IntechOpen

Artificial Neural Networks

Industrial and Control Engineering Applications

Edited by Kenji Suzuki

ARTIFICIAL NEURAL NETWORKS - INDUSTRIAL AND CONTROL ENGINEERING APPLICATIONS

Edited by **Kenji Suzuki**

Artificial Neural Networks - Industrial and Control Engineering Applications
http://dx.doi.org/10.5772/2041
Edited by Kenji Suzuki

Contributors

Mustafa Demetgul, Sezai Taskin, Ibrahim Nur Tansel, Alexander Koujelev, Siu-Lung Lui, Ali Taqi Hasan, Hayder M A A Al-Assadi, Ahmad Azlan Mat Isa, Maja Prevolnik, Marjana Novič, Dejan Škorjanc, Marjeta Čandek-Potokar, Jan Mareš, Petr Dolezel, Pavel Hrncirik, Bhaskar Prasad Rimal, Agus Budiyono, Dugki Min, Eunmi Choi, Idris E. Putro, Klemen Deželak, Gorazd Štumberger, Drago Dolinar, Beno Klopčič, Chang-Ching David Lin, Saifulnizam Abd. Khalid, Mohd Wazir Mustafa, Hussain Shareef, Azhar Khairuddin, Jiamei Deng, Richard Stobart, Bastian Maass, Esmaeil Jorjani, Farid Rafezi, Shahab Karimi, Mahmood Joorabian, Mehdi Baghdadi, Iman Sadinezhad, Mohammad Reza Toroghinejad, Mohsen Botlani Esfahani, Zahari Taha, Sarkawt Rostam Hassan, Chi Leung Patrick Hui, Sau Fun Ng, Connie Ip, Chang Hao Piao, Haider A.F. Almurib, Hayder M. A. Ali Al-Assadi, Pratima Shastri, Seyed Hosein Sadati, Javad Alizadeh Kaklar, Rahmatollah Ghajar, Mohammad Amani Tehran, Mahboubeh Maleki, Sanjoy Debnath, Chonghai Xu, Jingjie Zhang, Mingdong Yi

Notice

First published in Croatia, 2011 by INTECH d.o.o.
eBook (PDF) Published by IN TECH d.o.o.
Place and year of publication of eBook (PDF): Rijeka, 2019. IntechOpen is the global imprint of IN TECH d.o.o.
Printed in Croatia

Legal deposit, Croatia: National and University Library in Zagreb

Additional hard and PDF copies can be obtained from orders@intechopen.com

Artificial Neural Networks - Industrial and Control Engineering Applications
Edited by Kenji Suzuki

p. cm.

ISBN 978-953-307-220-3

eBook (PDF) ISBN 978-953-51-4505-9

Meet the editor

Kenji Suzuki received his Ph.D. degree in information engineering from Nagoya University in 2001. From 1993 to 2001, he worked at Hitachi Medical Corporation, and then Aichi Prefectural University as faculty. In 2001, he joined Department of Radiology at University of Chicago. Since 2006, he has been Assistant Professor of Radiology, Medical Physics, and Cancer Research Center there. Dr. Suzuki' research interests include computer-aided diagnosis and machine learning. He has published more than 190 papers including 70 peer-reviewed journal papers. He has been serving as the Editor-in-Chief and an Associate Editor of 12 leading international journals. He has received Paul Hodges Award, three Certificate of Merit Awards and Research Trainee Prize from RSNA, Young Investigator Award from Cancer Research Foundation, and Honorable Mention Poster Award at SPIE International Symposium on Medical Imaging. He has been a Senior Member of IEEE since 2004.

Contents

Preface

Artificial neural networks may probably be the single most successful technology in the last two decades which has been widely used in a large variety of applications in various areas. An artificial neural network, often just called a neural network, is a mathematical (or computational) model that is inspired by the structure and function of biological neural networks in the brain. An artificial neural network consists of a number of artificial neurons (i.e., nonlinear processing units) which are connected each other via synaptic weights (or simply just weights). An artificial neural network can "learn" a task by adjusting weights. There are supervised and unsupervised models. A supervised model requires a "teacher" or desired (ideal) output to learn a task. An unsupervised model does not require a "teacher," but it leans a task based on a cost function associated with the task. An artificial neural network is a powerful, versatile tool. Artificial neural networks have been successfully used in various applications such as biological, medical, industrial, control engendering, software engineering, environmental, economical, and social applications. The high versatility of artificial neural networks comes from its high capability and learning function. It has been theoretically proved that an artificial neural network can approximate any continuous mapping by arbitrary precision. Desired continuous mapping or a desired task is acquired in an artificial neural network by learning.

The purpose of this book series is to provide recent advances of artificial neural network applications in a wide range of areas. The series consists of two volumes: the first volume contains methodological advances and biomedical applications of artificial neural networks; the second volume contains artificial neural network applications in industrial and control engineering.

This second volume begins with a part of artificial neural network applications in textile industries which are concerned with the design and manufacture of clothing as well as the distribution and use of textiles. The part contains a review of various applications of artificial neural networks in textile and clothing industries as well as particular applications. A part of materials science and industry follows. This part contains applications of artificial neural networks in material identification, and estimation of material property, behavior, and state. Parts continue with food industry such as meat, electric and power industry such as batteries, power systems, and power allocation systems, mechanical engineering such as engines and machines, control and robotic engineering such as nonlinear system control, induction motors, system identification, signal and fault diagnosis systems, and robot manipulation.

Thus, this book will be a fundamental source of recent advances and applications of artificial neural networks in industrial and control engineering areas. The target audience of this book includes professors, college students, and graduate students in engineering schools, and engineers and researchers in industries. I hope this book will be a useful source for readers and inspire them.

Kenji Suzuki, Ph.D.
University of Chicago
Chicago, Illinois,
USA

Part 1

Textile Industry

1

Review of Application of Artificial Neural Networks in Textiles and Clothing Industries over Last Decades

Chi Leung Parick Hui, Ng Sau Fun and Connie Ip
Institute of Textiles and Clothing, The Hong Kong Polytechnic University, Hong Kong SAR, PRC. China

1. Introduction

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurones) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurones. The ANN has recently been applied in process control, identification, diagnostics, character recognition, sensory prediction, robot vision, and forecasting.

In Textiles and Clothing industries, it involves the interaction of a large number of variables. Because of the high degree of variability in raw materials, multistage processing and a lack of precise control on process parameters, the relation between such variables and the product properties is relied on the human knowledge but it is not possible for human being to remember all the details of the process-related data over the years. As the computing power has substantially improved over last decade, the ANN is able to learn such datasets to reveal the unknown relation between various variables effectively. Therefore, the application of ANN is more widespread in textiles and clothing industries over last decade.

In this chapter, it aims to review current application of ANN in textiles and clothing industries over last decade. Based on literature reviews, the challenges encountered by ANN used in the industries will be discussed and the potential future application of ANN in the industries will also be addressed. The structure of this chapter comprises of seven sections. The first section includes background of ANN, importance of ANN in textiles and clothing and the arrangement of this chapter. In forthcoming three sections, they include review of applications of ANN in fibres and yarns, in chemical processing, and in clothing over last decade. Afterwards, challenges encountered by ANN used in textiles and clothing industries will be discussed and potential future application of ANN in textiles and clothing industries will be addressed in last section.

2. Applications to fibres and yarns

2.1 Fibre classification

Kang and Kim (2002) developed an image system for the current cotton grading system of raw cotton involving a trained artificial neural network with a good classifying ability. Trash from a raw cotton image can be characterized by a captured color by a color CCD camera and acquire color parameters. The number of trash particles and their content, size, size distribution, and spatial density can be evaluated after raw cotton images of the physical standards are thresholded and connectivity was checked. The color grading of raw cotton can be influenced by trash. Therefore, the effect of trash on color grading was investigated using a color difference equation that measured the color difference between a trash-containing image and a trash-removed image. The artificial neural network, which has eight color parameters as input data, was a highly reliable and useful tool for classifying color grades automatically and objectively.

She *et al.*, (2002) developed an intelligent system using artificial neural networks (ANN) and image processing to classify two kinds of animal fibres objectively between merino and mohair; which are determined in accordance with the complexity of the scale structure and the accuracy of the model. An unsupervised artificial neural network was used to extract eighty, fifty, and twenty implicit features automatically while image processing technique was used to extract nine explicit features. Then the supervised ANN was employed to classify these fibers, based on the features extracted with image processing and unsupervised artificial neural networks. The classification with features extracted explicitly by image processing is more accurate than with features from unsupervised artificial neural networks but it required more effort for image processing and more prior knowledge. On the contrary, the classification with combined unsupervised and supervised ANN was more robust because it needed only raw images, limited image processing and prior knowledge. Since only ordinary optical images taken with a microscope were employed, this approach for many textile applications without expensive equipment such as scanning electron microscopy can be developed.

Durand *et al.*, (2007) studied different approaches for variable selection in the context of near-infrared (NIR) multivariate calibration of the cotton–viscose textiles composition. First, a model-based regression method was proposed. It consisted of genetic algorithm optimization combined with partial least squares regression (GA–PLS). The second approach was a relevance measure of spectral variables based on mutual information (MI), which can be performed independently of any given regression model. As MI made no assumption on the relationship between X and Y, non-linear methods such as feed-forward artificial neural network (ANN) were thus encouraged for modeling in a prediction context (MI–ANN). GA–PLS and MI–ANN models were developed for NIR quantitative prediction of cotton content in cotton–viscose textile samples. The results were compared to full spectrum (480 variables) PLS model (FS-PLS). The model required 11 latent variables and yielded a 3.74% RMS prediction error in the range 0–100%. GA–PLS provided more robust model based on 120 variables and slightly enhanced prediction performance (3.44% RMS error). Considering MI variable selection procedure, great improvement can be obtained as 12 variables only were retained. On the basis of these variables, a 12 inputs of ANN model was trained and the corresponding prediction error was 3.43% RMS error.

2.2 Yarn manufacture

Beltran *et al.*, (2004) developed an artificial neural network (ANN) trained with back-propagation encompassed all known processing variables that existed in different

spinning mills, and then generalized this information to accurately predict yarn quality of worsted spinning performance for an individual mill. The ANN was then subsequently trained with commercial mill data to assess the feasibility of the method as a mill-specific performance prediction tool. The ANN was a suitable tool for predicting worsted yarn quality for a specific mill.

Farooq and Cherif (2008) have reported a method of predicting the leveling action point, which was one of the important auto-leveling parameters of the drawing frame and strongly influences the quality of the manufactured yarn, by using artificial neural networks (ANN). Various leveling action point affecting variables were selected as inputs for training the artificial neural networks, which was aimed to optimize the auto-leveling by limiting the leveling action point search range. The Levenberg–Marquardt algorithm was incorporated into the back-propagation to accelerate the training and Bayesian regularization was applied to improve the generalization of the networks. The results obtained were quite promising that the accuracy in computation can lead to better sliver CV% and better yarn quality.

2.3 Yarn-property prediction

Kuo *et al*., (2004) applied neural network theory to consider the extruder screw speed, gear pump gear speed, and winder winding speed of a melt spinning system as the inputs and the tensile strength and yarn count of spun fibers as the outputs. The data from the experiments were used as learning information for the neural network to establish a reliable prediction model that can be applied to new projects. The neural network model can predict the tensile strength and yarn count of spun fibers so that it can provide a very good and reliable reference for spun fiber processing.

Zeng *et al.*, (2004) tried to predict the tensile properties (yarn tenacity) of air-jet spun yarns produced from 75/25 polyester on an air-jet spintester by two models, namely neural network model and numerical simulation. Fifty tests were undergone to obtain average yarn tenacity values for each sample. A neural network model provided quantitative predictions of yarn tenacity by using the following parameters as inputs: first and second nozzle pressures, spinning speed, distance between front roller nip and first nozzle inlet, and the position of the jet orifice in the first nozzle so that the effects of parameters on yarn tenacity can be determined. Meanwhile, a numerical simulation provided a useful insight into the flow characteristics and wrapping formation process of edge fibers in the nozzle of an air-jet spinning machine; hence, the effects of nozzle parameters on yarn tensile properties can be predicted. The result showed that excellent agreement was obtained between these two methods. Moreover, the predicted and experimental values agreed well to indicate that the neural network was an excellent method for predictors.

Lin (2007) studied the shrinkages of warp and weft yarns of 26 woven fabrics manufactured by air jet loom by using neural net model which were used to determine the relationships between the shrinkage of yarns and the cover factors of yarns and fabrics. The shrinkages were affected by various factors such as loom setting, fabric type, and the properties of warp and weft yarns. The neural net was trained with 13 experimental data points. A test on 13 data points showed that the mean errors between the known output values and the output values calculated using the neural net were only 0.0090 and 0.0059 for the shrinkage ratio of warp ($S1$) and weft ($S2$) yarn, respectively. There was a close match between the actual and predicted shrinkage of the warp (weft) yarn. The test results gave R^2 values of 0.85 and 0.87 for the shrinkage of the warp (i.e., $S1$) and weft (i.e., $S2$), respectively. This showed that the

neural net produced good results for predicting the shrinkage of yarns in woven fabrics. Different woven fabrics manufactured on different looms like rapier, gripper, etc., raw material yarn ingredients (e.g., T/C x T/R, T/R x T/R, T/C x C, etc.), and fabric structural class (e.g., twill, satin, etc.) were examined to measure the shrinkage ratio of warp and weft yarns. The developed neural net model was then used to train the obtained data and the result showed that the prediction of yarn shrinkage in the off-loomed fabrics can be fulfilled through a prediction model constructed with neural net.

Xu *et al.*, (2007) studied a neural network method of analyzing cross-sectional images of a wool/silk blended yarn. The process of original yarn cross-sectional images including image enhancement and shape filtering; and the determination of characteristic parameters for distinguishing wool and silk fibers in the enhanced yarn cross-sectional images were in the study. A neural network computing approach, single-layer perceptrons, was used for learning the target parameters. The neural network model had a good capability of tolerance and learning. The study indicated that preparation of the yarn sample slices was critically important to obtain undistorted fiber images and to ensure the accuracy of fiber recognition. The overall error estimated for recognizing wool or silk fiber was 5%.

Khan *et al.*, (2009) studied the performance of multilayer perceptron (MLP) and multivariate linear regression (MLR) models for predicting the hairiness of worsted-spun wool yarns objectively by examining 75 sets of yarns consisting of various top specifications and processing parameters of shrink-resist treated, single-ply, pure wool worsted yarns. The results indicated that the MLP model predicted yarn hairiness was more accurately than the MLR model and showed that a degree of nonlinearity existed in the relationship between yarn hairiness and the input factors considered. Therefore, the artificial neural network (ANN) model had the potential for wide mill specific applications for high precision prediction of hairiness of a yarn from limited top, yarn and processing parameters. The use of the ANN model as an analytical tool may facilitate the improvement of current products by offering alternative material specification and/or selection and improved processing parameters governed by the predicted outcomes of the model. On sensitivity analysis on the MLP model, yarn twist, ring size, average fiber length (hauteur) had the greatest effect on yarn hairiness with twist having the greatest impact on yarn hairiness.

Ünal *et al.*, (2010) investigated the retained spliced diameter with regard to splicing parameters and fiber and yarn properties. The yarns were produced from eight different cotton types in three yarn counts (29.5, 19.7 and 14.8 tex) and three different twist coefficients (α_{Tex} 3653, α_{Tex} 4038, α_{Tex} 4423). To investigate the effects of splicing parameters on the retained spliced diameter, opening air pressure, splicing air pressure and splicing air time were set according to an orthogonal experimental design. The retained spliced diameter was calculated and predicted by using an artificial neural network (ANN) and response surface methods. Analyses showed that ANN models were more powerful compared with response surface models in predicting the retained spliced diameter of ring spun cotton yarns.

2.4 Fibre and Yarn relationship

Admuthe and Apte (2010) used multiple regression model such as artificial neural network (ANN) in an attempt to develop the relationship between fiber and yarn in the spinning process. 30 different cotton fibres were selected covering all of the staple length groups of cotton grown in India. The yarn (output) produced from the spinning process had a unique

relationship with the fibers (input). However, ANN failed to develop exact relationships between the fiber and the yarn, then a hybrid approach was used to achieving the solution. Hence, a new hybrid technique, Adaptive Neuro-Fuzzy Inference System (ANFIS) which was combined with subtractive clustering was used to predict yarn properties. The result shown that the ANFIS gave better co-relation values. The test results show better accuracy for all datasets when compared it to the ANN model.

3. Applications to fabrics

3.1 Fabric manufacture

Yao *et al.*, (2005) investigated the predictability of the warp breakage rate from a sizing yarn quality index using a feed-forward back-propagation network in an artificial neural network system. An eight-quality index (size add-on, abrasion resistance, abrasion resistance irregularity, hairiness beyond 3 mm, breaking strength, breaking strength irregularity, breaking elongation, and breaking elongation irregularity) and warp breakage rates were rated in controlled conditions. A good correlation between predicted and actual warp breakage rates indicated that warp breakage rates can be predicted by neural networks. A model with a single sigmoid hidden layer with four neurons was able to produce better predictions than the other models of this particular data set in the study.

Behera and Karthikeyan (2006) described the method of applying artificial NNs for the prediction of both construction and performance parameters of canopy fabrics. Based on the influence on the performance of the canopy fabric, constructional parameters were chosen. Constructional parameters were used as input for predicting the performance parameter in forward engineering, and the parameters were reversed for the reverse engineering prediction. Comparison between actual results and predicted results was made. The results of the design prediction had excellent correlation with all the samples.

Behera and Goyal (2009) described the method of applying the artificial neural network for the prediction performance parameters for airbag fabrics. The results of the ANN performance prediction had low prediction error of 12% with all the samples and the artificial neural network based on Error Back-propagation were found promising for a new domain of design prediction technique. The prediction performance of the neural network was based on the amount of training. The diversity of the data and the amount of data resulted in better the mapping of the network, and better predictions. Therefore, airbag fabrics could be successfully engineered using artificial neural network.

3.2 Fabric-property prediction

Ertugrul and Ucar (2000) have shown how the bursting strength of cotton plain knitted fabrics can be predicted before manufacturing by using intelligent techniques of neural network and neuro-fuzzy approaches. Fabric bursting strength affected by fabric weight, yarn breaking strength, and yarn breaking elongation were input elements for the predictions. Both the multi-layer feed-forward neural network and adaptive network based fuzzy inference system, a combination of a radial basis neural network and the Sugeno-Takagi fuzzy system, were studied. Both systems had the ability to learn training data successfully, and testing errors can give an approximate knowledge of the bursting strength which fabric can be knitted.

Chen *et al.*, (2001) proposed a neural network computing technique to predict fabric end-use. One hundred samples of apparel fabrics for suiting, shirting, and blouse uses were selected and fabric properties of extension, shear, bending, compression, and friction and

roughness were measured by using the Kawabata KES instruments. Instrumental data of the fabric properties and information on fabric end-uses were input into neural network software to train a multilayer perceptron model. The prediction error rate from the established neural network model was estimated by using a cross-validation method. The estimated error rate for prediction was 0.07. The established neural network model could be upgraded by inputting new fabric samples and be implemented for applications in garment design and manufacture.

Shyr *et al.*, (2004) have taken new approaches in using a one-step transformation process to establish translation equations for total hand evaluations of fabrics by employing a stepwise regression method and an artificial neural network. The key mechanical properties selected from sixteen fabric mechanical properties based on a KES system, using the stepwise regression selection method, were the parameters. The translation equations were developed directly with parameters without a primary hand value transformation process. 114 polyester/cotton blended woven fabrics were selected for investigation. Four mechanical properties LC, 2HG, B, and WT were the parameters for developing the translation equations. The correlation coefficients of the translation equations developed from the stepwise regression and artificial neural network methods were 0.925 and 0.955, respectively. Both translation equations had high correlation coefficients between the calculated and practical values. The approaches were identified effectively to develop translation equations for new fabrics in the textile industry.

Behera and Mishra (2007) investigated the prediction of non-linear relations of functional and aesthetic properties of worsted suiting fabrics for fabric development by an engineered approach of a radial basis function network which was trained with worsted fabric constructional parameters. Therefore, an objective method of fabric appearance evaluation with the help of digital image processing was introduced. The radial basis function network can successfully predict the fabric functional and aesthetic properties from basic fibre characteristics and fabric constructional parameters with considerable accuracy. The network prediction was in good correlation with the actual experimental data. There was some error in predicting the fabric properties from the constructional parameters. The variation in the actual values and predicted values was due to the small sample size. Moreover, the properties of worsted fabrics were greatly influenced by the finishing parameters which are not taken into consideration in the training of the network.

Murrells *et al.*, (2009) employed an artificial neural network (ANN) model and a standard multiple linear regression model for the prediction of the degree of spirality of single jersey fabrics made from a total of 66 fabric samples produced from three types of 100% cotton yarn samples including conventional ring yarns, low torque ring yarns and plied yarns. The data were randomly divided into 53 and 13 sets of data that were used for training and evaluating the performance of the predictive models. A statistical analysis was undertaken to check the validity by comparing the results obtained from the two types of model with relatively good agreement between predictions and actual measured values of fabric spirality with a correlation coefficient, R, of 0.976 in out-of-sample testing. Therefore, the results demonstrated that the neural network model produced superior results to predict the degree of fabric spirality after three washing and drying cycles. Both the ANN and the regression approach showed that twist liveliness, tightness factor and yarn linear density were the most important factors in predicting fabric spirality. Twist liveliness was the major contributor to spirality with the other factors such as yarn type, the number of feeders,

rotational direction and gauge (needles/inch) of the knitting machine and dyeing method having a minor influence.

Hadizadeh *et al.*, (2009) used an ANN model for predicting initial load-extension behavior (Young's modulus) in the warp and weft directions of plain weave and plain weave derivative fabrics by modeling the relationship between a combination of the yarn modular length, yarn spacing, the ratio of occupied spacing to total length of yarn in one weave repeat, and the yarn flexural rigidity with satisfactory accuracy. A single hidden layer feed-forward ANN based on a back-propagation algorithm with four input neurons and one output neuron was developed to predict initial modulus in the warp and weft directions. Input values were defined as combination expressions of geometrical parameters of fabric and yarn flexural rigidity, which were obtained from Leaf's mathematical model. Data were divided into two groups as training and test sets. A very good agreement between the examined and predicted values was achieved and the model's suitability was confirmed by the low performance factor (PF/3) and the high coefficient of correlation.

Hadizadeh *et al.*, (2010) introduced a new model based on an adaptive neuro-fuzzy inference system (ANFIS) for predicting initial load–extension behavior of plain-woven fabrics. Input values defined as combination expressions of geometrical parameters of fabric and yarn flexural rigidity, yarn-spacing, weave angle and yarn modular length, which were extracted from Leaf's mathematical model. The results showed that the neuro-fuzzy system can be used for modeling initial modulus in the warp and weft directions of plain-woven fabrics. Outputs of the neuro-fuzzy model were also compared with results obtained by Leaf's models. The calculated results were in good agreement with the real data upon finding the importance of inputs.

3.3 Fabric defect

Hu and Tsai (2000) used best wavelet packet bases and an artificial neural network (ANN) to inspect four kinds of fabric defects. Multi-resolution representation of an image using wavelet transform was a new and effective approach for analyzing image information content. The values and positions for the smallest-six entropy were found in a wavelet packet best tree that acted as the feature parameters of the ANN for identifying fabric defects. They explored three basic considerations of the classification rate of fabric defect inspection comprising wavelets with various maximum vanishing moments, different numbers of resolution levels, and differently scaled fabric images. The results showed that the total classification rate for a wavelet function with a maximum vanishing moment of four and three resolution levels can reach 100%, and differently scaled fabric images had no obvious effect on the classification rate.

Shiau *et al.*, (2000) constructed a back-propagation neural network topology to automatically recognize neps and trash in a web by color image processing. The ideal background color under moderate conditions of brightness and contrast to overcome the translucent problem of fibers in a web, specimens were reproduced in a color BMP image file format. With a back-propagation neural network, the RGB (red, green, and blue) values corresponding with the image pixels were used to perform the recognition, and three categories (i.e., normal web, nep, and trash) can be recognized to determine the numbers and areas of both neps and trash. According to experimental analysis, the recognition rate can reach 99.63% under circumstances in which the neural network topology is 3-3-3. Both contrast and brightness were set at 60% with an azure background color. The results showed that both neps and

trash can be recognized well, and the method was suitable not only for cotton and man-made fibers of different lengths, but also for different web thicknesses as to a limit of 32.9 g/m^2.

Choi *et al.*, (2001) developed a new method for a fabric defect identifying system by using fuzzy inference in multi-conditions. The system has applied fuzzy inference rules, and the membership function for these rules to adopt a neural network approach. Only a small number of fuzzy inference rules were required to make the identifications of non-defect, slub (warp direction), slub (weft direction), nep, and composite defect. One fuzzy inference rule can replace many crisp rules. This system can be used to design a reliable system for identifying fabric defects. Experimental results with this approach have demonstrated the identification ability which was comparable to that of a human inspector.

Huang and Chen (2001) investigated an image classification by a neural-fuzzy system for normal fabrics and eight kinds of fabric defects. This system combined the fuzzification technique with fuzzy logic and a back-propagation learning algorithm with neural networks. Four inputs featured the ratio of projection lengths in the horizontal and vertical directions, the gray-level mean and standard deviation of the image, and the large number emphasis (LNE) based on the neighboring gray level dependence matrix for the defect area. The neural network was also implemented and compared with the neural-fuzzy system. The results demonstrated that the neural-fuzzy system was superior to the neural network in classification ability.

Saeidi *et al.*, (2005) described a computer vision-based fabric inspection system implemented on a circular knitting machine to inspect the fabric under construction. The detection of defects in knitted fabric was performed and the performance of three different spectral methods, namely, the discrete Fourier transform, the wavelet and the Gabor transforms were evaluated off-line. Knitted fabric defect-detection and classification was implemented on-line. The captured images were subjected to a defect-detection algorithm, which was based on the concepts of the Gabor wavelet transform, and a neural network as a classifier. An operator encountering defects also evaluated the performance of the system. The fabric images were broadly classified into seven main categories as well as seven combined defects. The results of the designed system were compared with those of human vision.

Shady *et al.*, (2006) developed a new method for knitted fabric defect detection and classification using image analysis and neural networks. Images of six different induced defects (broken needle, fly, hole, barré, thick and thin yarn) were used in the analysis. Statistical procedures and Fourier Transforms were utilized in the feature extraction effort and neural networks were used to detect and classify the defects. The results showed success in classifying most of the defects but the classification results for the barré defect were not identified using either approach due to the nature of the defect shape which caused it to interfere with other defects such as thick/thin yarn defects. The results of using the Fourier Transform features extraction approach were slightly more successful than the statistical approach in detecting the free defect and classifying most of the other defects.

Yuen *et al.*, (2009) explored a novel method to detect the fabric defect automatically with a segmented window technique which was presented to segment an image for a three layer BP neural network to classify fabric stitching defects. This method was specifically designed for evaluating fabric stitches or seams of semi-finished and finished garments.

A fabric stitching inspection method was proposed for knitted fabric in which a segmented window technique was developed to segment images into three classes using a

monochrome single-loop ribwork of knitted fabric: (1) seams without sewing defects; (2) seams with pleated defects; and (3) seams with puckering defects caused by stitching faults. Nine characteristic variables were obtained from the segmented images and input into a Back Propagation (BP) neural network for classification and object recognition. The classification results demonstrated that the inspection method developed was effective in identifying the three classes of knitted-fabric stitching. It was proved that the classifier with nine characteristic variables outperformed those with five and seven variables and the neural network technique using either BP or radial basis (RB) was effective for classifying the fabric stitching defects. By using the BP neural network, the recognition rate was 100%. The experiment results showed that the method developed in this study is feasible and applicable.

3.4 Sewing

Jeong *et al.*, (2000) constructed a neural network and subjoined local approximation technique for application to the sewing process by selecting optimal interlinings for woolen fabrics. Men's woolen suitings and ten optimal interlinings were selected and matched. A single hidden layer neural network was constructed with five input nodes, ten hidden nodes, and two output nodes. Both input and output of the mechanical parameters measured on the KES-FB system were used to train the network with a back-propagation learning algorithm. The inputs for the fabrics were tensile energy, bending rigidity, bending hysteresis, shear stiffness, and shear hysteresis, while outputs for the interlinings were bending rigidity and shear stiffness. This research presented a few methods for improving the efficiency of the learning process. The raw data from the KES-FB system were nonlinearly normalized, and input orders were randomized. The procedure produced a good result because the selection agreed well with the experts' selections. Consequently, the results showed that the neural network and subjoined techniques had a strong potential for selecting optimum interlinings for woolen fabrics.

Hui *et al.*, (2007) investigated the use of artificial neural networks (ANN) to predict the sewing performance of woven fabrics for efficient planning and control for the sewing operation. This was based on the physical and mechanical properties of fabrics such as the critical parameters of a fabric constructional and behavioural pattern as all input units and to verify the ANN techniques as human decision in the prediction of sewing performance of fabrics by testing 109 data sets of fabrics through simple testing system and the sewing performance of each fabric's specimen by the domain experts. Among 109 input-output data pairs, 94 were used to train the proposed back-propagation (BP) neural network for the prediction of the unknown sewing performance of a given fabric, and 15 were used to test the proposed BP neural network. A three-layered BP neural network that consists of 21 input units, 21 hidden units, and 16 output units was developed. The output units of the model were the control levels of sewing performance in the areas of puckering, needle damages, distortion, and overfeeding. After 10,000 iterations of training of BP neural network, the neural network converged to the minimum error level. The evaluation of the model showed that the overall prediction accuracy of the developed BP model was at 93 per cent which was the same as the accuracy of prediction made by human assessment. The predicted values of most fabrics were found to be in good agreement with the results of sewing tests carried out by domain experts.

3.5 Seam performance

Hui and Ng (2009) investigated the capability of artificial neural networks based on a back propagation algorithm with weight decay technique and multiple logarithm regression (MLR) methods for modeling seam performance of fifty commercial woven fabrics used for the manufacture of men's and women's outerwear based on seam puckering, seam flotation and seam efficiency. The developed models were assessed by verifying Mean Square Error (MSE) and Correlation Coefficient (*R*-value) of test data prediction. The results indicated that the artificial neural network (ANN) model has better performance in comparison with the multiple logarithm regression model. The difference between the MSE of predicting in these two models for predicting seam puckering, seam flotation, and seam efficiency was 0.0394, 0.0096, and 0.0049, respectively. Thus, the ANN model was found to be more accurate than MLR, and the prediction errors of ANNs were low despite the availability of only a small training data set. However, the difference in prediction errors made by both models was not significantly high. It was found that MLR models were quicker to construct, more transparent, and less likely to overfit the minimal amount of data available. Therefore, both models were effectively predicting the seam performance of woven fabrics.

Onal *et al.*, (2009) studied the effect of factors on seam strength of webbings made from polyamide 6.6 which were used in parachute assemblies as reinforcing units for providing strength by using both Taguchi's design of experiment (TDOE) as well as an artificial neural network (ANN), then compared them with the strength physically obtained from mechanical tests on notched webbing specimens. It was established from these comparisons, in which the root mean square error was used as an accuracy measure, that the predictions by ANN were better predictions of the experimental seam strength of jointed notched webbing in accuracy than those predicted by TDOE. An L8 design was adopted and an orthogonal array was generated. The contribution of each factor to seam strength was analyzed using analysis of variance (ANOVA) and signal to noise ratio methods. From the analysis, the TDOE revealed (based on SNR performance criteria) that the fabric width, folding length of joint and interaction between the folding length of joint and the seam design affected seam strength significantly. An optimal configuration of levels of factors was found by using TDOE.

4. Applications to chemical processing

Huang and Yu (2001) used image processing and fuzzy neural network approaches to classify seven kinds of dyeing defects including filling band in shade, dye and carrier spots, mist, oil stain, tailing, listing, and uneven dyeing on selvage. The fuzzy neural classification system was constructed by a fuzzy expert system with the neural network as a fuzzy inference engine so it was more intelligent in handling pattern recognition and classification problems. The neural network was trained to become the inference engine using sample data. Region growing was adopted to directly detect different defect regions in an image. Seventy samples, ten samples for each defect, were obtained for training and testing. The results demonstrated that the fuzzy neural network approach could precisely classify the defective samples by the features selected.

5. Applications to clothing

5.1 Pattern fitting prediction

Hu *et al.*, (2009) developed a system to utilize the successful experiences and help the beginners of garment pattern design (GPD) through optimization methods by proposing a

hybrid system (NN-ICEA) based on neural network (NN) and immune co-evolutionary algorithm (ICEA) to predict the fit of the garments and search optimal sizes. ICEA takes NN as fitness function and procedures including clonal proliferation, hypermutation and co-evolution search the optimal size values. A series of experiments with a dataset of 450 pieces of pants was conducted to demonstrate the prediction and optimization capabilities of NN-ICEA. In the comparative studies, NN-ICEA was compared with NN-genetic algorithm to show the value of immune-inspired operators. Four types of GPD method have been summarized and compared. The research was a feasible and effective attempt aiming at a valuable problem and provides key algorithms for fit prediction and size optimization. The algorithms can be incorporated into garment computer-aided design system (CAD).

5.2 Clothing sensory comfort

Wong *et al.*, (2003) investigated the predictability of clothing sensory comfort from psychological perceptions by using a feed-forward back-propagation network in an artificial neural network (ANN) system. Wear trials were conducted ten sensory perceptions (clammy, clingy, damp, sticky, heavy, prickly, scratchy, fit, breathable, and thermal) and overall clothing comfort (comfort) which were rated by twenty-two professional athletes in a controlled laboratory. Four different garments in each trial and rate the sensations above during a 90-minute exercising period were scored as input into five different feed-forward back-propagation neural network models, consisting of six different numbers of hidden and output transfer neurons. The results showed a good correlation between predicted and actual comfort ratings with a significance of $p<0.001$. Good agreement between predicted and actual clothing comfort perceptions proved that the neural network was an effective technique for modeling the psychological perceptions of clothing sensory comfort. The predicted comfort score generated from the model with the log-sigmoid hidden neurons and the linear output neuron had a better fit with the actual comfort score than other models with different combinations of hidden and output neurons. Compared with statistical modeling techniques, the neural network was a fast, flexible, predictive tool with a self-learning ability for clothing comfort perceptions.

Wong *et al.*, (2004) investigated the process of human psychological perceptions of clothing related sensations and comfort to develop an intellectual understanding of and methodology for predicting clothing comfort performance from fabric physical properties. Various hybrid models were developed using different modeling techniques by studying human sensory perception and judgement processes. By combining the strengths of statistics (data reduction and information summation), a neural network (self-learning ability), and fuzzy logic (fuzzy reasoning ability), hybrid models were developed to simulate different stages of the perception process. Results showed that the TS-TS-NN-FL model had the highest ability to predict overall comfort performance from fabric physical properties. The three key elements in predicting psychological perceptions of clothing comfort from fabric physical properties were data reduction and summation, self-learning, and fuzzy reasoning. The model was shown that these three elements can generated the best predictions compared with other hybrid models.

All research outputs in application of ANN in textiles and clothing areas over last decade are summarized as shown in Appendix.

6. Challenges encountered by ANN used in textiles and clothing industries

In the application of ANN in different disciplines of textiles and clothing industries, there are the following limitations which has encountered.

Fibre classification: More powerful learning strategies are required to improve the classification accuracy made by the ANN.

Yarn manufacture: Additional work is needed to accurately model the occurrence of spinning ends-down and neps by using the ANN. To improve the predictions on such parameters, additional mill-specific data and further developments of the ANN simulations are necessary.

Yarn-property prediction: Some researchers reported that yarn tenacity decreases when spinning speed exceeds a certain value, say, 210 m/min. Since we used an air-jet spinstester in this research, spinning speed could not exceed 200 m/min because of the restriction of the machine, so the decreasing trend of yarn tenacity could not be predicted. In addition, the difficulty in developing a universal empirical model that can accurately predict yarn hairiness for different mills stems from the variability in processing methodologies and equipment.

As many independent variables exist, further difficulty arises in covering the entire range of parameters with the capability of interpolating and extrapolating experimental observations or mill measurements and to take into account the interactive contribution between each input factor. It is, therefore, desirable to possess the capacity to discover regularities directly from the data being modeled, that can dynamically evolve with time taking into account changes in materials' specifications and processing techniques within a given mill. The MLP model, one of ANN model does possess this characteristic and has the potential for wider applicability in industry.

Fabric manufacture: To improve the correlation between actual and predicted values, in the case of reverse engineering, constraints are posed to limit the ranges of constructional parameters in ANN.

Fabric-property prediction: Besides the possibility of trying different ANN configurations, the quantity and the quality of training data are also very important to the results. Even though we do not include coefficient of variation (CV) values in the training pairs as inputs, we have concluded that the ANN has a higher chance of giving big errors if the data include many training pairs with high CV values because they feed inconsistent information to the ANN. For future work, we suggest that there should be enough training pairs and the CV values of these data should also be known for higher reliability.

In addition, prediction performance can be further improved by including these parameters as input during the training phase. In few cases, the network has predicted contradictory trends which are found difficult to be explained. Also, the neural network model outperformed the multiple regression models in predicting the angle of spirality using data that were not used to train the network. This indicates that it is worthwhile using the more complex ANN technique if a large amount of different types of data are available.

Fabric defect: Since neps and trash in a web can be recognized, yarn quality is able to improve using a reference for adjusting manufacturing parameters. In addition, the **CCD** (charge coupled device) must be mounted, despite the scanner, because of on-line considerations. Patterned and complex fabrics can be inspected as well as plain fabrics. Further research such as a neuro-fuzzy expert system can identify actual defect types like reed marks, mispicks, pilling, finger marks, and others.

Since this research is limited by the speed of the knitting machine, further studies are required to inspect the fabric defects in higher speed, circular knitting machines. Application of ANN in fabric defect is still needed to be done in two major aspects: (1) the applicability of the developed method in studying other manufacturing defects needs to be validated; and (2) the current 2-D-based investigation needs to be extended to three-dimensional (3-D) space for actual manual inspection.

Seam performance: In these comparisons, RMSE values were used as comparative metrics. As a result, it can be said that ANN appears to be a reliable and useful tool in characterizing the effect of some critical manufacturing parameters on the seam strength of webbing if a sufficient number of replicated experimental data are available to train the ANN.

Applications to Chemical Processing: Fuzzification maps the input feature value to fuzzy sets and the dimensions of the feature space are increased. When fuzzy sets are appropriately chosen, they can increase the separated ability of classes in the feature space. This allows the fuzzy neural network model to fit input-output data more accurately with enhanced classification ability.

Pattern fitting prediction: The current scale is definitely not enough to study all sizes of the garment. In order to present the fuzzy and stochastic nature of the garment and body sizes, it should be modeled as fuzzy vector or stochastic vector. In addition, it is valuable to incorporate NN-ICEA into garment CAD system and thus the 2D and 3D effects of garments can provide intuitive impressions.

Clothing sensory comfort: The functions and interrelationships of individual sensory perceptions and comfort are unknown. It is difficult to learn their relationships using ANN.

In conclusion, the major challenges of using ANN in textiles and clothing industries are lack of sufficient data for learning and long computational time required for handling a large size of dataset. To improve the performance of ANN models, some major factors shall be considered to include the determination of adequate model inputs, data division and preprocessing, choice of suitable network architecture, careful selection of some internal parameters that control the optimization method, stopping criteria, and model validation.

7. Potential future application of ANN in textiles and clothing industries

A large number of applications of ANN in textiles and clothing industries are used feedforward and Kohonen networks. The other types of artificial neural networks such as recurrent neural network, associative neural network and dynamic neural networks (refer to http://en.wikipedia.org/wiki/Types_of_artificial_neural_networks website) are rarely used. Meanwhile, quite a few areas remains insufficiently explored such as knitting, nonwoven fabrics and finishing control. Exploring such areas using new ANN models is a new trend in future research.

In the future research, the following issues shall be taken into consideration to the application of ANN in textiles and clothing industries.

a. improve the data collection method for training ANNs such as online data captured from the process
b. improve the feature-extraction procedures before the data can be fed to an ANN
c. improve extrapolation ability of the system to strengthen the prediction capability
d. improve the user-friendly interface between user and machine

These issues are important for further development of using ANN in textiles and clothing industries. Further research works shall deal with such issues in order to set up intelligent systems in textiles and clothing fields instead of human judgment.

8. Appendix

Study Area		No	Title	Author	Journal	Year	Vol(No),pp.	Findings	Limitations
2. Fibres and Yarn	2.1 Fibre classification	1	Objective Evaluation of the Trash and Color of Raw Cotton by Image Processing and Neural Network	Kang and Kim	Textile Research Journal	2002	72(9), 776-782.	an image system for the current cotton grading system of raw cotton	/
		2	Intelligent Animal Fiber Classification with Artificial Neural Networks	She et al.	Textile Research Journal	2002	72(7), 594-600.	classify two kinds of animal fibres objectively between merino and mohair	The classification accuracy of ANN will be improved by developing more powerful learning strategies.
		3	Genetic algorithm optimisation combined with partial least squares regression and mutual information variable selection procedures in near-infrared quantitative analysis of cotton–viscose textiles	Durand et al.	Analytica Chimica Acta	2007	595(1-2), 72-79.	variable selection in the context of near-infrared (NIR) multivariate calibration of the cotton–viscose textiles composition	/
	2.2 Yarn manufacture	4	Predicting Worsted Spinning Performance with an Artificial Neural Network Model	Beltran et al.	Textile Research Journal	2004	74(9), 757-763.	predict yarn quality of worsted spinning performance for an individual mill	Additional work is needed to accurately model the occurrence of spinning ends-down and neps

Study Area		No	Title	Author	Journal	Year	Vol(No),pp.	Findings	Limitations
		5	Use of Artificial Neural Networks for Determining the Leveling Action Point at the Auto-leveling Draw Frame	Farooq and Cherif	Textile Research Journal	2008	78(6), 502-509.	predicting the leveling action point, which was one of the important auto-leveling parameters of the drawing frame and strongly influences the quality of the manufactured yarn	/
	2.3 Yarn-property prediction	6	Using Neural Network Theory to Predict the Properties of Melt Spun Fibers	Kuo et al.	Textile Research Journal	2004	74(9), 840-843.	establish a reliable prediction model for tensile strength and yarn count of as-spun fibers so that it can provide a very good and reliable reference for as-spun fiber processing	/
		7	Predicting the Tensile Properties of Air-Jet Spun Yarns	Zeng et al.	Textile Research Journal	2004	74(8), 689-694.	predict the tensile properties (yam tenacity) of air-jet spun yarns	spinning speed could not exceed 200 m/min because of the

Study Area		No	Title	Author	Journal	Year	Vol(No),pp.	Findings	Limitations
								produced from 75/25 polyester on an air-jet spintester by two models, namely neural network model and numerical simulation	restriction of the machine, so the decreasing trend of yarn tenacity could not be predicted.
		8	Prediction of Yarn Shrinkage using Neural Nets	Lin	Textile Research Journal	2007	77(5), 336-342.	the shrinkages of warp and weft yarns of 26 woven fabrics manufactured by air jet loom by using neural net model which were used to determine the relationships between the shrinkage of yarns and the cover factors of yarns and fabrics	/
		9	Neural Network Technique for Fiber Image Recognition	Xu et al.	Journal of Industrial Textiles	2007	36(4), 329-336.	analyzing cross-sectional images of a wool/silk blended yarn	/

Study Area		No	Title	Author	Journal	Year	Vol(No),pp.	Findings	Limitations
		10	An Artificial Neural Network-based Hairiness Prediction Model for Worsted Wool Yarns	Khan et al.	Textile Research Journal	2009	79(8), 714-720	the performance of multilayer perceptron (MLP) and multivariate linear regression (MLR) models for predicting the hairiness of worsted-spun wool yarns objectively	The difficulty in developing a universal empirical model that can accurately predict yarn hairiness for different mills stems from the variability in processing methodologies and equipment. As many independent variables exist, further difficulty arises in covering the entire range of parameters with the capability of interpolating and extrapolating experimental observations or mill measurements and to take into account the interactive contribution between each

Study Area		No	Title	Author	Journal	Year	Vol(No),pp.	Findings	Limitations
									input factor. It is, therefore, desirable to possess the capacity to discover regularities directly from the data being modeled, that can dynamically evolve with time taking into account changes in materials' specifications and processing techniques within a given mill. The MLP model does possess this characteristic and has the potential for wider applicability in industry.
		11	The Effect of Fiber Properties on the Characteristics of Spliced Yarns: Part II: Prediction of Retained Spliced Diameter	Ünal et al.	Textile Research Journal	2010	0(0), 1-8.	the retained spliced diameter with regard to splicing parameters and fiber and yarn properties	/

Study Area		No	Title	Author	Journal	Year	Vol(No),pp.	Findings	Limitations
	2.4 Fibre and Yarn relationship	12	Adaptive Neuro-fuzzy Inference System with Subtractive Clustering: A Model to Predict Fiber and Yarn Relationship	Admuthe and Apte	Textile Research Journal	2010	80(9), 841-846.	to develop the relationship between fiber and yarn in the spinning process	/
3. Applications to Fabrics	3.1 Fabric manufacture	13	Predicting the Warp Breakage Rate in Weaving by Neural Network Techniques	Yao et al.	Textile Research Journal	2005	75(3), 274-278	the predictability of the warp breakage rate from a sizing yarn quality index using a feed-forward back-propagation network in an artificial neural network system	/
		14	Artificial Neural Network-embedded Expert System for the Design of Canopy Fabrics	Behera and Karthikeyan	Journal of Industrial Textiles	2006	36(2), 111-123.	prediction of both construction and performance parameters of canopy fabrics	To improve the correlation between actual and predicted values, in the case of reverse engineering, constraints are posed to limit the ranges of construc-tional parameters.
		15	Artificial Neural Network System for the Design of Airbag Fabrics	Behera and Goyal	Journal of Indu-strial Textiles	2009	39(1), 45-55.	the prediction performance parameters for airbag fabrics	/

Study Area		No	Title	Author	Journal	Year	Vol(No),pp.	Findings	Limitations
	3.2 Fabric-property prediction	16	Predicting Bursting Strength of Cotton Plain Knitted Fabrics Using Intelligent Techniques	Ertugrul and Ucar	Textile Research Journal	2000	70(10), 845-851.	how the bursting strength of cotton plain knitted fabrics can be predicted before manufacturing by using intelligent techniques of neural network and neuro-fuzzy approaches	Besides the possibility of trying different NN configurations, the quantity and the quality of training data are also very important to the results. Even though we do not include cv values in the training pairs as inputs,' we have concluded that the NN has a higher chance of giving big errors if the data include many training pairs with high cv values because they feed inconsistent information to the NN. For future work, we suggest that there should be enough training pairs and

Study Area		No	Title	Author	Journal	Year	Vol(No),pp.	Findings	Limitations
									the cv values of these data should also be known for higher reliability.
		17	Prediction of Fabric End-use Using a Neural Network Technique	Chen et al.	Journal of the Textile Institute	2001	92(2), 157-163.	to predict fabric end-use	/
		18	New Approaches to Establishing Translation Equations for the Total Hand Value of Fabric	Shyr et al.	Textile Research Journal	2004	74(6), 528-534.	a one-step transformation process to establish translation equations for total hand evaluations of fabrics by employing a stepwise regression method and an artificial neural network	/
		19	Artificial neural network-based prediction of aesthetic and functional properties of worsted suiting fabrics	Behera and Mishra	International Journal of Clothing Science and Technology	2007	19(5), 259-276.	investigated the prediction of non-linear relations of functional and aesthetic properties of worsted suiting fabrics for	Prediction performance can be further improved by including these parameters as input, during the training phase.

Study Area		No	Title	Author	Journal	Year	Vol(No),pp.	Findings	Limitations
								fabric development by an engineered approach of a radial basis function network which was trained with worsted fabric constructional parameters	In few cases, the network has predicted contradictory trends, which are found difficult to be explained
		20	An Artificial Neural Network Model for the Prediction of Spirality of Fully Relaxed Single Jersey Fabrics	Murrells et al.	Textile Research Journal	2009	79(3), 227-234.	the prediction of the degree of spirality of single jersey fabrics made from a total of 66 fabric samples produced from three types of 100% cotton yarn samples	The neural network model outperformed the multiple regression model in predicting the angle of spirality using data that were not used to train the network. This indicates that it is worthwhile using the more complex ANN technique if a large amount of different types of data are available

Study Area		No	Title	Author	Journal	Year	Vol(No),pp.	Findings	Limitations
		21	The Prediction of Initial Load-extension Behavior of Woven Fabrics Using Artificial Neural Network	Hadizadeh et al.	Textile Research Journal	2009	79(17), 1599-1609.	predicting initial load-extension behavior (Young's modulus) in the warp and weft directions of plain weave and plain weave derivative fabrics	/
		22	Application of an Adaptive Neuro-fuzzy System for Prediction of Initial Load--Extension Behavior of Plain-woven Fabrics	Hadizadeh et al.	Textile Research Journal	2010	80(10), 981-990.	predicting initial load–extension behavior of plain-woven fabrics based on an adaptive neuro-fuzzy inference system (ANFIS)	/
	3.3 Fabric defect	23	Fabric Inspection Based on Best Wavelet Packet Bases	Hu and Tsai	Textile Research Journal	2000	70(8), 662-670.	best wavelet packet bases and an artificial neural network (ANN) to inspect four kinds of fabric defects	/
		24	Classifying Web Defects with a Back-Propagation Neural Network by Color Image Processing	Shiau et al.	Textile Research Journal	2000	70(7), 633-640.	a back-propagation neural network topology to automatically recognize neps and trash in a web by color image processing	Since neps and trash in a web can be recognized, yarn quality not only can be assessed but also improved using a reference for adjusting manufacturing parameters

Study Area		No	Title	Author	Journal	Year	Vol(No),pp.	Findings	Limitations
		25	Detecting Fabric Defects with Computer Vision and Fuzzy Rule Generation. Part II: Defect Identification by a Fuzzy Expert System	Choi et al.	Textile Research Journal	2001	71(7), 563-573.	a fabric defect identifying system by using fuzzy inference in multicondi-tion	The CCD (charge coupled device) must be mounted, despite the scanner, because of on-line considerations. Patterned and complex fabrics can be inspected as well as plain fabrics. For further research such as a neuro-fuzzy expert system can identify actual defect types like reed marks, mispicks, pilling, finger marks, and others.
		26	Neural-Fuzzy Classification for Fabric Defects	Huang and Chen	Textile Research Journal	2001	71(3), 220-224.	an image classification by a neural-fuzzy system for normal fabrics and eight kinds of fabric defects	/

Study Area		No	Title	Author	Journal	Year	Vol(No),pp.	Findings	Limitations
		27	Computer Vision-Aided Fabric Inspection System for On-Circular Knitting Machine	Saeidi et al.	Textile Research Journal	2005	75(6), 492-497.	a computer vision-based fabric inspection system implemented on a circular knitting machine to inspect the fabric under construction	Since this research is limited by the speed of the knitting machine, further studies are required to inspect the fabric defects in higher speed, circular knitting machines.
		28	Detection and Classification of Defects in Knitted Fabric Structures	Shady et al.	Textile Research Journal	2006	76(4), 295-300.	for knitted fabric defect detection and classification using image analysis and neural networks	/
		29	Fabric Stitching Inspection Using Segmented Window Technique and BP Neural Network	Yuen et al.	Textile Research Journal	2009	79(1), 24-35.	a novel method to detect the fabric defect automatically with a segmented window technique which was presented to segment an image for a three layer BP neural network to classify fabric stitching defects	Work is still needed to be done in two major aspects: (1) the applicability of the developed method in studying other manufacturing defects needs to be validated; and (2) the current 2-D-based investigation needs to be

Study Area		No	Title	Author	Journal	Year	Vol(No),pp.	Findings	Limitations
									extended to three-dimensional (3-D) space for actual manual inspection.
	3.4 Sewing	30	Selecting Optimal Interlinings with a Neural Network	Jeong et al.	Textile Research Journal	2000	70(11), 1005-1010.	a neural network and subjoined local approximation technique for application to the sewing process by selecting optimal interlinings for woolen fabrics	/
		31	Application of artificial neural networks to the prediction of sewing performance of fabrics	Hui et al.	International Journal of Clothing Science and Technology	2007	19(5), 291-318.	to predict the sewing performance of woven fabrics for efficient planning and control for the sewing operation based on the physical and mechanical properties of fabrics	/

Study Area		No	Title	Author	Journal	Year	Vol(No),pp.	Findings	Limitations
	3.5 Seam performance	32	Predicting Seam Performance of Commercial Woven Fabrics Using Multiple Logarithm Regression and Artificial Neural Networks	Hui and Ng	Textile Research Journal	2009	79(18), 1649-1657.	the capability of artificial neural networks based on a back propagation algorithm with weight decay technique and multiple logarithm regression (MLR) methods for modeling seam performance of fifty commercial woven fabrics used for the manufacture of men's and women's outerwear	/
		33	Predicting the Seam Strength of Notched Webbings for Parachute Assemblies Using the Taguchi's Design of Experiment and Artificial Neural Networks	Onal et al.	Textile Research Journal	2009	79(5), 468-478.	the effect of factors on seam strength of webbings made from polyamide 6.6	In these comparisons, RMSE values were used as comparative metrics. As a result, it can be said that ANN appears to be a

Study Area		No	Title	Author	Journal	Year	Vol(No),pp.	Findings	Limitations
									reliable and useful tool in characterizing the effect of some critical manufacturing parameters on the seam strength of webbing, if a sufficient number of replicated experimental data are available to train the ANN.
4. Applications to Chemical Processing		34	Fuzzy Neural Network Approach to Classifying Dyeing Defects	Huang and Yu	Textile Research Journal	2001	71(2), 100-104.	image processing and fuzzy neural network approaches to classify seven kinds of dyeing defects	Fuzzification maps the input feature value to fuzzy sets and so increases the dimensions of the feature space. When fuzzy sets are appropriately chosen, they can increase the separability of classes in the feature space. This allows the fuzzy neural network

Study Area		No	Title	Author	Journal	Year	Vol(No),pp.	Findings	Limitations
									model to fit input-output data more accurately with enhanced classification ability.
5. Applications to Clothing	5.1 Pattern fitting prediction	35	A Hybrid Neural Network and Immune Algorithm Approach for Fit Garment Design	Hu et al.	Textile Research Journal	2009	79(14), 1319-1330.	to predict the fit of the garments and search optimal sizes	For future research directions, the dataset needs to be enriched. The current scale is definitely not enough to study all sizes of the garment. In order to present the fuzzy and stochastic nature of the garment and body sizes, it should be modeled as fuzzy vector or stochastic vector. In addition, it is valuable to incorporate NN-ICEA into garment CAD

Study Area		No	Title	Author	Journal	Year	Vol(No),pp.	Findings	Limitations
									system, thus the 2D and 3D effects of garments can provide intuitive impressions
	5.2 Clothing sensory comfort	36	Neural Network Predictions of Human Psychological Perceptions of Clothing Sensory Comfort	Wong et al.	Textile Research Journal	2003	73(1), 31-37.	the predictability of clothing sensory comfort from psychological perceptions by using a feed-forward back-propagation network in an artificial neural network (ANN) system	The functions and interrelationships of individual sensory perceptions and comfort are unknown.
		37	Predicting Clothing Sensory Comfort with Artificial Intelligence Hybrid Models	Wong et al.	Textile Research Journal	2004	74(1), 13-19.	to develop an intellectual understanding of and methodology for predicting clothing comfort performance from fabric physical properties	/

8. Reference

Admuthe, L.S. and Apte, S. Adaptive Neuro-fuzzy Inference System with Subtractive Clustering: A Model to Predict Fiber and Yarn Relationship. *Textile Research Journal*, 2010, 80(9), 841-846.

Behera, B.K. and Goyal, Y. Artificial Neural Network System for the Design of Airbag Fabrics. *Journal of Industrial Textiles*, 2009, 39(1), 45-55.

Behera, B.K. and Karthikeyan, B. Artificial Neural Network-embedded Expert System for the Design of Canopy Fabrics. *Journal of Industrial Textiles*, 2006, 36(2), 111-123.

Behera, B.K. and Mishra, R. Artificial neural network-based prediction of aesthetic and functional properties of worsted suiting fabrics. *International Journal of Clothing Science and Technology*. 2007, 19(5), 259-276.

Beltran, R., Wang, L. and Wang, X. Predicting Worsted Spinning Performance with an Artificial Neural Network Model. *Textile Research Journal*, 2004, 74(9), 757-763.

Chen, Y., Zhao, T. and Collier, B.J. Prediction of Fabric End-use Using a Neural Network Technique. *Journal of the Textile Institute*, 2001, 92(2), 157-163.

Choi, H.T., Jeong, S.H., Kim, S.R., Jaung, J.Y. and Kim, S.H. Detecting Fabric Defects with Computer Vision and Fuzzy Rule Generation. Part II: Defect Identification by a Fuzzy Expert System. *Textile Research Journal*, 2001, 71(7), 563-573.

Durand, A., Devos, O., Ruckebusch, C. and Huvenne, J.P. Genetic algorithm optimisation combined with partial least squares regression and mutual information variable selection procedures in near-infrared quantitative analysis of cotton–viscose textiles. *Analytica Chimica Acta*, 2007, 595(1-2), 72-79.

Ertugrul, S. and Ucar, N. Predicting Bursting Strength of Cotton Plain Knitted Fabrics Using Intelligent Techniques. *Textile Research Journal*, 2000, 70(10), 845-851.

Farooq, A. and Cherif, C. Use of Artificial Neural Networks for Determining the Leveling Action Point at the Auto-leveling Draw Frame. *Textile Research Journal*, 2008, 78(6), 502-509.

Hadizadeh, M., Jeddi, A.A.A., and Tehran, M.A. The Prediction of Initial Load-extension Behavior of Woven Fabrics Using Artificial Neural Network. *Textile Research Journal*, 2009, 79(17), 1599-1609.

Hadizadeh, M., Tehran, M.A. and Jeddi, A.A.A. Application of an Adaptive Neuro-fuzzy System for Prediction of Initial Load--Extension Behavior of Plain-woven Fabrics. *Textile Research Journal*, 2010, 80(10), 981-990.

Huang, C.C. and Chen, I.C. Neural-Fuzzy Classification for Fabric Defects. *Textile Research Journal*, 2001, 71(3), 220-224.

Huang, C.C. and Yu, W.H. Fuzzy Neural Network Approach to Classifying Dyeing Defects. Textile *Research Journal*, 2001, 71(2), 100-104

Hui, C.L. and Ng, S.F. Predicting Seam Performance of Commercial Woven Fabrics Using Multiple Logarithm Regression and Artificial Neural Networks. *Textile Research* Journal, 2009, 79(18), 1649-1657.

Hui, C.L.P., Chan, C.C.K., Yeung, K.W. and Ng, S.F.F. Application of artificial neural networks to the prediction of sewing performance of fabrics. *International Journal of* Clothing *Science and Technology*. 2007, 19(5), 291-318.

Hu, M.C. and Tsai, I.S. Fabric Inspection Based on Best Wavelet Packet Bases. *Textile Research Journal*, 2000, 70(8), 662-670.

Hu, Z.H., Ding, Y.S., Yu, X.K., Zhang, W.B. and Yan, Q. A Hybrid Neural Network and Immune Algorithm Approach for Fit Garment Design. *Textile Research Journal*, 2009, 79(14), 1319-1330.

Jeong, S.H., Kim, J.H. and Hong, C.J. Selecting Optimal Interlinings with a Neural Network. Textile *Research Journal*, 2000, 70(11), 1005-1010.

Kang, T.J. and Kim, S.C. Objective Evaluation of the Trash and Color of Raw Cotton by Image Processing and Neural Network. *Textile Research Journal*, 2002, 72(9), 776-782.

Khan, Z., Lim, A.E.K., Wang, L., Wang, X. and Beltran, R. An Artificial Neural Network-based Hairiness Prediction Model for Worsted Wool Yarns. *Textile Research Journal*, 2009, 79(8), 714-720.

Kuo, C.F.J., Hsiao, K.I. and Wu, Y.S. Using Neural Network Theory to Predict the Properties of Melt Spun Fibers. *Textile Research Journal*, 2004, 74(9), 840-843.

Lin, J.J. Prediction of Yarn Shrinkage using Neural Nets. *Textile Research Journal*, 2007, 77(5), 336-342.

Murrells, C.M., Tao, X.M., Xu, B.G. and Cheng, K.P.S. An Artificial Neural Network Model for the Prediction of Spirality of Fully Relaxed Single Jersey Fabrics. *Textile Research* Journal, 2009, 79(3), 227-234.

Onal, L., Zeydan, M., Korkmaz, M. and Meeran, S. Predicting the Seam Strength of Notched Webbings for Parachute Assemblies Using the Taguchi's Design of Experiment and Artificial Neural Networks. *Textile Research Journal*, 2009, 79(5), 468-478.

Saeidi, R.G., Latifi, M., Najar, S.S. and Saeidi, A.G. Computer Vision-Aided Fabric Inspection System for On-Circular Knitting Machine. *Textile Research Journal*, 2005, 75(6), 492-497.

Shady, E., Gowayed, Y., Abouiiana, M., Youssef, S. and Pastore, C. Detection and Classification of Defects in Knitted Fabric Structures. *Textile Research Journal*, 2006, 76(4), 295-300.

She, F.H., Kong, L.X., Nahavandi, S. and Kouzani, A.Z. Intelligent Animal Fiber Classification with Artificial Neural Networks. *Textile Research Journal*, 2002, 72(7), 594-600.

Shiau, Y.R., Tsai, I.S. and Lin, C.S. Classifying Web Defects with a Back-Propagation Neural Network by Color Image Processing. *Textile Research Journal*, 2000, 70(7), 633-640.

Shyr, T.W., Lai, S.S. and Lin, J.Y. New Approaches to Establishing Translation Equations for the Total Hand Value of Fabric. *Textile Research Journal*, 2004, 74(6), 528-534.

Ünal, P.G., Arikan, C., Özdil, N. and Taskin, C. The Effect of Fiber Properties on the Characteristics of Spliced Yarns: Part II: Prediction of Retained Spliced Diameter. Textile *Research Journal*, 2010, 0(0), 1-8.

Wong, A.S.W., Li, Y., Yeung, P.K.W. and Lee, P.W.H. Neural Network Predictions of Human Psychological Perceptions of Clothing Sensory Comfort. *Textile Research* Journal, 2003, 73(1), 31-37.

Wong, A.S.W., Li, Y., Yeung, P.K.W. Predicting Clothing Sensory Comfort with Artificial Intelligence Hybrid Models. *Textile Research Journal*, 2004, 74(1), 13-19.

Xu, B., Dong, B. and Chen, Y. Neural Network Technique for Fiber Image Recognition. *Journal of Industrial Textiles*, 2007, 36(4), 329-336.

Yao, G., Guo, J. and Zhou, Y. Predicting the Warp Breakage Rate in Weaving by Neural Network Techniques. *Textile Research Journal*, 2005, 75(3), 274-278.

Yuen, C.W.M., Wong, W.K., Qian, S.Q., Fan, D.D., Chan, L.K. and Fung, E.H.K. Fabric Stitching Inspection Using Segmented Window Technique and BP Neural Network. *Textile Research Journal*, 2009, 79(1), 24-35.

Zeng, Y.C., Wang, K.F. and Yu, C.W. Predicting the Tensile Properties of Air-Jet Spun Yarns. *Textile Research Journal*, 2004, 74(8), 689-694.

2

Artificial Neural Network Prosperities in Textile Applications

Mohammad Amani Tehran and Mahboubeh Maleki
Amirkabir university of Technology
Islamic Republic of IRAN

1. Introduction

Such as other fields, textile industry, deal with numerous large inputs and possible outputs parameters and always feed with a complex interdependence between parameters, it is highly unlikely that an exact mathematical model will ever be developed. Furthermore, since there are many dependent and independent variables during different textile progress, it becomes difficult to conduct and to cover the entire range of the parameters. Moreover, the known and unknown variables cannot be interpolated and extrapolated in a reasonable way based on experimental observations or mill measurements due to the shortage of knowledge on the evaluation of the interaction and significance at weight contributing from each variable. For example, it is quite difficult to develop some universal practical models that can accurately predict yarn quality for different mills (Chattopadhyay & Guha, 2004).
Statistical models have also shown up their limitations in use—not least their sensitivity to rogue data—and are rarely used in any branch of the textile industry as a decision-making tool. The mechanistic models proposed by various authors overtly simplify the case to make the equations manageable and pay the price with their limited accuracy. In any case, the vast volume of process parameter- related data is hardly ever included in these models, making them unsuitable for application in an industrial scenario.
By using neural networks, it seems to be possible to identify and classify different textile properties (Guruprasad & Behera, 2010). Some of the studies reported in recent years on the application of neural networks are discussed hereunder.

2. Fiber classification

The usual tests for fiber identification (usually chemical tests), in addition to being difficult to perform, are almost always destructive in nature.
Leonard et al., 1998 had used Near-infrared (NIR) spectroscopy as input data to a neural network to identify fibers in both original and normalised spectra. The performance of the network was judged by computing the root mean square error of prediction (RMSEP) and was compared with similar results given by multiple linear regressions (MLR).
Accurate classification of animal fibers used in the wool industry is very difficult. Some techniques distinguish these fibers from patterns of their cuticular scales and others from their physical and chemical properties. However, classification of animal fibers is actually a typical task of pattern recognition and classification (Leonard et al., 1998). She et al., 2002

developed an intelligent fiber classification system to objectively identify and classify two types of animal fibers, merino and mohair, by two different methods based on image processing and artificial neural network. There are considerable variations in the shape and contour of the scale cells and their arrangement within the cuticle. They used these two systems based on how the scale features of the animal fibers were extracted. The data was cast images of fibers captured by optical microscopy. Then they applied principal component analysis (PCA) to reduce the dimension of input images and extract an optimal linear feature before applying neural network. Furthermore neural network classifiers generalize better when they have a small number of independent inputs. Finally they used an unsupervised neural network in which the outputs used as inputs in the supervised network (a multilayer perception with a back propagation algorithm) for classification while the fiber classes were the outputs of the output layer. For the unsupervised network, learning rate at 0.005 (step size) was set which linearly decayed to 0.0005 within the first 100 epochs and three different numbers of units in the hidden layer (80, 50, and 20) was used. Multilayer perception used for fiber classification had a hyperbolic tangent activation function in the processing elements of the hidden layer and output layer. They also compared their two systems and concluded that neural network system was more robust since only raw images were used and by developing more powerful learning strategies, the classification accuracy of model would be improved (She et al., 2002).
There are some studies which have been introduced different design of neural network classifier to categorize different type of fibers based on their colors too.
Raw cotton contains various kinds of trash, such as leaf, bark, and seed coat. The content of each of these trash particles is vital for deciding upon the cleaning process (Xu et al., 1999). For instance, the trash and color of raw cotton are very important and decisive factors in the current cotton grading system that determine spinning quality and market value.
For many years, the USDA (United States Department of Agriculture) has used both a visual grading method by trained classers and an instrumental method with HVI (High Volume Instrument) systems to evaluate the color and trash of raw cotton. However it is expensive, slow, and a time consuming process (Kang & Kim, 2002). Xu et al., 1999 used three classification techniques (sum of squares, fuzzy, and neural network) into four groups (bark, leaf, hairy seed coats, and smooth seed coat). They applied two hidden layer with four and six neurons and their results showed that the neural network clustering method outperformed the other used two methods (Xu et al., 1999).
Kang & Kim, 2002 developed an image system to characterize trash from a raw cotton image captured by a color CCD camera and acquired color parameters. They trained and tested neural network based on back propagation algorithm using color parameters as input data from physical standard samples. A sigmoid function was used for an error back propagation model and the number of input and output nodes was eight and seven respectively in accordance with the color parameters and seven grades in the subcategories. The results predicted by neural network were compared with the grades that classers judged (Kang & Kim, 2002).

3. Yarn, fabric, nonwoven and cloth defect detection and categorization

In general, textile quality control is determined by measuring a large number of properties (including mechanical and physical properties, and etc), which in many cases can only be done by skilled workers or expensive equipments (Lien & Lee, 2002). Generally, In textile

industry, textiles are inspected manually for defects, but some problems arise in this visual inspection, such as excessive time consumed, human subjective factors, stress on mind and body, and fatigue. These problems further influence production volume and inspection accuracy. Therefore, techniques that can replace manual inspection have emerged (Kuo & Lee, 2003). In recent years, neural networks have been used to inspect yarn, fabric and cloth defects and to identify their types (Kuo, 2003). Neural networks are among the best classifier used for fault detection due to their non-parametric nature and ability to describe complex decision regions.

A key issue in many neural network applications is to determine which of the available input features should be used for modeling (Kumar, 2003). Mostly, researchers have used different ways for feature selection based on image processing methods in conjunction with neural network. An image acquisition setup that yields suitable images is crucial for a reliable and accurate judgment. This system is usually including the specimen, the camera or scanner and the illumination assembly (Bahlmann et al., 1999). Some studied have used near sensor image processing (NSIP) technology as well. Most researchers had converted the original color image to gray level image to improve the computer processing speed and reducing the dimensions of information. However, Tilocca et al., 2002 presented a method to fabric inspection based both on gray levels and 3D range profile data of the sample (Tilocca, 2002). Most studies usually have employed histogram equalization, noise reduction operation by filtering, etc to improve visual appearance of the image (Jeon, 2003). When they use image technology in conjunction with neural networks, some problems may occur; For example recognizable rate of defect may be related to light source conditions (Kuo & Lee, 2003). Since a fine feature selection can simplify problem identification by ranking the feature and those features that do not affect the identification capability can be removed to increase operation efficiency and decrease the cost of evaluation systems without losing accuracy (Lien & Lee, 2002). So some studies have applied principal component analysis (PCA) as pre processing methods to reduce the dimension of feature vectors (Kumar, 2003).

Usually, in ANN, the available data are divided into three groups. The first group is the training set. The second group is the validation set, which is useful when the network begins to over-fit the data so the error on the validation set typically begins to rise; during this time the training is stopped for a specified number of iterations (max fails) and the weights and biases at the minimum of the validation error are returned. The last group is the performance test set, which is useful to plot the test set error during the training process (Liu, 2001).

Data are further processed to extract specific features which are then transmitted to either supervised or unsupervised neural network for identification and classification. This feature extraction step is in accordance with textural structure, the difference in gray levels, the shape and size of the defects and etc (Kuo et al., 2003) and it is necessary to improve the performance of the neural network classifier (Tilocca, 2002). Consequently, a large amount of study is usually related to this step to extract useful information from images and feed them to neural network as input to recognize and categorize yarn, nonwoven, fabric, and garment defects.

In supervised systems, the neural network can establish its own data base after it has learned different defects with different properties. Most researchers have been used multi layer feed forward back propagation Neural network since it is a nonlinear regressional algorithm and can be used for learning and classifying distinct defects.

There are numerous publications on neural network applications addressing wide variety of textile defects including yarn, fabric and garment defects. Some of the studies reported on this application of neural networks are discussed hereunder.

3.1 Yarn defects

Sliver levelness is one of the critical factors when producing quality yarn products in spinning processes. However, it is difficult to model the drafting process exactly since these controls do not need to model the process and can handle very complicate processes, they are useful. Moreover, they possess the ability to improve the intelligence of systems working in an uncertain, imprecise, noisy environment. Therefore, Huang & Chang, 2001 developed an auto leveling system with a drawing frame using fuzzy self-organizing and neural network applied on a laboratory scale drawing frame with two drafting zones and two-sliver doubling samples. They used a three layer neural network model to compute the Jacobean matrix, which was needed in training the weights and thresholds on-line. A back propagation learning algorithm was used to tune the connection weights and thresholds and the unipolar sigmoid function as the activation function to compute the output of a node. Levelness performance was evaluated by the CV% of sliver products in which their results showed that neural network controller yielded more level slivers than the fuzzy self-organizing controller. The neural network controller kept learning from the feedback of the output linear density and generated the control action by the feed linear density and the desired output linear density. The weight and thresholds of the neural network controller were tuned on-line, leading to reduced variance in the output with respect to the desired value (Huang & Chang, 2001).

It is well known that spinning process is a complex manufacturing system with the uncertainty and the imprecision, in which raw materials, processing methodologies, and equipments and so on all influence the yarn quality (Yin & Yu, 2007). Yarn physical properties like strength, appearance, abrasion and bending are the most important parameters, affecting on the quality and performance of end products and also cost of the yarn to fabric process (Cheng & Lam, 2003).

Lien & Lee, 2002 reported feature selection for textile yarn grading to select the properties of minimum standard deviation and maximum recognizable distance between clusters to achieve effectiveness and reduce grading process costs. Yarn features were ranked according to importance with the distance between clusters (EDC) which could be applied to either supervised or unsupervised systems. However, they used a back propagation neural network learning process, a mathematical method and a normal algebraic method to verify feature selection and explained the observed results. A thirty sets data were selected containing twenty data as training sets and the other ten data as testing sets. Each of these data were the properties of single yarn strength, 100 meter weight, yarn evenness, blackboard neps, single yarn breaking strength, and 100-meter weight tolerance (Lien & Lee, 2002).

A performance prediction of the spliced cotton yarns was estimated by Cheng & Lam, 2003 using a regression model and also a neural network model. Different spliced yarn properties such as strength, bending, abrasion, and appearance were merged into a single score which was then used to analyze the overall performance of the yarns by those two models. The appearance of the spliced yarns was expressed as the retained yarn appearance (RYA) which 5 was identical, 3 was acceptable and 1 was fail values. They used the transfer functions of hyperbolic tangent sigmoid transfer function and linear transfer function.

According to their analytical results, the neural network model (R=0.98) gave a more accurate prediction that the regression model (R=0.74) (Cheng & Lam, 2003).

It is well known that worsted spinning process is a complex manufacturing system and there are many dependent and independent variables during spinning which becomes difficult to conduct and cover the entire range of the parameters using mathematical and empirical models. Yin & yu, 2007 firstly analyze all the variables collected from the mill through grey superior analysis (GS) in order to select the important variables and as a result better improve the yarn quality before ANNs model (multi-layer perceptron) was used by adopting the back-propagation neural network (BP) to estimate the validity of the input variables. In their research, they evaluated yarn qualities i.e. yarn unevenness, strength, extension at break, and ends-down per 1000 spindle hours; by means of inputs including the processing parameters such as fiber properties, spinning method, and process variables influencing on the yarn properties and spinning performance. From the 77 sets of data, 69 lots were selected at random to serve as learning set and the residual eight sets data were recorded as test sets. A one layer hidden layer was decided based on experiments by achieving the highest coefficient using back propagation learning. The prediction accuracy, A (%) and relative coefficient, R (%), between the predicted values and achieved values were calculated in order to validate the approaches of the variables selection. The comparison of the performance of ANNs model using grey superior analysis (GS), subjective and empirical approach (SE), and multilinear regress method (MLR) showed that the model using the input variables selected by GS was superior to that by SE and MLR. They also simulated the spinning of the worsted yarn with the high coincidence using the processing data in the mills based on the artificial neural networks and grey superior analysis (Yin & yu, 2007).

One of the important properties of yarns is unevenness. Mass or weight variation per unit length of yarn is defined as unevenness or irregularity. It can adversely influence many of the properties of textile materials such as tenacity, yarn faults, twist variation, abrasion, pilling, soil retention, drape, absorbency, reflectance or luster. Unevenness in blended yarns is depended mainly on the physical properties of fibers (fiber cross section deviation, length and length uniformity etc.), number of fibers and fiber location or positioning in the yarn cross section, blend ratio and working performance of the yarn spinning machine. Therefore, Demiryurek & Koc, 2009 developed an artificial neural network and a statistical model to predict the unevenness of polyester/viscose blended open-end rotor spun yarns. They used a back propagation multi layer perceptron network and a mixture process crossed regression model with two process variables (yarn count and rotor speed). They selected blend ratio, yarn count and the rotor speed as input parameters and unevenness of the yarns as output parameter. Sigmoid function was used as activation function, and number of hidden layer was determined as 25, the learning rate and momentum were optimized at 0.2 and 0.0 respectively in this study. They compared the result of both presented model and it was concluded that both models had satisfactory and acceptable results, however the correlation coefficient of neural network (0.98) was slightly greater than statistical model (0.93) and the mean square errors (0.077) were identical. The mean absolute percentage error was also calculated and was %1.58 and %0.73 for the ANN and statistical model respectively. Contrary to general opinion of the more reliable prediction of ANN than statistical models, they reported that statistical model developed was more reliable than ANN and by increasing the number of experiments, prediction performance of ANN would increase (Demiryurek & Koc, 2009).

2.2 Woven fabric defects

Image processing analyses in conjunction with neural networks have been widely used for woven and knitted fabric defect detection and grading.

Karras et al., 1998 investigated a vision based system to detect textile defects from the textural properties of their corresponding wavelet transformed images. They applied supervised (multilayer perceptrons trained with the back propagation algorithm) and unsupervised (Kohonen's self organizing feature maps) neural classification techniques by exploiting information coming from textural analysis and SVD in the wavelet transformed original images to provide second order information about pixel intensities and localize important information respectively. They considered defect detection as the approximation of the defect spatial probability distribution within the original image. The inputs to the MLP and SOFM networks were the 24 features contain 1009 patterns of the feature vector extracted from each sliding window. 280 out of the 1009 patterns belonged to the long and thin defective area of the upper side, while the rest belonged to the class of non defective areas. Reported classification accuracy was an overall 98.50% (Karras et al., 1998).

Tilocca et al., 2002 presented a direct method to fabric inspection based both on gray levels and 3D range profile data of the sample. They used a smart vision sensor for image acquisition system. The neural network was trained to classify three different categories which were normal fabric, defect with a marked 3D component and defect with no 3D component. A three layered feed forward neural network with sigmoid activation function and back propagation learning algorithm by a fixed learning rate at 0.2. They extracted 1500 training patterns including nondefective region, defects with marked 3D characteristics, and defects without 3D marks and another group of 500 patterns constituted the test sets. The number of hidden neurons was adjusted by trial and error at 24. They obtained the percentage of right, unknown, and wrong classifications for each class, both for the training and test sets. Percentage of test clean patterns correctly classified was almost 92%, showing that the ANN was able to identify and separate defective from nondefective regions. They suggested using this system for on-line monitoring of fabric defects since no further transformation of the data was needed before classification (Tilocca et al., 2002).

At present, fabric inspection still relies on the human eye, and the reliability and accuracy of the results are based on inspectors. Wrinkles in cloth usually develop with deformation during wearing, after washing and drying, and with folding during storage and it is not easy even for trained observers to judge the wrinkles. Mori & Komiyama, 2002 used gray scale image analysis of six kinds of plain fabrics to evaluate visual features of wrinkles in plain fabrics made from cotton, linen, rayon, wool, silk, and polyester using neural network. The angular second moment, contrast, correlation, and entropy were extracted from the gray level co-occurrence matrix and fractal dimension from fractal analysis of the image as input and the mean sensory value presenting the grade of wrinkled fabrics as output. The hidden units had logistic function as transfer function. Eight sets of data were selected arbitrarily as training data and the seven remaining data sets for testing the neural networks were used. They used a training algorithm with Kalman filter to tune the network in order to maximize the accuracy of the visual evaluation system. Sum of the square error (SSE) was used as total output error of the network. Overtraining was occurred in the region of more than 200 learning cycles, therefore they decided 150 learning cycles for checking or testing the network. They also compared the accuracy of the evaluating system for wrinkled images captured by the digital camera method with that for wrinkled images captured by the color scanner method and observed better accuracy for the color scanner than digital camera (Mori & Komiyama, 2002).

Kuo & Lee, 2003 used a back-propagation neural network for recognizing woven fabric defects. They used an image system (filtered and threshold images) to distinguished holes, oil stains, wrap-lacking and weft-lacking defects. Maximum length, maximum width and gray level of the defects were presented as the input units of the neural network. They used a back propagation neural network by eight defect samples for off line training. The initial learning rate was 0.1; keeping reducing to 0.01 and the momentum factor was 0.5. The error mean square value converged to 0.05 after 45000 iterations. According to their test, the recognizable rate of warp-lacking and weft-lacking was up to 95%, and up to 100% for holes and oil stains (Kuo & Lee, 2003). Kuo et al., 2003 used an image system for dynamic inspection of plain white fabrics using a linear scan digital camera with direct light to take images. The corresponding fabric conveying speed was 50 cm/s. the back propagation neural network of this research comprised an input layer with three input units (maximum length of the defect, maximum width of defect, and gray level value of the defect), a hidden layer, and an output layer by three output units. They reported average overall recognition rates up to 90% (Kuo et al., 2003).

Segmentation of defects provides accurate distinguishing of size and location of defects. Therefore, Kumar, 2003 investigated an approach to segment a variety of local textile (twill and plain weave fabrics) defects using feed-forward neural network. Since every fabric defect alters the gray-level arrangement of neighboring pixels, he extracted the feature vector for every pixel of backlighting captured images and applied a pre-processing using normalization of the feature vectors followed by principal component analysis (PCA) to reduce the dimension of feature vectors. He also used post-processed operation (a 9*9 median filtering) to generate the required output values. Hyperbolic tangent sigmoid activation function was chosen and the weights were updated using Levenberg-Marquardt algorithm for faster convergence rate. The network was trained for the maximum of 1000 steps with the learning rate of 0.01 and the training was stopped if the maximum performance gradient of 1e-10 was reached. Finally, a low-cost web inspection system based on linear neural network with a single layer to evaluate real fabric samples was proposed since the web inspection based on defect segmentation required additional DSP hardware, which would increase the cost of the inspection system (Kumar, 2003).

Pilling may be defined as a surface fabric fault comprising of circular accumulations of entangled fibers that cling to the fabric surface thereby affecting the appearance and handle of the fabric. The pilling of fabrics is a serious problem for the apparel industry and in particular wool knitwear fabrics. The formations of pills occur as a consequence of mechanical action during washing or wear (Beltran et al., 2005). The development of pills on a fabric surface, spoils the original appearance and hand, initiates garment attrition and reduces serviceability. Therefore evaluating pilling degree (from grade 5 which means no pilling to grade 1 which is very severe pilling) of fabric is important and usually it is inspected visually. Because of the inconsistency and inaccuracy of rating results obtained with the visual method, more reliable and objective methods for pilling evaluation are desirable for the textile industry. Chen & Huang, 2004 evaluated and graded fabric pilling based on light projection using image analysis and neural network to overcome the common difficulty of interference with fabric pill information from fabric color and pattern. Firstly, they eliminated interference with pilling information from fabric color and pattern. Their method was included a device to acquire the projected cross-sectional images, detecting the profile of projected images, segmenting pills appearing on converted gray images, extracting of a pill's feature index, and finally assessing pilling grade by Kohonen self

organizing feature map neural network. There were ten input neurons corresponding to ten feature indexes and five output nodes representing five cluster centers (five pilling grades) by training twenty kinds of samples including colored and patterned pilled worsted fabrics. The total number of iterations in the training process was 400, and the learning rate was initialized to be 0.02.They concluded that the objective pilling grade was in good agreement with the subjective pilling grade. The correlation coefficient for training and testing samples were reported up to 0.94 and 1 respectively (Chen & Huang, 2004).

Beltran et al., 2005 also used artificial neural networks to model the multi-linear relationship between fiber, yarn and fabric properties and their effect on the pilling propensity of pure wool knitted fabrics. They used key fiber (diameter, CV, diameter > 30 μm and curvature), top (Hauteur, CV, short fiber <30mm, bundle strength and strain), yarn (count, hairiness, thin and thick places, twist factor, folding twist ratio) and fabric properties (cover factor) as quantitative inputs (normalized data) along with their corresponding pilling intensities in an ANN to predict the pilling performance of knitted wool fabrics. The corresponding mean pill rating was served as the target output. 105 sets of randomized data were assigned to training, 20 sets were assigned for cross validation and 10 data sets were selected for testing the network. The network consisted of a single hidden layer multi layer perception trained with the error back propagation algorithm possessing hyperbolic tanh activation function in both the hidden and output layers (Beltran et al., 2005).

Zhang et al., 2010 investigated an approach for fabric defect classification using radial basis function (RBF) network improved by Gaussian mixture model (GMM). First, the gray level arrangement in the neighborhood of each pixel was extracted as the feature. This raw feature was subject to principal component analysis (PCA) which adopted the between class scatter matrix as the generation matrix to eliminate the variance within the same class. Second, the RBF network with Gaussian kernel was used as the classifier because of the nonlinear discrimination ability and support for multi-output. To train the classifier, GMM was introduced to cluster the feature set and precisely estimate the parameter in Gaussian RBF, in which each cluster strictly conforms to a multi-variance Gaussian distribution. Thus the parameter of each kernel function in RBF network could be acquired from a corresponding cluster. The proposed algorithm was experimented on fabric defect images with nine classes (mould, miss weft, damaged, double pick, cloud pick, coarse end, color smear, broken edge, and filling end) and achieved superior performance. Fabric images were collected under the back-lighting condition with the cloth moving speed of 100 m/min. in the training process, 30 images of each class were processed and repeated 5 times. They also compared the performance of three classifiers including ANN (9-16-10 feed forward structure using back propagation algorithm), SVM (Support Vector Machine which can automatically determine support vectors from the sample set which is normalized and preprocessed by PCA using Gaussian function as kernel), and RBF network on fabric defect classification. These schemes were evaluated on the same nine classes of fabric defect images. The training and test process was repeated five times to get an average performance. The result was measured by correct classification rate (CCR) which was defined as the number of correctly classified images divided by the number of total images. They found that ANN had the worst performance with an average CCR of 74% while the performance of RBF network was the best with CCR of 83.2% and the performance of SVM was sensitive to the parameters. Therefore, they reported that RBF network was an appropriate choice for the real time fabric defect classification. It has to be noted that this work was the first time that the RBF network was applied in fabric defect classification

which achieved excellent performance in combination with GMM in comparison with classical feed forward network (Zhang et al., 2010).

2.3 Knitted fabric defects

The apparent quality of knitted fabrics can be divided into two categories. First, the fabrics with a large number of area faults that were occurring in the knitting process and eventually make them useless. In the second category, there are inputted faults that originate from yarn faults and the apparent quality of yarn is directly related to the configuration of fibers on its surface (Liu et al., 2001). Different studies have been reported and identified both problems simultaneously or separately.

Detecting and classifying knitted fabric defects using image analysis and neural network were performed by Shady et al., 2006. They utilized two approaches including statistical procedures and fourier transforms to extract image features for six different knitted fabric defects using a defect free fabric as a control sample. All images were processed using histogram equalization and then converted to grayscale images. The feature vectors were used as input vectors to the network and six types of defects including broken needle, fly, hole, barre, thick yarn and thin yarn were identified and classified. Two neural networks were trained and tested for each feature extraction approach. The first one contained seven neurons in the input layer representing the seven features of the statistical approach, and seven neurons in the output layer representing the sic different defects and the free defect sample. This network was successful only in classifying broken needle, hole, thick and thin yarn defects. In the second neural network, six neurons were used in the input layer representing the features and seven neurons in the output layer representing the six defects and the free defect sample. The worst results were observed for the barre defects. In their work, the neural network was trained by the learning vector quantization (LVQ) algorithm to detect and classify the knitted fabric defects. Their results showed success in classifying most of the defects excluding barre defects (Shady et al., 2006).

Fabric spirality is a problem which affects the esthetics and quality of knitted fabrics. This problem is complex and there is a large amount of data required to establish quantitative relationship to model this phenomenon accurately. an artificial neural network model was proposed by Murrells et al., 2009 for the prediction of the degree of spirality of single jersey fabrics made from 100% cotton conventional and modified ring spun yarns from a number of factors considered to have the potential to influence fabric spirality after wash and dry relaxation such as twist liveliness, yarn type, yarn linear density, fabric tightness factor, the number of feeders, rotational direction, gauge of knitting machine and dyeing method. They compared ANN model (R=0.976) with a multiple regression model (R=0.970) and concluded that ANN model produced superior results to predict the degree of fabric spirality after three washing and drying cycles. The hyperbolic tangent sigmoid transfer function was assigned as the activation function in the hidden layer and the linear function was used in the output layer. During the process, 60%, 20%, and remaining 20% of the original data were set aside for training, validation, and testing respectively. They also investigated the relative importance of the investigated factors influencing the spirality of the fabric and tried various network structures with one hidden layer and finally demonstrated that multilayer feed forward network based on Levenberg-Marquardt learning algorithm had better results. Furthermore, both the ANN and the regression approach showed that twist liveliness, tightness factor, and yarn linear density were the most important factors in predicting fabric spirality (Murrells et al., 2009).

Semnani & Vadood, 2009 applied the artificial neural network (ANN) to predict the apparent quality of weft knitted fabrics. They considered, only the appearance of the safe knitted fabric without any knitting faults, tightened fibers with uniform configuration, big faults with less area, non-uniform and extended faults with spread configuration, and small spread faults such as non-uniform coating fibers and short tangled hairs had been considered (Semnani & Vadood, 2009).
There are some variables in the applied neural network where their variation affects on the obtained results are significant. These variables include the number of hidden layers, the number of neurons in hidden layers, the value of max fail and the percentage of validation and testing data.
Therefore, Semnani & Vadood, 2009 applied genetic algorithm in their research because of its intuitiveness, ease of implementation and the ability to effectively solve highly nonlinear, mixed integer optimization problems. Their results showed that the ANN could be optimized very well by the genetic algorithm method and the designed ANN was very accurate and applicable to predict the apparent parameters. Their optimized ANN was formed from two hidden layers, in which the first hidden layer had 8 and the second layer had 7 neurons, one neuron for output layer, five epochs for max fail, 20% available data for test and 10% of available data for validation (Semnani & Vadood, 2009).

2.4 Nonwoven defects

Liu et al., 2010 proposed an algorithm based on wavelet transform (feature extraction procedure) and learning vector quantization (LVQ) neural network for nonwoven uniformity identification and grading. Six hundred and twenty-five nonwoven images of five different grades, 125 images of each grade, were decomposed at four different levels with five wavelet bases of Daubechies family, and two kinds of energy values L^1 and L^2 extracted from the high frequency subbands were used as the input features of the LVQ neural network solely and jointly. The network outputs were class labels, which were defined with five integer numbers, from 1 to 5, denoting five different uniformity grades. The number of neurons in hidden layer, training epochs and goal, of the LVQ neural network were as 5, 200 and 0.01 respectively. They used the identification accuracy of each grade and average identification accuracy (AIA%) of five grades as performance parameters. Their results were expressed and compared five wavelet bases (db_2, db_4, db_6, db_8, and db_{10}) and even different features (L^1, L^2, and L^1UL^2) at the four levels (level 1 to 4). They noted three points as Firstly, with the same feature set and decomposition level, the length of the filter had little effect in performance in all methods. Secondly, with the same feature set and wavelet base, the decomposition level had a significant effect in the performance in all methods. Thirdly, the highest identification accuracy was gotten at the crossing point db_4 or db_6 and level 3 (Liu et al., 2010).
Liu et al., 2010 presented a method to recognize the visual quality of nonwoven by combining wavelet texture analysis, Bayesian neural network and outlier detection. Each nonwoven image was decomposed with orthogonal wavelet bases at four levels and two textural features, norm-1 and norm-2, which were used as the input of Bayesian neural network for training and test. To detect the outlier in the training set, the scaled outlier probability was introduced to increase its robustness. All nonwoven samples were classified into five grades according to visual qualities (such as surface uniformity, the condition of pilling, wrinkles and defects). Each image was individually normalized to zero mean and

unit variance before wavelet transform. They reported with the increase of decomposition level, the average classification error and cross entropy of training and test set decreased sharply and the recognition accuracy of the five grades was also affected (Liu et al., 2010).

2.5 Cloth defects

Quality inspection of garments is an important aspect of clothing manufacturing. For many textile products, a major quality control requirement is judging seam quality visually by human experts. Presently, this is still accomplished by human experts, which is very time consuming and suffers from variability due to human subjectivity. Consequently, investigations about automated seam quality classification and an implementation of an automated seam classificator are highly desirable. Bahlmann et al., 1999 presented a method for automated quality control of textile seams by a scale of five grades (from grade 5 which was best to grade 1 which was worst). Their system was consisting of an image acquisition setup (to record seams structures), an algorithm for locating the seam (transforming acquired seam images to normalize position), a feature extraction stage (based on fourier coefficients of one dimensional image columns) and a neural network of the self organizing map type (SOFM) for feature classification. The classification results were documented by three aspects including the classification confusion matrix, the inspection of the NMSE (normalized mean square error), and an investigation of the resulting Kohonen map. The classification rate amounted to 80% correct classifications, the rest differed from the correct grade by one and their results were not worse than the human exports error (Bahlmann et al., 1999).
Because of the special property of the knitted fabric which is very easy to be pleated, puckered or distorted in stitching, automatic inspection of stitching is necessary. Yuen et al., 2009 proposed a hybrid model (integration of genetic algorithm and neural network) to classify garment defects. Firstly, to process the garment sample images captured by digital camera, they used a morphological filter and a method based on genetic algorithms to find out an optimal structuring element. They also presented a segmented window technique to segment images into pixel blocks under three classes using monochrome single-loop ribwork of knitted garments caused by stitching (seams without swing defects, seams with pleated defects and seams with puckering defects). Four characteristic variables (size of the seams and defective regions, average intensity value, standard deviation and entropy value) were collected to describe the segmented regions and input into back propagation neural network to provide decision support in defect classification. The number of the nodes was set as 10 by many experiments. The training function of the neural network was a gradient-descending method based on momentum and an adaptive learning rate. The learning function of connection weights and threshold values was a momentum-learning method based on gradient descending. Twenty two images of each class were used as training samples and the other ten images were testing samples. They did not report any misclassified sample and the identification rate was 100% (Yuen et al., 2009).

3. Yarn and fabric properties prediction and modeling

The main objective of many scientific studies in textile is to reveal the complex functional relationships that exist between structural parameters of fiber, yarn and fabric properties. If the relationships between different parameters that determine the specific yarn or fabric property are known, they can be used to optimize that particular property for different end-use applications so as to minimize the cost. Predictive modeling methodologies, which are

complex and inherently nonlinear, can be used to identify the different levels of combinations of process parameters and material variables that yield the desired fabric property. Since the network can accurately capture the nonlinear relationships between input and output parameters, they have extremely good predictive power (Behera & Muttagi, 2005). The use of an artificial neural network model as an analytical tool may facilitate material specification/selection and improved processing parameters governed by the predicted outcomes of the model (Khan et al., 2002).
An ANN model adjusts itself to establish the relation between the input and the output. In spite to this, an ANN model does not require any explicit formula but instead it is an implicit model by itself where it can be trained to adopt and adjust itself to perform certain tasks (Nirmal, 2010).

3.1 Mechanical behavior prediction of textiles

Breaking elongation properties of yarns influence the performance of them during winding, warping, and weaving. Yarn elongation like other yarn properties is chiefly influenced by fiber properties, yarn twist, and yarn count. Because there is a strong correlation between yarn elongation and loom efficiency, it would be very helpful if a prediction model could forecast yarn elongation accurately (Majumdar & Majumdar, 2002). Furthermore, breaking strength of yarn is the one of the most important physical property of yarn as it is the main parameter for physical quality control. It takes a long time for the yarn producer to get the experimental results for the physical properties of yarn. Therefore, faster determination of yarn physical properties is needed (Dayik, 2009). Generally, modeling and prediction of yarn properties based on fiber properties and process parameters have been considered by many researchers such as mechanistic models, statistical regression models (Gharehaghaji et al., 2007). In recent years, artificial neural network models have been widely used to predict different kind of yarn and fabric mechanical properties based on process parameters and fiber and yarn parameters. Among the various kinds of learning algorithms for the neural network, back propagation is the most widely used.
Majumdar & Majumdar, 2004 predicted the breaking elongation of ring cotton yarns by three modeling methodologies including mathematical, statistical and artificial network by back propagation learning algorithm. 72 and 15 samples, respectively, were used for training and testing the three prediction models. They tried five different network structures with one hidden layer by different number of neurons (6, 8, 10, 12, and 14) in the hidden layer. Learning rate and momentum were optimized at 0.1 and 0.0, respectively. The neural network with ten nodes in the hidden layer had the best prediction results in the testing sets after 2500 iterations. Inputs to these models were constituent cotton fiber properties (fiber bundle tenacity, elongation, upper half mean length, uniformity index, micronaire, reflectance degree, and yellowness) measured by high-volume instruments (HVI) along with yarn count (Ne). They used statistical parameters such as the correlation coefficient (R) between the actual and predicted breaking elongation, mean squared error, mean absolute error (%), cases with more than 10% error, maximum error (%), and minimum error (%) to judge the predictive power of various models and concluded that neural network model had showed the best prediction results. The correlation coefficient between actual and predicted elongation was R=0.938 for the ANN model, R=0.731 for the mathematical model and R=0.870 for the statistical model. Percent of maximum error was also reported for ANN, mathematical and statistical models which were 13.23%, 34.04%, and 15.60% respectively. The only output of each prediction model was the breaking elongation

of yarns. They also measured the relative importance of various cotton fiber properties using neural network model (Majumdar & Majumdar, 2004).

Behera & Muttagi, 2005 compared the ability of three modeling methodologies based on mathematical, empirical and artificial neural network based on radial basis function (RBF) (using orthogonal least square learning procedure) to predict fabric properties. The inputs to the network were fabric constructional parameter, yarn bending rigidities and outputs were fabric initial tensile moduli. Before feeding to network, the input-output data set was scaled down to be within (0, 1), by dividing each value by the maximum value of the overall data. Data were randomly divided into 14 sets and 4 sets of input-output pairs for training and testing the network respectively. They also studied the effect of network design parameters on error of prediction. The effects of neurons number of the hidden layer, error goal, and bias constant on prediction performance of RBF network were assessed. They observed that ANN model produced the lease error as well as minimum range of error as compared to the other modeling methods and ANN required a much smaller data set than the one required for conventional regression analysis. For example, percentage prediction error for warp and weft way fabric tensile modulus were respectively 10.2% and 8.63% for ANN, 20.4% and 12.33% for empirical model and 20.53% and 13.65% for mathematical model. They also predicted bending rigidity of woven fabric by these three models and ANN had a better and accurate result than those two models (Behera & Muttagi, 2005).

Gharehaghaji et al., 2007 investigated tensile properties modeling of cotton-covered nylon core yarns by artificial neural networks based on back propagation algorithm and multiple linear regression methods which the first method had better performance than the second. They predicted breaking strength and breaking elongation simultaneously as output and by using count of core part, count of sheath part, twist factor of core-spun yarn and pretension as input. In order to eliminate the effect units of input and output parameters, data normalizing was carried out. The data set of 54 samples was divided randomly into 5 subsets, each containing 10 or 11 samples, to train and test the network five times by using four sets as training set and one subset as testing set. Overfitting was prevented by using weight decay technique. The adaptive learning rate with momentum training algorithm (optimized at 0.9) was used to enhance the training performance. They determined the number of hidden neurons and the number of hidden layers by trial and error by using 20 topologies with different number of hidden layers and numbers. Their results showed a two hidden layers by eight nodes into first hidden layer and six nodes into second hidden layer gave the best topology. They assessed their models using verifying mean square error (MSE) and correlation coefficient (R-value). The difference between the MSE value of two models for predicting breaking elongation and breaking strength of testing data were 0.119 and 0.365 respectively (Gharehaghaji et al., 2007).

Dayik, 2009 determined the breaking strength of 100% cotton yarn properties by using Gene expression programming, neural network and classical statistical approach (multiple regression algorithms) and compared the predictive power of them by correlation coefficient (R-square) and mean square error (MSE). The inputs were included foreign matter, micronaire, uniformity, elongation, strength of fiber, length of fiber, short fiber index and neps which were collected for a three month period data. He used seven different neural network architectures which were including multilayer perception, Generalized feed forward, Modular network, Jordan/Elman, Self organizing map, Principal component and Recurrent network to identify the best one. However the best results were obtained from the generalized feed forward neural network algorithms. He examined the predictive power by

multiple linear regression analysis. The statistical method showed very much worse performance than genetic and neural network since physical properties of yarn depends on many various factors and the relations between these factors are highly nonlinear and complex. Performance of genetic model (98.88%) was better than artificial neural network (94.00%) in his research (Dayik, 2009).

The effects of splicing parameters, fiber and yarn properties on the tenacity and elongation of spliced yarns were investigated by Unal et al., 2010 using artificial neural network (ANN) and response surface model (RSM). In the ANN analysis, a multilayer feed-forward network with one hidden layer trained by back propagation algorithm was used. In the first phase, the back propagation algorithm was applied for 100 epochs. The optimum learning rate of 0.01 and momentum coefficient of 0.3 used in back propagation was determined in terms of several trials. In the second phase of training, 500 epochs were performed for conjugate gradient descent algorithm. As activation functions, a hyperbolic function was used in the hidden layer and linear functions were used in the input and output layers. Of the 89 yarn samples, 76 samples were chosen as the training set at random, while 22 samples (25%) were chosen for the testing set.

They produced yarns from eight different cotton types, having three different counts and three different twist coefficients. Six parameters including fiber length, fiber diameter, yarn count, yarn twist, opening air pressure and splicing air pressure in the input layer were selected and a neural network with seven hidden neurons for yarn tenacity analysis and another neural network with six parameters including fiber length, short fiber content, yarn count, yarn twist, opening air pressure and splicing air pressure in the input layer and six hidden neurons for breaking elongation were determined as well. The results of the ANN analysis were similar to the results of RSM except for the effect of splicing air pressure and ANN showed more powerful results in comparison RSM model since it is more capable of explaining non-linear relations (Unal et al., 2010).

ANN appears to be a reliable and useful tool in characterizing the effect of some critical manufacturing parameters on the seam strength of webbing, if a sufficient number of replicated experimental data are available to train the ANN. Onal et al., 2009 studied the effect of fabric width, folding length of joint, seam design and seam type on seam strength of notched webbings for the parachute assemblies using both Taguchi's design of experiment (TDOE) and an artificial neural network (ANN) and then compared them with strength physically obtained from mechanical tests on notched webbing specimens. They used a four layer, feed forward, back propagation ANN model with a five hidden layer neurons and one output neuron to output seam strength. Input variables were fabric width, folding length of joint, seam design and seam type. 60 training patterns and 10 testing patterns were used to train and test the network. It was established from these comparisons, in which the root mean square error was used as an accuracy measure, that the predictions by ANN were better in accuracy than those predicted by TDOE (Onal et al., 2009).

Hadizadeh et al., 2009 presented an ANN model for predicting initial load-extension behavior of plain weave and plain weave derivative fabrics. They developed a single hidden layer feed forward ANN based on a back propagation algorithm with four input neurons (using a combination of parameters of Leaf's equation instead of individual parameters) and one output neuron to predict initial modulus in both warp and weft directions. In their research, the input and measured values were normalized so that they would have zero mean and unity standard deviation and they used Levenberg-Marquardt learning algorithm. Five different cases of ANN with different number of neurons in hidden layer

and different data were considered to train and test the network. The number of neurons in the hidden layer was experimentally verified on the basis of the performance factor. In case one, 18 Leaf and Kandil's data were inputted to the network. In case two, the ANN was consist of 31 samples of plain weave experimental values of produced fabrics. They used their data in conjunction Leaf's data in case three. In case four, their fabric samples of plain weave and plain weave derivatives were considered while in case five, Leaf's data in addition to their data were applied to network. The model's suitability was confirmed by the low performance factor (PF/3) and the high coefficient of correlation. Their proposed ANN model was suitable for the prediction load-extension behavior of plain weave and plain weave derivatives of fabrics (Hadizadeh et al., 2009).

Shear stiffness is one of the important properties of worsted fabrics which depends on yarn properties and fabric parameters. As a nonlinear problem, predicting the shear stiffness can be realized by an alternative modeling method, that is, by using the artificial neural network (ANN) model. Chen et al., 2009 modeled the relationship between yarn properties, fabric parameters, and shear stiffness of worsted fabrics using two stage neural network models. First, the yarn properties and fabric parameters were selected by utilizing an input variable selection method to find the most relevant yarn properties and fabric parameters as the input variables to fit the small-scale artificial neural network model. The first stage was consisting two parts. The first part took the human knowledge on the shear stiffness into account (VAk) and the second part was a data sensitivity criterion based on a distance method (Sk). Second, the artificial neural network model of the relationship between yarn properties, fabric parameters, and shear stiffness of fabrics was established.

They used a feed forward ANN by six yarn properties and fabric parameters (warp cover factor, warp twist factor, weft twist factor, warp linear density, weft linear density, and fiber specific surface area) as inputs, one hidden layer with four neurons, and shear stiffness of fabrics as output trained with the help of the error back propagation algorithm. In order to avoid overfitting, the Bayesian framework were used in the training procedure. 39 data points and 1 data point were used for training and testing set respectively. They used the primitive variables to rank data, not their transformations as those in the PCA. Hence, the variables had clear physical meanings. Their results showed accurate prediction (up to average error of 0.209%) by the small-scale artificial neural network model and a reasonably good artificial neural network model could be achieved with relatively few data points by integrating with the input variable selecting method developed in their research (Chen et al., 2009).

Needle punching is a well-known nonwoven process of converting fibrous webs into self-locking or coherent structures using barbed needles. The barbed needles pull the fibers from the surface of web and reorientate them in the thickness direction leading to a complex three-dimensional (3D) structure. The nonwoven structural depends on different parameters. Rawal et al., 2009 predict the bulk density and tensile properties of needle punched nonwoven structures from main process parameters including web area density, depth of needle penetration, and punch density by Artificial Neural Network (ANN) modeling technique (back propagation learning algorithm). Two different ANN models were developed, one for predicting fabric bulk density and another for predicting the tensile strength in the machine and cross machine directions. Only one hidden layer with 8 nodes was used and transfer function in the hidden and output layers was log-sigmoid. Learning rate and momentum was optimized at 0.6 and 0.8 respectively. Web area density, punch density, and depth of needle penetration were considered as inputs. Training was stopped when the error in the unseen or testing data sets approached at the minimum level. 21 data

sets were randomly chosen for the training of ANN and 6 sets were used for the testing purpose. The simultaneous effect of more than one parameter on bulk density and tensile properties of needle punched nonwoven structures have been investigated based upon the results of trained ANN models. A comparison was also made between the experimental and predicted values of fabric bulk density and tensile strength in the machine and cross machine directions in unseen or test data sets. It has been inferred that the ANN models had achieved good level of generalization that is further ascertained by the acceptable level of mean absolute error obtained between predicted and experimental results (Rawal et al., 2009).

3.2 Prediction of the other textile properties

The material properties of engineering fabrics that are used to manufacture airbag can not be modeled easily by the available nonlinear elastic-plastic shell elements. A nonlinear membrane element that incorporates an elaborate tissue material model has been widely used by the auto industry for the airbag simulation studies, this model is highly computation intensive and does not differentiate between the various physical properties of the fabrics like fiber denier, the polymer fiber, and weave pattern. Keshavaraj et al., 1996 introduced a feed forward neural network to determine permeability and biaxial stress-strain relationships for nylon and polyester fabrics used in airbags. The network used for permeability prediction was a three input nodes (281, 323, and 373 K temperature levels), four hidden nodes and one output node. The predictions provided by the neural network model were better for the polyester fabrics than they were for the nylon fabrics. The effects of the type of fabrics, i.e., denier and weave type, with nylons and calendering in case of polyester, biaxial strain, biaxial stress, and pressure drop while predicting biaxial stress of fabric under a biaxial deformation were considered in their model. The model prediction was within a ±3 MPa error limit which was agreed very well with the experimental data (Keshavaraj et al., 1996).

Classical pressure drop models set up for porous media do not accurately model pressure drops through fabric structures but they give information about the location of flow through fabric structures and about the specific characteristics of cloths which may influence pressure drop values. A neural network (NN) approach is then carried out in order to model experimental data by taking into account specific characteristics of cloths as input neurons, and to analyze the relative importance of each input variable on pressure drops. Brasquet & Cloirec., 2000 studied pressure drops through several textile fabrics using classical models (Ergun's equation, Carmen's dimensionless approach, and Comiti-Renaud's model), statistical tool, and neural network. The models were tested by three different definitions for the specific surface area, on the fabric, yarn, and opening scale, respectively. Different kinds of cloth were used, in terms of fiber type such as activated carbon fibers and their precursors, rayon fibers. In the first part, they measured air and water pressure drops induced by these different cloths as a function of fluid velocity experimentally and secondly, using classical models set up for particular media in order to locate the flow and then a statistical approach by neural network were considered. They chose input neurons in a multilayer perceptron network (fluid properties-μ, ρ, Re- and fabric characteristics – thickness, density, number of openings and raw material) in order to predict pressure drop values as the output neuron. The number of hidden neurons was statistically optimized as four with hyperbolic tangent function as transfer function. Network training set was carried out with 400 data and a validation set of 183 data was also performed. 200 data was used to

test the generalization ability of the trained neural network. They calculated absolute averaged relative errors (AARE) to assess the performance of their network (Brasquet & Cloirec., 2000).

The work 'bagging' is essentially a perception by people of the three dimensional shape of a bagged garment. Subjective judgments of the degree of garment bagging vary with different people and also depend on garment types. Garment bagging is a kind of three-dimensional residual deformation during wear, which can be characterized by a few parameters such as bagging height, volume, shape, and fabric surface pattern. Yeung et al., 2002 developed a method to evaluate garment bagging by image processing with three different modeling including multiple regression, liner modeling and neural network. These models were able to provide predictive powers of R^2 value of 0.92, 0.93, and 0.94 respectively. Firstly, they evaluated fabric bagging by capturing digitized images of bagged fabrics, image processing of the capture images, and recognizing bagging magnitude from these criteria. They used the eight criteria as input variables to predict subjective perceptions of bagging, employing a two-layer feed forward neural network with back propagation learning algorithm. The hidden layer included thirteen neurons with tan-sigmoid transfer functions to learn nonlinear and linear relationships between input and output. Ten samples and six samples were used to train and test the network respectively. The ability of network to predict bagging was reported R^2=0.94 (Yeung et al., 2002).

Tokarska, 2004 presented modeling of woven fabric permeability (dynamic air permeability) features by means of neural network (multilayer perceptron). His analysis of the flow properties was based on observations of their behavior during impact air flow. He used apparent density, warp twist, and weft twist for the input layer, while the output layer was the integral of the function $p(t)$, that is the actual pressure impulse generated on a fabric under impact air flow conditions, using a back propagation method to teach the network. He obtained the quality of his neural model by means of an index ρ, which is given the standard deviation of errors for the output variables divided to the standard deviation of the target output variable (Tokarska, 2004).

Fabric hand is commonly used for assessing fabric quality and prospective performance in a particular end use. Subjective assessments treat fabric hand as a psychological reaction obtained from the sense of touch, based on the experience and sensitivity of humans. Prediction of these psychological perceptions of hand based on fabric properties is very difficult. Hui et al., 2004 predicted sensory hand based on fabric properties using a resilient back propagation multilayer feed forward neural network. Twelve fabric properties were fed into the input layer then they propagated forward through two hidden layers and then fourteen biopolar pairs of sensory fabric hand attributes arrived at the output layer. The output was normalized since the log sigmoid activation functions were used on each layer. Mean square error (MSE) was set to 1e-8 and to avoid network over fitting, they used Larsen's early stopping methodology to reduce the generalization error of the network. Correlation between output and target values were reported greater than 0.9 (Hui et al., 2004).

There are numerous factors which broadly classified into yarn quality, condition of warp preparation, and loom actions and conditions which can affect the performance of warp yarns in weaving. The weaving performance of a yarn is generally expressed in terms of warp breakage rate in weaving. Yao et al., 2005 investigated the predictability of the wrap breakage rate from a sizing yarn quality index using a feed forward back propagation network. They rated an eight quality index including size add-on, abrasion resistance, abrasion resistance irregularity, hairiness beyond 3 mm, breaking strength, breaking

strength irregularity, breaking elongation and breaking elongation irregularity as input layer and warp breakage rates as output layer in controlled conditions. Hidden layer with 1, 4, and 8 neurons were tested. The learning method was back propagation with momentum, and single step learning with a sequential presentation sequence was selected as learning strategy. They suggested a model with a single sigmoid hidden layer with four neurons to produce better predictions than the other models and prepared sixty records for training and ten records for testing the network. The stop condition had been mean absolute error (MAE= 0.148), mean square error (MSE= 0.0364), root mean square error (RMSE= 0.191), and mean absolute percentage error (MAPE= 5.58) and correlation was reported R= 99.5%. (Yao et al., 2005).

Comfort is one of the most important attributes of textiles used in clothing. Clothing comfort is influenced by different fabric, environment and human factors. Thermal properties of clothing are one of the most important aspects of clothing comfort in which analyzing the relationship between various fabric parameters and comfort properties are essential. Bhattacharjee & Kothari, 2007 reported a study on the predictability of the steady-state and transient thermal properties of fabrics (thermal resistance and maximum instantaneous heat transfer) using a feed forward, back propagation artificial neural network system. They made a comparison with two different network architectures, one with two sequential networks working in tandem fed with a common input and another with a single network that gave two outputs and the first one (mean error percentage of 8.61%) gave better results than the second one (mean error percentage of 10.42%). First model was able to predict the steady-state and transient thermal behavior with a good coefficient of determination (R^2= 0.94) as compared with the second model (R^2= 0.69). A three layered network with two hidden layers was used in both of the cases. The input parameters including type of weave warp and weft count, thread density, thickness and areal density were considered. A sigmoid transfer function 'tansig' was used for input and hidden layers and a linear function was used for the output layer. The training function used was a quasi-Newton algorithm based on the Broyden, Fletcher, Goldfarb, and Shanno (BFGS) update and regularization was carried out to avoid over fitting (Bhattacharjee & Kothari, 2007).

Capillary rise in porous media is a frequently occurring phenomenon which occurs in dyeing of textile fabrics, and a variety of other fields. An artificial neural network was employed by Ahadian et al., 2007 to predict the time of capillary rise for a known given height. Their network's inputs were density, surface tension, and viscosity for the liquids and particle size, bulk density, packing density, and surface free energy for the powders. The output layer of the network corresponded to the time of capillary rise in order to reach a given height (i.e. 0.036 m). A training set (136 times of capillary rise) and a testing set (18 times of capillary rise) was chosen for the network. Networks were trained using the Levenberg- Marquardt back propagation algorithm. A linear activation function was used in output layer of the networks. All the input and output data were normalized to the interval [-1 to 1] before training and testing. Two statistical parameters namely the product moment correlation coefficient (r^2) and the performance factor (PF/3) were used to correlate the actual experimentally obtained times of capillary rise. The results showed that their artificial neural network was able to predict the time of capillary rise (i.e. r^2 = 0.91, PF/3=55). In comparison, the Lucas-Washburn's calculations gave the worst correlations (r^2 = 0.11, PF/3 = 1016) (Ahadian et al., 2007).

Furthermore, thermodynamic and transport properties of liquids are fundamental in processes involving liquid flow and heat and mass transfer. Two most important of these

properties are surface tension and viscosity which are modeled using two artificial neural networks (ANNs) by Ahadian et al., 2008. The surface tension predictor network had six inputs, namely: particle size, bulk density, packing density and surface free energy of the powders as well as the density of the probe liquids together with the capillary rise time of the liquids in the corresponding powders. The viscosity predictor network had surface tension as an extra input. The results of the present work clearly showed that the artificial neural network approach is able to predict the surface tension (i.e. r2 = 0.95, PF/3 = 16) and viscosity (i.e. r2 = 0.998 , PF/3 = 13) of the probe liquids with unsurpassed accuracy (Ahadian et al., 2008).

There is a need for a reliable forecasting system which can quantitatively predict the hairiness of a resultant yarn from its processing parameters prior to yarn formation. The development of such a system is potentially challenging owing to the complex nature of the worsted spinning pipeline, where wool fibers undergo a series of different processes before being converted into a yarn. Khan et al., 2009 evaluated the performance of multilayer perceptron (MLP) and multivariate linear regression (MLR) models for predicting the hairiness of worsted-spun wool yarns from various top, yarns and processing parameters. Their results indicated that MLP model predicted yarn hairiness more accurately than the MLR model. They used some factors including yarn twist, ring size, average fiber length, fiber diameter and yarn count on the basis of sensitivity analysis as inputs. Five different random partitions of the database into training and validation sets were generated. For each partition, both models were independently trained using the training set and their responses to the validation set assessed. Both MLR and MLP models were capable of achieving a good fit to the measured hairiness values, as evidenced by the high mean R2 values of 0.910 and 0.949, respectively. This study also demonstrated that the hairiness of a yarn could be predicted to a high precision from limited top, yarn and processing parameters, and that the ANN-based yarn hairiness prediction model had the potential for wide mill specific applications (Khan et al., 2009).

One of the most important properties of clothes is their ability to help the body's thermal system to keep the body temperature in its natural range, even if the environmental conditions or physical activities are outside the body's ideal range. Perspiring is one of the most important effects of physical activities in warm weather for shedding the body's excessive heat. Therefore, the basic requirement of a fabric worn next to the skin is to transfer this moisture to the atmosphere to reach comfort through the avoidance of a feeling of wetness and clamminess and also through the generation of a situation for the best surface evaporation of moisture. Mokhtari Yazi et al., 2009 evaluated the transmission of heat and moisture by differential modeling as an artificial neural network a double-surface knitted fabric containing hydrophilic and hydrophobic fibers. Input data was made from temperature and moisture values for the bottom and top surfaces of the fabric; the number depended on the time of each experiment and was different for each sample. The connections of network nodes corresponded to the partial differential equation of propagation as forward time-centered spaces (three advanced Euler methods). The results were analyzed to find a suitable fabric with optimum comfort. The final results showed that a fabric made of micro polyester filaments and cotton yarns on the bottom and top surfaces, respectively, had the best heat and moisture transfer (Mokhtari Yazi et al., 2009).

Giri Dev et al., 2009 modeled and predicted water retention capacities of the membranes under different hydrolyzing conditions using empirical as well as artificial neural network (ANN model) by alkali concentration, temperature and time as inputs. Both statistical model

and ANN model had showed a very good relationship (R^2) between the experimental and predicted response values and both models had an error percentage less than 2% indicating the reliability of the model developed (Giri Dev et al., 2009).

Needle punching is a well-known nonwoven process of converting fibrous webs into self-locking or coherent structures using barbed needles. Rawal et al., 2009 predict the bulk density and tensile properties of needle punched nonwoven structures from main process parameters including web area density, depth of needle penetration, and punch density by Artificial Neural Network (ANN) modeling technique (back propagation learning algorithm). Two different ANN models were developed, one for predicting fabric bulk density and another for predicting the tensile strength in the machine and cross machine directions. The number of nodes in the hidden layer and learning parameters, i.e., learning rate and momentum was optimized at 8, 0.6, and 0.8, respectively. Training was ceased when the error in the unseen or testing data sets approached at the minimum level. Out of 27 available data sets, 21 sets were randomly chosen for the training of ANN and remaining six sets were used for the testing purpose. The simultaneous effect of more than one parameter on bulk density and tensile properties of needle punched nonwoven structures have been investigated based upon the results of trained ANN models. A comparison was also made between the experimental and predicted values of fabric bulk density (R= 0.907) and tensile strength in the machine (R= 0.986) and cross machine directions (R= 0.982) in unseen or test data sets. It has been inferred that the ANN models had achieved good level of generalization that is further ascertained by the acceptable level of mean absolute error obtained between predicted and experimental results (Rawal et al., 2009).

Bio-composite materials are gaining high popularity due to its various advantages such as renewable, biodegradable, low in cost, light weight, low density, widely available and possess high specific mechanical properties. Nirmal, 2010 predict frictional performance of treated betelnut fiber reinforced polyester (T-BFRP) composite using artificial neural network configuration. To predict the friction coefficient of the T-BFRP composite, the ANN model was subjected to three different input parameters; normal loads (5–30 N), sliding distances (0–6.72 km) and fiber orientations (anti-parallel, parallel and normal orientations). Prior to inputting the data to the ANN network, data coding was performed to the input parameters. Network had a two hidden layer with 10 neurons in the first hidden layer followed by 20 neurons in the second hidden layer. The learning process of a developed ANN model was based on a gradient search with least preferred sum squared errors between the predicted and the actual values. He considered the trial and error ANN model based on method where various neuron configuration, layer configuration and transfer function configuration. Results obtained from the developed ANN model were compared with experimental results. It was found that the experimental and numerical results showed good accuracy when the developed ANN model was trained with Levenberg– Marqurdt training function (Nirmal, 2010).

4. Process behaviour prediction

Yarn properties and spinning performance are influenced by fiber properties (mean fiber diameter, mean fiber length, diameter distribution, fiber strength, and etc), yarn specifications (linear density, twist level), and operational parameters (ring size, traveler weight, spinning speed). Because there are many independent variables, it becomes difficult to cover the entire range of parameters in order to interpolate and extrapolate experimental

observations or mill measurements and take into account the interactive contribution of each independent variable. Unlike conventional techniques, which are often limited by strict assumptions of normality, linearity, and variable independence, ANN's are universal approximators, which, by possessing the capacity to learn directly from the data being modeled, are able to find associations or discover regularities within a set of patterns, where the volume or variation within the data is large or the relationships between variables are dynamic and nonlinear. For a given fiber spun to pre-determined yarn specifications, the spinning performance of the yarn usually varies from mill to mill. For this reason, it is necessary to develop an empirical model that can encompass all known processing variables that exist in different spinning mills, and then generalize this information and be able to accurately predict yarn quality for an individual mill (Beltran et al., 2004).

The degree of spinnability of a fiber is very difficult to assess with the current range of instruments available. Pynckels et al., 1995 described an experiment of 29 fiber properties of twenty types of cotton to predict spinnability of fibers. A yarn was considered to be unspinnable if there were more than five breakages during the first three minutes of spinning. They trained a neural network with 700 spinnable and 700 unspinnable yarns data to predict the spinnability from fiber properties and process parameters. In the test data set, 90% of spinnable fibers and 95% of the unspinnable fibers was classified correctly (Pynckels et al., 1995).

Beltran et al., 2004 reported a method for predicating worsted spinning performance with an artificial neural network trained with back propagation learning rule. The applicability of ANN for predicting spinning performance was first evaluated against a well established prediction and benchmarking tool. The ANN was then subsequently trained with commercial mill data to assess the feasibility of the method as a mill specific performance prediction tool. Incorporating mill specific data resulted in an improved fit to the commercial mill data set. Top properties, yarn specifications, and processing information were designated as the input vectors for the input layer. They found that as the number of mill-specific data sets increased, further improvements in prediction accuracy would arise (Beltran et al., 2004).

Mean fiber diameter, diameter distribution, hauteur, fiber length distribution, fiber bundle tenacity, curvature, short fiber content, yarn count, twist, draft, spinning speed, ring size, and traveler weight served as inputs to the neural network and the number of fibers in a cross section, unevenness CV%, unevenness U%, thin places per kilometer, neps per kilometer, yarn tenacity, elongation at break, breaking force, end-down per 1000 spindle hours, index of irregularity, thick places per kilometer, and hairiness served as the target spinning performance outputs. A total of 250 sets of training data were randomly generated. The first 180 data sets were used for network training, 20 data sets were set aside for cross-validation, and the last 50 data sets were used to evaluate the trained network's performance. The input data were normalized so those were bounded within the prescribed range of 1 and 0. They tested different numbers of neurons in hidden layer and indicated that a reduction in the training error occurred as the number of hidden nodes increased. To overcome the likelihood of over-fitting from excessive training, they invoked the cross-validation stop criteria. They observed that the cross validation mean squared error exponentially fell to $6.0 * 10^{-3}$ over 800 training epochs. Therefore 800 epochs represented the point where sufficient training had occurred prior to over fitting of the specific solutions within the training set. By incorporating mill-specific data results in an improved fit to the

commercial mill data set, suggesting that their proposed method had the ability to predict the spinning performance of a specific mill accurately (Beltran et al., 2004).

Melt spinning is the most economically useful method for producing artificial fibers in the industry. In melt spinning, as-spun fibers are evaluated according to yarn count and tensile strength, and the draw ratio is the major factor affecting the quality of as-spun fibers. The neural network computation can be divided into two parts: pre-teaching computation and reversing the adjusted weight value. Kuo et al., 2004 considered the extruder screw speed, gear pump gear speed, and winder winding speed of a melt spinning system as the inputs and the tensile strength and yarn count of as-spun fibers as the outputs for neural network by the delta learning rule. The data from experiments were used as learning information for the neural network to establish a reliable prediction model. They had adopted a three layer neural network consisting of a three neuron input layer, a twelve neuron hidden layer, and a two neuron output layer; focusing on the tensile strength and yarn count of as-spun fibers. They applied the delta learning rule to the neural network, with a sigmoid transfer function. The neural network prediction model was verified by ten entries of new data. In tensile strength prediction, the error of the neural network was ±2%. When compared with one standard difference of the experiment, 96.86% of the predictive values lied within ±1 σ=3.1419%. In yarn count prediction, the error of the neural network was ±2%. When compared with one standard difference of the experiment, 97.96% of the predictive values lied within ±1 σ=2.0418%. Their neural network model could predict the tensile strength and yarn count of as-spun fibers to provide a very good and reliable reference for as-spun fiber processing (Kuo et al., 2004).

Meltblowing has become an important industrial technique because of its ability to produce fabrics of microfiber structure, which are ideally suited for filtration media, thermal insulators, battery separators, and oil sorbents. In this process, the fiber forming mechanism is very complicated and the quality of the produced web depends on many processing variables such as die temperature, air temperatures, air flow rate, extruder temperature, die to collector distance, polymer throughput rate, resin melt flow rate, die geometry parameters and etc. therefore meltblowing is a highly complex, multivariable, and nonlinear process, leading to the extreme difficulty in theoretically modeling the process. However, process modeling is essential for the control of optimization and an on-line prediction is very useful for process monitoring and quality control. Melt blown process is of highly dimensional and nonlinear complexity. Sun et al., 1996 investigated back-propagation neural networks (BPNNs) for modeling the melt blown process and on-line predicting the product specifications such as fiber diameter and web thickness. By comparison of several network topology structures (6-3-1, 6-4-1, 6-5-1, 6-6-1, 6-4-3-1, etc) and different transfer functions (sigmoid, quadratic), the network 6-4-1 (i.e. six nodes in the input layer, four nodes in the hidden layer and two nodes in the output layer) was chosen using a sigmoid function as its transfer function. The network inputs were included extruder temperature, die temperature, melt flow rate, air temperature at die, air pressure at die, and die-to collector distance (DCD) and they were normalized. The output of the fiber diameter was obtained by neural computing. The network training was based on 160 sets of the training samples and the trained network was tested with 70 sets of test samples which were different from the training data. The test results showed a good agreement to the actual measurements. The maximum absolute error between the predicted fiber diameter and the actual values was less than 1.5 μm. By using the tested neural network, they also predicted the effect of process variables on the fiber diameter. The most valuable result of their

research was development of a technique which had been proved to be suitable for modeling and on-line preidicting of the meltblowing process in order to optimal control of the process and of practical significance to advanced meltblown processes (Sun et al., 1996).

5. Color coordinates conversion, color separation and categorization, color matching recipe prediction

One of the most important textile characteristics is undoubtedly color (Thevenet et al., 2002). Color quality control is one very important step in any textiles, however excellent the fabric material itself is, if it lacks good color, then it may still result in dull sale (Kuo et al., 2007). Many transformations affect the color of textile materials. Nevertheless, they can be divided into two groups. The first group concerns dyeing and printing stages, and is mainly governed by chemical rules, because the color attributes, which are added to the textile structure, are chemically fixed to the product (Thevenet et al., 2002). Expected depth of shade, color, color fastnesses and surface characteristics etc. are very important qualities which are necessary to be achieved in the dyed goods. If, these properties are different from that of the expected standard, the product has to either been reprocessed or discarded (Balci et al., 2008). Color separation is most important item in pattern printing process so as to secure integrity of printed fabrics product (Kuo et al., 2007). Furthermore, in textile printing it is very difficult to control all the process parameters; therefore using artificial neural networks for recipe calculation (concentration of each dye in the printing paste) have been investigated which enable the relationship between reflectance values and concentrations to be mapped.

Once the selected neural network is sufficiently trained with a set of known input (colour values) and output data (concentrations of each dye), it will predict the concentrations for an unknown set of coloured samples. One of the advantages of neural networks is their capability to establish relations between input and output data without explicit programming of Kubelka-Munk equations or analytical knowledge into the model (Golob et al., 2008).

The second group concerns blending and the transformations of structure of roving (assembly of fibers), which is spun and then woven or knitted. In this case, the color transformation is not governed by chemical rules, because during blending or spinning, no chemical compounds are added. So this group is rather physically governed, because color alterations are just produced by a different fibers organization. The aim of the model is to predict the color obtained when fibers, with different colors, are blended. When the blend is homogenous, the color obtained can be predicted very well by theoretical and empirical models (Thevenet et al., 2002).

Thevenet et al., 2002 described a model based on neural networks to predict color alteration after spinning process (roving to yarn). Their network was a multilayer feed-forward network. The first system using to predict the entire reflectance spectra was wavelength dependent, but its performance is not very satisfactory. The scaled conjugate gradient algorithm was incorporated into the back propagation procedure to reduce the training phase. Once the wavelength independence of the transformation was established, a second system, whose performances agree with the experimental curves, was proposed (Thevenet et al., 2002).

Kuo et al., 2007 proposed a printed fabrics computerized color separation system based on backward-propagation neural network, whose primary function was to separate rich color

of printed fabrics pattern so as to reduce time-consuming manual color separation color matching of current players. What it adopted was RGB color space, expressed in red, green, and blue. Genetic algorithm was used to pre-process as the first phase of operation process. A gene algorithm was used to find smaller sub images alternative of original fabric in color distribution, for later color separation algorithm use to reduce the operation of color separation. In order to find sub-images with same color distribution as original image, they adopted Histogram Intersection to measure color similarity of sub-image and original image. In terms of color separation algorithm, their research relied on supervised backward-propagation neural network to conduct color separation of printed fabrics RGB sub-image, and utilized PANTONE® standard color ticket to do color matching, so as to realize accurate color separation (Kuo et al., 2007).

Balci et al., 2008 presented an artificial neural network (multilayer perceptron) modeling by Levenberg-Marquardt (LM) algorithm for predicting the colorimetric values of the stripped cotton woven fabrics dyed using commercial reactive dyes. They used 90 different network structures with 15 different number of nodes in the hidden layer, 3 level of inputs (10 inputs, 7inputs, and 6 inputs) and 2 level of MSE value of results as stopping criteria in order to get the best fitting model to predict the L* and ΔE colorimetric values of stripped cotton samples. In order to establish these networks, they used type of the reactive dyes, type of the reducing agents, concentration of the reducing agents and caustic, working temperature and time, presence of the leveling agent and original colorimetric values (L*, a*, b*) of dyed samples measured before stripping processes as inputs, and L* and ΔE values of stripped samples measured after stripping process as outputs. After the prediction, the suitable working parameters can be chosen and the processes can be started. Therefore, this may make the stripping process for re-correction of the faulty materials more cost-effective [A5].

Golob et al., 2008 demonstrated the possibility of using counter-propagation neural networks (based on Kohonen ANN) to identify the combinations of dyes in textile printing paste formulations. An existing collection of 1430 printed samples produced with 10 dyes was used for neural network training. The reflectance values served as input data and the known concentrations of single dye or two dyes were used for printing each sample. Some variations of neural network parameters were tested to determine the best model, and a cross-validation method was used to estimate the generalization error. Also, some modifications of input and output data were made to improve the learning capabilities (Balci et al., 2008).

Metamerism is one of the most fundamental perceptual phenomena of the visual system and can be visualized when a part of colored samples in spite of having different spectral reflectance data give the same color coordinates (i. e. match in color) under one specified condition. Moradian & Amani Tehran, 2000 studied the application of artificial neural network (fully connected feed forward network) for the quantification of metamerism. Data from 98 real metameric pairs with visual assessment values were used for training (90 data set) and testing (8 data set) of the network. Many types of networks with different architecture, activation function and input were examined to achieve the best results. A network comprising of one hidden layer with 5 nodes with Tansig as the activation function provided the best prediction. The normalized L*C*H was regarded as the best-input candidate for the network. The final trained network showed a good degree of correlation with visual assessment deviating only by 20% (PF/4=20) and could therefore be a good candidate as a substitute for the previously proposed metameric indices. Metameric indices at their best, deviate by approximately 40% (PF/4=36) from visual assessments.

Fluorescent dyes present difficulties for match prediction due to their variable excitation and emission characteristics, which depend on a variety of factors. An empirical approach is therefore favored, such as that used in the artificial neural network method. Bezerra & Hawkyard, 2000 described the production of a database with four acid dyes (two fluorescent and two non-fluorescent) along with the large number of mixture dyeing that were carried out. The data were used to construct a network connecting reflectance values with concentrations in formulations. Their multilayer perceptron network was trained using back propagation algorithm. Network topology was constituted of one input layer (three nodes), one hidden layer (four nodes) and one output layer (five nodes). the networks' input layers were fed with SRF, XYZ or L*a*b* values of each sample in order to predict, in the output layer, the dye concentrations (C) applied. A linear activation function was used in the input and output layers, and the logistic sigmoid function in the hidden layers. All the data were normalized before training and testing, and all the networks were trained using the same learning rate (0.5 ® 0.01) and momentum term (0.5 → 0.1). The 311 samples produced were divided in two groups: a training set (283 samples) and a testing set (28 samples). Their results showed that, although time consuming, the presented approach was viable and accurate (Bezerra & Hawkyard, 2000).

Ameri et al., 2005 used the fundamental color stimulus as the input for a fixed optimized neural network match prediction system. Four sets of data having different origins (i.e. different substrate, different colorant sets and different dyeing procedures) were used to train and test the performance of the network. The input layer was consistent of the measured surface spectral reflectance of the target color centers at 16 wavelengths of 20 nm intervals throughout the visible range of the spectrum between 400-700 nm. The output layer was corresponded to the concentrations of the colorants. The network was trained using the scaled conjugate gradient back propagation algorithm. A positive linear activation function was used in the output layer whilst the logsig function was used in the hidden layer. Training was made to continue over 100000 epochs running three times. The results showed that the use of fundamental color stimulus greatly reduced the errors as depicted by the MSE and Δ Cave data and improved the performance of the neural network prediction system (Ameri et al., 2005).

Ameri et al., 2006 used different transformed reflectance functions as input for a fixed genetically optimized neural network match prediction system. Two different sets of data depicting dyed samples of known recipes but metameric to each other were used to train and test the network. The transformation based on matrix R of the decomposition theory showed promising results, since it gave very good colorant concentration predictions when trained by the first set data dyed with one set of colorants while being tested by a completely different second set of data dyed with a different set of colorants (PF/4 always being less than 10). The network was trained using the Levenberg-Marquardt back propagation algorithm. The error goal was fixed at MSE 0.001. All the input and output data were normalized before training and testing (Ameri et al., 2006).

6. Conclusion

Neural network technique is used to model non-linear problems and predict the output values for given input parameters. Most of the textile processes and the related quality assessments are non-linear in nature and hence, neural networks find application in textile technology.

ANN may be defined as structures comprised of densely interconnected adaptive simple processing elements that are capable of performing massively parallel computations for data processing and knowledge representation. There are many different types of neural networks varying fundamentally. The most commonly used type of ANN in textile industry is the multilayered perceptron (MLP) trained neural network. MLP is a feed-forward neural network. In most textile applications a feed-forward network with a single layer of hidden units is used with a sigmoid activation function for the units (Balci et al., 2008).
Some studies have decided the number of unites in the hidden layer upon by conducing the trail and error, or genetic algorithm or other optimizing methods and a network with the minimum test-set error is to be used for further analysis.
The number of input and output neurons depends on the type of textile problems.
Many of the techniques reported require many feature extraction procedures before the data can feed to a neural network and data is afforded by different measurements including feature extracted from images, experiments based on standards based on their own tests or other gathered measurements.
Some studies have discussed different type of pre processing and post processing methods.
Many papers have applied and compared the performance of different mathematical, statistical, or experimental models and predictions with neural network for different textile applications and in most of them, neural network models predict process, grading, or behavior of features more accurate than other methods.
The performance of the network is judged by computing the root mean square error (MSE), Sum of the square error (SSE), moment correlation coefficient (r), percentage error (%E), coefficient of variation (%CV), gamma factor (γ), performance factor (PF/4), and etc in order to analyze the results.
Since neural networks are known to be good at solving classification problems, it is not surprising that much research has been done in the area of textile classification, particularly fault identification and classification. The current 2D-based investigation needs to be extended to 3D space for actual manual inspection.

7. References

Ahadian, S.; Moradian, S.; Sharif, F.; Amani Tehran, M. & Mohseni, M. (2007). Prediction of Time of Capillary Rise in Porous Media Using Artificial Neural Network (ANN). *Iranian Journal of Chemistry and Chemical Engineering*, Vol.26, No.1, pp. 71-83, ISSN 1021-9986

Ahadian, S.; Moradian, S.; Sharif, F.; Amani Tehran, M. & Mohseni, M. (2008). Determination of Surface Tension and Viscosity of Liquids by the Aid of the Capillary Rise Procedure Using Artificial Neural Network (ANN). *Iranian Journal of Chemistry and Chemical Engineering*, Vol.27, No.1, pp. 71-83, ISSN 1021-9986

Ameri, F.; Moradian, S.; Amani Tehran, M. & Faez, K. (2005). The Use of Fundamental Color Stimulus to Improve the Performance of Artificial Neural Network Color Match Prediction Systems. *Iranian Journal of Chemistry and Chemical Engineering*, Vol.24, No.4, pp. 53-61, ISSN 1021-9986

Ameri, F.; Moradian, S.; Amani Tehran, M. & Faez, K. (2006). Use of Transformed Reflectance Function for Neural Network Color Match Prediction Systems. *Indian Journal of Fibre & Textile Research*, Vol.31, pp. 439-443, ISSN 0971-0426

Bahlmann, C.; Heidemann, G. & Ritter, H. (1999). Artificial Neural Networks for Automated Quality Control of Textile Seams. *Pattern Recognition*, Vol.32, No.6, pp. 1049-1060, ISSN 0031-3203

Balci, O.; Ogulata, S. N.; Sahin, C. & Ogulata, R. T. (2008). An Artificial Neural Network Approach to Prediction of the Colorimetric Values of the Stripped Cotton Fabrics. *Fibers and Polymers*, Vol.9, No.5, pp. 604-614, ISSN 1229-9197

Behera, B. K. & Muttagi, S. B. (2005). Comparative Analysis of Modeling Methods for Predicting Woven Fabric Properties. *Journal of Textile Engineering*, Vol.51, No.1, pp. 1-9, ISSN 1346-8235

Beltran, R.; Wang, L. & Wang, X. (2004). Predicting Worsted Spinning Performance with an Artificial Neural Network Model. *Textile Research Journal*, Vol.74, No.9, pp. 757-763, ISSN 0040-5175

Beltran, R.; Wang, L. & Wang, X. (2005). Predicting the Pilling Propensity of Fabrics through Artificial Neural Network Modeling. *Textile Research Journal*, Vol.75, No.7, pp. 557-561, ISSN 0040-5175

Bezerra, C. M. & Hawkyard, C. J. (2000). Computer Match Prediction for Fluorescent Dyes by Neural Networks. *Coloration Technology*, Vol.116, No.5-6, pp. 163-169, ISSN 1472-3581

Bhattacharjee, D. & Kothari, V. K. (2007). A Neural Network System for Prediction of Thermal Resistance of Textile Fabrics. *Textile Research Journal*, Vol.77, No.1, pp. 4- 12, ISSN 0040-5175

Brasquet, C. & Le Cloirec, P. (2000). Pressure Drop through Textile Fabrics-Experimental Data Modeling using Classical Models and Neural Networks. *Chemical Engineering Science*, Vol.55, pp. 2767-2778, ISSN 0009-2509

Chattopadhyay, R. & Guha, A. (2004). Artificial Neural Networks: Applications to Textiles. *Textile Progress*, Vol.35, No.1, pp. 1-46, ISSN 0040-5175

Chen, X. & Huang, X. B. (2004). Evaluating Fabric Pilling with Light-Projected Image Analysis. *Textile Research Journal*, Vol.74, No.11, pp. 977-981, ISSN 0040-5175

Chen, T.; Zhang, C.; Chen, X. & Li, L. (2009). An Input Variable Selection Method for the Artificial Neural Network of Shear Stiffness of Worsted Fabrics. *Statistical Analysis and Data Mining*, Vol.1, pp. 287-295, ISSN 1932-1864

Cheng, K. P. S. & Lam, H. L. I. (2003). Evaluating and Comparing the Physical Properties of Sliced Yarns by Regression and Neural Network Techniques. *Textile Research Journal*, Vol.73, No.2, pp. 161-164, ISSN 0040-5175

Dayik, M. (2009). Predicting of Yarn Properties using Evaluation Programing. *Textile Research Journal*, Vol.79, No.11, pp. 963-972, ISSN 0040-5175

Demiryurek, O. & Koc, E. (2009). Predicting the Unevenness of Polyester/Viscose lended Open-end Rotor Spun Yarns Using Artificial Neural Network and Statistical Models. *Fibers and Polymers*, Vol.10, No.2, pp. 237-245, ISSN 1229-9197

Dev, V. R. G.; Venugopal, J. R.; Senthilkumar, M.; Gupta, D. & Ramakrishna, D. (2009).Prediction of Water Retention Capacity of Hydrolysed Electrospun Polyacrylonitrile Fibers using Statistical Model and Artificial Neural Network. *Journal of Applied Polymer Science*, Vol.113, pp. 3397-3404, ISSN 0021-8995

Gharehaghaji, A. A.; Shanbeh, M. & Palhang, M. (2007). Analysis of Two Modeling Methodologies for Predicting the Tensile Properties of Cotton-Covered Nylon Core Yarns. *Textile Research Journal*, Vol.77, No.8, pp. 565-571, ISSN 0040-5175

Golob, D.; Osterman, D. P. & Zupan, J. (2008). Determination of Pigment combinations for Textile Printing Using Artificial Neural Networks. *Fibers & Textiles in Eastern Europe*, Vol.16, No.3, pp. 93-98, ISSN 1230-3666

Guruprasad, R. & Behera, B. K. (2010). Soft Computing in Textiles. *Indian Journal of Fibre & Textile Research*, Vol.35, pp. 75-84, ISSN 0971-0426

Hadizadeh, M.; Jeddi, A. A. A. & Amani Tehran, M. (2009). The Prediction of Initial Load-Extension Behavior of Woven Fabrics using Artificial Neural Network. *Textile Research Journal*, Vol.79, No.17, pp. 1599-1609, ISSN 0040-5175

Huang, C. C. & Chang, K. T. (2001). Fuzzy Self-Organizing and Neural Network Control of Sliver Linear Density in a Drawing Frame. *Textile Research Journal*, Vol.71, No.11, pp. 987-992, ISSN 0040-5175

Hui, C. L.; Lau, T. W.; Ng, S. F. & Chan, K. C. C. (2004). Neural Network Prediction of Human Psychological Perceptions of Fabric Hand. *Textile Research Journal*, Vol.74, No.5, pp. 375-383, ISSN 0040-5175

Jeon, B. S.; Bae, J. H. & Suh, M. W. (2003). Automatic Recognition of Woven Fabric Patterns by an Artificial Neural Network. *Textile Research Journal*, Vol.73, No.7, pp. 645-650, ISSN 0040-5175

Kang, T. J. & Kim, S. C. (2002). Objective Evaluation of the Trash and Color of Raw Cotton by Image Processing and Neural Network. *Textile Research Journal*, Vol.79, No.9, pp. 776-782, ISSN 0040-5175

Karras, D. A.; Karkanis, S. A. & Mertzios, B. G. (1998). Supervised and Unsupervised Neural Network Methods applied to Textile Quality Control based on Improved Wavelet Feature Extraction Techniques. *International Journal of Computer Mathematics*, Vol.67, No.1&2, pp. 169-181, ISSN 0020-7160

Keshavaraj, R.; Tock, R.W. & Haycook, D. (1996).Airbag Fabric Material Modeling of Nylon and Polyester Fabrics Using a Very Simple Neural Network Architecture. *Journal of Applied Polymer Science*, Vol.60, pp. 2329-2338, ISSN 0021-8995

Khan, Z.; Lim, A. E. K.; Wang, L.; Wang, X. & Beltran, R. (2002). An Artificial Neural Network-based Hairiness Prediction Model for Worsted Wool Yarns. *Textile Research Journal*, Vol.79, No.8, pp. 714-720, ISSN 0040-5175

Kumar, A. (2003). Neural Network Based Detection of Local Textile Defects. *Pattern Recognition*, Vol.36, pp. 1645-1659, ISSN 0031-3203

Kuo, C. F. J.; Hsiao, K. I. & Wu, Y. S. (2004). Using Neural Network Theory to Predict the Properties of Melt Spun Fibers. *Textile Research Journal*, Vol.74, No.9, pp. 840-843, ISSN 0040-5175

Kuo, C. F. J. & Lee, C. J. (2003). A Back-Propagation Neural Network for Recognizing Fabric Defects. *Textile Research Journal*, Vol.73, No.2, pp. 147-151, ISSN 0040-5175

Kuo, C. F. J.; Lee, C. J. & Tsai, C. C. (2003). Using a Neural Network to Identify Fabric Defects in Dynamic Cloth Inspection. *Textile Research Journal*, Vol.73, No.3, pp. 238 244, ISSN 0040-5175

Kuo, C. F. J.; Su, T. L. & Huang, Y. J. (2007). Computerized Color Separation System for Printed Fabrics by using Backward-Propagation Neural Network. *Fibers and Polymers*, Vol.8, No.5, pp. 529-536, ISSN 1229-9197

Leonard, J.; Pirotte, F. & Knott, J. (1998). Classification of Second Hand Textile Waste Based on Near-infrared Analysis and Neural Network. *Melliand International*, Vol.4, pp. 242-244, ISSN 0947-9163

Lien, H. C. & Lee, S. (2002). A Method of Feature Selection for Textile Yarn Grading Using the Effective Distance Bettween Clusters. *Textile Research Journal*, Vol.72, No.10, pp. 870-878, ISSN 0040-5175

Lin, J. J. (2007). Prediction of Yarn Shrinkage Using Neural Nets. *Textile Research Journal*, Vol.77, No.5, pp. 336-342, ISSN 0040-5175

Liu, Y.; Liu, W. & Zhang, Y. (2001). Inspection of Defects in Optical Fibers based on Back-Propagation Neural Networks. *Optics Communications*, Vol.198, pp. 369-378, ISSN 0030-4018

Liu, J.; Zuo, B.; Zeng, X.; Vroman, P. & Rabenasolo, B. (2010). Nonwovens Uniformity Identification using Wavelet Texture Analysis and LVQ Neural Network. *Expert Systems with Applications*, Vol.37, pp.2241-2246, ISSN 0957-4174

Liu, J.; Zuo, B.; Vroman, P.; Rabenasolo, B.; Zeng, X. & Bai, L. (2010). Visual Quality Recognition of Nonwovens using Wavelet Texture Analysis and Robust Bayesian Neural Network. *Textile Research Journal*, Vol.80, No.13, pp.1278-1289, ISSN 0040- 5175

Majumdar, P. K. & Majumdar, A. (2002). Predicting the Breaking Elongation of Ring Spun Cotton Yarns using Mathematical, Statistical, and Artificial Neural Network Models. *Textile Research Journal*, Vol.74, No.7, pp. 652-655, ISSN 0040-5175

Mokhtari Yazdi, M.; Semnani, D. & Sheikhzadeh, M. (2009). Moisture and Heat Transfer in Hybrid Weft Knitted Fabric with Artificial Intelligence. *Journal of Applied Polymer Science*, Vol.114, pp. 1731-1737, ISSN 0021-8995

Moradian, S. & Amani Tehran, M. (2000). Predicting the Degree of Metamerism by Artificial Neural Network, *Proceedings of Fifth Seminar on Polymer Science and Technology*, pp. 153-156, Amirkabir University of Technology, Tehran, Iran, September 12-14, 2000

Mori, T. & Komiyama, J. (2002). Evaluating Wrinkled Fabrics with Image Analysis and Neural Networks. *Textile Research Journal*, Vol.72, No.5, pp. 417-422, ISSN 0040-5175

Murrells, C. M.; Tao, X. M.; Xu, B. G. & Cheng, K. P. S. (2009). An Artificial Neural Network Model for the Prediction of Spirality of Fully Relaxed Single Jersey Fabrics. *Textile Research Journal*, Vol.79, No.3, pp. 227-234, ISSN 0040 5175

Nirmal, U. (2010). Prediction of Friction coefficient of Treated Betelnut Fiber Reinforced polyester (T-BFRP) Composite using Artificial Neural Networks. *Tribology International*, Vol.43, pp. 1417-1429, ISSN 0301-679X

Onal, L.; Zeydan, M.; Korkmaz, M. & Meeran, S. (2009). Predicting the Seam Strength of Notched Webbings for Parachute Assemblies using the Taguchi's Design of Experiment and Artificial Neural Networks. *Textile Research Journal*, Vol.79, No.5, pp. 468-478, ISSN 0040-5175

Pynckels, F.; Kiekens, P.; Sette, S.; Van Langenhove, L. & Impe, K. (1995). Use of NeuralNets for Determining the Spinnability of Fibers. *Journal of Textile Institute*, Vol.86, No.3, pp. 425-437, ISSN 0040-5000

Rawal, A.; Majumdar, A.; Anand, S. & Shah, T. (2009). Predicting the Properties of Needlepunched Nonwovens using Artificial Neural Network. *Journal of Applied Polymer Science*, Vol.112, pp. 3575-3581, ISSN 0021-8995

Semnani, D. & Vadood, M. (2010). Improvement of Intelligent Methods for Evaluating the Apparent Quality of Knitted Fabrics. *Engineering Applications of Artificial Intelligence*, Vol.23, pp. 217-221, ISSN 0952-1976

Shady, E.; Gowayed, Y.; Abouiiana, M.; Youssef, S. & Pastore, C. (2006). Detection and Classification of Defects in Knitted Fabric Structures. *Textile Research Journal*, Vol.76, No.4, pp. 295-300, ISSN 0040-5175

She, F. H.; Kong, L. X.; Nahavandi, S. & Kouzani, A. Z. (2002). Intelligent Animal Fiber Classification with Artificial Neural Network. *Textile Research Journal*, Vol.72, No.7, pp. 594-600, ISSN 0040-5175

Sun, Q.; Zhang, D.; Chen, B. & Wadsworth, L. C. (1996). Application of Neural Networks to Meltblown Process Control. *Journal of Applied Polymer Science*, Vol.62, pp. 1605-1611, ISSN 0021-8995

Tilocca, A. (2002). Detecting Fabric Defects with a Neural Network using Two Kinds of Optical Patterns. *Textile Research Journal*, Vol.72, No.6, pp. 545-550, ISSN 0040-5175

Thevenet, L.; Dupont, D. & Jolly-Desodt, A. M. (2002). Modeling Color change after Spinning Process using Feedforward Neural Networks. *Color Research and Application*, Vol.28, No.1, pp. 50-58, ISSN 1520-6378

Tokarska, M. (2004). Neural Model of the Permeability Features of Woven Fabrics. *Textile Research Journal*, Vol.74, No.12, pp. 1045-1048, ISSN 0040-5175

Unal, P. G.; Ozdil, N. & Taskin, C. (2002). The Effect of Fiber Properties on the Characteristics of Spliced Yarns Part 1: Prediction of Spliced Yarns Tensile Properties. *Textile Research Journal*, Vol.80, No.5, pp. 429-438, ISSN 0040-5175

Xu, B.; Fang, C. & Watson, M. D. (1999). Clustering Analysis for Cotton Trash Classification. *Textile Research Journal*, Vol.69, No.9, pp. 656-662, ISSN 0040-5175

Yin, X. & Yu, W. (2007). The Virtual Manufacturing Model of the Worsted Yarn based on Artificial Neural Networks and Grey Theory. *Applied Mathematics and Computation*, Vol.185, pp. 322-332, ISSN 0096-3003

Yao, G.; Guo, J. & Zhou, Y. (2005). Predicting the Warp Breakage Rate in Weaving by Neural Network Techniques. *Textile Research Journal*, Vol.75, No.3, pp. 274-278, ISSN 0040-5175

Yeung, K. W.; Li, Y.; Zhang, X. & Yao, M. (2002). Evaluating and Predicting Fabric Bagging with Image Processing. *Textile Research Journal*, Vol.72, No.8, pp. 693-700, ISSN 0040-5175

Yuen, C. W. M.; Wong, W. K.; Qian, S. Q.; Chan, L. K. & Fung, E. H. K. (2009). A Hybrid Model using Genetic Algorithm and Neural Network for Classifying Garment Defects. *Expert Systems with Applications*, Vol.36, pp. 2037-2047, ISSN 0957-4174

Zhang, Y.; Zhaoyang, L. & Li, J. (2010). Fabric Defect Classification using Radial Basis Function Network. *Pattern Recognition Letters*, Vol.31, pp. 2033-2042, ISSN 0167- 8655

3

Modelling of Needle-Punched Nonwoven Fabric Properties Using Artificial Neural Network

Dr. Sanjoy Debnath
National Institute of Research on Jute & Allied Fibre Technology
Indian Council of Agricultural Research
12, Regent Park, Kolkata – 700 040, West Bengal
India

1. Introduction

Needle-punched nonwoven is an industrial fabric used in wide range of applications areas. The physical structure of needle-punched nonwoven is very complex in nature and therefore engineering the fabric according the required properties is difficult. Because of this, the basic mathematical modeling is not very successful for predicting various important properties of the fabrics.

In recent days, artificial neural networks (ANN) have shown a great assurance for modeling non-linear processes. Rajamanickam et al., 1997 and Ramesh et al., 1995 used ANN to model the tensile properties of air jet yarn. The ANN model had also been used to model to assess the set marks and also the relaxation curve of yarn after dynamic loading (Vangheluwe et al., 1993 and 1996). Luo & David, 1995 used the HVI experimental test results to train the neural nets and predict the yarn strength. Researchers also made an attempt to build models for predicting ring or rotor yarn hairiness using a back propagation ANN model by Zhu & Ethridge, 1997. Fan & Hunter, 1998 developed ANN for predicting the fabric properties based on fibre, yarn and fabric constructional parameters and suggested the suitable computer programming for development of neural network model using back-propagation simulator. Wen et al., 1998 used back-propagation neural network model for classification of textile faults. Postle, 1997 enlighten on measurement and fabric categorisation and quality evaluation by neural networks. Park et al., 2000 also enlightened the use of fuzzy logic and neural network method for hand evaluation of outerwear knitted fabrics. Gong & Chen, 1999 found that the use of neural network is very effective for predicting problems in clothing manufacturing. Xu et al., 1999 used three clustering analysis technique viz. sum of squares, fuzzy and neural network for cotton trash classification. They found neural network clustering yields the highest accuracy, but it needs more computational time for network training. Vangheluwe et al., 1993 found Neural nets showed good results assessing the visibility set marks in fabrics. The review of literature shows that the ANN model is a powerful and accurate tool for predicting a nonlinear relationship between input and output variables.

Jute, polypropylene, jute-polypropylene blended and polyester needle punched nonwoven fabrics have been prepared using series of textile machinery normally used in needle-punching process for preparation of the fabric samples. Textile materials are compressive in

nature. It has been reported by various authors that the effect of compression behaviour of jute-polypropylene (Debnath & Madhusoothanan, 2007) and polyester (Midha et al., 2004) is largely influenced by fibre linear density, blend ratios of fibres, fabric weight, web laying type, needling density and depth of needle penetration. Kothari & Das, 1992 and 1993 explained that the compression behaviour of needle-punched nonwoven fabrics is dependent on fibre fineness, proportion of finer fibre present in different layers of nonwoven fabrics, and fabric weight for polyester and polypropylene fibres. In the present study, some of these important factors, viz. fabric weight, blend proportion, three different types of fibres and needling density, have been taken into consideration for modeling of the compression behaviour. Jute, polypropylene and polyester fibres have been used in this study. Woollenisation of jute has been done to develop crimp in the fibre. This study also elaborates the effect of number of hidden layers and simulation cycles for jute-polypropylene blended and polyester needle-punched nonwoven fabrics. Different fabric properties like fabric weight, needling density, blend composition of the fibres are the basic variables selected as input variables. The output variables are selected as air permeability, tensile, and compression properties.

Under tensile properties, tenacity and initial modulus of jute-polypropylene blended needle punched nonwoven fabric both in machine (lengthwise) and transverse (width wise) directions have been predicted accurately using artificial neural network. Empirical models have also been developed for the tensile properties and found that artificial neural network models are more accurate than empirical models. Prediction of tensile properties by ANN model shows considerably lower error than empirical model when the inputs are beyond the range of inputs, which were used for developing the model. Thus the prediction by artificial neural network model shows better results than that by empirical model even for the extrapolated input variables.

The jute-polypropylene blended needle-punched nonwoven fabric samples were produced as per a statistical factorial design for prediction of air permeability. The efficiency of prediction of two models has been experimentally verified wherein some of the input variables were beyond the range over which the models were developed. The predicted air permeability values from both the models have been compared statistically. An attempt has also been made to study the effect of number of hidden layer in neural network model. The highest correlation has been found in artificial neural network with three hidden layers. The neural network model with three hidden layer shows less prediction error followed by two hidden layers, empirical model and artificial neural network with one hidden layer. Artificial neural network model with three hidden layers predicts the value of air permeability with minimum error when inputs are beyond the range of inputs used for developing the model.

Initial thickness, percentage compression, thickness loss and percentage compression resilience are the compression properties predicted using artificial neural network model of needle-punched nonwoven fabrics produced from polyester and jute-polypropylene blended fibres varying fabric weight, needling density, blend ratio of jute and polypropylene, and polyester fibre. A very good correlation (R^2 values) with minimum error between the experimental and the predicted values of compression properties have been obtained by artificial neural network model with two and three hidden layers. An attempt has also been made for experimental verification of the predicted values for the input variables not used during the training phase. The prediction of compression properties by artificial neural network model in some particular sample is less accurate due to lack of learning during

training phase. The three hidden layered artificial neural network models take more time for computation during training phase but the predicted results are more accurate with less variations in the absolute error in the verification phase. This study will be useful to the industry for designing the needle-punched nonwoven fabric made out of jute-polypropylene blended or polyester fibres for desired fabric properties. The cost for design and development of desired needle-punched fabric property of the said nonwovens can also be minimised.

2. Materials and methods

2.1 Materials

Polypropylene fibre of 0.44 tex fineness, 80 mm length; jute fibres of Tossa-4 grade and polyester fibre of 51 mm length and 0.33 tex fineness fibre of were used to prepare the fabric samples. Some important properties of fibres are presented in Table 1. Sodium hydroxide and acetic acid were used for woollenisation of the jute.

Property	Jute	Polypropylene	Polyester
Fibre fineness (tex)	2.08	0.44	0.33
Density (g/cm^3)	1.45	0.91	1.38
Tensile strength (cN/tex)	30.1	34.5	34.83
Breaking elongation (%)	1.55	54.13	51.00
Moisture regain (%) at 65% RH	12.5	0.05	0.40

Table 1. Properties of jute, polypropylene and polyester fibres

2.2 Methods

2.2.1 Preparation of jute, jute-polypropylene blended and polyester fabrics

The raw jute fibres do not produce good quality fabric because there is no crimp in these fibres. To develop crimp before the fabric production, the jute fibres were treated with 18% (w/v) sodium hydroxide solution at 30°C using the liquor-to-material ratio of 10:1, as suggested by Sao & Jain, 1995. After 45 min of soaking, the jute fibres were taken out, washed thoroughly in running water and treated with 1% acetic acid. The treated fibres were washed again and then dried in air for 24 h. This process apart from introducing about 2 crimps/cm also results in weight loss of ~ 9.5%.

The jute reeds were opened in a roller and clearer card, which produces almost mesh-free stapled fibre. The woollenised jute and polypropylene fibres were opened by hand separately and blended in different blend proportions (Table 2). The blended materials were thoroughly opened by passing through one carding passage.

The blended fibres were fed to the lattice of the roller and clearer card at a uniform and predetermined rate so that a web of 50 g/m^2 can be achieved. The fibrous web coming out from the card was fed to feed lattice of cross-lapper and cross-laid webs were produced with cross-lapping angle of 20°. The web was then fed to the needling zone. The required needling density was obtained by adjusting the throughput speed.

Different web combinations, as per fabric weight (g/m^2) requirements were passed through the needling zone of the machine for a number of times depending upon the punch density required. A punch density of 50 punches/cm^2 was given on each passage of the web, changing the web face alternatively. The fabric samples were produced as per the variables presented in Table 2.

Fabric code	Fabric weight g/m²	Needling density punches/cm²	Woollenised jute %	Polypropylene fibre %	Polyester fibre %
1	250	150	40	60	-
2	250	350	40	60	-
3	450	150	40	60	-
4	450	350	40	60	-
5	250	250	60	40	-
6	250	250	20	80	-
7	450	250	60	40	-
8	450	250	20	80	-
9	350	150	60	40	-
10	350	150	20	80	-
11	350	350	60	40	-
12	350	350	20	80	-
13	350	250	40	60	-
14	350	250	40	60	-
15	350	250	40	60	-
16	393	150	0	100	-
17	440	150	0	100	-
18	410	250	0	100	-
19	392	350	0	100	-
20	241	150	100	0	-
21	310	250	100	0	-
22	303	350	100	0	-
23	300	150	80	20	-
24	276	250	80	20	-
25	205	350	80	20	-
26	415	300	-	-	100
27	515	300	-	-	100
28	680	300	-	-	100
29	815	300	-	-	100

Table 2. Experimental design of fabric samples

The polyester fabric samples were made from parallel-laid webs, which were obtained by feeding opened fibres in the TAIRO laboratory model with stationary flat card (2009a). The fine web emerging out from the card was built up into several layers in order to obtain desired level of fabric weight (Table 2). The needle punching of all parallel-laid polyester fabric samples was carried out in James Hunter Laboratory Fiber Locker [Model 26 (315 mm)] having a stroke frequency of 170 strokes/min. The machine speed and needling density were selected in such a way that in a single passage 50 punches/cm² of needling density could be obtained on the fabric. The web was passed through the machine for a number of times depending upon the needling density required, e.g. the web was passed 6 times through the machine to obtain fabric with 300 punches/cm². The needling was done alternatively on each side of the polyester fabric.

The needle dimension of 15 × 18 × 36 × R/SP 3½ × ¼ × 9 was used for all jute-polypropylene, jute and polyester samples. The depth of needle penetration was also kept constant at 11 mm in all the cases.
The actual fabric weights of the final needle-punched fabric samples were measured considering the average weight of randomly cut 1 m^2 sample at 5 different places from each sample.

2.2.2 Measurement of tenacity and initial modulus

The mechanical properties like tenacity and initial modulus were measured both in the machine and transverse directions (Debnath et al., 2000a) of the fabric using an Instron tensile tester (Model 4301). The size of sample and the rate of straining were chosen according to ATSM standard D1117-80 (sample size 7.6 cm x 2.5 cm, cross head transverse speed 300 mm/min). Breaking load verses elongation curves were plotted for all the tests. The tenacity was calculated by normalising the breaking load by fabric weight and width of the specimen as suggested by Hearle & Sultan, 1967. The initial modulus was calculated from the load elongation curves.

2.2.3 Measurement of air permeability

The air permeability measurements were done using the Shirley (SDL-21) air permeability tester (Debnath & Madhusoothanan, 2010b). The test area was 5.07 cm^2. The pressure range = 0.25 mm and flow range = 0.04 – 350 cc/sec. The airflow in cubic cm at 10 mm water head pressure was measured. The air permeability of fabric samples was calculated using the formula (1) given below (Sengupta et al., 1985 and Debnath et al., 2006).

$$AP = \frac{AF}{TA} \times 10^{-2} \tag{1}$$

Where, AP = air permeability of fabric in $m^3/m^2/sec$, AF = air flow through fabric in cm^3/sec at 10 mm water head pressure and TA = test specimen area in cm^2 for each sample.

2.2.4 Measurement of compression properties

The initial thickness (Debnath & Madhusoothanan, 2010a), compression, thickness loss and compression resilience were calculated from the compression and decompression curves. For measuring these properties, a thickness tester was used (Subramaniam et al., 1990). The pressure foot area was 5.067 cm^2 (diameter = ϕ2.54 cm). The dial gauge with a least count of 0.01 mm and maximum displacement of 10.5 mm was attached to the thickness tester. The compression properties were studied under a pressure range between 1.55 kPa and 51.89 kPa.
The initial thickness of the needle-punched fabrics was observed under the pressure of 1.55 kPa (Debnath & Madhusoothanan, 2007). The corresponding thickness values were observed from the dial gauge for each corresponding load of 1.962 N. A delay of 30 s was given between the previous and next load applied. Similarly, 30 s delay was also allowed during decompression cycle at every individual load of 1.962 N. This compression and recovery thickness values for corresponding pressure values are used to plot the compression-recovery curves.
The percentage compression (Debnath & Madhusoothanan, 2007), percentage thickness loss (Debnath & Madhusoothanan, 2009a and Debnath & Roy, 1999) and percentage

compression resilience (Debnath & Madhusoothanan, 2007, 2009a and 2009b), were estimated using the following relationships (2,3,4):

$$\text{Compression (\%)} = \frac{T_0 - T_1}{T_0} \times 100 \tag{2}$$

$$\text{Thickness loss (\%)} = \frac{T_0 - T_2}{T_0} \times 100 \tag{3}$$

$$\text{Compression resilience (\%)} = \frac{W_c'}{W_c} \times 100 \tag{4}$$

where T_0 is the initial thickness; T_1, the thickness at maximum pressure; T_2, the recovered thickness; W_c, the work done during compression; and W_c', the work done during recovery process.

The average of ten readings from different places for each sample was considered. The coefficient of variation was less than 6% in all the cases.

All these tests were carried out in the standard atmospheric condition of 65 ± 2% RH and 20 ± 2°C. The fabrics were conditioned for 24 h in the above mentioned atmospheric conditions before testing.

2.2.5 Empirical model

An empirical equation of second order polynomial (Box & Behnken, 1960) was derived to predict the mechanical properties (Debnath et al. 2000a) like tenacity and initial modulus, and physical property like air permeability (Debnath et al. 2000a) were predicted from the results obtained from the samples produced using Box and Behnken factorial design.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_{11} {X_1}^2 + \beta_{22} {X_2}^2 + \beta_{33} {X_3}^2 + \beta_{12} X_1 X_2 + \beta_{13} X_1 X_3 + \beta_{23} X_2 X_3 \tag{5}$$

Where, Y = predicted fabric property (tenacity or initial modulus or air permeability), X_1 = fabric weight, X_2 = needling density, X_3 = percentage of polypropylene, β_0 is the constant and β_i is the coefficient of the variable X_i. The predicted values of fabric properties were then compared with the actual values and error (6) was calculated.

$$E\ (\%) = \frac{A - P}{A} \times 100 \tag{6}$$

Where, E is error in percentage, A is the actual experimental values and P is the predicted values from models.

2.2.6 Artificial neural network model

The physiology of neurons present in biological neural system such as human nervous system was the fundamental idea behind developing the ANNs. This computational model was trained to capture nonlinear relationship between input and output variables with scientific and mathematical basis. In recent days, commonly used model is layered feed-forward neural network with multi layer perceptions and back propagation learning algorithms (Vangheluwe et al., 1993, Rajamanickam et al., 1997, Zhu & Ethridge, 1997 and Wen et al., 1998).

The ANNs are computing systems composed of a number of highly interconnected layers of simple neuron like processing elements, which process information by their dynamic response to external inputs. The information passed through the complete network by linear connection with linear or nonlinear transformations. The weights were determined by training the neural nets. Once the ANN was trained, it was used for predicting new sets of inputs. Multi layer feed-forward neural network architecture (Figure 1) was used for predicting the tenacity, initial modulus, air permeability, initial thickness, percentage compression, thickness loss and compression resilience properties of fabrics (Debnath et al., 2000a, 2000b and Debnath & Madhusoothanan, 2008). The circle in Figure 3.5 represents the neurons arranged in five layers as one input, one output and three hidden layers. Three neurons in the input layer, three hidden layers, each layer consisting of three neurons and one neuron in the output layer. HL-1, HL-2 and HL-3 are 1st, 2nd and 3rd hidden layers respectively, whereas i and j are two different neurons in two different layers. The neuron (i) in one layer was connected with the neuron (j) in next layer with weights (W_{ij}) as presented in the Figure 1.

The data were scaled down between 0 and 1 by normalizing them with their respective values. The ANN was trained with known sets of input-output data pairs.

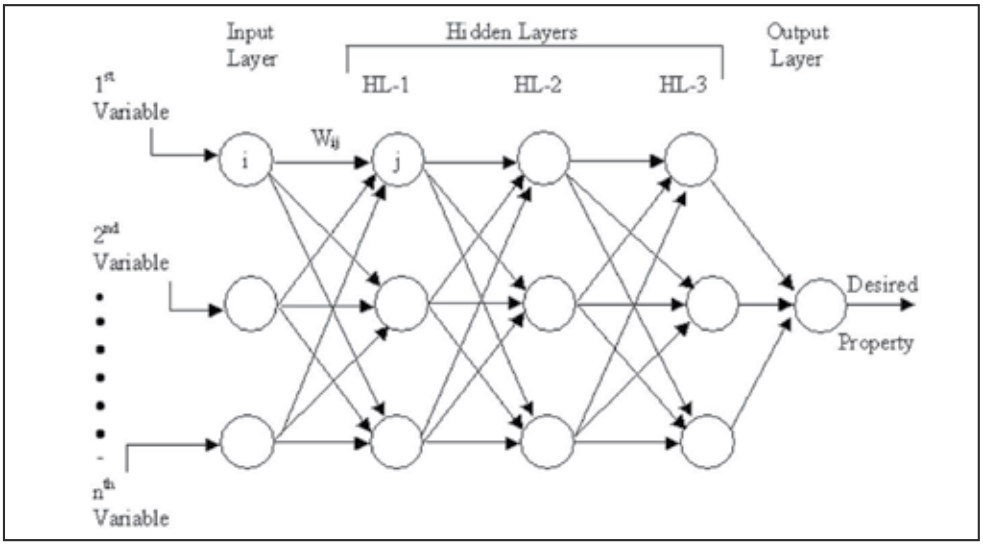


Fig. 1. Neural architecture of the fabric property

3. Results and discussion

3.1 Modelling of tenacity and initial modulus

The empirical and ANN models for tensile properties have been developed from the experimental values (Debnath et al., 2000a) of fifteen sets of selected fabric samples as shown in Table 3.

The constants and coefficients of the empirical model for the fifteen fabric sample sets (Table 3) were calculated with the help of multiple regression analysis, are given in Table 4.

The ANN was trained up to 64,000 cycles to obtain optimum weights for the same sample sets used to develop emperical model (Table 3). The weights of ANN for tenacity and initial

modulus on both machine and transverse direction were presented in Table 5. Tables 6 and 7 show the experimental, predicted values and their prediction error for tenacity and initial modulus respectively.

The Table 6 shows a very good correlation (R^2 values) between the experimental and predicted tenacity values by ANN than by empirical model in both the machine and transverse directions of the fabrics. Similar trend was also observed in the case of initial modulus (Table 7).

The ANN models of tenacity and initial modulus show much lower absolute percentage error and mean absolute percentage error than that of empirical model (Tables 6 and 7). The standard deviation of mean absolute percentage error also follows the similar trend. This

Fabric code	Fabric weight g/m²	Needling density punches/cm²	Woollenised jute %	Polypropylene fibre %
1	250	150	40	60
2	250	350	40	60
3	450	150	40	60
4	450	350	40	60
5	250	250	60	40
6	250	250	20	80
7	450	250	60	40
8	450	250	20	80
9	350	150	60	40
10	350	150	20	80
11	350	350	60	40
12	350	350	20	80
13	350	250	40	60
14	350	250	40	60
15	350	250	40	60

Table 3. Fabric samples for development of Emperical and ANN models

	Tenacity		Initial Modulus	
	Machine direction	Transverse direction	Machine direction	Transverse direction
β_0	-9.882	-9.157	-7.448E-01	-2.832E-01
β_1	1.484E-02	1.228E-02	1.925E-03	2.806E-03
β_2	3.129E-02	2.610E-02	6.544E-03	5.279E-03
β_3	1.362E-01	1.833E-01	-4.700E-03	-2.063E-02
β_{11}	-6.084E-06	-1.817E-06	-3.908E-06	-7.840E-06
β_{22}	-2.838E-05	-2.682E-05	-1.388E-05	-1.941E-05
β_{33}	-5.033E-04	-3.787E-04	-3.216E-05	6.992E-05
β_{12}	-3.068E-05	-2.155E-05	1.835E-06	1.147E-05
β_{13}	-5.0170E-05	-1.157E-04	1.817E-05	2.775E-05
β_{23}	-1.251E-04	-1.849E-04	2.242E-05	2.596E-05

Table 4. Coefficients and constants of empirical models of tenacity and initial modulus

Weights between the layers number		Tenacity		Initial modulus	
		Machine direction	Transverse direction	Machine direction	Transverse direction
1st and 2nd	W_{11}	-4.053	1.185	0.379	-6.844
	W_{12}	1.363	-2.341	11.313	1.539
	W_{13}	2.035	5.420	2.564	-2.829
	W_{21}	-4.530	-0.496	0.919	16.684
	W_{22}	3.401	-0.667	-16.856	4.141
	W_{23}	7.707	5.064	-9.534	-0.370
	W_{31}	5.997	3.669	-4.380	-1.518
	W_{32}	-6.298	0.890	2.876	-7.049
	W_{33}	-7.736	-9.883	4.257	1.298
2nd and 3rd	W_{11}	1.207	3.113	-2.472	-0.752
	W_{12}	1.689	-6.265	10.783	3.987
	W_{13}	-3.273	0.630	-3.429	-2.242
	W_{21}	-17.135	-8.309	1.478	2.702
	W_{22}	5.736	3.556	-2.926	-0.151
	W_{23}	10.765	2.652	0.811	6.455
	W_{31}	3.907	-12.208	-5.815	-8.148
	W_{32}	-6.176	5.439	3.362	-3.522
	W_{33}	4.880	-5.658	0.882	9.483
3rd and 4th	W_{11}	-12.307	3.779	1.784	-1.669
	W_{12}	3.732	-5.345	6.455	4.879
	W_{13}	-11.562	6.306	-5.127	-4.866
	W_{21}	10.984	-2.423	-0.415	2.262
	W_{22}	0.739	1.605	-9.454	2.647
	W_{23}	6.466	-1.513	0.686	-2.908
	W_{31}	2.598	-2.440	-0.643	-0.846
	W_{32}	-13.977	3.412	4.862	-7.376
	W_{33}	-1.486	-4.109	0.810	7.533
4th and 5th	W_{10}	1.979	4.550	2.702	5.054
	W_{20}	12.652	-7.022	11.945	8.722
	W_{30}	-9.348	7.491	-3.734	-4.757

Table 5. Weights of ANN model for tenacity and initial modulus

Fabric code	Tenacity in the machine direction					Tenacity in the transverse direction				
	Exp tenacity (cN/Tex)	Predicted tenacity (cN/Tex)		Absolute error (%)		Exp tenacity (cN/Tex)	Predicted tenacity (cN/Tex)		Absolute error (%)	
		Emp	ANN	Emp	ANN		Emp	ANN	Emp	ANN
1	0.513	0.827	0.514	61.65	00.04	2.220	2.540	2.222	14.43	00.09
2	1.357	1.214	1.355	10.57	00.20	2.000	1.775	1.961	11.23	01.97
3	1.279	1.423	1.277	11.22	00.20	2.484	2.708	2.462	09.05	00.89
4	0.896	0.579	0.901	35.32	00.55	1.402	1.081	1.402	22.86	00.04
5	0.544	0.466	0.545	14.39	00.22	0.827	1.020	0.845	23.36	02.13
6	1.837	1.743	1.838	05.15	00.01	3.819	3.530	3.818	07.56	00.02
7	0.551	0.646	0.544	17.17	01.23	0.931	1.220	0.922	31.02	00.95
8	1.443	1.521	1.444	05.43	00.07	2.998	2.805	2.994	06.44	00.33
9	0.435	0.197	0.433	54.71	00.51	1.611	1.098	1.603	31.88	00.50
10	1.996	1.774	1.996	11.12	00.01	3.916	3.885	3.914	00.81	00.07
11	0.247	0.468	0.248	90.00	00.69	0.610	0.641	0.601	05.18	01.35
12	0.806	1.044	1.001	29.55	24.22	1.435	1.949	1.425	35.79	00.71
13	1.345	1.356	1.348	00.84	00.22	2.296	2.313	2.315	00.75	00.80
14	1.391	1.356	1.348	02.51	03.11	2.609	2.313	2.315	11.33	11.28
15	1.332	1.356	1.348	01.78	01.15	2.035	2.313	2.315	13.68	13.75
'R^2' values		0.879	0.990				0.911	0.994		
Mean absolute percentage error				23.43	02.16				15.03	02.33
SD of absolute percentage error				26.34	06.15				11.34	04.21
Exp – Experimental; Emp – Empirical model and ANN – Artificial Neural Network Model										

Table 6. Experimental and predicted tenacity values by empirical and ANN models

indicates that the prediction by ANN model is closer to the experimental values and variations of error among the samples were also lower than the prediction by empirical model. This could be due to the fact that the prediction by empirical model is not very accurate when the relationship between the inputs and outputs is nonlinear (Debnath et al. 2000a).

3.1.1 Verification of tenacity and initial modulus models

An attempt was made to predict the tenacity and initial modulus in machine direction and in transverse direction to understand the accuracy of the models. The ANNs and empirical models were then presented to three sets of inputs, which have not appeared during the modeling phase as shown in Table 8. The input variables were selected in such a way that one input variable is beyond the range with which the ANN was trained or empirical model was developed. The Table 8 indicates that the prediction errors of ANNs were lower in both the directions of the fabric for tenacity and initial modulus in comparison with that of empirical model (Debnath et al., 2000a).

In Table 8 the predicted tenacity and initial modulus values by ANN gives higher absolute percentage error than the predicted values in Tables 6 and 7. This may be due to the fact that the selected input variables (Table 8) were beyond the range over which the empirical or ANN models were developed (Debnath et al., 2000a). However, in most of the cases of prediction ANNs give lesser absolute percentage error than the empirical model.

Fabric code	Initial modulus in the machine direction					Initial Modulus in the transverse direction				
	Exp (cN/Tex)	Predicted initial modulus (cN/Tex)		Absolute error (%)		Exp (cN/Tex)	Predicted initial modulus (cN/Tex)		Absolute error (%)	
		Emp	ANN	Emp	ANN		Emp	ANN	Emp	ANN
1	0.396	0.307	0.394	22.44	00.38	0.550	0.377	0.556	31.42	01.11
2	0.736	0.589	0.736	19.96	00.08	0.451	0.377	0.433	16.46	04.12
3	0.271	0.418	0.270	54.19	00.30	0.444	0.518	0.445	16.75	00.36
4	0.685	0.773	0.685	12.97	00.00	0.804	0.976	0.805	21.51	00.19
5	0.494	0.542	0.495	09.76	00.12	0.400	0.578	0.422	43.77	05.40
6	0.418	0.606	0.420	44.85	00.36	0.551	0.623	0.552	13.05	00.20
7	0.805	0.617	0.804	23.30	00.06	0.906	0.834	0.908	07.93	00.18
8	0.874	0.826	0.874	05.51	00.02	1.279	1.104	1.278	13.70	00.06
9	0.325	0.365	0.326	12.50	00.34	0.529	0.527	0.520	00.45	01.74
10	0.511	0.412	0.511	19.33	00.02	0.480	0.581	0.479	21.01	00.27
11	0.496	0.594	0.496	19.89	00.00	0.753	0.652	0.752	13.40	00.12
12	0.861	0.820	0.860	04.72	00.09	0.912	0.914	0.908	00.25	00.43
13	0.644	0.700	0.718	02.34	04.94	0.836	0.835	0.847	00.13	01.40
14	0.688	0.700	0.718	01.64	04.23	0.815	0.835	0.847	02.47	04.04
15	0.727	0.700	0.718	03.73	01.23	0.854	0.835	0.847	02.21	07.71
'R^2' values		0.703	0.997				0.803	0.997		
Mean absolute percentage error				17.14	00.81				13.63	01.36
SD of absolute percentage error				15.23	01.57				12.48	01.73
Exp – Experimental; Emp – Empirical model and ANN – Artificial Neural Network Model										

Table 7. Experimental and predicted initial modulus values by empirical and ANN models

3.2 Modelling of Air permeability

The emperical and ANN models were developed from selected fifteen sets of fabric samples as shown in Table 3. The empirical model (7) derived using Box and Behnken factorial design for predicting the air permeability is given below.

$$AP = -8.54\text{E-}3X_1 + 2.695\text{E-}3X_2 - 4.58\text{E-}2X_3 + 3.05\text{E-}6X_1^2 + 9.925\text{E-}6X_2^2 + 3.578\text{E-}4X_3^2 - 1.79\text{E-}5X_1X_2 + 5.076\text{E-}5X_1X_3 - 3.846\text{E-}5X_2X_3 + 5.401 \quad (7)$$

Where, AP= air permeability ($m^3/m^2/s$) X_1 = fabric weight (g/m^2), X_2 = needling density ($punches/cm^2$) and X_3 = percentage polypropylene content in the blend ratio of polypropylene and woollenised jute. Since the coefficient of determination ($R^2 = 0.97$) value is very high, we can conclude that the empirical model fits the data very well.

During training the ANN models for air permeability, the minimum prediction error for all ANN models was obtained within 40,000 cycles (Debnath et al., 2000b). Table 9 depicts the interconnecting weights used for calculating the air permeability of ANN model with three hidden layers, where, W_{mn} – Interconnecting weights between the neuron (m) in one layer and neuron (n) in next layer.

Fabric code	D	Tenacity (cN/Tex)					Initial Modulus (cN/Tex)				
		Exp	Prediction		AE (%)		Exp	Prediction		AE (%)	
			Emp	ANN	Emp	ANN		Emp	ANN	Emp	ANN
16	MD	1.6730	1.9886	1.9960	18.86	19.31	0.4968	0.4445	0.4750	10.53	04.38
	CD	3.7860	4.6575	3.9150	23.02	03.41	0.3123	0.7559	0.2366	142.0	24.24
18	MD	2.2947	1.4784	1.9958	35.57	13.02	0.8467	0.8582	0.8401	01.36	00.77
	CD	4.3700	3.3917	3.9157	22.38	10.40	1.2551	1.2542	1.2434	00.07	00.93
21	MD	0.0240	-2.2031	0.0221	-	07.91	0.3194	0.3875	0.2968	21.32	7.08
	CD	0.0850	-2.3606	0.0975	-	14.71	0.9759	0.8271	1.0112	15.24	3.62
D – Test direction of sample; MD - Machine direction; CD – Cross direction, Exp – Experimental; Emp – Empirical model and ANN – Artificial Neural Network model, AE – Absolute error											

Table 8. Experimental verification of predicted results (tenacity and initial modulus)

Weights between the layers		1st and 2nd	2nd and 3rd	3rd and 4th
	W_{11}	6.110	-21.555	-2.205
	W_{12}	1.811	11.242	-0.073
	W_{13}	-9.048	0.859	-2.135
	W_{21}	-14.213	-2.992	-0.163
	W_{22}	8.363	0.675	-23.549
	W_{23}	-3.274	4.588	-25.085
	W_{31}	-11.762	-10.013	16.168
	W_{32}	1.202	-13.005	-4.871
	W_{33}	-11.006	-2.470	-11.349
Weights between 4th and 5th layers	W_{10}	W_{20}	W_{30}	
	10.465	-8.925	5.433	

Table 9. Weights of ANN model with three hidden layers for air permeability

The Table 10 shows the correlation between experimental and predicted values of air permeability. It is clear that the 'R^2' values for ANN of three hidden layers were maximum followed by empirical model, two layers and single hidden layer ANN respectively. From the Table 10 it can also be observed that the average absolute error was found minimum while using ANN with three hidden layers, followed by ANN with two hidden layers, empirical model and ANN by single hidden layer respectively. The standard deviation of absolute error also follows the same trend. The ANN model with single hidden layer has low correlation between the experimental and predicted values (Debnath et al., 2000b). This may be because the ANN with one hidden layer has only two neurons. Both the number of neurons and the hidden layers are responsible for the accuracy in the predicted model. The ANN with three hidden layers shows the best, predicted results. The empirical model is not as good as ANN of three hidden layers. Though, the correlation between the experimental and predicted values of empirical model is higher than ANN model with two hidden layers, but the mean percentage absolute error is quite high in the case of empirical model than ANN with two or three hidden layers. This is probably due to the fact that the empirical model may require a larger sample size when the relationship between input and output variables is nonlinear (Fan & Hunter, 1998).

Fabric code	Exp AP	Empirical Model		Artificial neural network models					
		Pre AP	AE, %	1 HL Pre AP	AE, %	2 HL Pre AP	AE, %	3 HL Pre AP	AE, %
1	2.285	2.368	03.36	2.426	06.71	2.516	10.10	2.311	01.15
2	2.659	2.543	04.39	2.629	01.27	2.672	00.47	2.671	00.42
3	1.308	1.585	11.40	1.467	12.19	1.506	15.13	1.334	01.98
4	0.966	0.617	36.10	1.425	47.45	0.887	08.21	0.962	00.49
5	2.663	2.495	06.30	2.244	15.72	2.580	03.10	2.665	00.07
6	2.682	2.503	06.67	2.620	02.31	2.612	02.61	2.670	00.47
7	0.786	0.725	07.74	1.379	75.38	0.901	14.66	0.796	01.22
8	1.262	1.391	10.19	1.519	20.31	1.366	08.19	1.395	10.54
9	1.856	1.693	08.75	1.534	17.35	1.639	11.67	1.898	02.26
10	2.361	2.058	12.81	2.197	06.96	2.216	06.15	2.382	00.89
11	1.627	1.664	02.25	1.732	06.45	1.684	03.45	1.701	04.54
12	1.824	1.722	05.63	2.015	10.46	1.867	02.31	1.826	00.09
13	1.675	1.542	07.93	1.676	00.05	1.674	00.70	1.677	00.14
14	1.677	1.542	08.02	1.676	00.05	1.674	00.17	1.677	00.04
15	1.672	1.542	07.79	1.676	00.20	1.674	00.07	1.677	00.29
'R^2'		00.97		00.82		00.96		00.99	
Mean Absolute Error (%)			09.28		14.85		05.79		01.58
SDER			07.94		20.67		05.23		02.73
Exp – Experimental; Emp – Empirical model ; Pre – Predicted; HL – Hidden layer; AE – Absolute error; AP - Air permeability in $m^3/m^2/s$ and SDER – Standard deviation of percentage absolute error									

Table 10. Experimental and predicted air permeability values by empirical and ANN models – absolute error and correlation

3.2.1 Verification of air permeability models

The trained ANN with three hidden layers (3HL) and the empirical models were then used to predict the air permeabilityproperty of six different sets of input pairs. The input variables are selected in such a way that one or two input variables are beyond the range, with which the ANN was trained and empirical model was developed (Table 11).

It can be observed that, the percentage absolute error with ANN, ranges between 00.60 and 14.62. However, the percentage absolute error is between 04.32 and 30.00, while predicting with empirical model. The prediction of air permeability was more accurate with ANN, compared to empirical model even when the inputs are beyond the range of modeling (Debnath et al., 2000b).

3.3 Modelling of compression properties

The ANN models for initial thickness (IT), percentage compression (C), percentage thickness loss (TL) and percentage compression resilience (CR) have been developed from the selected twenty-five sets of fabric samples and corresponding experimental values of compression properties shown in (Table 12).

Fabric code	Fabric weight (g/m^2)	Needling density (punches/cm^2)	Blend ratio (Polypropylene:Jute)	Air permeability ($m^3/m^2/s$)				
				Exp	Predicted values		Absolute error, (%)	
					ANN	Emp	ANN	Emp
20	241	150	00 :100	2.6923	2.6760	3.5000	00.60	30.00
21	310	250	00 :100	2.5641	2.6692	2.9528	04.10	15.15
22	303	350	00 :100	2.8679	2.6728	3.3924	06.80	18.28
23	300	150	20 : 80	2.4576	2.6292	2.3512	06.98	04.32
24	276	250	20 : 80	2.4951	2.6523	2.6497	06.30	06.19
25	205	350	20 : 80	3.1381	2.6791	3.8188	14.62	21.69
Exp – Experimental; Emp – Empirical model and ANN – Artificial Neural Network Model								

Table 11. Experimental verification of predicted results of air permeability values

Fabric code	Fabric weight g/m^2	Needling density punches/cm^2	Woollenised jute %	Polypropylene fibre %	Polyester fibre %	IT mm	C %	TL %	CR %
1	250	150	40	60	-	3.54	53.64	25.46	32.67
2	250	350	40	60	-	3.02	46.73	25.98	32.29
3	450	150	40	60	-	4.41	44.8	20.68	32.92
4	450	350	40	60	-	3.8	36.47	17.68	33.87
5	250	250	60	40	-	3.02	52.48	30.69	29.48
6	250	250	20	80	-	4.27	54.88	27.82	32.27
7	450	250	60	40	-	4.39	37.24	20.69	30.99
8	450	250	20	80	-	3.88	37.8	18.63	31.28
9	350	150	60	40	-	3.45	50.24	25.16	32.77
10	350	150	20	80	-	4.48	50.06	24.49	31.52
11	350	350	60	40	-	3.12	44.91	25.51	31.73
12	350	350	20	80	-	3.38	43.75	23.25	30.99
13	350	250	40	60	-	3.29	45.16	22.06	33.25
14	350	250	40	60	-	3.94	42.45	21.84	33.15
15	350	250	40	60	-	3.66	44.09	21.68	33.33
16	393	150	0	100	-	5.87	54.92	25.05	28.56
17	440	150	0	100	-	5.77	54.97	25.15	28.2
18	392	350	0	100	-	4.08	37.51	17.4	35.05
19	241	150	100	0	-	2.51	41.18	20.61	30.29
20	303	350	100	0	-	2.84	41.85	22.23	30.43
21	300	150	80	20	-	3.18	39.98	18.47	35.32
22	205	350	80	20	-	2.47	47.42	25.22	28.98
23	415	300	-	-	100	3.54	42.93	9.89	54.33
24	515	300	-	-	100	4.14	37.00	8.36	56.69
25	815	300	-	-	100	5.62	23.78	6.65	53.85

Table 12. Experimental design for compression properties

The ANN was trained separately up to certain number of cycles to obtain optimum weights for each compression properties. The number of cycles to achieve optimum weights for

initial thickness, percentage compression, thickness loss (%) and percentage compression resilience are found between 320000 and 5120000 cycles as presented in Table 13. A very large number of simulation cycles was required because more number of input variables was used to develop the ANN model (Debnath & Madhusoothanan, 2008)..

Compression property	Number of cycle		
	One hidden layer	Two hidden layers	Three hidden layers
Initial thickness, mm	2560000	2560000	2560000
Percentage compression	1280000	2560000	5120000
Percentage thickness loss	320000	1280000	2560000
Compression resilience, %	640000	2560000	5120000

Table 13. Optimum number of cycles of one, two and three hidden layered ANN models for compression properties

The optimum weights of ANN for initial thickness, percentage compression, thickness loss (%) and percentage compression resilience are shown in Table 14.

Weights between the layers number	Initial thickness	Percentage compression	Percentage thickness loss	Percentage compression resilience
1st and 2nd				
W_{11}	-7.825	-9.697	-0.797	1.497
W_{12}	-3.144	6.650	1.176	-1.003
W_{13}	0.821	-1.560	1.221	-4.777
W_{14}	3.338	2.949	8.374	14.286
W_{21}	0.394	4.034	2.738	5.181
W_{22}	0.801	-11.441	-4.945	8.240
W_{23}	2.356	-12.284	-0.218	3.091
W_{24}	3.839	0.981	-7.399	-8.415
W_{31}	0.587	4.742	-0.658	-3.937
W_{32}	0.418	2.487	8.743	-2.320
W_{33}	5.436	9.689	-3.318	-2.272
W_{34}	-2.470	8.814	-0.340	0.617
W_{41}	4.336	-0.697	-1.058	2.704
W_{42}	1.140	6.674	-5.424	2.298
W_{43}	-2.877	-11.909	8.539	-3.649
W_{44}	-1.919	-2.500	1.827	4.803
W_{51}	2.555	3.046	0.206	0.552
W_{52}	0.428	-1.342	-1.456	4.349
W_{53}	-3.728	-0.608	-2.002	0.192
W_{54}	-0.958	1.000	1.431	0.350
2nd and 3rd				
W_{11}	-1.958	5.796	2.126	0.474

W_{12}	8.015	10.795	-5.784	-0.253
W_{13}	1.747	0.628	-3.575	6.556
W_{21}	6.622	2.771	0.908	3.378
W_{22}	-2.664	-5.510	4.585	13.901
W_{23}	-2.217	-2.485	0.170	0.471
W_{31}	-1.255	0.661	-1.004	-2.508
W_{32}	-4.467	-1.092	3.731	-8.715
W_{33}	-3.381	7.313	2.431	4.162
W_{41}	-1.670	-6.856	0.762	9.749
W_{42}	-4.480	-3.497	-8.304	-11.644
W_{43}	-1.602	0.590	3.243	-6.180
3rd and 4th				
W_{11}	1.780	-0.951	-1.025	7.269
W_{12}	-4.432	5.588	-6.411	–
W_{21}	-1.488	-0.675	0.401	-14.560
W_{22}	7.351	5.949	9.564	–
W_{31}	-1.375	0.999	3.754	7.599
W_{32}	1.381	-11.087	3.248	–
4th and 5th				
W_{10}	-1.442	-0.432	-1.923	–
W_{20}	13.259	8.769	12.222	–

Table 14. Weights of ANN model for compression properties

Tables 15 to 18 show the experimental and predicted values of initial thickness, compression (%), percentage thickness loss and percentage compression resilience respectively. These tables also indicate the effect of number of hidden layers on the percentage error, standard deviation and correlation between the experimental and predicted results for the corresponding compression properties.

Table 15 shows a very good correlation (R^2 values) between the experimental and the predicted initial thickness values by ANN. Among the results obtained, the ANN with three hidden layers presents comparatively highest R^2 value with lowest error. The standard deviation of percentage absolute error is also found to be less in the case of ANN model with three hidden layers. Similar trend has also been observed in case of percentage compression and percentage thickness loss as depicted in Tables 14 and 15 respectively. The ANN model with two hidden layers performs better in terms of percentage error and standard deviation of percentage error in the case of percentage compression resilience (Table 16). In the cases where average error for the ANN models with three different hidden layers shows more or less similar values, the priority is given to the standard deviation of errors (Debnath & Madhusoothanan, 2008). This study shows that in majority of the cases, the three hidden layered ANN models present better results for predicting compression properties of needle-punched fabrics. Though the three hidden layered ANN models take more time during training phase, the predicted results are more accurate in comparison to ANN models with one and two hidden layers, with less variations in the absolute error (Debnath et al., 2000a).

Fabric code	Initial thickness, mm						
	Exp	ANN Predicted			Absolute error, %		
		1 HL	2 HL	3 HL	1 HL	2 HL	3 HL
1	3.54	3.531	3.539	3.546	0.259	0.034	0.171
2	3.02	3.046	3.019	3.036	0.868	0.030	0.520
3	4.41	4.369	4.398	4.351	0.932	0.266	1.349
4	3.8	3.785	3.780	3.783	0.399	0.524	0.443
5	3.02	3.012	3.012	2.995	0.272	0.261	0.821
6	4.27	4.287	4.267	4.272	0.399	0.071	0.041
7	4.39	4.398	4.383	4.407	0.187	0.149	0.384
8	3.88	3.930	3.878	3.916	1.298	0.053	0.939
9	3.45	3.601	3.538	3.580	4.379	2.564	3.771
10	4.48	4.456	4.482	4.472	0.540	0.043	0.181
11	3.12	3.133	3.166	3.139	0.432	1.479	0.598
12	3.38	3.364	3.389	3.359	0.484	0.256	0.634
13	3.29	3.627	3.648	3.630	10.229	10.870	10.343
14	3.94	3.627	3.648	3.630	7.956	7.421	7.861
15	3.66	3.627	3.648	3.630	0.915	0.338	0.812
16	5.87	5.867	5.870	5.869	0.053	0.002	0.025
17	5.77	5.777	5.771	5.773	0.117	0.017	0.056
18	4.08	4.074	4.087	4.083	0.159	0.168	0.061
19	2.51	2.578	2.614	2.558	2.724	4.124	1.904
20	2.84	2.847	2.857	2.831	0.262	0.603	0.333
21	3.18	3.038	3.030	3.062	4.469	4.708	3.712
22	2.47	2.460	2.440	2.478	0.415	1.200	0.332
23	3.54	3.540	3.540	3.540	0.000	0.003	0.010
24	4.14	4.140	4.140	4.140	0.001	0.006	0.005
25	5.62	5.620	5.620	5.621	0.000	0.004	0.016
R^2	–	0.9868	0.9872	0.9875	–	–	–
Mean of % absolute error		–	–	–	1.51	1.41	1.41
SD of % absolute error		–	–	–	2.6071	2.6932	2.55
Exp – Experimental; 1HL – One hidden layer; 2HL – Two hidden layers; 3HL – Three hidden layers; and SD – Standard deviation							

Table 15. Experimental and predicted values of initial thickness by ANN model

Fabric code	Percentage compression, %						
	Exp	ANN Predicted			Absolute error, %		
		1 HL	2 HL	3 HL	1 HL	2 HL	3 HL
1	53.64	54.126	53.638	53.648	0.906	0.003	0.015
2	46.73	48.817	46.729	46.727	4.467	0.003	0.006
3	44.8	44.536	44.807	44.789	0.589	0.016	0.025
4	36.47	36.223	36.473	36.453	0.677	0.007	0.047
5	52.48	50.449	52.638	52.486	3.869	0.301	0.011
6	54.88	54.333	54.883	54.872	0.997	0.006	0.015
7	37.24	37.576	38.740	37.240	0.902	4.028	0.001
8	37.8	38.590	38.159	37.800	2.089	0.951	0.001
9	50.24	48.230	50.358	50.224	4.001	0.234	0.031
10	50.06	50.703	50.411	50.078	1.285	0.701	0.037
11	44.91	45.650	44.035	44.912	1.648	1.949	0.004
12	43.75	43.949	43.581	43.756	0.454	0.386	0.013
13	45.16	44.244	43.780	43.863	2.028	3.056	2.871
14	42.45	44.244	43.780	43.863	4.227	3.133	3.329
15	44.09	44.244	43.780	43.863	0.350	0.704	0.514
16	54.92	54.807	54.930	54.951	0.205	0.019	0.056
17	54.97	54.896	54.954	54.943	0.135	0.029	0.050
18	37.51	36.873	37.269	37.515	1.699	0.641	0.012
19	41.18	41.666	40.616	41.178	1.181	1.369	0.005
20	41.85	42.787	41.536	41.842	2.240	0.751	0.019
21	39.98	40.793	39.785	39.984	2.033	0.489	0.009
22	47.42	47.242	47.570	47.423	0.376	0.316	0.007
23	42.93	42.933	42.928	42.927	0.007	0.004	0.007
24	37	36.997	37.002	37.003	0.007	0.005	0.007
25	23.78	23.780	23.780	23.791	0.001	0.001	0.047
R^2	–	0.9839	0.9941	0.9971	–	–	–
Mean of % absolute error		–	–	–	1.453	0.764	0.285
SD of % absolute error		–	–	–	1.386	1.117	0.856
Exp – Experimental; 1HL – One hidden layer; 2HL – Two hidden layers; 3HL – Three hidden layers; and SD – Standard deviation							

Table 16. Experimental and predicted values of percentage compression by ANN model

Fabric code	Thickness loss, %							
	Exp	ANN Predicted				Absolute error, %		
		1 HL	2 HL	3 HL		1 HL	2 HL	3 HL
1	25.46	25.547	26.448	25.462		0.341	3.881	0.007
2	25.98	27.399	26.468	25.976		5.462	1.879	0.017
3	20.68	20.574	21.035	20.676		0.515	1.717	0.018
4	17.68	17.147	17.720	17.662		3.013	0.225	0.100
5	30.69	30.660	30.689	30.688		0.096	0.003	0.007
6	27.82	26.361	26.453	27.813		5.244	4.913	0.025
7	20.69	20.634	20.739	20.686		0.271	0.235	0.019
8	18.63	18.564	18.189	18.621		0.357	2.369	0.047
9	25.16	25.200	25.057	25.157		0.159	0.410	0.011
10	24.49	24.554	24.250	24.508		0.261	0.981	0.073
11	25.51	25.488	25.465	25.509		0.087	0.176	0.002
12	23.25	23.236	23.087	23.264		0.060	0.702	0.060
13	22.06	22.064	21.843	21.851		0.017	0.982	0.946
14	21.84	22.064	21.843	21.851		1.024	0.015	0.052
15	21.68	22.064	21.843	21.851		1.770	0.753	0.790
16	25.05	24.994	25.279	25.016		0.225	0.914	0.134
17	25.15	24.733	25.035	25.169		1.657	0.456	0.075
18	17.4	17.817	17.708	17.401		2.396	1.772	0.008
19	20.61	21.149	20.642	20.611		2.614	0.154	0.005
20	22.23	21.340	22.208	22.229		4.002	0.100	0.003
21	18.47	18.334	18.472	18.469		0.734	0.011	0.004
22	25.22	25.207	25.219	25.220		0.053	0.005	0.002
23	9.89	9.876	9.881	9.892		0.144	0.091	0.020
24	8.36	8.368	8.358	8.357		0.096	0.027	0.036
25	6.65	6.652	6.652	6.657		0.037	0.025	0.101
R^2	–	0.9926	0.9954	0.9999		–	–	–
Mean of % absolute error		–	–	–		1.225	0.912	0.102
SD of % absolute error		–	–	–		1.655	1.259	0.234
Exp – Experimental; 1HL – One hidden layer; 2HL – Two hidden layers; 3HL – Three hidden layers; and SD – Standard deviation								

Table 17. Experimental and predicted values of percentage thickness loss by ANN model

Fabric code	Compression resilience, %							
	Exp	ANN Predicted				Absolute error, %		
		1 HL	2 HL	3 HL		1 HL	2 HL	3 HL
1	32.67	32.864	32.568	32.684		0.594	0.312	0.044
2	32.29	32.041	32.253	31.838		0.772	0.115	1.401
3	32.92	30.169	32.805	32.923		8.356	0.350	0.009
4	33.87	33.917	33.640	33.624		0.139	0.679	0.725
5	29.48	29.334	29.375	29.514		0.495	0.357	0.115
6	32.27	32.324	31.931	31.832		0.169	1.051	1.358
7	30.99	31.959	30.700	30.997		3.126	0.935	0.022
8	31.28	30.803	30.890	31.256		1.523	1.248	0.076
9	32.77	33.355	32.304	32.802		1.784	1.422	0.097
10	31.52	30.943	31.071	31.445		1.830	1.425	0.237
11	31.73	31.471	31.735	31.374		0.817	0.016	1.122
12	30.99	31.581	31.029	32.012		1.907	0.127	3.297
13	33.25	33.123	33.162	33.307		0.383	0.266	0.172
14	33.15	33.123	33.162	33.307		0.083	0.035	0.474
15	33.33	33.123	33.162	33.307		0.622	0.505	0.069
16	28.56	29.678	28.624	28.577		3.915	0.223	0.058
17	28.2	29.141	28.083	28.212		3.337	0.414	0.041
18	35.05	34.855	35.006	35.083		0.557	0.125	0.094
19	30.29	30.234	30.215	30.319		0.183	0.249	0.096
20	30.43	30.477	30.399	29.597		0.154	0.103	2.736
21	35.32	35.221	35.130	35.283		0.281	0.537	0.105
22	28.98	29.010	28.998	30.004		0.105	0.064	3.533
23	54.33	54.335	54.340	54.330		0.008	0.018	0.001
24	56.69	56.684	56.687	56.689		0.010	0.005	0.001
25	53.85	53.851	53.837	53.850		0.002	0.025	0.001
R^2	–	0.9919	0.9996	0.9977		–	–	–
Mean of % absolute error		–	–	–		1.2461	0.424	0.635
SD of % absolute error		–	–	–		1.8555	0.450	1.055
Exp – Experimental; 1HL – One hidden layer; 2HL – Two hidden layers; 3HL – Three hidden layers; and SD – Standard deviation								

Table 18. Experimental and predicted values of compression resilience by ANN model

3.3.1 Verification of Models for compression properties

Further, attempts have been made to predict the compression properties to understand the perfection of the models. The ANNs models were then used to four sets of inputs, which have not been utilized during the modeling phase as shown in Table 19. Table 20 indicates the prediction of compression properties and respective absolute errors by ANNs models during verification phase.

Fabric code	Fabric weight g/m²	Needling density punches/cm²	Woollenised jute %	Polypropylene %	Polyester %
18	410	250	0	100	0
21	310	250	100	0	0
24	276	250	80	20	0
28	680	300	0	0	100

Table 19. Samples for experimental verification of ANN model for compression properties

Table 20 presents the predicted compression values of untrained fabric samples by ANN models, showing higher absolute percentage error than the predicted compression values of trained fabric samples as shown in Tables 15 to 18. Specifically, in case of sample code 28, all the properties predicted during verification are high. Two samples of this category (100% jute) have been used during the training phase (Table 10). This might be the reason for higher error in sample code 28 (Debnath & Madhusoothanan, 2008). Hence, the learning process by ANN itself is very poor compared to other samples, this ultimately increases the error during verification (Table 20).

Fabric code	Initial thickness with 3 hidden layer mm			Compression with 3 hidden layer %			Thickness loss with 3 hidden layer %			Compression resilience with 3 hidden layer %		
	E	P	A	E	P	A	E	P	A	E	P	A
18	5.07	5.25	3.53	38.07	54.88	44.17	17.87	19.17	7.26	34.12	30.73	9.95
21	2.47	3.17	28.47	43.96	29.20	33.59	22.38	27.85	24.46	32.89	17.41	47.05
24	3.00	2.91	3.09	41.53	48.57	16.95	21.59	30.68	42.12	30.71	27.80	9.48
28	5.13	5.26	2.54	22.35	23.95	7.15	6.19	6.76	9.24	54.21	56.62	4.44
E – Experimental; P – Predicted and A – Absolute error %												

Table 20. Experimental verification of predicted results on compression properties

4. Conclusions

From this study it is clear that the tensile and air permeability property of needle punched non-woven fabric can be predicted from two different methodologies– empirical and ANN models. The ANN model for prediction of tensile properties of needle punched non-woven is much more accurate compared to the empirical model. Prediction of tensile properties by ANN model shows considerably lower error than empirical model even when the inputs were beyond the range of inputs, which were used for developing the model.

It can also be concluded that ANNs can be used effectively even for predicting nonlinear relationship between the process parameters and fabric properties.
Both the methods can be implemented successfully as far as the air permeability of such needled fabric is concerned. The prediction accuracy of the ANN with three hidden layers is the best amongst all the predicting models used in this work. The ANN with three hidden layers is the best, which, gives highest correlation with lowest prediction error between actual and predicted values of air permeability of needle punched non-woven. The ANN with three hidden layers also shows lesser error when compared to an empirical model even when input variables are extrapolated over which the models were developed.
ANNs can be used effectively for predicting nonlinear relationship between the process parameters and the fabric compression properties.
The number of cycles to achieve optimum weights for initial thickness, percentage compression, thickness loss (%) and percentage compression resilience are found between 320000 and 5120000 cycles.
There is a very good correlation (R^2 values) with minimum error between the experimental and predicted initial thickness, percentage compression and thickness loss values by ANN with three hidden layers.
The standard deviation of percentage absolute error is also found to be less in the case of ANN model with three hidden layers for initial thickness, percentage compression and percentage thickness loss. The ANN model with two hidden layers performs better in terms of percentage error and standard deviation in the case of percentage compression resilience.
The three hidden layered ANN models take more time for computation during training phase but the predicted results are more accurate with less variations in the absolute error in the verification phase.
Based on the experiences the ANN model can be well used to model and predict other important properties of needle-punched nonwoven fabrics made of different fibre materials.

5. References

Box, G. E. P. & Behnken, D. W. (1960). Some New Three Level Designs for the Study of Quantitative Variables. *Technometrics*, Vol.2, No.4, 455-475, ISSN 0040-1706

Debnath, C. R. & Roy, A. N. (1999). Mechanical behaviour of needle punched textiles of jute nonwovens. *Indian Textile Journal*, Vol.110, No.3, 50-53, ISSN 0019-6436

Debnath, S.; Madhusoothanan, M. & Srinivasmoorthy, V. R. (2000a). Modelling of tensile properties of needle-punched nonwovens using artificial neural networks. *Indian Journal of Fibre & Textile Research*, Vol.25, No.1, 31-36, ISSN 0971-0426

Debnath, S.; Madhusoothanan, M. & Srinivasmoorthy, V. R. (2000b). Prediction of air permeability of needle-punched nonwoven fabrics using artificial neural network and empirical models. *Indian Journal of Fibre and Textile Research*, Vol.25, No.4, 251-255, ISSN 0971-0426

Debnath, S., Nag, D., De, S. S., Ganguly, P. K., & Ghosh, S. K. (2006). Studies on mechanical and hydraulic properties of JGT for geo-technical applications. *Journal of The Institution of Engineers (India)*, Vol.TX86, No.2, 46-49, ISSN 0257-4438

Debnath, S. & Madhusoothanan, M. (2007). Compression behaviour of jute-polypropylene blended needle-punched nonwoven fabrics. *Indian Journal of Fibre and Textile Research*, Vol.32, No.4, 427-433, ISSN 0971-0426

Debnath, S. & Madhusoothanan, M. (2008). Modeling of compression properties of needle-punched nonwoven fabrics using artificial neural network. *Indian Journal of Fibre & Textile Research*, Vol.33, No.4., 392-399, ISSN 0971-0426

Debnath, Sanjoy & Madhusoothanan, M. (2009a). Compression properties of polyester needlepunched fabric. *Journal of Engineered Fibres and Fabrics*, Vol.4, No.4, 14-19, ISSN 1558-9250

Debnath, S. & Madhusoothanan, M. (2009b). Studies on compression behaviour of polypropylene needle punched non-woven fabrics. *Journal of The Institution of Engineers (India)*, Vol.TX89, No.2, 34-37, ISSN 0257-4438

Debnath, Sanjoy & Madhusoothanan, M. (2010a). Water absorbency of jute-polypropylene blended needle-punched nonwoven. *Journal of Industrial Textiles*, Vol.39, No.3, 215-231, ISSN 1528-0837, DOI: 10.1177/1528083709347121.

Debnath, Sanjoy & Madhusoothanan, M. (2010b). Thermal insulation, compression and air permeability of polyester needle-punched nonwoven. *Indian Journal of Fibre and Textile Research*, Vol.35, No.1, 38-44, ISSN 0971-0426

Fan, J. & Hunter, L. (1998). A worsted fabric expert system, Part-2: An artificial neural network model for predicting the properties of worsted fabrics. *Textile Research Journal*, Vol.68, No.10., 763-771, ISSN 0040-5175

Gong, R. H. & Chen, Y. (1999). Predicting the performance of fabrics in garment manufacturing with artificial neural networks. *Textile Research Journal*, Vol.69, No.7., 477-482, ISSN 0040-5175

Hearle, J.W.S. & Sultan, M.A.I. (1967). A study of needled fabrics, Part-I: Experimental methods and properties. *Journal of Textile Institute*, Vol.58, Part-1, No.6, 251-265, ISSN 0040-5000

Kothari, V. K., & Das, A. (1992). Compression behaviour of nonwoven geotextiles. *Geotextiles and Geomembranes*, Vol.11, 235-253, ISSN 0266-1144

Kothari, V. K., & Das, A. (1993). Compression behaviour of layered needle-punched nonwoven geotextiles. *Geotextiles and Geomembranes*, Vol.12, 179-191, ISSN 0266-1144

Luo, C., and David, A. L. (1995). Yarn strength prediction using neural networks, Part I: Fibre properties and yarn strength relationship. *Textile Research Journal*, Vol.65, No.9., 495-500, ISSN 0040-5175

Midha, V. K., Alagirusamy, R. & Kothari, V. K. (2004). Studied on properties of hollow polyester needle-punched fabrics. *Indian Journal of Fibre & Textile Research*, Vol.29, No.4., 391-399, ISSN 0971-0426

Park, S. W., Hwang, Y. G., Kang, B. C. & Yeo, S. W. (2000). Applying fuzzy logic and neural networks to total hand evaluation of knitted fabrics. *Textile Research Journal*, Vol.70, No.8., 675-681, ISSN 0040-5175

Postle, R. (1997). Fabric catagorisation by means of objective measurement and neural networks. *Textile Asia*, Vol.28, No. 2, 33-34, ISSN 0049-3554

Rajamanickam, R., Hansen, S. M. & Jayaraman, S. (1997). Analysis of the modelling methodologies for predicting the strength of air-jet spun yarns. *Textile Research Journal*, Vol.67, No.1., 39-44, ISSN 0040-5175

Ramesh, M. C., Rajamanickam, R., & Jayaraman, S. (1995). The Prediction of yarn tensile properties by using artificial neural networks. *Textile Research Journal*, Vol.86, No.3., 459-469, ISSN 0040-5175

Sao, K.P. & Jain, A. K. (1995). Mercerization and crimp formation in jute. *Indian Journal of Fibre and Textile Research*, Vol.20, No.4, 185-191, ISSN 0971-0426

Sengupta, A. K., Sinha, A. K. & Debnath, C. R. (1985). Needle-punched non-woven jute floor coverings: Part III – Air permeability and thermal conductivity. *Indian Journal of Fibre & Textile Research*, Vol.10, No.4., 147-151, ISSN 0971-0426

Subramaniam, V., Malathi, Lokanadam, B., Kumari. N. & Chandramohan, G. (1990). A simple method of measuring the handle of fabrics and softness of yarns. *Journal of Textile Institute*, Vol.81, Part-1, No.1., 94-97, ISSN 0040-5000

Vangheluwe, L., Sette, S., & Pynckels, F. (1993). Assessment of set marks by means of neural nets. *Textile Research Journal*, Vol.63, No.4., 244-246, ISSN 0040-5175

Vangheluwe, L., Sette, S., & Kiekens P. (1996). Modelling relaxation behaviour of yarns, Part-II: Backpropagation neural network model. *Journal of Textile Institute*, Vol.87, Part-1, No.2., 305-310, ISSN 0040-5000

Wen Chen P., Chun Liang T., Fai You H., Li Sun W., Chueh Wang N., Chyilin H. and Cherng Lien R. (1998). Classifying textile faults with a back propagation neural network using power spectra. *Textile Research Journal*, Vol.68, No.2., 121-126, ISSN 0040-5175

Xu, B., Fang, C. & Watson, M. D. (1999). Clustering analyses for cotton trash classification. *Textile Research Journal*, Vol.69, No.9., 656-662, ISSN 0040-5175

Zhu, R. & Ethridge, M. D. (1997). Predicting hairiness for ring and rotor spun yarns and analysing the impact of fibre properties. *Textile Research Journal*, Vol.67, No.9., 694-698, ISSN 0040-5175

Part 2

Materials Science and Industry

4

Artificial Neural Networks for Material Identification, Mineralogy and Analytical Geochemistry Based on Laser-Induced Breakdown Spectroscopy

Alexander Koujelev and Siu-Lung Lui
Canadian Space Agency[1]
Canada

1. Introduction

Artificial Neural Networks (ANN) are used nowadays in a broad range of areas such as pattern recognition, finances, data mining, battle scene analysis, process control, robotics, etc. Application of ANN in the field of spectroscopy has generated a long-standing interest of scientists, engineers and application specialists. The ANN' capability of producing fast, reliable and accurate spectral data processing has become, in many cases, a bridging mechanism between science and application. A particular example of how ANN can transform plasma emission spectroscopy, that is quit challenging to model, into a turnkey ready to use device is described in this Chapter.

Laser-Induced Breakdown Spectroscopy (LIBS) is a material-composition analytical technique gaining increased interest last decade in various application fields, such as geology, metallurgy, pharmaceutical, bio-medical, environmental, industrial process control and others (Cremer & Radziemski, 2006; Miziolek et al., 2006). It is in essence a spectroscopic analysis of light emitted by the hot plasma created on a sample by the laser-induced breakdown. LIBS offers numerous advantages as compared to the standard elemental analysis techniques (X-ray fluorescence or X-ray diffraction spectroscopy, inductively coupled plasma spectroscopy, etc.), such as: capability of remote analysis in the field, compact instrumentation, detection of all elements and high spatial resolution. Such features as minimum or no sample preparation requirement and dust mitigation using "cleaning" laser shots are especially important for field geology and remotely operated rover-based instruments.

As result, LIBS instruments have been selected as payloads for the 2011 Mars Science Laboratory mission led by the National Aeronautics and Space Administration (Lanza et al., 2010) and the ExoMars mission on Mars planned for 2018 and led by European Space Agency (Escudero-Sanz et al., 2008).

Despite of the advantages, the main challenge is still the retrieval of accurate information from measured spectra. LIBS spectral signals, composed mostly of narrow emission lines, are complex nonlinear functions of concentrations of measured constituents and instrument

parameters. The most important contributors to this nonlinearity are spectral overlapping, self-absorption, and the so-called matrix effects. These effects are caused by chemical properties and morphological features of the sample matrix that can change the intensity of the emitted lines (Eppler et al., 1996; Harmon et al., 2006). In addition, the ambience such as pressure, temperature and gas type can vary the heat loss and confinement effect in LIBS that results in a change of spectra (Iida, 1990; Lui & Cheung, 2003). All this leads to large errors in concentration measurements of minor or trace elements performed in different materials. This became a serious impeding factor for using full advantages of LIBS in analytical geochemistry in either field geology or planetary exploration.

Common quantitative spectral data processing algorithms, based on calibration curve method have been successfully applied in some cases (St-Onge et al., 2002; Cho et al., 2001), but they are limited to application in one class of material and require *a priory* knowledge about the tested sample. An alternative method, called calibration-free method, relies on plasma model to calculate plasma temperature using several spectral lines. It shows encouraging results, however also subject to a number of limitations (Ciucci et al., 1999; Aguilera et al., 2009).

Classification & identification techniques are also used in conjunction with LIBS to define material identity and even composition. In relatively simple cases classification and identification of samples can be achieved by evaluating the line ratios or the patterns of a LIBS spectrum (Mönch et al., 1997; Samek et al., 2001; Sattmann et al., 1998). More sophisticated classification methods such as, principle components analysis (PCA), soft independent modeling of class analogy (SIMCA) and partial least-squares discriminant analysis (PLS-DA), have been studied and produced very promising results (Sirven et al., 2007; Clegg et al., 2009). However, the above techniques being based on linear processing have difficulty to take into account nonlinear effects.

ANN data processing offers to address the above challenges as having the potential to solve nonlinear problems (Gurney, 1997; Haykin, 1999). The capabilities of ANN in this area have started to be explored recently almost simultaneously by few groups. Inakollu (Inakollu et al., 2009) used ANN to predict the element concentrations in aluminium alloys from its LIBS spectrum. Ferreira (Ferreira et al., 2008) selected a set of wavelengths through the "wrapper" algorithm and then determined the concentration of copper in soil samples by ANN. Sattmann (Sattmann et al., 1998) discriminated PVC from other polymers with the distinct chlorine 725.66 nm line. Ramil (Ramil et al., 2008) classified the LIBS spectra of 36 archaeological ceramics into three groups by ANN. The possibility of using ANN to predict composition in natural rocks explored in our earlier works by Motto-Ros (Motto-Ros et al., 2008) and Koujelev (Koujelev et al., 2009). We also demonstrated the capability of mineral and rock sample identification with LIBS combined with ANN (Koujelev et al., 2010). The potential of ANN to analyse LIBS spectra has been proven in these studies.

It is important to note that performing LIBS on geological material: minerals, rocks, and soils, is especially challenging. These materials can vary from silica-based basalt rock to iron-rich hematite mineral. They exhibit serious matrix effect thus the conventional calibration curve method will not be applicable for quantitative study (retrieval of composition). Most importantly, without prior knowing the matrix identity, choosing an appropriate calibration curve is impossible. Identification, or qualitative analysis, is also difficult to achieve since there are over few thousand types of minerals so learning all their spectra seems impractical. In fact, applying LIBS on rocks, soils or minerals have been reported in several studies. Menut (Menut et al., 2006) demonstrated the potential of probing europium in argillaceous rocks preconditioned in europium solution. Sharma

(Sharma et al., 2007) combined LIBS with Raman spectroscopy to evaluate mineral rocks. Bousquet (Bousquet et al., 2007) measured the chromium concentration in 22 soil samples doped with chromium. Calibration curve was obtained from the five kaolinite soil samples only. They also performed classification by principal components analyses (Sirven et al., 2006). Belkov (Belkov et al., 2009) showed the possibility of measuring the carbon content in 11 soil samples. The calibration curve was fit by an exponential function within 2% to 8% range. Gaft (Gaft et al., 2009) evaluated the performance of LIBS in sulphur analyses of minerals, alloys, and coal mixtures. Two calibration curves were established for two sets of coal mixtures. From these examples one can observe that the application of LIBS on geological analysis is mainly demonstrative and descriptive. Samples were artificially doped and the calibration curves were conditional, either limited by sample type or concentration range. In term of quantitative and qualitative aspects, the application on geological samples still remains challenging to the LIBS community.

This chapter presents a review of our earlier work as well as some new results. The focus is made on how we apply and optimise ANN in a particular spectroscopy application. The chapter is structured in the following way. After the introduction, the section describing the principles of LIBS will be presented so, that the particularities of the LIBS data are introduced. Some pre-processing techniques are presented in the LIBS section. The next, section is devoted to different ANN architectures used for particular types of data analysis and the targeted applications. The first sub-section describes material identification analysis, the second sub-section describes quantitative mineralogy analysis, and the third sub-section describes quantitative elemental analysis. Conclusions and future works are discussed in the last section of the chapter.

2. LIBS technique

Before we discuss different ANN spectral processing schemes, it is important to define the experimental settings where the raw spectra are obtained. It is also very important to address what types of materials are studied and what pre-processing routines are applied before the data are inputted to the network.

A typical LIBS system includes a laser, optical elements to focus laser beam and to collect plasma emission, and a spectrometer (Fig. 1). In our studies, the laser source is a Q-switch Nd:YAG laser (Spectra Physics, LPY150, 1064nm, 7 ns) operating at 1 Hz repetition rate with pulse energy of 20 mJ. The pulse energy is monitored by Joule-meter and adjusted by a $\lambda/2$ plate and a polariser. The beam is focused to a 50 μm spot to ablate the sample. The plasma emission is collected and delivered to the Ocean Optics LIBS 2000 spectrometer (200 – 970 nm, 0.1 mm resolution) through an optical fibre. In the majority of our experiments, the distance between the sample and the collection optics was 10 cm. The delays between instruments are controlled by a pulse delay generator (BNC 575). The spectra are recorded and analysed with a computer and dedicated software. It worth noting that these parameters are typical for a low-power LIBS system that may be used on a remote platform, such as planetary rover, or as a hand-held instrument in the field conditions.

Different types of geological materials are studied in our experiments. The samples of standard geological materials, mostly silicates in our cases, are supplied in form of powder with certified elemental composition by the Brammer Standard Company Inc. They are pressed into tablets for easy handling and sampling. Another set of natural rock and mineral samples is obtained from Miners Inc (part number: K4009). For these samples, only major

composition elements were known based on the type of mineral. Powder-based samples are used to train, validate and test the composition retrieval algorithm, while the natural rocks and minerals are used only to test the mineral identification capability.

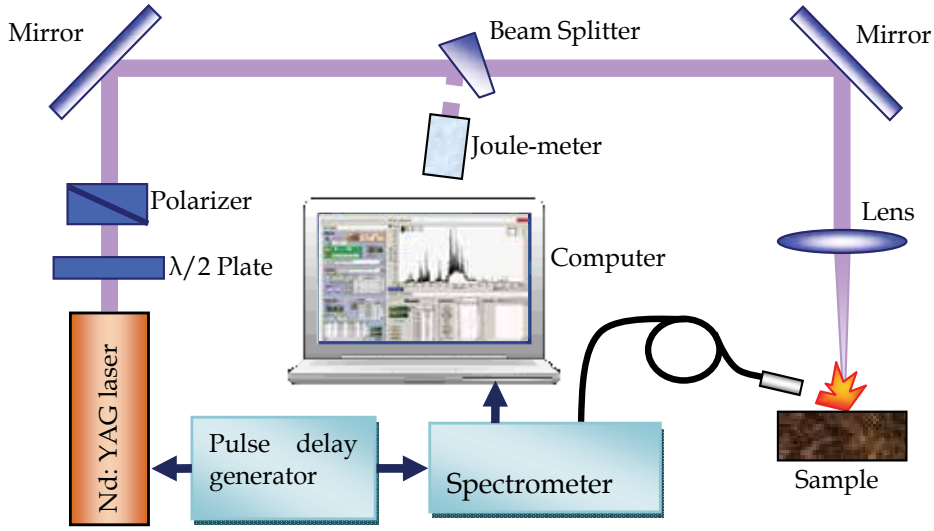


Fig. 1. Experimental configuration of a LIBS system.

	Concentration (fraction)								
Std name	SiO_2	Al_2O_3	MgO	CaO	Na_2O	K_2O	TiO_2	Fe_2O_3	MnO
Rock71306	0.0062	0.001	0.218	0.3002	0.0003	0.00038	0.00015	0.0021	0.00108
AndesiteJA1	0.6397	0.1522	0.0157	0.057	0.0384	0.0077	0.0085	0.0707	0.00157

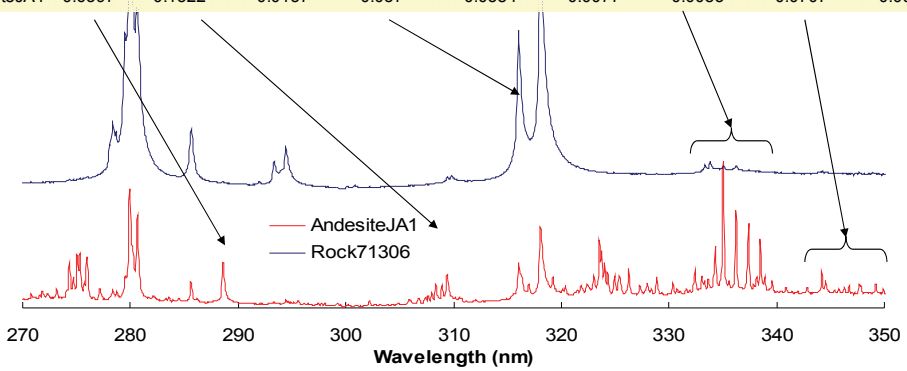


Fig. 2. Examples of LIBS spectra for materials with different composition.

Let us consider few examples of raw LIBS spectra. Spectral signatures of a carbonate rock (Rock 71306) and an andesite (JA1) are shown in Fig. 2. Due to large difference in compositions of these two materials, their discrimination can be easily arranged. Here, a monitoring of intensities of several key atomic lines (Si, Al, Ca, Ti and Fe in this case) can be employed. Therefore, identification or classification of types of minerals with a strong difference in composition can be easily achieved using simple logic algorithms. In this case, we rather care about the presence of specific spectral lines than the exact measurement of their intensity and correspondence to elemental concentration.

The situation however, can be much more complex when one deals with identification of materials with high degree of similarity, or with retrieval of compositional data (quantitative analysis). Such an example is presented in Fig. 3. Here the strategy for these two applications may diverge. Such, that for material identification the spectral lines showing the largest deviations between materials (Mg in this example) should be used. However, for quantitative analysis it is rather useful to select the spectral lines that exhibit near-linear correspondence of the intensity and the element concentration (Ti 330 nm – 340 nm lines in this example). This is why the material identification and quantitative analysis that will be discussed in the following sections rely on different spectral line selection.

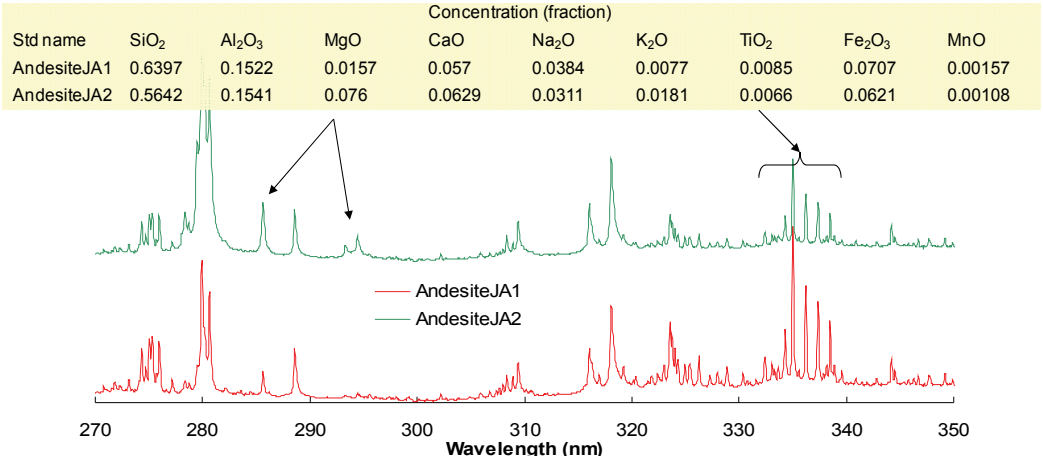


Fig. 3. Examples of LIBS spectra for materials with similar composition.

Once LIBS spectra are acquired from the sample of interest, several pre-processing steps are performed. Pre-processing techniques are very important for proper conditioning of the data before feeding them to the network and account for about 50 % of success of the data processing algorithm. The following major steps in data conditioning are employed before the spectral data are inputted to the ANN.

a. Averaging of LIBS spectra. Usually, averaging of up to a hundred of spectral samples (laser shots) may be used to increase signal to noise ratio. The averaging factor depends on experimental conditions and the desired sensitivity.
b. Background subtraction. The background is defined as a smooth part of the spectrum caused by several factors, such as, dark current, continuum plasma emission, stray light, etc. It can be cancelled out by use of polynomial fit.
c. Selection of spectral lines for the ANN processing. Each application requires its own set of selected spectral lines for the processing. This will be discussed in greater details in the following sections.
d. Calculation of normalised spectral line intensities. In order to account for variations in laser pulse energy, sample surface and other experimental conditions the internal normalization is employed. In our studies, we normalize the spectra on the intensity of O 777 nm line. This is the most convenient element for normalization since all our samples contain oxygen and there is always a contribution of atmospheric oxygen in the spectra in normal ambient conditions. The line intensities are calculated by integrating the corresponding spectral outputs within the full width half-maximum (FWHM) linewidth.

After this pre-processing, the amount of data is greatly reduced to the number of selected normalized spectral line intensities, which are submitted to the ANN.

3. ANN processing of LIBS data

The ANN usually used by researchers to process LIBS data and reported in our earlier works is a conventional three-layer structure, input, hidden, and output, built up by neurons as shown in (Fig. 4). Each neuron is governed by the log-sigmoid function. The first input layer receives LIBS intensities at certain spectral lines, where one neuron normally corresponds to one line.

A typical broadband spectrometer has more than a thousand channels. Inputting to the network the whole spectrum increases the network complexity and computation time. Our attempts to use the full spectrum as an input to ANN were not successful. As a result, we selected certain elemental lines as reference lines to be an input to ANN. General criteria for the line selection are the following: good signal to noise ratio (SNR); minimal overlapping with other lines; minimal self-absorption; and no saturation of the spectrometer channel.

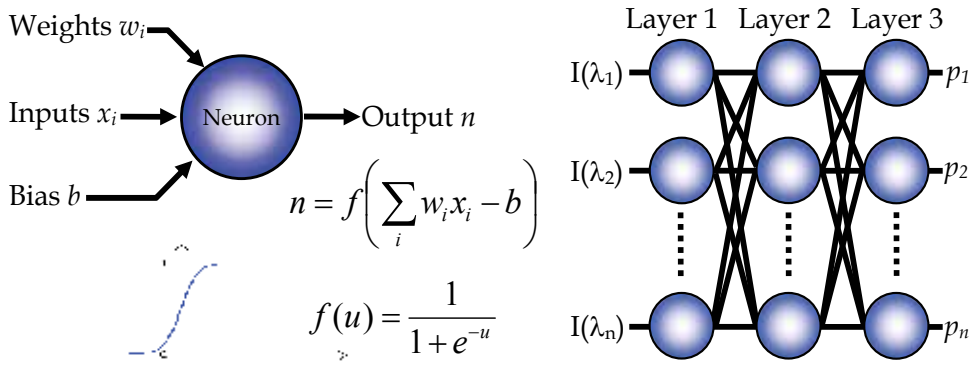


Fig. 4. Basic structure of an artificial neural network.

These criteria eliminate many lines which are commonly used by other spectroscopic techniques. For example, the Na 589 nm doublet saturates the spectrometer easily, thus is not selected. The C 247.9 nm can be confused with Fe 248.3 nm, therefore is avoided. At the same time, the relatively weak Mg 881 nm line is preferred to 285 nm line since it is located in a region with less interference from other lines. In addition to these general rules, some specific requirements for line selection imposed by particular applications are discussed in the following sections.

The number of neurons in the hidden layer is adjusted for faster processing and more accurate prediction. Each neuron at the output layer is associated either to a learnt material (identification analysis) or an element which concentration is measured (quantitative analysis). The output neurons return a value between 0 and 1 which represents either the confidence level (CL) in identification or a fraction of elemental composition in quantitative processing.

The weights and biases are optimized through the feed-forward back-propagation algorithm during the learning or training phase. To perform ANN learning we use a

training data set. Then to verify the accuracy of the ANN processing we use validation data set. Training and validation data sets are acquired from the same samples but at different locations (Fig. 5). In this particular example ten spectra collected at each location and averaged to produce one input spectrum per location. Five cleaning laser shots are fired at each location before the data acquisition.

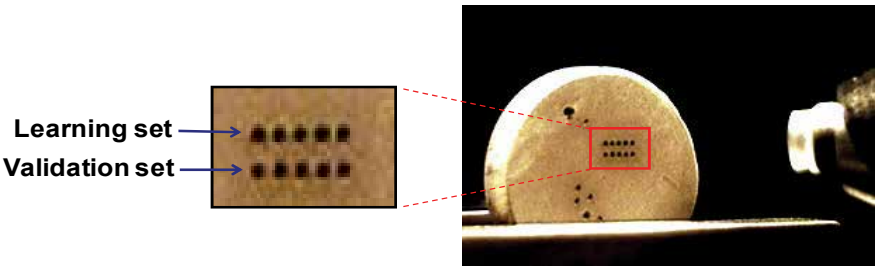


Fig. 5. Acquiring learning and validation spectra from a pressed tablet sample. The ten spots on the left are laser breakdown craters corresponding to the data sets. An emission collection lens is shown on the right in the picture.

3.1 Material identification

Material identification has been demonstrated recently with a conventional three-layer feed-forward ANN (Koujelev et al., 2010). High success rate of the identification algorithm has been demonstrated with using standard samples made of powders (Fig. 6). However, a need for improvements has been identified to ensure the identification is stable with given large variations of natural rocks in terms of surface condition, inhomogeneity and composition variations (Fig. 7). Indeed, the drop in identification success rate between validation set and the test set composed of natural minerals and rocks is from 87 % to 57 % (Fig. 6). Note, at the output layer, the predicted output of each neuron may be of any value between 0 (complete mismatch) and 1 (perfect match). The material is counted as identified when the ANN output shows CL above threshold of 70 % (green dashed line). If all outputs are below this threshold, the test result is regarded as unidentified. Additional, soft threshold is introduced at 45 % (orange dashed line) such that if the maximum CL falls between 45 % and 70 %, the sample is regarded as a similar class.

An improved design of ANN structure incorporating a sequential learning approach has been proposed and demonstrated (Lui & Koujelev, 2010). Here we review those improvements and provide a comparative analysis of the conventional and the constructive leaning network.

Achieving high efficiency in material identification, using LIBS requires a special attention to the selection of spectral lines used as input to the network. In addition to the above described considerations, we added an extra rational for the line selection. Lines with large variability in intensity between different materials, having pronounced matrix effects were preferred. In such a way we selected 139 lines corresponding to 139 input nodes of the ANN. The optimized number of neurons in the hidden layer was 140, and the number of output layer nodes was 41 corresponding to the number of materials used in the training phase.

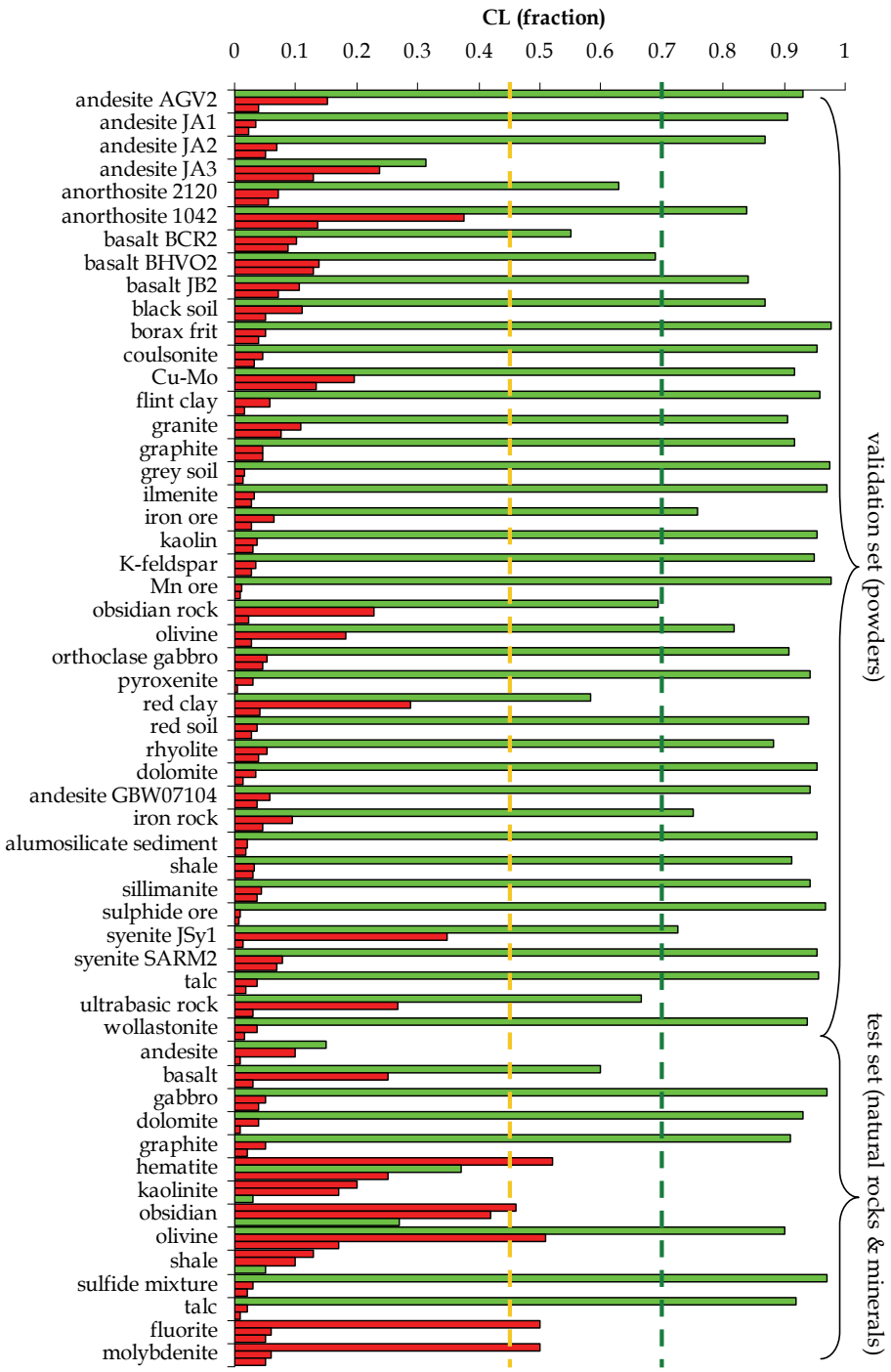


Fig. 6. Identification results for ANN with conventional training: powder tablets validation and natural rock & mineral test. Green colour corresponds to confidence levels for correct identification and red colour corresponds to mis-identification ANN outputs.

Fig. 7. Natural rock & mineral samples and their powder tablets counterparts.

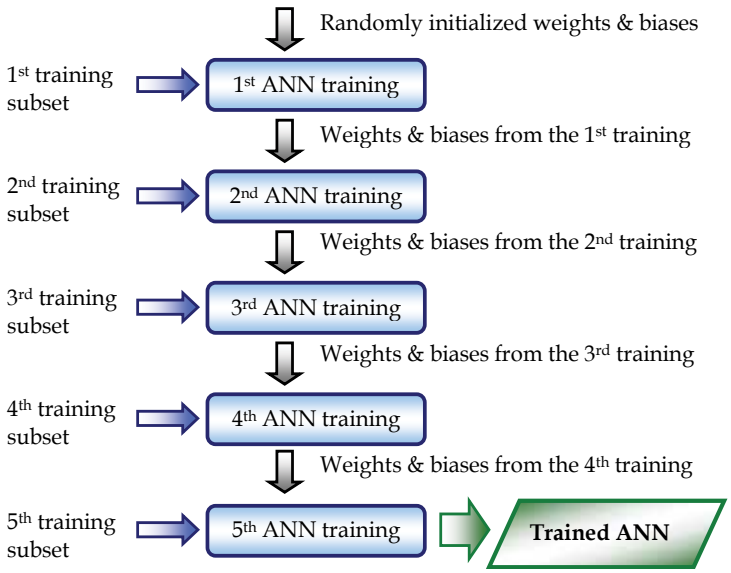


Fig. 8. Sequential training diagram.

When dealing with a conventional training the identification success rate drops rapidly if natural rock samples are subject to measurement on the ANN trained with powder made samples. This is, as we believe, due to overfitting of ANN. To avoid overfitting, the number of training cases must be sufficiently large, usually a few times more than the number of variables (i.e., weights and biases) in the network (Moody, 1992). If the network is trained

only by the average spectrum of each sample corresponding to 41 training cases, then the ANN is most likely to be overfitted. To improve the generalization of the network, the sequential training was adopted as an ANN learning technique (Kadirkamanathan et al., 1993; Rajasekaran et al., 2002 and 2006).

The early stopping also helps the performance by monitoring the error of the validation data after each back-propagation cycle during the training process. The training ends when the validation error starts to increase (Prechelt, 1998). In our LIBS data sets there are five averaged spectra per sample, each used in its own step of the training sequence. At each step, the ANN is trained by a subset of spectra with the early stopping criterion and the optimized weights and biases are transferred as the initial values to the second training with another subset. This procedure repeats until all subsets are used.

The algorithm implementation is illustrated in (Fig. 9). While the mean square error (MSE) decreases going through 5 consecutive steps (upper graph), the validation success rate grows up (bottom graph).

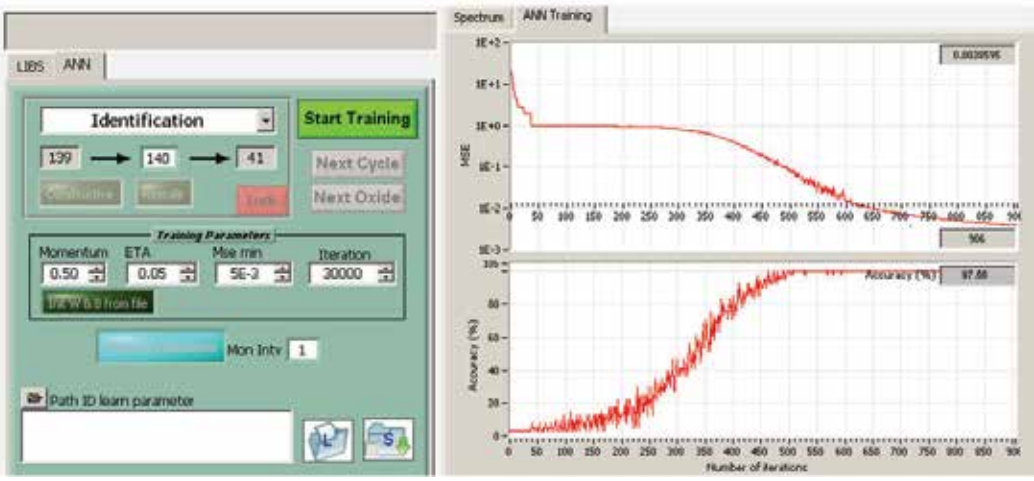


Fig. 9. Identification algorithm programmed in the LabView environment: the training phase.

Using a standard laptop computer the learning phase is usually completed in less than 20 minutes. Once the learning is complete, the identification can be performed in quasi real time. The LIBS-ANN algorithm and control interface is shown in (Fig. 10).

Identification can be performed on each single laser shot spectrum, on the averaged spectrum, or continuously. The acquired spectrum displayed is of the Ilmenite mineral sample in the given example. When the material is identified, the composition corresponding to this material is displayed. Note, that the identification algorithm does not calculate the composition based on the spectrum, but takes the tabular data from the training library. The direct measurement of material's composition is possible with quantitative ANN analysis.

In the event if the sample shows low CL for all ANN outputs it is treated as unknown. In such a case, more spectra may be acquired to clarify the material identity. If it is confirmed by several measurements that the sample is unknown to the network, it can be added to the

training library and the ANN can be re-trained with the updated dataset. Thus, for a remote LIBS operation, this mode "learn as you go" adds frequently encountered spectra on the site as the reference spectra. This mode offers a solution for precise identification without dealing with too large database of reference materials spectra beforehand. The exact identity or a terrestrial analogue (in case of a planetary exploration scenario) can be defined based on more detailed quantitative analysis, possibly, in conjunction with data from other sensors.

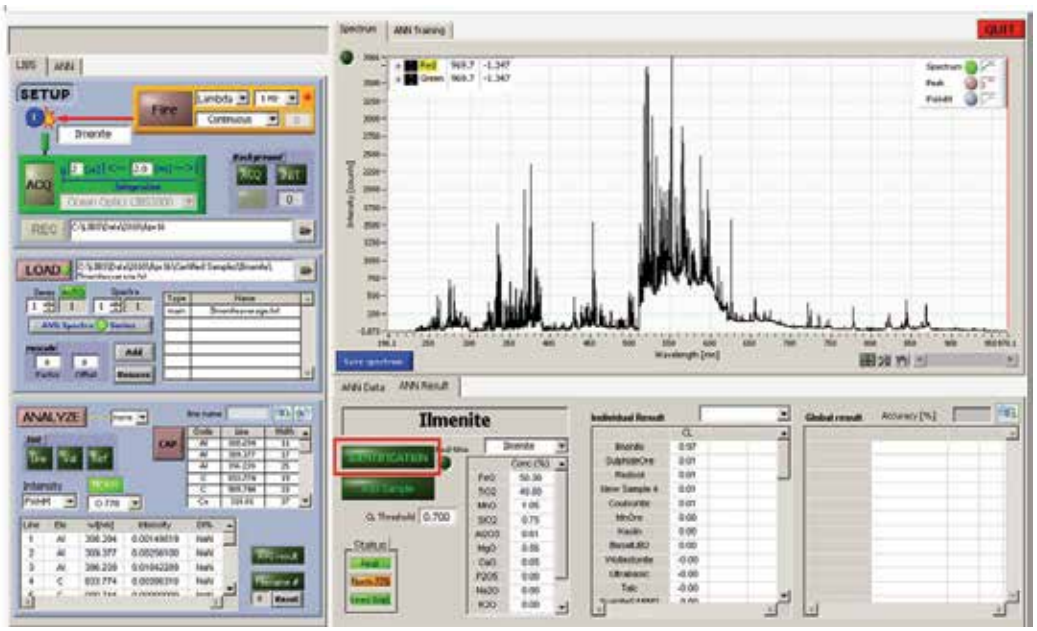


Fig. 10. Identification algorithm programmed in the LabView environment: how it works for a test sample that has been identified. Upper-left section defines the hardware control parameters. Bottom-left section defines the spectral analysis parameters (spectral lines). Top-right part displays the acquired spectrum. Bottom-right section displays identification results.

The results of validation and natural rock test identification are shown in (Fig 11) in the form of averaged CL outputs. The CL values corresponding to mis-identification (red) are lower than for the conventional training, especially for the part with natural rocks. All identifications are correct in this case. The standard powder set includes similar powders of andesite, anorthosite and basalt which are treated as different classes during the trainings. Therefore, non-zero outputs may be obtained for their similar counterparts. The lower red outputs in sequential training suggests it is more subtle to handle similar class. Note that both training methods confuse andesite JA3, with other andesites. According to the certified data, the concentrations of major oxides for JA3 always lie between those of other andesites. As a result, there are no distinct spectral features to differentiate JA3 from other andesites. Therefore, mis-identification in this particular case can be acceptable.

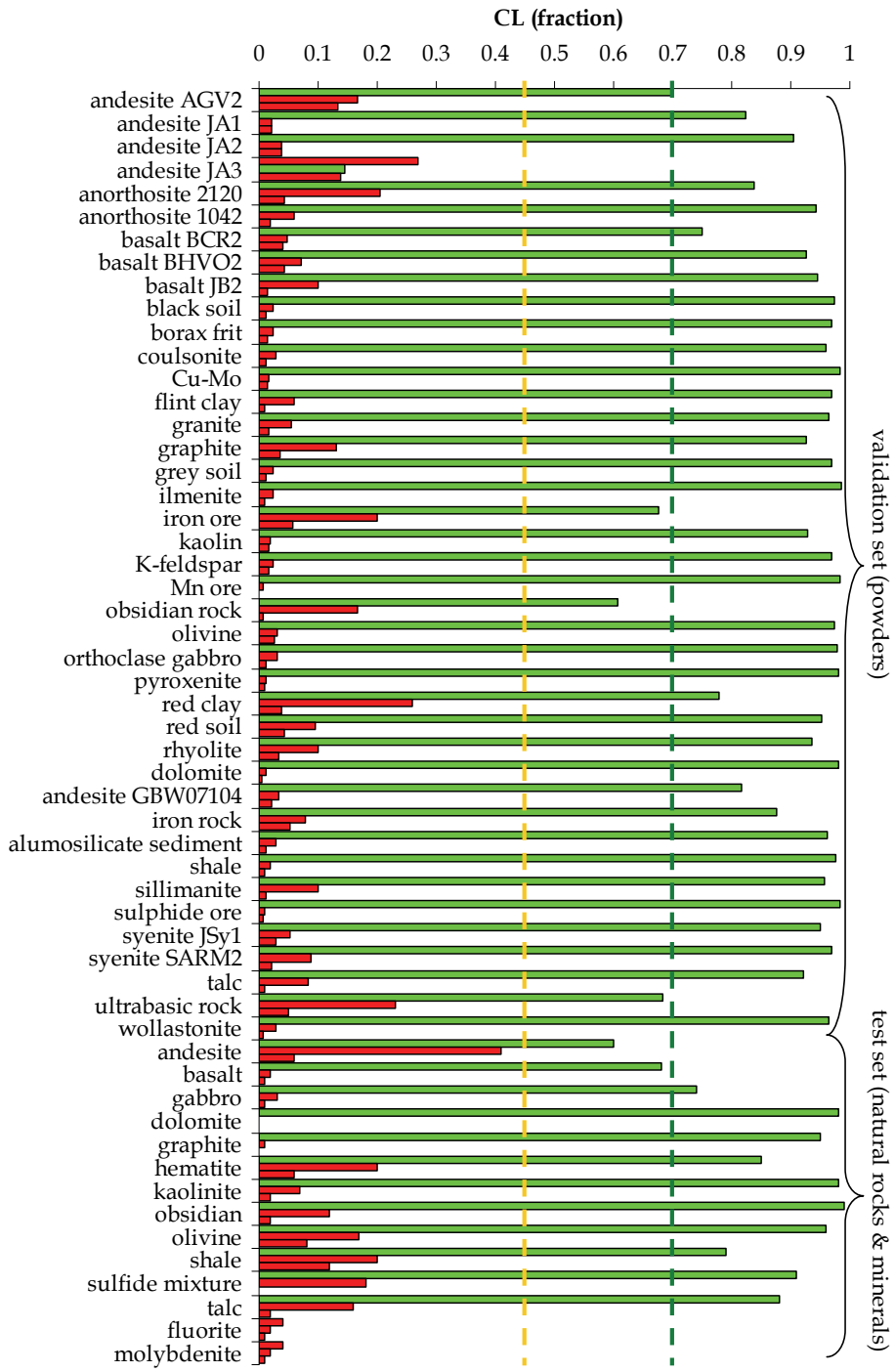


Fig. 11. Identification results for ANN with sequential training: powder tablets validation and natural rock & mineral test. Green colour corresponds to confidence levels for correct identification and red colour corresponds to mis-identification ANN outputs.

The last two samples, fluorite and molybdenite, are selected to evaluate the network's response to an unknown sample. The technique is capable of differentiating new samples. Certainly, if our certified samples included fluorite or molybdenite, the ANN would have been spotted these samples easily due to the distinct Mo and F emission lines.
The comparative of summary the results of the ANN with sequential training with those of another ANN trained by conventional method are shown in Table 1. Here, the conventional method is referred as a single training with one average spectrum for each sample. The prediction of the sequential LIBS-ANN improves with the increasing number of sequential trainings. After the 5th training, its performance surpasses that of the conventional LIBS-ANN. The rate of correct identification rises from 82.4% to 90.7%, while the incorrect identification rate drops from 2% to 0.5%. This is equivalent to only two false identifications out of 410 test spectra from the validation set. The rock identification shown is done on 50-averaged spectra. The correct identification rate for the sequential training method is 100%. In conventional training, it is only 57% with the rest results regarded as "undetermined". The outstanding performance of the sequential ANN shows a better generalization and robustness of the network.

Material set	Training method		Average rate (%)			
			Classified			
			Correct	Misidentified	Success within classified samples	Unidentified
Validation set (powders)	Conventional		87.1	2.0	97.9	11.0
	Sequential training	After 1st	82.4	2.0	96.7	15.6
		After 3rd	88.5	1.7	97.5	9.8
		After 5th	90.7	0.5	99.5	8.8
Test set (natural rocks & minerals)[1]	Conventional		57.1	0	100	42.9
	Five level sequential training		100	0	100	0

Table 1. Validation and test result of the ANN trained by sequential and conventional methods. Average spectrum of a sample is used for testing.

3.2 Mineralogy analysis

Measuring presence of different minerals in natural rock mixtures is an important analysis that is commonly done in geological surveys. On one hand, LIBS relies on atomic spectral signatures directly indicating elemental composition of the material, therefore material crystalline structure does not appear to be present in the measurement. On the other hand, the information on the material physical and chemical parameters is present in the LIBS signal in a form of matrix effect. This, in fact, means that materials with the same elemental

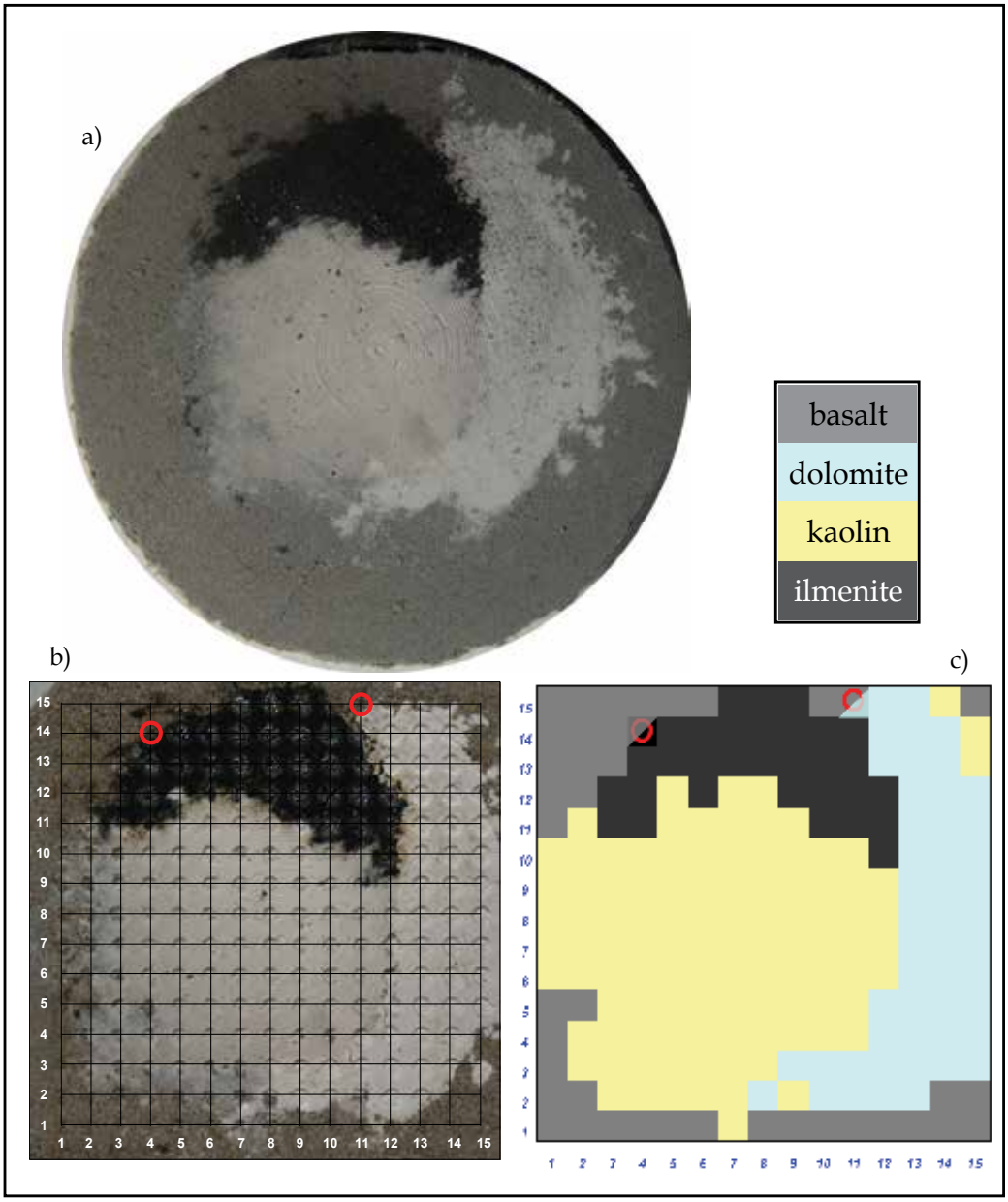


Fig. 12. Mineralogy analysis on the sample made of mixture of basalt, dolomite, kaolin and ilmenite. Red circles indicate unidentified prediction.

composition but different crystalline structure (or other physical or chemical properties) produce LIBS spectra with different ratios of spectral line intensities. Thus, mineralogy analysis can be done based on LIBS measurement where the ratios & intensities of the spectral lines are processed to deduce the identity of the mineral matrix.

One can implement this using the identification algorithm described in the previous section. The methodology relies on a series of measurement produced in different locations of the

rock, soil or mixture, where only one mineral type is identified in each location. Then, the quantitative mineralogy content in percents is generated for the sample based on the total result.

In this section, we describe a mineralogy analysis algorithm and tests that were performed in a particular low-signal condition. LIBS setup, described earlier, was used with a larger distance between the collection aperture and a sample. The distance was increased up to 50 cm thus resulting in 25 times smaller signal-to-noise ratio. This simulates realistic conditions of a field measurement. Since a lens of longer focal length was used, a larger crater was produced.

Because of low-signal condition, we adjusted ANN structure to produce result that is more reliable. First, the peak value is used in this case instead of FWHM-integrated value used earlier to represent the spectral line intensity. In a condition of weak lines, the FWHM value is difficult to define. Second, the intensities of several spectral lines per element were averaged to produce one input value to the ANN. Consequently, the ANN structure included 10 input nodes (first layer) corresponding to the following input elements: Al, Ca, Fe, K, Mg, Mn, Na, P, Si and Ti. The output layer contained 38 nodes corresponding to the number of mineral samples in the library. The hidden layer consisted of 40 neurons. The sequential training described above was used.

In order to test the performance of quantitative mineralogy, an artificial sample was made based on the mixture of certified powders. Four minerals such as, ilmenite, basalt, dolomite and kaolin, were placed in a pellet so that clusters with visible boundaries can be formed after pressing the tablet (Fig. 12a). The measurements were produced by a map of 15x15 locations with a spacing of 1 mm where LIBS spectra were taken (Fig. 12b). Ten measurement spectra were taken at each location. They are averaged and processed by ANN algorithm.

Figure 12c shows the resulting mineralogy surface map. Since the colours of mineral powders were different, one may easily compare the accuracy of the LIBS mineralogy mapping with the actual mineral content. The results of the scan are summarised in the Table 2. The achieved overall accuracy is 2.5 % that is an impressive result demonstrating the high potential of the technique.

Mineral	Basalt	Dolomite	Kaolin	Ilmenite
LIBS-ANN measurement, %	17.8	21.8	45.8	13.8
True value, %	22.2	18.2	46.9	12.7
Deviation, %	4.4	3.6	1.1	1.1
Average deviation, %	2.5			

Table 2. Test result of the LIBS-ANN mineralogy mapping.

It should be noted that the true data are calculated as percentages of the mineral parts present on the scanned surface. These percentages are not representative of the entire surface of the sample or volume content. This becomes an obvious observation if one

considers that the large non-scanned area at the edge of the sample is covered by basalt, while its abundance is small on the scanned area. Therefore, the selection of the scanning area becomes very important issue if the results are to be generalised on entire sample.

3.3 Quantitative material composition analysis

The mineralogy analysis based on identification ANN can be used to estimate material elemental composition. This estimation however may largely deviate from true values, because it is based on the assumption that each type of mineral (or reference material) has well defined elemental composition. In reality, the concentrations of the elements may vary in the same type of mineral. Moreover, very often one element can substitute another element (either partially or completely) in the same type of mineral.

This section describes the ANN algorithm for quantitative elemental analysis based directly on the intensities of spectral lines obtained by LIBS. The ANN for quantitative assay requires much higher precision than the sample identification. The output neurons now predict the concentrations, which can range from parts per million up to a hundred percents. Thus, to improve the accuracy of the prediction, we introduce the following changes to the structure of a typical ANN and the learning process.

In our earlier development of quantitative analysis of geological samples, the ANN consisted of multiple neurons at the output layer. Each output neuron returned the concentration of one oxide (Motto-Ros et al., 2008). This network, however, can suffer from undesirable cross-talk. During training process, an update of any weights or biases by one output can change the values of other output neurons, which may be optimized already. Therefore, in this current algorithm, we propose using several networks and each network has only one output neuron dedicated to one element's concentration (Fig. 13). For geological materials, we use conventional representation of concentration of element's oxide form.

Similar to identification algorithm in low-signal condition, the spectral lines identified for the same element are averaged producing one input value per element. This minimizes the noise due to individual fluctuation of lines.

Since the concentration of the oxide can cover a wide range, during the back-propagation training, the network unavoidably favour the fitting of high concentration values and cause inaccurate predictions at low concentration elements. To minimize this bias, the input and desired output values are rescaled with their logarithm to reduce the data span and increase the weight of the low-value data during the training.

Without the matrix effect, the concentration of an element can simply be determined by the intensity of its corresponding line by using a calibration curve. In reality, the presence of other elements or oxides introduces non-linearity. To present this concept in an ANN, additional inputs corresponding to other elements are added. Those inputs however should be allowed to play only secondary role as compared to the input from the primary element. In other words, the weights and biases of the primary neurons should weight more than others should.

To implement this idea, the ANN training is split into two steps. In the first training, only the average line intensity of the oxide of interest is fed to the network. This average intensity is duplicated to several input neurons to improve the convergence and accuracy. The weights and biases obtained from this training are carried forward to the second training of

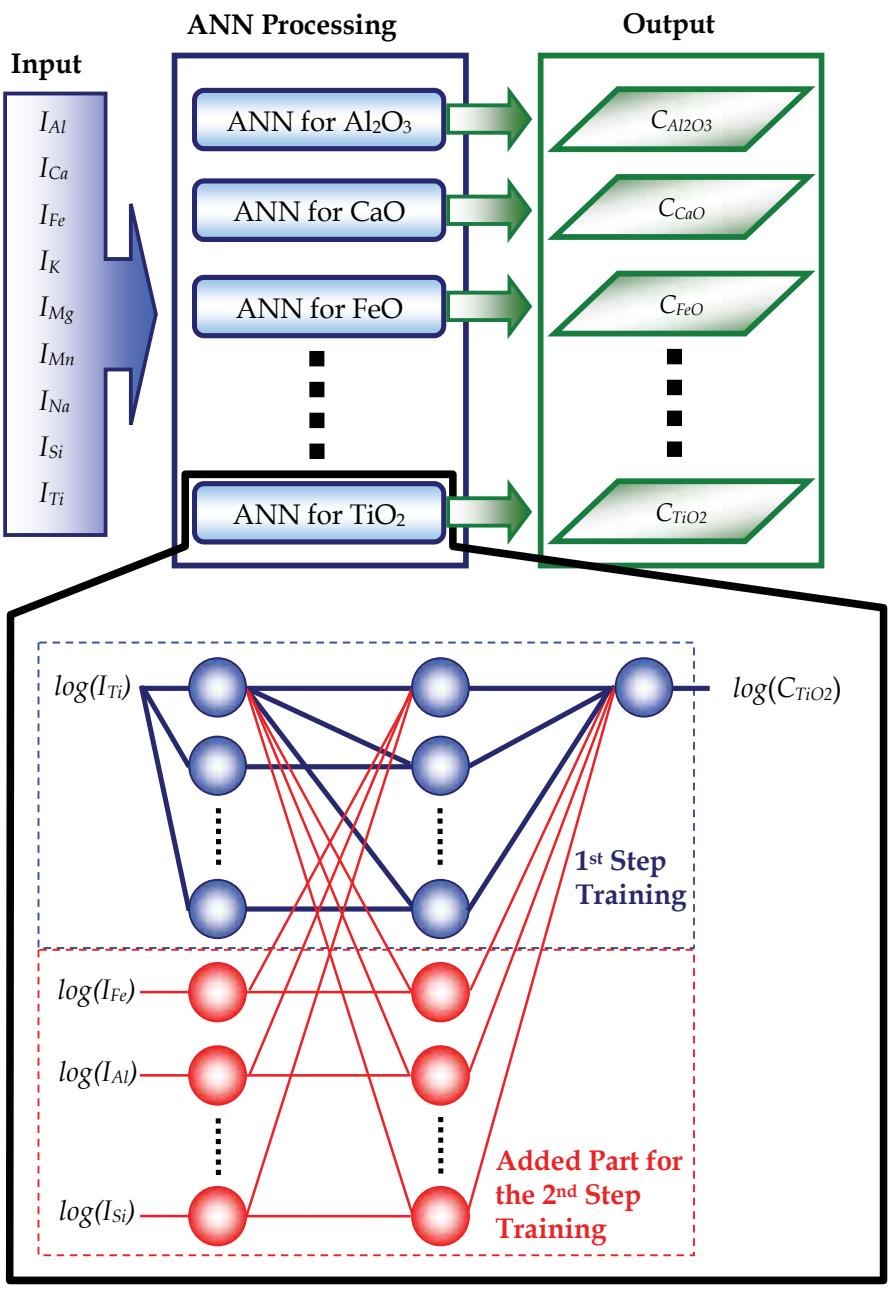


Fig. 13. Architecture of the expanded ANN for the constructive training. The blue dashed box indicates the structure of the ANN corresponding to the 1st step training. The red dashed box shows the neurons and connections added to the initial network (blue) during the 2nd training (constructive). In the 2nd training, the weights and biases of the blue neurons are initialled with the values obtained from the first training, while the weights and biases of the red neurons are initialized with small values much lower than those of blue neurons.

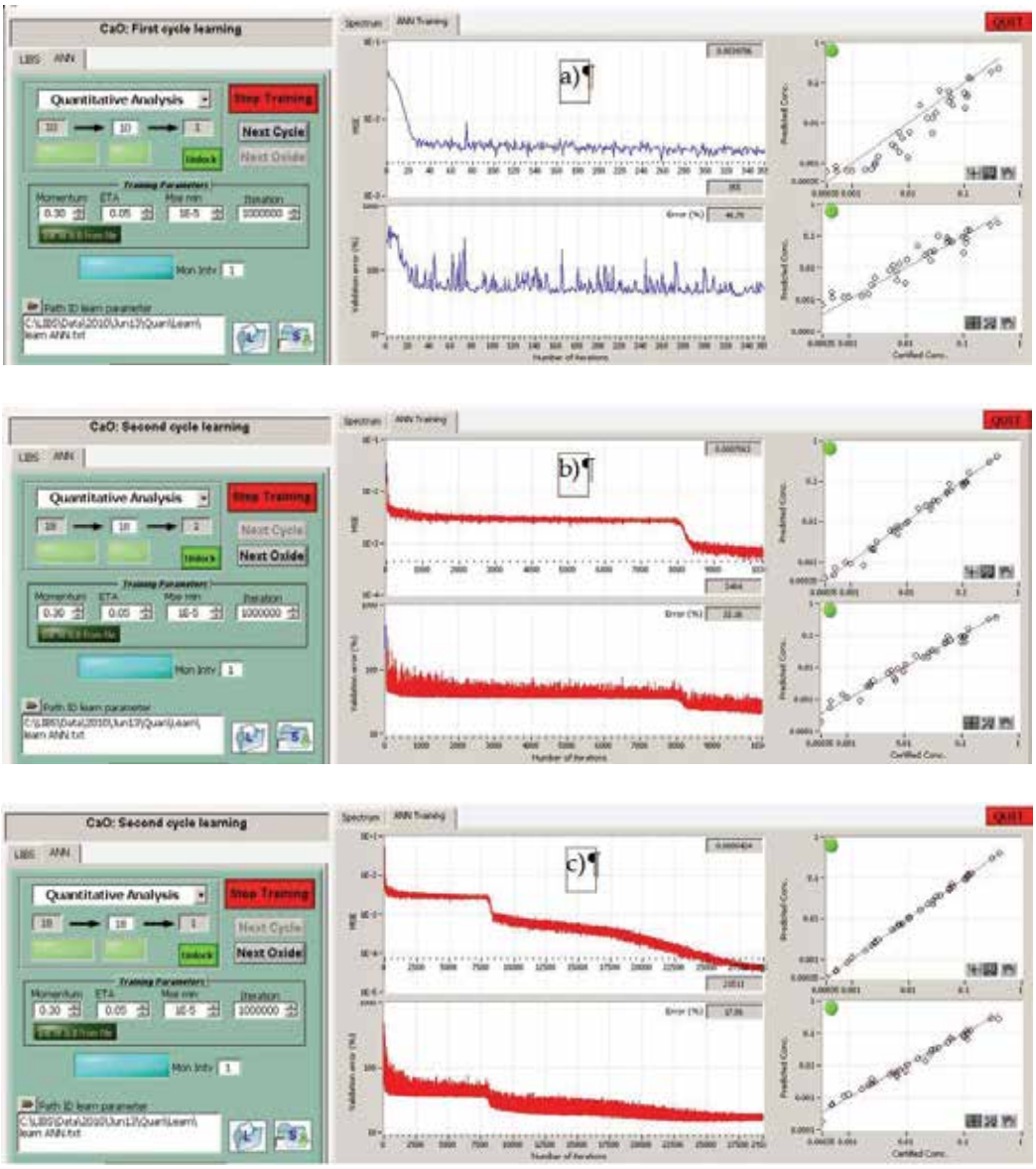


Fig. 14. Screenshots of the training interface of the quantitative LIBS-ANN algorithm programmed in LabView environment. Dynamics of the ANN learning and validation error while training is shown: (a) – during the 1st step training; (b) – in the beginning of the 2nd step training; (c) – at the end of the training. On each screenshot: the menu on the left defines training parameters; the graph in middle-top shows mean square error (MSE) for the training set; the graph in middle-bottom shows MSE for the validation set; the graph in right-top shows predicted concentration *vs.* certified concentration for the training set; the graph in right-bottom shows predicted concentration *vs.* certified concentration for the validation set.

a larger network. The expanded network is constructed from the first network with additional neurons which handle other spectral lines. This two-step training is referred as constructive training. Accuracy is verified by validation data set simultaneously with training (Fig. 14).

This figure illustrates training dynamics on the ANN part responsible for CaO measurement. In the first step of training the ANN has one input value per material that is copied to 10 input neurons. The number of hidden neurons is 10 and there is only one output neuron. As we see, the validation error is very noisy and reaches rather big value at the end of the training (~50%) (Fig. 14a). Concentration plot shows large scattering. When the second training starts the error goes down abruptly. In this case the network is expanded to 18 input neurons (10 for CaO line and 8 for the rest of elements, one input per element). The number of hidden neurons is 18 and there is one output neuron corresponding to CaO concentration. The validation error and the level of noise get gradually reduced. At the end of the training it reaches 17 % (averaged value for the data set). Taking into account that the span of data reaches four orders of magnitude, this is a very good unprecedented performance.

A comparison of the performance between a typical ANN using conventional training and a re-structured ANN with constructive training is shown in (Fig. 15a, b). In general, the predictions by the constructive ANN fall excellently on the ideal line (i.e., predicted output corresponds to certified value). Although the performance is similar at high concentration region (>10%), the data from the conventional ANN method start to deviate at low concentration regime. The scattering of data becomes very large at the very low concentration region (< 0.1%). Some data points fall outside the displayable range of the plot (e.g. the low concentrated TiO_2 and MnO). This observation supports the importance of data rescaling for accurate predictions at low concentration range.

The performance of validation for different oxides is summarized in Table 3. The validation by the constructive method is significantly better than that of the conventional training. The deviation of all predictions is less than 20%. The prediction of SiO_2 concentration is similar in both approaches since it is the most abundant oxide in almost all samples. For the conventional ANN method, the deviations of most prediction are in general higher. This is attributed to the cross-talk of the neurons. The deviation for MnO is incredibly large as it is usually in the form of impurity of tens of ppm. Thus the bias in training makes the prediction of these low concentrated oxides less accurate.

Oxide	Al_2O_3	CaO	FeO	K_2O	MgO	MnO	Na_2O	SiO_2	TiO_2
Constructive ANN error (%)	17.7	14.1	14.3	16.9	14.0	18.9	10.7	7.7	16.6
Conventional ANN error (%)	21.3	33.3	44.2	33.4	53.2	152.5	35.9	7.3	86.6

Table 3. A comparison of the validation error between the constructive and conventional ANN.

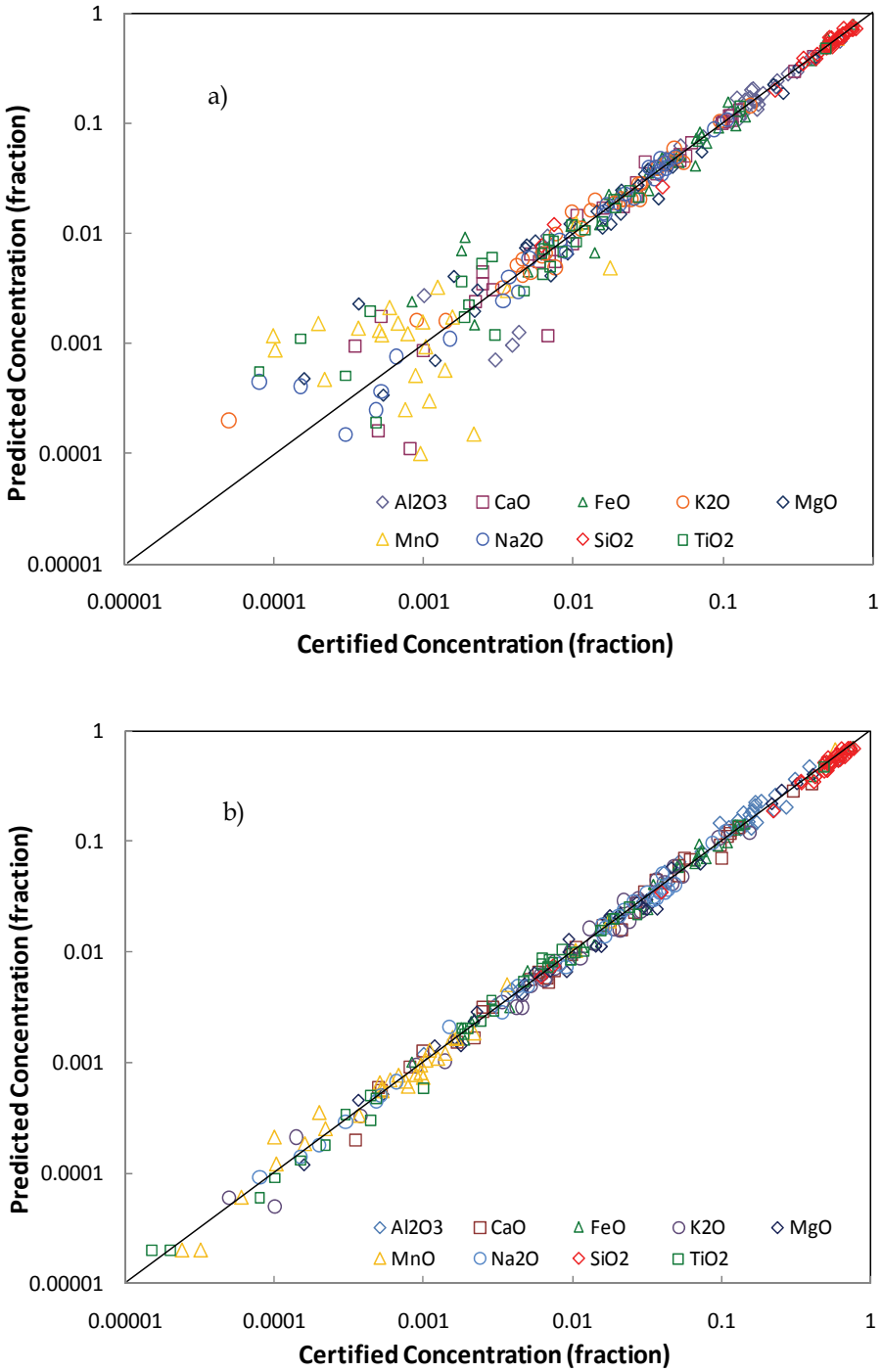


Fig. 15. A comparison of the validation performance between a typical ANN with conventional training (a) and the ANN with constructive training (b).

The prediction of oxide concentration by the constructive ANN is evaluated by four certified samples, which were not part of the training process. They were unknown to network thus simulating a new sample. The oxide concentrations obtained are compared with those calculated using the calibration curve method and a conventional ANN algorithm (Fig. 16). Among these three techniques, both the calibration curve method and the conventional ANN give inaccurate prediction for most oxides (Table 4).
For the calibration curve method, the deviation is mainly due to the serious matrix effects of the geological samples.

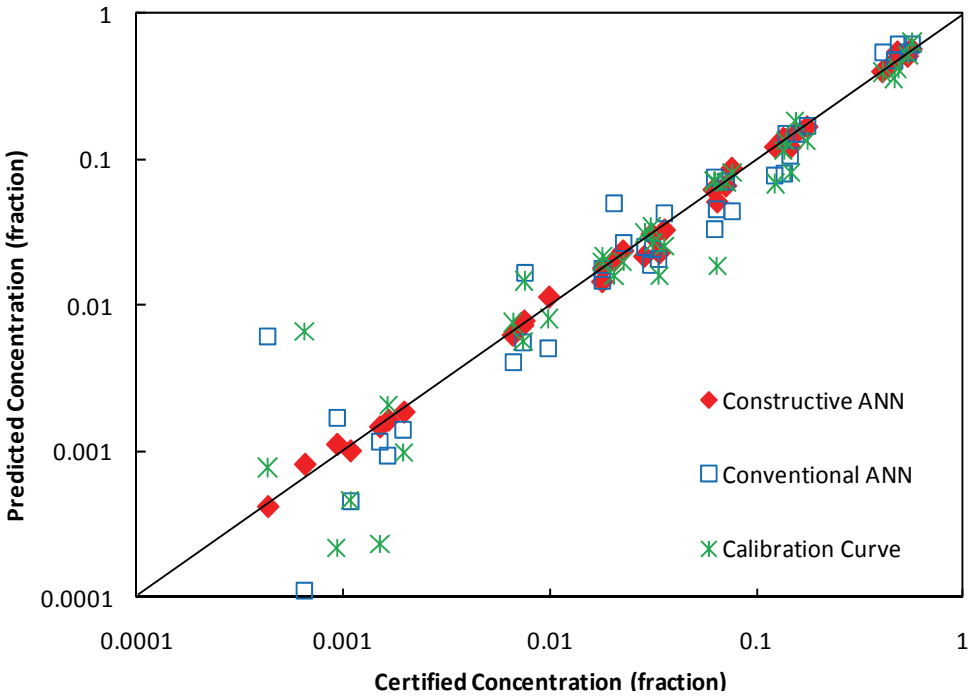


Fig. 16. Comparison of the concentration prediction of the four samples (andesite JA2, basalt BCR2, iron ore, orthoclase gabbro) by the constructive ANN, conventional ANN and the calibration curve method.

The prediction of SiO_2 has the least deviation as it is the major constitution (i.e., the matrix) of the samples. Minor components such as Al_2O_3, CaO, FeO and MgO have errors of about 20 to 30%. Impurities, like MnO, Na_2O and TiO_2, suffer most from the matrix effect and have the worst predictions, which is 40% to 250% inaccuracy.
The conventional ANN has comparable result as that of the calibration curve. Yet their deviation is caused by the limitation of the ANN discussed earlier. The errors for MnO, Na_2O and TiO_2 are still the worst at 50% to over 300% level. For Al_2O_3, CaO and FeO, the variations are around 20%. However, due to cross-talking of the output neutrons, the prediction of SiO_2 is even worse than that obtained from the calibration curve method. Nevertheless, the predictions at low concentration scattered seriously, revealing the bias of high-concentration fitting during the training process.
With the modified ANN, the accuracy of the prediction is drastically enhanced. Those scattered data from the calibration curve method and classical ANN at the low

concentration region are now brought back to the ideal line. Both the major oxides (SiO_2 and Al_2O_3) and the impurities (MnO and Na_2O) have similar performance of deviations below 20%. The matrix effect and the poor accuracy at low concentration that appear in other methods are no longer observed in the optimized constructive ANN technique.

Oxide	Al_2O_3	CaO	FeO	K_2O	MgO	MnO	Na_2O	SiO_2	TiO_2
Constructive ANN deviation (%)	2.8	10.2	0.6	6.0	16.7	8.0	8.1	5.6	10.7
Conventional ANN deviation (%)	18.1	24.1	22.9	47.0	25.3	47.2	71.6	17.8	360.3
Calibration curve deviation (%)	20.3	19.6	20.9	37.6	29.0	67.2	241.3	8.3	40.0

Table 4. The average deviation of the prediction from the certified value for each oxide of the four unknown samples.

Given the success of these two types of analysis demonstrated above: identification and quantitative, we merged them in one software tool to facilitate data analysis (Fig. 17).
The identification part uses ANN with 139 input neurons, 140 hidden and 41 output neurons, and the quantitative ANN uses constructive architecture. Two outputs are produced from a single LIBS data acquisition: material identification and its composition prediction. Even if the sample cannot be identified, its composition is still accurately predicted.

4. Conclusion

We demonstrate application of supervised ANN architectures to spectroscopic analysis based on LIBS data. Two distinct processing approaches are described targeting material identification and quantitative material composition analysis.
In the first application, such features as early stopping and sequential training are introduced enabling exceptional robustness of the algorithm. While the algorithm was trained using standard powder-based samples, a 100% successful identification is achieved using set of natural rocks and minerals as test samples. Application of material identification in quantitative mineralogy analysis is demonstrated using artificial mineral mixture. Overall accuracy of 2.5% is achieved.
In the second application, we introduced constructive learning to ensure algorithm stability and robustness, but at the same time to account for matrix effects. The accuracy better than 20% is achieved for nine elements measured in their oxide form (Al_2O_3, CaO, FeO, K_2O, MgO, MnO, Na_2O, SiO_2 and TiO_2) in the working range from 10 parts per million up to a hundred percent. It is worth noting that this accuracy is reached with no assumption on the type of the material. Geological samples of mineralogy different than those used for training the algorithm were successfully tested. This demonstrates the ability of the constructive ANN technique to overcome highly nonlinear multi-dimensional problem caused by matrix effects in LIBS data.

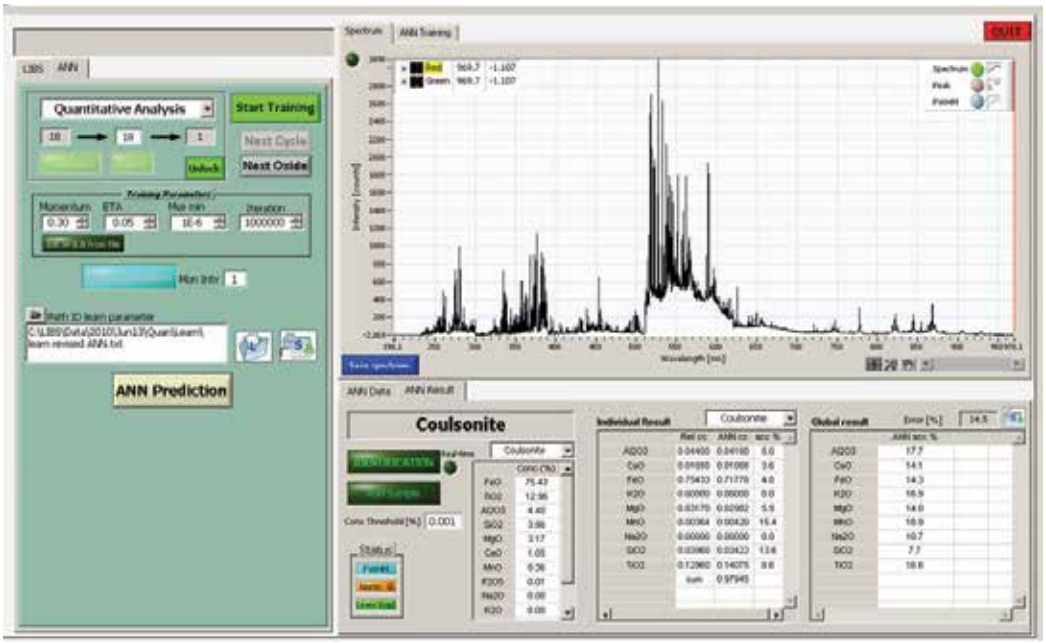


Fig. 17. Measurement of a new sample composition by quantitative ANN-LIBS algorithm implemented in LabView environment complemented by material identification ANN analysis. Upper-left section defines the network parameters and hardware control parameters. Top-right part displays the acquired spectrum. Bottom-right section displays the results of ANN analysis (from left to right): sample identity (Coulsonite in this case) and its tabulated composition, then the sample composition predicted by quantitative ANN, and finally the difference between the predicted composition and the tabulated composition.

Based on the above algorithms, the integrated software tool has been developed. It provides identification, mineralogy, and composition analysis with a single acquisition of LIBS spectra. The future works will be directed toward verification of stability of the algorithms with data acquired in different experimental settings. Use of sequential training for quantitative composition analysis is proposed to enhance this stability. We plan to implement comprehensive validation tests in laboratory and in field conditions.

5. Acknowledgements

The authors wish to thank the following scientists and engineers who contributed to success of this project: A. Dudelzak, J. Lucas, V. Motto-Ros, M. Sabsabi, D. Gratton, J. Spray and A. Hollinger.

6. References

Aguilera, J.A.; Aragón, C.; Cristoforetti, G. & Tognoni, E. (2009) Application of calibration-free laser-induced breakdown spectroscopy to radially resolved spectra from a copper-based alloy laser-induced plasma, *Spectrochimica Acta Part B*, Vol. 64, No. 7, (July 2009) pp. 685-689, ISSN: 05848547

Belkov, M.V.; Burakov, V.S.; De Giacomo, A.; Kiris, V.V.; Raikov, S.N. & Tarasenko, N.V. (2009) Comparison of two laser-induced breakdown spectroscopy techniques for total carbon measurement in soils, *Spectrochimica Acta Part B*, Vol. 64, No. 9, (September 2009) pp. 899-904, ISSN: 05848547

Bousquet, B.; Sirven, J.–B. & Canioni, L. (2007) Towards quantitative laser-induced breakdown spectroscopy analysis of soil samples, *Spectrochimica Acta Part B*, Vol. 62, No. 12, (December 2007) pp. 1582-1589, ISSN: 05848547

Cho, H.H.; Kim, Y.J.; Jo, Y.S.; Kitagawa, K.; Arai, N. & Lee, Y.I. (2001) Application of laser-induced breakdown spectrometry for direct determination of trace elements in starch-based flours, *Journal of Analytical Atomic Spectrometry*, Vol. 16, No. 6, (June 2001) pp. 622-627, ISSN: 02679477

Ciucci, A.; Corsi, M.; Palleschi, V.; Rastelli, S.; Salvetti, A. & Tognoni, E. (1999) New procedure for quantitative elemental analysis by laser-induced plasma spectroscopy, *Applied Spectroscopy*, Vol. 53, No. 8, (August 1999) pp. 960-964, ISSN: 00037028

Clegg, S.M.; Sklute, E.; Dyar, M.D.; Barefield, J.E. & Wien, R.C (2009) Multivariate analysis of remote laser-induced breakdown spectroscopy spectra using partial least squares principal, component analysis, and related techniques, *Spectrochimica Acta Part B*, Vol. 64, No. 1, (January 2009) pp. 79-88, ISSN: 05848547

Cremers, D.A. & Radziemski, L.J. (2006) *Handbook of Laser-Induced Breakdown Spectroscopy*, John Wiley & Sons, ISBN: 978-0-470-09299-6, USA

Eppler, A.S.; Cremers, D.A.; Hickmott, D.D.; Ferris, M.J. & Koskelo, A.C. (1996) Matrix effects in the detection of Pb and Ba in soils using laser-induced breakdown spectroscopy, *Applied Spectroscopy*, Vol. 50, No. 9, (September 1996) pp. 1175-1181, ISSN: 00037028

Escudero-Sanz, I.; Ahlers, B. & Courrèges-Lacoste, G.B. (2008) Optical design of a combined Raman–laser-induced-breakdown-spectroscopy instrument for the European Space Agency ExoMars Mission, *Optical Engineering*, Vol. 47. No. 3, (March 2008) pp. 033001-1 - 033001-11, ISSN: 00913286

Ferreira, E. C.; Milori, D.M.B.P.; Ferreira, E.J.; Da Silva, R.M. & Martin-Neto, L. (2008) Artificial neural network for Cu quantitative determination in soil using a portable laser induced breakdown spectroscopy system, *Spectrochimica Acta Part B*, Vol. 63., No. 10, (October 2008) pp. 1216-1220, ISSN: 05848547

Gaft, M.; Nagli, L.; Fasaki, I.; Kompitsas, M. & Wilsch, G. (2009) Laser-induced breakdown spectroscopy for on-line sulfur analyses of minerals in ambient conditions, *Spectrochimica Acta Part B*, Vol. 64, No. 10, (October 2009) pp. 1098-1104, ISSN: 05848547

Garrelie, F. & Catherinot, A. (1999) Monte Carlo simulation of the laser-induced plasma-plume expansion under vacuum and with a background gas, *Applied Surface Science*, Vol. 138-139, No. 1-4, (January 1999) pp. 97-101, ISSN: 01694332

Gurney K. (1997) *An Introduction to Neural Networks*, UCL Press, ISBN: 0-203-45151-1, UK

Harmon, R.S.; DeLucia, F.C.; McManus, C.E.; McMillan, N.J.; Jenkins, T.F.; Walsh, M.E. & Miziolek, A. (2006) Laser-induced breakdown spectroscopy – An emerging chemical sensor technology for real-time field-portable, geochemical, mineralogical, and environmental applications, *Applied Geochemistry*, Vol. 21, No. 5, (May 2006) pp. 730-747, ISSN: 08832927

Haykin. S. (1999) *Neural Networks: A Comprehensive Foundation*, Prentice Hall, ISBN: 0132733501, US

Iida, Y. (1990) Effects of atmosphere on laser vaporization and excitation processes of solid samples, *Spectrochimica Acta Part B*, Vol. 45, No. 12, (December 1990) pp. 1353-1367, ISSN: 05848547

Inakollu, P.; Philip, T.; Rai, A.K.; Yueh, F.-Y. & Singh, J.P. (2009) A comparative study of laser induced breakdown spectroscopy analysis for element concentrations in aluminum alloy using artificial neural networks and calibration methods, *Spectrochimica Acta Part B*, Vol. 64, No. 1, (January 2009) pp. 99-104, ISSN: 05848547

Kadirkamanathan, V. & Niranjan, M. (1993) A function estimation approach to sequential learning with neural networks, *Neural Computation*, Vol. 5, No. 6, (June 1993) pp. 954-975, ISSN 0899-7667

Koujelev, A.; Motto-Ros, V.; Gratton, D. & Dudelzak, A. (2009) Laser-induced breakdown spectroscopy as geological tool for field planetary analogue research, *Canadian Aeronautics and Space Journal*, Vol. 55, No. 2, (August 2009) pp. 97–106, ISSN: 00082821

Koujelev, A.; Sabsabi, M.; Motto-Ros, V.; Laville, S. & Lui, S.L. (2010) Laser-induced breakdown spectroscopy with artificial neural network processing for material identification, *Planetary and Space Science*, Vol. 58, No. 4, (April 2010) pp. 682-690, ISSN: 00320633

Lanza, N.; Wiens, R.C.; Clegg, S.M.; Ollila, A.M.; Humphries, S.D.; Newsom, H.E.; Barefield, J.E. & ChemCam Team (2010) Calibrating the ChemCam laser-induced breakdown spectroscopy instrument for carbonate minerals on Mars, *Applied Optics*, Vol. 49, No. 13, (May 2010) pp. C211-C217, ISSN: 00036935

Lui, S.L. & Cheung, N.H. (2003) Resonance-enhanced laser-induced plasma spectroscopy: ambient gas effects, *Spectrochimica Acta Part B*, Vol. 58, No. 9, (September 2003) pp. 1613-1623, ISSN: 05848547

Lui, S.L. & Koujelev, A.S. (2011) Accurate identification of geological samples using artificial neural network processing of laser-induced breakdown spectroscopy data, *Journal of Analytical Atomic Spectrometry,* (to be published)

Menut, D.; Descostes, M.; Meier, P.; Radwan, J.; Mauchien, P. & Poinssort, C. (2006) Europium migration in argillaceous rocks: on the use of micro laser-induced breakdown spectroscopy as a microanalysis tool, *Materials Research Society Symposium Proceedings*, Vol. 932, (September 2006) pp. 913-918, ISSN: 02729172

Miziolek, A.W.; Palleschi, V. & Schechter, I. (2006) Laser Induced Breakdown Spectroscopy, Cambridge University Press ISBN-13: 9780521852746, ISBN-10: 0521852749, UK

Mönch, I.; Sattmann, R. & Noll, R. (1997) High speed identification of polymers by laser-induced breakdown spectroscopy, *Proceedings of SPIE,* Vol. 3100, No. 1, (September 1997) pp. 64-74, ISSN: 0277786X

Moody, J.E. (1992) The effective number of parameters: an analysis of generalization and regularization in nonlinear learning systems, In: *Advances in neural information processing systems 4*, Moody, J.E.; Hanson, S.J. & Lippmann, R.P., (Eds.), pp. 847-854, Morgan Kaufmann Publishers, ISSN: 1-55860-222-4, USA

Motto-Ros, V.; Koujelev, A.S.; Osinski, G.R. & Dudelzak, A.E. (2008) Quantitative multi-elemental laser-induced breakdown spectroscopy using artificial neural networks, *Journal of the European Optical Society – Rapid Publications*, Vol. 3, (March 2008) 08011, ISSN: 19902573

Prechelt, L. (1998) Early stopping – but when?, In: *Neural Networks: Tricks of the trade*, Orr, G.B. & Müller, K.-R., (Eds.), pp. 55-69, Springer Verlag, ISBN-10: 3540653112, ISBN-13: 9783540653110, Heidelberg, USA

Rajasekaran, S.; Suresh, D. & Vijayalakshmi Pai, G.A. (2002) Application of sequential learning neural networks to civil engineering modeling problems, *Engineering with Computers*, Vol. 18, No. 2, (August 2002) pp. 138-147, ISSN: 01770667

Rajasekaran, S.; Thiruvenkatasamy, K. & Lee, T.L. (2006) Tidal level forecasting using functional and sequential learning neural networks, *Applied Mathematical Modeling*, Vol. 30, No. 1, (January 2006) pp. 85-103, ISSN: 0307904X

Ramil, A.; López, A.J. & Yáñez A. (2008) Application of artificial neural networks for the rapid classification of archaeological ceramics by means of laser induced breakdown spectroscopy (LIBS), *Applied Physics A*, Vol. 92, No. 1, (January 2008) pp. 197-202, ISSN: 09478396

Samek, O.; Telle, H.H. & Beddows, D.C.S. (2001) Laser-induced breakdown spectroscopy: a tool for real-time, in vitro and in vivo identification of carious teeth, *BMC Oral Health*, Vol. 1, PMC64785, ISSN: 14726831

Sattmann, R.; Mönch, I.; Krause, H.; Noll, R.; Couris, S.; Hatziapostolou, A.; Mavromanolakis, A.; Fotakis, C.; Larrauri, E. & Miguel, R. (1998) Laser-induced breakdown spectroscopy for polymer identification, *Applied Spectroscopy*, Vol. 52 No. 3, (March 1998) pp. 456-461, ISSN: 00037028

Sharma, S.K.; Misra, A.K.; Lucey, P.G.; Wiens, R.C. & Clegg, S.M. (2007) Combined remote LIBS and Raman spectroscopy at 8.6 m of sulfur-containing minerals, and minerals coated with hematite or covered with basaltic dust, *Spectrochimica Acta Part A*, Vol. 68, No. 4, (December 2007) pp. 1036-1045, ISSN: 13861425

Sirven, J.–B.; Bousquet, B.; Canioni, L.; Sarger, L.; Tellier, S.; Potin-Gautier, M. & Hecho, I. Le (2006) Qualitative and quantitative investigation of chromium-polluted soil by laser-induced breakdown spectroscopy combined with neural networks analysis, *Analytical and Bioanalytical Chemistry*, Vol. 385, No. 2, (May 2006) pp. 256-262, ISSN:16182642

Sirven, J.-B.; Sallé, B.; Mauchien, P.; Lacour, J.-L.; Maurice, S. & Manhès, G. (2007) Feasibility study of rock identification at the surface of Mars by remote laser-induced breakdown spectroscopy and three chemometric methods, *Journal of Analytical Atomic Spectrometry*, Vol. 22, No. 12, (December 2007) pp. 1471-1480, ISSN: 02679447

St-Onge, L.; Kwong, E.; Sabsabi, M. & Vadas, E.B. (2002). Quantitative analysis of pharmaceutical products by laser-induced breakdown spectroscopy. *Spectrochimica Acta Part B*, Vol. 57, No. 7, (July 2002) pp. 1131-1140, ISSN: 05848547

5

Application of Artificial Neural Networks in the Estimation of Mechanical Properties of Materials

Seyed Hosein Sadati, Javad Alizadeh Kaklar and Rahmatollah Ghajar
K. N. Toosi University of Technology
Iran

1. Introduction

In today's industry, it is imperative that a thorough knowledge of the mechanical properties of materials be known to the designer in order to come up with a design of parts, tools, or instruments that will meet the highly competitive industrial requirements. It is well known that mechanical properties of various materials are in turn highly affected by the manner in which they are subjected to loadings of both static and fatigue types, and by its manufacturing process, in particular the heat treatment the material receives during its manufacturing. This further makes it required to perform the proper experiments and laboratory tests with regard to fatigue in the field of fatigue mechanics in order to obtain the necessary knowledge for the material properties for design purposes. It is emphasized that such properties obtained from monotonic tests are of no value and by no means recommended. To this end, on one hand metallurgical engineers often attempt to obtain their desired material properties and efficiencies by making variations in the parameters governing the manufacturing process. On the other hand, yet, the high costs of fatigue tests as compared with those of the simple monotonic tests, as well as the need for complex testing equipment are the major drawbacks in the way of such tests, encouraging the use of approximate and empirical mathematical models based on the data obtained from the monotonic tests. This has been quite evident among researchers and industry alike, as indeed indicated by the variety of ongoing articles published in the field. In the area of materials engineering as well, the knowledge of the effect of different manufacturing processing parameters on the material properties in view of the highly expensive nature of the tests are also of particular interest. Use of Artificial Neural Network (ANN) models is considered as a less expensive, less tedious, more efficient, and highly reliable alternative means for the estimation of the material fatigue properties using the data obtained from the monotonic tests. In addition, the ANN methodology was also employed for the parameter estimation related to the manufacturing process of materials. The method was also used to investigate and infer the manner in which such material properties are affected by variations in the parameters that are the main governing elements of these properties. Many researchers have indeed pursued such applications in their studies (Bucar et al., 2006; Genel, 2004; Han, 1995; Lee et al., 1999; Liao et al., 2008; Malinov et al., 2001; Mathew et al., 2007; Mathur et al. 2007; Park & Kang, 2007; Pleune & Chopra, 2000; Srinivasan et al., 2003;

Venkatessh & Rack, 1999). Once the ANN model is trained properly, it will be able to offer an appropriate estimate of the required output using the given input parameters.
In this chapter, it is first attempted to give an account of the necessity and benefits of the ANN methodology as pertained to the mechanical properties of materials followed by an exposition of the necessary knowledge for the proper use of this strong and valuable technique. This chapter will then close by the introduction and discussion of a case study.

2. Artificial Neural Network; an overview

In recent years, Artificial Neural Network (ANN) has been applied in many fields including function approximation and prediction. Artificial neural network is a kind of information processing technology, good at handling problems in which complex nonlinear relations exist among the input and output variables. The main idea of neural network approach resembles the human brain functioning. Artificial neural networks are based on the structure and functioning of the biological nervous system. Neurons are the basic unit or building blocks of the brain. The human brain consists of about 10^{11} neurons, leading in about 1000 trillion connections. A neuron receives many input signals but it produces only one output signal at a time.
Back propagation network is made up of a large number of interconnected neurons. The neurons are arranged in layers: one input layer, one output layer, and one or more hidden layer(s) between the input layer and the output layer. Each neuron in the input layer is connected to every neuron in the hidden layer which in turn is connected to the neuron in the output layer. This topology results in a network commonly known as the Multilayer Perceptron, abbreviated as MLP. In the conventional MLP network, there is no connection between neurons in the same layer. The connection between two neurons is called synapse, and each synapse has an associated strength or weight, which influences the output of the neuron. Neurons in the input layer receive the input signals from each training pattern. The outputs of the neurons in the input layer are exactly the same as the input signals to those neurons. The neurons in the hidden layer then receive the output of the input neurons. This signal is then run through a nonlinear activation function to produce the output of each neuron of the hidden layer. The output of the neurons of the last hidden layer is in turn sent as an input to each output neuron. The more the number of hidden neurons, the more complex the model becomes. The predicted output is compared with the desired output and the error is sent back to the hidden layer for improving the prediction. The neural network architecture is described by the number of hidden layers, the number of neurons in each layer, the form of activation function used to nonlinearise the input-output relationship, training algorithms, the learning rate, momentum rate, and other pertinent parameters used in the network.
Implementation of a neural network requires one to make three main decisions, namely the structure, i.e., the network topology, the type of activation functions, and the learning algorithm. The structure of the network deals with the number of hidden layers used in the network as well as the number of nodes used in each layer. The activation function refers to the transfer function for the neurons of each layer except for the input layer which uses an identity activation function. The notion of learning refers to the use of a suitable learning algorithm in the network training process.
Before training, the network architecture must be defined. As a general rule, the number of neurons must be large enough to be able to map the implicit relationship existing between

the input and output variables for a given problem. On the other hand, it must not be overly large, since otherwise its connection weights may not be accurately estimated from the available training data.

Depending on the structure of a neural network designed for use in a certain problem, two general neural networks can be designed, namely feedback and feedforward neural networks. The most widely used algorithms are in general feedforward networks, which is simple from the viewpoint of structure and easily analyzed mathematically. The back propagation neural network scheme, with a strong learning ability in training and mapping the relations between inputs and outputs, is the most commonly used network model (Koker et al., 2007).

The ANN model applied in materials science belongs to a novel branch subject-Material Metrics. ANNs are parallel-interconnected networks of simple computational elements that are intended to interact with the objects of the real-world in a similar way to the biological nervous system (Muc & Gurba, 2001). Formally, an ANN is an oriented graph in which the nodes represent a set of processing units, called neurons, or processing elements, and the connections represent the information flow channels. Each connection between two neurons has an associated value called weight (Wij) which specifies the strength of the connection from unit i to unit j (Bahrami et al., 2005). The choice of a specific class of networks for the simulation of a nonlinear and complex problem depends on a variety of factors such as the accuracy desired and the prior information concerning the input–output pairs (Mousavi Anijdan et al., 2005). The most popular ANN in materials science and engineering investigations is the feedforward multi-layer perceptron, where the neurons are arranged into an input layer, one or more hidden layers, and an output layer (Muc & Gurba, 2001; Bahrami et al., 2005; Mousavi Anijdan et al., 2005; Song et al., 1995). A schematic description of a three-layer feedforward network is given in Fig. 1. Assuming that the network consists of n, p, and m neurons in the input, hidden, and output layers, respectively, the net input (z_j) to node j in the hidden layer is of the form

$$z_j = \sum_{i=1}^{n} W_{ij} x_i + b_j \quad , \quad j = 1,2,...,p \tag{1}$$

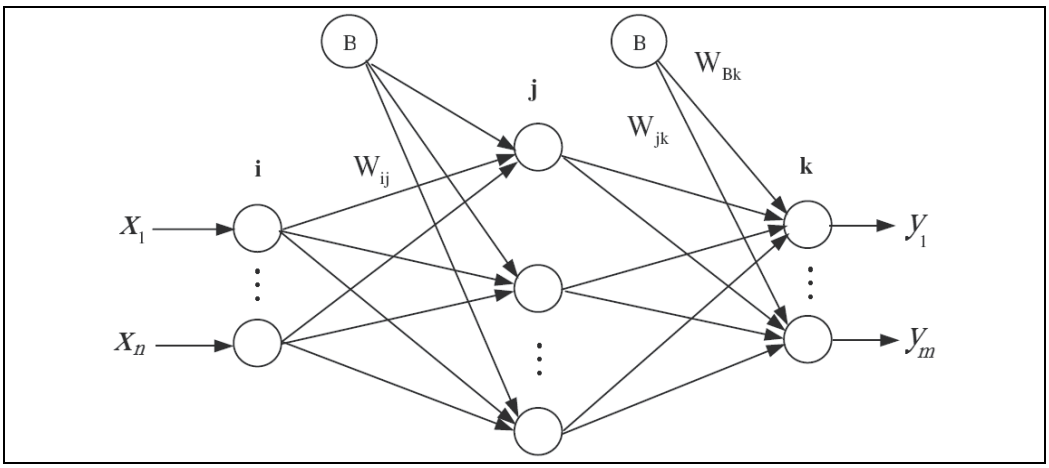


Fig. 1. A schematic description of artificial neural network configuration

, where x_i is the input of node j of the input layer, W_{ij} is the connection weight associated with node i of the input layer and node j of the hidden layer, and b_j is the bias associated with node j of the hidden layer. The bias neurons do not take any input and they emit a constant output value across weighted connections to the neurons in the next layer.
Each neuron consists of a transfer function expressing internal activation level. The output (h) from a neuron is determined by transforming its input using a suitable transfer function as follows:

$$h_j = f\left(\sum_{i=1}^{n} W_{ij} x_i + b_j \right) \quad , \quad j = 1,2,...,p \tag{2}$$

Generally, the transfer functions are sigmoid function, hyperbolic tangent and linear function (Fogel, 1994; Wong & Y. W. Wong, 1995).
In the output layer, the net input z_k to node k is of the form:

$$z_k = \sum_{j=1}^{p} W_{jk} h_j + b_k \quad , \quad k = 1,2,...,m \tag{3}$$

The output y_k of node k of the output layer is then written as:

$$y_k = g\left(\sum_{j=1}^{p} W_{jk} h_j + b_k \right) \quad , \quad k = 1,2,...,m \tag{4}$$

To estimate the degree of accuracy of the network, the database is split into two sub-groups: the so-called learning phase used to determine the weights associated with each interconnection (training process), and the test or validation phase, which verifies that the network is able to predict examples not previously learnt (validation process). The training process consists of determining the weights that produce from the inputs the best fit of the predicted outputs over the entire training data set. An input vector is then introduced in the input layer and is propagated through the network all through the output layer. The difference between the computed output vector and the target vector is then used to determine the weights using an optimization procedure in order to minimize the suitable error function. This form of training is termed the back propagation training algorithm (Mousavi Anijdan, 2005). There are several variables that have an effect on the ANN training. These variables are the number of training data points (N), network size (number of hidden layer and neurons in each layer), and number of training iterations or epochs (C). To find the best set of these variables and parameters, these parameters should be varied and then the best combinations chosen.
An important step in building a neural network model is called training, the process of fitting the network to the experimental data and this is a computationally intensive process. Learning in an MLP neural network model involves the use of a gradient descent algorithm in an iterative manner to minimize the mean square error between the actual outputs of the network and the desired outputs in response to given inputs. Training in an MLP network is performed by a forward followed by a backward operation. The network produces its actual output for a certain input pattern using the current connection weights. Subsequently, the backward operation is carried out to modify the weights in an attempt to decrease the error between the actual and desired outputs. The updates in the weights are affected by two

parameters, namely the learning rate η and momentum coefficient α. The learning rate defines the speed for the changes that are taking place in the connection weights. The momentum coefficient is introduced to improve the learning process to make the learning process faster by adding a term to the weight adjustment proportional to the previous weight change.
The error is computed using Eqs. (5) known as average squared error. Here, N denotes the total number of samples in training set:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(t_i - a_i)^2 \tag{5}$$

Once each input pattern is presented to the neural network and the network error corresponding to that input pattern is calculated, this error is propagated back through the network and the weights are adjusted according to the well-known backpropagation of error learning rule. The next input pattern is then presented to the network and the process repeats until all the input patterns get the chance to be presented to the network. When all the input patterns are presented to the network once, one epoch is said to be completed in the training process. This process is iterated for as many epochs as needed for the error to reach a desired minimum level, at which time, the network is said to have learned the problem in consideration. Finally, the test data are used to verify the non-linear relationship between the input and output data sets.
NN models can be used for accurate interpolation within the range of input variables used for building the model. As far as extrapolation beyond the experimental range used in the training set, one must be careful not to place overly trust in the network response, as it may not produce a reliable result for inputs too far beyond.

2.1 Neural network training algorithms

Artificial neural networks are used as an interdisciplinary tool in many types of nonlinear problems. In order to design a neural network for a certain problem, one needs a training algorithm. As neural networks function based on samples (patterns), it is necessary to prepare a set of examples representing the problem in the forms of system inputs and outputs. During the training process, the weights and biases in the network are adjusted to minimize the error to obtain a high-performance for the problem.
There are various training algorithms used in neural network applications. It is normally not a simple task to predict which of these training algorithms will be the most appropriate one for any problem. It generally depends on a number of factors such as the network architecture, and the application at hand (pattern recognition, classification, or function approximation). Other issues that may also be important could be the data structure and uniformity of the training set, as these will affect the system accuracy and performance.
During the training process, it is important to avoid overtraining in an effort to obtain the best fit. This is a potential problem with the use of powerful non-linear regression methods in neural network modeling. An over-trained model tends to remember the relationship between input and output variables and therefore lacks generalization capability (Mathew et al., 2008).
During the training session, the network weights are continuously adjusted until the difference between the predicted output and experimental value is minimized, i.e. the error function defined as the sum of squares of the difference between predicted and

experimental value on all the input patterns reaches a set limit or the number of predetermined training operations or epochs are completed, whichever comes first. Levenburg-Marquardt algorithm, Quasi-Newton algorithm, and Steepest-Descent method are some of the optimization techniques employed in the training of the neural networks. From these learning algorithms, the one most used in training the MLP networks is the Levenburg-Marquardt algorithm due to its fast convergence. Accordingly, a brief account of this algorithms is presented below.

The back propagation learning algorithm which is based on the first order gradient of the network error enjoys the benefit of simple implementation as well as the ease of its use. Yet, it suffers from the disadvantage of a slow convergence. The Levenberg-Marquardt algorithm (Hagan & Menhaj, 1994) is a step taken towards solving the problem of slow convergence of the BP method. The LM method is based on a second order gradient of the network error. The fast convergence is the immediate consequence of the fact that the algorithm takes advantage of making use of only an approximation for the Hessian matrix instead of doing a thorough computation of this matrix. It also avoids the singularity of that matrix by adding a small term in the approximate calculation of the matrix. It is well known that the LM algorithm performs much faster than the usual BP rule at the cost of requiring more memory. The Levenberg–Marquardt training algorithm was indeed found to be the fastest training algorithm to date.

Once the training of the network is completed, the ability of the trained neural network to correctly generalize must be checked out using some input-output data not included in the training set. This set is commonly known as the test set or validation set. This set is normally prepared by randomly taking some 20 to 25 percent of the original data set. It is noted that each pattern from the validation set must lie within the range defined by the entire training set.

3. Some remarks in the use of Neural Networks

To develop a neural network with good performance, an adequate quantity of experimental data must be available. During the training and testing sessions, the network architecture, learning algorithm, and other parameters of the neural network should also be optimised to the specific problem under investigation. When the neural network is sufficiently optimal, and trained based on these data, it then becomes possible to generate satisfactory results when presented with any new input pattern it has never experienced before.

As the number of neurons in the hidden layer increases, so does the number of connections and weights to be fitted. The number of neurons and the number of hidden layers cannot be increased without limit because one may reach a situation where the number of the connections to be fitted is larger than the number of the data pairs available for training. Though the neural network can still be trained, the case is mathematically undetermined. Mathematically it is not possible to determine more fitting parameters than the available data points. For example, two data points are required as a minimum for linear regression, three data points for second order polynomial (parabolic) regression and so on. In practice, for a reliable regression, much more data than the minimum amounts are used to increase statistical significance. For example, if we use two points to determine a slope through linear regression, the standard error of the slope calculated will be infinitely large. A slope determined through two points has no statistical significance (Sha & Edwards, 2007).

In order to increase the efficiency of the neural network training techniques, it is necessary to use a large database consisting of sufficient number of training patterns to cover the

entire working space of the problem under study. Neural network modelling cannot replace experimentation; in fact it relies entirely on past experimental results, and because of the fact that experiments can be expensive, time consuming, and dangerous, researchers focus on the neural network methods. The neural network technique is actually most effective when large amount of data are already available in the literature. In such cases, the neural network is an effective way of storing and analyzing, with some artificial intelligence, the large amount and wide range of data from different sources. Because of the randomness of the data, the large size of the data, and the multiple natures of the input variables in most cases, it is usually difficult to extract physical rules governing the large data set or quantitative physical theories that can describe the problem data. Hence, a neural network model is ideal for such situations. In fact, if a limited amount of systematic data is available, efforts would be better spent on developing physical models rather than statistical or neural network models.

It is noted that the errors obtained for the testing sets are more important than the errors obtained for the training sets, due to the fact that the errors for the testing sets are normally much larger and would not compare well with the errors for the training sets. One must also note that it is more effective to develop separate models for individual output variables. This is because the training time increases significantly when the number and nature of outputs increases. Therefore, setting up a series of neural network models, with each model dealing with only one output variable significantly improves the network learning and simplifies and speeds up the training of the neural network model.

When determining the inputs, it is imperative to use independent variables as inputs. In fact, if there was only one input variable, a neural network would be no better than the conventional, simple regression. A neural network in this case does no more or better than the "Moving Average" or similar "Trend Line" functions, or even "Smoothed Line" in Microsoft Excel. A neural network is best used to model complicated interactions between several numbers of input parameters (Sha & Edwards, 2007).

All through experiences in the use of neural networks in engineering problems, it is determined that certain types of networks are normally more appropriate for certain problems. For instance, and to be more specific, it is realized that to predict the fatigue life of materials or to predict the mechanical behaviour of materials under monotonic and cyclic loading, the Multilayer Neural Network along with the Back-Propagation learning algorithms have been proven to be very effective and accurate, especially when there is sufficient training data (Abdalla & Hawileh; 2010).

4. Case study: Estimation of cyclic strain hardening exponent and cyclic strength coefficient of steels by artificial neural networks (Ghajar et al., 2008)

In many field test situations, it may be desirable to convert the measured strains to stress in order to estimate fatigue life. Stress-strain response of some steels can change significantly when subjected to inelastic strains as this may occur at notch roots due to cyclic loading. When fatigue failure occurs, particularly at low cycle fatigue, such inelastic straining is generally present. Hence, the cyclic stress-strain curve may better represent the steel's stress-strain response than the monotonic stress-strain curve (Society of Automotive Engineers [SAE], 2000). The relationship between cyclic strain amplitude, $\Delta\varepsilon/2$, and cyclic stress amplitude, $\Delta\sigma/2$, can be expressed as (Stephens et al., 2001):

$$\Delta\varepsilon/2 = \Delta\sigma/2E+(\Delta\sigma/2K')^{1/n'} \quad (6)$$

, where K' is the cyclic strength coefficient, n' is the cyclic strain hardening exponent, and E is the modulus of elasticity. The two fatigue properties needed in this correlation are K' and n'.
The cyclic strength coefficient, K', and the cyclic strain hardening exponent, n', are often determined from the cyclic stress plastic strain curve. A family of stabilized hysteresis loops at different strain amplitudes can be used to obtain the cyclic stress-strain curve for a given material. The tips from the family of multiple loops are connected, as shown in Fig. 2, to form the cyclic stress-strain curve. Three methods commonly used to obtain the cyclic stress-strain curve are the companion, incremental-step, and multiple-step test methods (Stephens et al., 2001). These test methods are time-consuming and the testing equipment is more complicated and expensive than that required for monotonic tension tests, while monotonic stress-strain properties are commonly available in handbooks. Therefore, it is more desirable to use approximation methods for estimating the values of K' and n'.

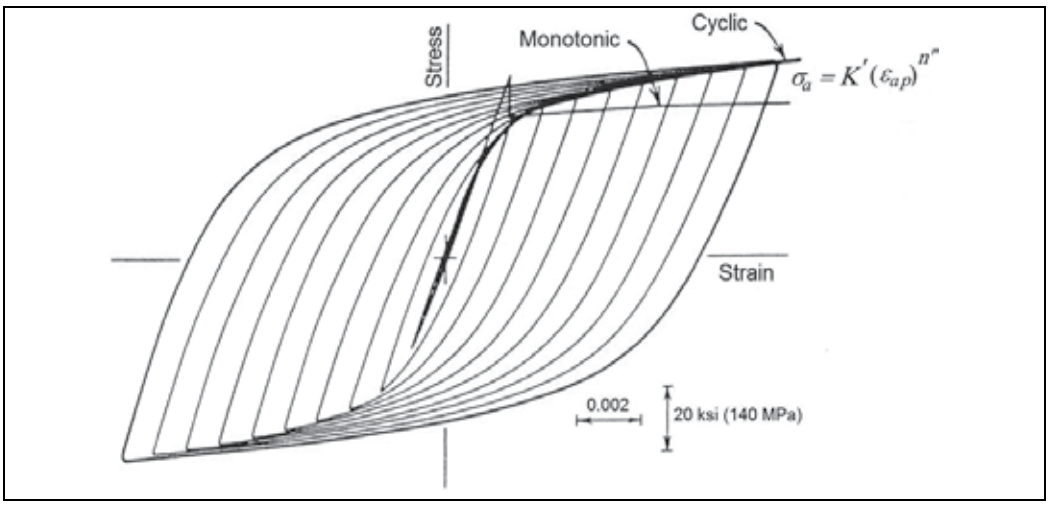


Fig. 2. Stable hysteresis loops for determining the cyclic stress-strain curve and comparison with the monotonic stress-strain curve (Stephens et al., 2001).

An approximation of K' and n' can also be calculated from the low-cycle fatigue properties by using (Stephens et al., 2001):

$$K' = \sigma'_f/(\varepsilon'_f)^{b/c}$$
$$n' = b/c \quad (7)$$

, where σ'_f is the fatigue strength coefficient, ε'_f is the fatigue ductility coefficient, b is the fatigue strength exponent, and c is the fatigue ductility exponent. This estimation method has its problems and errors. It requires, in the first place, the four empirical constants that must be obtained from fatigue tests. Furthermore, estimating cyclic stress-strain curves based on fatigue properties could lead to considerable errors in certain situations (Kim et al., 2002). So, it is recommended that the values of K' and n' obtained from direct fitting of the experimental data are used in fatigue design rather than those calculated from Eq. 7 (Stephens et al., 2001).

It is therefore useful to estimate cyclic strength coefficient and cyclic strain hardening exponent on the basis of monotonic tensile tests properties, reported in handbooks or simply obtainable from experiments. By doing so, one can covert cyclic strain to cyclic stress only by using tensile test properties.

In this work, the MLP network with back propagation algorithms is used for the estimation of cyclic strain hardening exponent, n', and the cyclic strength coefficient, K', of steels. The K' and n' are estimated by two separate networks. For these estimations steels tensile data used as input to the ANN model, are extracted from the literature (SAE, 2001; Kim et al., 2002; Roessle & Fatemi, 2000). In order to enhance training performance, both input and output variables are normalized before the network is trained. In order to investigate the influence of input parameters on the estimation of n' and K', several networks having different combinations of tensile data are considered. The mean square error (MSE) between the desired output and the ANN response is used.

A broad range of 82 steels is used for modeling n'. One set of data consist of 60 values was used for training the network and another consisting of 22 values was used for testing the trained network. Preliminary examinations were performed on different combinations of σ_y, S_u, RA%, BHN and E as input data to the ANNs in order to determine the parameters affecting the n' estimation. Finally three combinations of tensile data were selected from among them as follows: (σ_y, S_u and BHN), (σ_y, S_u, RA% and BHN), and (σ_y, S_u , RA%, BHN and E).

A number of neural network architectures with different number of neurons in the hidden layer (2 to 10 neurons) were also investigated to select the best one. A summary of the results is presented in Table 1. The results indicate that the best architecture involves 7 neurons for the first combination (σ_y , S_u and BHN), 9 neurons for the second (σ_y , S_u , RA% and BHN), and 6 neurons for the last (σ_y , S_u , RA% , BHN and E).

Input sets	Neurons in the hidden layers	Regression for the training data	Regression for the test data	No. Of Iterations
σ_y, S_u, BHN	6	0.890	0.563	363
	7	0.913	0.716	367
$\sigma_y, S_u, BHN, RA\%$	7	0.973	0.726	550
	8	0.967	0.792	120
	9	0.973	0.865	112
$\sigma_y, S_u, BHN, RA\%, E$	5	0.913	0.379	700
	6	0.962	0.702	700
	7	0.910	0.205	1000

Table 1. Networks details and architectures of n'

As mentioned earlier, the performance of the networks was evaluated by calculating MSE errors. In order to assess the validity of the networks and their accuracy, it is often useful to perform regression analysis between the network response and the corresponding target. Obviously, the closer these two data are, the better the performance of the network is. Fig. 3 shows the regression analysis for the best set of input for the test and training data. The regression results of the training data illustrate that networks were trained with a high accuracy. Furthermore, comparison of the regression results of the test data indicates that the set of inputs (σ_y , S_u , RA% and BHN) provided the best prediction, R=0.866, followed by

the set (σ_y , S_u and BHN). The difference in accuracy observed among the different input sets shows the importance of input parameters for predicting n'. It may be concluded that σ_y , S_u , RA% and BHN have relatively established effects on the prediction of n' while the effect of E is not only immaterial, but also confusing.

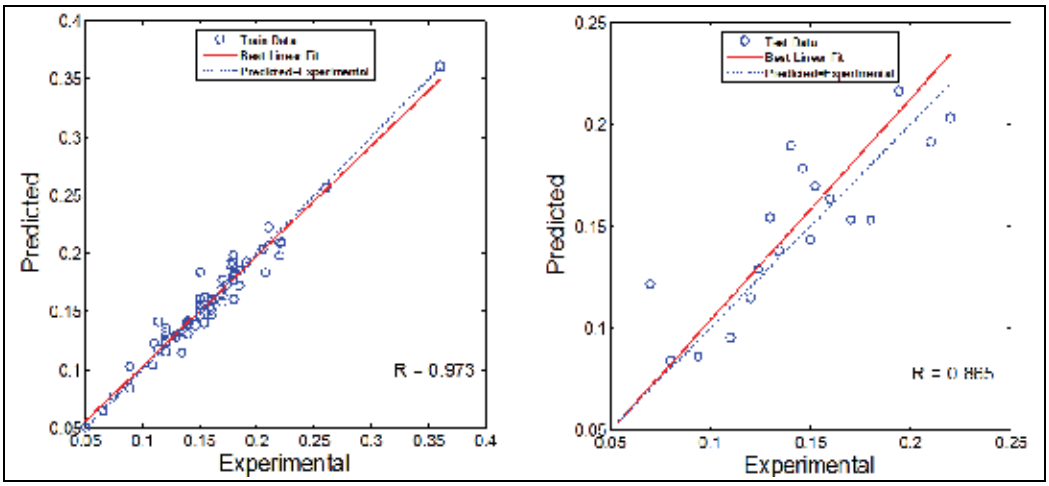


Fig. 3. Regression analysis of n' for the train and test data and (σ_y , S_u , RA% and BHN) as the ANN input.

In addition, the test data were used for a new prediction based on Eq. 7 to evaluate ANN test results. Fig. 4 shows the results of this estimation. By comparing ANN and Eq. 7 results (Fig. 3 and 4) it may be concluded that the ANN estimations were more accurate than Eq.7. Therefore, such estimations seem desirable, especially considering the time and effort that are required to obtain the fatigue properties used in the approximations by Eq.7, as compared with the monotonic tensile properties used in ANN predictions.

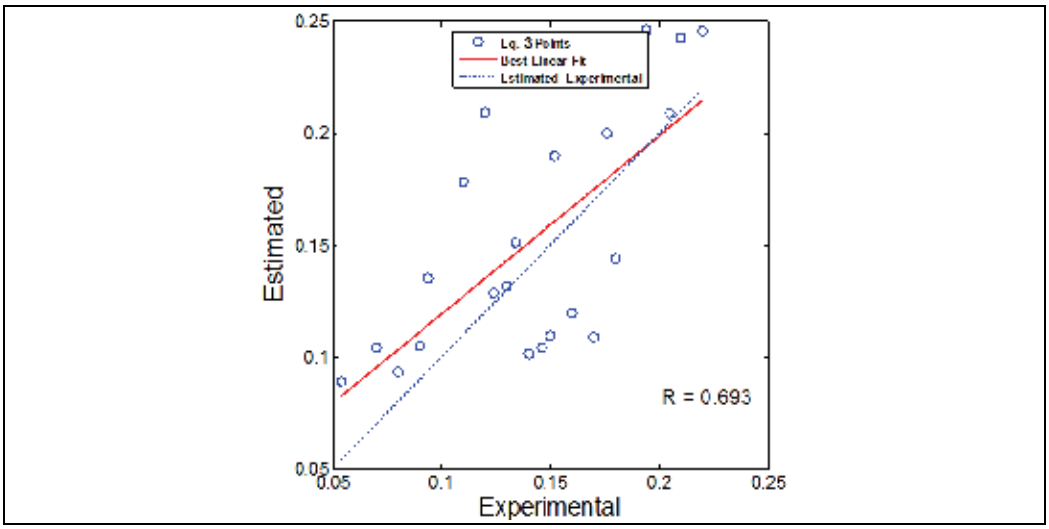


Fig. 4. Regression analysis of approximated n' based on Eq. 7 for the test data.

Finally, based on the regression analysis results, it is possible to claim that the ANN with (σ_y, S_u, RA% and BHN) input set is a useful method for the prediction of cyclic strain hardening exponent.

For K' estimation, the properties of 48 steels reported in the literature (SAE, 2001; Kim et al., 2002; Roessle & Fatemi, 2000) were used. 36 values of data were used for training the network and the others consisting of 12 data values were used for testing the trained network.

Three combinations of tensile data consisting of σ_y, S_u, RA%, BHN and E were used to determine the parameters affecting the K' estimation in the same manner as in the case of n'.

A number of neural network architectures with different number of neurons in the hidden layer (2 to 10 neurons) were also investigated to select the best one. The summary of the results are provided in Table 2. Clearly, the best architecture is associated with 6 neurons for the combination (σ_y, S_u and BHN), 8 neurons for (σ_y, S_u, RA% and BHN), and 7 neuron for (σ_y, S_u, RA% , BHN and E).

Input sets	Neurons in the hidden layers	Regression for the training data	Regression for the test data	No. Of Iterations
σ_y, S_u, BHN	6	0.998	0.901	320
	7	0.995	0.896	200
$\sigma_y, S_u, BHN, RA\%$	6	0.999	0.926	229
	7	0.994	0.931	87
	8	0.996	0.953	100
$\sigma_y, S_u, BHN, RA\%, E$	6	0.999	0.913	166
	7	0.999	0.925	185
	8	0.999	0.853	200

Table 2. Networks details and architectures of K'

As mentioned previously, the performance of the networks was evaluated by calculating the MSE errors. In order to assess the validity of the networks and their accuracies, the

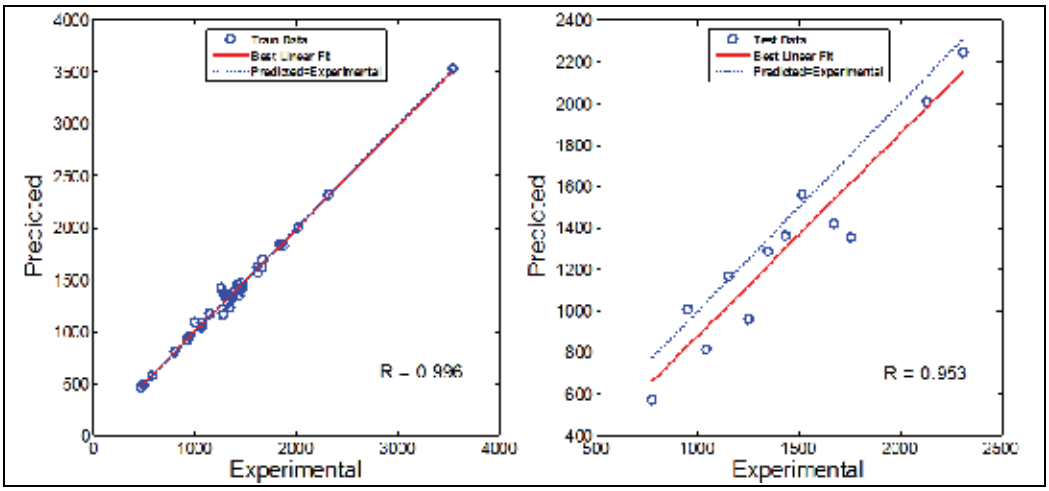


Fig. 5. Regression analysis of K' for the train and test data and (σ_y , S_u , RA% and BHN) as ANN input.

regression analysis was performed between the network response and the corresponding target. Fig. 5 indicates the regression analysis for the best set of input for the test and training data. It may be seen from this figure that the value of K' obtained from the trained network is in close agreement with its experimental value. Moreover, the regression results of test data illustrate that amongst the input sets, the set (σ_y , S_u , RA% and BHN) set of inputs yielded the best prediction, R=0.953, followed by the set (σ_y, S_u, RA%, BHN and E). Similar to the case of n' estimation, it can be concluded that σ_y, S_u, RA% and BHN have relatively effects on the prediction of K' while the effect of E is not only immaterial, but also confusing.
Additionally, the test data were used for a new prediction based on Eq. 7. Comparing the results of this estimation and experimental values for K' is depicted in Fig. 6. There is a poor agreement between the experimental values of K' and the predictions obtained from Eq. 7. From Fig. 5 and 6, it can be concluded that the ANN estimations are more accurate than Eq. 7. Therefore, the ANN method is preferred, especially by considering that it only requires monotonic tensile properties.
Finally, similar to n', based on the regression analysis results, it is possible to claim that the ANN with (σ_y, S_u, RA% and BHN) input set is a useful method for the prediction of cyclic strength coefficient.
Cyclic strain hardening exponent and cyclic strength coefficient of steels, which characterize the stable curves of true stress amplitude versus true plastic strain amplitude, were predicted by ANN with high accuracy of 0.865 and 0.953% respectively while accuracy of estimations based on approximate relations (Eq. 7) are 0.693 and 0.726%.
It was concluded that predicted stable cyclic true stress-strain curve properties by trained neural network are more accurate compared to approximate relations based on low-cycle fatigue properties.

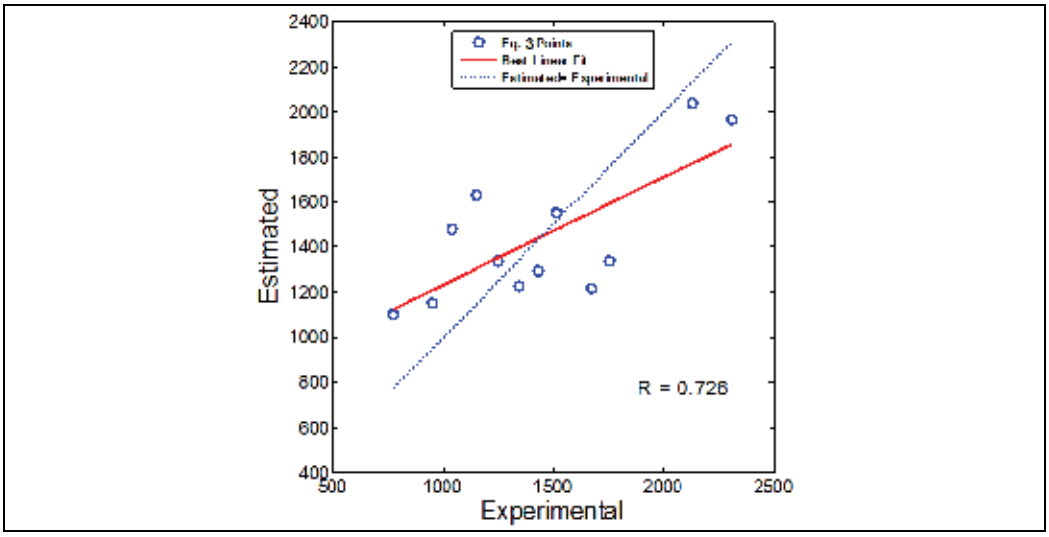


Fig. 6. Regression analysis of of approximated K' based on Eq. 7 for test data.

5. Conclusion

This chapter presents an exposition of the benefits and advantages of the neural network technique in the solution of engineering problems as a whole and materials science

problems in particular. A general overview of the neural network models is given followed by the introduction of a case study related to some fatigue properties of steels. It is emphasized that neural network models are effective techniques for modelling the problems in material science as the technique will help a material scientist with the determination and estimation of the complex and often nonlinear relationship governing the input/output data obtained within an experimental setup. As such, neural network techniques are still an ongoing research area as applied to the problems in material science and engineering.

6. References

Abdalla, J. A., & Hawileh, Rami., (in press). Modeling and simulation of low-cycle fatigue life of steel reinforcing bars using artifiial neural network. *Journal of the Franklin Institute,* ISSN 0016-0032

Bahrami, A., Mousavi Anijdan, S. H., & Ekrami, A., (2005). Prediction of Mechanical Properties of DP Steels Using Neural Network Model. *Journal of Alloys and Compounds*, Vol.392, No.1-2, (April 2005), pp. 177-182, ISSN 0925-8388

Bucar, T., Nagode, M., & Fajdiga, M., (2006). A Neural Network Approach to Describing the Scatter of S–N Curves. *International Journal of Fatigue*, Vol.28, No.4, (April 2006), pp. 311-323, ISSN 0142-1123

Fogel, D. B., (1994). An Introduction to Simulated Evolutionary Optimization. *IEEE Transactions on Neural Networks*, Vol.5, No.1, (1994), pp. 3-14, ISSN 1045-9227

Genel, K., (2004). Application of Artificial Neural Network for Predicting Strain-Life Fatigue Properties of Steels on the Basis of Tensile Data. *International Journal of Fatigue*, Vol.26, No.10, (October 2004), pp. 1027-1035, ISSN 0142-1123

Ghajar, R.; Alizadeh, J., & Naserifar, N., (2008). Estimation of cyclic strain hardening exponent and cyclic strength coefficient of steels by artificial neural networks, *Proceedings of ASME 2008 International Mechanical Engineering Congress and Exposition*, pp. 639-648, ISBN 978-0-7918-4873-9, Boston, Massachusetts, USA, November 2-6, 2008.

Hagan, M. T., & Menhaj, M. B., (1994). Training Feedforward Networks with the Marquardt Algorithm. *IEEE Transactions on Neural Networks*, Vol.5, No.6, (1994), pp. 989-993, ISSN 1045-9227

Han, Y. L., (1995). Artificial Neural Network Technology as a Method to Evaluate the Fatigue Life of Weldments with Welding Defects. *International Journal of Pressure Vessels & Piping*, Vol.63, No.2, (1995), pp. 205–209, ISSN 0308-0161

Kim, K. S., Chen, X., Han, C., & Lee, H. W., (2002). Estimation Methods for Fatigue Properties of Steels under Axial and Torsional Loading. *International Journal of Fatigue*, Vol.24, No.7, (July 2002), pp. 783-793, ISSN 0142-1123

Koker, R., Altinkok, N., & Demir, A., (2007). Neural network based prediction of mechanical properties of particulate reinforced metal matrix composites using various training algorithms. *Materials and Design,* Vol.28, No.2, (2007), pp. 616-627, ISSN 0264-1275

Lee, J. A., Almond, D. P., & Harris, B., (1999). The Use of Neural Networks for the Prediction of Fatigue Lives of Composite Materials. *Composites: PartA: Applied Scienceand Manufacturing*, Vol.30, No.10, (October 1999), pp. 1159–1169, ISSN 1359-835X

Liao, X., Xu, W., & Gao, Z., (2008). Application of Artificial Neural Network to Forecast the Tensile Fatigue Life of Carbon Material. *Key Engineering Materials*, Vol.385-387, (July 2008), pp. 385-387, ISSN 1662-9795

Malinov, S., Sha, W., & McKeown, J. J., (2001). Modelling the Correlation Between Processing Parameters and Properties in Titanium Alloys Using Artificial Neural Network. *Computational Materials Science*, Vol.21, No.3, (July 2001), pp. 375-394, ISSN 0927-0256

Mathew, M. D., Kim, D. W., & Ryu, W. S., (2008). A neural network model to predict low cycle fatigue life of nitrogen-alloyed 316L stainless steel. *Materials Science and Engineering A,* Vol.474, No.1-2, (February 2008), pp. 247-253, ISSN 0921-5093

Mathur, S., Gopeand, P. C., & Sharma, J. K., (2007). Prediction of Fatigue Lives of Composites Material by Artificial Neural Network, *Proceedings of the SEM2007 Annual Conference and Exposition*, Paper 260, Springfield, Massachusetts, USA, June4–6, 2007

Mousavi Anijdan, S. H., Bahrami, A., & Mater, J., (2005). A New Method in Prediction of TCP Phases Formation in Superalloys. *Materials Science and Engineering A*, Vol.396, No.1-2, (April 2005), pp. 138-142, ISSN 0921-5093

Muc, A., & Gurba, W., (2001). Genetic Algorithms and Finite Element Analysis in Optimization of Composite Structures. *Composite Structures*, Vol.54, No.2-3, (November-December 2001), pp. 275-281, ISSN 0263-8223

Park, J. M., & Kang, H. T., (2007). Prediction of Fatigue Life for Spot Welds Using Back-Propagation Neural Networks. *Materials and Design*, Vol.28, No.10, (2007), pp. 2577–2584, ISSN 0261-3069

Pleune, T. T., & Chopra, O. K., (2000). Using Artificial Neural Networks to Predict the Fatigue Life of Carbon and Low-Alloy Steels. *Nuclear Engineering and Design*, Vol.197, No.1-2, (April 2000), pp. 1–12, ISSN 0029-5493

Roessle, M. L., & Fatemi, A., (2000). Strain-Controlled Fatigue Properties of Steels and Some Simple Approximations. *International Journal of Fatigue*, Vol.22, No.6, (July 2000), pp. 495-511, ISSN 0142-1123

SAE Standards (2002). Technical Report on Low Cycle Fatigue Properties: Ferrous and Nonferrous Materials, *SAE*, Report Number: J1099, Warren dale, PA

Sha, W., & Edwards, K. L., (2007). The use of artificial neural networks in materials science based research. *Materials and Design,* Vol.28, No.6, (2007), pp. 1747-1752, ISSN 0261-3069

Song, R. G., Zhang, Q. Z., Tseng, M. K., & Zhang, B. J., (1995). The Application of Artificial Neural Networks to the Investigation of Aging Dynamics in 7175 Aluminium Alloys. *Materials Science and Engineering C*, Vol.3, No.1, (October 1995), pp. 39-41, ISSN 0928-4931

Srinivasan, V. S., Valsan, M., Roa, K. B. S., Mannan, S. L., & Raj, B., (2003). Low Cycle Fatigue and Creep–Fatigue Interaction Behavior of 316L(N) Stainless Steel and Life Prediction by Artificial Neural Network Approach. *International Journal of Fatigue*, Vol.28, No.12, (December 2003), pp. 1327-1338, ISSN 0142-1123

Stephens, R. I., Fatemi, A., Stephens, R. R., & Fuchs, H. O., (2001). *Metal Fatigue in Engineering*, John Wiley & Sons, ISBN 9780471510598, Canada

Venkatessh, V., & Rack, H. J., (1999). A Neural Network Approach to Elevated Temperature Creep–Fatigue Life Prediction. *International Journal of Fatigue*, Vol.21, No.3, (March 1999), pp. 225-234, ISSN 0142-1123

Wong, K. P., & Wong, Y. W., (1995). Thermal Generator Scheduling Using Hybrid Genetic/ Simulated-Annealing Approach. *IEEE Proceedings on Generation, Transmission and Distribution*, Vol.142, No.4, (July 1995), pp. 372-380, ISSN 1350-2360

6

Optimum Design and Application of Nano-Micro-Composite Ceramic Tool and Die Materials with Improved Back Propagation Neural Network

Chonghai Xu[1,2], Jingjie Zhang[1] and Mingdong Yi[1,2]
[1]*Shandong Institute of Light Industry*
[2]*Shandong University*
PR China

1. Introduction

At present, the main method of the ceramic tool and die materials research is still the traditional 'trial-error' method which needs a large number of experiments to determine the optimum material compositions. This traditional method requires researchers to repeat experiments and to face to the complex preparation processes as well as the high cost of the experiments, and so on. Therefore, the utilization of advanced and even intelligent design technologies for ceramic material design is extremely necessary.

The computational intelligence (CI) technique, as an offshoot of artificial intelligence (AI), is a kind of heuristic algorithm including three categories: neural network, fuzzy system and evolutionary computation. Genetic algorithm (GA) and artificial neural network (ANN) are the two important computational intelligence techniques.

In recent, the two techniques especially the ANN have got successful application in the material design of ceramics and metal matrix composites, etc. For instance, some researchers used ANN to predict the functional properties of ceramic materials from compositions (Scott et al, 2007) or the bending strength and hardness of particulate reinforced Al-Si-Mg aluminum matrix composites (Altinkok & Korker, 2004) or the mechanical properties of ceramic tool (Huang et al, 2002) or the percentage of alumina in Al_2O_3/SiC ceramic cakes and the pore volume fraction (Altinkok & Korker, 2005), etc.

ANN is a kind of self-learning technology and back propagation (BP) neural network is one of the simply and commonly used network architectures. BP is based on the gradient descent method where connection weights and thresholds are modified in a direction corresponding to the negative gradient of a backward-propagated error measure (Jiang & Adeli, 2004). Although BP neural network has an advantage of high accuracy, it is often plagued by the local minimum point, low convergence or oscillation effects. In order to overcome the disadvantage of BP neural network, GA is usually used to improve the BP neural network. GA has a strong searching capability and high probability in finding the global optimum solution which is suitable for the early stage of data searching. Although these two techniques seem quite different in the number of involved individuals and the process scheme, they can provide more power of problem solving than either alone (Yen &

Lu, 2002; Yao, 1999; Gen & Cheng, 2000). Therefore, many researchers have attempted to use GA to improve BP neural network in order to achieve the complementary advantages (Sexton, 1998; Gupta & Sexton, 1999).

Some successful examples of the improved BP neural network which were optimized by GA had been reported to optimize successfully the flow stress of 304 stainless steel under cold and warm compression (Anijdan et al, 2007) or the surface roughness in end milling Inconel 718 (Ozcelik et al, 2005) or the plasma processes (Kim & Bae, 2005), etc. In literature (Zemin et al, 2010), BP neural network was used to predict punch radius based on the results of air-bending experiments of sheet metal. This prediction model was proved to be effective by experiments.

The compositions and hot pressing parameters are two important factors which can greatly affect the mechanical properties of ceramic materials. In the present study, the standard BP neural network and the improved BP neural network are used in the optimum design of both compositions and hot pressing parameters of $ZrO_2/TiB_2/Al_2O_3$ nano-micro-composite ceramic tool and die material.

2. The improved BP Neural Network

BP neural network is multi-layered forward feed neural network which is based on the error back-propagation algorithm. And the study of BP neural network can be divided into two steps which named forward-propagation process and back-propagation process, respectively. In forward-propagating process, the input is the known sample data and the information will be transmitted in turn for the hidden layer and the output layer. And the error between actual output and expected output is calculated in output layer. The back-propagation process is that the calculated error will modify each connection weight and threshold along the original way. The above two processes are iterated and repeated until the error satisfies the condition.

Fig. 1 is the structure of BP neural network. The network is multilayer which is composed of some connection neurons according to certain rules. It mainly consists of input layer, hidden layer and output layer, and each layer has independent neuron constitution. The layers are connected by the weights which can express the link degree between the neurons. And the hidden layer is composed of at least one or more layers.

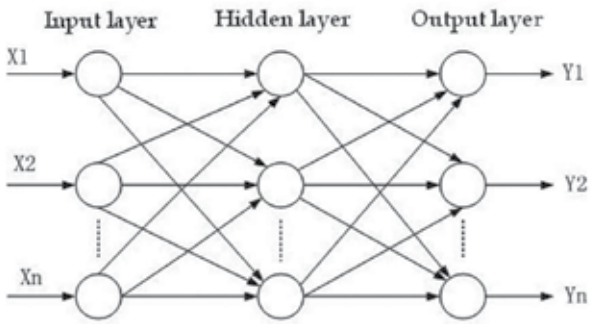


Fig. 1. The structure of BP neural network

The improved BP neural network means using GA to optimize the BP neural network. The commonly improved BP neural network mainly has three methods. One is using GA to improve the structure of BP neural network which is marked as GA-BP I; the second is using

GA to identify the initial connection weight and threshold of BP neural network which is marked as GA-BP II; while the third is using GA not only to identify the initial connection weight and threshold but also to improve the structure of BP neural network which is marked as GA-BP III. The latter two kinds of algorithms will further be discussed in the present study.

2.1 The GA-BP II algorithm

BP neural network is very sensitive to the initial vectors and different initial values may lead to completely different results. Especially in the specific calculation process, the related initial values are usually determined randomly or by experience. Once the initial value is not properly determined, it would lead to effect of oscillation or seldom convergence. Even if it is convergent, the process will be quite slow because of the too long time of training or falling into local minimum. And the best connection weights distribution can not be achieved. Used GA to optimize the connection weight and threshold of BP neural network (GA-BP II) can solve the kind of problem.

The principle of the GA-BP II algorithm is as follows: using GA to optimize the connection weights and thresholds of BP neural network from its searching space which contains all the available individuals. Then, the BP network is trained with these connection weights and thresholds so that the error between BP actual output and target output could be reduced.

2.2 The algorithm of GA-BP III

Most of the research literatures focused on the utilization of various improved GA to optimize the connection weight and threshold ignoring the importance of the structure of BP neural network and its close relationship between the structure and the connection weight. In the present study, an improved algorithm of BP neural network with GA (GA-BP III) is used for the optimum design of nano-micro-composite ceramic tool and die materials. In this algorithm, GA is used to fully optimize BP neural network including the comprehensive optimization of the structure, the initial connection weight and the threshold.

It is reported that the structure of BP neural network could greatly affect the network processing capabilities. Redundant nodes and connections are not allowed existing in a good structure. However, the design of the structure of BP neural network had not rigorously and systematically theoretical guidance and remains largely dependent on a person's experience. Using GA to solve the optimization problem of the structure can be transformed into the process of biological evolution that can be obtained through the selection, crossover and mutation, etc.

According to the Kolmogorov theorem, for three-layer BP neural network, it can achieve any given mapping. When the number of the hidden layer neurons is enough, it can use any degree of accuracy to approximate any non-linear mapping. The neurons in the input layer and output layer are determined on the specific problem; only the number of neurons in the hidden layer is variable. Thus, how to determine the number of the hidden layer neurons has become a very important issue which is the optimum object of the structure of BP neural network. If the number of neurons in the hidden layer is too little, the network may not be trained satisfyingly with the results, or the network is not robust enough with the poor fault-tolerance. If too many, they will make learning time too long and the error is not necessarily the smallest. So there exist an optimal number of the hidden layer neurons.

It is assumed that the BP neural network is hierarchically fully connected and only the neurons of two adjacent layers are possible to be connected and must be connected. If the

input and output vector values are in the real number space and there are no effects between the connected two neurons, the weight of the two connected neurons will be zero. Under the known condition of the input and output neurons, the number of the neurons in the hidden layer could only correspond to the number of the connection weight.
Thus, the principle of the GA-BP III algorithm is as following: Before the optimization, GA is used to optimize the number of connection weight, the best connection weight and threshold for BP neural network from its searching space which contains all the available individuals. After that, a global optimum solution can be achieved. Then the last generation of individuals is decoded and the corresponding structure of BP neural network, initial connection weights and thresholds can be achieved. With these values work as the structure and the initial value, samples are then trained to obtain the precise optimization. The optimum structure of BP neural network and these connection weights and thresholds could reduce the error between the output of BP neural network and the target output. Therefore, the results became more accurate.

2.2.1 Encoding

For the BP neural network with *n-d-m* three-layer where *n* is the number of neurons of the input layer, *d* is the number of neurons of the hidden layer and *m* is the number of neurons of the output layer, the floating-point type number is used for the connection weight and threshold to be encoded. Link the encoding which is encoded by the order of first connection weights then thresholds to a long string. The length of the string *L* is:

$$L=n\times d+d+d\times m+m \tag{1}$$

The scope of *d* can be ascertained by the empirical formula of the hidden layer neurons (Zhu & Shi, 2006) given below:

$$d=\sqrt{n+m}+\alpha \tag{2}$$

Where *n* and *m* can be determined based on the actual problem, α is a constant in the range of 1 to 10. Thus, once the length of the string *L* is determined, the number of hidden layer neurons and then the network structure of BP neural network can be determined. The individual value after decoding is the corresponding connection weight and threshold.

2.2.2 Determination of the fitness function

The relationship between the input and output of the network is available as following (Gu et al, 2006):

$$Y_k=\sum_{j=1}^{d}V_{jk}\cdot f\left[\sum_{i=1}^{n}W_{ij}\cdot X_i+\theta_j\right]+r_k \tag{3}$$

where *f* is the transfer function between layers, X_i is the actual input of the neuron *i* of the input layer, W_{ij} is the connection weight from the neuron *i* of the input layer to the neuron *j* of the hidden layer, θ_j is the threshold of the neuron *j* of the hidden layer, V_{jk} is the connection weight from the neuron *j* of the hidden layer to the neuron *k* of the output layer, r_k is the threshold of the neuron *k* of the output layer, and Y_k is the actual output of the

neuron *k* of the output layer. According to the error between the actual output and the target output, a least-squares error function *E* can be defined as (Gu et al, 2006):

$$E(W,V,\theta,r)=\frac{1}{2p}\sum_{q=1}^{p}\sum_{i=1}^{m}(T_i^q-Y_i^q)^2 \tag{4}$$

Where p is the total number of the training samples, T_i^q and Y_i^q is the target output and the actual output of the neuron i of the input layer, respectively when the q^{th} training sample trains.

In order to integrate GA and BP, the fitness function of GA is selected as following (Gu et al, 2006):

$$f(W,V,\theta,r)=\frac{1}{E(W,V,\theta,r)+1} \tag{5}$$

In this way, once the outputs are available through the BP computation, the relating outputs are transferred to the fitness function for comparing and determining the final value. While the fitness values are being updated from generation to generation, a new generation of the population will be produced and do the same evaluation. When fitness of the population reaches the maximum, the output error of the network will become the minimum. This process will continue until the end of predetermined generation.

3. Experimental

$ZrO_2/TiB_2/Al_2O_3$ nano-micro-composite ceramic tool and die material is a typical three phase composite material in which zirconia is the matrix reinforced with titanium diboride and alumina. High purity nanometer sized ZrO_2 and micrometer sized TiB_2 and Al_2O_3 powders were used as the starting materials with average sizes of 39nm, 1.5μm and 1.0μm, respectively. According to the required volume fraction, the raw material powders were blended. The mixtures were subsequently homogenized with absolute alcohol media and Polyethylene Glycol (PEG) in a ball mill for 48h. After milling the slurry was dried in vacuum and screened.

In the experiment of compositions optimization, the samples were then formed by vacuum hot pressing (HP) technique under the hot pressing temperature of 1445°C, pressure of 30MPa and time duration of 60min. Sintered bodies were cut with a diamond wheel into samples of 3mm×4mm×30mm. The flexural strength was measured in an electronic universal testing machine (model INSTRON-5569) by means of the three-point bending method with a span of 20mm and a loading rate of 0.5mm/min. The Vickers hardness was tested by the testing machine (model Hv-120) with a load of 196N and a holding time of 15s. The fracture toughness was determined by the indentation method. The experimental data for the compositions optimization are listed in Table 1.

In the optimization process of hot pressing parameters, the pressure was kept as 35MPa limited by the hot pressing equipment. The sintering temperature was initially selected from 1420 to 1480°C and the holding time was initially selected in the range of 20-80min. All the selected hot pressing parameters are shown in Table 2. According to the processing technologies mentioned above, the materials were prepared and the mechanical properties were tested.

Number	V_{ZrO2} (%)	V_{TiB2} (%)	V_{Al2O3} (%)	Hardness (GPa)	Flexural strength (MPa)	Fracture toughness (MPa·m$^{1/2}$)
1	90	5	5	10.03	619	9.76
2	85	5	10	10.20	501	10.59
3	80	5	15	10.36	509	9.95
4	85	10	5	10.37	617	10.51
5	80	10	10	10.71	612	11.37
6	75	10	15	10.19	565	12.20
7	80	15	5	9.82	513	7.86
8	75	15	10	10.22	524	7.91
9	70	15	15	10.14	520	8.11

Table 1. The compositions and mechanical properties of $ZrO_2/TiB_2/Al_2O_3$ ceramic material

Number	Sintering temperature (°C)	Holding time (min)	Hardness (GPa)	Flexural strength (MPa)	Fracture toughness (MPa·m$^{1/2}$)
1	1430	60	13.59	1055	10.57
2	1440	60	13.78	1010	10.26
3	1450	60	13.48	878	9.54
4	1460	60	13.15	914	9.74
5	1470	60	13.26	835	9.27
6	1450	20	13.23	569	8.68
7	1450	40	12.93	671	9.91
8	1450	80	13.69	785	9.49

Table 2. The hot pressing parameters and mechanical properties of $ZrO_2/TiB_2/Al_2O_3$ ceramic material

4. The compositions optimization

4.1 The compositions optimization based on the standard BP algorithm

The BP neural network can achieve the nonlinear relationship between the compositions and the mechanical properties. If there are sufficient training data, proper change of the structure of the BP neural network which includes the number of neurons in input layer, hidden layer and output layer, and the number of the hidden layer, the BP neural network model of the optimal compositions can be established. Material compositions can then be optimized through the complex non-linear relationship between the compositions of the materials preparation and the mechanical properties. In this paper, the training sample data of standard BP neural network are the experimental data of the compositions optimization (Table 1).

The hardness, flexural strength and fracture toughness are the main mechanical properties of ceramic tool and die materials. When the processing techniques are determined, the mechanical properties of ceramic tool and die material are mainly decided by the compositions. Therefore, the inputs of the BP neural network model are the contents of each composition and the outputs are the three mechanical properties of the given materials. Therefore the model has three input neurons and three output neurons. The sigmoid-type function is adopted for the input layer to the hidden layer as the transfer function and

linear-type function is adopted for the hidden layer to the output layer. And the simulated data are listed in Table 3.

Number	V_{ZrO2} (%)	V_{TiB2} (%)	V_{Al2O3} (%)	Number	V_{ZrO2} (%)	V_{TiB2} (%)	V_{Al2O3} (%)
1	85	6	9	12	80	14	6
2	85	7	8	13	75	11	14
3	85	8	7	14	75	12	13
4	85	9	6	15	75	13	12
5	80	6	14	16	75	14	11
6	80	7	13	17	60	10	30
7	80	8	12	18	60	15	25
8	80	9	11	19	60	20	20
9	80	11	9	20	60	25	15
10	80	12	8	21	60	30	10
11	80	13	7				

Table 3. The simulated data in compositions optimization

According to the theory of the BP neural network, the computing process is programmed with neural network toolbox in MATLAB. Training function is using 'trainlm' function and network performance parameters are using MSE function which is the mean square error between the expected output value and the actual output value to measure the network performance. The training parameters are set as following:

net.trainparam.show=10

net.train.param.goal=0.001

net.trainParam.epochs=100

net.trainParam.lr=0.01

Other parameters are set by default.

Through the calculation of the error between the actual output value and the expected output value, and according to the BP neural network model, the number of hidden layer neurons is initially chosen as 6. So, the final structure of standard BP neural network is 3×6×3. Based on this BP model, the compositions are optimized and the mechanical properties are then predicted. The predicted mechanical properties are listed in Table 4. After 62 times of iterations, the training curve of BP neural network is converged to the specified accuracy of 0.001 (Fig. 2). And the mean square error MSE is 1.24.

According to the predicted results, the best flexural strength is 643MPa and the best hardness of the materials is 9.94GPa with the corresponding volume fractions of 85vol%ZrO_2, 8vol%TiB_2 and 7vol%Al_2O_3, and the corresponding fracture toughness is 11.14MPa·$m^{1/2}$. The highest fracture toughness is 11.76MPa·$m^{1/2}$ with the corresponding volume fractions of 75vol%ZrO_2, 14vol%TiB_2 and 11vol%Al_2O_3, but the corresponding hardness and flexural strength is low. From comprehensive consideration, it seems that the mechanical properties of $ZrO_2/TiB_2/Al_2O_3$ nano-micro-composite ceramic tool and die material with the corresponding volume fractions of 85vol%ZrO_2, 8vol%TiB_2 and 7vol%Al_2O_3 is the best. So, this composition is the optimum composition in prediction.

Number	V_{ZrO2} (%)	V_{TiB2} (%)	V_{Al2O3} (%)	Hardness (GPa)	Flexural strength (MPa)	Fracture toughness (MPa $m^{1/2}$)
1	85	6	9	9.22	546	10.91
2	85	7	8	9.06	611	11.05
3	85	8	7	9.94	643	11.14
4	85	9	6	9.89	643	10.38
5	80	6	14	9.89	506	11.27
6	80	7	13	9.88	510	11.11
7	80	8	12	9.15	543	11.11
8	80	9	11	9.87	594	9.33
9	80	11	9	9.89	594	7.36
10	80	12	8	9.00	565	6.87
11	80	13	7	9.72	547	7.24
12	80	14	6	9.75	530	11.54
13	75	11	14	9.04	568	9.68
14	75	12	13	9.06	542	8.39
15	75	13	12	9.88	528	7.95
16	75	14	11	9.28	525	11.76
17	60	10	30	9.24	451	5.91
18	60	15	25	9.81	504	6.28
19	60	20	20	9.11	576	9.77
20	60	25	15	9.12	483	10.97
21	60	30	10	9.46	454	11.05

Table 4. The predicted results of standard BP algorithm in compositions optimization

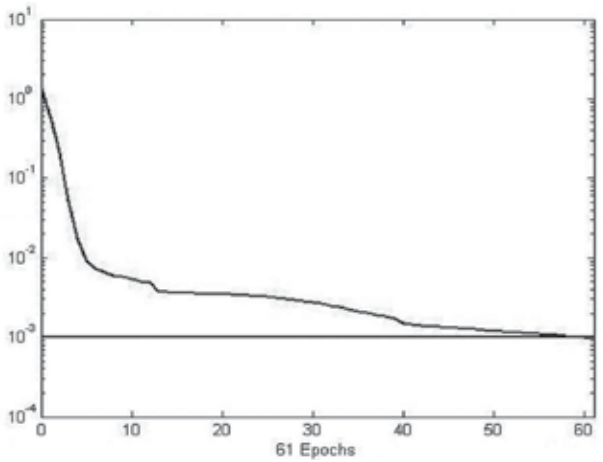


Fig. 2. The training curve of BP neural network of standard BP algorithm

4.2 The compositions optimization based on GA-BP II algorithm

According to the formerly established BP model in which the number of the neurons of hidden layer is 6 and the structure of the BP model is 3×6×3, GA-BP II algorithm is used to optimize the compositions and the predicted mechanical properties are listed in Table 5.

Number	V_{ZrO2} (%)	V_{TiB2} (%)	V_{Al2O3} (%)	Hardness (GPa)	Flexural strength (MPa)	Fracture toughness (MPa·m$^{1/2}$)
1	85	6	9	10.29	563	10.49
2	85	7	8	10.36	625	10.16
3	85	8	7	10.43	645	10.07
4	85	9	6	10.36	636	10.25
5	80	6	14	10.35	496	10.88
6	80	7	13	10.38	505	11.72
7	80	8	12	10.29	558	11.73
8	80	9	11	10.23	599	11.51
9	80	11	9	10.24	617	11.10
10	80	12	8	10.22	614	10.53
11	80	13	7	10.25	585	9.54
12	80	14	6	10.10	541	8.49
13	75	11	14	10.25	595	11.85
14	75	12	13	10.26	590	11.14
15	75	13	12	10.26	565	9.87
16	75	14	11	10.25	538	8.61
17	60	10	30	10.12	511	9.94
18	60	15	25	9.92	458	10.49
19	60	20	20	9.97	517	10.16
20	60	25	15	9.63	516	10.07
21	60	30	10	9.12	462	10.25

Table 5. The predicted results of GA-BP II algorithm in compositions optimization

After about 100 generations of searching, the fitness and square error have been stabilized respectively as shown in Fig.3. After 12 times of iterations, the training curve of BP neural network of GA-BP II algorithm is converged to the specified precision of 0.001 which is shown in Fig.4. The mean square error MSE is 1.05 and the elapsed-time is 144.20s.

According to the predicted results in Table 5, the maximum flexural strength and hardness of the materials is 645MPa and 10.43GPa,respectively, when the volume fractions of ZrO_2, TiB_2 and Al_2O_3 is 85vol%, 8vol% and 7vol%respectively while the fracture toughness is 10.07MPa·m$^{1/2}$ which is only the better one. The maximum fracture toughness of the material is 11.85MPa·m$^{1/2}$ with the corresponding volume fractions of 70vol%ZrO_2, 11vol%TiB_2 and 14vol%Al_2O_3, while the corresponding flexural strength and hardness is only 595MPa and 10.25GPa, respectively. Compared with the two compositions, the mechanical properties of the material with the volume fractions of 85vol%ZrO_2, 8vol%TiB_2 and 7vol%Al_2O_3 is the better.

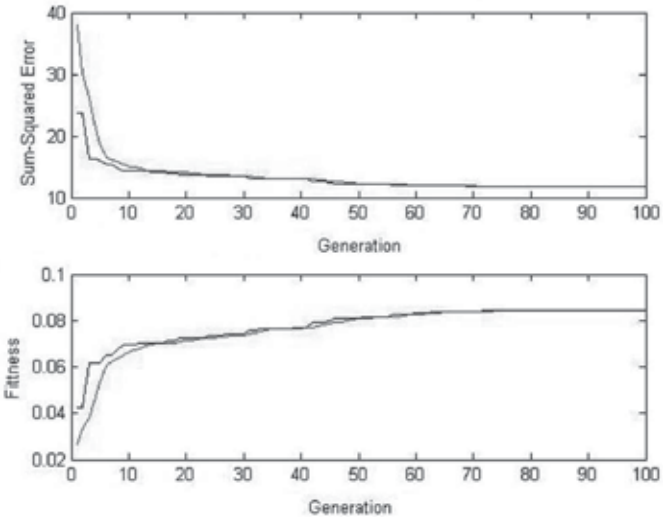


Fig. 3. The curve of square error and fitness of GA-BP II in compositions optimization

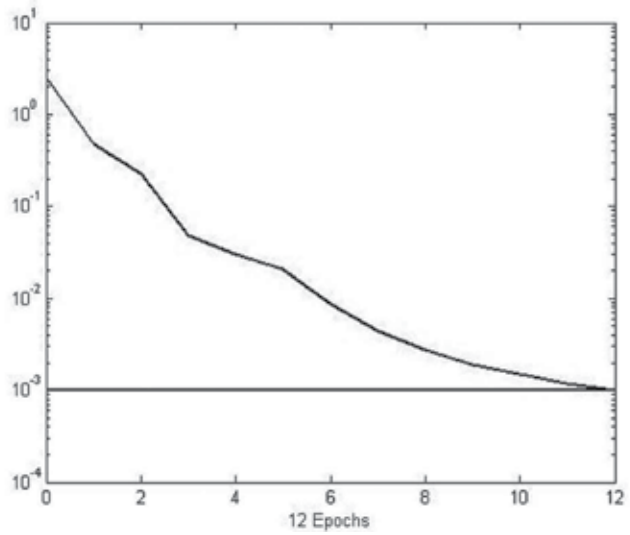


Fig. 4. The training curve of BP neural network of GA-BP II algorithm in compositions optimization

4.3 The compositions optimization based on GA-BP III algorithm

According to the compositions optimization, the input layer neuron number is 3, the output layer neuron number is 3, and the number of hidden layer neurons is set to d. According to GA-BP III algorithm, the string length L can be determined as $L=3+7d$. In accordance with the empirical formula (Eq. 2) which can determine the range of hidden layer neurons, the range of d is 4-13. According to the principle of GA-BP III algorithm, the corresponding computing process is programmed and run with MATLAB 7.0 software. The corresponding parameters are set as following: the initial population number N=30, the cross probability

P_c=0.8, the mutation probability P_m=0.1 and the error e=0.001. When the error reaches the intended target, the training process of BP neural network is then stopped.

In the process of GA optimization, with the increase of the evolution of generation, the fitness and square error are becoming convergent and finally achieve the best value. At this stage, the corresponding connection weights and thresholds of the BP neural network become the optimum. Their individuals are decoded as follows: -0.33, 1.00, 0.00, -0.64, -0.09, 0.18, -0.61, -0.38, 0.13, -0.27, -0.27, 0.91, -0.55, 0.72, 0.57, 0.33, -0.48, 0.36, -0.51, -0.19, -0.19, -0.05, 0.13, -0.32, -0.52, 0.24, -0.78, 0.29, 0.39, 0.13, -0.46, 0.00, 0.00, 0.47, 1.00, -0.32, -0.59, 0.36, -0.07, -0.40, -0.34, -0.28, -0.22, -1.00, -0.28, -0.61, 0.19, 0.49, -0.82, 0.00, 0.10, 0.52, 0.63, -0.48, 0.96, -0.89, 0.23, 0.11, -0.59. Based on the above 59 parameters and $L=3+7d$, the number of hidden layer neurons is ascertained as 8. Therefore the structure of BP neural network is 3×8×3 and the last 11 parameters are the threshold values. Some connection weights in the list above are found to be 0.00 which indicate that the connection between the two neighboring neurons is invalid.

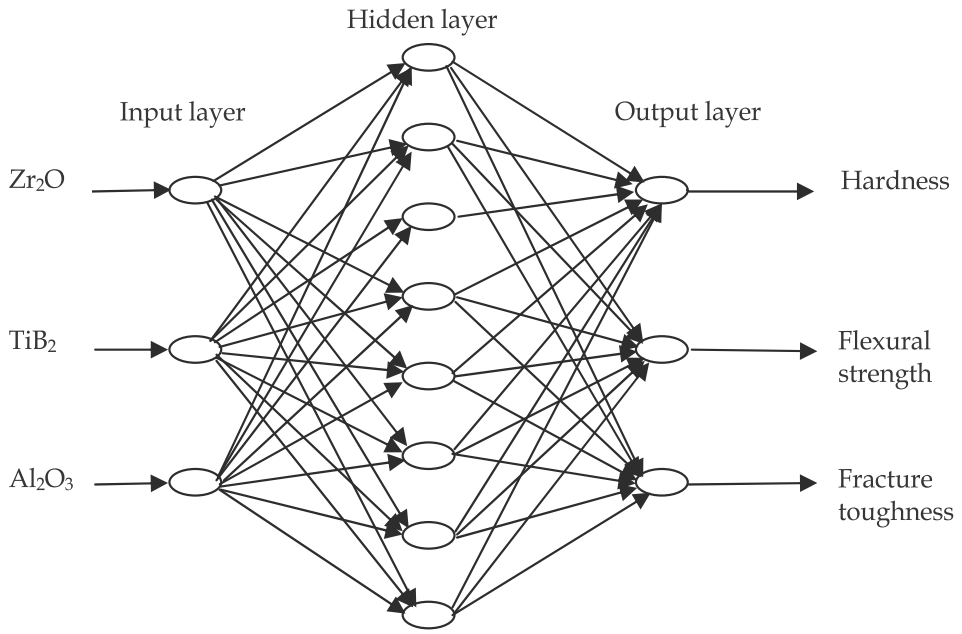


Fig. 5. The structure of BP neural network of GA-BP III algorithm in compositions optimization

The concrete structure of BP neural network is the improved BP neural network optimized by GA which is shown in Fig.5. It can be seen that the first neuron of input layer and the third neuron of hidden layer is no connection. The third neuron of hidden layer and the second and the third neurons of output layer are also connectionless. The data within the range of the experimental results are selected as the data for prediction in order to get the optimum compositions corresponding to the best mechanical properties.

After about 100 generations of searching, the fitness and the square error have been stabilized respectively as shown in Fig.6. The curve of BP training target is shown in Fig.7. It

indicates that the BP neural network has 8 iterations convergence to the specified accuracy. The elapsed-time is 129.939s and MSE is 0.1491.

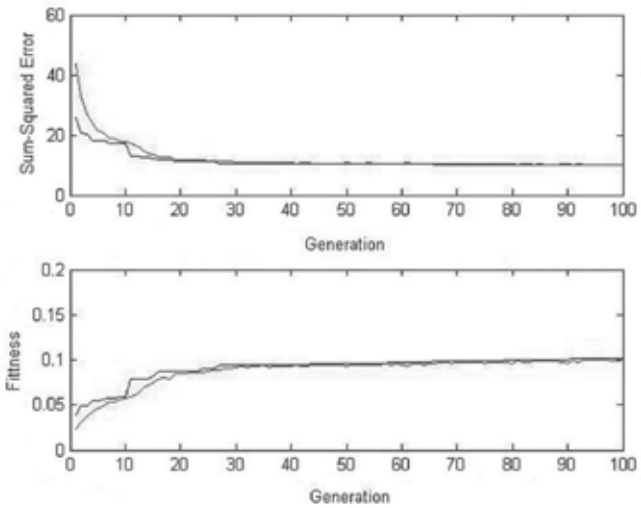


Fig. 6. The curve of square error and fitness of GA-BP III algorithm in compositions optimization

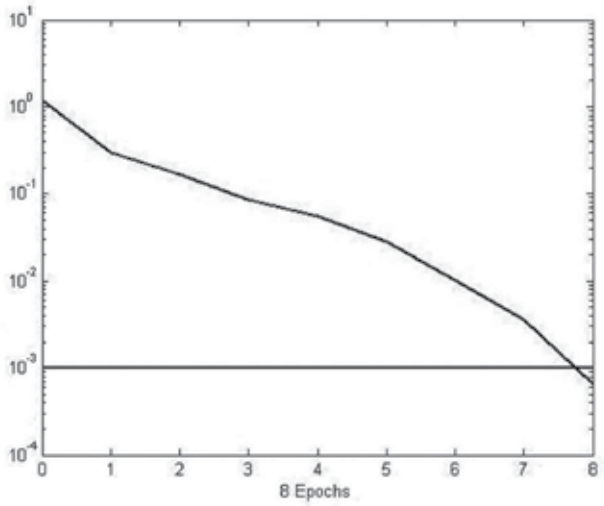


Fig. 7. The training curve of BP neural network of GA-BP III algorithm in compositions optimization

The predicted results of GA-BP III algorithm are given in Table 6. It indicates that the highest flexural strength is 685MPa and the highest hardness is 10.74GPa with the corresponding volume fractions of 85vol%ZrO_2, 8vol%TiB_2 and 7vol%Al_2O_3. The fracture toughness with the same compositions is 10.38MPa.$m^{1/2}$ which is slightly less than the best value 11.72 MPa.$m^{1/2}$ when the volume fraction of ZrO_2, TiB_2 and Al_2O_3 is 80%, 9% and 11%,

respectively. While the flexural strength and hardness with the latter compositions is only 568 MPa and 10.72GPa, respectively. It suggests that comprehensive good mechanical properties of the namo-micro-composite ceramic tool and die material $ZrO_2/TiB_2/Al_2O_3$ can be achieved when the volume fraction of ZrO_2, TiB_2 and Al_2O_3 is 85%, 8% and 7%, respectively.

Number	V_{ZrO2} (%)	V_{TiB2} (%)	V_{Al2O3} (%)	Hardness (GPa)	Flexural strength (MPa)	Fracture toughness (MPa $m^{1/2}$)
1	85	6	9	10.41	581	10.33
2	85	7	8	10.62	652	10.24
3	85	8	7	10.74	685	10.38
4	85	9	6	10.68	674	10.50
5	80	6	14	10.58	525	10.73
6	80	7	13	10.69	537	11.28
7	80	8	12	10.72	547	11.63
8	80	9	11	10.72	568	11.72
9	80	11	9	10.66	662	10.66
10	80	12	8	10.47	657	9.94
11	80	13	7	10.10	590	9.15
12	80	14	6	9.89	538	8.39
13	75	11	14	10.33	539	11.41
14	75	12	13	10.42	519	10.50
15	75	13	12	10.46	510	9.69
16	75	14	11	10.43	517	8.90
17	60	10	30	9.74	567	7.33
18	60	15	25	9.75	567	7.27
19	60	20	20	9.76	567	7.24
20	60	25	15	9.06	407	7.05
21	60	30	10	9.76	506	5.75

Table 6. The predicted results of GA-BP III algorithm in compositions optimization

4.4 Results and discussion

According to the above predicted results of three algorithms (BP/GA-BP II/GA-BP III) and the analysis, 85%ZrO_2, 8vol%TiB_2 and 7vol%Al_2O_3 are chosen as the optimum compositions since material with the ingredients will have the best flexural strength, the best hardness and the better fracture toughness. Then, $ZrO_2/TiB_2/Al_2O_3$ nano-micro-composite ceramic tool and die material with the above optimum compositions is prepared with the vacuum hot pressing techniques described in section 3. Compared with the above two algorithms, the GA-BP III algorithm has less iteration number, shorter elapsed-time and smaller MSE.

Both the experimental data and the predicted data of these kinds of methods mentioned above are all listed in Table 7 as well as the relative errors between the predicted and

experimental data. It can be seen that the two kinds of the improved algorithms of both GA-BP II algorithm and GA-BP III algorithm all have higher prediction accuracy than the standard BP algorithm. However, the GA-BP III algorithm has the least relative error among the three algorithms. The least relative error of the hardness, flexural strength and fracture toughness is 1.8%, 1.4% and 0.7%, respectively which is approximately 38%, 20% and 32% of that of GA-BP II algorithm and 20%, 19% and 9% of that of standard BP algorithm. The predicted data of GA-BP III algorithm better coincide with the experimental data. Therefore, it can well be used in the compositional design of ceramic tool and die materials with high accuracy of prediction and high reliability.

	Hardness (GPa)	Relative error (%)	Flexural strength (MPa)	Relative error (%)	Fracture toughness (MPa·m$^{1/2}$)	Relative error (%)
Experimental	10.95	/	694	/	10.30	/
Standard BP	9.94	9.2	643	7.4	11.14	8.1
GA-BPII	10.43	4.7	645	7.1	10.07	2.2
GA-BPIII	10.74	1.8	685	1.4	10.38	0.7

Table 7. Comparison of the optimal results of three algorithms and experimental results of the ZrO_2 based ceramic tool and die material with 8vol%TiB_2 and 7vol%Al_2O_3

5. The optimization of hot pressing parameters

As is known, the mechanical properties of ceramic materials depend on the composition and microstructure of the material. So in addition to the material compositions, the hot pressing parameters are the main factors affecting the microstructure and the mechanical properties. When one of the hot pressing parameters is changed, the sample material is needed to prepare and the mechanical properties have to be tested. If it is necessary, microstructural and phase analysis will even be needed to do. This will result in the disadvantages of high cost and long time-consuming, etc. In this section, the standard BP neural network and the improved BP neural network GA-BP II and GA-BP III are used to optimize the hot pressing parameters of $ZrO_2/TiB_2/Al_2O_3$ namo-micro-composite ceramic tool and die materials. And based on the optimum results, the materials are then prepared and mechanical properties are tested in order to validate the optimization algorithms.

5.1 The optimization of hot pressing parameters based on the standard BP algorithm

BP neural network can also be used to achieve the nonlinear mapping relationship between the hot pressing parameters and the mechanical properties of the ceramic tool and die material.

The training sample data of BP neural network are the experimental data (Table 2). The input is the hot pressing parameters, including the sintering temperature and holding time. And the output is the main mechanical properties, including hardness, flexural strength and fracture toughness. Simulated data are selected from all the data in range of the sintering temperature and holding time, which are listed in Table 8.

Based on the actual optimal problem, there are two inputs and three outputs of the BP neural network model. Therefore, the BP model is then established, which has two input neurons and three output neurons. The transfer function is sigmoid-type and linear-type in the hidden layer and output layer, respectively.

Number	Sintering temperature (°C)	Holding time (min)	Number	Sintering temperature (°C)	Holding time (min)
1	1420	20	11	1460	20
2	1420	40	12	1460	40
3	1420	60	13	1460	80
4	1420	80	14	1470	20
5	1430	20	15	1470	40
6	1430	40	16	1470	80
7	1430	80	17	1480	20
8	1440	20	18	1480	40
9	1440	60	19	1480	60
10	1440	80	20	1480	80

Table 8. The simulated data in the optimization of hot pressing parameters

According to the theory of the BP neural network, the computing process is programmed with neural network toolbox in MATLAB. Training function is using 'trainlm' function and network performance parameters is using MSE function. The training parameters are set as the same as that in the compositions optimization. And other parameters are set by default.

Number	Sintering temperature (°C)	Holding time (min)	Hardness (GPa)	Flexural strength (MPa)	Fracture toughness (MPa·m$^{1/2}$)
1	1420	20	13.55	726	10.26
2	1420	40	13.58	751	10.03
3	1420	60	12.94	1151	12.15
4	1420	80	12.55	1104	11.10
5	1430	20	13.55	722	10.23
6	1430	40	13.49	764	10.47
7	1430	80	12.70	1085	11.97
8	1440	20	13.45	673	9.76
9	1440	60	13.78	1001	10.27
10	1440	80	13.24	984	11.23
11	1460	20	13.19	543	8.52
12	1460	40	12.72	700	10.13
13	1460	80	13.80	722	8.54
14	1470	20	13.29	521	8.21
15	1470	40	12.80	700	9.91
16	1470	80	14.43	614	6.37
17	1480	20	14.00	371	6.04
18	1480	40	13.20	635	8.73
19	1480	60	13.16	816	9.40
20	1480	80	14.53	618	5.90

Table 9. The predicted results of standard BP algorithm in the optimization of hot pressing parameters

According to the BP neural network model, the number of hidden neurons is initially chosen as 6, so the neural network structure is 2×6×3. Based on this BP model, the hot pressing parameters are optimized and the mechanical properties are obtained by prediction. Because of the differences of the initial data, the BP neural network is easy to be shocked, especially in the optimization parameters. Under such circumstances, four times of the separate BP neural network prediction and simulation is carried out, but the result of each MSE is not the same. The MSE = 6.45 is selected which is nearly the average value in the four times, and the predicted results are listed in Table 9. After 40 times of iterations, the training curve of BP neural network is converged to the specified precision of 0.001.
According to the predicted results, the highest flexural strength and fracture toughness of the materials is 1151MPa and 12.15 MPa·$m^{1/2}$, respectively when the sintering temperature is 1420°C and the holding time is 60min, while the hardness is just 12.94GPa. The highest hardness of the material is 14.53GPa which corresponds to the sintering temperature of 1480°C and the holding time of 80min. In this case, the flexural strength of the material is 618MPa and the fracture toughness is 5.90 MPa·$m^{1/2}$. The hardness of the material which is prepared with these hot pressing parameters reaches the highest, but both flexural strength and fracture toughness are relative low. Compared with the mechanical properties of the ceramic tool and die materials prepared with different hot pressing parameters, it seems that the ceramic tool and die material which is fabricated with sintering temperature of 1420°C and holding time of 60min has better comprehensive mechanical properties. Therefore, these hot pressing parameters are the optimum hot pressing parameters for the fabrication of $ZrO_2/TiB_2/Al_2O_3$ namo-micro-composite ceramic tool and die material.

5.2 The optimization of hot pressing parameters based on GA-BP II algorithm

According to the formerly established BP model where the number of the neurons of hidden layer is 6 and the structure of the BP model is 2×6×3, the GA-BP II algorithm is then utilized to optimize the hot pressing parameters. The mechanical properties are obtained and given in Table 10. After 40 times of iterations, the training curve of BP neural network of GA-BP II algorithm is converged to the specified precision of 0.001. The mean square error MSE is 4.27.
After analyzing the predicted results, the material is prepared with the sintering temperature of 1420°C and the holding time of 60min. It has the best flexural strength and the best fracture toughness which is 1052MPa and 10.59 MPa·$m^{1/2}$, respectively. Under the same hot pressing parameters, however, the hardness of the material is 13.36GPa which is slightly lower. The highest hardness of the material amounts to be 14.28GPa where the corresponding sintering temperature is 1420°C and the holding time is 80min, while the flexural strength is 1051MPa and the fracture toughness is 10.40 MPa·$m^{1/2}$. Compared with the mechanical properties of ceramic tool and die material which is prepared in different hot pressing parameters, it suggests that the comprehensive good mechanical properties of $ZrO_2/TiB_2/Al_2O_3$ namo-micro-composite ceramic tool and die material can be achieved when the sintering temperature is 1420°C and the holding time is 60min.

5.3 The optimization of hot pressing parameters based on GA-BP III algorithm

According to the actual problem, the input layer neuron number is 2, the output layer neuron number is 3, and the number of hidden layer neurons is set to d. According to GA-BP III algorithm, the string length L can be determined as $L=3+6d$. In accordance with the

Number	Sintering temperature (°C)	Holding time (min)	Hardness (GPa)	Flexural strength (MPa)	Fracture toughness (MPa·m $^{1/2}$)
1	1420	20	14.25	1042	10.39
2	1420	40	14.27	1035	10.51
3	1420	60	13.36	1052	10.59
4	1420	80	14.28	1051	10.40
5	1430	20	13.37	776	9.91
6	1430	40	14.17	1037	10.31
7	1430	80	13.26	1050	10.30
8	1440	20	12.82	624	9.92
9	1440	60	13.78	1010	10.26
10	1440	80	13.31	1035	10.54
11	1460	20	12.83	857	8.42
12	1460	40	12.42	870	9.77
13	1460	80	13.86	597	8.21
14	1470	20	12.15	1006	8.94
15	1470	40	12.29	1005	8.92
16	1470	80	13.29	985	8.87
17	1480	20	12.23	1000	8.87
18	1480	40	13.62	826	7.63
19	1480	60	14.05	704	7.53
20	1480	80	13.25	831	9.11

Table 10. The predicted results of GA-BP II algorithm in the optimization of hot pressing parameters

empirical formula (Eq. 2) which can determine the range of hidden layer neurons, the range of d is 3-12. According to the principle of GA-BP III algorithm, the computing process are programmed and run with MATLAB 7.0 software. The corresponding parameters are set as following: the initial population number N=30, the cross probability P_c=0.8, the mutation probability P_m=0.1 and the error e=0.001. When the error reaches the intended target, the training parameters of BP neural network is then stopped.

The individuals of the connection weight and thresholds are decoded as follows: 0.32, -0.14, 0.36, -0.29, 0.24, 0.16, 0.24, -0.88, -0.24, 0.16, -0.16, 0.60, 0.44, -0.69, -0.40, 0.03, 0.26, 1, 0.39, -0.29, 0.21, -0.49, 0.00, 1, -0.20, -1, -1, -0.68, 0.00, 0.00, 0.35, 0.02, 0.32, -0.27, 1, 0.09, -0.13, -0.23, 0.15.

Based on the above 39 parameters and $L=3+6d$, the number of hidden layer neurons is ascertained as 6. Therefore, the structure of BP neural network is 2×6×3 and the last 9 parameters are the threshold values. The structure is shown in Fig.8 which is the optimal BP neural network of GA-BP III algorithm. It can be seen that the second neuron of input layer and the fifth neuron of hidden layer is no connection. The sixth neuron of hidden layer and the second and third neurons of output layer are also connectionless. After about 100 generations of searching, the fitness and the square error have been stabilized respectively as shown in Fig. 9. The curve of BP training target is shown in Fig. 10. It is shown that the BP neural network has 45 iterations convergence to the specified accuracy. The elapsed-time is 155.584s and the MSE is 0.1643.

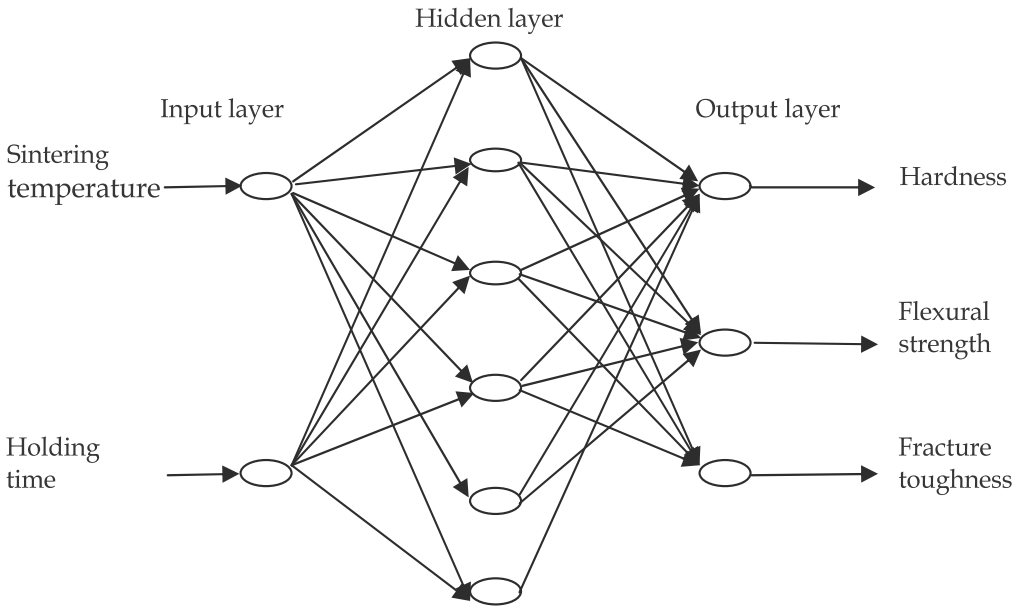


Fig. 8. The structure of BP neural network for GA-BP III algorithm simulation in the optimization of hot pressing parameters

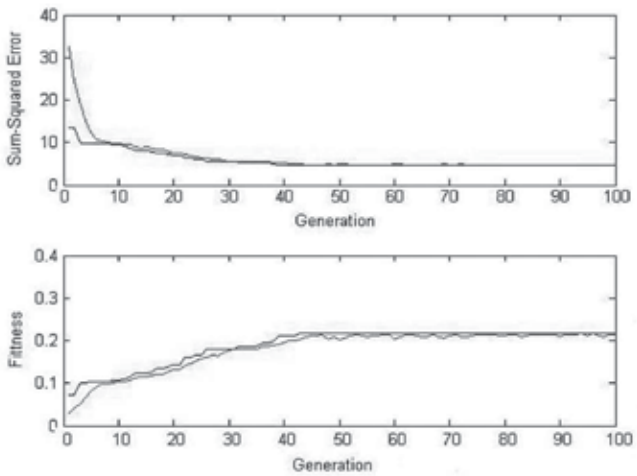


Fig. 9. The curve of square error and fitness of GA-BP III algorithm in the optimization of hot pressing parameters

The predicted results of GA-BP III algorithm are given in Table 11. It can be seen that the optimum flexural strength and the optimum fracture toughness is 1010MPa and 10.40 MPa·m$^{1/2}$ respectively when the material is prepared with the sintering temperature of 1420°C and the holding time of 60min. The hardness of the material fabricated in these hot pressing parameters is 13.43GPa. The optimum hardness is 14.14GPa which is corresponding to the sintering temperature of 1420°C and the holding time of 80min, while

Number	Sintering temperature (°C)	Holding time (min)	Hardness (GPa)	Flexural strength (MPa)	Fracture toughness (MPa·$m^{1/2}$)
1	1420	20	13.72	1002	10.91
2	1420	40	13.70	1004	10.38
3	1420	60	13.43	1010	10.40
4	1420	80	14.14	804	9.55
5	1430	20	13.71	996	10.34
6	1430	40	13.71	1005	10.38
7	1430	80	14.02	858	8.06
8	1440	20	13.62	818	9.80
9	1440	60	13.78	1005	10.27
10	1440	80	14.06	897	8.20
11	1460	20	12.08	768	9.82
12	1460	40	11.95	827	9.31
13	1460	80	13.52	493	8.56
14	1470	20	11.69	850	9.96
15	1470	40	12.30	885	9.67
16	1470	80	13.63	427	8.38
17	1480	20	11.70	857	9.96
18	1480	40	12.96	909	9.19
19	1480	60	13.42	715	8.46
20	1480	80	13.66	431	8.37

Table 11. The predicted results of GA-BP III algorithm in the optimization of hot pressing parameters

the flexural strength and fracture toughness is just 804MPa and 9.55MPa·$m^{1/2}$, respectively. Both values are obviously lower than the optimum. Therefore, the optimum hot pressing parameters are that the sintering temperature is 1420°C and the holding time is 60min which is the same as that of GA-BP II algorithm.

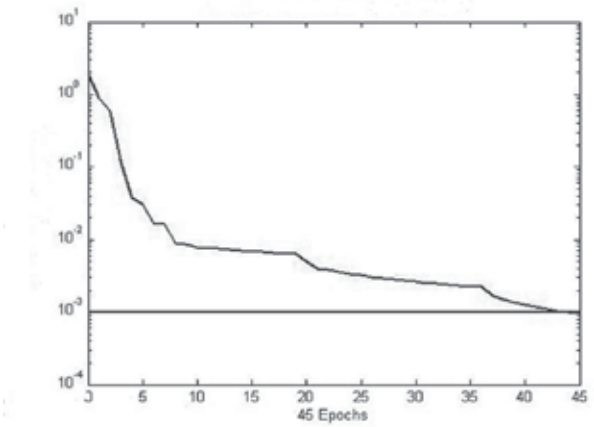


Fig. 10. The training curve of BP neural network of GA-BP III algorithm in the optimization of hot pressing parameters

5.4 Results and discussion

According to the predicted results of three algorithms, the sintering temperature of 1420°C and holding time of 60min are determined as the optimum hot pressing parameters. Then, with these optimized hot pressing parameters, $ZrO_2/TiB_2/Al_2O_3$ nano-micro composite ceramic tool and die material with the above optimum compositions is prepared by means of the vacuum hot pressing technique described in section 3 and mechanical properties are tested.

	Hardness (GPa)	Relative error (%)	Flexural strength (MPa)	Relative error (%)	Fracture toughness (MPa·$m^{1/2}$)	Relative error (%)
Experimental	13.3	/	937.0	/	10.17	/
Standard BP	12.9	2.8	1151.5	22.8	11.10	9.1
GA-BP II	13.4	0.9	1052.5	12.3	10.60	4.2
GA-BP III	13.4	0.9	1009.7	7.8	10.40	2.2

Table 12. Comparison of the optimal results of three algorithms and experimental results in the optimization of hot pressing parameters

Table 12 gives the experimental mechanical properties of the $ZrO_2/TiB_2/Al_2O_3$ nano-micro-composite ceramic tool and die material which is prepared under the optimum hot pressing parameters. The predicted results and the relative errors are both listed.

Compared with the experimental values, the least relative error of flexural strength and fracture toughness is 7.8% and 2.2% obtained by GA-BP III algorithm which is approximately 63% and 48% of that of GA-BP II algorithm and 34% and 24% of that of standard BP algorithm. The least relative error of hardness is 0.9% obtained by GA-BP III algorithm which is the same as that obtained by GA-BP II algorithm. In addition to the same relative error of hardness by GA-BP II algorithm, other relative errors of mechanical properties by GA-BP III are the least. So the predicted results of GA-BP III algorithm are the most accurate in these three algorithms. The predicted data of GA-BP III algorithm better coincide with the experimental data. Therefore, it can well be utilized for the optimum design of hot pressing parameters of ceramic tool and die materials with high accuracy of prediction and reliability.

6. Conclusion

With the utilization of GA-BP III algorithm for the compositional design of nano-micro-composite ceramic tool and die material, the iteration number could noticeably be reduced and results are more accurate. It can avoid the local minimum problem and can present more accurate and reliable results. And it also can overcome the disadvantages of both long time and slow speed of the standard BP neural network. Preparation experiments of $ZrO_2/TiB_2/Al_2O_3$ nano-micro-composite ceramic tool and die material indicate that the relative error between the experimental and predicted results of the hardness, flexural strength and fracture toughness is 1.8%, 1.4% and 0.7%, respectively by the GA-BP III algorithm which is the least relative error among three kinds of algorithms. The predicted data better coincide with the experimental data high accuracy of prediction.

The GA-BP III algorithm can also well be used in the optimization of hot pressing parameters of nano-micro-composite ceramic tool and die material. It can reduce the

number of iterations. The optimization results are more precise. The GA-BP III algorithm can avoid falling into local minimum which is the shortcoming of standard BP algorithm, and can obtain more accurate and reliable optimization results. Compared with the experimental results and the predicted result of standard BP neural network, it indicates that the improved BP algorithms, especially GA-BP III algorithm are suitable for the optimization of hot pressing parameters of $ZrO_2/TiB_2/Al_2O_3$ nano-micro-composite ceramic tool and die materials.

Therefore, the GA-BP III algorithm is one of the fast, effective and reliable algorithms in the optimum design of both compositions and hot pressing parameters of nano-micro-composite ceramic tool and die materials. It suggests that it can also be effectively applied in the material design area of other ceramic composites.

7. Acknowledgement

This work was supported by Shandong Provincial Natural Science Foundation, China (Grant No. ZR2009FZ005), the Program for New Century Excellent Talents in University of China (Grant No. NCET-10-0866), the National Natural Science foundation of China (Grant No. 51075248), Shandong Provincial Natural Science Foundation for Distinguished Young Scientists, China (Grant No. JQ201014).

8. References

Altinkok N. & Korker R. N. (2004). Neural Network Approach to Prediction of Bending Strength and Hardening Behaviour of Particulate Reinforced (Al-Si-Mg) Aluminium Matrix Composites. *Materials and Design*, Vol. 25, No. 7, Oct, 2004, pp. 595-602, ISSN 0261-3069

Altinkok N. & Korker R. N. (2005). Mixture and Pore Volume Fraction in Al_2O_3/SiC Ceramic Cake Using Artificial Neural Networks. *Materials and Design*, Vol. 26, No. 4, Jun, 2005, pp. 305-311, ISSN 0261-3069

Gen M. & Cheng R. (2000). *Genetic algorithm and Engineering Optimization*, Wiley, ISBN 0-471-31531-1, New York

Gu X.Q.; Yi D.X & Liu C.H. (2006). Optimization of Topological Structure and Weight Value of Artifical Neural Network using Genetic Algorithm. *Journal of Guangdong University of Technology*, Vol. 23, No. 4, 2006, pp. 64-69, ISSN 1007-7162 (in Chinese)

Gupta J.N.D. & Sexton R.S. (1999). Comparing Back propagation with a Genetic Algorithm for Neural Network Training. *The International Journal of Management Science*. Vol. 27, Dec, 1999, pp. 679-684, ISSN 1226-0797

Huang C.Z.; Zhang L.; He L.; Sun J.; Fang B. & B. Zou. J. (2002). A Study on the Prediction of the Mechanical Properties of a Ceramic Tool Based on an Artificial Neural Network. *Journal of Materials Processing Technology*, Vol. 129, Oct, 2002, pp. 399-402, ISSN 0924-0136

Jiang X. & Adeli H. (2004). Clustering-Neural Network Models for Freeway Work Zone Capacity Estimation. *International Journal of Neural Systems*, Vol. 14, No. 3, Jun, 2004, pp. 147-163, ISSN 0129-0657

Kim B. & Bae J. (2005). Prediction of Plasma Processes Using Neural Network and Genetic Algorithm. *Solid-State Electronics*, Vol. 49, Oct, 2005, pp. 1576-1580, ISSN 0038-1101

Mousavi Anijdan S.H.; Madaah-Hosseini H.R. & Bahrami A. (2007). Flow stress Optimization for 304 Stainless Steel under Cold and Warm Compression by Artificial Neural Network and Genetic Algorithm. *Materials and Design*, Vol. 28, 2007, pp. 609-615, ISSN 0261-3069

Ozcelik B.; Oktem H. & Kurtaran H. (2005). Surface Roughness in End Milling Inconel 718 by Coupling Neural Network Model and Genetic Algorithm. *The International Journal of Advanced Manufacturing Technology*, Vol. 27, 2005, pp. 234-241, ISSN 0268-3768

Scott D.J.; Coveney P.V.; Kilner J.A.; Rossing J.C.H.; Alford N.M. & Ceram J. (2007). Prediction of the Functional Properties of Ceramic Materials from Composition Using Artificial Neural Networks. *Journal of the European Ceramic Society*, Vol. 27, Dec, 2007, pp. 4425-4435, ISSN 0955-2219

Sexton R.S.; Dorsey R.E. & Johnson J.D. (1998). Toward Global Optimization of Neural Network: A Comparison of the Genetic Algorithm and Backpropagation. *Decision Support Systems*. Vol. 22, Feb, 1998, pp. 171-185, ISSN 0167-9236

Yao X. (1999). Evolving Artificial Neural Network, *Proceedings of IEEE*, pp. 1423-1447, ISBN 0018-9219, Sep 1999, IEEE Press. Vol. 87

Yen G.& Lu H. M. (2002). Hierarchical Genetic Algorithm for Near-Optimal Feed Forward Neural Network Design. *International Journal Neural Systems*, Vol. 12, No. 1, Feb, 2002, pp. 31-43, ISSN 0129-0657

Zemin F.; Jianhua M. & Lin C. (2010). Using Genetic Algorithm-Back Propagation Neural Network Prediction and Finite-element Model Simulation to Optimize the Process of Multiple-step Incremental Air-bending Forming of Sheet Metal. *Materials and Design*, Vol. 31, Jan, 2010, pp. 267-277, ISSN 0261-3069

Zhu D.Q. & Shi H. (2006). *The Principle and Application of Artificial Neural Networks*, Science Press, ISBN 7-03-016570-5, Beijing (in Chinese)

7

Application of Bayesian Neural Networks to Predict Strength and Grain Size of Hot Strip Low Carbon Steels

Mohammad Reza Toroghinejad and Mohsen Botlani Esfahani
Dept. of Materials Engineering, Isfahan University of Technology, Isfahan, 84156-83111, Iran

1. Introduction

Low alloy steels are the most demanding materials that are used in industrial processes such as hot stripping. Hot stripping is a severe plastic deformation which is applied on cast steels for a variety of shapes and sizes. A hot strip mill consists of, from start to finish, reheat furnaces, roughing mill, finishing mill, runout table with accelerated cooling and finally a coiler, as shown in Figure 1.

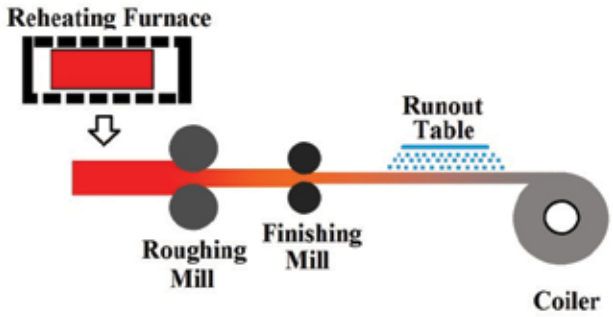


Fig. 1. Schematic illustration of hot strip mill.

The process enhances the properties of steels by several metallurgical mechanisms which take place in different parts of the hot strip mill. This process is illustrated in Figure 2 which includes following metallurgical phenomena:

1. Austenitization, dissolution of microalloy compounds and homogenization of the chemical segregation in the reheating furnace.
2. Deformation and reduction of reheated slab to intermediate thickness which is accompanied with recrystallization, grain growth and precipitation of alloying and microalloy elements in roughing and finishing mills.
3. Phase transformation and precipitation during cooling and decreasing the heat to room temperature (Ryu, 2008), (Gonzalez, 2002).

These mechanisms by refinement of structure bring about a simultaneous improvement in strength and toughness (Singh et al., 1998).

1.1 Grain size effect on mechanical properties

Grain size is an important aspect of microstructure with respect to mechanical properties of steels. The ferrite in low carbon steels is typically strengthened by grain refinement, precipitation hardening, and, to a lesser extent, solid-solution strengthening. Grain refinement is the most desirable strengthening mechanism because as mentioned earlier it improves not only strength but also toughness. According to equation (1) indicating the Hall-Petch relation, fine grain size produces higher yield strength, (σ_{yield}) (Parker, 1997):

$$\sigma_{yield} = \sigma_{init} + k_y d_\alpha^{-1/2} \tag{1}$$

where σ_{init} is the yield strength for a polycrystalline material, k_y is a constant, and d_α is a measure of the ferrite grain size. Grain size also has an effect on the ultimate tensile strength by changing work-hardening rate. Work-hardening takes place within the grains during plastic deformation according to Morrison (Ryu, 2008):

$$n = 5 / (10 + d_\alpha^{-1/2}) \tag{2}$$

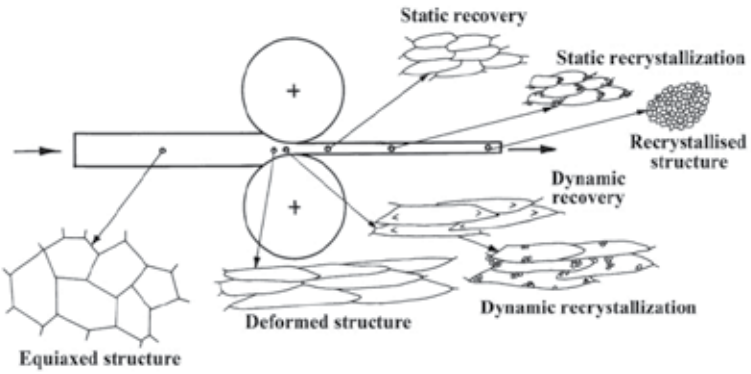


Fig. 2. Related metallurgical phenomena.

where n is work-hardening exponent and d_α is grain size. Ferrite mainly nucleates at the austenite grain boundaries and thus a finer austenite grains produces fine ferrite grains. Further ferrite refinement can be achieved by transformation from deformed austenite grains because, deformation increases ferrite nucleation rate (Parker, 1997). The effects of chemical composition on these properties are an important parameter as well as thermo-mechanical processing features such as temperature and final dimensions (Ryu, 2008). The additions of some alloying elements affect ferrite transformation and thus control the amount of phases present in the final matrix. The presence of microalloying elements generally control the grain size and provide precipitation strengthening and have a significant impact on the strength (Singh et al., 1998). Therefore, estimating of strength and grain size of hot stripped steel products depends on thermo-mechanical behavior of steel, microstructure evolution and phase transformation, during hot rolling stages and cooling period. These are complicated metallurgical phenomena and strongly depend on chemical composition, therefore developing a physical model to analysis these parameters and predict strength as well as final grain size, is cumbersome. Also, the accuracy of the models developed so far is somehow questionable and

are not suitable for practical purposes. Traditionally, setting the tolerances is carried out by making several samples and checking the final results by trial and error approach. Generally, these procedures are expensive and time-consuming especially in such a complex metallurgical phenomena. Consequently, the overall effects of these features can have an effect on rolling design and therefore too many experimental trials are needed to achieve ideal tolerances. Since estimating these properties of low carbon steel strips in terms of chemical composition and thermo-mechanical parameters is desirable from engineering view point, several models are introduced based upon different neural network methods. These models are capable of understand very complex and unknown relationships between inputs and output data. Furthermore, the models can explore the effect of the individual input on output which can be extremely difficult in the experimental tasks. Achieved model for estimating tensile strength can be used as a quantitative tool to predict the final tensile strength of these commercial low carbon steels with different of input variables. Moreover analysis of the effect of input parameters on results may leads to design new steels with different input parameters. In the present work also, by selecting more relevant inputs and using hybrid Bayesian Artificial Neural Network (ANN) model assisted with Reversible-Jump Markov Chain Monte Carlo (RJMCMC, also known as trans-dimensional MCMC), the prediction of final grain size in low carbon steel strips is carried out.

2. Method

2.1 Artificial Neural Networks

A neural network is an interconnected network of a set of simple processing units which are connected by a set of connections called "weights". They can learn the given information by a set of examples and transfer them to their structure. The method which is inspired from studying the human brain, is capable of recognizing complex patterns of the training data and can be applied to regression and classification tasks. The training is an optimization procedure by finding a set of weights which combined with processing units, describes the data pattern. There are several advantages in this method. Firstly, there is no need to choose the behavior of the model in advance. Secondly, its need to train data, does not grow as fast as other conventional regression methods and therefore, growing the complexity and dimensionality of the problem doesn't need any further data (Botlani-Esfahani et al., 2009a). Basic ANN model with k outputs is

$$f_k(x,w) = w_{k0} + \sum_{j=1}^{m} w_{kj} \tanh(w_{j0} + \sum_{i=1}^{d} w_{ji} x_i) \tag{3}$$

where x is a d-dimensional input vector, w denotes the weights, and indices i and j correspond to input and hidden units, respectively (Lampinen & Vehtari, 2001). Arrangement of layers and units in an ANN called architecture (Doan & Yuiliong, 2004). Figure 3 sketches schematic architecture of a feed forward ANN model. In each layer, units receive their input from previous layer's units and send their output to units in the following layer. Output of each hidden unit is the transfer function response to the weighted sum of its inputs.

The number of units in input and output layers are dictated by the problem, but the number of hidden units which control the complexity of the model, must be determined.

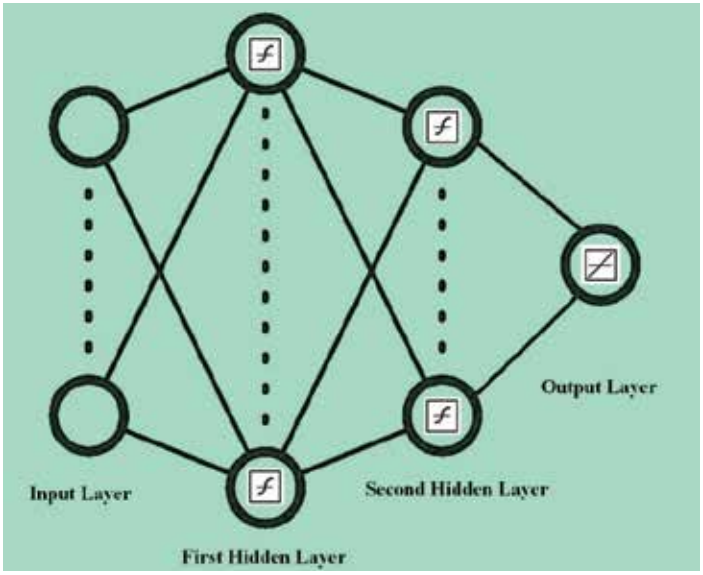


Fig. 3. Schematic architecture of Artificial Neural Network model.

The processing units for computational convenience, like nonlinear hyperbolic tangent sigmoid functions are easily differentiable, and are employed in the present model, Equation (4):

$$\tanh(x) = \frac{2}{(1+\exp(-2x))-1} \tag{4}$$

Traditionally the complexity of the ANN has been controlled with early stopping. In which part of the training data is used to train the network and other part is used to control the complexity of the model. Early stopping is inefficient because the effective complexity may be much less than the number of parameters in the model. Consequently, tow different Bayesian Learning paradigm for ANN was employed to train models.

2.2 Bayesian learning for ANN

In the Bayesian framework which has introduced by MacKay (MacKay, 1992) the weights of the network are considered as random variables and the posterior distribution of the weights updated according to Bayes' rule (Xu et al., 2006):

$$Posterior = \frac{likelihood \times prior}{Evidence} \tag{5}$$

This equation in terms of Artificial Neural Networks is:

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} \propto L(\theta|D)p(\theta) \tag{6}$$

where $p(\theta)$ is prior distribution for the model parameters θ ,D = {(x(1),y(1)),…, (x(n),y(n))} is observing data and $L(\theta|D)$ is likelihood function that gives the probability of the observed data as function of the unknown model parameters (Lampinen & Vehtari, 2001).

2.3 Reversible jump Markov Chain Monte Carlo method

Neal has introduced an implementation of Bayesian learning for ANN in which, difficult integrations accompanied with this framework are performed using Markov Chain Monte Carlo (MCMC). In this application samples (in model parameters space) are generated using a Markov Chain Monte Carlo to estimate the desired posterior distributions (Lampinen & Vehtari, 2001). In practical problems like the present study, it is usual to measure many variables, but it is not necessarily known which one of them is relevant and required to solve the problem. To make the model more explainable or to reduce the measurement cost and the computation time, it may be useful to select a model with smaller set of input variables (Lampinen & Vehtari, 2001). As a consequence, RJMCMC method is applied for this modeling. This algorithm allows jumps between models with different dimensional parameter spaces with respect to the number of inputs chosen in the model. RJMCMC visits the models according to their posterior probability which allows it to be used for model selection (Vehtari & Lampinen, 2002). The grain size model was achieved by this method.

2.4 Bayesian Regularized Neural Network (BRNN)

Conventional performance function of neural network which optimization applied on it, has general form of:

$$F = mse = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(t_i - a_i)^2 \tag{7}$$

where mse is mean of squared error. If the performance function is changed by adding a term that contains mean of squared weights (msw), yield:

$$msereg = \gamma mse + (1-\gamma)msw \tag{8}$$

where γ is the performance ratio, and

$$msw = \frac{1}{n}\sum_{j=1}^{n} w_j^2 \tag{9}$$

Using this performance function leads to smaller network weights and biases, which makes the network response to be smoother and less likely to over-fit (MathWorks). The main remaining problem is to find the ideal value for the performance ratio. Choosing too large ratio increases over-fitting likelihood and too small ratio prevents network to fit adequately the training data (MacKay, 1992). To find out the best regularization, as mentioned before MacKay in his Bayesian framework suggests, assuming the weights and biases as random variables with specified distributions and related the regularization parameters to these distributions. Another approach suggested by Foresee (Botlani-Esfahani et al., 2009b) in which the Levenberg-Marquardt method that is a rapid optimization algorithm employed for training. The (BRNN) automatically can control the complexity of the model and prevent the over-fitting of training data set. As a result, this model has good prediction accuracy and according to MacKay, (MacKay, 1992) in Bayesian framework, there is no need for test data set to control the specified network architecture. The acquired model by this approach can reveal a good generalization, even if its architecture is an over optimized (MathWorks). Consequently, the trial and error approach for finding ideal architecture is reduced. This approach was applied to acquire ideal model for prediction of tensile strength of steel strips because of its good accuracy as well as fast convergence speed.

2.5 Experimental database

Since an ANN model is empirical, its performance depends on the dataset used for training. Annual products data report of Isfahan Mobarakeh Steel Company (MSC) were used for this modeling, which input parameters for tensile strength modeling consisted of:

i. Final thickness
ii. Initial and final weight
iii. Initial width
iv. Reheating furnace temperature, roughing temperature, finishing temperature and coiling temperature
v. The chemical composition, consisting 14 different elements
vi. The carbon equivalent according to the following formula:

$$Ceq = C + Si/25 + (Mn+Cr)/16 + (Cr+Ni+Mo)/20 + V/15 \quad (10)$$

where elements are expressed in weight percent.

About 70234 examples each consisting of corresponding input and output were available for modeling tensile strength. Some further information about the variables are given in Table 1. These examples were normalized so that they had zero mean and unity standard deviation before computations.

Data set that was used to model grain size consisted of 624 metallographic images. At this company, these images are classified into three groups according to ASTM (E-112), this standard assigns larger numbers to finer grain structures. Figure 4 shows an example of such a database. Further information is also given in Table 2. The input parameters are chemical composition of the strips which include 14 elements. Additional input variables are given in Table 3.

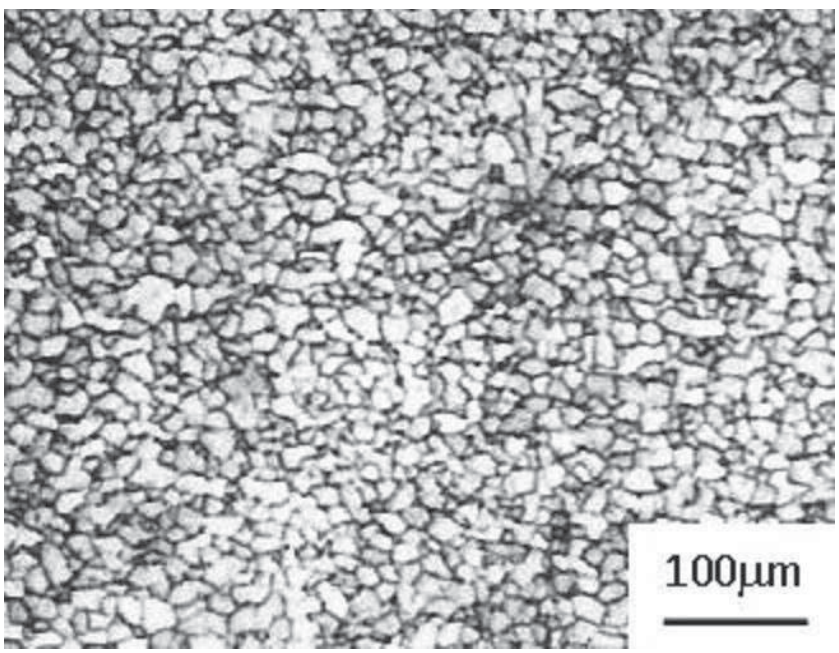


Fig. 4. One sample of data, microstructure of produced steel with ASTM grain size no. 9.0.

	Variables	min	max	mean	SD
Inputs	Final Thickness(mm)	1.5	16	5.244903	3.155532
	Final Weight(kg)	5097	28030	18502.91	3214.769
	Initial Weight(kg)	5202	28660	18874.26	3264.811
	Initial Width(mm)	650	1850	1277.022	205.7713
	Furnace Temp(°C)	1164	1296	1229.77	23.4407
	Roughing Temp(°C)	932	1122	1058.281	14.00645
	Finishing Temp(°C)	782	960	881.1131	23.32006
	Coiling Temp(°C)	517	729	610.5108	18.02052
	C (wt %)	0.03	0.21	0.126968	0.02545
	Si (wt %)	0	0.347	0.070235	0.084277
	Mn (wt %)	0.175	1.38	0.658662	0.206133
	P (wt %)	0.001	0.026	0.006786	0.002377
	S (wt %)	0	0.02	0.008637	0.002686
	Cu (wt %)	0	0.264	0.029318	0.011597
	Al (wt %)	0.007	0.093	0.045926	0.010957
	N (ppm)	15	90	39.784	9.221
	Nb (wt %)	0	0.06	0.004854	0.009032
	V (wt %)	0	0.043	0.003378	0.001607
	Ti (wt %)	0	0.042	0.001654	0.002318
	Mo (wt %)	0	0.022	0.003654	0.004104
	Cr (wt %)	0.001	0.194	0.011992	0.008007
	Ni (wt %)	0.016	0.243	0.028205	0.004679
	Ceq (wt %)	0.068032	0.437799	0.2443845	0.0534388
Output	Strength (MPa)	299	659	444.64	48.68

SD: Standard Deviation Ceq: Carbon Equivalent

Table 1. Input and output parameter information.

Number of Samples	ASTM (E-112) grain no
162	8.5
294	9
167	9.5

Table 2. Output Data Information

2.6 Network training

As mentioned the (BRNN) models have a good predictive accuracy (generalization) and specified network architecture in Bayesian framework doesn't need of test data to adjust its

complexity. However, there is still a need for an independent set of data to evaluate the ideal network predictive accuracy on unseen data. In this respect, 10% of the total data were kept for validation. In the case of tensile strength several models were examined for finding the best network architecture. The network architecture was started with a few hidden units in a single hidden layer and as the number of hidden units increased the squared error sum on both training and test data, decreased. As expected from performance function of the BRNN, (Equation 8), when the number of training data is raised, the number of weights must also increase. When the number of hidden units is placed within one layer, the accuracy of the results is less than when two layers are used. This also indicates that more weights are needed in two layer network. Finally, the ideal network was determined with 23-60-50-1 architecture. Training stopped when the squared error sum, the squared weights sum and performance ratio (which are the criteria for training evaluation) became stable. Since network training is an optimization procedure, the calculations can become cumbersome. For example, for 4541 parameters, more than 55 hours was taken when a dual 3.2 GHz processor, with 2 gigabyte memory, was used.
For finding ideal model for grain size prediction the modeling database was divided into training and test sets, which include 60 and 40 percents of data respectively. Training was started with (14-8-3) architecture and model selection procedure was evaluated by an internal procedure of RJMCMC algorithm, as mentioned in sec. 2.3. The result of this training indicated a chain of network parameters. When this chain converged into a stable distribution, a sample of the chain (network parameters) was selected on the bases of minimum classification error of the model on test dataset.

No.	Inputs	min	max	Mean	SD
1	C (wt%)	0.032	0.179	0.1272	0.0312
2	Si (wt%)	0.008	0.218	0.0637	0.0798
3	Mn (wt%)	0.191	1.15	0.6466	0.211
4	P (wt%)	0.002	0.025	0.0072	0.0022
5	S (wt%)	0.001	0.02	0.0089	0.0029
6	Cu (wt%)	0.004	0.078	0.03	0.0109
7	Al (wt%)	0.015	0.075	0.0454	0.0119
8	N (ppm)	16	75	38	8.8
9	Nb (wt%)	0	0.045	0.0051	0.0105
10	V (wt%)	0	0.011	0.003	0.0014
11	Ti (wt%)	0	0.042	0.0017	0.0031
12	Mo (wt%)	0	0.019	0.0038	0.0045
13	Cr (wt%)	0.004	0.194	0.0131	0.012
14	Ni (wt%)	0.02	0.042	0.03	0.0034

SD: Standard Deviation

Table 3. Input parameter information

2.7 Calculation of the weights of individual input variable

Extracting effective information from a neural network model is not as easy as conventional linear regression because the discovered relationships with neural network are much more complicated. However when the output layer only consists of one neuron the dependency of

output variable on inputs is same as network dependency to input parameters (Botlani-Esfahani et al., 2009b). On the other hand, in feed-forward networks the path which the effects of the input parameters carried is straightforward from input layer to output layer. Therefore, the weights which fan out the input units can be considered as their significance, like the impact of inputs on output in linear models. The relative importance of individual input variable on output variable can be expressed as: (Xu et al., 2006)

$$I = \frac{\sum_{j=1}^{S} \left| W_{ji} \right|}{\sum_{i=1}^{N} \sum_{j=1}^{S} \left| W_{ji} \right|} \tag{11}$$

where W_{ji} is the connection weight from i input neuron to j hidden neuron, N, S are the number of input parameters and hidden neurons, respectively. This approach was employed to investigate the relative importance of input parameters on tensile strength however, in case of grain size such task has carried out automatically by the algorithm.

3. Results and discussion

3.1 Performance of the model

Scatter diagrams of model predictions versus experimental data for both training data and validation data are used as a means of showing the tensile strength model generalization. Figure 5, indicates that the correlation coefficients of training and validation data are close to one, and their differences are negligible. Therefore, it is clear that, the network predictions are in good agreement with experimental data.

Calculation of the misclassification error on test data is a popular way to show the prediction accuracy (generalization) of a classifier model. This error is calculated according to:

$$Misclassification\ error = \frac{\sum \left| test\ data - model\ result \right|}{number\ of\ test\ data} \times 100 \tag{12}$$

Therefore, grain size model revealed just 2.439 percent misclassification error, which is very low and indicates that, this model has good generalization. More information about misclassified error is available in Table 4.

Number of Misclassified	Test Target Data	Model Result
2	9	8.5
2	9.5	9

Table 4. misclassified case

3.2 Sensitivity analysis

Figure 6 shows the importance of input variable relevancies on tensile strength which were analyzed by the method mentioned in section 2.7. Figure 6 shows that silicon, carbon, manganese, copper, nickel and chromium give a large contribution to the strength. Moreover, microalloy elements such as niobium, vanadium and titanium, though less than other elements, have a similar effect of strength. Among the processing features, the width and thickness of the strip revealed remarkable influence on tensile strength (Botlani-Esfahani et al., 2009b).

The depicting effect of mentioned factors and their interactions with one another, two parameters were altered at a time and other parameters were kept on their mean values which are tabulated in Table 1. As mentioned RJMCMC method can select potentially useful inputs according to marginal probabilities of inputs. The result of this analysis indicates the importance of Si, Mn and C contents on grain refinement which is significantly greater than the concentration of other elements. The most effective element for grain refinement is recognized to be that of vanadium. However, its concentration in these steels is very low.

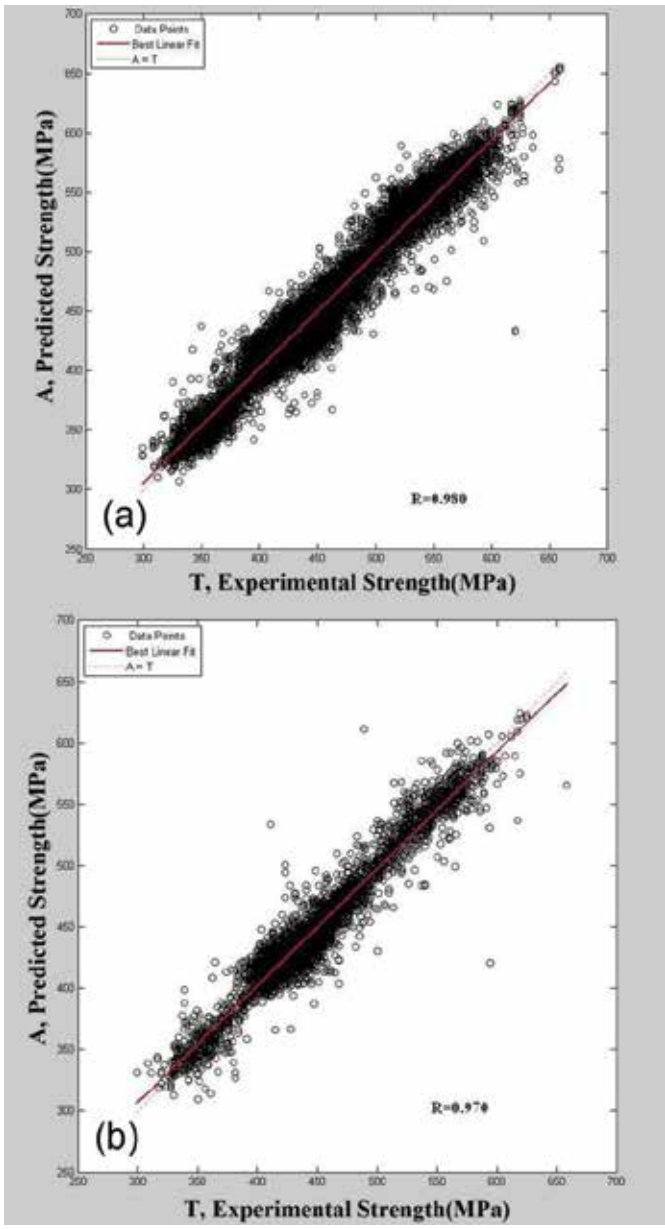


Fig. 5. Behavior of tensile strength model on (a) training data (b) test data.

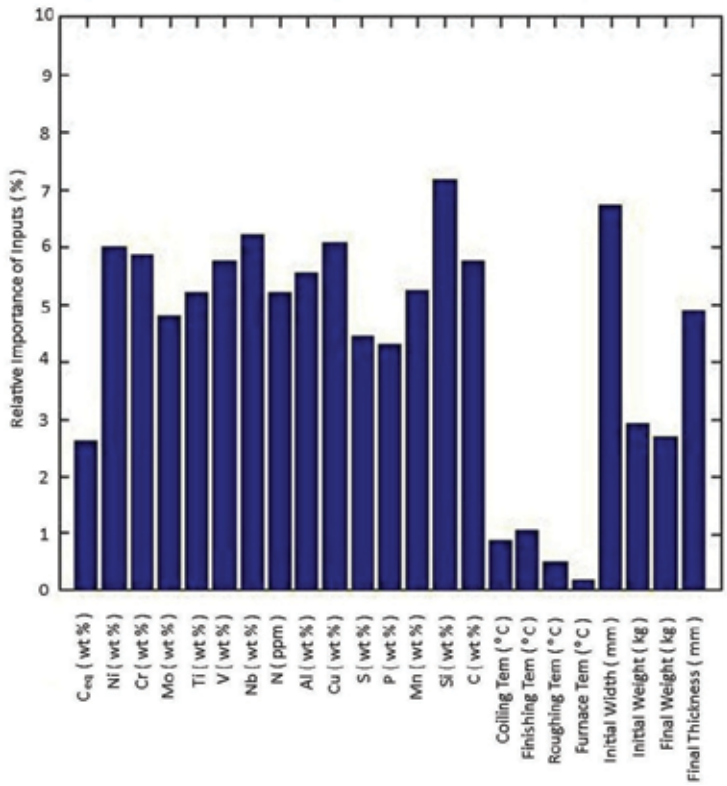


Fig. 6. Relative importance of inputs.

3.3 Tensile strength model

Carbon has a major effect on steel properties and increases the strength by interstitial solid solution strengthening. This effect is more pronounced in ferritic steels. In ferritic-pearlitic steels, the carbon content raises pearlite volume which in turn leads to the increase of alloy strength (Singh et al., 1998). Silicon is one of the principal deoxidizers used in steel-making, Figure 7a shows silicon effect which enhances the strength by suppressing precipitation of cementite from austenite. Thus carbon remains in austenite for subsequent strengthening (Bhadeshia et al., 2003). The effect is more pronounced in steels with lower carbon concentration because silicon dissolves in the ferrite. Manganese promotes stronger steels by stabilizing austenite and solid solution strengthening (Singh et al., 1998). The increase in strength is dependent upon the carbon content as is shown in Figure 7b. However the concentration of microalloy elements is low, they have a significant influence on several stages of rolling. Unlike alloying elements that alter the structure of iron, microalloy elements have a great affinity to combine with other elements such as carbon and nitrogen. This results in precipitation of several secondary phases (Meyer, 2001). Model reveals the effect of niobium concentration as the most effective microalloy. Niobium contributes towards the prevention of austenite grain coarsening during reheating period and retards the recrystallization temperature during rolling. Niobium also reduces the transformation temperature by solute drag effect (Singh et al., 1998),(Hulka, 2003). Figure 7c shows that the addition of 0.025 wt% Nb, improves tensile strength more than that of 0.04 wt%. For

instance in a steel with a carbon content of 0.15wt%, addition of 0.025% Nb increases tensile strength by 150 MPa.

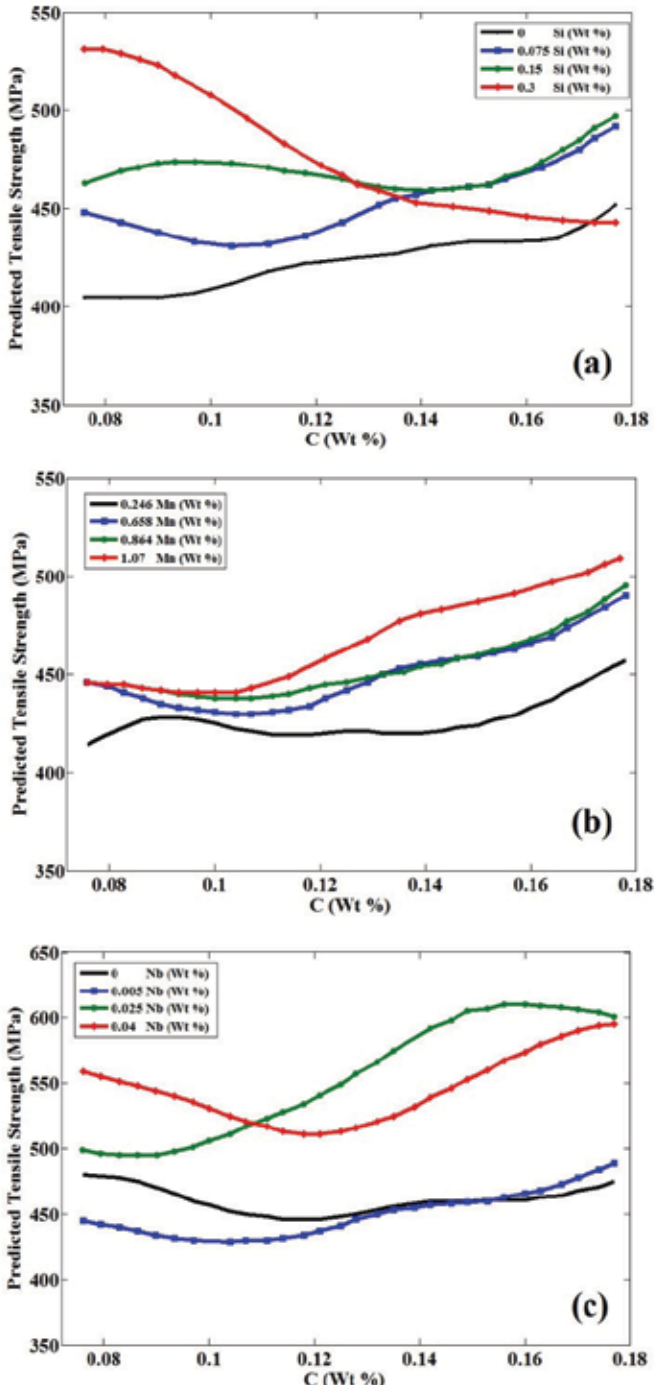


Fig. 7. Carbon concentration effect in combination with (a) Silicon (b) Manganese (c) Niobium.

Figure 8a, displays the effect of strip thickness versus manganese content on the final tensile strength. The results indicate a drop in tensile strength when final thickness is increased. This can be attributed to lower cooling rate of thicker strips. Therefore, coarsening takes place and the tensile strength decreases (Singh et al., 1998). This figure also illustrates the more influential effects of manganese on thinner strips. Figure 8b reveals the significance of finishing temperature verses the carbon concentration on tensile strength. It shows that by decreasing finishing temperature, the final tensile strength increases. Inter-pass recrystallization and grain growth prevention my causes this effect (Preloscan et al., 2002). The influence of temperatures on tensile strength is not significant when compared with that of chemical composition (in specified ranges) (Botlani-Esfahani et al., 2009b).

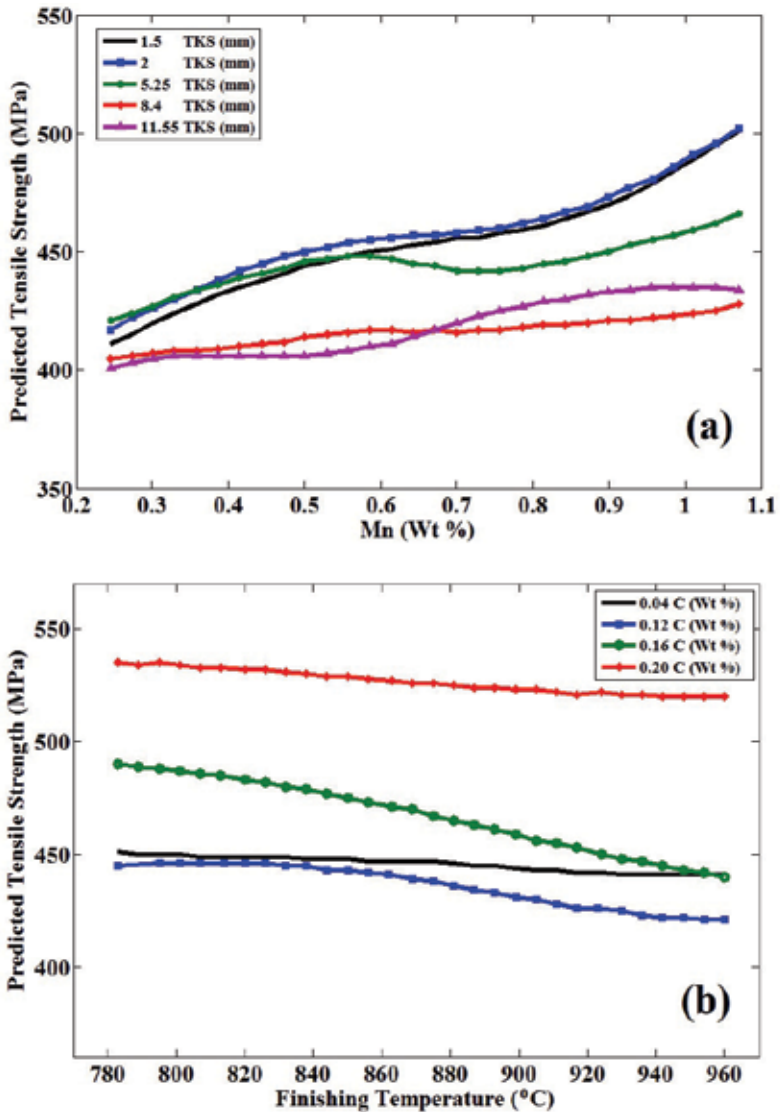


Fig. 8. Interaction of processing feature (a) Final thickness and manganese concentration, (b) Finishing temperature and carbon concentration.

3.4 Grain size model results

The result of this analysis indicates the importance of Si, Mn and C contents on grain refinement which is significantly greater than the concentration of other elements. The most effective element for grain refinement is recognized to be that of vanadium. However, its concentration in these steels is very low. For testing, the results of the model are depicted when the concentrations of elements are on their mean values which mentioned in Table 2 and the microalloying elements (i.e. Nb, Ti and V) are not present. Figure 9 shows the model result of this analysis. Manganese stabilizes austenite, therefore decreases austenite to ferrite transformation temperature and hence refines the grain structure. In addition, manganese

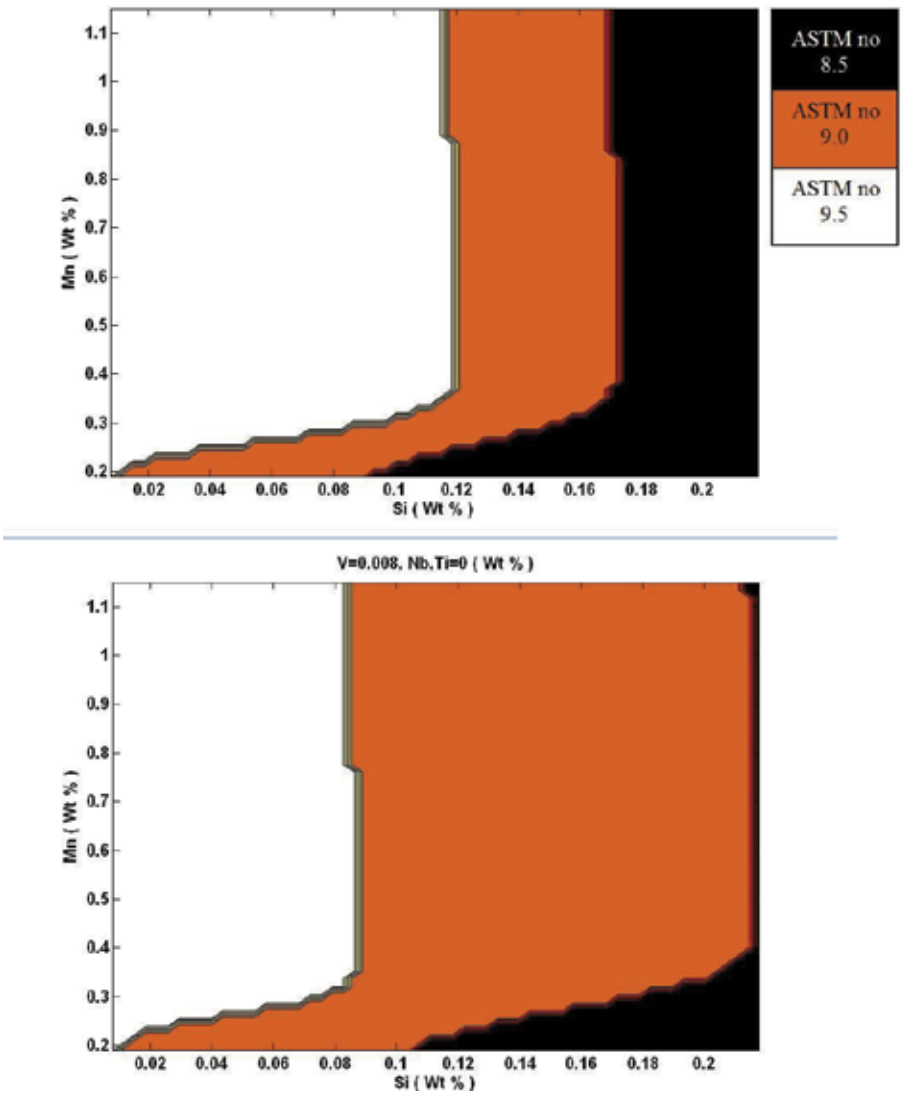


Fig. 9. Model result in respect of silicon and manganese concentration in 0.015 wt %C and 0.035 wt%Al. (a) Absence micro-alloying elements. (b) Minor addition of vanadium (0.008 wt %).

can enhance the precipitation strengthening of vanadium microalloyed steels and to a lesser extent, niobium microalloyed steels (keytosteel). Figure 9a reveals determining role of silicon on grain size in the absence of microalloying elements (i.e. Nb, Ti and V). The figure shows that silicon concentration divides the figure into three regions include finer, mild and coarser grain structures. This figure also indicates that increasing Si content, increases grain size. This is because silicon is a ferrite stabilizer and promotes ferrite grain growth (Umemoto et al., 2001). Figure 9b shows that addition of small amount of vanadium (0.008wt %) to steel severely contracts the coarser grain region. Vanadium acts as a scavenger for oxides, and forms nano-scale inter-phase precipitations. This is mainly due to the rapid rate of austenite to ferrite transformation which produces these nano-scale precipitates (Bhadeshia & Honeycombe, 2006). Furthermore, addition of vanadium also reduces the finer grain area somewhat. This is because, vanadium is strong carbide former and the majority of such elements is ferrite stabilizer and therefore, promotes ferrite grain growth (Zhang & Ren, 2003). The net effect of this minor vanadium addition is to decrease the sensitivity of grain size to silicon content, and also reduction of coarse grain area.

4. Conclusions

1. The effects of chemical composition and process variables on the tensile strength of hot strip mill products were modeled by Artificial Neural Network (ANN) moreover a Bayesian ANN model assisted by RJMCMC is capable of predicting the grain size of hot strip low carbon steels and can be used as a function of steel composition. The results of both models are shown to be consistent with experimental data (acquired from Mobarakeh Steel Company data).
2. The relative importance of each input variable was evaluated by sensitivity analysis for tensile strength. The influence of chemical composition on final tensile strength is much more pronounced than process parameters. Furthermore, grain size model recognizes the effects of relevant elements in grain refining. These are manganese, silicon and vanadium. Silicon concentration shows determining role this effect have not reported in the literature and vanadium reveals great impact on grain refining phenomena.
3. The results show the effects of the parameters are too complex to model with a simple linear regression technique. The developed ANN models can be used as guide to control the final mechanical properties of commercial carbon steel products. The major advantage of these methods is selection of useful inputs in complex problems with many inputs. Because many problems in materials science and engineering are similar, this method is useful for solving them.

5. References

Bhadeshia. H.K.D.H., Honeycombe. R.W.K. (2006) Steels Microstructure and Properties. 3rd ed., Elsevier, London, U.K, 57.

Bhadeshia. H.K.D.H., Lordand. M. Svensson. L.E. (2003) Silicon–Rich Bainitic Steel Welds Proc. of Int. Conf.: Joining & Welding Solutions to Industrial Problems, JWRI, Osaka University, Japan, 43-52.

Botlani-Esfahani. M, M. R. Toroghinejad and Key Yeganeh. A. R. (2009a) Modeling the Yield Strength of Hot Strip Low Carbon Steels by Artificial Neural Network. Materials and Design 30:9, 3653-3658

Botlani-Esfahani. M, Toroghinejad. M. R. and Abbasi. Sh. (2009b) Artificial Neural Network Modeling the Tensile Strength of Hot Strip Mill Products. ISIJ International 49:10, 1583-1587

Doan. C. D. and Yuiliong. S. (2004) Generalization for Multilayer Neural Network Bayesian Regularization or Early Stopping. Proc. of Asia Pacific Association of Hydrology and Water Resources 2nd Conference, APHW, Singapore, 1

Gonzalez. JEG. (2002) Study of the effect of hot rolling processing parameters on the variability of HSLA steels, Master thesis, University of Pittsburgh, USA

Hulka. K. (2003): Niobium Information, 17/98, http://www.cbmm.com.br

Keytosteel.com. Control of high strength low alloy (HSLA) steel properties. www. keytosteel.com

Lampinen. J. and Vehtari. A. (2001) Bayesian techniques for neural networks - review and case studies. In K. Wang, J Grundespenkis, and A. Yerofeyev, editors, Applied Computational Intelligence to Engineering and Business, 7-15.

MacKay DJC. (1992) A practical Bayesian framework for back-propagation networks. Neural Computation. 4, 415-47.

MathWorks,Inc.http://www.mathworks.com/access/helpdesk/help/pdf-doc/nnet/nnet.pdf, Nat-ick, MA, USA

MEYER, L (2001). History of Niobium as a microalloying element." In: Proceedings of the International Symposium Niobium 2001. Niobium Science and Technology. Niobium 2001 Ltd. Bridgeville: Pa, USA. 359-377

Preloscan. A., Vodopivec. F., Mamuzic. I. (2002) Fine-Grained Structural Steel with Controlled Hot Rolling. Materiali in Tehnologije, 36, 181.

Parker. S.V. (1997) Modeling phase transformation in hot-rolling steels. PhD Thesis, University of Cambridge, UK

Ryu. J. (2008). Model for mechanical properties of hot-rolled steels, Master thesis, Pohang University of Science and Technology, Korea

Singh. S. B., Bhadeshia. H. K. D. H, MacKay. D. J. C., Carey. H, and Martin. I. (1998) Neural Network Analysis of Steel Plate Processing. Ironmaking Steelmaking, 25, 355.

Umemoto. M., Liu. Z.G., Masuyama. K., Tsuchiya. K. (2001): Influence of Alloy Additions on Production and Propeties of Bulk Centite. Scripta. Materialia., 45, 39.

Zhang. Y. B., Ren. D.Y. (2003) Distribution of strong carbide forming elements in hard facing weld metal. Materials. Science and Technology., 19:8. 1029-103.

Vehtari. A., and Lampinen. J. (2002), Bayesian model assessment and comparison using cross-validation predictive densities, Neural Computation, 14, 2439.

Xu. M., Zeng. G., Xu. X., Huang. G., Jiang. R. and Sun. W. (2006) Application of Bayesian Regularized BP Neural Network Model for Trend Analysis, Acidity and Chemical Composition of Precipitation in North Carolina. Water, Air, and Soil Pollution, 172, 167.

8

Adaptive Neuro-Fuzzy Inference System Prediction of Calorific Value Based on the Analysis of U.S. Coals

F. Rafezi, E. Jorjani and Sh. Karimi
Science and Research Branch, Islamic Azad University, Tehran
Iran

1. Introduction

Coal is a chemically and physically heterogeneous and combustible substance that consists of both organic and inorganic compounds. It currently is a major energy source worldwide, especially among many developing countries, and will continue to be so for many years (Miller, 2005).The chemical analysis of coal includes proximate and ultimate analyses. The proximate analysis gives the relative amounts of moisture, volatile matter, and ash, as well as the fixed carbon content of the coal. The ultimate or elemental analysis gives the amounts of carbon, hydrogen, nitrogen, sulfur, and oxygen in the coal (Miller, 2005).

The measure of the amount of energy that a given quantity of coal will produce when burned is kown as calorific value or heating value. Heating value is a rank parameter and a complex function of the elemental composition of the coal, but it is also dependent on the maceral and mineral composition (Hower and Eble, 1996). It can be determined experimentally using a calorimeter.

Many equations have been developed for the estimation of gross calorific value (GCV) based on proximate analysis and/or ultimate analysis (Mason and Gandhi, 1983; Mesroghli et al., 2009; Given et al., 1986; Parikh et al., 2005; Custer, 1951; Spooner, 1951; Mazumdar, 1954; Channiwala and Parikh, 2002; Majumder et al., 2008).

Regression analyses and data for 775 U.S. coal samples (with less than 30% dry ash) were used by Mason and Gandhi (1983) to develop an empirical equation that estimates the calorific value (CV) of coal based on its C, H, S, and ash contents (all on dry basis). Their empirical equation, expressed in SI units, is:

$$CV = 0.472C + 1.48H + 0.193S + 0.107A - 12.29 \text{ (MJ/kg)} \quad (1)$$

Given et al. (1986) developed an equation to calculate the calorific value of U.S. coals from their elemental composition; expressed in SI units, their equation is:

$$CV = 0.3278C + 1.419H + 0.09257S - 0.1379O + 0.637 \text{ (MJ/Kg)} \quad (2)$$

Neural networks, as a new mathematical method, have been used extensively in research areas related to industrial processes (Zhenyu and Yongmo, 1996; Jorjani et al., 2007; Specht,

1991; Chen et al., 1991; Wasserman, 1993; Chehreh Chelgani et al., 2008; Hansen and Meservy, 1996; Patel et al., 2007; Mesroghli et al., 2009; Bagherieh et al., 2008; Jorjani et al., 2008; Chehreh Chelgani et al., 2010; Khandelwal and Singh, 2010 ; Sahu et al., 2010; Yao et al., 2005; Patel et al., 2007; Salehfar and Benson, 1998; Wu et al., 2008; Karacan, 2007).

Patel et al. (2007) predicted the GCV of coal utilizing 79 sets of data using neural network analyses based on proximate analysis, ultimate analysis, and the density of helium. They found that the input set of moisture, ash, volatile matter, fixed carbon, carbon, hydrogen, sulfur, and nitrogen yielded the best prediction and generalization accuracy.

Mesroghli et al. (2009) investigated the relationships of ultimate analysis and proximate analysis with GCV of U.S. coal samples by regression analysis and artificial neural network methods. The input set of C, $H_{\text{exclusive of moisture}}$ (H_{ex}) , N, $O_{\text{exclusive of moisture}}$ (O_{ex}), S, moisture, and ash was found to be the best predictor.

The adaptive neuro-fuzzy inference system (ANFIS), which consists of both artificial neural networks and fuzzy logic, has been used widely in research areas related to industrial processes (Boyacioglu and Avci, 2010; Esen and Inalli, 2010; Soltani et al., 2010; Pena et al., 2010; Chong-lin et al., 2009).

The aim of the present work is to assess the properties of 4540 samples of U.S. coal from 25 states with reference to the GCV and possible variations with respect to ultimate and proximate analyses using multi-variable regression, the SPSS software package, and the ANFIS, MATLAB software package.

This work is an attempt to answer the following important questions:

a. Is it possible to generate precise linear or non-linear equations between ultimate and proximate analysis parameters and GCV for different U.S. coal samples that have a wide range of calorific values from 4.82 to 34.85 MJ/kg?
b. Is ANFIS a better tool than regression analysis for improving accuracy and decreasing errors in the estimation of the calorific value of coal?
c. Is it possible to improve the accuracy of predictions by changing "total hydrogen and oxygen in coal (H and O)" to "H_{ex}, O_{ex}, and moisture?"

This work is different from previously published work because it involves the first use of ANFIS to predict the GCV of coal.

2. Experimental data

The data that were used to examine the proposed approaches were obtained from the U.S. Geological Survey Coal Quality (COALQUAL) database, open file report 97-134 (Bragg et al., 2009). Samples with more than 50% ash and samples that had a proximate analysis and/or an ultimate analysis different from 100% were excluded from the database.

Analysis results for a total of 4540 coal samples were used.

The sampling procedures and chemical analytical methods are available at the following website: http://energy.er.usgs.gov/products/databases/CoalQual/index.htm. The number of samples and the range of GCV for different states are shown in Table 1.

Table 2 shows the ranges of input variables, i.e., C, H, H_{ex}, N, O, O_{ex}, total sulfur, ash, moisture, and volatile matter, that were used in predicting GCV.

State	Number of samples	Range of GCV (MJ/kg)
Alabama	679	6.05-34.80
Alaska	51	8.65-27.42
Arizona	10	18.54-24.36
Arkansas	52	5.57-34.68
Colorado	172	7.24-33.81
Georgia	25	24.03-34.85
Indiana	101	19.23-28.96
Iowa	73	16.03-26.59
Kansas	19	20.87-28.86
Kentucky	720	18.68-34.03
Maryland	40	23.04-33.48
Missouri	68	23.83-28.63
Montana	140	5.55-20.63
New Mexico	114	8.81-32.15
North Dakota	124	4.85-13.61
Ohio	398	16.43-31.14
Oklahoma	25	23.89-33.31
Pennsylvania	498	13.58-33.10
Tennessee	42	24.61-33.48
Texas	33	9.54-27.74
Utah	103	4.82-30.14
Virginia	368	19.49-34.80
Washington	10	13.14-27.45
West Virginia	340	14.29-34.75
Wyoming	335	6.27-34.23

Table 1. Number of samples and range of GCV (as-received) for different U.S. states

Variable (%)	Minimum	Maximum	Mean	Std. Deviation
Moisture	0.4	49.60	8.90	9.90
Volatile matter	3.80	55.70	32.30	6.32
Ash	0.90	32.90	10.84	5.97
Hydrogen	1.70	8.10	5.27	0.69
Carbon	24.10	89.60	65.72	12.02
Nitrogen	0.20	2.41	1.29	0.33
Oxygen	0.90	54.70	14.86	11.27
Sulfur	0.07	17.30	1.90	1.73
H_{ex}	0.19	5.86	4.36	0.79
O_{ex}	0.09	22.14	7.50	3.27

Table 2. Ranges of proximate and ultimate analyses of coal samples (as-received)

3. Methods

3.1 Regression analysis

Regression nalysis is a statistical tool that is used to investigate the relationships between variables. Usually, the investigator seeks to ascertain the causal effect of one variable upon another. To explore such issues, the investigator assembles data on the underlying variables of interest and employs regression analysis to estimate the quantitative effect of the causal variables upon the variable that they influence. The investigator also typically assesses the statistical significance of the estimated relationships, that is, the degree of confidence that the true relationship is close to the estimated relationship (An introduction to regression analysis, Alan O. Sykes).

Linear regression estimates the coefficients of the linear equation, involving one or more independent variables, which are required to have a reliable prediction of the value of the dependent variable. All variables must pass the tolerance criterion to be entered in the equation, regardless of the entry method specified. The default tolerance level is 0.0001. Also, a variable is not entered if it would cause the tolerance of another variable already in the model to drop below the tolerance criterion. All independent variables selected are added to a single regression model. However, different entry methods can be specified for different subsets of variables. Method selection allows specifying how independent variables will be entered into the analysis. Using different methods, a variety of regression models can be selected from the same set of variables (SPSS Inc., 2004).

Non-linear regression is a method of finding a non-linear model of the relationship between the dependent variable and a set of independent variables. Unlike traditional linear regression, which is restricted to estimating linear models, non-linear regression can estimate models with arbitrary relationships between independent and dependent variables. This is accomplished using iterative estimation algorithms (SPSS Inc., 2004).

In this study, both single-variable and multi-variable regressions were used to develop correlations between ultimate and proximate analyses of coal samples with their gross calorific value (GCV). A stepwise procedure for selecting variables was used, and the variables were entered sequentially into the model. The first variable considered for use in the equation was the one with the largest positive or negative correlation with the dependent variable. This variable was entered into the equation only if it satisfied the criterion for entry. The next variable, with the largest partial correlation, was considered as the second input to the equation. The procedure stops when there are no variables that meet the entry criterion (SPSS Inc., 2004).

3.2 Adaptive neuro fuzzy inference system

In the artificial intelligence field, the term "neuro-fuzzy" refers to combinations of artificial neural networks and fuzzy logic. Fuzzy modeling and neural networks have been recognized as powerful tools that can facilitate the effective development of models and integrate information from different sources, such as empirical models, physical laws, or measurements and heuristics (Babuska, 1998); these two tools were combined in order to achieve readability and learning ability at the same time (Jantzen, 1998). The neuro-fuzzy approach in the fuzzy modeling research field is divided into two areas: 1) linguistic fuzzy modeling that is focused on interpretability, mainly the Mamdani model and 2) precise fuzzy modeling that is focused on accuracy, mainly the Takagi-Sugeno-Kang (TSK) model (Wikimedia Foundation Inc., 2009). ANFIS is an architecture that is functionally equivalent to a Takagi-Sugeno-Kang-type fuzzy

rule base (Jang & Sun, 1995); it is a class of adaptive, multi-layer, feed-forward networks that is functionally equivalent to a fuzzy inference system.
A fuzzy rule in a Sugeno fuzzy model has the form of:

$$\text{If x is A and y is B then } z = f(x, y)\ , \tag{3}$$

where A and B are input fuzzy sets in the antecedent, and, usually, z = f(x, y) is a zero- or first-order polynomial function in the consequent. The fuzzy reasoning procedure for the first-order Sugeno fuzzy model and equivalent ANFIS structure is shown in Fig. 1.
Here, the defuzzification procedure in the Mamdani fuzzy model is replaced by the operation of the weighted average in order to avoid the time-consuming procedure of defuzzification. Defuzzification refers to the way a crisp value is extracted from a fuzzy set as a representative value (Jang and Sun, 1995).
Jang and Sun (1995) and Jantzen (1998) have provided more details about the ANFIS architecture, learning algorithms, and training methods.

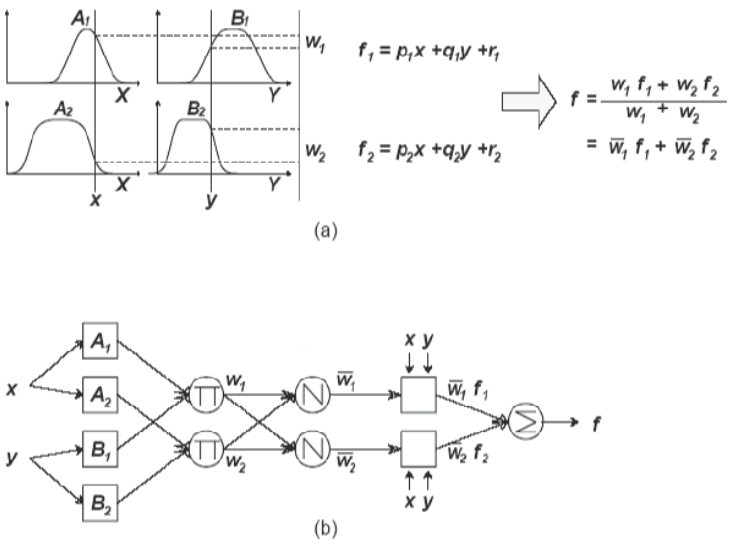


Fig. 1. (a) The Sugeno fuzzy model reasoning; (b) equivalent ANFIS structure (Jang and Sun, 1995)

4. Results and discussion

4.1 Relationships between GCV and individual input variables

By a least squares mathematical method, the correlation coefficients (R^2) of C, H, H_{ex}, N, O, O_{ex}, total sulfur, ash, moisture, and volatile matter with GCV were determined to be +0.99, -0.25, +0.72, +0.52, -0.86, -0.51, +0.01, -0.05, -0.85, and +0.03, respectively. From the above-mentioned results, it can be concluded that the worthy relationships are for carbon with positive effect and oxygen with negative effect, because they are rank parameters; and moisture with negative effect, because it is also a rank parameter at low rank coals and because it is a diluent with respect to heating value. Non-linear relationships between individual input variables and GCV were examined as well, but the results were not better than the results obtained when the linear procedure was used.

4.2 Multi-variable relationships of GCV with ultimate and proximate analysis parameters

The best-correlated linear equations, using a stepwise procedure between the various mentioned parameters and GCV, can be presented as follows:

a. Ash, moisture, and volatile matter inputs:

$$GCV\ (MJ/kg) = 37.777 - 0.647M - 0.387A - 0.089VM \qquad R^2 = 0.97 \qquad (4)$$

b. Carbon, hydrogen, nitrogen, oxygen, sulfur, and ash inputs:

$$GCV\ (MJ/kg) = 5.833 + 0.284C - 0.321O + 1.031H + 0.519N - 0.046Ash$$

$$R^2 = 0.994 \qquad (5)$$

c. Carbon, hydrogen exclusive of moisture, nitrogen, oxygen exclusive of moisture, sulfur, moisture, and ash inputs:

$$GCV\ (MJ/kg) = 26.452 + 0.074C - 0.405M + 0.89H_{ex} - 0.446\ O_{ex} - 0.256Ash - 0.195S$$

$$R^2 = 0.995 \qquad (6)$$

Estimated deviations of GCV from target values for equations (4) through (6) are shown in Table 3.

GCV deviation from target (MJ/kg)	Eq. (4)	Eq. (5)	Eq. (6)
Less than 0.5	39.4%	71.7%	78.2%
Less than 1	72.5%	95.2%	96.5%
More than 1	27.2%	4.8%	3.5%

Table 3. Estimated deviations of GVC from target values for various linear regression equations

The non-linear equations were examined as well, and the exponential equation was the best predictor of GCV. The results for the input sets of (a), (b), and (c) are shown in the following equations:

a. Ash, moisture, and volatile matter inputs:

$$GCV = 182.667 + 37.564e^{-0.027M} - 0.381e^{0.042VM} - 182.79e^{0.002A} \qquad R^2 = 0.988 \qquad (7)$$

b. Carbon, hydrogen, nitrogen, oxygen, sulfur, and ash inputs:

$$GCV = -156.641 - 0.091e^{-0.073A} + 60.15e^{0.004C} - 13.95e^{-0.322H} + 0.33e^{0.648N} + 109.885^{-0.003O} - 0.318\ e^{-0.363S}$$

$$R^2 = 0.995 \qquad (8)$$

c. Carbon, hydrogen exclusive of moisture, nitrogen, oxygen exclusive of moisture, sulfur, moisture, and ash inputs:

$$GCV = -278.474 + 4.487e0.016C + 24.485e\text{-}0.019M + 7.173e0.013N + 76.532e0.012Hex +$$

$$189.349e\text{-}0.001Oex - 0.033e0.221S - 4.727e0.021A \qquad R^2 = 0.999 \qquad (9)$$

The estimation of GCV deviations from target values for equations (7) through (9) are shown in Table 4. By comparing Tables 3 and 4, it can be concluded that exponential equations are more precise than linear equations for predicting the GCV of coal.

GCV deviation from target (MJ/kg)	Eq. (7)	Eq. (8)	Eq. (9)
Less than 0.5	60%	28.98%	74.8%
Less than 1	86.65%	71.34%	99.1%
More than 1	13.35%	28.66%	0.9%

Table 4. Estimation of the deviations of GCV from target values for various non-linear regression equations

4.3 ANFIS prediction

Three input sets, (a), (b) and (c), were used to determine whether ANFIS is able to predict GCV better than regression. This was done using the ANFIS menu in the MATLAB software package to identify the relationships between GCV and input variables.

In a neuro-fuzzy inference system, the first step is to determine the system inputs and outputs that will be used to predict GCV. In this study, input set (a) was comprised of three variables, i.e., ash, volatile matter, and moisture; input set (b) was comprised of six variables, i.e., C, H, N, O, S, and ash; input set (c) was comprised of seven variables, i.e., C, H_{ex}, N, O_{ex}, S, ash, and moisture.

The Sugeno fuzzy inference system was used in this research. The output functions in the Sugeno system are linear or constant. A rule in the fuzzy Sugeno model is:

$$\text{If input 1 = x and input 2 = y, then the output is } z = ax + by + c \tag{10}$$

In the Sugeno system, for a zero-order model, the z plane is constant (a = b = 0). The plane of z_i, the output of any rule, is weighted by w_i. The final output of the system is the weighted average of all outputs, which is calculated as follows:

$$\text{final output} = \frac{\sum_{i=1}^{N} w_i z_i}{\sum_{i=1}^{N} w_i} \tag{11}$$

The subtractive clustering scheme was used to cluster data; the best-designed, neuro-fuzzy system for input sets (a), (b), and (c) were systems with three, five, and twelve clusters, respectively. For input set (a), the range of influence, squash factor, accept ratio, and reject ratio were selected as 0.5, 1.25, 0.5, and 0.15, respectively; for input set (b), they were 0.35, 1.25, 0.5, and 0.15, respectively; and, for input set (c), they were 0.25, 1.2, 0.5, and 0.125, respectively. The Gaussian membership function was used. For training of the ANFIS, the hybrid method was used with 3200 sets of data; the remaining 1340 sets of data were used

Model	Basis	Model inputs	Training set size	Testing set size	Number of membership functions	R^2
a	As received	Ash, volatile matter, moisture	3200	1340	3	0.997
b	As received	C, H, N, O, S, ash	3200	1340	5	0.999
c	As received	C,H_{ex}, N, O_{ex}, S, ash, moisture	3200	1340	12	0.999

Table 5. Details of the best-correlated neuro-fuzzy models

for testing. For the training stage, we selected 100 epochs. Details of the best-correlated neuro-fuzzy models are shown in Table 5. As Table 5 shows, the designed neuro-fuzzy systems can predict the GCV with acceptable correlation coefficients (R^2) of 0.997 , 0.999, and 0.999 for the (a), (b), and (c) input sets, respectively.
As an example, the neuro-fuzzy design structure for model (c) to predict GCV is shown in Fig. 2.
The estimates of the deviations of the GCV from target values produced by the neuro-fuzzy models are shown in Table 6. It can be seen that the prediction precision of GCV from ANFIS and using all three input sets (a), (b), and (c) (Table 6) are better than those from linear and non- linear regression (Tables 3 and 4).

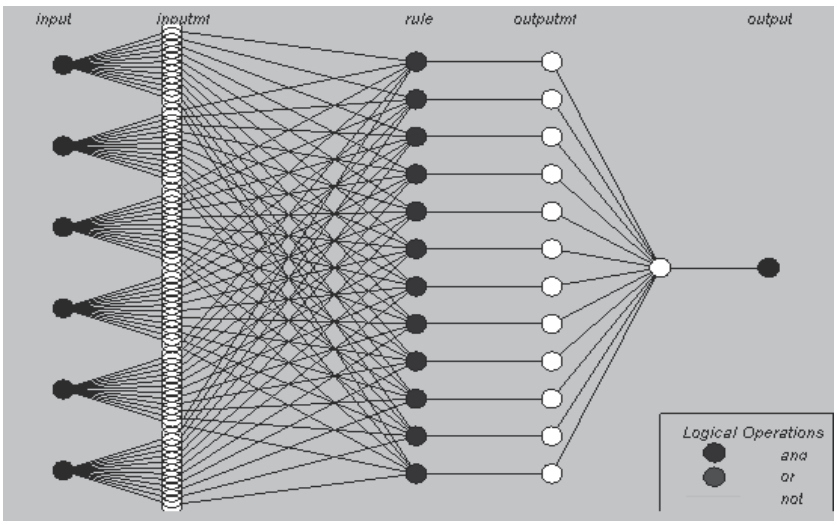


Fig. 2. ANFIS model structure for the prediction of GCV using input set (c)

GCV deviation from target (MJ/kg)	Model a (3-member function)	Model b (5-member function)	Model c (12-member function)
Less than 0.5	83%	97.6%	99.4%
Less than 1	99.4%	100%	100%
More than 1	0.5%	0%	0%

Table 6. Estimation of deviations of GCV from target values for neuro-fuzzy models

The GCV predicted (GCV_P) by ANFIS in the testing stage for input sets (a), (b), and (c) compared to the actual values determined in the laboratory (GCV_a) are shown in Figs. 3, 4, and 5, respectively. The distributions of the differences between actual and estimated GCVs are shown in Figs. 6, 7, and 8 for input sets (a), (b), and (c), respectively.

5. Technical considerations

According to Eqs. (4) through (9) and the results presented in Tables 3 and 4, it can be seen that the exponential equations are better than linear equations for predicting GCV; among the exponential equations, Eq (9) is the most suitable equation. A correlation coefficient of 0.999 and a deviation from experimentally calculated GCVs that was only 0.9 % more than

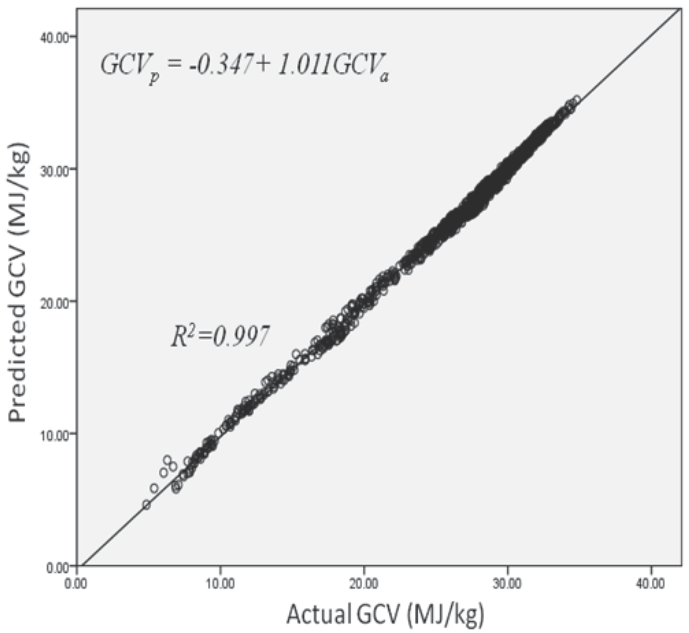


Fig. 3. ANFIS-estimated GCV in testing stage versus actual determined value (model a)

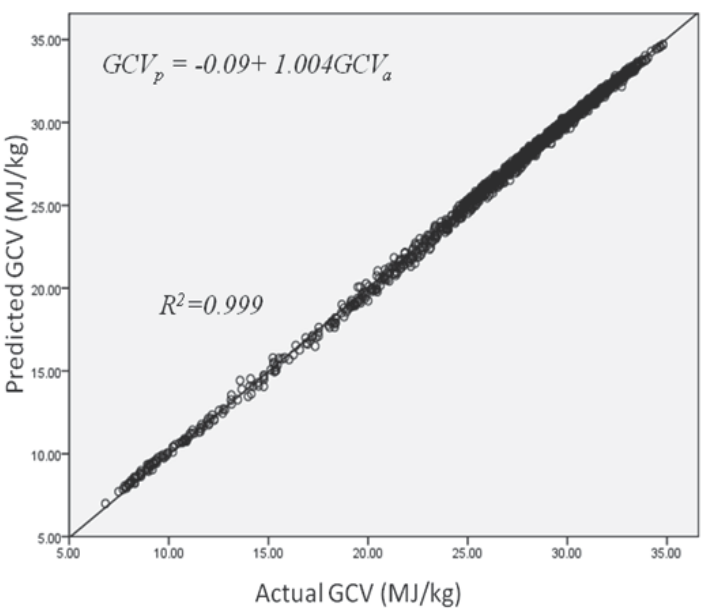


Fig. 4. ANFIS-estimated GCV in testing stage versus actual determined value (model b)

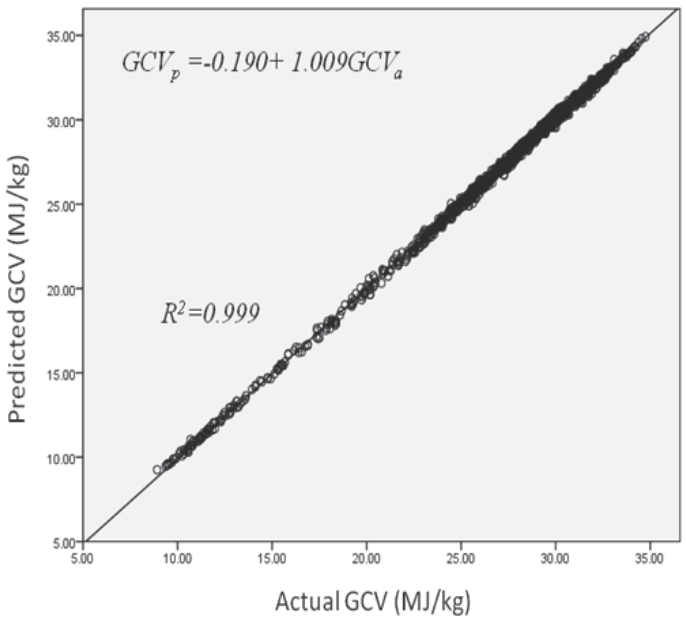


Fig. 5. ANFIS-estimated GCV in testing stage versus actual determined value (model c)

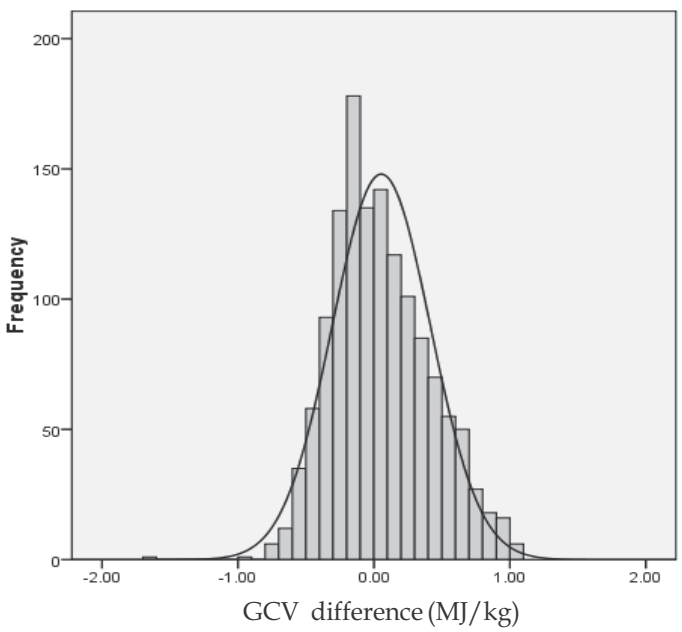


Fig. 6. Distribution of difference between actual and estimated GCV in testing stage (model a)

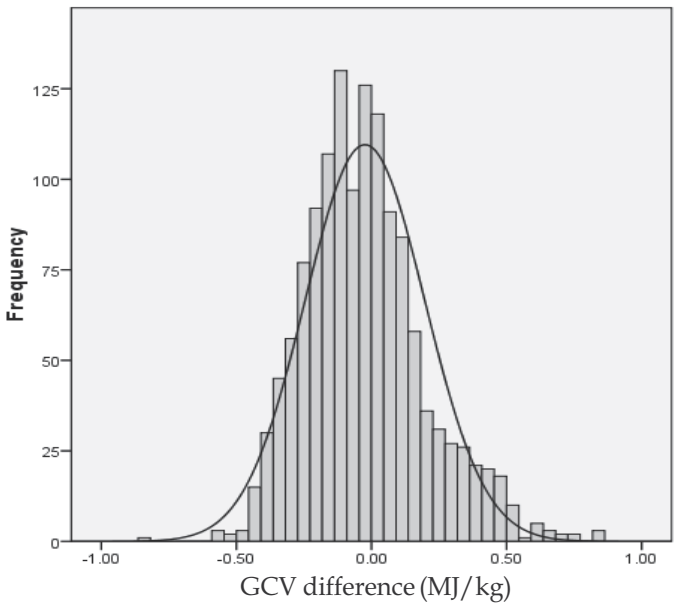


Fig. 7. Distribution of difference between actual and estimated GCV in testing stage (model b)

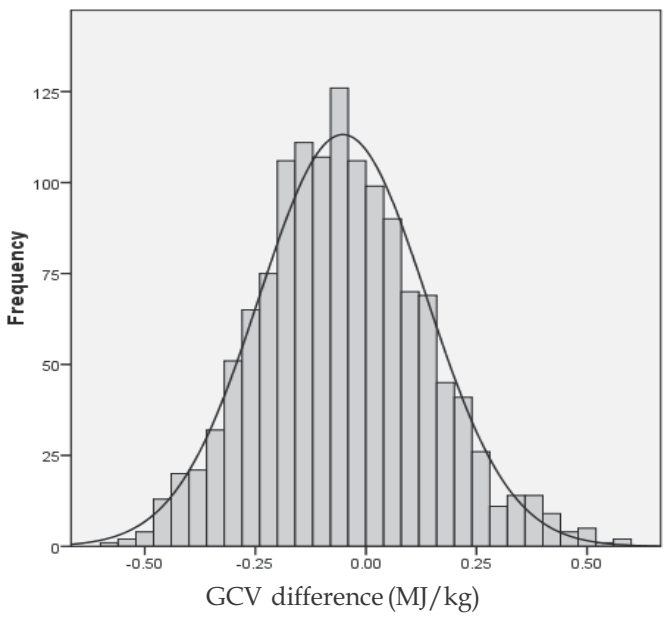


Fig. 8. Distribution of difference between actual and estimated GCV in testing stage (model c)

0.5 (MJ/kg) were achieved by Eq (9). With reference to the above results, it can be concluded that the input set of carbon, hydrogen exclusive of moisture, nitrogen, oxygen exclusive of moisture, sulfur, moisture, and ash can be used as the best and most-reliable input for the

prediction of the GCV of coal using exponential equations. Restating "hydrogen and oxygen" in the form of "hydrogen exclusive of moisture, oxygen exclusive of moisture, and moisture" can decrease the errors and deviations from experimentally calculated GCV by regression. According to Table 5, which presents the correlation coefficients of the ANFIS models for the (a), (b), and (c) input sets, the correlation coefficients in the test stage were determined ot be 0.997 (model a) to 0.999 (models b and c). In addition, Table 6, which presents the deviations of the ANFIS model predictions from targets values, shows that the errors and deviations from experimentally calculated GCVs in ANFIS models are less than those produced by regression models. Although Mesroghli et al. (2009) reported that artificial neural network is not better or very different from regression results when the proximate and ultimate analyses are the GCV predictors. However, in the current work, a suitable, structured ANFIS model predicted GCV with a high precision that has not been reported in previous published works.

6. Conclusions

- In this work, proximate and ultimate analyses of 4540 coal samples from 25 U.S. states and two mathematical modelling methods, i.e., multi-variable regression and adaptive neuro-fuzzy interface systems were used to estimate GCV.
  - The best-correlated linear equation was achieved using input set (c) (C, H_{ex}, N, O_{ex}, S, M, ash) with a correlation coefficient of 0.995. The results also showed that, for input set (c), the difference between actual and predicted values of GCV in about 78% of the data was less than 0.5 MJ/kg, and, in 96% of the data, the difference was less than 1 MJ/kg.
  - Exponential equations provided improved correlation coefficients in comparison to linear equations. The best result was achieved using input set (c) with a correlation coefficient of 0.999. The difference between actual and predicted values of GCV in about 75% of the data was less than 0.5 MJ/kg, and, in 99% of the data, the difference was less than 1 MJ/kg.
  - The neuro-fuzzy modeling system improved prediction accuracy for input sets (a), (b), and (c).
  - The neuro-fuzzy rules that were designed using 3, 5, and 12 membership functions can predict the GCV with R^2 = 0.997, 0.999, and 0.999, respectively. They also produced a deviation from target values of less than 0.5 MJ/kg for about 83, 97, and 99% of data, respectively, and less than 1 MJ/kg for about 99, 100, and 100% of data for input sets (a), (b), and (c), respectively.
  - The GCV prediction precision achieved in the current work using neuro-fuzzy systems has not been reported previously in the literature.

7. References

Babuska, R. (1998). *Fuzzy modeling for control*, Kluwer Academic Publisher, Boston.

Boyacioglu, M.A. & Avci, D. (2010). An Adaptive Network-Based Fuzzy Inference System (ANFIS) for the prediction of stock market return: The case of the Istanbul Stock Exchange, *Expert Systems with Applications*, Volume 37, Issue 12, 7908-7912.

Bagherieh, A.H. ; Hower, J.C. ; Bagherieh, A.R. & Jorjani, E. (2008). Studies of the relationship between petrography and grindability for Kentucky coals using

artificial neural network . *International Journal of Coal Geology*, Volume 73, Issue 2, 130-138.

Bragg, L.J. ; Oman, J.K. ; Tewalt, S.J. ; Oman, C.J. ; Rega, N.H. ; Washington, P.M. & Finkelman, R.B. (2009). U.S. *Geological Survey Coal Quality (COALQUAL) database version 2.0. open-file report 97-134*, http://energy.er.usgs.gov/products/databases/CoalQual/index.htm.

Chong-lin, W. ; Cao-yuan, M. ; Jian-hua, L. ; Guo-xin, L. ; Dong-liang, Z. & Jie-jie, T. (2009). Study on coal face stray current safety early warning based on ANFIS, *Procedia Earth and Planetary Science*, Volume 1, Issue 1, 1332-1336.

Chen, S. ; Cowan, C.F.N. & Grant, P.M. (1991). Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Trans. Neural Networks,* 2 (2), 302–309.

Custer, V.F. (1951). Uber die Berechnung des Heizwertes von Kohlen der Immediatzusammensetzung. Brennst.-Chem, 32, 19–20.

Channiwala, S.A. & Parikh, P.P. (2002). A unified correlation for estimating HHV of solid, Liquid and gaseous fuels. *Fuel,* 81, 1051–1063.

Chehreh Chelgani, S. ; Hower, J.C. ; Jorjani, E. ; Mesroghli, Sh. & Bagherieh, A.H. (2008). Prediction of coal grindability based on petrography, proximate and ultimate analysis using multiple regression and artificial neural network models, *Fuel Processing Technology*, Volume 89, Issue 1, 13-20.

Chehreh Chelgani, S. ; Mesroghli, Sh. & Hower, J.C. (2010). Simultaneous prediction of coal rank parameters based on ultimate analysis using regression and artificial neural network. *International Journal of Coal Geology*, Volume 83, Issue 1, 31-34.

Esen, H. & Inalli, M. (2010). ANN and ANFIS models for performance evaluation of a vertical ground source heat pump system , *Expert Systems with Applications*, Volume 37, Issue 12, 8134-8147.

Given, P.H. ; Weldon, D. & Zoeller, J.H. (1986). Calculation of calorific values of coals from ultimate analyses: theoretical basis and geochemical implications. *Fuel,* 65, 849–854.

Hower, J.C. & Eble, C.F. (1996). Coal quality and coal utilization. *Energy Miner. Div*. Hourglass 30 (7), 1–8.

Hansen, J.V. & Meservy, R.D. (1996). Learning experiments with genetic optimization of a generalized regression neural network. *Decis. Support Syst*., 18 (3–4), 317–325.

Jorjani, E., Chehreh Chelgani, S. & Mesroghli, Sh. (2007). Prediction of microbial desulfurization of coal using artificial neural networks , *Minerals Engineering*, Volume 20, Issue 14, 1285-1292.

Jorjani, E. ; Mesroghli, Sh. & Chehreh Chelgani, S. (2008). Prediction of operational parameters effect on coal flotation using artificial neural network. *Journal of University of Science and Technology Beijing, Mineral, Metallurgy, Material,* Volume 15, Issue 5, 528-533.

Jantzen J. (1998). *Neurofuzzy modelling*, Technical University of Denmark, Department of Automation, Tech. report no 98-H-874, 1-28.

Jang, J.S.R. & Sun, C.T. (1995). Neuro-fuzzy modeling and control, *Proceedings of the IEEE*, 83(3): 378–406.

Khandelwal, M. & Singh, T.N. (2010). Prediction of macerals contents of Indian coals from proximate and ultimate analyses using artificial neural networks, *Fuel,* Volume 89, Issue 5, 1101-1109.

Karacan, C.O. (2007). Development and application of reservoir models and artificial neural networks for optimizing ventilation air requirements in development mining of coal seams, *International Journal of Coal Geology,* Volume 72, Issues 3-4, 221-239.

Mason, D.M. & Gandhi, K.N. (1983). Formulas for calculating the calorific value of coal and chars. *Fuel Process. Technol*. 7, 11–22.

Miller, B.G. (2005). *Coal Energy Systems*, Elsevier Academic Press, ISBN: 0-12-497451-1, USA.

Mazumdar, B.K. (1954). Coal systematics: deductions from proximate analysis of coal part I. *J. Sci. Ind. Res*., 13B (12), 857–863.

Majumder, A.K. ; Jain, R. ; Banerjee, J.P. & Barnwal, J.P. (2008). Development of a new proximate analysis based correlation to predict calorific value of coal. *Fuel,* 87, 3077–3081.

Mesroghli, Sh. ; Jorjani, E. & Chehreh Chelgani, S. (2009). Estimation of gross calorific value based on coal analysis using regression and artificial neural networks. *International Journal of Coal Geology*, 79, 49–54.

Patel, S.U. ; Kumar, B.J. ; Badhe, Y.P. ; Sharma, B.K. ; Saha, S. ; Biswas, S. ; Chaudhury, A. ; Tambe, S.S. & Kulkarni, B.D. (2007). Estimation of gross calorific value of coals using artificial neural. *Fuel*, Volume 86, Issue 3, 334-344.

Pena, B. ; Teruel, E. & Diez, L.I. (2010). Soft-computing models for soot-blowing optimization in coal-fired utility boilers, *Applied Soft Computing*, In Press, Corrected Proof.

Parikh, J. ; Channiwala, S.A. & Ghosal, G.K. (2005). A correlation for calculating HHV from proximate analysis of solid fuels. *Fuel*, 84, 487–494.

Soltani, F. ; Kerachian, R. & Shirangi, E. (2010). Developing operating rules for reservoirs considering the water quality issues: Application of ANFIS-based surrogate models, *Expert Systems with Applications*, Volume 37, Issue 9, 6639-6645.

Specht, D.F. (1991). A generalized regression neural network. *IEEE Trans. Neural Netw.,* 2(5), 568–576.

Salehfar, H. & Benson, S.A. (1998). Electric utility coal quality analysis using artificial neural network techniques, *Neurocomputing*, Volume 23, Issues 1-3, 195-206.

Spooner, C.E. (1951). Swelling power of coal. Fuel, 30, 193–202.

SPSS. (2004). Help Files, Version 13, SPSS Inc.

Wu, Q.; Ye, S. & Yu, J. (2008). The prediction of size-limited structures in a coal mine using Artificial Neural Networks. *International Journal of Rock Mechanics and Mining Sciences,* Volume 45, Issue 6, 999-1006.

Wasserman, P.D. (1993). *Advanced methods in neural computing*. Van Nostrand Reinhold, New York, 155–161.

Zhenyu, Z. & Yongmo, X. (1996). *Introduction to fuzzy theory, neural networks, and their applications.* Beijing/Nanning: Tsinghua University Press/Guangxi Science and Technology Press, in Chinese.

Sahu, H.B. ; Padhee, S. & Mahapatra, S.S. (2010). Prediction of spontaneous heating susceptibility of Indian coals using fuzzy logic and artificial neural network models, *Expert Systems with Applications*, In Press, Uncorrected Proof.

9

Artificial Neural Network Applied for Detecting the Saturation Level in the Magnetic Core of a Welding Transformer

Klemen Deželak[1], Gorazd Štumberger[1], Drago Dolinar[1] and Beno Klopčič[2]
[1]*University of Maribor, Faculty of Electrical Engineering and Computer Science*
[2]*Indramat elektromotorji d. o. o.*
Slovenia

1. Introduction

This chapter deals with the detector of saturation level in the magnetic (iron) core of a welding transformer. It is based on an artificial neural network (ANN) and requires only the measurement of the transformer's primary current. The saturation level detector could be the substantial component of a middle frequency resistance spot welding system (RSWS), where the welding current and the flux density in the welding transformer's iron core are closed-loop controlled by two hysteresis controllers. The resistance spot welding systems, described in different realizations (Brown, 1987), are widely used in the automotive industry. Although the alternating or direct currents (DC) can be used for welding, this chapter focuses on the resistance spot welding system (Fig. 1) with DC welding current. The resistances of the two secondary windings R_2, R_3 and characteristics of the rectifier diodes, connected to these windings, can slightly differ. Reference (Klopčič et al., 2008) shows that combination of these small differences can result in increased DC component in welding transformer's iron core flux density. It causes increasing iron core saturation with the high impact on the transformer's primary current i_1, where currents spikes eventually appear, leading to the over-current protection switch-off of the entire system. However, the problematic current spikes can be prevented either passively or actively (Klopčič et al., 2008). When the current spikes are prevented actively, closed-loop control of the welding current and iron core flux density is required (Klopčič et al., 2008). Thus, the welding current and the iron core flux density must be measured. While the welding current is normally measured by the Rogowski coil (Ramboz, 1996), the iron core flux density can be measured by the Hall sensor or by a probe coil wound around the iron core. In the case, the flux density value is obtained by the analogue integration of the voltage induced in the probe coil (Deželak et al., 2008). Integration of the induced voltage can be unreliable due to the unknown integration constant in the form of the remanent flux and the drift in analogue electronic components. The drift can be kept under control by the use of closed-loop compensated analogue integrator.

An advanced, the two hysteresis controllers based control of the RSWS, where the current spikes are prevented actively by the closed-loop control of the welding current and flux

density in the welding transformer's iron core, is presented in (Klopčič et al., 2008). The modified solution requires measuring of the welding current, while instead of measured flux density only information about saturation level in the iron core is required (Deželak et al., 2010). Some methods, tested on welding transformer's iron core, that can be applied for saturation level detection are presented in (Deželak et al., 2008). All these methods require the Hall sensor or probe coils which make them less interesting for applications in the industrial RSWS, due to the relatively high sensitivity on vibrations, the mechanical stresses and the high temperatures. In order to overcome these problems, an ANN based iron core saturation level detector is introduced in this work. Additionally the method proposed for the detecting saturation level of the complete loaded RSWS, completed by ANN, is presented. Its only (single) input is the measured transformer's primary current.
The ANN, applied in the iron core saturation level detector, is trained to recognize the waveform of the current spikes, which appear in the primary current when the iron core is approaching the saturated region. Before the ANN can be applied, its structure must be defined first, and then the ANN must be trained using an appropriate learning method (Pihler et al., 1997). In this paper, the ANN structure appropriate for saturation detection in the transformer's iron core and the appropriate learning method are found with the help of properly build dynamic model of the RSWS (Deželak et al., 2010). The mentioned dynamic model includes models of the hysteresis controllers and the ANN based saturation level detector. The well-known trial and error method was used for defining ANN structure. It is shown that the three-layer ANN with 30 neurons in the first layer, 7 neurons in the second layer, and 1 neuron in the third layer, gives acceptable results. ANN is trained by the resilient backpropagation rule, where the measured and calculated samples of transformer's primary current, with different known levels of saturation in the iron core, are used. The calculated and measured results, presented in this paper, show that the proposed ANN based iron core saturation level detector can be used as a part of the discussed RSWS, improving performances of the entire system

2. Dynamic model of the resistance spot welding system

The resistance spot welding system consists of a full wave output rectifier, a single phase transformer, an H-bridge inverter and an input rectifier. It is shown in Fig. 1 and described in (Klopčič et al., 2008). The three-phase alternating current (AC) voltages u_1, u_2 and u_3, supplied from the electric grid, are rectified in the input rectifier in order to produce the DC bus voltage. This voltage is used in the H-bridge inverter, where different switching patterns and modulation techniques can be applied, to generate AC voltage u_H, required for supply of the welding transformer. The welding transformer has one primary and two secondary windings. They are marked with indices 1, 2 and 3, respectively. The currents, the number of turns, the resistances and the leakage inductances of the primary and two secondary welding transformer's windings are denoted by i_1, i_2, i_3, N_1, N_2, N_3, R_1, R_2, R_3, and $L_{\sigma 1}$, $L_{\sigma 2}$, $L_{\sigma 3}$. The effects of the eddy current losses are accounted for by the resistor R_{Fe}, while R_L and L_L are the resistance and inductance of the load. The output rectifier diodes D_1 and D_2 are connected to both transformer's secondary coils. They generate the DC welding current i_L which has a DC value a few times higher than the amplitudes of AC currents i_2 and i_3 that appear in the transformer's secondary coils without rectifier diodes.

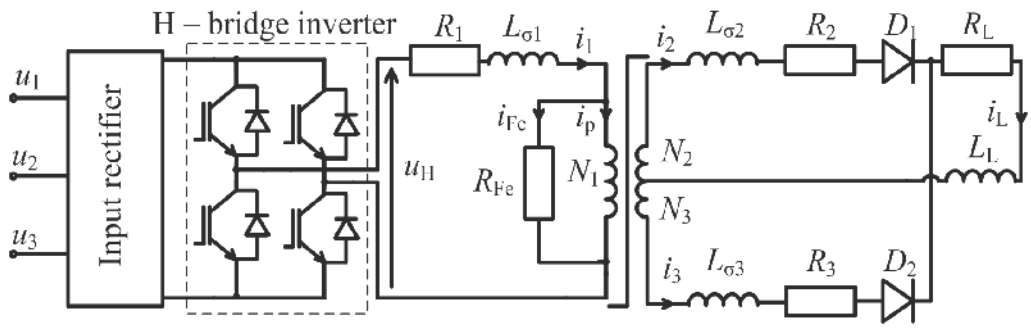


Fig. 1. The resistance spot welding system

The supply voltage of the primary coil of the transformer could be generated on the different ways (Štumberger et al., 2010). In the existent system, widely spread in the automotive industry, this voltage is generated by the H-bridge inverter applying pulse width modulation (PWM) at switching frequency of f = 1 kHz. The PWM principle is shown in Fig. 2a, where u_t is the triangular voltage, U_{ref} is the reference voltage for PWM, T_p is the period of H-bridge inverter output voltage, u_m is the gate driver input voltage, S_1, S_4 and S_2, S_3 are the pairs of IGBT-s in the H-bridge inverter (Sabate et al., 1990).
Additionally Fig. 2b shows the AC voltage generated by the H-bridge applied by PWM, where U_{DC} is the DC-bus voltage.

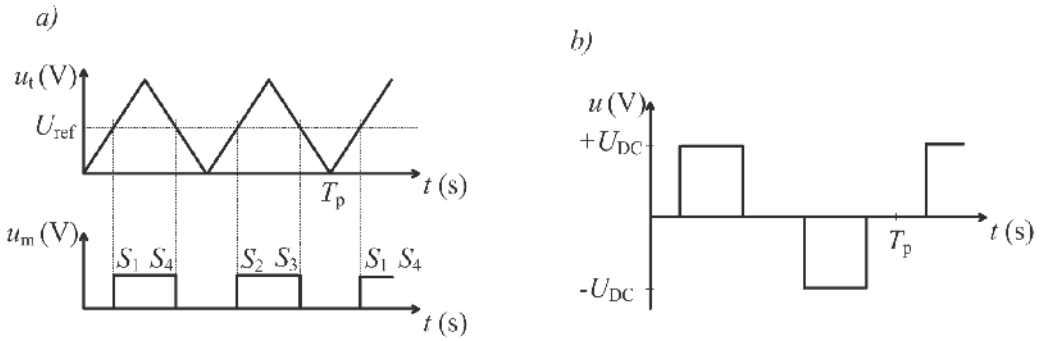


Fig. 2. The PWM principle (a) and the AC voltage generated by the H-bridge applied by PWM (b)

As references (Klopčič et al., 2008) and (Deželak et al., 2010) show, the resistances of the secondary windings R_2, R_3 and the characteristics of the rectifier diodes could be slightly different. These differences can cause DC component in welding transformer's iron core flux density, which causes increasing iron core saturation with the essential impact on the transformer's primary current i_1, where currents spikes appear, leading to the over-current protection switch-off of the entire system.
Aforementioned phenomena could be confirmed by the appropriate dynamic model (Leon & Semlyen, 1994) of the complete resistance spot welding system. In this work the model is built with the programme package Matlab/Simulink based on the following set of equations (1) – (8).

$$u_H = R_1 i_1 + L_{\sigma 1}(di_1/dt) + N_1(d\phi/dt) \quad (1)$$

$$0 = R_2 i_2 + L_{\sigma 2}(di_2/dt) + N_2(d\phi/dt) + dip_1 + R_L i_L + L_L(d(i_2 + i_3)/dt) \quad (2)$$

$$0 = R_3 i_3 + L_{\sigma 3}(di_3/dt) - N_3(d\phi/dt) + dip_2 + R_L i_L + L_L(d(i_2 + i_3)/dt) \quad (3)$$

$$N_1 i_p + N_2 i_2 - N_3 i_3 = H(B) l_{ic} + 2\delta B/\mu_0 \quad (4)$$

$$i_L = i_2 + i_3 \quad (5)$$

$$i_1 = i_{Fe} + i_p \quad (6)$$

$$i_{Fe} = N_1(d\phi/dt)/R_{Fe} \quad (7)$$

$$\phi = BA_{Fe} \quad (8)$$

$$\theta = N_1 i_1 + N_2 i_2 - N_3 i_3 \quad (8)$$

In set of equations (1) – (8) ϕ stands for magnetic flux, dip_1 and dip_2 are nonlinear characteristics of the output rectifier diodes $D1$ and D_2, $H(B)$ is the magnetizing curve of the iron core material, δ is the air gap, B is the iron core flux density, μ_0 is the permeability of the vacuum, l_{ic} is the average length of the magnetic flux line in the iron core, A_{Fe} is the cross-section of the transformer's iron core and θ is the magnetomotive force. Parameters that appear in (1) – (8) are shown in Table 1 and in Table 2.

Parameter	*Value*	*Unit*
A_{Fe}	0.001385	m^2
δ	10	μm
l_{ic}	0.313	m
f	1000	Hz
R_1	0.01357	Ω
R_2	20	μΩ
R_3	20	μΩ
R_L	100	μΩ
$L_{\sigma 1}$	0.035	mH
$L_{\sigma 2}$	1	nH
$L_{\sigma 3}$	1	nH
L_L	1	μH
N_1	55	/
N_2	1	/
N_3	1	/

Table 1. Parameters of RSWS dynamic model

dip_1 - i (A)	dip_1 - u (V)	dip_2 - i (A)	dip_2 - u (V)
0	0	0	0
0.003	0.6	0.011	0.6
0.014	0.65	0.053	0.65
0.059	0.7	0.25	0.7
0.247	0.75	1.17	0.75
1.05	0.8	5.52	0.8
4.43	0.85	25.9	0.85
18.75	0.9	121.5	0.9
79.3	0.95	570	0.95
335	1	2675	1
1418	1.05	12555	1.05
6000	1.1	58912	1.1
25378	1.15	400416	1.15
107334	1.2	1297043	1.2

Table 2. Nonlinear characteristics of the output rectifier diodes D_1 - (dip_1) and D_2 - (dip_2)

With the appropriate dynamic model the two behaviours of RSWS, the symmetrical and asymmetrical, could be simulated. Firstly, the symmetrical behaviour is considered by parameters shown in Table 3, while obtained results are shown in Fig. 3. The resistances R_2 and R_3 in the two secondary welding transformer's windings are equal, as well the characteristics of the output rectifier diodes D_1 and D_2. Fig. 3 shows the time dependent primary current i_1 and the magnetic flux density B in the time window of t = 0.1s.

Parameter	*Value*	*Unit*
R_2	20	μΩ
R_3	20	μΩ
D_1	characteristic - dip_1	/
D_2	characteristic - dip_1	/

Table 3. Parameters for symmetrical behaviour of the resistance spot welding system

Different resistances R_2 and R_3 and different characteristics of the output rectifier diodes D_1 and D_2 could cause undesired asymmetry of the spot welding system. In case of considering values shown in Table 4, the asymmetrical time dependent magnetic flux density B could be obtained by the appropriate model of RSWS. In this way, when the magnetic flux density B reaches the saturation level the current spikes appear in the primary current i_1, as shown in Fig. 4.

Parameter	*Value*	*Unit*
R_2	20	μΩ
R_3	15	μΩ
D_1	characteristic - dip_1	/
D_2	characteristic - dip_2	/

Table 4. Parameters for asymmetrical behaviour of the resistance spot welding system

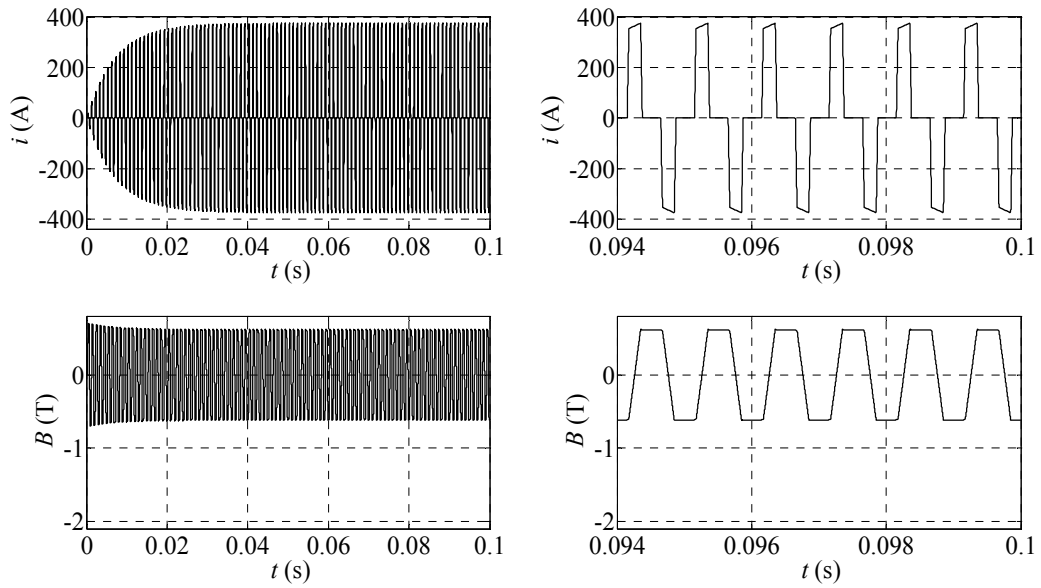


Fig. 3. Symmetrical behaviour of the resistance spot welding system

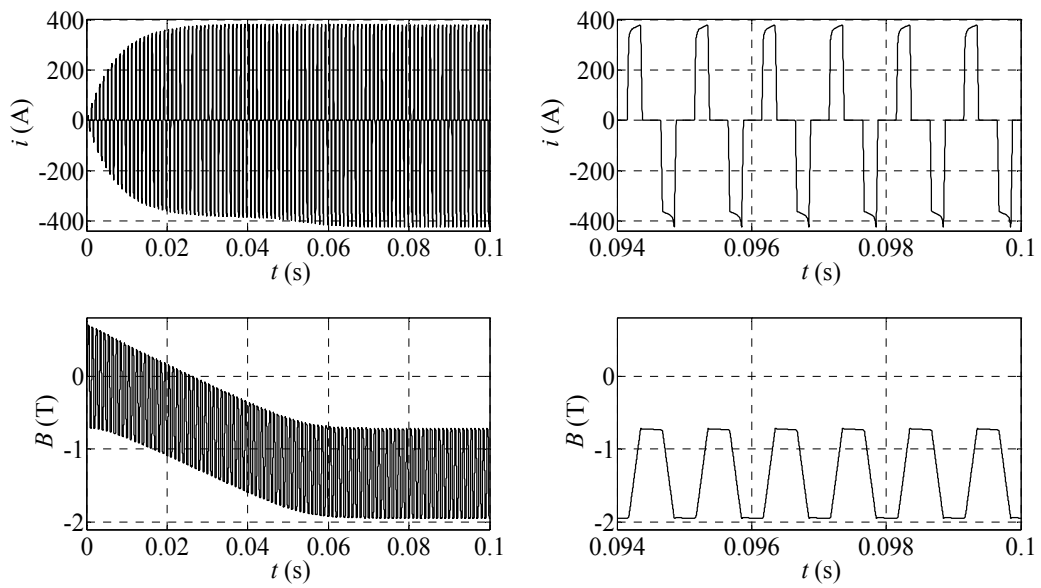


Fig. 4. Asymmetrical behaviour of the resistance spot welding system

As Fig. 4 shows, the iron core becomes more and more saturated, which leads to currents spikes in the primary current i_1 and finally to the over-current protection switch-off. The unwanted current spikes can be prevented passively by using pairs of rectifier diodes with matched characteristics, or actively (Klopčič et al., 2008) by controlling the saturation level in the iron core. In the letter case, a saturation detector, which generates a signal when the preset saturation level is reached, is indispensable for preventing the iron core saturation.

This work evaluates the methods appropriate for the detecting saturation level in the stand alone transformer's iron core and in the transformer operating in the resistance spot welding system. These methods actually detect the instant when the iron core starts to become saturated and generate signals which are used in the control algorithm to prevent iron core saturation. All of the presented methods are based on the ANN which is applied as an additionally tool for the detection of transformer's core saturation.

3. The detecting saturation level in the transformer's iron core

In this section, the saturation level detection of the stand alone transformer's iron core, is described (Deželak et al., 2008). The iron core of a welding transformer, which is normally installed in an industrial resistance spot welding system, is shown in Fig. 5. For test purposes, the actual primary and secondary windings were replaced with only one primary coil, which was able to produce the same magnetomotive force as the primary and secondary winding, together. In Fig. 5, u denotes the primary voltage, i_1 is the primary current, δ is the length of the air gap, while A_{Fe} is the cross-section of the iron core. A measurement coil is wound around the iron core for measurement purposes. The primary and measurement coils have the same number of turns N.

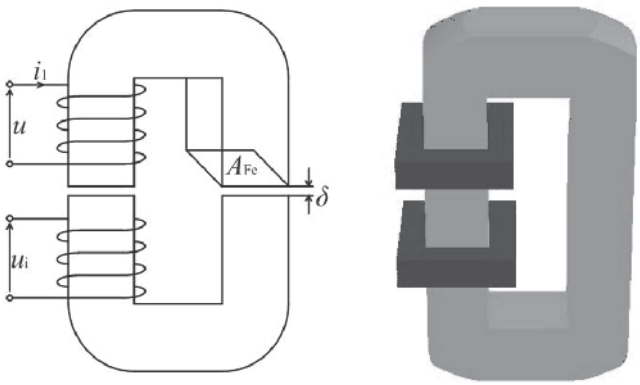


Fig. 5. The iron core of a welding transformer

The proposed method is based on calculation of dynamic inductance L_d (9), where u is the measured induced voltage (10) and i is the measured transformer's primary current. Fig. 6 shows the dynamic inductance as a function of primary current i. The dynamic inductance is defined by (11), where $\psi(i)$ is the magnetically nonlinear iron core characteristic shown in Fig. 6. In the given case, the magnetically nonlinear characteristic of the welding transformer's iron core $\psi(i)$ was determined experimentally using numerical integration (12).

$$L_d = u/(di/dt) \tag{9}$$

$$u = d\psi/dt \tag{10}$$

$$L_d = \partial\psi/\partial I \tag{11}$$

$$\psi(t) = \psi(0) + \int (u(\tau) - R_1\, i(\tau))d\tau \tag{12}$$

In (12) $\psi(t)$ is the time behaviour of the flux linkage, $i_1(t)$ and $u(t)$ are the measured primary current and voltage, while R_1 stands for the resistance of the primary winding.

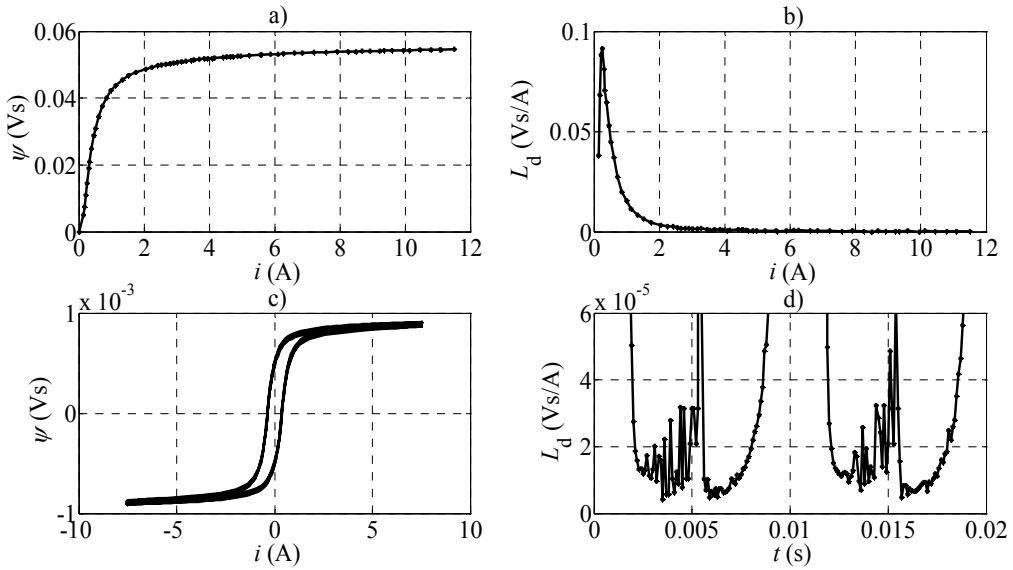


Fig. 6. The dynamic inductance and magnetically nonlinear iron core characteristic

When the value of dynamic inductance $L_d(i)$, shown in Fig. 6b, drops under the value of $L_d(i)$ = 0.0003 Vs/A, which is reached at i = 8 A, the iron core can be considered as saturated. However, the signal that represents dynamic inductivity $L_d(i)$ calculated by (9) is contaminated with noise, as shown in Fig. 6d. The contamination with the noise is substantially increased in the vicinity of the reversal points of the hysteresis, which makes reliable iron core saturation level detection almost impossible.

This problem can be effectively solved by supplementing the calculated values of the dynamic inductivity in the vicinity of the reversal points of the hysteresis by a signal generated by an artificial neural network. The artificial neural network is a parallel multi layer information processing structure, with possibility to include expert knowledge into existent process. Fig. 7 shows the three-layer artificial neural network, where x_1, x_2, … stand for input parameters or signals, $w(x_1)_1$, $w(x_2)_2$, $w(x_1)_2$, $w(x_2)_2$, …, w_{56} are the weight coefficients, 1, 2, 3, …, 6 are the sum blocks, while tan_{sig} and lin stand for the sigmoid and linear activation functions. Additionally z_1, z_2, z_3, …, z are the output signals of the sum blocks, while y_1, y_2, y_3, ..., y are output signals of the neurons (Hassoun, 1995). The number of neurons used in the three-layer artificial neural network shown in Fig. 7 is six - three in the first layer, 2 in the second layer and 1 in the output layer. The neural network can be supplemented with the bias vector for the each individual neuron. The artificial neural networks accumulate the knowledge during the training process, while the effectiveness of the artificial neural network depends on the quality of the training procedure. The fundamental aim of the training procedure is to adjust all weights in artificial neural network to obtain minimal deviations between the target and calculated outputs (Hoyong et al., 1993).

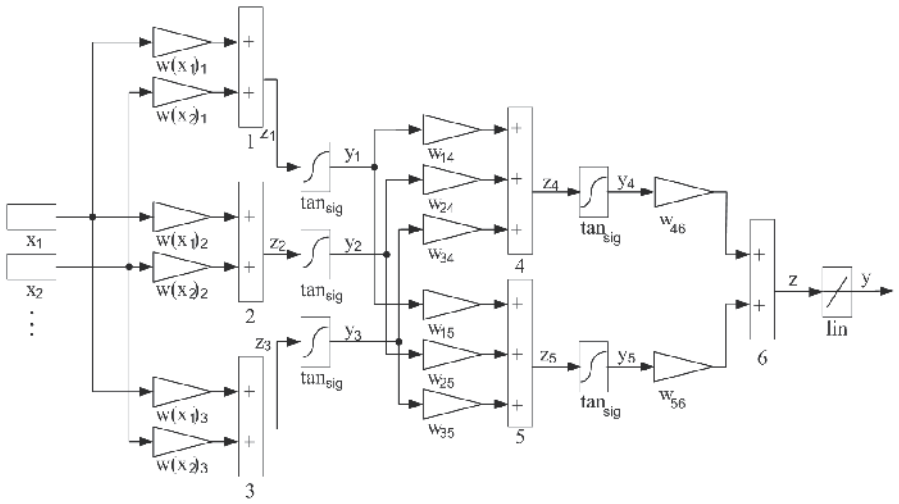


Fig. 7. The example of the three layers artificial neural network

In this chapter the error backpropagation method is applied. For that reason the characteristic input patterns must be selected, while the target signal is generated with respect to the input patterns. Training with the error backpropagation learning rule consists of the initialization of all weights (and bias) with randomly selected initial values and calculations of all outputs signals from each neuron. As soon as the output value of the last neuron in the output layer is calculated, the squared error for this (last) neuron can be calculated and then errors for the rest of neurons, from the output layer towards the input layer can be defined too. Finally, when errors of each neuron are obtained, the new values of all weights (and bias) can be calculated and the entire procedure can start with the new iteration. The number of iterations of described procedure is called the number of epochs. The iterations stop when the error reaches predefined value or the maximal number of epochs (iterations) is reached. In the given case, the learning signal was build of 170 patterns (signals) obtained by measurements, while the target signal was defined afterwards and was set to the values one (saturated) or zero (not saturated). Fig. 8 shows five of these patterns. In the case when the value of the target signal (*Tar.*) equals one, the iron core is considered as saturated. After extensive testing of different net configurations, the final artificial neural network configuration was defined. It contains three-layer with 50 neurons in the first layer, 8 neurons in the second - hidden layer, and with one neuron in the third - output layer.

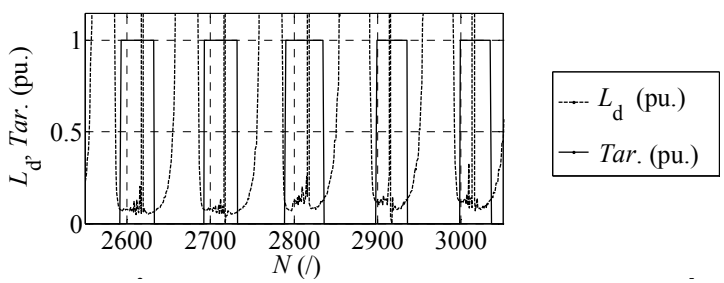


Fig. 8. The learning and targets signals in the case of the saturation level detection of the transformer's iron core

Once the artificial neural network is trained, which means that all of the weights and bias are set, it can be tested. First with the patterns used in the training procedure (Fig. 9 left) and then with the new samples which were not used in the training procedure (Fig. 9 right). Results presented in Fig. 9 show that the artificial neural network trained in this way is appropriate for saturation level detection in the transformer's iron core. However, the results of extended analysis showed that the proposed method gives unreliable results when the level of iron core saturation further increases. Fig. 10 shows the results obtained by the artificial neural network for the case when transformer's iron core was highly saturated. Fig. 10 shows output signal of artificial neural network before (*Out*) and after (*Out'*) the final bias value (*Out'* = 0 means iron core is not saturated, *Out'* = 1 means iron core is saturated). According to the artificial neural network output signals the highly saturated iron core is not saturated at all.

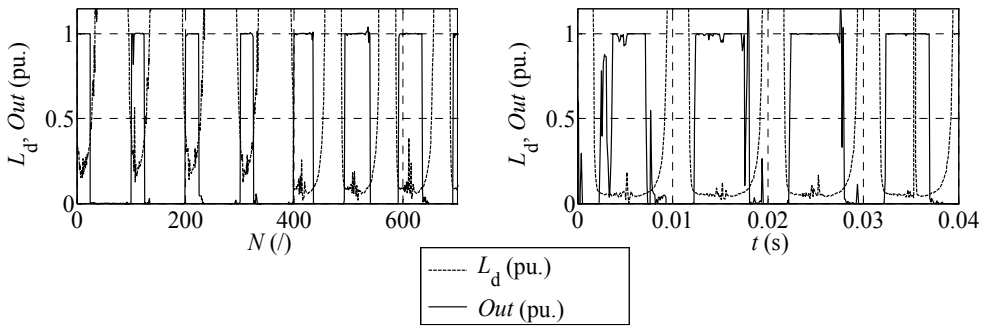


Fig. 9. Testing of the ANN with the patterns used in the training procedure (left) and with the new samples which were not used in the training procedure (right)

The results presented show that the artificial neural network is not reliable enough to be used for iron core saturation level detection as a stand alone algorithm. However, it could be very useful as a supplement to the existing algorithms for iron core saturation detection which fails when approaching reversal point on the hysteresis loop. In this region the artificial neural network can provide reliable information that the system is approaching reversal point of the hysteresis loop, which can be used to stabilize existing algorithms for iron core saturation level detection.

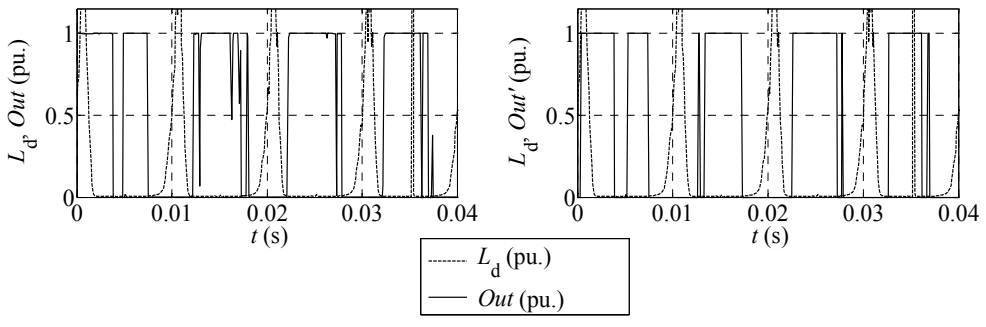


Fig. 10. Testing of the ANN with the patterns of the highly saturated transformer's iron core before *Out* (left) and after *Out'* (right) the final bias value

4. The detecting saturation level in the resistance spot welding system

Now the algorithm for detecting the saturation level in the iron core of the transformer operating in the resistance spot welding system can be presented. One of the possible solutions for active prevention of the current spikes, is the closed-loop control of the iron core flux density and welding current with two hysteresis controllers (Klopčič et al., 2008). Thus, the iron core flux density and the welding current must be measured. The iron core flux density can be measured by the Hall sensor or by a probe coil wound around the iron core, while the welding current is normally measured by the Rogowski coil. The flux density value is obtained by an analogue integration of the voltage induced in the probe coil. It is well known that the integration of the induced voltage could be unreliable due to the drift in analogue electronic components and the unknown integration constant in the form of remanent flux. Because of the mentioned drawback, this work proposes an improved solution. Instead to measure the flux density only the information about the saturation level in the welding transformer's iron core is required (Fig. 11). In Fig. 11 S_1 – S_4 stand for the adequate switching of the transistors.

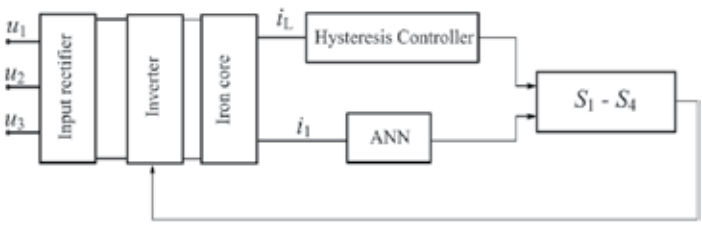


Fig. 11. The closed-loop control of the transformer primary current i_1 and welding current i_L using the hysteresis controller and the ANN

To obtain the information about the saturation level an ANN can be applied as an effective tool, where the ANN single input is the measured transformer's primary current i_1 (Deželak et al., 2010). The idea is to replace one hysteresis controller with ANN. Basically the ANN is trained to recognize the waveform of the current spikes, which appear in the primary current when the iron core is approaching the saturated region.

4.1 Hysteresis control with saturation detector

The H – bridge inverter output voltage u_H (Fig. 1) is equal to DC voltage, while its polarity depends on the pair of transistors that are switched on. When all four transistors are switched off, the voltage u_H equals zero. The welding current i_L increases when the primary voltage of the transformer u_H differs from zero. On the other hand the welding current tends towards zero when u_H equals zero. The magnetic flux density increases when +DC bus voltage is applied and decreases when–DC bus voltage is applied. As soon as the magnetic flux density exceeds its limit, the transformer's iron core becomes highly saturated, which causes current spikes in the transformer's primary current i_1. The advanced control of the RSWS can be applied to prevent the current spikes. The authors in (Klopčič et al., 2008) proposed an advanced hysteresis control of the RSWS based on two hysteresis controllers. The first one is used for the closed-loop control of the welding current while the second one is used for the closed-loop control of a flux density in the transformer's iron core. The advanced hysteresis control requires information that the preset saturation level in the iron core is exceeded. Measurement of the iron core flux density B can be avoided.

This subsection proposes an ANN based detector of saturation level in the iron core. The only input for the proposed detector is the measured welding transformer's primary current i_1. The output signal of the ANN based saturation detector (*Tar.*) is set to the value one when the preset saturation level in transformer's iron core is exceeded. Operation of the advanced hysteresis control, supplemented by the proposed ANN based saturation level detector, is illustrated in Fig. 12, where i_L is the welding current, u_H is the applied H-bridge inverter output voltage, and t stands for the time. The upper (i_{Lu}) and lower (i_{Ll}) welding current limits must be set before start.

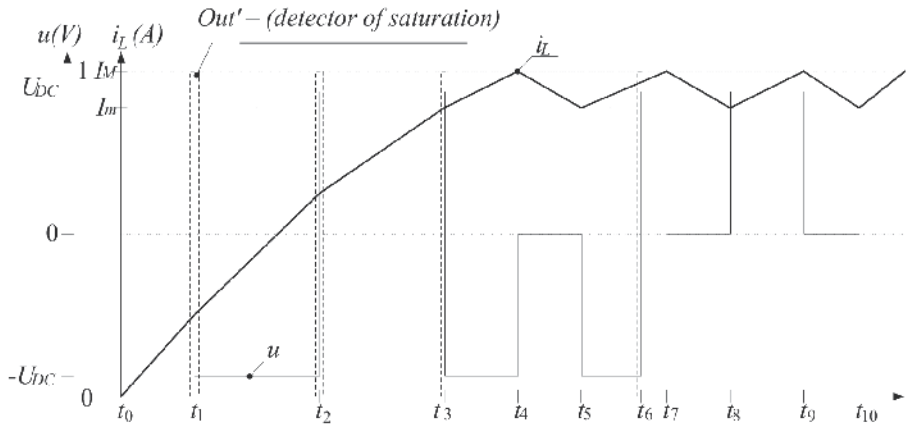


Fig. 12. Operation of the advanced hysteresis control, supplemented by the saturation level detector

The advanced hysteresis control of the RSWS, supplemented by the proposed saturation level detector, starts at the time t = 0s. The H-bridge inverter generated transformer's supply voltage $u_H = U_{DC}$, is applied. The welding current i_L and the iron core flux density B start to increase. At the time t_1, the ANN based saturation level detector detects that the preset saturation level in the iron core is exceeded. Its output signal *Out'* is set to the value 1, while the transformer's supply voltage is changed to $u_H = -U_{DC}$. The welding current still increases while the iron core flux density starts to decrease. At time t_2, the saturation level detector detects the exceeded saturation level again. In this case, it is caused by a high negative value of the iron core flux density. The transformer's supply voltage changes to $u_H = U_{DC}$. The welding current still increases while the flux density value starts to increase again.

Thus, whenever the ANN based saturation level detector detects the exceeded saturation level, the polarity of the transformer's supply voltage changes, causing change in the sign of the flux density increment, while the welding current increases all the time. The advanced RSWS hysteresis control switches the transformer's supplied voltage according to the described pattern as long as the welding current does not reach its upper limit, which happens at the time t_4. When the upper limit for the welding current is reached, the supply voltage $u_H = 0$ is applied. The welding current starts to decrease. At the time t_5, it reaches its lower limit. The supply voltage with the same polarity as before the time t_4 ($u_H = -U_{DC}$) is applied again. At the time t_6, the saturation level detector detects the increased saturation level again, which causes change in the polarity of the transformer's supply voltage $u_H = U_{DC}$. The RSWS operates as described, keeping the welding current between its lower and upper limits, until the welding cycle is completed. The transformer's supply voltage is set to

u_H = 0 while the welding current decreases to value zero. The advanced hysteresis control controlled RSWS is ready for the new welding cycle.

4.2 Test result

The ANN structure used to detect the iron core saturation in the stand alone transformer cannot be applied when the transformer operates as a part of the RSWS. In the case of the stand alone transformer the ANN was used to supplement the calculated value of dynamic inductance L_d in order to evaluate the iron core saturation level. In the case of transformer operating as a part of RSWS the primary current i_1 supplemented by an ANN is used to detect the iron core saturation level. Thus the new ANN structure and learning method must be defined. This can be obtained with the proper dynamic model of the RSWS with included models of the advanced hysteresis control and the ANN based saturation level detector. Once the model is built, the proper structure of the ANN and learning method can be easily defined by running simulations with different ANN structures. The trial and error procedure was applied in the testing. As already mentioned, the ANN accumulates knowledge during the learning process. Fig. 13 shows both signals, which are involved in the learning process.

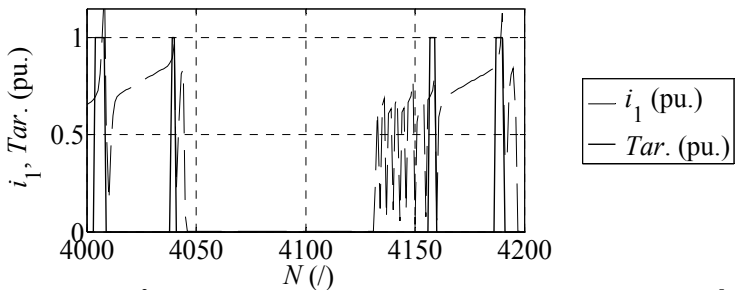


Fig. 13. The learning and targets signals in the case when the transformer operates as a part of the RSWS

The input learning signal (pattern) of the ANN in the learning process is the absolute value of the transformer's primary current i_1 (1 pu = 400 A) obtained with the RSWS dynamic model. According to the saturation level in the iron core, the output learning signal (*Tar.*) was set to the value zero or one. The current spikes in the transformer's primary current clearly show that the iron core becomes highly saturated. In that case, the output of the ANN based saturation level detector must be set to the value one, which changes the polarity of the transformer's supply voltage in the RSWS controlled by the advanced hysteresis control. Fig. 13 shows learning signals during samples 4000 and 4200, while the number of all samples is 6000.

The ANN output signals are very sensitive on the ANN net configuration, therefore an extensive testing of different net configurations was performed. The proper net structure can be defined with the proper model of a whole RSWS. The high computational effort required for simulations of the whole system forced us to apply the trial and error method in determining the ANN structure, instead of applying optimization techniques. The correlation coefficients between the target signal and calculated outputs were the root mean square errors (RMSE). The learning rates were controlled. Based on results of the extensive numerical analysis, the ANN structure with 30 neurons in first, 7 neurons in second and 1

neuron in last layer was chosen (RMSE = 3.89 %). More or less neurons in the first layer gave worse RMSE. In addition for determining the ANN structure, the model was applied also for determining the most appropriate learning rule. From all learning rules tested, the resilient and Levenberg-Marquardt backpropagation algorithms gave the best RMSE values. However, the resilient backpropagation was adopted due to the lowest computational effort required.

As soon as the structure of the ANN and learning rule is defined, they can be applied on to the measured signals, while the ANN trained with the measured signals can be applied as a saturation level detector in the RSWS controlled by the advanced hysteresis control. Fig. 14 shows the output signals from the ANN for two different transformer's primary currents (absolute, per unit value, 1 pu = 400 A) measured on the RSWS.

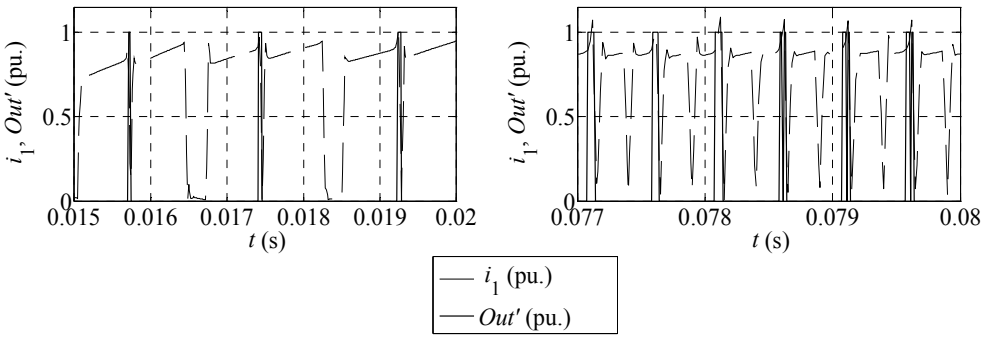


Fig. 14. Testing of the ANN with the different patterns

As soon as the ANN, through the characteristic form of the primary current, detects that the saturation level in the iron core is high enough, the value of the ANN output (*Out'*) changes, causing change in the polarity of the applied supply voltage. This leads to the change in the sign of the magnetic flux density derivative. The magnetic flux density move in the opposite direction until the ANN detects increased saturation level again. The polarity of the applied supply voltage is changed again and the complete procedure is repeated.

5. Conclusion

This chapter deals with an evaluation of method appropriate for the iron core saturation level detection in welding transformer built into resistance spot welding system. The welding transformer is a part of a resistance spot welding system, where current spikes in the transformers primary current, caused by the iron core saturation, can cause the over-current protection switches-off of the entire spot welding system. For that reason the saturation level detection is an indispensable part of modern resistance spot welding systems. It makes the control of the iron core flux possible, which leads to better iron core exploitation and prevents over-current protection switch-offs. The main aim of this work is to present a reliable method for detection of the iron core saturation that does not require any additional sensor.

Firstly, an artificial neural network supplemented algorithm for detecting the saturation level in the iron core of a welding transformer is described. In order to prevent iron core saturation and current spikes in the primary current, the iron core saturation level detection,

based on the dynamic inductances, is investigated. The drawbacks of the applied method could be effectively eliminated by using an artificial neural network as a supplement to the existing method.

After that the algorithm for detecting the saturation level in the iron core of the transformer operating as a part of the resistance spot welding system was presented. The proposed ANN based detector requires measurement of the welding transformer's primary current. The applied ANN is trained to recognize the waveform of the currents spikes which is used for saturation level detection. The applied ANN contains 3 layers with 30, 7 and 1 neuron in the first, second and third layer, respectively. It is trained by the resilient backpropagation rule using samples obtained by measurements and the dynamic model of the RSWS. Performances of the trained ANN were evaluated by the tests performed with the different measured samples. The results of the laboratory tests, shown in Fig. 13, are very promising and show reliable recognition of the iron core saturation.

6. References

Brown, B. M. (1987). A Comparison of AC and DC Resistance Welding of Automotive Steels. *The Welding Journal,* Vol. 66, No. 1, January 1987, pp. 8-23

Deželak, K.; Pihler, J.; Štumberger, G.; Klopčič, B. & Dolinar, D. (2010). Artificial Neural Network Applied for Detection of Magnetization Level in the Iron core of a Welding Transformer. *IEEE Trans. On Magnetics,* Vol. 46, No. 2, February 2010, pp. 634-637

Deželak, K.; Klopčič, B.; Štumberger, G. & Dolinar, D. (2008). Detecting Saturation Level in the Iron Core of a Welding Transformer in a Resistance Spot-Welding System. *Journal of Magnetism and Magnetic Materials,* Vol. 320, No. 20, October 2008, pp. 878-883

Deželak, K.; Štumberger, G.; Klopčič, B.; Dolinar, D. & Pihler, J. (2008). Iron Core Saturation Detector Supplemented by an Artificial Neural Network. *Prz. Elektrotech.,* Vol. 84, No. 12, 2008, pp. 157-159

Hassoun, H. M. (1995). *Fundamentals of Artificial Neural Networks,* Library of Congress Cataloging, 0-262-08239-X, Massachusetts Institute of Technology

Hoyong, K.; Yunseok, K. & Kyung-Hee, J. (1993). Artificial Neural Network Based Feeder Reconfiguration for Loss Reduction in Distribution Systems. *IEEE Trans. On Power Delivery,* Vol. 8, No. 3, 1993, pp. 1356-1366

Klopčič, B.; Dolinar, D. & Štumberger, G. (2008). Advanced Control of a Resistance Spot Welding System. *IEEE Trans. On Power Electronics,* Vol. 23, No. 1, January 2008, pp. 144-152

Leon, F. & Semlyen, A. (1994). Complete Transformer Model for Electromagnetic Transients. *IEEE Trans. On Power Delivery,* Vol. 9, No. 1, January 1994, pp. 231-239

Pihler, J.; Grčar, B. & Dolinar, D. (1997). Improved Operation of Power Transformer Protection Using Artificial Neural Network. *IEEE Trans. On Power Delivery,* Vol. 12, No. 3, 1997, pp. 1128-1136

Ramboz, D. J. (1996). Machinable Rogowski Coil, Design, and Calibration. *IEEE Trans. Instrumentation and Measurement,* Vol. 45, No. 2, April 1996

Sabate, J. A.; Vlatkovic, V.; Ridley, R. B.; Lee, F. C. & Cho, B. H. (1990). Design Considerations for High-voltage High-power Full Bridge Zero Voltage-switching PWM Converter. *IEEE Appl. Power Electron.,* Conference, 1990, pp. 275-284

Štumberger, G.; Klopčič, B.; Deželak, K. & Dolinar, D. (2010). Prevention of Iron Core Saturation in Multi-Winding Transformer for DC-DC Converters. *IEEE Trans. On Magnetics,* Vol. 46, No. 2, February 2010, pp. 582-585

Part 3

Food Industry

10

Application of Artificial Neural Networks to Food and Fermentation Technology

Madhukar Bhotmange and Pratima Shastri
Laxminarayan Institute of Technology, Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur 440033.
India

1. Introduction

Every system is controlled by certain parameters and works at its best for a certain combination of the values of these parameters Input parameters of the system are defined as the independent variables or causes, which affect the values of output parameters commonly identified as effects. The relationship in many case is typically nonlinear, and complex. Different input parameters –apart from their individual influences – may affect the output parameter in synergistic or antagonistic way.

The knowledge of cause-and-effect relationships is important in the solution of problems in all fields of endeavor. In the simplest of cases, these relationships may take on a linear form, while in others, highly nonlinear and complex, relationships may be appropriate. Some relationships are static, while others involve dynamic or time varying elements.

A complex system like thermal processing requires maximum destruction of undesirable microorganisms with minimum loss of freshness, taste, texture and flavor as the outputs, with time temperature, can size, etc. as extrinsic causes, along with the composition, viscosity, and thermal properties of food material as intrinsic causes. Product development happens to be an equally complex system where level and proportion of ingredients are the inputs, which determine the sensory parameters, cost and marketability. Modeling of bioprocesses for engineering applications is equally challenging task, due to their complx nonlinear dynamic behaviour.

The conditions of best functioning are called optimum operating / functioning conditions. Large number of experiments need to be performed under certain set of conditions, for obtaining these optimum parameters. Still, the results at selected data points need not necessarily represent the optimum functioning of a process, specially for typical nonlinear systems. Performing permutations and combination with experimental parameters till the optimum combination of parameters is achieved is not only time consuming and laborious, but also contributes to increased expenses, hazard possibility and error incorporations. In such situation, several structured and unstructured models can be developed from the available data, and the possible outputs can be successfully predicted at any combination of values, within the frame work. Artificial Neural Network (ANN) is one such tool for prediction of outputs for nonlinear systems at various combinations. The process is based on learning of the network with the experimental values, thus knowing the system behavior, & then predicting the output values of the desired set of parametric combinations. Food

science and technology represents a potential area for application of ANN. Critical review by Huang et al. (2007) discusses the basic theory of the ANN technology and its applications in food science, providing food scientists and the research community an overview of the current research and future trend of the applications of ANN technology in the field.

2. What is Neural Network?

Mother nature's most complex creation, the human brain has evolved over million of years and has very complex and powerful architecture. It consists of large number of nerve cells called neurons. The axon or output path of a neuron splits up and connects to dendrites or input paths of other neurons through a junction known as a synapse (Fig.1) The transmission across this junction is chemical in nature, and the amount of signal transferred depends on the amount of chemicals (Acetylchloline) released by the axon and in turn received by the dendrites. This synaptic strength is modified when the brain learns. Each neuron will have of the order of 10,000 dendrites through which they accept inputs.

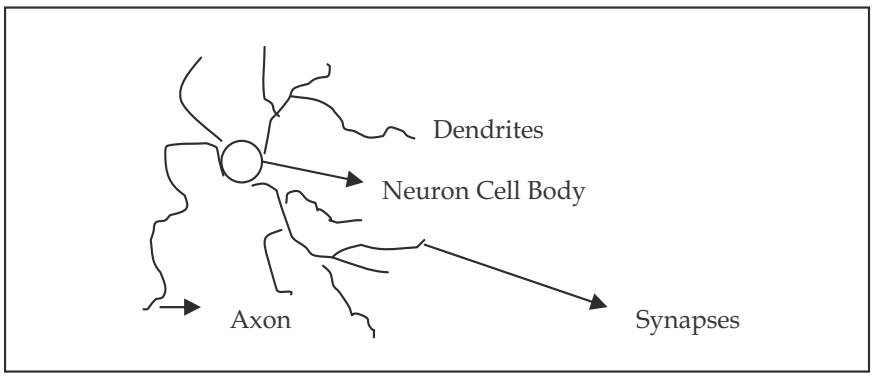


Fig. 1. Biological Neuron

2.1 Artificial Neural Network (ANN)

An artificial neural network (ANN) is a data processing system based on the structure of the biological neural simulation by learning from the data generated experimentally or using validated models.

Some terms required to be defined for ANN users are:

- ANN: A neural network is a processing device, either an algorithm, or actual hardware, whose inspired by the design in and functioning of animal brains and components thereof. It is computer program designed to simulate the brain neurons.
- Processing element: In an ANN, the unit analogous to the biological neuron is a processing elements (PE). Each PE has many inputs and outputs. The network consists of many units or neurons, each possibly having a small amount of local memory. The unit by undirectional communication channels "connections" which carry numeric data. The units operate only on their local data and on the inputs they receive connection.
- Connection weight: The output path of a processing element is connected to input paths of other PEs through connection weights, analogous to the synaptic strength of neural connections.

- Input, output and hidden layers: A network consists of a sequence of layers with connections between successive layers. Data to the network is presented at input layer and the response of the network to the given data is produced in the output layer. There may be several layers between these two principal layers, which are called hidden layers.
- Training: Most neural networks have some sort of "training" rule whereby the weights of connection are adjusted on the basis of presented patterns. In other words, neural network patterns "learn from example".
- Error: It is defined as the total sum of the difference between desired output and output produced by the network for the set of inputs.
- Learning rate: A learning rule, which changes the connection weights of the network in response to the example inputs and desired output to those inputs. The training of neural network model is similar to the way humans or animals are trained by reinforcement technique, where certain synapses that connect the neurons selectively get strengthened leading to increase in the gain.
- Recall: Recall refer to how the network processes a data set presented at its input layer and produces a response at the output layer. The weights are not changed during the recall process.

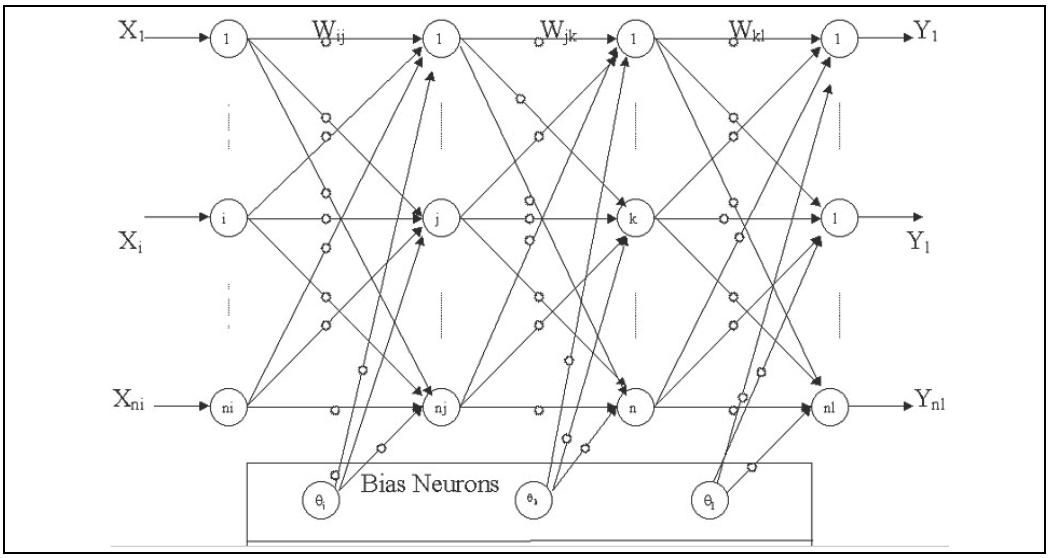


Fig. 2. Artificial Neural Network : A Multilayer Perceptron

Derived from their biological counterparts, ANNs are based on the concept that a highly inter-connected system of simple processing elements can learn complex inter relationships between independent and dependent variables. ANNs offer an attractive approach to the black-box modeling of highly complex, nonlinear systems having a large number of inputs and out puts in the form of massively connected parallel structures. It has three-layered system, an input layer, and intermediate layer called hidden layer, and an output layer (Fig.2). Each layer contains a number of neurons. The number of neurons in the input layer equals the number on inputs to the neural network while the number of neurons in the output layer equals the number outputs in the system. Although numerous guidelines have

been proposed for selecting the number of units in the hidden layer, they do not work in all situations, and the number is often determined heuristically. Each neuron is connected to all the neurons in the next layer by means of a "connection weight". The output from neurons can be calculated by suitable "transform equations" provided the inputs and the connection weights are known.
The sequence of neural network modeling is to assume a set of weights initially, compute the outputs and the predict error, and then adjust the weights according to an error minimization technique until the prediction error falls to an acceptable level. This activity of finding optimal weight is called network training. Once the network is so trained, the black –box model is ready, and may be used to predict outputs for a set of new inputs, not originally part of those used in training.

2.2 Types of ANN

1. Back Propogation Network (BPN)

Back Propogation Network has been extensively studied, theoretically, and has been the most successful. The BPN is usually built from a three layered system consisting of input, hidden, and output layers. An equation in the hidden layers (transfer function) determines whether inputs are sufficient to produce an output (Hornik et al 1989). There are several kinds of transfer functions, e.g. threshold or sigmoid functions. In training a NN, the values predicted by the net work are compared to experimental values using the delta rule, an equation which minimizes error between experimental values and net work predicted values. The errors are then back propagated to hidden and input layers to adjust weights. This is repeated many times until errors between predicted and experimental values are minimized. General reviews, and references of NN procedure are discussed by Eberhart and Dobbins (1990) .

2. General Regression Neural Network (GRNN)

General Regression Neural Network are memory based feed forward networks meaning that all the training samples are stored in the network. It possess a special property that they do not require iterative training.

3. Neural network vs statistical regression

In statistical regression, the parameters or constants of the equation are determined for a given mathematical equation, which relates the inputs to the output(s), so that the difference between the desired output and the output of the equation for the set of inputs is a minimum. Here the type and nature of the equation relating the inputs with the output has to be initially formulated clearly. Neural Network (NN) doesn't require such explicit relationship between the inputs and the output(s). In Neural network parameter values cannot be extracted after the simulation. In statistics the analysis is limited to a certain number of possible interactions. However, more terms can be examined for interaction and included in Neural Network. By allowing more data to be analyzed at the same time, more complex and subtle interactions can be determined. Fuzzy and not so clear data sets can also be analyzed and their interaction studied with Neural Network, whereas statistical regression analysis will fail in such situation.
It can perform better than statistical regression analysis for prediction, modeling & optimization even if the data is noisy and incomplete. It is also ideally suited when the inputs are qualitative in nature and when the inputs or the output can not be represented as

mathematical terms (Pandharipande, 2004). Unlike other modeling such as expert system, an ANN can use more than two parameters to predict two or more parameters. In addition, ANN differs from traditional methods due to their ability to learn about the system to be modeled without a prior knowledge of the process parameter. ANN results are straight forward and do not need any transformations. ANN is amongst various intelligent modeling methods which are able to solve a very important problem –processing of unstructured ,scarce and incomplete numerical information about nonlinear and non stationary systems , as well as biotechnological processes (Vassileva et al, 2000).. ANN has the ability for relearning according to new data., and it is possible to add new observations at any time. Unlike ANN, when new observations are added to the data set in PCR, principal components have to be calculated before regression analysis is applied (Vallejo-Cordoba et al ,1995)

4. Applications of ANN in food technology

Artificial Neural Networks (ANNs) have been applied in almost every aspect of food science over the past two decades, although most applications are in the development stage. ANNs are useful tools for food safety and quality analyses, which include modeling of microbial growth and from this predicting food safety, interpreting spectroscopic data, and predicting physical, chemical, functional and sensory properties of various food products during processing and distribution. ANNs hold a great deal of promise for modeling complex tasks in process control and simulation and in applications of machine perception including machine vision and electronic nose for food safety and quality control.

4.1 ANN for prediction of food quality, properties and shelf life

Quality of food is complex term, and is assessed by suitable combination of physical, chemical and organoleptic tests. Physical / chemical parameters- though convenient to measure - do not always have straightforward correlations with the sensory evaluation results. However, frequent sensory evaluation is restricted due to the availability of trained judges, and proper ambience. Several investigators have attempted to apply ANN models for prediction of food properties, and changes during processing and storage of foods.

Zhang and Chen (1997) introduced a method of food sensory evaluation employing artificial neural networks. The process of food sensory evaluation can be viewed as a multi-input and multi-output (MIMO) system in which food composition serves as the input and human food evaluation as the output. It has proved to be very difficult to establish a mathematical model of this system; however, a series of samples have been obtained through experiments, each of which comprises input and output data. On the basis of these sample data, the back-propagation algorithm (BP algorithm) is applied to "train" a three-layer feed-forward network. The result is a neural network that can successfully imitate the food sensory evaluation of the evaluation panel. This method can also be applied in other fields such as food composition optimizing, new product development and market evaluation and investigation.

Lopez et al (1999) have applied ANN for identification of registered designation of origin areas of portugese cheese defined by microbial phenotypes and artificial neural networks.

The human sense of smell is the faculty which has very important role to play in industries such as beverages, food and perfumes. Studies have been carried out to construct an instrument that mimics the remarkable capabilities of the human olfactory system (Gardner et al 1990). The instrument or electronic nose consists of a computer-controlled multi-sensor

array, which exhibits a differential response to a range of vapors and odors. The authors report on a novel application of artificial neural networks (ANNS) to the processing of data gathered from the integrated sensor array or electronic nose. This technique offers several advantages, such as adaptability, fault tolerance, and potential for hardware implementation over conventional data processing techniques. Results of the classification of the signal spectra measured from several alcohols are reported and they show considerable promise for the future application of ANNs within the field of sensor array processing. Electronic/artificial nose, developed as systems for the automated detection and classification of odors, vapors, and gases is generally composed of a chemical sensing system (e.g., sensor array or spectrometer) and a pattern recognition system (e.g., artificial neural network). Electronic noses for the automated identification of volatile chemicals for environmental, medical and food industry applications are being developed

A similar report on application of electronic nose for classification of pig fat has been reported by Carrapsio et al. (2001). Fatty acid analysis is frequently performed in fat and other raw materials to classify them according to their fatty acid composition, but the need to carry out online determinations has generated a growing interest in more rapid options. This research was done to evaluate the ability of a polymer-sensor based electronic nose to classify Iberian pig fat samples with different fatty acid compositions. Significant correlations were found between individual fatty acids and sensor responses, proving that sensor response data were not fortuitously sorted. Significant correlations also appeared between some sensors and water activity, which was considered during the sample classification. Two supervised pattern recognition techniques were attempted to process the sensor responses: 85.5% of the samples were correctly classified by discriminant analysis, but the percentage increased to 97.8% using a one-hidden layer back-propagation artificial neural network.

An artificial olfactory system based on Gas Sensor Array and Back-Propagation Neural Network is constructed to determine the individual gas concentrations of gas mixture (CO and H_2) with high accuracy. Back-Propagation (BP) neural network algorism has been designed using MATLAB neural network toolbox, and an effective study to enhance the parameters of the neural network, including pre-processing techniques and early stopping method is presented in this paper. It is showed that the method of BP artificial neural improves the selectivity and sensitivity of semiconductor gas sensor, and is valuable to engineering application (Tai et al., 2004). The electronic nose (sensor responses analyzed by a neural network) achieved success similar to that obtained using the more usual fatty acid analysis by gas chromatography. Similar application in fatty acid analysis of soyabean oil is reported by Kovalenko et al (2006).

An artificial neural network model is presented for the prediction of thermal conductivity of food as a function of moisture content, temperature and apparent porosity. (Sablani and Rahman, 2003).The food products considered were apple, pear, corn starch, raisin, potato, ovalbumin, sucrose, starch, carrot and rice. The thermal conductivity data of food products (0.012-2.350W/mK) were obtained from literature for the wide range of moisture content (0.04-0.98 on wet basis fraction), temperature (-42-130^{o}C)and apparent porosity(0.0-0.7). Several configurations were evaluated while developing the optimal ANN model. The optimal model ANN consisted two hidden layers with four neurons in each layer. This model was able to predict thermal conductivity with a mean relative error of 12.6%,a mean absolute error of 0.081 W/mK. The model can be incorporated in heat transfer calculations during food processing. Rahman's model (at 0^{o}C) and a simple multiple regression model predict thermal conductivity with mean relative error of 24.3%.

An interesting application of ANN for identification of organically farmed atlantic salmon from wild salmon is by analysis of stable isotopes and fatty acids is discussed by Molkentin et al (2007). Using isotope ratio mass spectrometry (IRMS), the ratios of carbon (δ ^{13}C) and nitrogen (δ ^{15}N) stable isotopes were investigated in raw fillets of differently grown Atlantic salmon (*Salmo salar*) in order to develop a method for the identification of organically farmed salmon. IRMS allowed to distinguish organically farmed salmon (OS) from wild salmon (WS), with δ ^{15}N-values being higher in OS, but not from conventionally farmed salmon (CS). The gas chromatographic analysis of fatty acids differentiated WS from CS by stearic acid as well as WS from CS and OS by either linoleic acid or α-linolenic acid, but not OS from CS. The combined data were subjected to analysis using an artificial neural network (ANN). The ANN yielded several combinations of input data that allowed to assign all 100 samples from Ireland and Norway correctly to the three different classes. Although the complete assignment could already be achieved using fatty acid data only, it appeared to be more robust with a combination of fatty acid and IRMS data, i.e. with two independent analytical methods. This was also favorable with respect to a possible manipulation using suitable feed components. A good differentiation was established even without an ANN by the δ ^{15}N-value and the content of linoleic acid. The general applicability in the context of consumer protection is recommended be checked with further samples, particularly regarding the variability of feed composition and possible changes in smoked salmon.

Experimental measurements of the variation in the solid fraction during crystallization of lipid mixtures are often correlated in terms of the so-called Avrami model. Jose et al (2007) employed above model to describe measurements taken during the crystallization of blends of tripalmitin in olive oil at high concentrations. Although the blends appeared to behave ideally, the Avrami model failed to describe the experimental results over the entire range of tripalmitin concentration investigated. As an alternative to the description of lipid crystallization experiments, the use of continuous-time artificial neural network (ANN) approximators is proposed. ANN successfully reproduced the experimentally observed behavior for all temperatures and tripalmitin concentrations used.

ANN based automatic grading and sorting systems for fruits and vegetables have been developed by various investigators. Saito et al (2003) have developed eggplant grading system using image processing and artificial neural network. The lighting conditions are discussed for taking color components of the eggplant image effectively. The shape parameters such as length, girth, etc. are measured using image processing. On the other hand, bruises of the eggplants are detected and classified based on the color information by using artificial neural network. Development of electronc nose for determination of fruit ripeness has been reported by Salim et al. (2005).

A combination of machine vision and artificial neural network model for guava sorting which classify from size, weight and defect of guava has been described by Chokananporn and Tansakul (2008) and the system was evaluated by comparing with human sorting. Furthermore, the surface area of guava could be estimated from the artificial neural network model. The major diameter, intermediate diameter, minor diameter, and sphericity were used to classify the shape and used as the input parameters of the network. The sorting process was controlled by computer software which was well designed and created on visual basic 6.0. The experiments were carried out with fresh guava. The results from machine vision system were compared with those from human classifying capability. One hundred percent coincidence for the extra size and 73.3 percent coincidence for the class I and II size were obtained. For surface area estimation, the predicted surface area was found

to be nearly the same as that from the standard method. The lowest mean relative error (MRE) and mean absolute error (MAE) values were 0.15% and 0.39 cm^2, respectively. Similar combination system for classification of beans is reported by Kilik et al (2007).

Prediction of Milk shelf – life based on Artificial Predicting Neural networks and head space gas chromatographic data has been reported by Vellejo-Cordoba et al. (1995 **).** Pasteurized milk was sampled during refrigerated storage at 4°C until termination of shelf life, as determined by sensory evaluation, sub samples were incubated at 24 $\pm$1°C for 18 hours prior to detection of volatiles by dynamic head space gas chromatograph (Cordoba & Nakai, 1994)). Several volatiles consisting mainly of aldehydes, ketones & alcohols were identified in milk. Not only increased peak areas of the compounds already present appeared in poor-quality milk, new volatiles were also detected, including esters. Cross validation was used with 113 training sets, and 21 test sets. In PCR, the independent variables were the first 30 principal components and the dependent variable was flavor – based shelf life in days. The shelf life predictability of ANN was superior to PCR as indicated by carrying out regression analysis for experimental vs predicted shelf life and the squared correlation (r^2) and the standard error of the estimate (SEE).

The power of computational neural networks (CNN) for growth prediction of three strains of *Salmonella* as affected by pH level, sodium chloride concentration and storage temperature was evaluated by Herv's et al (2001). The architecture of CNN was designed to contain above three input parameters and growth as output parameter. The standard error of prediction (%SEP) obtained was under 5% and was significantly less than the one obtained using regression equations. Similar study by Zurera-Cosano et al (2005) reported an Artificial Neural Network-based predictive model (ANN) for *Leuconostoc mesenteroides* growth in response to temperature, pH, sodium chloride and sodium nitrite, was validated on vacuum packed, sliced, cooked meat products and applied to shelf-life determination. Lag-time (*Lag*), growth rate (*Gr*), and maximum population density (*yEnd)* of *L. mesenteroides*, estimated by the ANN model, were compared to those observed in vacuum-packed cooked ham, turkey breast meat, and chicken breast meat stored at 10.5°C, 13.5°C and 17.7°C. From the three kinetic parameters obtained by the ANN model, commercial shelf-life were estimated for each temperature and compared with the tasting panel evaluation. The commercial shelf life estimated microbiologically, i.e. times to reach $10^{6.5}$ cfu/g, was shorter than the period estimated using sensory methods.

Application of ANN for prediction of shelf life of green chilli powder (GCP) is reported by Meshram (2008).Green Chilli Powder (GCP) prepared by dehydration of *Jwala* variety of chilli in air–Radio Frequency (RF) combo dryer had 1.13% moisture content with 19% ERH. Danger and critical points were identified at 60.5 % and 63% ERH corresponding to 7.12% and 8.0% moisture content respectively. Storage study was carried out under ambient (25°C, 65% RH) and accelerated (38°C, 90% RH) conditions for GCP packed in Laminated aluminium foil (LAM) and Polypropylene (PP). Half Value Period (HVP) and shelf life at different combinations of temperature (T) and relative Humidity (RH%) for 100 g GCP pack was calculated based on WVTR (LAM =2.35, PP =4.16 units at 38°C,90% RH) and packaging constant.(Ranganna). Application of Artificial Neural Network (***elite-ANN*** ©) for prediction of shelf life as function of T and RH% gave R^2 value >0.99 for both packings.

4.2 ANN in food processing

Various processing parameters are required to be monitored and controlled simultaneously, and it is quite difficult to derive classical structured models, on account of practical

problems in conducting required number of experiments and lack of sufficient data. Possibility for application of ANN for optimizing the process parameters is an interesting area, with many potential applications.

The effect of agglomerate size and water activity on attrition kinetics of some selected agglomerated food powders was evaluated by Hong Yan and Barbosa-Canovas (2001) by application of ANN. Investigation of the attrition of agglomerates is very important for assessing the agglomerate strength, compaction characteristics, and quality control. A one-term exponential attrition index model and the Hausner ratio were used to study the effects of agglomerate size and water activity on the attrition kinetics of some selected agglomerated food powders. It was found that the agglomerate size and water activity played significant roles in affecting the attrition: the larger the agglomerate size and higher the water activity, higher was the attrition index under the same tap number. The Hausner ratio was well correlated with the attrition index at high tap numbers and might be used as a simple index to evaluate attrition severity for agglomerates. Knowing the effects of agglomerate size and water activity is very useful to minimize the attrition phenomenon during the handling and processing of agglomerated powders.

Modeling and control of a food extrusion process using artificial neural network and an expert system is discussed by Popescue et al. (2001). A neural network model is proposed and its parameters are determined. Simulation results with real data are also presented. The inputs and outputs of the model are among those used by the human operator during the start-up process for control. An intelligent controller structure that uses an expert system and "delta-variations" to modify inputs is also proposed.

A hypothesis on coating of food is put forward by Bhattacharya et al (2008), who have also discussed development of a system analytical model based on simulation studies and artificial neural network The process of coating of foods is a complex process due to the presence of a large number of variables, and unknown relationship between the coating variables and coating characteristics. Needs exists to develop a model that can relate the important variables and coating parameters that would be helpful in developing coated products. A system analytical model for coating of foods has been hypothesized. The model relates influencing variables to derived parameters that in turn relates the target coating parameters. The concentration of solids and temperature of coating dispersions are the examples of the influencing variables, whereas rheological parameters (apparent viscosity, yield stress, flow and consistency indices) are the derived parameters that finally decide the coating parameters such as total uptake, solid uptake and dimensionless uptake according to the hypothesized relations $y = f(x)$ and $z=g(y)$. The proposed hypothesis was initially examined by performing simulation studies conducted on steel balls (small and big) using sucrose solution and malt – maltodextrin dispersions at different concentrations (20-60%) and temperatures (5-80°C), and applying the theory of artificial neural network (ANN) for prediction of target parameters. The hypothesis was tested in actual system using corn balls and sucrose solution. The proposed analytical model has been employed to develop sweetened breakfast cereals and snacks.

Application of ANN in baking has been studied out by few investigators. The bake level of biscuits is of significant value to biscuit manufacturers as it determines the taste, texture and appearance of the products. Previous research explored and revealed the feasibility of biscuit bake inspection using feed forward neural networks (FFNN) with a back propagation learning algorithm and monochrome images (Yeh et al 1995). A second study revealed the existence of a curve in colour space, called a baking curve, along which the

bake colour changes during the baking process. Combining these results, an automated bake inspection system with artificial neural networks that utilises colour instead of monochrome images is evaluated against trained human inspectors .

Comparison of Neural Networks Vs Principal component regression for prediction of wheat flour loaf volume in baking tests has been reported by Harimoto et al. (1995). The objective here was to determine values of four parameters which minimize the standard error of estimate (SEE) between prediction of NN & actual, measured remix loaf volumes of the flour. Two hundred patterns (i.e. quality test results of 200 flours) were used for training the NN. The training tolerance specifies how close each output (remix loaf volume) of the network must be to the empirical response to be considered "correct" during training. The training tolerance is a percentage of the range of the out put neuron. Net works with smaller tolerances require longer time to train. If a network is slow in learning, it is sometimes helpful to begin with a wide tolerance and then narrow tolerance. A back-propagation neural network has been developed by Ruan et al (1995) to accurately predict the farinograph peak, extensibility, and maximum resistance of dough using the mixer torque curve. This development has significant potential to improve product quality by minimizing process variability. The ability to measure the rheology of every batch of dough will enable online process control through modifying process conditions.

Razmi Rad et al (2007) have shown the ability of artificial neural network (ANN) technology for predicting the correlation between farinographic properties of wheat flour dough and its chemical composition. With protein content, wet gluten, sedimentation value and falling number as input parameters six farinographic properties including water absorption, dough development time, dough stability time, degree of dough softening after 10 and 20 min and valorimeteric value as output parameters. The ANN model predicted the farinographic properties of wheat flour dough with average RMS 10.794. indicating that the ANN can potentially be used to estimate farinographic parameters of dough from chemical composition. A neural network based model was developed for the prediction of sedimentation value of wheat flour as a function of protein content, wet gluten and hardness index (Razmi et al 2008). The optimal model, which consisted of one hidden layer with nine neurons, was able to predict the sedimentation value with acceptable error. Thus, ANN can potentially be used to estimate other chemical and physical properties of wheat flour.

Ismail et al (2008) have compared chemometric methods including classical least square (CLS), principle component regression (PCR), partial least square (PLS),and artificial neural networks (ANN) for estimation of dielectric constants (DC) dielectric loss factor (DLF) values of cakes by using porosity, moisture content and main formulation components, fat content, emulsifier type (Purawave™, Lecigran™), and fat replacer type (maltodextrin, Simplesse). Chemometric methods were calibrated firstly using training data set, and then they were tested using test data set to determine estimation capability of the method. Although statistical methods (CLS,PCR and PLS) were not successful for estimation of DC and DLF values, ANN estimated the dielectric properties accurately (R^2, 0.940 for DC and 0.953 for DLF). The variation of DC and DLF of the cakes when the porosity value, moisture content, and formulation components were changed were also visualized using the data predicted by trained network ANN is applied for prediction of temperature and moisture content of frankfurters during thermal processing (Mittal and Zhang, 2000). Lou, and Nakai (2001). Have discussed application of artificial neural networks for predicting the thermal inactivation of bacteria as a combined effect of temperature, pH and water activity.

Linear Regression, NN & Induction Analysis to determine harvesting & processing effects on surimi quality is reported by (Peters et al 1996). Surimi production is highly technical process requiring considerable skill. Harvesting & Processing input combinations and product quality attributes for the pacific writing surimi industrial were collected and analyzed. Multiple linear regression (MLR), NN, & MS – Induction were used to determine significant variables in the industry. MLR incorporated time, temperature and date of harvest as the variables, whereas ANN could incorporate other significant variable factors intrinsic to the fish (moisture content, salinity, pH , length, weight) and processing variables (processing time, storage temp, harvest date, wash time, wash ratios) in addition to the above three variables. Most variables were highly interactive and non linear. The back propagation NN algorithm was used to relate the influences of the variables (inputs) and their effects on quality (output) as defined by gel strength the NN model was trained so that the model predication was = 10% of the actual value for all data points.

Comparison of three analytical systems, MLR, NN, & MS –I showed that time from capture to final production, temp of storage and date of harvest were indicated to be critical to get desired gel strength by all systems. ANN & MS-I also identified fish weight and length, salinity & moisture of flesh as important processing parameters. In addition, NN analysis indicated flesh pH, wash ratios and geographic location were important factors that affect quality. NN and MS-I were effective computer based methods for analyzing large data sets of complex biological system. They were especially useful for determining factors that affect final product quality in a multi-process operation.

A three-layer feed forward neural network was successfully applied by Paquet et al (2000) to model and predict the pH of cheese curd at various stages during the cheese-making process. An extended database, containing more than 1800 vats over 3 yr of production of Cheddar cheese with eight different starters, from a large cheese plant was used for model development and parameter estimation. Very high correlation coefficients, ranging from 0.853 to 0.926, were obtained with the validation data. A sensitivity analysis of neural network models allowed the relative importance of each input process variable to be identified. The sensitivity analysis in conjunction with a prior knowledge permitted a significant reduction in the size of the model input vector. A neural network model using only nine input process variables was able to predict the final pH of cheese with the same accuracy as for the complete model with 33 original input variables. This significant decrease in the size of neural networks is important for applications of process control in cheese manufacturing.

Optimization of the process of extraction of soy-fiber from defatted soy-flour is reported by Gupta and Shastri (2005). Defatted soya flour (DSF) is a good source of proteins, which are extracted in alkaline medium. The concept of integrated processing of DSF involves simultaneous recovery of soya proteins and fiber, which find use in dietetic foods. Process needs to be optimized to solubiise maximum protein, which is recovered afterwards as Soya Protein Isolate (SPI), with minimum fiber disintegration, and maximum recovery. DSF (obtained from Rasoya Ltd. Nagpur) contained 40.3% protein and 25% fiber. Extraction of soy-fiber was carried out by alkaline extraction at 11 different concentration-time combinations with alkali concentration (range 0.1-0.5N) as variable I, and extraction time (range 0.5-1.5 hours) as variable II. Maximum recovery of the fiber after protein solubilization was the required output. ANN elite; software (Pandharipande &. Badhe,2003) was applied by selecting three hidden layers with 5 neurons, 0.9 learning rate and 0.001 back propagation error. Learning of the network was carried out using 9 data points from the

experimental data, whereas remaining two data points were used for assessment of the learning status of the network. The comparison between the experimental and predicted results is given in Fig. (3)

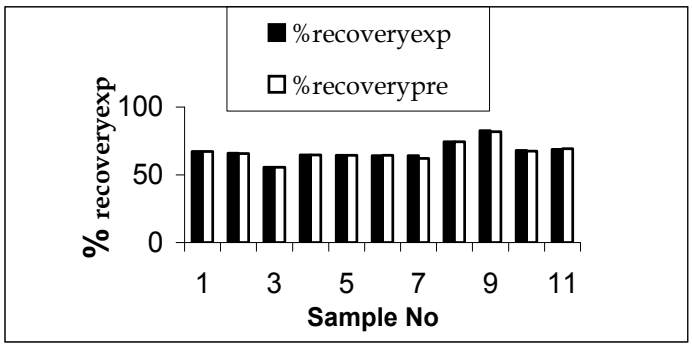


Fig. 3. Experimental and predicted values for recovery of fiber from DSF

The optimum conditions predicting maximum percentage recovery under the above consideration were found to be 0.5 hrs extraction time with 0.5 N alkali(condition I) and 0.35N alkali for 60 minutes (condition II). Validity of the model was established by confirming the recovery under the selected combinations of alkali concentration and time which showed excellent correlation (R^2=0.998) with the predicted values, Thus, it can be concluded that the developed Artificial Neural Network model has been used effectively as a tool in optimizing the process parameter for removal of fiber from DSF.

5. ANN in the field of biotechnology

ANN can be a boon in the field of biotechnology in view of the complex nature of biocatalysts and microorganisms and their interactions with the environment. Prediction of models is usually very difficult on account of the lack of information about the physiological and biochemical constraints of biocatalysts, and their effect on physical phenomena like solubility of nutrients, oxygen transfer, and availability of water. ANN has the advantage that it can make accurate forecast even when the process behavior is non linear and data is unstructured. Since network training is fast, the method is suitable for on-line forecasting.
Characteristic of the beer production process is the uncertainty caused by the complex biological raw materials and the yeast, a living organism. Thus, predicting the speed of the beer fermentation process is a non-trivial task. Data sets from laboratory-scale experiments as well as industrial scale brewing process were used to develop the neural network and descision tree. Simple decision trees were able to predict the classes with 95%–98% accuracy. Utility of these methods was checked in a real brewery environment. The neural network could, on average, predict the duration of the fermentation process within a day of the true value; an accuracy that is sufficient for today's brewery logistics. The accuracy of the decision tree in detecting slow fermentation was around 70%, which is also a useful result. (Rousu et al 1999). Beluhan and Beluhan (2000) describe estimation of yeast biomass concentration in industrial fed-batch yeast cultivation process with separate arificial neural networks combined with balance equations. Static networks with local recurrent memory structures were used for on line estimation of yeast biomass concentration in industrial

bioreactor , and the inputs were standard cultivation state variables: respiratory quotient, molasses feed rate, ethanol concentration, etc. This hybrid approach is generally applicable to state estimation or prediction when different sources of process information and knowledge have to be integrated.

Multivariate statistical methods namely, principal component analysis (PCA) and partial least squares (PLS), which perform dimensionality reduction and regression, respectively, are commonly used in batch process modeling and monitoring. A significant drawback of the PLS is that it is a linear regression formalism and thus makes poor predictions when relationships between process inputs and outputs are nonlinear. For overcoming this drawback of PCA, an integrated generalized regression neural networks (GRNNs) is introduced for conducting batch process modeling and monitoring. The effectiveness of the proposed modeling and monitoring formalism has been successfully demonstrated by conducting two case studies involving penicillin production and protein synthesis.(Kulkarni et al 2004). Application of neural network (ANN) for the prediction of fermentation variables in batch fermenter for the production of ethanol from grape waste using *Saccharomyces cerevisiae* yeast has been discussed by Pramanik (2004). ANN model, based on feed forward architecture and back propagation as training algorithm, is applied in this study. The Levenberg- Marquardt optimization technique has been used to upgrade the network by minimizing the sum square error (SSE). The performance of the network for predicting cell mass and ethanol concentration is found to be very effective. The best prediction is obtained using a neural network with two hidden layers consisting of 15 and 16 neurons, respectively.

Online biomass estimation for bioprocess supervision and control purposes is addressed by Jenzsch et al (2006), for the concrete case of recombinant protein production with genetically modified *Escherichia coli* bacteria and perform a ranking. As the biomass concentration cannot be measured online during the production to sufficient accuracy, indirect measurement techniques are required. At normal process operation, the best estimates can be obtained with artificial neural networks (ANNs). Simple model-based statistical correlation techniques such as multivariate regression and principle component techniques analysis can be used as alternative. Estimates based on the Luedeking/Piret-type are not as accurate as the ANN approach; however, they are very robust. Techniques based on principal component analysis can be used to recognize abnormal cultivation behavior. All techniques examined are in line with the recommendations expressed in the process analytical technology (PAT)-initiative of the FDA.

Badhe et al (2002) extended application of ANN to study hydrolysis of castor oil by Pancreatic lipase *(Biocon India Ltd.)* at 35 ^{o}C at pH 7.5 in immobilized membrane bio reactor to investigate the application of free and immobilized lipase for oil hydrolysis. Effect of three variables, e.g. enzyme concentration (range 0.1-0.5 ml), substrate concentration (Range 0.25 to 2.0g) and reaction time (range 2 –8 hours) on percent hydrolysis was investigated. Total 30 data points in the above mentioned range were subjected to training and validation using *elite*^{ANN} software (Pandharipande & Badhe, 2003) with feed forward, sigmoidal activation function & delta learning rule. The topology of the system is described as in Table 1. ANN predictions were accurate (R^2 =0.998) for predicting the percentage hydrolysis of castor oil by lipase enzyme as a function of enzyme concentration, ratio of substrate to buffer concentration and reaction time.

Number of neurons	input	3
	Output	1
First hidden layer		15
Second hidden layer		07
Training data points		26
Test		04
Learning rate		0.8

Table 1. Topology of the ANN network applied for prediction for hydrolysis of castor oil using pancreatic lipase.

Cheese whey proteolysis, carried out by immobilized enzymes, can either change or evidence functional properties of the produced peptides, increasing the potential applications of this byproduct of the dairy industry. However, no information about the distribution of peptides' molecular sizes is supplied by the mass balance equations and Michelis Menten like kinetics. Sousa et al (2003) present a hybrid model of a batch enzymatic reactor, consisting of differential mass balances coupled to a "neural-kinetic model," which provides the molecular weight distributions of the resulting peptides.

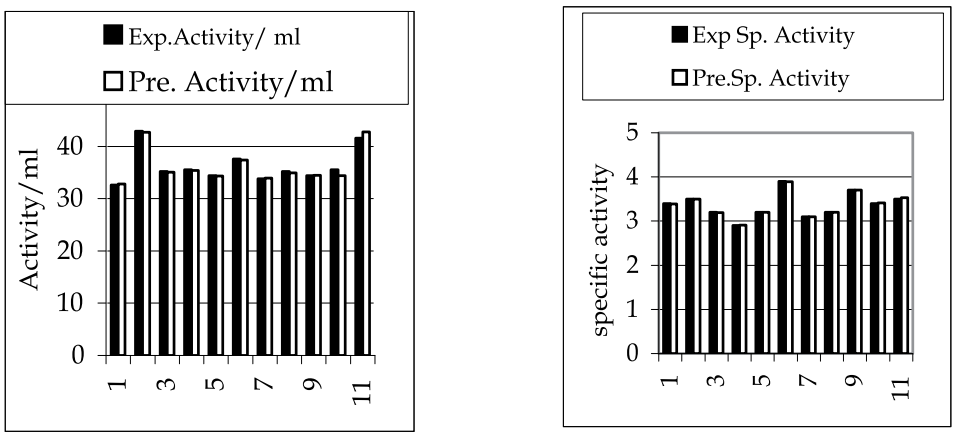
6. ANN for prediction of enzyme production

Mazutti et al (2009) have studies production of inulinase employing agroindustrial residues as the substrate to reduce production costs and to minimize the environmental impact of disposing these residues in the environment. This study focused on the use of a phenomenological model and an artificial neural network (ANN) to simulate the inulinase production during the batch cultivation of the yeast *Kluyveromyces marxianus NRRL Y-7571,* employing a medium containing agroindustrial residues such as molasses, corn steep liquor and yeast extract. It was concluded that due to the complexity of the medium composition it was rather difficult to use a phenomenological model with sufficient accuracy. For this reason, an alternative and more cost-effective methodology based on ANN was adopted. The predictive capacity of the ANN was superior to that of the phenomenological model, indicating that the neural network approach could be used as an alternative in the predictive modeling of complex batch cultivations.

SSF is defined as cultivation of microorganisms on a moist insoluble substrate, which binds sufficient water to solubilize the nutrients. The desirable a_w is 0.88 to 0.85 and the amount of water to be added is determined by the water binding capacity of the solid substrate. Although wheat bran is widely recommended ingredient in SSF, several other lignocellulosic agrowastes may be incorporated as inducers for specific products. (Deshpande et al 2008). On account of difference in water binding capacity of such varied substrates, it becomes necessary to optimize the amount of water for achieving maximum productivity. The system can be described as a unstructured model, on account of several undefined parameters and interactions. Possibility of application of ANN for prediction of extracellular enzyme production under SSF conditions was examined for several systems, specially to define optimum level of water in combination of solid substrate containing components with different water binding properties.

Production of Pectin Trans Eliminase (PTE) by *Penicillium oxalicum* was carried out on wheat bran medium by incorporation of de-oiled orange peel (DOP), which was incorporated at

different levels (range 25 – 75%) as first input parameter and levels of substrate: moisture ratio (range 2-3) as second input variable. Enzyme activity units /ml of crude enzyme extract (CEE) was first output parameters and specific activity (enzyme activity units/mg proteins) was second output parameter. ANN topology employed for the study had three hidden layers, each with 10 neurons, learning rate 0.9 and back propogation error =0.0014.
The model was used for prediction of experimental conditions within the system framework for optimum enzyme production and the output predicted by ANN showed excellent concurrence with experimental results (Fig.4). Results clearly indicate that DOP is a good inducer, because increase in orange peel % increases the enzyme activity but to enhance the activity, it is necessary to increase moisture content simultaneously since orange peel has more moisture binding capacity. Optimum combination for high productivity as per the ANN analysis was found to be 60-65% DOP, with 90% moisture. (Yadav et al. 2003).



(a) (b)

Fig. 4. Experimental and predicted values for production of Pectin Trans Eliminase by *Penicillium oxalicum* on Wheat bran : Deoiled Orange Peel medium under SSF conditions
(a) Enzyme units /ml of Crude enzyme extract
(b) Specific activity

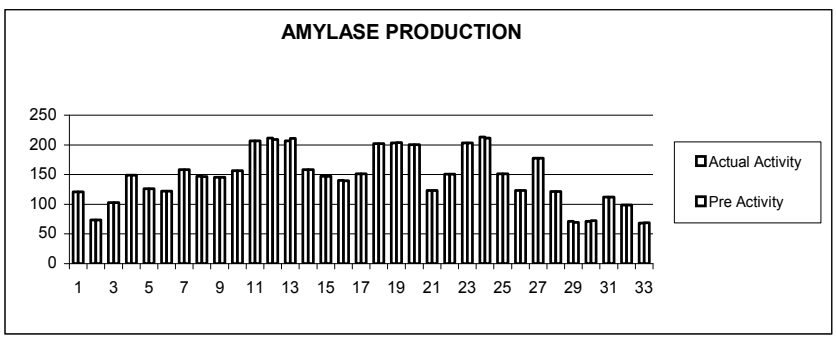


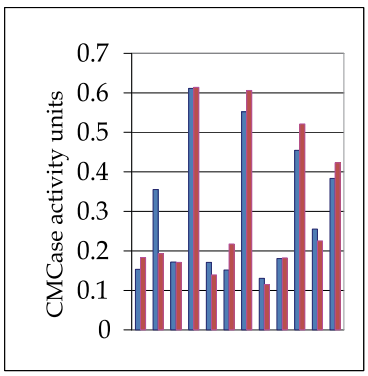
Fig. 5. Experimental and predicted values for specific activity of Amylase produced by *Aspergillus oryzae* on sorghum grit and sorghum stalk medium under SSF conditions

Similar Study was carried out for production of amylase by *Aspergillus oryzae* by using combination of sorghum stalk and sorghum grits as substrate (Pandharipande et al. 2003). Sorghum stalk content varied between 0-100%, and the level of moisture varied between (30-70%) with total 33 data sets. Amylase activity units/ml in CEE, as well as the specific activity were the output parameters. The data was processed by *elite*^{ANN} software (Pandharipande and Badhe, 2003), with three hidden layers of 20 neurons each, learning rate of 0.9, and back propogation error 0.0001. The experimental results are shown in Fig. 5.
It was observed that the amount of inducer influenced the amount of water to be added. The optimum specific activity was obtained at inducer level 70% and moisture level 65% experimentally as against the predicted values of 85% inducer and 60% moisture.
Application of ANN for prediction of cellulase and xylanase production by Solid State Fermentation (SSF) was studied using microorganisms *Trichoderma reesei* and *Aspergillus niger* (Singh et al. 2008, 2009). Experiments were performed with three variables on the production of xylanase and cellulase enzyme by *T.reesei* and *A.niger* by SSF. Total 60 different combination of wheat bran-sugarcane bagasse composition, water: substrate ratio and incubation time were selected as shown in Table 2.

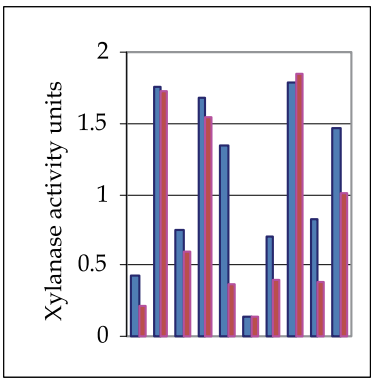
Variable	Range	
	Low	High
1. Bran%	0	100
2.Water Substrate Ratio(v/w) W:S	1.875	3.125
3.Time of Incubation (hrs)	24	168

Table 2. Range of variables selected for study of cellulase & xylanase production

Experimental data was divided into two data series. First set, consisting of about 75-80% of the data points, named as 'Training Set'. It was used for training of ANN to develop independent models for xylanase and cellulase production, each containing three inputs (%wheat bran, W:S Ratio, and Hours of incubation) and one output (IU/ml), three hidden layers (10 nodes each) , learning rate 0.6 and final error 0.002. Adequacy and predictability of the model was tested by giving input parameters for the second data set named as 'Test set' and comparing the predicted and experimental values for *T. reese* (Fig. 6a & 6b,) and *A.niger* (Fig. 7a & 7b) respectively by using *elite-ANN*© software.

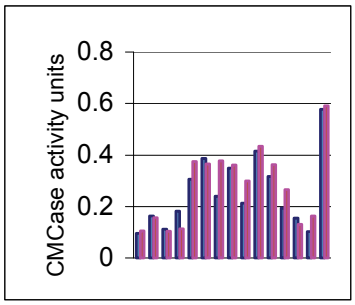


(a)CMCase (R^2=0..846; RMSE0=.082)

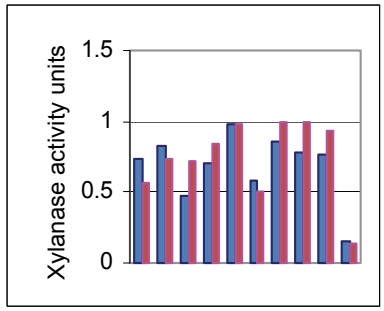


(b) Xylanase (R^2= 0.900 RMSE= 0.371)

Fig. 6. Comparison between actual and predicted values of enzyme production for test data set of *T.reesei* (a) CMCase (b) Xylanase



(a) CMCase R^2= 0.875 RMSE= 0.152



(b) Xylanase R^2= 0.800 MSE = 0.085598

Fig. 7. Comparison between actual and predicted values of enzyme production for test data set of *A niger* (a) CMCase (b) Xylanase

Adequacy and predictability of the developed ANN mode l is judged by the comparison of the actual and the predicted values (Fig.6 &7), which show a satisfactory match as indicated by the correlation coefficients (0.0.90 & 0.81 for xylanase and 0.85 & 0.87 for cellulase) and root mean square error (0.35 & 086 for xylanase and 0.082 & 0.15 for cellulase) for *T.reesei* and *A. niger* respectively. Minor variations in the prediction may be due to complexity and inherent variability of biological system. (Pandharipande et al. 2007)

Production of CMCase and Xylanase by *A.niger* and *T.reesei* under SSF condition is a function of %baggasse (which acts as an inducer) and a_w (which supports the growth). Since wheat bran and baggasse differ in their water absorption capacity (WAC), proportion of water required to achieve desirable a_w in combined substrate needs to be predicted. The observations indicate that CMCase and Xylanase production is optimum (>0.6 units) with greater than 50% baggasse, and ratio of water to substrate being 2.0. Thus it can be concluded that the model developed is validated for the given set & range of process conditions and can be used for the prediction of the enzyme activity at different combinations of parameters and selection of most appropriate fermentation conditions.

Xylanase				Cellulase			
Wheat Bran %	W : S Ratio	Hours	PredictedActivity IU/ml-	Wheat Bran %	W : S Ratio	Hours	Predicted Activity IU/ml-
90	2.75	168	1.769	55	2.5	120	0.527
65	2.5	156	1.491	50	2.25	144	0.497
60	2.5	168	1.387	45	2.25	120	0.443
55	2.25	120	1.524	40	2.75	120	0.426

Table 3. ANN based predicted combinations for optimized production of enzymes by *T. reesei*

Xylanase				Cellulase			
Wheat Bran %	W : S Ratio	Hours	Predicted Activity IU/ml	Wheat Bran %	W : S Ratio	Hours	Predicted Activity IU/ml
90	1.875	108	0.9503	80	1.750	108	0.5489
80	1.750	144	0.9741	75	1.875	120	0.5483
75	2.259	120	0.9027	70	2.000	136	0.4315
60	2.000	136	0.8836	65	2.250	144	0.4832

Table 4. ANN based predicted combinations for optimized roduction of enzymes by *A.niger,*

Above data was subjected to Response Surface Methodology, for Box Behnken model using second order regression equation obtained for the model expressed as follows:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{33} x_3^2 + \beta_{12} x_1 x_2 + \beta_{23} x_2 x_3 + \beta_{31} x_3 x_1$$

where x_1, x_2,and x_3 are inputs, y is the output, The Statistical analysis was done using Minitab1511. The correlation coefficient and MSE obtained by these two models is compared in Table 3 indicating suitability of both models.

Parameter	T.reesei		A,niger	
	Xylanase	CMCase	Xylanase	CMCase
$R^2$by ANN	0.9	O.846	0.8	0.875
R^2 by RSM	0.987	0.79	0.99	0.87
MSE ANN	0.371	0.082	0.856	0.152
MSE RMS	0.034	0.028	0.06	0.076

Table 5. Comparison of ANN and RSM for prediction of cellulase and xylanase production by Solid State Fermentation (SSF)

7. Future prospects

Modern systems with diverse application areas demand expert & accurate calculations within a nick of time. For Such diverse and cutting-edge technology conventional systems have proved expendable and arduous. It is when the Artificial Neural Networks and Fuzzy Systems have proved their speed competitive potentials and expandability. In the last years several propositions for hybrid models, and especially serial approaches, were published and discussed, in order to combine analytical prior knowledge with the learning capabilities of Artificial Neural Networks (ANN). The intelligent modeling approach of models employing Artificial Neural Network in combination with other data analysis systems is able to solve a very important problem - processing of scarce, uncertainty and incomplete numerical and linguistic information about multivariate non-linear and non-stationary systems as well as biotechnological processes (Vassileva et al ,2000, Beluhan and Beluhan, 2000).

8. Acknowledgement

Authors are thankful to Mr. S. L. Pandharipande for making available the ANN software and support for analysis of the experimental data. Authors express their gratitude towards Director, Laxminarayan Institute of Technology, Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur for the encouragement and facilities provided at the institute.

9. References

Badhe YP, Joshi SW, Bhotmange MG, & Pandharipande SL (2002). Modelling of hydrolysis of castor oil by pancreatic lipase using artificial neural network *Proceedings of National Conference on Instrumentation and Controls for Chemical Industries, ICCI 2002*,

Paper 1.2, 8-9th August 2002 Nagpur , organized by Laxminarayan Institute of Technology, Nagpur, India

Beluhan Damir, & Beluhan Sunica (2000). Hybrid modeling approach to on-line estimation of yeast biomass concentration in industrial bioreactor, *Biotechnology Letters* 22(8), pp. 631-635

Bhattacharya Suvendu, Patel Bhavesh K, & Agarwal Kalpesh (2003). Enrobing of foods: Simulation study and application of artificial neural network for development of products. Proceedings of *International Food Convention "Innovative Food Technologies and Quality Systems Strategies for Global Competitiveness" IFCON 2003*, Poster no TC-32, pp 172, Mysore, December 2003.(AFSTI), Mysore India

Carapiso Ana I, Ventanas Jesus, Jurado Angela, & Garcia Carmen (2001). An Electronic Nose to Classify Isberian Pig Fats with different Fatty Acid Composition, *Journal of the American Oil Chemists' Society*, 78(4), pp. 415-418

Deshpande SK, Bhotmange MG, Chakrabarti T, & Shastri PN (2008). Production of cellulase and xylanase by *T. reesei* (QM9414 mutant), *A. niger* and mixed culture by Solid State Fermentation (SSF) of Water Hyacinth (*Eicchornia crassipes*), *Indian Journal of Chemical Technology, 15*(5), pp. 449-456

Eberhart, R. C., & Dobbins, R.W. (1990). *Network* analysis. In Neural Network *PC Tools*. A. *Practical Guide*, R.C. Eberhart and R.W. Dobbins (Ed.), Academic Press, San Diego, CA.

Gardner, JW, Hines, EL, & Wilkinson M (1990). Application of artificial neural networks to an electronic olfactory system, *Journal Measurement Science and Technology*, 1(5), pp. 446.

Gupta R, Pandharipande SL, & Shastri PN (2005). Optimization of the process of extraction of fiber from defatted soyflour using ANN, *National Seminar on Global perspectives for India Food Industry by 2020- Food Vision 2020, organized by Laxminarayan Institute of Technology, Nagpur University, Nagpur.*

Harimoto Y, Durance T, Nakai S, & Lukow O.M. (1995). Neural Networks Vs Principal Component Regression for Prediction of Wheat Flour Loaf Volume in Baking Tests, *Journal of Food Science*, 60(3), pp. 429–433

Herv's, C. G.,Zurera, Garcfa, R M., & Martinez J. A. (2001). Optimization of Computational Neural Network for Its Application in the Prediction of Microbial Growth in Foods *Food Science and Technology International*, 7: 159-163

Hong Yan, G.V., & Barbosa-Canovas (2001). Attrition Evaluation for selected Agglomerated Food Powders: The effect of agglomerate size and water activity, *Journal of Food Process Engineering*, 24(1), pp. 37-49

Hornik, K, Stichcombe, M, & White , H (1989). Multilayer Feed forward Neural Network are universal Approximate . *Neural Network.* 2, pp. 359-366 Huang Y, Kangas LJ, & Rasco BA. (2007). Applications of artificial neural networks (ANNs) in food science. *Crit. Rev Food Sci. Nutr.* 47(2), pp. 113-26

Huiling Tai, Guangzhong Xie, & Yadong Jiang (2004). An Artificial Olfactory system based on Gas Sensor Array and Back-Propogation Neural Network, Lecture Notes in Computer Science, *Advances in Neural Networks*, Vol. 3174, pp. 323-339

Igor V. Kovalenko, Glen R. Rippke, & Charles R. Hurburgh (2006). Measurement of soybean fatty acids by near-infrared spectroscopy: Linear and nonlinear calibration methods *Journal of the American Oil Chemists' Society*, 83(5)

İsmail Hakkı Boyacı, Gulum Sumnu, & Ozge Sakiyan (2008). Estimation of Dielectric Properties of Cakes Based on Porosity, Moisture Content, and Formulations Using Statistical Methods and Artificial Neural Networks, *Food Bioprocess Technol* 2(4), pp. 353-360

Jenzsch, Marco, Simutis, Rimvydas, Eisbrenner, Günter, Stückrath, Ingolf, & Lübbert, Andreas (2006). Estimation of biomass concentrations in fermentation processes for recombinant protein production, *Bioprocess and Biosystems Engineering,* 29(1), pp. 19-27

Jose Alberto Gallegos-Enfante, Nuria E.Roha Guzman, Ruben F.Gonzalez-Laredo, & Ramiro Rico- Martinez (2007). The kinetics of crystallization of tripalmitin in olive oil: an artificial neural network approach *Journal of Food Lipids*, 9(1), pp. 73–86

Kılıç, K., Boyacı, İ.-H., Köksel, H., & Küsmenoğlu, İ. (2007). A classification system for beans using computer vision system and artificial neural networks. *Journal of Food Engineering*, 78, pp. 897–904.

References and further reading may be available for this article. To view references and further reading you must this article.

Kulkarni Savita G., Chaudhary Amit Kumar Nandi , Somnath Tambe, & Kulkarni Bhaskar D. (2004). Modeling and monitoring of batch processes using principal component analysis (PCA) assisted generalized regression neural networks (GRNN), *Biochemical Engineering Journal*, 18(3), pp. 193-210

Lenz, J, Hofer, M, Krasenbrink, J, B, & Holker U (2004). A Survey of Computational and physical methods applied to solid state fermentation, *Applied Microbiology and biotechnology*, 65(1), pp. 9-17

Lopes, M.F.S., Pereira C. I., Rodrigues F.M.S., Martins M. P., Mimoso M.C., Barros T. C., Figueiredo Marques J. J., Tenreiro R. P., Almeida J. S., & Barreto Crespo M. T. (1999). Registered designation of origin areas of fermented food products defined by microbial phenotypes and artificial neural networks. *Appl. Environ. Microbiol*., 65, pp. 4484–4489

Lou, W., & Nakai, S. (2001). Application of artificial neural networks for predicting the thermal inactivation of bacteria: A combined effect of temperature, pH and water activity. *Food Research International*, 34, pp. 573–579

Mazzuti Marcio M, Corrazza Marcos L, Filho Francisco Maugeri, Rodrigues Marai Isabel, Corraza Fernanda C, & Triechel Helen (2009). Inulinase production in a batch bioreactor using agroindustrial residues as the substrate: experimental data and modeling, *Bioprocess and Biosystems engineering,* 32(1), pp. 85-95

Meshram C.N. (2008). Studies on dehydration of agro based products using radio frequency dryer, M.Tech (ChemTech.) Thesis sumitted to Nagpur University

Mittal G.S., & Zhang , J. (2000). Prediction of temperature and moisture content of frankfurters during thermal processing using neural network, *Meat Science*, 55(1), pp. 13-24

Molkentin Joachim, Meisel Hans, Lehmann Ines, & Rehbein Hartmut (2007). Identification of Organically Farmed Atlantic Salmon by Analysis of Stable Isotopes and Fatty acids, *European food Research and Technology*, 224 (5) pp. 535-543

Pandharipande, M.S., Pandharipande, S.L., Bhotmange, M.G., & Shastri P.N. (2003). Application of ANN for prediction of Amylase production by *Aspergillus oryzae* under SSF conditions; Proceedings of *International Food Convention"Innovative Food*

Technologies and Quality Systems Strategies for Global Competitiveness "IFCON 2003", Poster no PD 37, pp. 259, Mysore, December 2003.AFST(I), Mysore, India

Pandharipande S. L., & Badhe Y.P.(2003). Software copyright for 'elit-ANN' No. 103/03/CoSw *dated 20/3/03*

Pandharipande S.L. (2004). *Artificial Neural Networks*, Central Techno Publications, Nagpur.

Paquet, J., Lacroix, C., & J. Thibault J, 2000. Modeling of pH and Acidity for Industrial Cheese Production, *J Dairy Sci*., 83, pp. 2393–2409

Peters, G., Morrissey, M., Sylvia, G., & Bolte, J. (1996). Linear Regression, Neural Network and Induction Analysis to Determine Harvesting and Processing Effects on Surimi Quality, *Journal of Food Science*, 61, pp. 876–880

Pramanik K., (2004). Use of Artificial Neural Networks for Prediction of Cell Mass and Ethanol Concentration in Batch Fermentation using *Saccharomyces cerevisiae* Yeast, *IE (I) Journal.CH*, 85, pp. 31-35

Popescu Otelia, popescu Dimitrie, wilder Joseph, & Karwe Mukund (2001). A New Approach To Modelling and Control of a Food Extrusion Process Using Artificial Neural Network and an Expert System, *Journal of Food Process Engineering*, 24(1), pp. 17-36

References and further reading may be available for this article. To view references and further reading you must this article.

Razmi-Rad E., Ghanbarzadeh B., Mousavi S.M., Emam-Djomeh Z., & Khazaei J. (2007). Prediction of rheological properties of Iranian bread dough from chemical composition of wheat flour by using artificial neural networks, *Journal of Food Engineering*, 81(4), pp. 728-734

Razmi-Rad, E., Ghanbarzadeh B., & Rashmekarim, J. (2008). An artificial neural network for prediction of zeleny sedimentation volume of wheat flour. *Int. J. Agri. Biol.,* 10, pp. 422–426

Rousu, J., Elomaa, T., & Aarts, R.J.(1999). Predicting the Speed of Beer Fermentation in Laboratory and Industrial Scale. Engineering Applications of Bio-Inspired Artificial Neural Networks, *Lecture Notes in Computer Science,* 1607, pp. 893–901

Ruan R, Almaer S, & Zhang J (1995). Prediction of Dough Rheological Properties Using Neural Networks, *Cereal Chem*., 72(3), pp. 308-311

Rumelhart, D.E., Hinton, G.E., & Williams, R.J.(1986). Learning internal representations by error propagation. Parallel distributed Processing: Explorations in the Microstructure of Cognition, 1, MIT Press, pp. 318–362

Sablani Shyam S., & M. Shafiur Rahman (2003). Using neural networks to predict thermal conductivity of food as a function of moisture content, temperature and apparent porosity, *Food Research International*, 36,pp. 617–623

Salim Siti Nordiyana Md, Shakaff1 Ali Yeon Md, Ahmad Mohd Noor, Adom Abdul Hamid, & Husin Zulkifl (2005). Development of electronic nose for fruits ripeness determination, 1st International Conference on Sensing Technology, November 21-23, 2005 Palmerston North, New Zealand pp. 515-518

Singh Aruna, Tatewar Divya, Shastri P.N., & Pandharipande S.L. (2008). Application of ANN for prediction of cellulase and xylanase production by *Trichoderma reesei* under SSF conditions, *Indian Journal of Chemical Technology*, 15(1), pp. 53-58.

Singh Aruna, Tatewar Divya, Shastri P.N., & Pandharipande S.L. (2009). Validity of artificial neural network for predicting effect of media components on enzyme production

by *A. niger* in solid state fermentation. *Asian Journal of Microbiology Biotechnology Environmental Science,* 11(4), pp. 777-782

Sousa Ruy, Resende Mariaam M, Giordano Raquel L.C., & Giordano Roberto C, (2003). Hydrolysis of cheese whey proteins by alcalase immobilized in agarose gel particles, *Applied Biochemistry and Biotechnology*, 106(1-3)

Vallejo-Cordoba B., Arteaga G.E., & Nakai S. (1995). Predicting Milk Shelf-life Based on Artificial Neural Networks and Headspace Gas Chromatographic Data, *Journal of Food Science*, 60(5), pp. 885–888

Vassileva S., B. Tzvetkova B., Katranoushkova C. and Losseva L.(2000) Neuro-fuzzy prediction of uricase production *Bioprocess and Biosystems Engineering* 22,(4), pp 363-367

Wongsapat Chokananporn, & Ampawan Tansakul (2008). Artificial Neural Network Model for Estimating the Surface Area of Fresh Guava, *Asian Journal of Food and Agro-Industry*, 1(3), pp. 129 – 136

Yadav Sangeeta, Shastri N.V., Pandharipande S.L., & Shastri P. N. (2003). Optimization of water and DOP level for production of pectin trans eliminase by Penicillium oxalatum under SSF condition by Artificial Neural Network, Proceedings of *International Food Convention"Innovative Food Technologies and Quality Systems Strategies for Global Competitiveness" IFCON 2003*, Poster no FB 28, pp. 48, Mysore , December 2003.(AFSTI) Mysore, India

Yasuo Saito, Toshiharu Hatanaka, Katsuji Uosaki, & Hidekazu Shigeto (2003). Neural Network application to Eggplant Classification, *Lecture Notes in Computer Science*, Vol. 2774, pp. 933-940

Yeh Jeffrey C.H., Hamey Leonard G. C., Westcott Tas, & Sung Samuel K.Y. (2005). In Proceedings of the IEEE International Conference on Neural Networks 2005, pp. 37--42, {IEEE}

Zhang, Jun, & Chen, Yixin (1997). Food sensory evaluation employing artificial neural networks *Sensor Review*, 17(2), pp. 150-158(9)

11

Application of Artificial Neural Networks in Meat Production and Technology

Maja Prevolnik[1,2], Dejan Škorjanc[2],
Marjeta Čandek-Potokar[1,2] and Marjana Novič[3]
[1]*Agricultural Institute of Slovenia, Hacquetova ulica 17, 1000 Ljubljana,*
[2]*University of Maribor, Faculty of Agriculture and Life Sciences, Pivola 10, 2311 Hoče,*
[3]*National Institute of Chemistry, Hajdrihova 19, 1001 Ljubljana,*
Slovenia

1. Introduction

The market of meat and meat products is growing continuously. In the sector of meat, there are many problems and challenges associated with the evaluation of meat quality at industrial level. The methods with the potential of industrial application should be accurate but also rapid, non-destructive, with no health or environment hazards, with benefits of automation and lower risk of human error. The lack of such methods represents a drawback for meat industry and the research focusing on the possible application of rapid methods is emerging. Many new promising techniques are being tested in meat science such as NIR (near infrared) and FT-IR (Fourier transformed infrared) spectroscopy, mass spectrometry, hyper- and multispectral imaging techniques, machine/computer vision, biosensors, electronic noses (array of sensors), ultrasound techniques, *etc.* However, the enormous amount of information provided by these instruments demands an advanced data treatment approach. The artificial intelligent methods can be used for such purposes since their primary target is to distinguish objects or groups or populations. Artificial neural networks (ANN) are a well-known mathematical tool widely used and tested lately for the problems in meat production and technology. Its advantages are in the ability to handle with non-linear data, highly correlated variables and the potential for identification of problems or classification. In particular promising applications of ANN in relation to meat sector is in carcass classification, quality control of raw material, meat processing, meat spoilage or freshness and shelf-life evaluation, detecting off-flavours, authenticity assessment, *etc.* In this chapter an overview of published studies dealing with the application of ANN in meat science is given. In the first part of the chapter basic concepts of artificial neural networks (ANN) are presented and described. The next part of the chapter summarizes the relevant publications on the use of ANN in case of meat production and technology issues and is divided in several paragraphs presenting the relevant research work with the most interesting applications of ANN.

2. Basic concepts of ANN

The ANN is a machine learning method evolved from the idea of simulating the human brain (Rosenblatt, 1961; Zou et al., 2008). Once regarded as an eccentric and unpromising

algorithm for the analysis of scientific data, ANN has been developed into a powerful computational tool in the past decades (Cartwright, 2008) used in many fields of chemistry and biology. The key characteristic of ANN is its ability to learn. Important assets of ANN are related to its ability to handle large data sets, to find out interesting relationships or behaviour among complex data. It is highly adaptable and has an excellent fault tolerance. When a data set is well explained by an appropriate mathematical model (e.g. linear regression), a neural network is unlikely to be needed. It becomes useful in the cases where the rules that underlie the data are not known, or are only partially known. In this case a mechanistic model cannot be derived; instead, a data-driven model may be developed and for this purpose the ANN method is well suited. The functional relationship between input and output is formed during the learning process. This chapter will give only a brief and elementary description of the ANN used in various studies related to meat production and technology. For more detailed information a reader is advised to address the literature specialized in description and mathematical concepts of ANN. Different types of ANN are known, Kohonen, counter-propagation (CP), back-propagation ANN, the latter being the most often applied in studies on meat. Like in the biological neural network, the artificial ANN has an interconnection of neurons with three vital components: i) node character which controls signals *i.e.* the number of inputs and outputs, the weights and activation function associated with the node, ii) network topology defining how nodes are organized and connected and iii) learning rules for the initialization and adjustment of weights. There are two groups of ANN, supervised and unsupervised, which differ in the strategy of learning. In unsupervised learning, the input data is organised and processed without reference to the target, whereas in supervised learning, both the input and target (output) are used. Kohonen ANN is an example of unsupervised learning, where no referential (output) data are used in training of the network, and the algorithms used are excellent for establishing the relationship among complex sets of data. Counter-propagation ANN represents an up-grade of Kohonen ANN and is based on two-step learning procedure, unsupervised in the first step, and supervised in the second. CP-ANN is the most suitable method for classification of data, but can be used also as a method for developing predictive models for new objects of unknown properties. Back-propagation ANN is another example of supervised learning, where one or more target values are predicted from input data, meaning that both inputs and outputs should be known for the training dataset. A special type of ANN is radial basis function network which ordinarily does not involve the training of network, but is determined using a certain transformed function. However, the majority of algorithms work according to an iterative principle, which is similar to training of the network.

2.1 Feed-forward neural network

Feed-forward neural network was the first type of ANN developed. In this network, the information moves only in one direction, forward from the input neurons through the hidden neurons (if any) to the output nodes. There are no cycles or loops in the network. Perceptron (a linear classifier) is the simplest kind of feed-forward ANN. The most popular form is back-propagation (BP) ANN, a multilayer feed-forward network based on back-propagation learning algorithm. The BP-ANN consists of supervised learning algorithm that corrects the weights within each layer of neurons in proportion to the error of the preceding layer level *i.e.* backwards, from the last (output) layer towards the first (input) layer of neurons (Zupan, 1994). Giving the input vectors and targets, this network can approximate a function or classify input vectors in a way defined by the user. Typical BP-ANN has three layers (Fig. 1): the input neurons that receive the information from a data file, the output

neurons that provide a response to the input data, in between are the hidden neurons which communicate with other neurons, they are a part of the internal pattern which provides a solution. In BP-ANN the information flows from one processing element to another within a set of weights. During the training, the interconnections can strengthen or weaken, so that a neural network produces a more correct answer. The number of neurons in the hidden layer influences the number of connections, which affect significantly the network performance and should be optimised. If the number of hidden neurons is too low the learning process can be obstructed, if the number of hidden neurons is too big the network can be over-trained. When developing BP-ANN, besides the mentioned number of neurons in hidden layer, the following parameters of network should be optimized: learning rate (0.1-0.9), momentum term (0.0-1.0), and number of epochs (starting with sample size, optimized on test-set error). When ANN is trained to a satisfactory level, the weighted links among the units are saved and later used as an analytical tool to predict results for a new set of input data.

2.2 Self-organizing maps (SOM) or Kohonen neural networks

Kohonen ANN was initially developed with the aim to mimic human brain functioning. In human brains similar information is stored in certain regions (neighbouring neurons) of cortex. This is related to the mapping of inputs in the Kohonen map which represents a type of unsupervised learning strategy and can be rationalised by the way how young children learn to recognize objects. They do not have to know the words of objects, they just look at the images and they automatically relay *e.g.* the houses in the same group of objects, no matter how many windows or chimneys they have. For the unsupervised learning strategy, only the description of objects are needed, *i.e.* the independent variables for the input vectors. The properties are not given, so the map obtained shows only the relationship between the independent variables of the objects, regardless of their property that may be known, but is not represented in object vectors. The main goal of Kohonen ANN is to project or map objects from *m*-dimensional into two-dimensional space on the basis of input data (similarity among objects). Thus Kohonen ANN is most frequently applied for visualization and clustering purposes.

The Kohonen ANN (Fig. 1) has only one (active) layer of *N* neurons represented as weights $W_j=(w_{j1}, w_{j2}, w_{ji},\ldots w_{jm})$. Each neuron ($j=1...N$) has several weight levels ($i=1...m$). There are as many weight levels as there are input variables. The learning in the Kohonen network is based on unsupervised competitive learning, where only one neuron from the layer is selected for each input. Input is a vector of variables *i.e.* descriptors ($X_s=x_{s1}, x_{s2}, x_{si},\ldots x_{sm}$). The winning neuron W_c is the neuron with weights closest to the input *Xs* according to the Euclidean distance. The weights of the winning neuron and its neighbouring neurons are corrected so that their weights become more similar to the input variable. A trained Kohonen network consists of m-dimensional neurons organised in $N_x \times N_y$ matrix with weights accommodated to the training set objects. Presenting the entire set of objects to the trained network we obtain the locations of the winning neurons in the $N_x \times N_y$ map, excited by individual objects. If we mark the excited neurons in the map by labels corresponding to individual objects, we obtain so-called top-map. The labels can be chosen according to known properties of the objects (e.g. feeding regime, breed, quality class, geographical location). In the top-map one can find clusters of objects, empty spaces (neuron that were not excited by any of the training objects), or conflicts (neurons, excited by two or more

objects from different classes or having different properties). Clusters and empty spaces can be inspected without prior knowledge of property (dependent variables) of the studied objects, while the conflicts can only be determined knowing the properties as well. When developing Kohonen ANN the following parameters of network should be optimised: net size (number of neurons in x and y direction), boundary condition, neighbourhood correction, learning rate (minimal, maximal), number of epochs (starting with sample size, optimized on test-set error), the latest being the most influential parameter.

2.3 Counter-propagation artificial neural networks (CP-ANN)

The CP-ANN (Fig. 1) is based on a two-steps learning procedure, which is unsupervised in the first step. The first step corresponds to the mapping of objects in the input layer (also called Kohonen layer). This part is identical to the Kohonen learning procedure described above. The second step of the learning is supervised, which means that for the learning procedure the response or target value is required for each input. The network is thus trained with a set of input-target pairs $\{X_s,T_s\}$, where T_s is the vector representing dependent variables.

The training of the CP-ANN means adjusting the weights of the neurons in such a way that for each input sample X_s from the training set the network would respond with the output Out_s identical to the target T_s. The training is an iterative procedure similar to the procedure described for the Kohonen neural network, only that dependent variables or target vectors are considered as well. It involves the feeding of all input-output pairs $\{X_s,T_s\}$ to the network, finding the central neuron in the input layer for each X_s, and correction of weights of the neurons, not only in the input but also in the output layer, according to the differences between the targets and current outputs (T_s - Out_s). As already stressed, the targets are needed only in the last part of each iterative learning step. The unsupervised element in the CP-ANN learning procedure is the mapping of the objects vectors into the Kohonen layer, which is based solely on the independent variables, *i.e.* X-part of the $\{X_s,T_s\}$ pairs of the objects from the training set. For this step no knowledge about the target vector (property) is needed. Once the position (central neuron c) of the input vector is defined, the weights of the input and output layer of the CP-ANN are corrected accordingly.

When developing CP-ANN the same network parameters should be optimised as previously explained for Kohonen ANN. Properly trained CP-ANN can be used as predictive models for new objects of unknown properties. First the object is located in the Kohonen layer (on the most similar neuron) regarding the independent variables, which describe the unknown object. Then the position of the neuron is projected to output layer, which gives us the prediction of the sought properties. CP-ANN is also a suitable device for clustering, classification and determination of outliers.

2.4 Differences between CP-ANN and BP-ANN

There are two main differences between CP-ANN and BP-ANN which relate to the learning strategy and the connection between layers (Novič, 2008). Firstly, in contrast to BP-ANN, the learning strategy of CP-ANN is not supervised in all subsequent stages of the training process. The two steps are iteratively repeated for all objects of the training data: (i) finding the position of the object in the two-dimensional map (the most similar neuron in the input or Kohonen layer), which is unsupervised routine based solely on the object representation or independent variables, and (ii) correction of the weights, which also encompasses the output neurons and consequently the property or target values are needed for this purpose.

In this stage, the supervised part is introduced into the training process. Secondly, there is no hidden layer in the CP-ANN. The output (Grossberg) layer is positioned directly below input (Kohonen) layer, with a one-to-one correspondence of neurons. This means that each neuron from the input layer at a position (N_x, N_y) has an ascribed property stored in the output layer at the same position (N_x, N_y). In fact, the output layer, when properly trained, serves as a lookup table for all neurons from the input layer. It has to be stressed here that, in the process of training, all the neurons are affected, not only the central neurons fired by the object. The neighbouring neurons around the central one may remain "unoccupied" at the end of the training; consequently, the output layer contains also values different from the properties of the training objects (interpolated values between those from two occupied neurons). However, there is no chance to obtain predictions out of the range of properties of the training data, which means that extrapolations are not feasible with the CP-ANN. This can be regarded as an advantage, because it prevents unreliable extrapolated predictions, not viable in the experience-based ANN.

2.5 Radial basis function networks

Radial basis function networks (RBF networks) represent a special type of ANN, which are closely related to density estimation methods. A thorough mathematical description of RBF

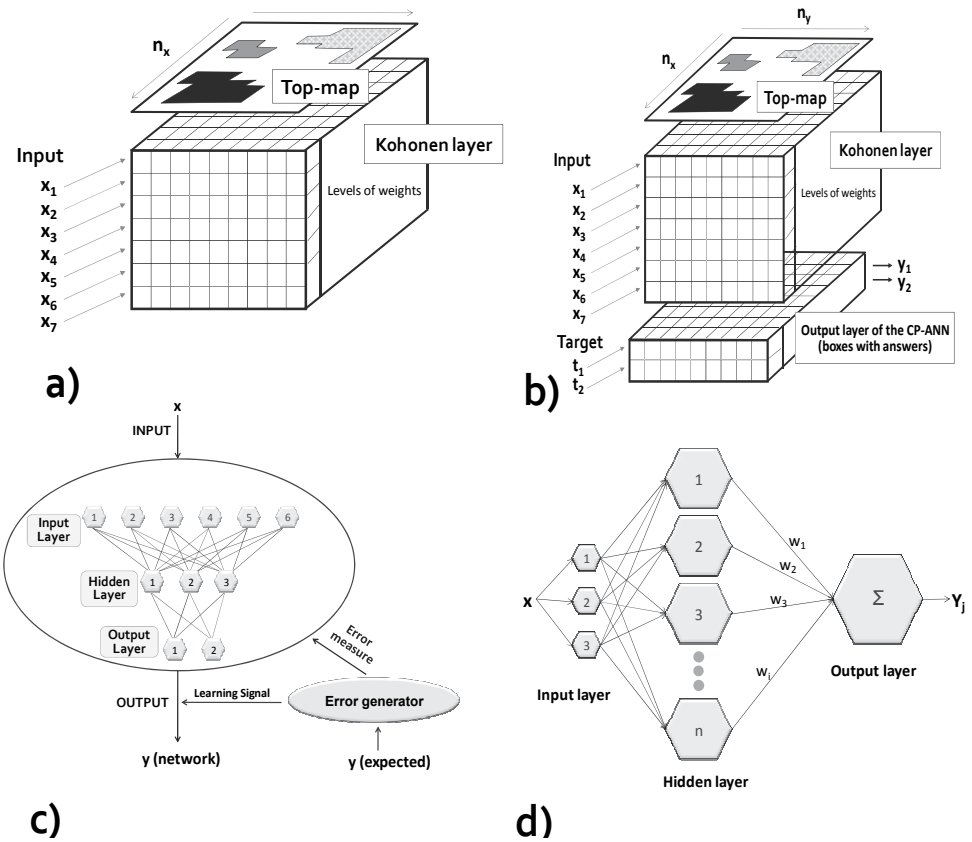


Fig. 1. The structure of the a) Kohonen ANN, b) CP-ANN, c) BP-ANN and d) RBF network

networks is given by Broomhead & Lowe (1988), a short introduction can be found in Lohninger (1993). RBF networks are considered as intermediate between regression models and nearest neighbour classification schemes, which can be looked upon as content-addressable memories (some workers in the field do not regard it as neural networks at all). The behaviour of a RBF network can be controlled by a single parameter which determines if the network behaves more like a multiple linear regression or a content-addressable memory. RBF networks (Fig. 1) have a special architecture, they have only three layers (input, hidden, output) and there is only one layer where the neurons show a nonlinear response (Lohninger, 1999). Some authors have suggested including some extra neurons which serve to calculate the reliability of the output signals (extrapolation flag). The input layer has, as in many other network models, no calculating power and serves only to distribute the input data among the hidden neurons. The hidden neurons show a non-linear transfer function which is derived from Gaussian bell curves. The output neurons in turn have a linear transfer function which makes it possible to simply calculate the optimum weights associated with these neurons.

3. Novel technologies using ANN in meat quality evaluation and control

Meat quality is a very complex term and it comprises various aspects which can differ according to the user's standpoint *i.e.* different factors or properties are important for animal producer, meat processor or consumer. From the animal production perspective the quality mainly refers to lean meat content on which the payment to the farmer is based. Processing industry on the other hand is interested in meat technological quality (suitability for further processing) and factors affecting consumer's choice. The consumer is sensitive about meat appearance (colour, lean to fat ratio), its sensory quality, nutritional value (macro and micro nutrients) and safety (presence/absence of toxic compounds, drugs, and pathogen or spoilage micro flora). Other factors like the way meat is produced (animal welfare, ecology) can also affect consumer's choice. In meat production and technology, different properties can play an important role in quality classification of meat for different purposes or can be critically appraised by consumers (often their basis for meat selection or rejection). In pork for example, water-holding capacity of meat has big significance, whereas in beef, tenderness is an important attribute. Spoilage detection or meat shelf-life is also an important issue in meat sector. In the last decades, the methods used in meat evaluation, meat quality control, or inspection have undergone important developments with the application of novel technologies like computer (machine) vision, spectral imaging, spectroscopy, electronic nose or bio-sensing technologies. Since the application of ANN in meat science and technology is mainly associated with novel technologies, a brief presentation of technologies encountered is given.

Electronic nose (also electronic sensing or e-sensing) is an array of electronic chemical sensors with partial specificity and an appropriate pattern-recognition system, capable of detecting specific or complex odours (Craven et al., 1996). These instruments contain an array of sensors that utilize various technologies like organic polymers, metal oxides (Harper, 2001). The recognition process is similar to human olfaction and is performed for identification, comparison, quantification and other applications. These instruments show potential but are presently still in developmental phase due to many weaknesses (sensitivity to humid conditions, high alcohol concentration, instrumental drift, sensor span life) that should be overcome (Harper, 2001).

Computer vision is concerned with the theory behind artificial systems that extract information from images. The image data can take many forms, such as video sequences,

views from multiple cameras, or multi-dimensional data from a medical scanner. The application of computer vision in the industry, where information is extracted for the purpose of supporting a manufacturing process, is called machine vision. Ultrasonography is a kind of imaging technique which uses ultrasound for diagnostic purposes. The reflection signature can reveal details about the inner structure of the medium. Spectral imaging or spectral analysis comprises different techniques such as hyper-, multi- and ultraspectral imaging. In contrast to the human eye, which can see only visible light, hyperspectral imaging collects and processes information from across the electromagnetic spectrum. Certain objects leave unique 'fingerprints' across electromagnetic spectrum. The differences among hyper-, multi- and ultraspectral imaging are based mainly on the type of measurements *i.e.* discrete or continuous bands, broad or narrow bands.

Near infrared (NIR) spectroscopy is a spectroscopic method which extracts the information about chemical and physical properties of organic substances on the basis of vibrations of bounds caused by NIR light (800 nm to 2500 nm). The characteristics of NIR spectra are molecular overtones and combination vibrations which are typically very broad in this part of the electromagnetic spectrum. It can be very useful in probing bulk material with little or no sample preparation.

Bio-sensing technology combines a sensitive biological element (e.g. enzymes, microorganisms, antibodies, *etc.*) with a physicochemical detector of an analyte (optical, piezoelectric, electrochemical). The physicochemical detector transforms the interaction of the analyte with the biological element into a signal which can be measured and quantified. The results are displayed in a user-friendly way.

The mentioned techniques generally produce enormous amounts of very complex information (spectra, images, *etc.*) which require sophisticated data treatment *i.e.* multivariate calibration methods. Due to its dynamic self-adapting system using a learning strategy ANN is able of pattern recognition, dealing with complexity of data and non-linear relationships, performing complex prediction and classification tasks. ANN has thus been applied also for solving the problems in meat science and technology. New methods were developed to either complete or replace subjective sensory testing (e.g. analysis of odour or flavour), to handle complex properties (e.g. meat tenderness), to speed up the process or replace human operator in on-line inspection. Literature review (Tables 1-4) demonstrates examples of successful or promising applications of ANN in meat technology in association with novel technologies.

4. Application of ANN in meat quality evaluation and meat chemical composition analysis

Artificial intelligence methods (ANN) were mainly investigated for the evaluation of meat sensory quality *i.e.* the properties that are subjectively evaluated or classified such as tenderness (Li et al., 1999; Li et al., 2001; Tian et al., 2005; Chandraratne et al., 2006), colour (Santé et al., 1996; Lu et al. 2000; Tan et al., 2000; Sheridan et al., 2007) or marbling score/level (Brethour, 1994; Qiao et al., 2007a). There were also studies dealing with water-holding capacity of pork (Prevolnik et al., 2009; Qiao et al., 2007b), quality of meat products (Dong, 2009; Valous et al., 2010) and categorization to different pork (Qiao et al., 2007a) or beef quality classes (Shiranita et al, 2000). The majority of studies were carried out on beef and pork (Table 1), and only a few of them to other *species* such as poultry (Santé et al., 1996) and lamb (Sebastian et al., 2004; Chandraratne et al., 2006). Contrary to the frequent use of ANN for meat quality assessment, this approach was seldom used for the prediction of meat chemical properties (Mittal & Zhang, 2000; Sebastian et al., 2004; Prevolnik et al., 2009). In the studies of

meat quality assessment, variants of computer (machine) vision were often applied. The overview of the studies (Table 1) shows that ANN was used for the assessment of meat properties based on digital images of meat surface (Li et al., 1999; Lu et al., 2000; Shiranita et al., 2000; Tan et al., 2000), or based on near infrared spectra (Prevolnik et al., 2009), mass spectroscopy (Sebastian et al., 2004), hyperspectral imaging (Qiao et al., 2007a,b), ultrasonography (Brethour, 1994; Huang et al., 1998). Only few studies based application of ANN for meat quality assessment using just several simple physical measurements of meat (Santé et al., 1996; Prevolnik et al., 2009) or carcass traits (Hill et al., 2000). In the vast majority of the reported studies a supervised learning strategy of ANN (multi-layer perceptron neural networks with back-propagation learning, back-propagation ANN, feed-forward ANN, multi-layer perceptron) was used for addressing the issues of meat quality and composition, denoting an interest for prediction ability. There was only one study (Prevolnik et al., 2009) where a combination of unsupervised (Kohonen ANN) and supervised learning (CP-ANN and BP-ANN) and was used. Generally the presented studies (an overview is given in Table 1) demonstrate good results, and an improvement when compared to other multivariate techniques of data the analysis. The accuracy of classification

OBJECTIVE	SAMPLE	INPUT DATA	RESULTS	REFERENCE
$C_{Marbling}$	Bovine LD n=161	Ultrasonography, pattern recognition	84% correctness	Brethour, 1994
$C_{Meat\ colour}$	Turkey breast, n=68+40	pH, L^*, a^*, b^*, T, haem pigment, dielectric loss factor	70% correctness	Santé et al., 1996
$P_{WBSF,\ fat,\ moisture,\ collagen,\ sacromere\ length,\ calpastatine}$	Bovine LD	Wavelet textural features from ultrasonic elastograms	R^2=0.91-0.99	Huang et al., 1998
$P_{Cooked\ meat\ tenderness}$	Bovine loin n=97	Computer vision (digital colour image of meat)	R^2=0.70	Li et al., 1999
C,P_{WBSF}	Bovine LTL n=1452	Carcass traits	P_{R^2}=0.37-0.45 $C_{51-53\%}$	Hill et al., 2000
$P_{Meat\ colour}$	Pork LD n=44	Computer vision, image analysis	R^2=0.56	Lu et al., 2000
$P_{Temperature\ and\ moisture\ during\ cooking}$	Frankfurters	ratio fat/protein, initial & ambient T, radius, initial moisture, relative humidity	System is convenient and accurate	Mittal & Zhang, 2000
$C_{Meat\ grade}$	Bovine loin n=36	Image processing	Effective system, difference in grades < 1	Shiranita et al., 2000
$C_{Meat\ colour}$	Pork LD n>200	Colour machine vision	86% correctness	Tan et al., 2000
$C_{Meat\ tenderness\ (tough\ or\ tender)}$	Bovine loin n=59	Image texture analysis	83% correctness	Li et al., 2001
$P_{WBSF,\ collagen\ and\ lipid\ content}$	Lamb LD n=120	Curie point pyrolysis-mass spectrometry	r=0.85-0.90, 10-12% error	Sebastian et al., 2004

Table 1. The application of ANN in meat chemical composition and quality analysis

OBJECTIVE	SAMPLE	INPUT DATA	RESULTS	REFERENCE
^{P}Cooked meat tenderness	Bovine LD n=50	Computer vision technology	R^2=0.62	Tian et al., 2005
^{P}WBSF	Lamb loin n=160	Image surface texture analysis	R^2=0.62-0.75	Chandraratne et al., 2006
^{P}Moisture content	Cooked bovine joints	Computer vision (colour features)	r=0.75	Zheng et al., 2007
^{C}Pork quality class, marbling	Pork loin n=40	Hyperspectral imaging	>70 % correctness	Qiao et al., 2007a
^{P}Drip loss, L*, pH ^{C}WHC classes	Pork loin n=80	Hyperspectral imaging	^{P}r=0.77, 0.55 and 0.86 for drip loss, pH and L*, respectively ^{C}successful	Qiao et al., 2007b
^{C}Discoloration (fading)	Cured ham	L*, a*, b or spectral reflectance	Successful discriminating different stages of fading	Sheridan et al., 2007
^{P}WHC (drip loss)	Pork LD n=312	pH, L*, a*, b* NIR spectroscopy	R^2=0.37-0.51, error=2.2-2.5%	Prevolnik et al., 2009
Sensory texture	Cooked sausage	Instrumental texture measurements	Lower errors as compared to regression analysis	Dong, 2009
Quality class	Cooked ham	Computer vision	84-96% correctness	Valous et al., 2010

LD – *longissimus dorsi*; TB – *triceps brahii*; LTL – *longissimus thoracis et lumborum*; R^2 – coefficient of determination; r – correlation coefficient; P – prediction; C – classification; WHC – water holding capacity; WBSF – Warner-Bratzler shear force; NIR – near infrared.

Table 1. Continued. Application of ANN in meat chemical composition and quality analysis reported is high (70 to 85 %). In case where ANN approach was used for prediction, the results varied from moderate to excellent; however, for the most part the authors consider application of ANN as promising and successful.

5. Application of ANN for carcass quality or classification

Meat industry is interested in lean and conformed carcasses which provide high meat yields. The so called carcass grading or classification (used for pig, bovine, lamb carcasses) is performed at the end of the slaughter line and represents a basis for the payment to the farmer. Another example is in poultry, where the carcasses are inspected at the slaughter line for the wholesomeness and those with an abnormal aspect (tumorous, bruised, skin-torn, septicemic, cadaver, air-sacculitis) are discarded. The mentioned procedures are mostly based on the visual appraisal and thus subjected to human limitations (speed, error, fatigue). The overview of the problems encountered in this field of research, where possible application of

ANN was investigated, is given in Table 2. In regard to carcass classification of domestic mammals, the research was mainly focused on either improving or replacing methods that are currently used. Many studies were carried out on classification or carcass quality evaluation in bovine carcasses (Borggaard et al., 1996; Hwang et al., 1997; Díez et al., 2003; Hatem et al., 2003; Lu & Tan, 2004), but also for lamb (Chandraratne et al., 2007) or goat (Peres et al., 2010). In these *species* the principle of grading is similar and consists of visual notes given by the classifier, which are the indicators of lean meat quantity. In these cases the aim was either to predict carcass lean meat content (Hwang et al., 1997; Berg et al., 1998; Lu & Tan, 2004) or to replace the classifier using automated grading (Borggaard et al., 1996).

OBJECTIVE	SAMPLE	INPUT DATA	RESULTS	REFERENCE
Inspection – carcass wholesomeness	Poultry n=87	Multispectral imaging	83-97% success in sorting	Park & Chen, 1994
$P_{}$Conformation, fatness, fat colour, rib eye area, saleable meat %	Bovine n=3,500	Computer vision	R^2=0.66-0.93, 20% lower error as classifier	Borggaard et al., 1996
Inspection – carcass wholesomeness	Poultry n=559	VIS-NIR spectroscopy	93–97% success in sorting	Chen et al., 1996
Inspection – carcass wholesomeness	Poultry n=288	Multispectral imaging	91% success in sorting	Park et al., 1996
Inspection – carcass wholesomeness	Poultry	VIS-NIR spectroscopy	>95% success in sorting	Chen et al., 1998a
Inspection – carcass wholesomeness	Poultry	VIS-NIR spectroscopy	98% success in sorting	Chen et al., 1998b
Lean meat content prediction (carcass and prime cuts)	Pig n=50	Electromagnetic scanning	Improvement in comparison to linear regression	Berg et al., 1998
Inspection – carcass wholesomeness	Poultry n=91	Multispectral imaging	90-93% success in sorting	Park et al., 1998
Inspection – carcass wholesomenes	Poultry	Machine vision (dual-camera)	80-100% success in sorting	Chao et al., 2000

Table 2. Application of ANN for carcass classification

Other studied applications were interested in prediction of fat depots based on *in vivo* measurements (Peres et al., 2010) or prediction of carcass maturity (Hatem et al., 2003). In the case of pig classification, studies using ANN are rare (Berg et al., 1998), probably because the current classification methods are based on objective measurements on the carcass which are well correlated to lean meat content thus providing sufficient accuracy using standard regression methods. There was an interesting study in bovine carcass classification addressing the problem of classifier effect and repeatability in bovine carcass grading (Díez et al., 2003), demonstrating another possible application of ANN for the purposes of monitoring. Much work has also been devoted to the automatic inspection of wholesomeness of chicken carcasses using different optical techniques (Park & Chen, 1994; Chen et al., 1996, Park et al., 1996, 1998; Chen et al., 1998a,b; Chao et al., 2000, 2002; Ibarra et

OBJECTIVE	SAMPLE	INPUT DATA	RESULTS	REFERENCE
Inspection – carcass wholesomeness	Poultry	Spectral imaging	83–91% success in sorting	Park & Chen, 2000
Inspection – carcass wholesomeness	Poultry n=14,591	Machine vision (dual-camera)	87-94% success in sorting	Chao et al., 2002
Inspection – carcass wholesomeness-diseased air sacks	Poultry n=100	Computer vision - color classification	97% success rate	Ibarra et al., 2002
Classifier effect and repeatability	Bovine n=227	Computer vision (image analysis)	Higher uncertainty when grading light than standard carcasses	Díez et al., 2003
Skeletal maturity grading	Bovine cartilage n=138	Machine vision (colour features of cartilage)	65-75% correctness	Hatem et al., 2003
Lean weight and lean percentage prediction	Bovine n=241	Computer vision (image analysis)	No advantage to linear methods	Lu & Tan, 2004
Carcass grading	Lamb n=160	Computer vision (image analysis)	87-100% correctness	Chandraratne et al., 2007
Fat depots assessment	Goats, n=56	Ultrasound technology	R^2=0.82-0.96, RPD=1.7-4.3	Peres et al., 2010

LD – *longissimus dorsi*; VIS – visible; NIR – near infrared; R^2 – coefficient of determination; r – correlation coefficient; P – prediction; C – classification; RPD – residual predictive deviation.

Table 2. Continued. Application of ANN for carcass classification

al., 2002). The usefulness of ANN as coupled with computer vision for such purposes has been demonstrated by several studies. The success rate of such classification is very high, typically above 90%. In all studies dealing with carcass classification or inspection a supervised learning strategy was applied, mainly BP-ANN, with the exception of a few studies using other types of ANN such as RBF networks (Peres et al., 2010) or learning vector quantization (Ibarra et al., 2002). In general, ANN showed its potential and advantage over conventional regression methods especially in case of non-linearity between system inputs and outputs.

6. Application of ANN for spoilage identification/storage time assessment

Meat and meat products are highly susceptible to spoilage or contamination, affecting the quality and safety of the products. Many of the methods used for the detection of spoiled or contaminated meat are based on immunological or nucleic acid based procedures which are time consuming, laborious and demand trained personnel. At present no method is available for a real-time, non-destructive, reagentless, quantitative and relatively inexpensive monitoring. According to Ellis & Goodacre (2001) interesting analytical approaches include biosensors, electronic noses, infrared spectroscopy upgraded with machine learning methods (ANN, genetic algorithms).

OBJECTIVE	SAMPLE	INPUT DATA	RESULTS	REFERENCE
$P_{storage\ time,\ spoiled\ meat}$	Ground beef, pork n=20	Electronic nose	Successful	Winquist et al., 1993
$P_{Meat\ freshness}$	Chicken	Electronic nose	Successful prediction of storage time	Galdikas et al., 2000
$P_{Bacterial\ growth}$ (*L. sake)*	Cooked meat products	T, a_w, CO_2	Max. specific growth rate R^2=0.94, RMSE=0.011 Lag phase λ R^2=0.97, RMSE=6.70	Lou & Nakai, 2001
$P_{Bacterial\ growth}$ (*L. monocytogenes)*	Meat broth	Fluctuating conditions (T, pH, NaCl, a_w)	ANN can be used to describe/predict bacterial growth in dynamic conditions	Cheroutre-Vialette & Lebert, 2002
$P_{Internal\ temperature\ estimation}$	Chicken n=85	IR and laser range imaging	R^2=0.94-0.96	Ma & Tao, 2005
$P_{Shelf\text{-}life\ estimation}$	Cooked meat products	T, pH, NaCl, $NaNO_2$	Error, bias and accuracy factors show successful validation	Zurera-Cosano et al., 2005
$C_{Identification\ of\ spoiled\ meat}$	Bovine LD n=156	Electronic nose	83-100% correctness	Panigrahi et al., 2006
$P_{Survivival\ of}$ *Escherichia coli*	Fermented sausage	pH, a_w, iso-thiocyanate concentration	Accurate ANN based models	Palanichamy et al., 2008
$C,P_{Meat\ spoilage\ identification}$	Bovine LD n=156	Electronic nose	Sorting accuracy >90% Microbial count R^2>0.70	Balasubramanian et al., 2009
$C,P_{Spoilage\ identification}$	Beef fillets n=74	FT-IR spectroscopy	Sorting accuracy 81-94% Satisfactory prediction of microbial counts	Argyri et al., 2010

LD – *longissimus dorsi*; R^2 – coefficient of determination; r – correlation coefficient; P – prediction; C – classification; IR – infrared.

Table 3. Application of ANN for spoilage or storage time prediction

7. Various other applications of ANN in meat science and technology

In addition to the mentioned subjects of interest for ANN application in meat science there are various other applications related to meat technology issues (Table 4). These involve identification of animal *species* in ground meat mixtures (Winquist et al., 1993) or fat tissue (Beattie et al., 2007), recognition of animal origin (distinction between Iberian and Duroc

OBJECTIVE	SAMPLE	INPUT DATA	RESULTS	REFERENCE
Species recognition	Ground beef, pork, n=20	Electronic nose	Successful	Winquist et al., 1993
Visual guidance of evisceration	Pig carcasses	Computer vision	Efficient ANN based system	Christensen et al., 1996
Lean tissue extraction (image segmentation)	Bovine LD n=60	Computer vision (hybrid image)	Better efficiency and robustness of ANN based system	Hwang et al., 1997
Fermentation monitoring	Sausage	Electronic nose	Lowest error in case of ANN compared to regression	Eklöv et al., 1998
Estimation of meat internal T	Cooked chicken meat	IR imaging	Great potential for monitoring of meat doneness (error of ±1°C)	Ibarra et al., 2000
Determination of RN^- phenotype	Pig n=96	NIR spectroscopy	96% correctness	Josell et al., 2000
Identification of feeding and ripening time	Pig; dry-cured ham	Electronic nose	Best prediction for N at 250°C; misclassified hams ≈8%	Santos et al., 2004
Species recognition on adipose tissue	Lamb, beef chicken,pork n=255	Raman spectroscopy	>98% correctness	Beattie et al., 2007
$P_{Cooking\ shrinkage}$	Bovine TB n=25	Computer vision technique	r=0.52-0.75	Zheng et al., 2007
Walk-through weighing	Pigs	Machine vision	relative error ≈3%	Wang et al., 2008
Differentiation of Iberian and Duroc	Pigs n=30	VIS-NIR spectroscopy	>95% correctness	del Moral et al., 2009

LD – *longissimus dorsi*; TB – *triceps brachii*; R^2 – coefficient of determination; r – correlation coefficient; P – prediction; C – classification; VIS – visible; NIR – near infrared; IR - infrared.

Table 4. Other applications of ANN in meat science and technology

pigs) as affected by rearing regime and/or breed (del Moral et al., 2009), hybrid image processing for lean tissue extraction (Hwang et al., 1997), detection of RN- phenotype in pigs (Josell et al., 2000), the "walk-through" weighing of pigs (Wang et al., 2008), the efficiency of ANN for visual guidance of pig evisceration at the slaughter line (Christensen et al., 1996) and the use of ANN for the processing control of meat products (Eklöv et al., 1998; Ibarra et al., 2000; Santos et al., 2004). Again, in the majority of studies, ANN approach was an instrument to deal with the complex output signal of novel technologies applied. Again, based on the literature reports, supervised learning strategy of ANN (BP-ANN, RBF) was applied in the majority of studies. There were also a few studies where unsupervised learning has been tested (Winquist et al., 1993; Beattie et al., 2007). A bibliographic overview given in Table 4 demonstrates the efficiency and successful classification rate of ANN based systems.

8. Conclusions and future perspectives

The existing research work of ANN application in meat production and technology provided many useful results for its application, the majority of them in association with novel technologies. Among interesting ideas that have not been encountered in the literature review is the combination of ANN with bio-sensing technology. ANN shows great potential for carcass and meat (product) quality evaluation and monitoring under industrial conditions or bacterial growth and shelf-life estimation. However, the potentially interesting relevance of ANN, for which the literature information is scarce, is its application for meat authenticity or meat (product) quality forecast based on the information from rearing phase. Overall the presented applications are relatively new and the full potential has not yet been discovered.

9. References

Argyri, A. A., Panagou, E. Z., Tarantilis, P. A., Polysiou, M. & Nychas, G. J. E. (2010). Rapid qualitative and quantitative detection of beef fillets spoilage based on Fourier transform infrared spectroscopy data and artificial neural networks. *Sensors and Actuators B-Chemical*, 145, 1, 146-154, ISSN: 0925-4005

Balasubramanian, S., Panigrahi, S., Logue, C. M., Gu, H. & Marchello, M. (2009). Neural networks-integrated metal oxide-based artificial olfactory system for meat spoilage identification. *Journal of Food Engineering*, 91, 1, 91-98, ISSN: 0260-8774

Beattie, J. R., Bell, S. E. J., Borggaard, C., Fearon, A. M. & Moss, B. W. (2007). Classification of adipose tissue species using Raman spectroscopy. *Lipids*, 42, 7, 679-685, ISSN: 0024-4201

Berg, E. P., Engel, B. A. & Forrest, J. C. (1998). Pork carcass composition derived from a neural network model of electromagnetic scans. *Journal of Animal Science*, 76, 1, 18-22, ISSN: 0021-8812

Borggaard, C., Madsen, N. & Thodberg, H. (1996). In-line image analysis in the slaughter industry, illustrated by beef carcass classification. *Meat Science*, 43, 151-163, ISSN: 0309-1740

Brethour, J. (1994). Estimating marbling score in live cattle from ultrasoud images using pattern-recognition and neural-network procedures. *Journal of Animal Science*, 72, 6, 1425-1432, ISSN: 0021-8812

Broomhead, D. S. & Lowe, D. (1998). Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2, 312-355, ISSN: 0891-2513

Cartwright, H. M. (2008). Artificial neural networks in biology and chemistry. In: *Artificial neural networks : methods and applications*. Livingstone, D. (Ed.), 1-13, Humana Press, ISBN: 978-1-58829-718-1, New York

Chandraratne, M. R., Samarasinghe, S., Kulasiri, D. & Bickerstaffe, R. (2006). Prediction of lamb tenderness using image surface texture features. *Journal of Food Engineering*, 77, 3, 492-499, ISSN: 0260-8774

Chandraratne, M., Kulasiri, D. & Samarasinghe, S. (2007). Classification of lamb carcass using machine vision: Comparison of statistical and neural network analyses. *Journal of Food Engineering*, 82, 1, 26-34, ISSN: 0260-8774

Chao, K., Park, B., Chen, Y. R., Hruschka, W. R. & Wheaten, F. W. (2000). Design of a dual-camera system for poultry carcasses inspection. *Applied Engineering in Agriculture*, 16, 5, 581-587, ISSN: 0883-8542

Chao, K., Chen, Y. R., Hruschka, W. R. & Gwozdz, F. B. (2002). On-line inspection of poultry carcasses by a dual-camera system. *Journal of Food Engineering*, 51, 3, 185-192, ISSN: 0260-8774

Chen, Y. R., Huffman, R. W., Park, B. & Nguyen, M. (1996). Transportable spectrophotometer system for on-line classification of poultry carcasses. *Applied Spectroscopy*, 50, 7, 910-916, ISSN: 0003-7028

Chen, Y. R., Nguyen, M. & Park, B. (1998a). Neural network with principal component analysis for poultry carcass classification. *Journal of Food Process Engineering*, 21, 5, 351-367, ISSN: 0145-8876

Chen, Y. R., Park, B., Huffman, R. W. & Nguyen, M. (1998b). Classification o in-line poultry carcasses with back-propagation neural networks. *Journal of Food Processing Engineering*, 21, 1, 33-48, ISSN: 1745-4530

Cheroutre-Vialette, M. & Lebert, A. (2002). Application of recurrent neural network to predict bacterial growth in dynamic conditions. *International Journal of Food Microbiology*, 73, 2-3, 107-118 ISSN: 0168-1605

Christensen, S. S., Andersen, A. W., Jørgensen, T. M. & Liisberg, C. (1996). Visual guidance of a pig evisceration robot using neural networks. *Pattern Recognition Letters*, 17, 4, 345-355, ISSN: 0167-8655

Craven, M. A., Gardner, J. W. & Bartlett, P. N. (1996). Electronic noses – Development and future prospects. *Trends in Analytical Chemistry*, 15, 9, 486-493, ISSN: 0167-2940

Del Moral, F. G., Guillén, A., del Moral, L. G., O'Valle, F., Martínez, L. & del Moral, R. G. (2009). Duroc and Iberian pork neural network classification by visible and near infrared reflectance spectroscopy. *Journal of Food Engineering*, 90, 4, 540-547, ISSN: 0260-8774

Díez, J., Bahamonde, A., Alonso, J., López, S., del Coz, J. J., Quevedo, J. R., Ranilla, J., Luaces, O., Alvarez, I., Royo, L. J. & Goyache, F. (2003). Artificial intelligence techniques point out differences in classification performance between light and standard bovine carcasses. *Meat Science*, 64, 3, 249-258, ISSN: 0309-1740

Dong, Q. L. (2009). BP neural network for evaluating sensory texture properties of cooked sausage. *Journal of Sensory Studies*, 24, 6, 833-850, ISSN: 0887-8250

Eklöv, T., Johansson, G., Winquist, F. & Lundström, I. (1998). Monitoring sausage fermentation using an electronic nose. *Journal of the Science of Food and Agriculture*, 76, 4, 525-532, ISSN: 0022-5142

Ellis, D. I. & Goodacre, R. (2001). Rapid and quantitative detection of the microbial spoilage of muscle foods: Current status and future trends. *Trends in Food Science & Technology*, 12, 11, 414-424, ISSN: 0924-2244

Galdikas, A., Mironas, A., Senulienė, D., Strazdienė, V., Šetkus, A. & Zelenin, D. (2000). Response time based output of metal oxide gas sensors applied to evaluation of meat freshness with neural signal analysis. *Sensors and Actuators B: Chemical*, 69, 3, 258-265, ISSN: 0925-4005

Harper, W. J. (2001). The strengths and weaknesses of the electronic nose. In: *Headspace analysis of foods and flavours*, Rouseff R. L. & Cadwallader K. R. , (Ed.), 59-71, Kluwer Academic/Plenum Publ., ISBN: 978-0-306-46561-1, New York

Hatem, I., Tan, J. & Gerrard, D. E. (2003). Determination of animal skeletal maturity by image processing. *Meat Science*, 65, 3, 999-1004, ISSN: 0309-1740

Hill, B. D., Jones, S. D. M., Robertson, W. M. & Major, I. T. (2000). Neural network modeling of carcass measurements to predict beef tenderness. *Canadian Journal of Animal Science*, 80, 2, 311-318, ISSN: 0008-3984

Huang, Y., Lacey, R. & Whittaker, A. (1998). Neural network prediction modeling based on elastographic textural features for meat quality evaluation. *Transactions of the ASAE*, 41, 4, 1173-1179, ISSN: 0001-2351

Hwang, H., Park, B., Nguyen, M. & Chen, Y. R. (1997). Hybrid image processing for robust extraction of lean tissue on beef cut surfaces. *Computers and Electronics in Agriculture*, 17, 3, 281-294, ISSN: 0168-1699

Ibarra, J. G., Tao, Y. & Xin, H. W. (2000). Combined IR imaging-neural network method for the estimation of internal temperature in cooked chicken meat. *Optical Engineering*, 39, 11, 3032-3038, ISSN: 0091-3286

Ibarra, J. G., Tao, Y., Newberry, L. & Chen, Y. R. (2002). Learning vector quantization for color classification of diseased air sacs in chicken carcasses. *Transactions of the ASAE*, 45, 5, 1629-1635, ISSN: 0001-2351

Josell, Å., Martinsson, L., Borggaard, C., Andersen, J. R. & Tornberg, E. (2000). Determination of RN- phenotype in pigs at slaughter-line using visual and near-infrared spectroscopy. *Meat Science*, 55, 3, 273-278, ISSN: 0309-1740

Li, J., Tan, J., Martz, F. A. & Heymann, H. (1999). Image texture features as indicators of beef tenderness. *Meat Science*, 53, 1, 17-22, ISSN: 0309-1740

Li, J., Tan, J. & Shatadal, P. (2001). Classification of tough and tender beef by image texture analysis. *Meat Science*, 57, 4, 341-346, ISSN: 0309-1740

Lohninger, H. (1993). Evaluation of neural networks based on radial basis functions and their application to the prediction of boiling points from structural parameters. *Journal of Chemical Information and Computer Sciences*, 33, 736-744, ISSN: 0095-2338

Lohninger, H. (1999). Teach/Me Data Analysis, Springer-Verlag, Berlin-New York-Tokyo, ISBN: 978-3-54014-743-5

Lou, W. & Naka, S. (2001). Artificial Neural Network-based predictive model for bacterial growth in a simulated medium o modified-atmosphere-packed coked meat products. *Journal of Agricultural and Food Chemistry*, 49, 4, 1799-1804, ISSN: 0021-8561

Lu, J., Tan, J., Shatadal, P. & Gerrard, D. E. (2000). Evaluation of pork color by using computer vision. *Meat Science*, 56, 1, 57-60, ISSN: 0309-1740

Lu, W. & Tan, J. (2004). Analysis of image-based measurements and USDA characteristics as predictors of beef lean yield. *Meat Science*, 66, 2, 483-491, ISSN: 0309-1740

Ma, L. & Tao, Y. (2005). An infrared and laser range imaging system for non-invasive estimation of internal temperatures in chicken breasts during cooking. *Transactions of the ASAE, 48, 2, 681-690*, ISSN: 0001-2351

Mittal, G. S. & Zhang, J. (2000). Prediction of temperature and moisture content of frankfurters during thermal processing using neural network. Meat Science, 55, 1, 13-24, ISSN: 0309-1740

Novič, M. (2008). Kohonen and counter-propagation neural networks applied for mapping and interpretation of IR spectra. In: *Artificial neural networks : methods and applications*. Livingstone, D. (Ed.), 45-60, Humana Press, ISBN: 978-1-58829-718-1, New York

Palanichamy, A., Jayas, D. S. & Holley, R. A. (2008). Predicting survival of Escherichia coli O157 : H7 in dry fermented sausage using artificial neural networks. *Journal of Food Protection*, 71, 1, 6-12, ISSN: 0362-028X

Panigrahi, S., Balasubramanian, S., Gu, H., Logue, C. M. & Marchello, M. (2006). Design and development of a metal oxide based electronic nose for spoilage classification of beef. *Sensors and Actuators B: Chemical*, 119, 1, 2-14, ISSN: 0925-4005

Park, B. & Chen, Y.-R. (1994). Intensified multispectral imaging system for poultry carcass inspection. *Transactions of the ASAE*, 37, 6, 1983-1988, ISSN: 0001-2351

Park, B. Chen, Y. R., Nguyen, M. & Hwang, H. (1996). Characterizing multispectral images of tumorous, bruised, skin-torn, and wholesome poultry carcasses. *Transactions of the ASAE*, 39, 5, 1933-1941, ISSN: 0001-2351

Park, B., Chen, Y. R. & Nguyen, M. (1998). Multi-spectral Image Analysis using Neural Network Algorithm for Inspection of Poultry Carcasses. *Journal of Agricultural Engineering Research*, 69, 4, 351-363, ISSN: 1095-9246

Park, B. & Chen, J. Y. (2000). Real-time dual-wavelength image processing for poultry safety inspection. *Journal of Food Processing Engineering*, 23., 5., 329-351, ISSN: 1745-4530

Peres, A. M., Dias, L. G., Joy, M. & Teixeira, A. (2010). Assessment of goat fat depots using ultrasound technology and multiple multivariate prediction models. *Journal of Animal Science*, 88, 2, 572-580, ISSN: 0021-8812

Prevolnik, M., Čandek-Potokar, M., Novič, M. & Škorjanc, D. (2009). An attempt to predict pork drip loss from pH and colour measurements or near infrared spectra using artificial neural networks. *Meat Science*, 83, 3, 405-411, ISSN: 0309-1740

Qiao, J., Ngadi, M. O., Wang, N., Gariépy, C. & Prasher, S. O. (2007a). Pork quality and marbling level assessment using a hyperspectral imaging system. *Journal of Food Engineering*, 83, 1, 10-16, ISSN: 0260-8774

Qiao, J., Wang, N., Ngadi, M. O., Gunenc, A., Monroy, M., Gariépy, C. & Prasher, S.O. (2007b). Prediction of drip-loss, pH, and color for pork using a hyperspectral imaging technique. *Meat Science*, 76, 1, 1-8, ISSN: 0309-1740

Rosenblatt, F. (1961). Principles of neurodynamics: perceptrons and the theory of brain mechanisms. Spartan Books, Washington D. C., Washington

Santé, V. S., Lebert, A., Le Pottier, G. & Ouali, A. (1996). Comparison between two statistical models for prediction of turkey breast meat colour. *Meat Science*, 43, 3-4, 283-290, ISSN: 0309-1740

Santos, J. P., García, M., Aleixandre, M., Horrillo, M. C., Gutiérrez, J., Sayago, I., Fernández, M. J. & Arés, L. (2004). Electronic nose for the identification of pig feeding and ripening time in Iberian hams. *Meat Science*, 66, 3, 727-732, ISSN: 0309-1740

Sebastián, A., Viallon-Femandez, C., Toumayre, P., Berge, P., Sañudo, C., Sánchez, A. & Berdague, J.-L. (2004). Evaluation of collagen and lipid contents and texture of meat by Curie point pyrolysis-mass spectrometry. *Journal of Analytical and Applied Pyrolysis*, 72, 2, 203-208, ISSN: 0165-2370

Sheridan, C., O'Farrell, M., Lewis, E., Flanagan, C., Kerry, J. & Jackman, N. (2007). A comparison of CIE L*a*b* and spectral methods for the analysis of fading in sliced cured ham. *Journal of Optics A-Pure and Applied Optics*, 9, 6, 32-39, ISSN: 1464-4258

Shiranita, K., Hayashi, K., Otsubo, A., Miyajima, T. & Takiyama, R. (2000). Grading meat quality by image processing. *Pattern Recognition*, 33, 1, 97-104, ISSN: 0031-3203.

Tan, F. J., Morgan, M. T., Ludas, L. I., Forrest, J. C. & Gerrard, D. E. (2000). Assessment of fresh pork color with color machine vision. *Journal of Animal Science*, 78, 12, 3078-3085, ISSN: 0021-8812

Tian, Y. Q., McCall, D. G., Dripps, W., Yu, Q. & Gong, P. (2005). Using computer vision technology to evaluate the meat tenderness of grazing beef. *Food Australia*, 57, 8, 322-326, ISSN: 1032-5298

Valous, N. A., Mendoza, F., Sun, D.-W. & Allen, P. (2010). Supervised neural network classification of pre-sliced cooked pork ham images using quaternionic singular values. *Meat Science*, 84, 3, 422-430, ISSN: 0309-1740

Wang, Y., Yang, W., Winter, P. & Walker, L. (2008). Walk-through weighing of pigs using machine vision and an artificial neural network. *Biosystems Engineering*, 100, 1, 117-125, ISSN: 1537-5110

Winquist, F., Hörnsten, E. G., Sundgren, H. & Lundström, I. (1993). Performance of an electronic nose for quality estimation of ground meat. *Measurement Science & Technology*, 4, 12, 1493-1500, ISSN: 0957-0233

Zheng, C., Sun, D.-W. & Zheng, L. (2007). Predicting shrinkage of ellipsoid beef joints as affected by water immersion cooking using image analysis and neural network. *Journal of Food Engineering*, 79, 4, 1243-1249, ISSN: 0260-8774

Zou, J., Han, Y. & So, S.-S. (2008). Overview of artificial neural networks. In: *Artificial neural networks : methods and applications*. Livingstone, D. (Ed.), 15-23, Humana Press, ISBN: 978-1-58829-718-1, New York.

Zupan, J. (1994). Introduction to artificial neural network (ANN) methods: What they are and how to use them. *Acta Chimica Slovenica*, 41, 3, 327-352, ISSN: 1318-0207

Zurera-Cosano, G., Garcia-Gimeno, R. M, Rodriguez-Perez, M. R. & Hervas-Martinez, C. (2005). Validating an artificial neural network model of Leuconostoc mesenteroides in vacuum packaged sliced cooked meat products for shelf-life estimation. *European Food Research and Technology, 221, 5, 717-724,* ISSN: 1438-2377

Part 4

Electric and Power Industry

12

State of Charge Estimation of Ni-MH battery pack by using ANN

Chang-Hao Piao[1,2,3], Wen-Li Fu[1,3], Jin-Wang[3],
Zhi-Yu Huang[1,3] and Chongdu Cho[4]
[1]*Chongqing University of Posts and Telecommunications (Key Laboratory of Network Control & Intelligent Instrument),*
[2]*Chongqing Changan New Energy Automobile CO, LTD,*
[3]*Chongqing University of Posts and Telecommunications(Research Institution of Pattern Recognition and Application),*
[4]*INHA University of Korea (Department of mechanical Engineering)*
[1,2,3]*China*
[4]*Korea*

1. Introduction

1.1 Background and significance of the research

Currently, the world's fuel vehicle is growing by the rate of 30 million per year. It is estimated that the total amount of the world's fuel vehicle for the whole year will reach one billion. The sharp increase demand in oil's resources, further aggravate the shortage of oil resources in the world [1-2]. Fuel vehicle exhaust emission is the main source of urban air pollution today, and the negative impact on the environment is enormous. Environment is closely related to the survival and development of human society. In the case of the energy shortage and environmental protection urgent need to improve, governments invest enormous human and material resources to seek new solutions. This is also bringing the development of electric vehicle [3-6].

As power source and energy storage of HEV, battery is the main factors of impacting on the driving range and driving performance of HEV [7-8]. At present, the most important question is the capacity and battery life issues with HEV application. Only estimate SOC as accurate as possible can we ensure the realization of fast charging and balanced strategy. The purpose of that is to prevent over charge or discharge from damaging battery, and improve battery life. This also has practical significance in increasing battery safety and reducing the battery cost [9].

How accurate tracking of the battery SOC, has been the nickel-hydrogen battery's researchers concerned about putting in a lot of energy to study. Currently, it is very popular to estimate the SOC with Ampere hours (Ah) algorithm as this method is easy to apply in HEV. The residual capacity is calculated by initial capacity minus capacity discharged. But Ah algorithm has two shortcomings. First, it is impossible to forecast the initial SOC. Second, the accumulated error cannot be ignored with the test time growing [10]. The researchers also used a new method that the battery working conditions will be divided into

static, resume, three states of charge and discharge. Then estimate on the three state of SOC separately. It can disperse and eliminates the factors that affect the SOC value in the estimation process. Particularly in the charge-discharge state, they improve Ah algorithm by using the dynamic recovery value based on the coulomb efficiency factor. It solves the cause of the problem of accumulated error by Ah counting method, but this method cannot be displayed its accuracy in the complex conditions [11]. After analysis the large amounts of data under different charge or discharge test conditions, the researchers developed the battery model by cell theory and the external characteristics of the battery pack. Through a large number of experiments, the battery model is improved step by step. At last they completed the final model for measuring SOC in online and real-time. Through Digital model, the battery system's state equation and observation equation can be established. Kalman Filter is use to achieve the minimum mean square error (MMSE) of SOC estimation. The precision of the algorithm is analyses by a experiment in Different charge and discharge test conditions. Through continuous improvement, they can get the algorithm which does not demand exact conformity to initial SOC value. However, this method need researcher's high capacity and is too complicated to fit for the current application [12].
In addition, there have some other methods, such as open circuit voltage, resistance measurement, discharge experiment, the load voltage method and so on [13-15]. But they still cannot meet the requirements of the control requirement of HEV.

1.2 Main content

EV or hybrid electric vehicles (HEV) are mainly used secondary battery in power batteries. Than any other batteries, Ni-MH battery has many advantages: rapid charge or discharge high current, high resistance to charging and discharging capacity, low temperature performance, high mass-power ratio, environmentally friendly (no cadmium mercury or lead) and so on [16].Therefore, this paper studies how to fast and accurately track the SOC based on Ni-MH battery.
This paper designs an artificial neural network (ANN) for predicting Ni-MH batteries in EVs. For achieving the predictability of the network, the text use some basic characteristics of the ANN algorithm such as the ability of non-linear mapping, adapting to the self-learning, parallel processing method, and so on[16-17]. The influence between the current SOCt of Ni-MH battery and the previous SOCt-1 is not considered in most of published paper for the sake of tracking SOC by ANN when they select input variable. So the previous SOCt-1 is interpolated into input variable in this paper. That is to say, the input variable of this discourse are: battery discharging current I, battery terminal voltage U, and previous SOCt-1. Through training a lot of samples, ANN can study and adapt the unknown system's dynamic characteristics, and the characteristic will conserve inside the connected weight of ANN. Simulation results show that the proposed ANN algorithm can accurately predict Ni-MH hybrid vehicle battery SOC, and the average error of output results to reach about 5% in a short time.

2. General layout of ANN

2.1 Basic principles of ANN

The ANN comprises by input layer, hidden layer and output layer. The hidden layer may be one or more layers. The topology of the network is illustrated as figure 1[18-20]:

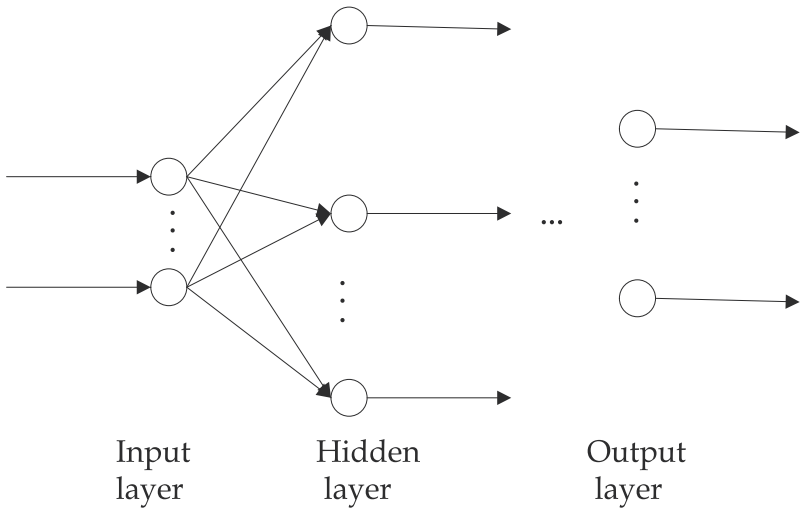


Fig. 1. The model of multilayer perceptron

The number of neurons in input layer is equal to the dimensions of the input signal, the number of hidden layers and hidden nodes depends on the special details, and the number of neurons in output layer is equal to the dimensions of the output signal. In addition to input and output layer, the multilayer perceptron includes one or more hidden units. The hidden units make the network be able to complete a more complex task by picking up more useful information from the input mode. Many synapses of the multilayer perceptron make the network more connective, the changes of the connection domain and connection weights will influence its connectivity. Multilayer perceptron has a unique learning method, which is the famous BP algorithm. Therefore the multilayer perceptron is frequently called the BP network.

It is supposed that the input units are n, the output units are m, and the effect of the network is the map from n-dimension space to m-dimension space. It can be proved that anyone of the nonlinear maps f can accomplish by a 3-layer network. That is to say, it will come true only by one hidden layer. The dimensions m, n of the vector have no any limiting condition. This makes many practical problems with the ANN method to solve possible. In theory, the BPNN can realize any link function map and its range of application is very wide.

2.2 Selection of sample

The performance of ANN is related to the choosing of samples. To successfully develop the useful ANN, the extraction of data samples is the key step. It contains initial data collection, data analysis, variable selection and data pretreatment. Only by these measures can ANN be for effective learning and training.

In this text, we collect once data every 10ms in many driving cycle which set up different initial condition (such as charge and discharge current). After receiving the real-time data of current, voltage and other basic parameters of hybrid car batteries, we can calculate the real-time SOC of the battery by Ampere hours (Ah) algorithm.

The collected data have a certain similarity, for example, directly extract training samples result in containing many redundant data. So they need preliminary sorting. It contain

abandon various kinds of irrational points that causing severe mutations of SOC, periodicity or consistently data also be selected only one group.

To a complex issues, how many data should be selected, which is also the key issues. System input-output relationship is contained in these data samples, so generally the more select data, the more learning and training result reflect the relationship between input-output data. But selecting too many data would increase the cost of the collecting data, the analysis data and network training; of course, selecting too few data could not receive the correct result. In fact, the number of the data depends on many factors, such as the size of the network, the need of the network test and the distribution of the input-output and so on. The size of the network is the most important, and ordinarily the larger network need the more training data [21].

Be inclusive of needing to pay attention to the attending training neural network data, also consider after the neural network finished, needing other test data to chow test the network, and the text data should be independent data assemble.

2.3 Establish the ANN model

The article focus on how to predict battery SOC in real-time according to the battery tested data (cell current、voltage)based on neural network. Generally, its usual operation is that choose the simple network also meet the request. Design a new network type seems difficult. Currently, among the practical application of ANN, the most majority of neural network has adopted BP. Many studies have shown that BPNN with three layers could reach to factual function f( ), thus the article has introduced the triple layers most commonly used BP neural network. The battery current,voltage act as the measured parameter basis for the battery, it must compose the input parameter in neural network. Given the certain relationship between battery SOC changes and its previous SOC, therefore it has to elect the SOC_{t-1} as its input parameter among building the neural network.

Under current time t, determined that HEV Ni-MH battery SOC_t and the current I_t, voltage U_t as well as the raletionship with the preceding time SOC_{t-1}, this is a forecast to the function curve. We can also understand the SOC_t as a three circular function f which is constituted with I_t, U_t and SOC_{t-1}.This has determined the input and the output parameter of the neural network.

After having determined input and output variable, the node number of the network difference level and the output level also determined along with it. Regarding to the layer number of the hidden layer, we first only consider to how to choose a hidden layer, and the left question is how to choice the node point number of the hidden layer. In neural network's design, increases the number of the hidden layer's neurons can improve the precision which the network and the training set match, but the more of the hidden layer's neurons is not better. Too many number of the neuron will let the network remember all training data including noise. It will reduce pan-ability of the network. In the foundation of it can reflect correctly the relationships between input and output, selects the few hidden layer nodal point number. This makes the network to be as simple as possible [20]. After contrast simulation according to cut and try method the result discovered that neural network's hidden layer uses 10 neurons can describe curve relations about the input variable and the output variable quite accurately.

The ANN structure is used in this experiment shown in Fig. 2.

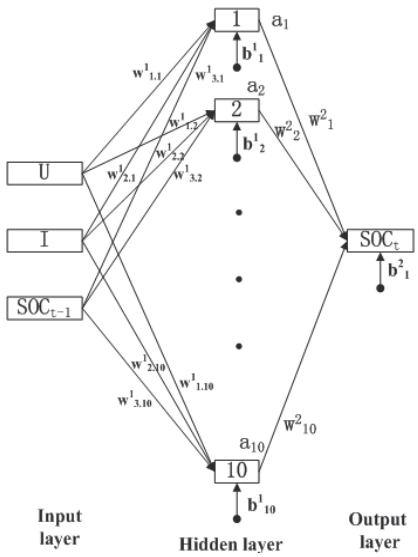


Fig. 2. Two layers of neural network structure

In the Fig. 2, w expresses the connection of weight between the two layers, b means every neuron's threshold, f expresses ANN's transfer function. Superscript on the $w^1_{1.1}$ expresses this value is the connection weights between input layer and hidden layer, while the weight of hidden layer and output layer expressed by number 2; the first number 1 of subscript means input is U_t, and the input I_t, SOC_{t-1} are expressed by number 2, 3; the second number 1 of subscript expresses the connection weights between the first neuron of the hidden layer and the input value. The superscript of b^1_1 means hidden layer neurons, and output layer neurons express with number 2. The subscript of b^1_1 means the first neuron of current layer. The a_1 expresses the first neuron's output in the hidden layer.
The output SOC of ANN's output layer defined as:

$$SOC = f[w^2 * f(w^1 * x + b^1)] + b^2 \tag{1}$$

x is the input value of ANN, and linear transfer function is $f(x)$ which equal to x in upper equation.

3. Training algorithm

In general, BP neural network is a kind of three or more than three multilayer neural network, it's about each neuron between the layers to achieve full connectivity, namely each layer in the left and right layers of neurons has a connection. BP network learning by a teacher's training. When a mode of learning provided to the network, its activation values of neurons will transmit from the input layer to the middle layer, land up output layer at last. Corresponds to the input mode, each neuron will export network response in the output layer. Then, follow the reduction of the desired output and actual output error principle, from the output layer through an intermediate layer, and finally back to the input layer connection weights layer by layer correction. This correction process is carried out from the output to the input layer. So it is called error back propagation algorithm. As this error back propagation constant training, the network input mode for the correct response rate is also rising.

It adopts the training and emulating alternate work model to avoid the net excess training. After the training samples achieve an net training, it keep the net weight value and threshold constant, validation samples data is used as the net input, running the net in forward direction and examining the average output error. During the simulation, the previous time simulation output is used for the next time simulation input, $SOC(i) = SOC'(i-1)$, $i > 1$ and is integral number. If continue training cannot decrease the average error, the net training is over. If we modify the parameter of NN such as learning rate slightly and keep the input and output constant, the average output error cannot decrease also, so we consider this net is the optimization result in the case of keeping the input and the Network Structure constant.

Commonly used BP algorithms exists a long time and slow convergence disadvantage etc. So this paper used the proportion conjugation gradient training algorithm. Conjugation gradient algorithm is required to search network linearly and then adjust its direction at each training cycle. This linear search at each search must be repeated for calculating all samples, this consumes a lot of time. While proportion conjugation gradient algorithm combines value trust region algorithm with the conjugation gradient algorithm, effectively reduce the search time mentioned above and improve the training speed of the network[22]. The BP neural network training process used in this article is shown in Fig. 3.

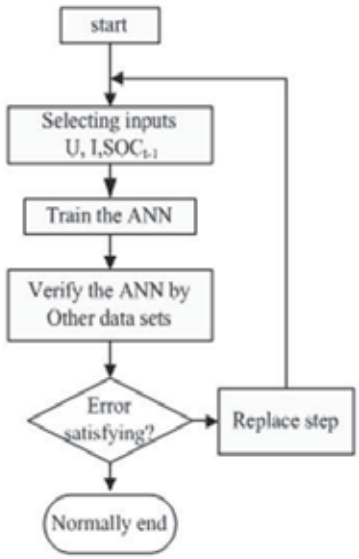


Fig. 3. The training flow chart of BPNN

Input training samples U and I are datum based on t moment in the Fig. 3. SOC is the data based on t-1 moment. ε represent a pre-set training ending goal. This goal is not the smaller the better, because over-training problem is existed in the network. Before we input the NN training sample, it must firstly assign the initial net parameter. ANN calculates the output of hidden neurons, and gets the Output of Output layer neuron. It also calculates every layer neuron output error. If the error is too big, we must modify the net weight value and threshold. After the sample are all trained, if the NN average error is smaller than the setting object for ending the training, the training is over, or else it keeps on new training after updating the total training steps.

4. Experiment results and analysis

4.1 Experiment and result

4.1.1 Training

In accordance with the above training methods, we first use the training sample to get a neural network and recorded it as network 1. In this topic research, we don't use the traditional authentication method, as lead the validation data into the model, and analyze difference value between the model prediction value and real value. The specific flow chart is shown in Figure 4. In the actual application, it only supplies the initial value or even the wrong initial value when we use battery management system to estimate electrokinetic cell SOC. In the research of this topic, we completely use the prediction technique. In the first time of prediction, we can get the input current, voltage and battery SOC, which the primary neural network model is needed. In the second prediction, we only input the collected current and voltage. The battery SOC is as the predication results as last time. In such a way, it can reflect the model's ability of self-adapting and tracing whole. When we have traced many times and amended the parameters such as network learning rate and so on, the output average error of the network still can't diminish. We will consider this network as the best result at present during the network input parameters are not changed. This paper will replace the network with the network 1 finally.

In the similar way, we can continue to add training samples b into the network 1, and obtain the network 2 by training. By parity of reasoning, when we have added the training sample of c, d and e, we can get network of 3, 4 and 5 respectively. The average error of each network at different time is shown in chart 1. As can be seen from chart 1, the average error of the output from the neural network 1 to neural networks 4 is gradually reduced, but it begin to increase from the network 5. It shows in the same case of input samples and training algorithm, network 4 is the best results we can get. This paper uses the network 4 as the neural network model, which will be tested finally. In which the training samples used as input of neural network's output comparison chart is shown in Fig.5. It uses the validation sample as input of neural network's output comparison chart is shown in Fig.6.

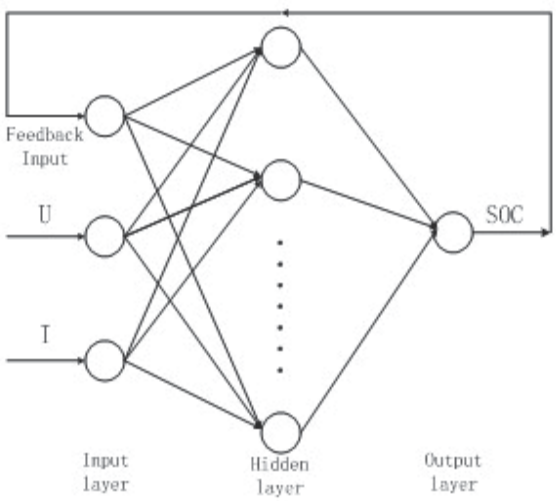


Fig. 4. The flow chart of checking model

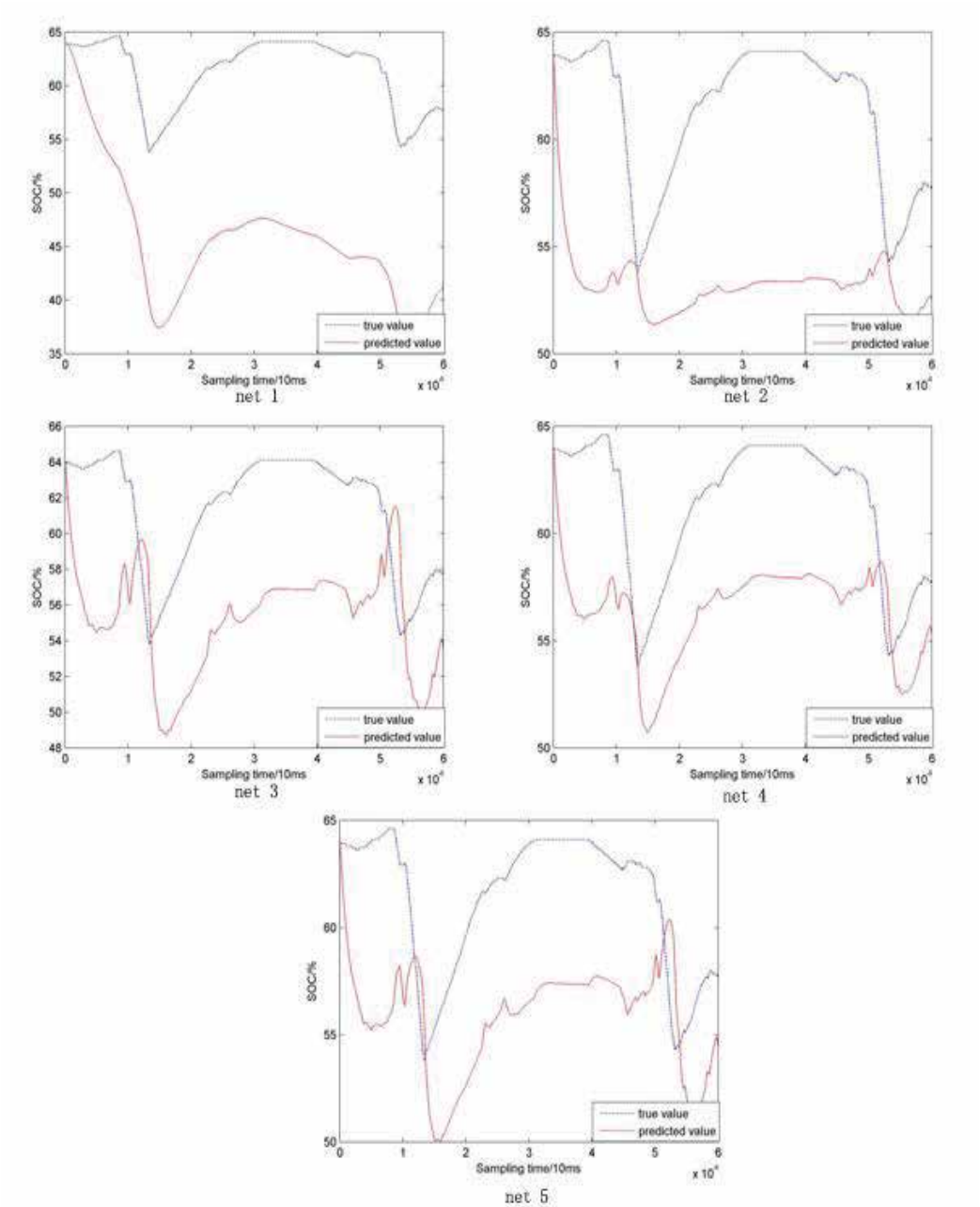


Fig. 5. The output result waveform of the training sample

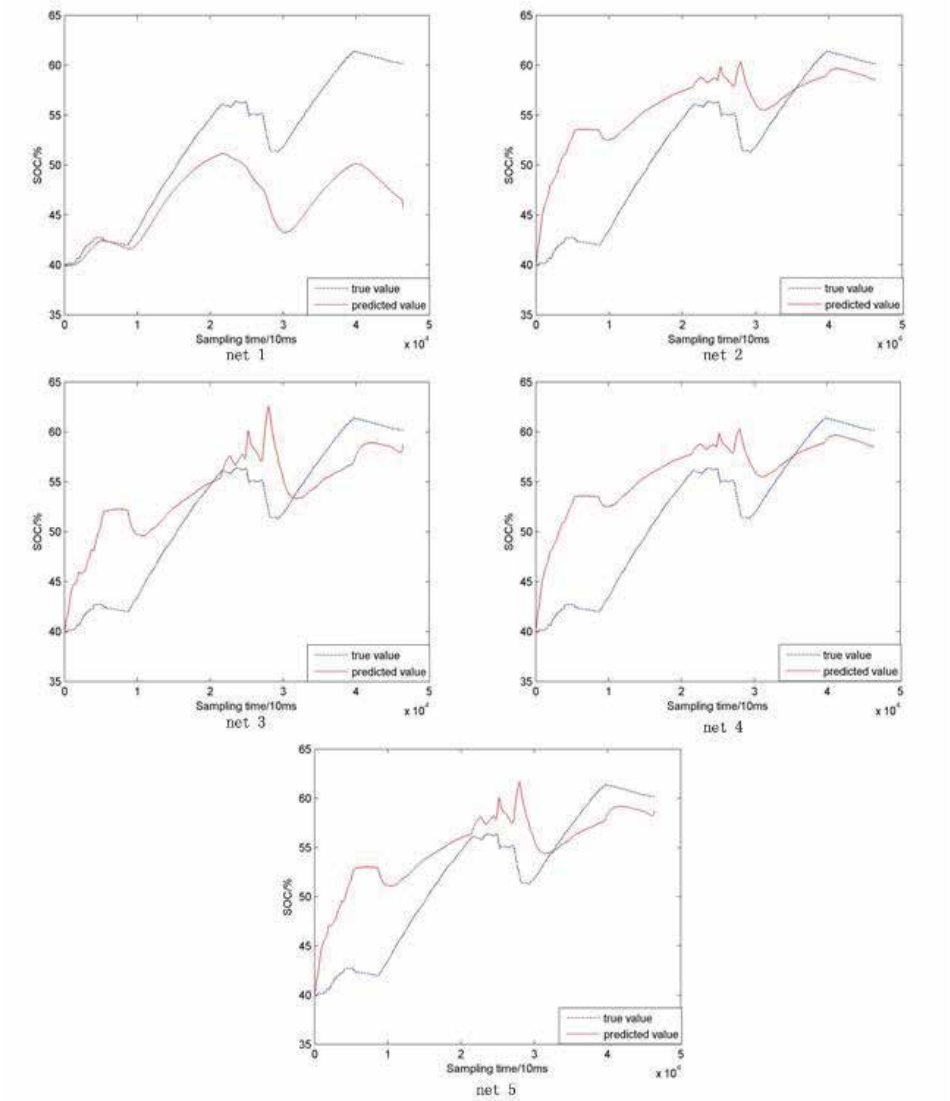


Fig. 6. The output result waveform of the checking sample is input.

		Net 1	Net 2	Net 3	Net 4	Net 5
start time error	training sample	29.9%	13.9%	8.9%	7.3%	8.1%
	checking sample	17.3%	9.7%	7.5%	9.5%	8.4%
3minutes 20seconds after error	training sample	29.8%	13.3%	8.2%	6.7%	7.5%
	checking sample	23.7%	7.1%	4.7%	4.2%	4.3%

Table 1. Different network's average error

4.1.2 Test and result

When you are sure the neural network which you have got is the best, use the validation sample and training sample to test the tracking of network respectively. That is in the any case of initial value setting of SOC, how long the neural network required to reduce output error to an acceptable extent. All samples' sampling time is 10ms. The initial value of SOC is obtained by Ah algorithm or is set arbitrarily (0, 30, 50, 70, 100 separately). Since the emulated waveform contains a number of fluctuations which are caused by current's mutation and voltage's mutation, this experiment uses a weighted filtering to process results. It consider the first few moments of factors in the current results (their own value instead of the output value at the start time). Through testing, the weighted parameters of all time choose the best one. Chart 2 shows the average error of some samples for the artificial neural network 4's forecast of results, which is in the condition of different initial value of SOC.

	Sample time t/s	average prediction error		Sample time t/s	average prediction error
Checking sample	87	6.9%	Training sample	64	7.2%
	150	4.8%		150	6.9%
	200	4.2%		200	6.7%

Table 2. The average error of ANN output

4.2 Result analysis

4.2.1 Training result

As can be seen from Table 1, we calculate the average error of the output from the initial moment. The inaccuracy of five networks which respectively use the training sample as the input are reduce at first, then its increase, and the NO.4 network's inaccuracy is the smallest. The inaccuracy of five networks changes to be quite disorderly which use the confirmation sample as the input. But apart from the network 1, the other four networks errors are about 8% and the differences are not large. As we calculate the average error of output from 3 minutes and 20 seconds, the inaccuracy of five networks are reduce at first which respectively use the training sample as the input, then its increase. The NO.4 network is the smallest. Regarding other abilities of ANN, generally we pay more attention to its generalization ability and tracking ability of the network running. From experimental results in Table 1, we know this paper focuses on change of the output's average error, which ANN uses the validation sample as input and start from 3 minutes and 20 seconds. At last the auto-adapted ability of Network 4 is the best, and the forecasting result is most accurate. From the comparison ware-form of output result in Fig.5, Fig.6 and Fig.7, we know, the output wave-form of network 4 is more close to the real value than other networks.

As Shown in Fig.7, it's the comparison chart of five network output value which ANN uses the validation sample as input value. Compares with other network's output result, the output result profile of network 4 and the network 5 are obviously closer the changes

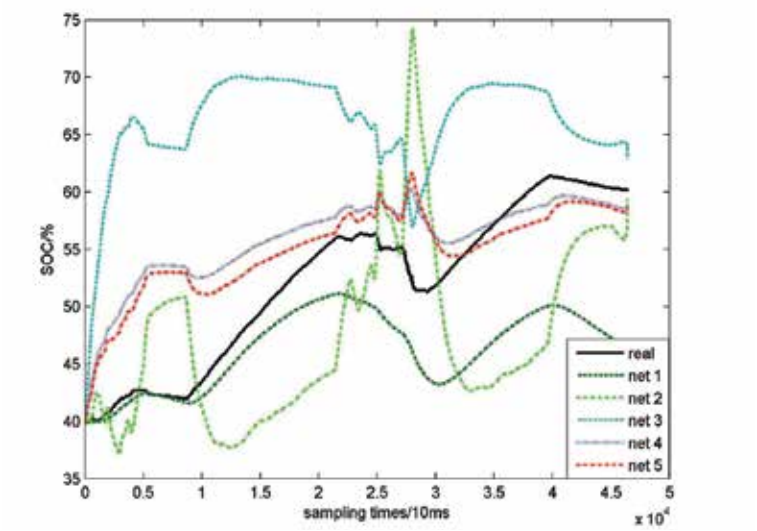


Fig. 7. Test Chart of five sample test network

waveform of real value in Fig.7. But from the Table 1 we know, the output average error of network 4 which calculated after a period of time after is smaller than the value of the network 5. Therefore, the network 4 is the networks which this laboratory needs.

Through the above analysis of training results, we use the network 4 as Neural Network which predicts the car battery SOC. Network structure of the network 4 as shown in Fig.8. And the weight of concealment level to output level is middle line of data in Fig.8. The weight of input level to conceals between the level as shown in Table 3.

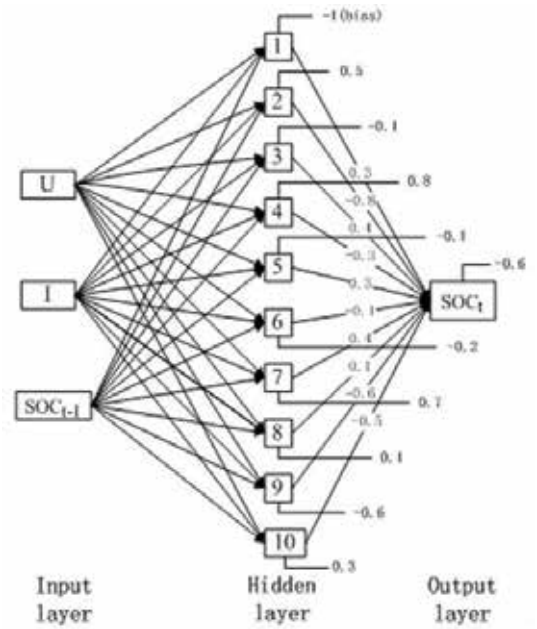


Fig. 8. Actual structure of neural network.

Hidden layer neuron	1	2	3	4	5
U/v	0.7	-0.3	0.1	0.02	0.6
I/A	0.3	0.5	0.9	0.7	-0.6
SOC/%	-0.8	-0.4	0.6	-1	-0.7
Hidden layer neuron	6	7	8	9	10
U/v	0.5	-0.2	-0.9	0.7	0.03
I/A	-0.2	0.9	0.8	-0.2	0.7
SOC/%	-0.6	-0.5	0.2	-0.5	-0.6

Table 3. The weight between input level and concealment level

4.2.2 Analysis of test results

In this paper, we illustrate the tracking performance of the neural network though the training sample and validation sample results. The SOC initial value of simulation respectively supposes 0, 30, 50, 70, 100 and the sample real value. The simulation result of training samples and confirmation sample of network 4 are shown in Fig.9.

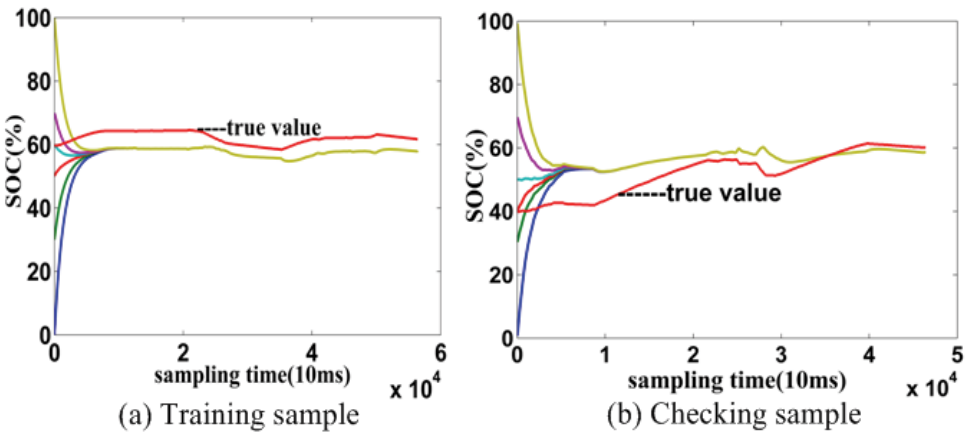


Fig. 9. The forecasting result of ANN as set the different SOC initial value

From Table 2, Table 3 and Fig. 9, we can draw that neural network can basically overcome prediction effect of the initial value of sample SOC set arbitrarily after training samples pass 64 seconds and checking samples pass 87 seconds. For checking samples, error comes to 6.9% after 87 seconds when the initial value is arbitrarily set 0, which is smaller than average error of 9.5% when the initial value is set true value. For training samples, error comes to 7.2% after 64 seconds when the initial value is arbitrarily set 0, which is smaller than average error of 7.3% when the initial value is set true value. As the time passes by, whatever the

initial value of experiment it is, the prediction average error of neural network is smaller and smaller, and the average error values are less than 10%. Through the above analysis, we can see that the neural network after training can be close to the target value of training network at a very short time and it has strong self-adaptive ability.
The waveform of the sample input variant U, I is shown in Fig. 10 and the sampling time is 10s.

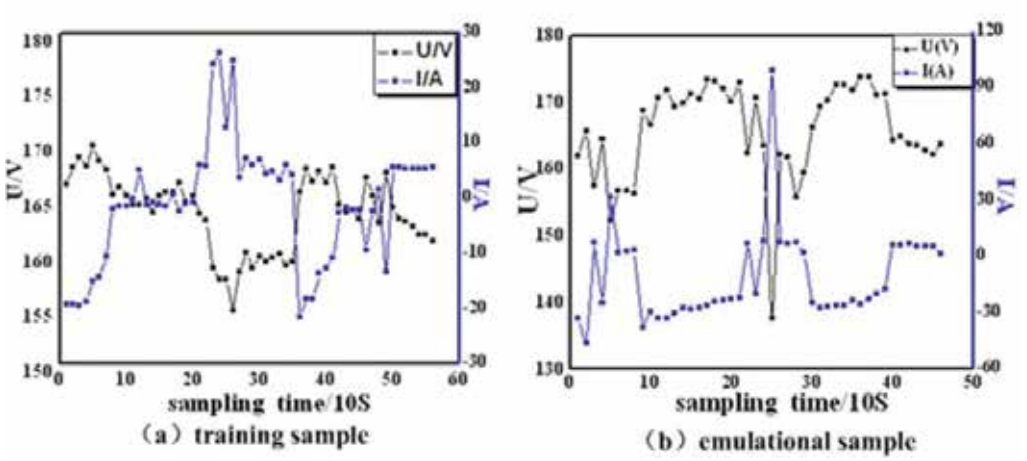


Fig. 10. Experimental data of current and voltage

The waveform of sample input variant U, I is shown in Fig.10 and the sampling time is 10s. Compared prediction waveform in Fig. 9 with the trend of sample current, voltage in Fig. 10, we can also see that the prediction ability of BP neural network algorithm can better reflect the trend of battery current and voltage. That is, when the current is negative, prediction result of SOC turns to decrease respectively. Output average error within 10% and the correspondence between input variable U, I demonstrate that it is very accurate to predicting the SOC of automotive power battery with BP neural network algorithm.

5. Conclusion

In order to predict the SOC of nickel hydrogen battery in real-time when the car is running, and at the same time guarantee the accuracy of prediction and good self-adaptive capacity, this paper designs a artificial neural network with three inputs and ten neurons and one output that can be used to predict the SOC of nickel hydrogen power battery. The neural network puts previous state of charge that is SOC_{t-1} to the prediction of the neural network, thus the effect of SOC_{t-1} toward predicting SOC_t is considered, so the self-adaptive ability of neural network is improved. Proportion conjugation gradient algorithm is used in the neural network training process, the connection weight value of the network is constantly changed via alternative training simulation and finally form fixed memory model for the prediction of the SOC. In the training network, but still pay attention to the selection of data, it has a great influence that different data sample finally forecast accuracy of network. Each training the neural network, it will gain a better simulation results, and then again add the data to back training network. At the same time, according to the comparison of the different neural networks, we can avoid over-training network. Simulation of the samples

indicate that artificial neural network built by experiment can accurately predict the SOC of the nickel hydrogen power battery of hybrid automobile and the self-adaptive is good. These features make the algorithm has a very high application value.

6. Future work

In recent years, the new energy industry and electric cars were pushed to unprecedented level, which will generate a new round of development opportunities. The battery is essential to new energy, automotive and other industries as the energy storage device. In response to industrial restructuring, the healthy development of the emerging industry requirements, the battery capacity technology have higher requirements, the development of remaining battery capacity of prediction is very urgent. In this environment, this article from the battery capacity forecasting technology's current present situation and the trend of development, combined with the actual situation of nickel-metal hydride batteries, and established a BP neural network model. The algorithm in predicting SOC values more consider the weight factors which can be measured, and other unpredictable factors do not consider. To further improve the algorithm accuracy and reliability in harsh environments, much work will need to be done:

1. In order to effectively improve the accuracy of the algorithm, so that parallel operation, it also need to increase the inclusiveness of the data and add another algorithm in the neural network.
2. To take further the reliability of the quantitative analysis, this method only changes in charge and discharge current mode of qualitative analysis and evaluation, and no failure mode of the system reliability parameters, such as the quantitative calculation of system failure.
3. In addition to the above-mentioned factors, there are other factors to consider in the battery of the work environment, such as battery temperature of their environment, the consumption of battery life and other factors. Future research needs to take these factors into account, so it makes BP neural network is more complete and has better predictability.

7. References

[1] Li Hongjian;Lin Yuchun. (2007). Vigorously the development and promotion of our natural gas vehicles. MAGNIFICENT WRTING, No.4, (2007) page numbers (174), ISSN l009-5489

[2] MA Youliang; CHEN Quanshi; QI Zhanning. (2001). Research on the SOC definition and measurement method of batteries used in EVs. Journal of Tsinghua University(Science and Technology), Vol.41, No.11, (2001) page numbers (95-98), ISSN 1000－0054

[3] Piao Changhao, Yang Xiaoyong, Teng Cong and Yang Huiqian(2010), An Improved Model Based on Artificial Neural Networks and Thevenin Model for Nickel Metal Hydride Power Battery, 2010 International Conference on Optics, Photonics and Energy Engineering, March 2010, Wuhan, China , pp.

[4] T.Shinpo. Development of Battery Management System for Electric Vehicle.Proc.of the 14 International Electric Vehicle Sysposium(1997)

[5] Lin Chengtao; CHEN Quanshi; WANG Junping. (2006). Improved Ah counting method for state of charge estimation of electric vehicle batteries. Journal of Tsinghua University(Science and Technology), Vol.26, No.2, (2006) page numbers (76-79), ISSN 1000－0054

[6] Zhao Huiyong; Luo Yongge; Yang Qiliang; She Jianqiang. (2004). Situation and Development on the Vehicle NiMHBattery Management System. Journal of Hubei Automotive Industries Institute, Vol.18, No.3, (SEP2004) page numbers (23-26), ISSN 1008-5483

[7] Jiang Jiuchun，Niu Liyong，Zhang Xin. (2004). Research 0n Battery Management System for Hybrid Electric Vehicle. High Technology Letters, Vol.11, No.2, (2004) page numbers (75-77), ISSN 1002-0470

[8] Chen Jun;Tao Zhanliang. (2006). Nickel-Metal Hydride Secondary Battery, Chemical Industry Press, ISBN 9787502583859,Bei Jing

[9] PILLER S, PERRIN M, JOSSEN A. Methods for state of charge determination and their application[J]. Journal of Power Sources, 2001,96(5):113-120.

[10] Zhang Hongmei, DENG Zhenglong. UKF-based attitude determination method for gyro less satellite[J]. Journal of System Engineering and Electronics, 2004, 15(2): 105-109.

[11] Tian Xiaohui;Diao Hainan;Fan Bo;Qiu Yunpeng. (2010). Research on estimation of lithium-ion battery SOC for electric vehicle. Chinese Journal of Power Sources, Vol.6, No.1, (2010) page numbers (51-54), ISSN 1002-087X

[12] Meng Siqi; Yang Honggeng. (2008). Short-Term Load Forecasting Based on Second-Order Correction of Kalman Filter. Advances of Power System & Hydroelectric Engineering, Vol.24, No.2, (200802) page numbers (73-75), ISSN 1674-0009

[13] LIN Chengtao, WANG Junping, CHEN Quanshi, Methods for state of charge estimation of EV batteries and their application[J], 2004, 34(5):772-774.

[14] SIMMONDS N, Device for indicating the residual capacity of secondary cells: U.S. Patent 5 518 835[P], 1996-5-21.

[15] SALAMEH M, CASACCA A. A mathematical model for lead2acid batteries [J]. IEEE Trans Energy Conversion, 1992 ,7 (1): 93 —97.

[16] Shen Wenxue, State of available capacity estimation for lead-acid batteries in electric vehicles using neural network[J], Energy Conversion and Management, 2007, 48(2):433-442.

[17] A. Salkind, C. Fennie, P. Singh, T. Atwater, D. Reisner, Determination of state-of-charge and state-of-health of batteries by fuzzy logic methodology[J], Journal of Power Sources , 1999, 80(2): 293-300.

[18] Simon Haykin. (2001). Neural network： A comprehensive foundation (Second edition), Tsinghua University Press, ISBN 730204936, Bei Jing

[19] Chang-hao Piao, Wen-li Fu, Gai-hui Lei and Chong-du Cho, Online Parameter Estimation of the Ni-MH Batteries Based on Statistical Methods, Energies, Volume 3, Issue 2 (February 2010), pp.206-215

[20] Martin T.Hagan. (2002). Neural network design, Machinery Industry Press, ISBN 7111075854., Bei Jing

[21] Jiao Huimin; Yu Qunming. Research on Capacity Predication of Battery based on BP network and Genetic Algorithm. Computer Simulation, Vol.26, No.11, (2006) page numbers (218-220), ISSN 1006-9348

[22] Wen Xin; Zhou Lu. (2000). MATLAB Neural network application and design, Science and Technology Press, ISBN 7030084802, Bei Jing
[23] Gao Juan. (2003). Artificial neural network theory and simulation, Machinery Industry Press, ISBN 9787111125914, Bei Jing
[24] Lin C, Reinforcement structure/parameter learning for neural network based fuzzy logic control systems [J], IEEE Trans. Fuzzy Syst, 1994, 2(1): 46–63

13

A Novel Frequency Tracking Method Based on Complex Adaptive Linear Neural Network State Vector in Power Systems

M. Joorabian, I. Sadinejad and M. Baghdadi
Shahid Chamran University
Iran

1. Introduction

In some digital application systems, power system frequency tracking is an important task. Accurate power frequency estimation is a necessity to check the state of health of power index, and a guarantee for accurate quantitative measurement of power parameters such as voltages, currents, active power and reactive power, in multi-function power meters under steady states. Many researches have been done in this area.
Three criteria that a frequency tracking method should satisfy is given as follows (Akke,1997):

1. Fast speed of convergence
2. Accuracy of frequency estimation
3. Robustness to noise.

He compares traditional modulation with new modulation. Traditional demodulation introduces a double frequency component that needs to be filtered away. For signals with low noise, the filter to reduce the double frequency component can often limit the speed of the frequency estimation algorithm. The purpose of this section is to show that the proposed method eliminates this problem. If other filters are the bottle-neck of the estimation algorithm, we will not capitalise on the benefits.
Many well-proven techniques such as zero-crossing technique, level-crossing technique, least squares error technique, Newton method, Kalman filter, Fourier transform, and wavelet transform have been used for power harmonic frequency estimation in the fields of measurement, instrumentation, control and monitoring of power systems. Besides, a comprehensive analysis of discrete Fourier transform (DFT) error is given in some researches, including the cases of synchronous sampling and error rises when sampling frequency does not synchronize with signal frequency. A frequency tracking method based on linear estimation of phase (LEP) has been introduced. Also, a processing unit for symmetrical components and harmonic estimation based on an adaptive linear combiner has been proposed.
This section presents the application of a complex adaptive linear neural network (CADALINE) in tracking the fundamental power system frequency. In this method, by using stationary-axes Park transformation in addition to producing a complex input measurement, the decaying DC offset is effectively eliminated. As the proposed method uses a first-order differentiator to estimate frequency changes, a Hamming filter is used to

smoothen the response and cancel high-frequency noises. The most distinguishing features of the proposed method are the reduction in the size of observation state vector required by a simple adaptive linear neural network (ADALINE) and increase in the accuracy and convergence speed under transient conditions. This section concludes with the presentation of the representative results obtained in numerical simulations and simulation in PSCAD/EMTDC software as well as in practical study.

2. ADALINE structure to track fundamental frequency

Figure 1 depicts the ADALINE structure to track fundamental frequency which is a proposed in this section.

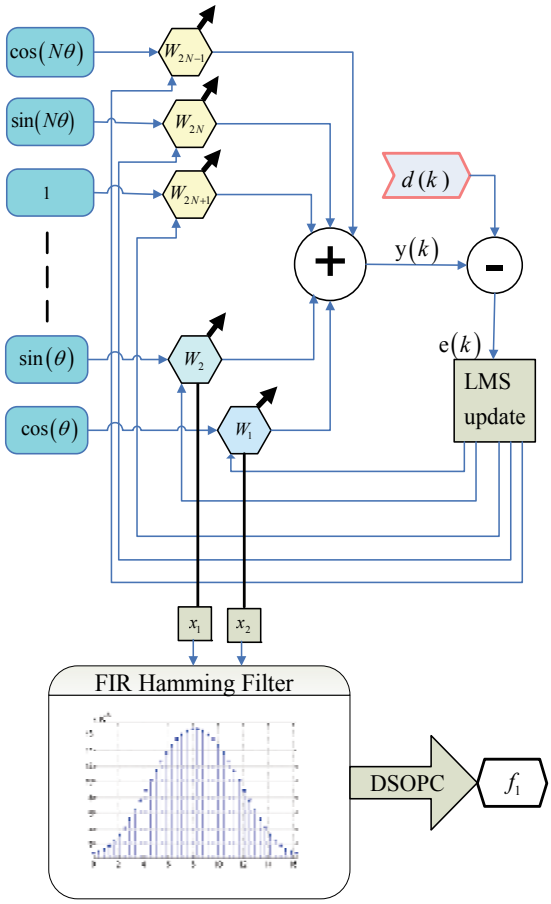


Fig. 1. ADALINE structure to track fundamental frequency

Assume that the voltage waveform of power system comprises unknown fundamental, harmonics and decaying DC offset components as:

$$V(t) = \sum_{l=1}^{N} V_l \sin\left(l\omega_0 t + \Phi_1\right) + A_v e^{\left(-\frac{t}{\tau}\right)} \tag{1}$$

where V_1 and Φ_1 are the amplitude and phase of the fundamental frequency respectively. A_v and τ are the amplitude and time constant of decaying DC offset respectively; N is the total number of harmonics; and ω_0 is the fundamental angular frequency in (rad/sec). Time-discrete expression of (1) is:

$$V(k)=\sum_{l=1}^{N} V_l \sin(l\theta+\Phi_1)+A_v e^{\left(-\frac{\theta}{\tau.\omega_0}\right)} \tag{2}$$

where $\theta = 2\pi k / N_s$ and N_s is sampling rate given by $N_s=\frac{f_s}{f_0}$, in which f_s is sampling frequency and f_0 is fundamental frequency of power system. By using the triangular equality:

$$\sin(\alpha+\beta)=\sin(\alpha)\cos(\beta)+\sin(\beta)\cos(\alpha) \tag{3}$$

Equation (2) can be rewritten as:

$$V(k)=\sum_{l=1}^{N} V_l \sin(\Phi_l)\cos(l\theta)+V_l\cos(\Phi_l)\sin(l\theta)+A_v e^{\left(-\frac{\theta}{\tau.\omega_0}\right)} \tag{4}$$

Rearranging the above equation in the matrix form, we obtain:

$$V(k)=\Psi_V \times X^T(k) \tag{5}$$

where $V(k)$ represents the measurement at each sampling, $X(k)$ is the time varying observation matrix and Ψ_V is the parameter at each iteration to be tracked. $X(k)$ and Ψ_V are shown in the following formula:

$$\begin{aligned}\Psi_V &= \begin{bmatrix} V_1\sin(\Phi_1) & V_1\cos(\Phi_1) & V_2\sin(\Phi_2) & V_2\cos(\Phi_2) \\ \dots & V_N\sin(\Phi_N) & V_N\cos(\Phi_N) & A_v \quad -\frac{A_v}{\tau} \end{bmatrix} \\ X(k) &= \begin{bmatrix} \cos(\theta) & \sin(\theta) & \cos(2\theta) & \sin(2\theta) \\ \dots\cos(N\theta) & \sin(N\theta) & 1 & -\frac{\theta}{\omega_0} \end{bmatrix}\end{aligned} \tag{6}$$

According to Fig. 1, at kth iteration, the input vector $X(k)$ is multiplied by the weighting vector $W(k)=\left[w_1(k)w_2(k)\dots w_P(k)\right]$, and then these weighted inputs are summed to produce the linear output $y(k)=W(k)\times X(k)^T$. In order for the ADALINE output to precisely mimic the desired value $d(k)$, the weight vector is adjusted utilizing an adaptation rule that is mainly based on least mean square (LMS) algorithm. This rule is also known as Widrow–Hoff delta rule [27] and is given by:

$$W(k+1)=W(k)+\frac{\alpha\, e(k)X(k)}{X(k)\times X(k)^T} \tag{7}$$

where α is the constant learning parameter and $e(k)=y(k)-d(k)$ is the error. When perfect learning is attained, the error is reduced to zero and the desired output becomes equal to $d(k)=W_0\times X(k)^T$, where W_0 is the weight vector after the complete algorithm convergence. Thus, the neural model exactly predicts the incoming signal. To track harmonic components of a voltage signal with ADALINE, the variables $\Psi_V(k)$ and $V(k)$ are simply assigned to $W(k)$ and $d(k)$ respectively, with $P=2\times N+2$.
After mentioned error converges to zero, the weight vector yields the Fourier coefficients of power signal as:

$$W_0=\begin{bmatrix} V_1\sin(\Phi_1) & V_1\cos(\Phi_1) & V_2\sin(\Phi_2) & V_2\cos(\Phi_2) \\ \dots & V_N\sin(\Phi_N) & V_N\cos(\Phi_N) & A_v \quad -\dfrac{A_v}{\tau} \end{bmatrix} \tag{8}$$

Voltage amplitude and phase angle of N^{th} harmonic are:

$$\begin{aligned} V_N &= \sqrt{W_0^2(2N-1)+W_0^2(2N)} \\ \Phi_{V,N} &= \tan^{-1}\left(W_0(2N-1)\Big/W_0(2N)\right) \end{aligned} \tag{9}$$

Voltage amplitude and phase angle of fundamental frequency extracted by (9) are:

$$\begin{aligned} V_1 &= \sqrt{W_0^2(1)+W_0^2(2)} \\ \Phi_{V,1} &= \tan^{-1}\left(W_0(1)\Big/W_0(2)\right) \end{aligned} \tag{10}$$

By sampling current signal with the same approach, discrete expression of current is:

$$I(k)=\sum_{l=1}^{N} I_l\sin(\Phi_l)\cos(l\theta)+I_l\cos(\Phi_l)\sin(l\theta)+A_i e^{\left(-\frac{\theta}{\omega_0\cdot\tau}\right)} \tag{11}$$

in which, variables $\Psi_I(k)$ and $I(k)$ are simply assigned to $W(k)$ and $d(k)$ respectively, with $P=2\times N+2$ [27]. $\Psi_I(k)$ is defined as:

$$\Psi_I=\begin{bmatrix} I_1\sin(\Phi_1) & I_1\cos(\Phi_1) & I_2\sin(\Phi_2) & I_2\cos(\Phi_2) \\ \dots & I_N\sin(\Phi_N) & I_N\cos(\Phi_N) & A_i \quad -\dfrac{A_i}{\tau} \end{bmatrix} \tag{12}$$

Current amplitude and phase of N^{th} harmonic are calculated as follows:

$$\begin{aligned} I_N &= \sqrt{W_0^2(2N-1)+W_0^2(2N)} \\ \Phi_{I,N} &= \tan^{-1}\left(W_0(2N-1)\Big/W_0(2N)\right) \end{aligned} \tag{13}$$

Current amplitude and phase of fundamental frequency are achieved by:

$$I_1 = \sqrt{W_0^2(1) + W_0^2(2)}$$
$$\Phi_{I,1} = \tan^{-1}\left(W_0(1) \Big/ W_0(2)\right) \tag{14}$$

To track frequency, a center frequency is assumed to be the actual value. It would be the operational frequency of the power system which is usually 50 Hz or 60 Hz. Under situations that the base power frequency changes, the k^{th} sample of fundamental component of voltage or current signal is modeled:

$$s(kT_s) = A \cdot \sin(2\pi f_x kT_s + \phi) \tag{15}$$

that can be rewritten as:

$$s(kT_s) = x_1 \sin(2\pi \cdot f_0 \cdot k \cdot T_s) + x_2 \cos(2\pi \cdot f_0 \cdot k \cdot T_s) \tag{16}$$

Where x_1 is the in-phase component, x_2 is the quadrature phase component, f_0 is the center frequency (60 Hz), f_1 is the frequency deviation, and T_s is the sampling interval $\left(\frac{1}{f_s}\right)$. Before calculating the frequency deviation (f_1), x_1 and x_2 pass through a FIR Hamming window and parameters y_1 and y_2 are obtained as:

$$y_1(kT_s) = \sum_{i=1}^{N_0} x_1\left((k-i+1)\cdot T_s\right) \times H(i)$$
$$y_2(kT_s) = \sum_{i=1}^{N_0} x_2\left((k-i+1)\cdot T_s\right) \times H(i) \tag{17}$$

where $H(i)$ is the i^{th} coefficient of the FIR Hamming window coefficients and N_0 is the sampling rate given by $N_0 = \left(\frac{f_s}{f_0}\right)$. The cut frequency for the low pass Hamming window is 20 Hz and the length of filter is 40. Fig. 2 shows the impulse response of this Hamming window. By using DSOPC principle, f_1 is obtained as:

$$f_1(kT_s) = \frac{1}{2\pi} \times \frac{y_1(kT_s)y_2'(kT_s) - y_2(kT_s)y_1'(kT_s)}{(y^2{}_2(kT_s) + y^2{}_1(kT_s))} \tag{18}$$

y_1' and y_2' are first-order discrete derivatives defined as:

$$y_1'(kT_s) = \frac{y_1(kT_s) - y_1(kT_s - T_s)}{T_s}$$
$$y_2'(kT_s) = \frac{y_2(kT_s) - y_2(kT_s - T_s)}{T_s} \tag{19}$$

Finally, the real value of fundamental frequency (f_x) is calculated by adding the frequency deviation to the assumed center frequency as:

$$f_x = f_0 + f_1 \tag{20}$$

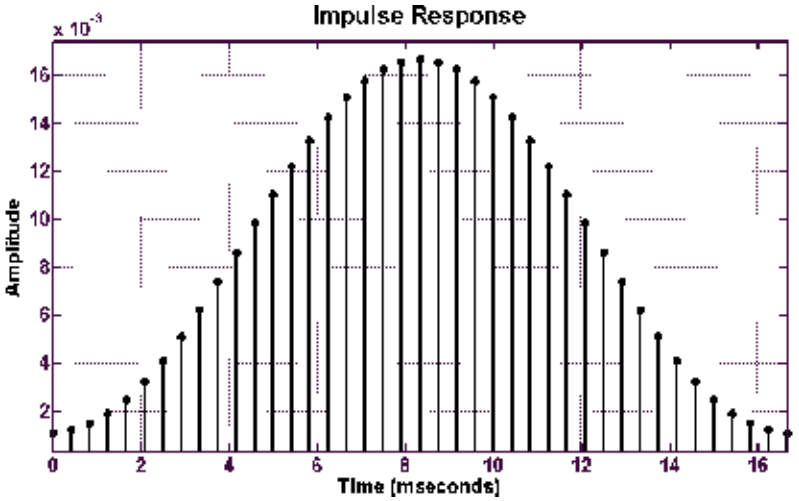


Fig. 2. Impulse response of the Hamming window with 20 Hz cut frequency

3. Complex ADALINE structure to track fundamental frequency

The proposed complex ADALINE (CADALINE) structure is based on the Widrow–Hoff delta rule, explained earlier. The improvement in ADALINE structure is made by introducing a complex observation vector. This approach reduces the number of weight updates, and so, the number of parameters to be estimated. To produce a complex vector measurement the use of the stationary-axes Park transformation is proposed. Stationary-axes Park transformation is widely employed to study the behavior of rotating electrical machines in transient conditions. However, it can be considered a more general and powerful tool to study the behavior of three-phase systems. This transformation applied to the signals $y_a(t), y_b(t)$ and $y_c(t)$ (voltages or currents) of a three-phase system leads to the Park components $y_d(t)$, $y_q(t)$ and $y_0(t)$ defined as:

$$\begin{bmatrix} y_d \\ y_q \\ y_0 \end{bmatrix} = [T]. \begin{bmatrix} y_a \\ y_b \\ y_c \end{bmatrix} \tag{21}$$

where $[T]$ is the orthogonal matrix defined as:

$$T = \begin{bmatrix} \sqrt{\frac{2}{3}} & -\sqrt{\frac{1}{6}} & -\sqrt{\frac{1}{6}} \\ 0 & \sqrt{\frac{1}{2}} & -\sqrt{\frac{1}{2}} \\ \sqrt{\frac{1}{3}} & \sqrt{\frac{1}{3}} & \sqrt{\frac{1}{3}} \end{bmatrix} \tag{22}$$

In the *d-q* frame, it is then possible to define the Park vector as a complex quantity as:

$$y = y_d + jy_q \tag{23}$$

This vector is used as a desired value. The complex observation matrix Z is introduced by:

$$Z(kT_s) = \left[e^{j\omega_0 kT_s}, \quad e^{j2\omega_0 kT_s}, \quad \ldots \quad e^{jN\omega_0 kT_s}, \quad 1, \quad -kT_s \right]^T \tag{24}$$

ω_0 is the center angular frequency (rad/sec), defined as $\omega_0 = 2\pi f_0$. The complex harmonic vector to be tracked at k^{th} sample is $\Gamma(kT_s)$ and is defined as:

$$\Gamma(kT_s) = \left[A_1(kT_s), \quad A_2(kT_s) \quad \ldots \quad A_N(kT_s), \quad A_{N+1}(kT_s), \quad A_{N+2}(kT_s) \right]^T \tag{25}$$

where $A_1(kT_s)$ is the complex phasorial expression of center frequency in the *d-q* frame. According to LMS rule, weight update is:

$$\Gamma(kT_s) = \Gamma(kT_s - T_s) + \alpha \cdot \vec{e}(kT_s - T_s) \frac{Z(kT_s - T_s)}{Z^T(kT_s - T_s) \cdot Z(kT_s - T_s)} \tag{26}$$

$\vec{e}(kT_s)$ is the complex error obtained as follows:

$$\begin{aligned} Y_s(kT_s) &= \Gamma^T(kT_s) \times Z(kT_s) \\ \vec{e}(kT_s) &= Y_s(kT_s) - y(kT_s) \end{aligned} \tag{27}$$

where $Y_s(kT_s)$ is the complex estimation of the actual values of $y(kT_s)$ in *d-q* frame.

It should be noted that under conditions where power system operates with the nominal frequency, $A_1(kT_s)$ is a constant vector, which does not rotate with respect to the time in the complex frame. When the base frequency changes, $A_1(kT_s)$ becomes a rotating vector. It is the result of the fact that when the base frequency changes, $A_1(kT_s)$ components appear as modulated signals and their carrier is the occurred frequency-drift. Therefore, the rate of this rotation is the key element to track the frequency deviation from the center frequency. The frequency deviation (f_1) is achieved by normalizing and differentiating $A_1(kT_s)$. For the types of power swing events studied here, it has been found that the non-fundamental components cannot be characterized as harmonics. A middle-filter is, therefore, required so that the signal is dominated by the fundamental component. The middle-filter, used here, is the FIR Hamming type filter as has been used in [18]. It is the same which has been used in Section 2. $A_1(kT_s)$ passes through the FIR Hamming window and $Ah_1(kT_s)$ is obtained as:

$$Ah_1(kT_s) = \sum_{i=1}^{N_0} \left[A_1\left((i-k+1)T_s\right) \cdot H(i) \right] \tag{28}$$

$Ah_1(kT_s)$ should be normalized to produce the rotating operator ($e^{(j2\pi f_1 kT_s)}$). $e^{(j2\pi f_1 kT_s)}$ stands for a normal rotating vector which its amplitude is unity and f_1 is the frequency deviation. Therefore, complex normalized rotating state vector $An_1(kT_s)$ is obtained as:

$$An_1(kT_s) = \frac{Ah_1(kT_s)}{abs\left(Ah_1(kT_s)\right)} \tag{29}$$

where $abs(x)$ stands for absolute value of x. By using the first-order discrete differentiator, f_1 is obtained as:

$$f_1(kT_s) = \left(\frac{1}{j2\pi \cdot An_1 \cdot k \cdot T_s}\right) \cdot \left(\frac{An_1(kT_s) - An_1(kT_s - T_s)}{T_s}\right) \tag{30}$$

It can be seen that observation matrix size and the parameters to be estimated have been reduced to $(N+2)$ elements in comparison with the simple ADALINE which uses $(2N+2)$ elements. Furthermore, owing to the fact that data from three phases are combined, the most important aspect of the proposed technique is that the convergence speed is considerably improved. After all, decaying DC offset is effectively eliminated by applying stationary-axes Park transformation and using CADALINE. Fig. 3 shows the complex ADALINE structure to track fundamental frequency.

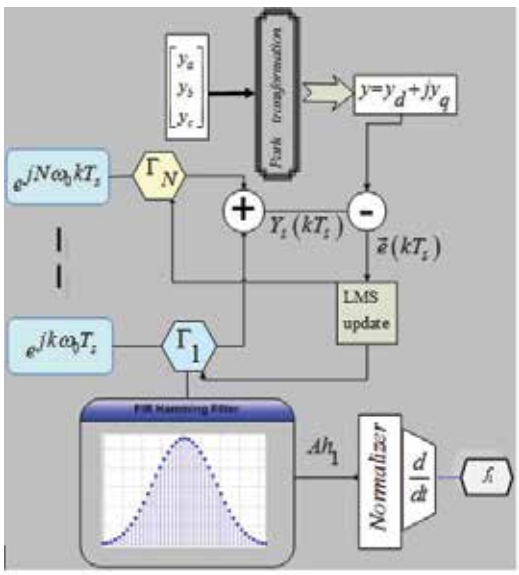


Fig. 3. Complex ADALINE structure to track fundamental frequency

4. Review of Kalman and DFT approaches

4.1 Kalman filter to track fundamental frequency

Kalman filter has also been used to track fundamental frequency in power system. Consider the following deterministic state-variable equation for a periodic signal having harmonic components up to N^{th} order with samples z_k, at time t_k, $(2n+1)$ samples per period.

$$\begin{aligned} x_{k+1} &= F \times x_k \\ z_k &= Q \times x_k \end{aligned} \tag{31}$$

where $(2N+1)$-dimensional state vector x_k is as follows:
$x_k(2i-1)$: real component of the i^{th} harmonic phasor,
$x_k(2i)$: imaginary component of the i^{th} harmonic phasor,
$x_k(2i+1)$: decaying DC component,
where the i^{th} element of x_k is represented by $x_k(i)$, and F is:

$$F=\begin{bmatrix} f(1\psi) & 0 & \cdots & 0 & 0 \\ 0 & f(2\psi) & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & f(N\cdot\psi) & \vdots \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \tag{32}$$

where $\psi=\omega_0 T_s$, ω_0 is the fundamental supply angular frequency in (rad/sec) and T_s is the sampling interval in seconds.

$$f(i\psi)=\begin{bmatrix} \cos(i\psi) & -\sin(i\psi) \\ \sin(i\psi) & \cos(i\psi) \end{bmatrix} \quad i=1,2,\ldots,N \tag{33}$$

and Q is a (1 × (2n + 1)) matrix which gives the connection between the measurement (z_k) and the state vector (x_k). The sampled value of the signal is considered to be the sum of the real components of the harmonic phasors and the decaying DC component. Therefore, Q is given by:

$$Q=[1,0,1,0,\cdots,1,0,1] \tag{34}$$

The harmonic components h_i (RMS) are given by:

$$h_i^2=\frac{\left(x_k^2(2i-1)+x_k^2(2i)\right)}{2} \quad i=1,2,\ldots,N \tag{35}$$

The problem of estimating the present state of the signal model (Eq. 31) from measurements (z_k) involves the design of standard state observers [33]. The observer state can be represented by:

$$\hat{x}_{k+1}=F\times\hat{x}_k+P\times\left(z_k-Q\hat{x}_k\right) \tag{36}$$

where $\hat{x}_k$ denotes the estimate of the state vector x_k and P is the observer gain matrix. The primary objective in choosing P is to obtain a stable observer, which is achieved by assigning the eigenvalues of the matrix $F-PQ$ within the unit circle. The locations of the eigenvalues determine, among other things, the transient response of the observer. For the purpose of frequency tracking, the speed of response and tracking ability are of particular importance. After studying various choices, the following case is considered.

$$\begin{aligned} P=[&0.248,0.0513,0.173,0.046,0.0674,0.0434,0.00916,\\ &0.0236,-0.000415,0.113,0.0802]^T \end{aligned} \tag{37}$$

To estimate fundamental frequency, the approach is based on DSPOC which has been described in Eqs. 15–20 is used.

4.2 DFT filter to track fundamental frequency

Under situation of frequency change, the k^{th} sample of fundamental voltage or current signal is described as denoted in Eq. 15. By using a DFT dynamic window, parameters x_1 and x_2 in Eq. 16 can be achieved as follows:

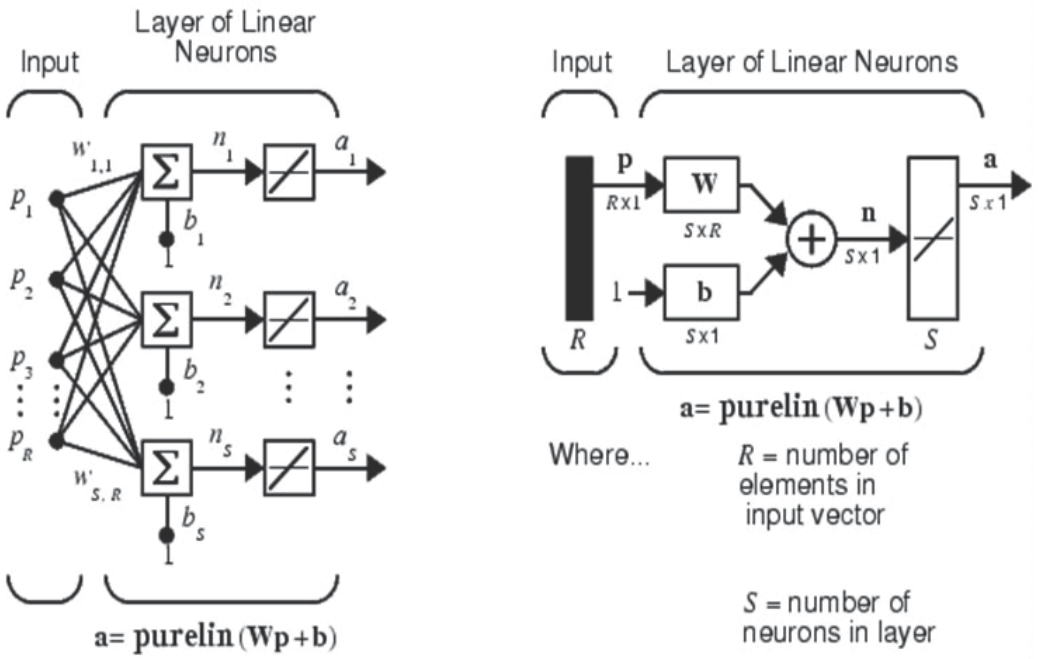
$$x_1(kT_s) = (\frac{2}{N_s}) \sum_{k=0}^{N_s-1} \left[s(kT_s) \cdot sin(2\pi \frac{k}{N_s}) \right]$$

$$x_2(kT_s) = (\frac{2}{N_s}) \sum_{k=0}^{N_s-1} \left[s(kT_s) \cdot cos(2\pi \frac{k}{N_s}) \right] \quad (38)$$

The fundamental frequency tracking process includes the same approach that has been expressed in Eqs. 15–20.

5. Adaptive linear element

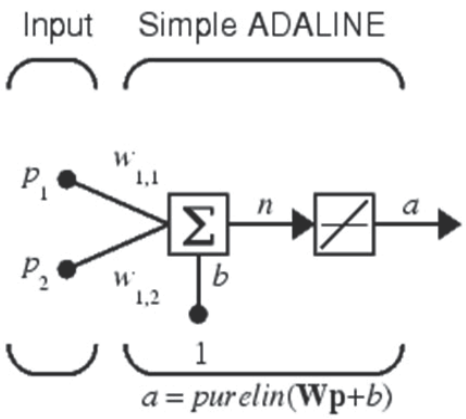
ADALINE (Adaptive Linear Neuron or later Adaptive Linear Element) is a single layer neural network as the 'least mean square' (LMS) learning procedure, also known as the delta



rule. It was developed by ProfessorBernard Widrow and his graduate student Ted Hoff at Stanford University in 1960. It is based on the McCulloch–Pitts neuron. It consists of a weight, a bias and a summation function.The difference between Adaline and the standard (McCulloch-Pitts) perceptron is that in the learning phase the weights are adjusted according to the weighted sum of the inputs (the net). In the standard perceptron, the net is passed to the activation (transfer) function and the function's output is used for adjusting the weights. The main functional difference with the perceptron training rule is the way the output of the system is used in the learning rule. The perceptron learning rule uses the output of the threshold function (either -1 or +1) for learning. The delta-rule uses the net output without further mapping into output values -1 or +1. The ADALINE network shown below has one layer of S neurons connected to R inputs through a matrix of weights W.
This network is sometimes called a MADALINE for Many ADALINEs. Note that the figure on the right defines an *S*-length output vector a.
The Widrow-Hoff rule can only train single-layer linear networks. This is not much of a disadvantage, however, as single-layer linear networks are just as capable as multilayer linear networks. For every multilayer linear network, there is an equivalent single-layer linear network.

5.1 Single ADALINE

Consider a single ADALINE with two inputs. The following figure shows the diagram for this network.



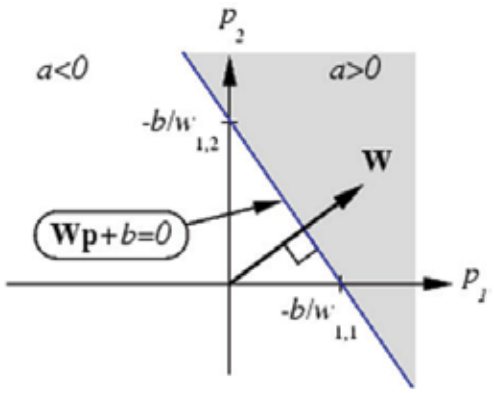
The weight matrix W in this case has only one row. The network output is:

$$a = purelin(n) = purelin(Wp + b) = (Wp + b) \tag{39}$$

Equation a can be written as follows:

$$a = w_{1,1}p_1 + w_{1,2}p_2 + b \tag{40}$$

Like the perceptron, the ADALINE has a decision boundary that is determined by the input vectors for which the net input n is zero. For n = 0 the equation Wp + b = 0 specifies such a decision boundary, as shown below:



Input vectors in the upper right gray area lead to an output greater than 0. Input vectors in the lower left white area lead to an output less than 0. Thus, the ADALINE can be used to classify objects into two categories.
However, ADALINE can classify objects in this way only when the objects are linearly separable. Thus, ADALINE has the same limitation as the perceptron.

5.2 Networks with linear activation functions: the delta rule

For a single layer network with an output unit with a linear activation function the output is simply given by:

$$y = \sum_{i=1}^{n} w_i x_i + \theta \tag{41}$$

Such a simple network is able to represent a linear relationship between the value of the output unit and the value of the input units. By thresholding the output value, a classifier can be constructed (such as Widrow's Adaline), but here we focus on the linear relationship and use the network for a function approximation task. In high dimensional input spaces the network represents a (hyper) plane and it will be clear that also multiple output units may be defined. Suppose we want to train the network such that a hyper plane is fitted as well as possible to a set of training samples consisting of input values d_p and desired (or target) output values d_p. For every given input sample, the output of the network differs from the target value d_p by $\left(d^p - y^p\right)$ where y_p is the actual output for this pattern. The delta-rule now uses a cost- or error-function based on these differences to adjust the weights. The error function, as indicated by the name least mean square, is the summed squared error. That is, the total error E is denoted to be:

$$E = \sum_p E^p = \frac{1}{2}\sum_p \left(d^p - y^p\right)^2 \tag{42}$$

Where the index p ranges over the set of input patterns and E_p represents the error on pattern p. The LMS procedure finds the values of all the weights that minimize the error function by a method called gradient descent. The idea is to make a change in the weight proportional to the negative of the derivative of the error as measured on the current pattern with respect to each weight:

$$\Delta_p w_j = -\gamma \frac{\partial E^p}{\partial w_j} \tag{43}$$

where γ is a constant of proportionality. The derivative is

$$\frac{\partial E^p}{\partial w_j} = \frac{\partial E^p}{\partial y^p}\frac{\partial y^p}{\partial w_j} \tag{44}$$

$$\frac{\partial y^p}{\partial w_j} = x_i \tag{45}$$

Because of the linearity, $\frac{\partial E^p}{\partial w_j}$ is as follows:

$$\frac{\partial E^p}{\partial w_j} = -\left(d^p - E^p\right) \tag{46}$$

Where $\delta^p = d^p - E^p$ is the difference between the target output and the actual output for pattern p .The delta rule modifies weight appropriately for target and actual outputs of either polarity and for both continuous and binary input and output units. These characteristics have opened up a wealth of new applications.

6. Simulation results

Simulation examples include the following three categories. Numerical simulations are represented in Section 5.1. for two cases, simulation in PSCAD/EMTDC software is presented in Section 5.2. Lastly, Section 5.3. presents practical measurement of a real fault incidence in Fars province, Iran.

6.1 Simulated signals

Herein, a disturbance is simulated at time 0.3 sec. Three-phase non-sinusoidal unbalanced signals, including decaying DC offset and third harmonic, are produced as:

$$\begin{cases} V_A = 220\sin(\omega_0 t) \\ V_B = 220\sin(\omega_0 t - \frac{2\pi}{3}) \\ V_C = 220\sin(\omega_0 t + \frac{2\pi}{3}) \end{cases} \quad 0 \le t \le 0.3 \tag{47}$$

After disturbance at 0.3 sec, signals are:

$$\begin{cases} V_A = 400\sin(\omega_x t) + 40\sin(3\omega_x t) + 400e^{(-10t)} \\ V_B = 800\sin(\omega_x t - \frac{2\pi}{3}) + 60\sin(3\omega_x t) + 800e^{(-10t)} \\ V_C = 800\sin(\omega_x t + \frac{2\pi}{3}) + 20\sin(3\omega_x t) \end{cases} \quad 0.3 \le t \le 0.6 \tag{48}$$

where ω_0 is the base angular frequency and ω_x is the actual angular frequency after disturbance.

6.1.1 Case 1

In this case, a 1-Hz frequency deviation occurs and tracked frequency using CADALINE, ADALINE, Kalman, and DFT approaches is revealed in Fig. 4; three-phase signals are shown in Fig. 5. Estimation error percentage according to the samples fed to each algorithm after frequency drift is shown in Fig. 6. Second set of samples including 100 samples, equivalent to two and half cycles, which is fed to all algorithms is magnified in Fig. 6. It can be seen that CADALINE converges to the real value after first 116 samples, less than three power cycles, with error of -0.4 %; and reaches a perfect estimation after having more few samples. Other methods' estimations are too fare from real value in this snapshoot. DFT, ADALINE and Kalman respectively need 120, 200 and 360 samples to reach less than one percent error in estimating the frequency drift. It should be considered that for 2.4-kHz sampling frequency and power system frequency of 60 Hz, each power cycle includes 40 samples. The complex normalized rotating state vector $An_1(kT_s)$ with respect to time and in *d-q* frame is shown in Fig. 7. It has been seen that for 1-Hz frequency deviation ($f_1 = 1$ Hz), CADALINE has the best convergence response in terms of speed and over/under shoot. ADALINE method convergence speed is half that in the CADALINE and shows a really high overshoot. Besides, Kalman approach shows the biggest error. in the first 7 power system cycles, it converges to 61.7 Hz instead of 61 Hz and its computational burden is considerably higher than other methods. In this case, presence of a long-lasting decaying DC offset affects the DFT performance. Consequently, its convergence speed and overshoot are not as improved as CADALINE.

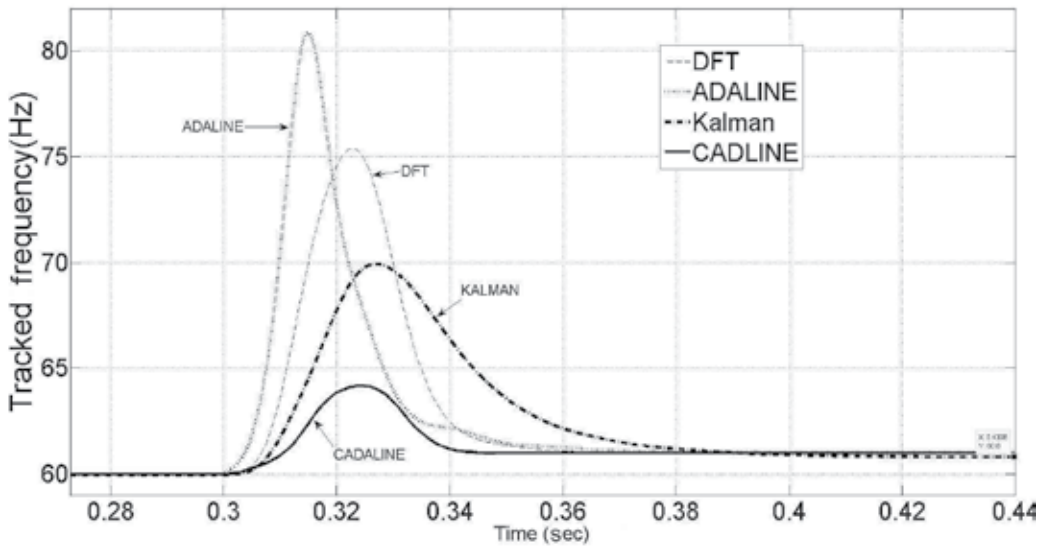


Fig. 4. Tracked frequency (Hz)

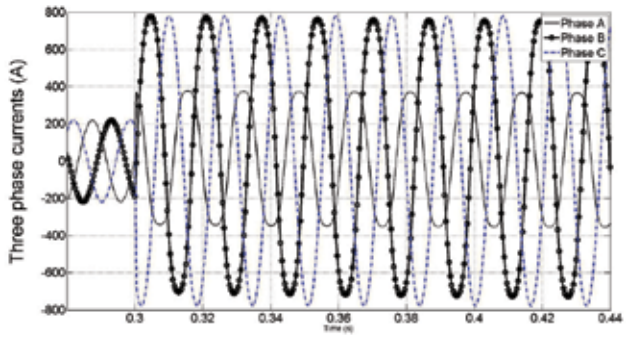


Fig. 5. Three-phase signals

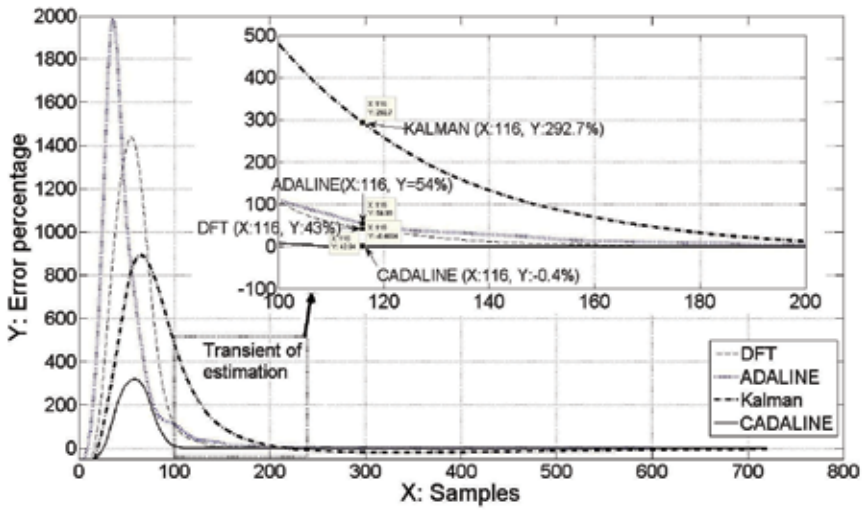


Fig. 6. Estimation error percentage according to samples fed to each algorithm after frequency drift

As can be seen in Fig. 7, $An_1(kT_s)$ starts rotation simultaneously when the frequency changes at time 0.3 sec.

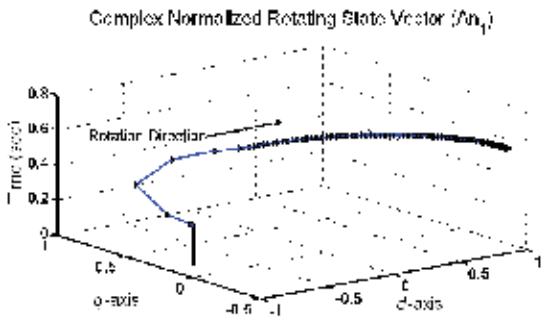


Fig. 7. Complex normalized rotating state vector (An_1)

6.1.2 Case 2

In this case, a three-phase balanced voltage is simulated numerically. The only change applied is a step-by-step 1-Hz change in fundamental frequency to study the steady-state response of the proposed method when the power system operates under/over frequency conditions. The three-phase signals are:

$$\begin{cases} V_A = 220\sin(\omega_x t) \\ V_B = 220\sin(\omega_x t - \frac{2\pi}{3}) \\ V_C = 220\sin(\omega_x t + \frac{2\pi}{3}) \end{cases} \tag{49}$$

where $\omega_x = 2\pi f_x$, and values of f_x are shown in Table I. The range of frequency that has been studied here is 50–70 Hz. Results are revealed in Table I and average convergence time is shown in Fig. 8 for CADALINE, ADALINE, Kalman filter and DFT approaches. The results from this section can give an insight into the number of samples that each algorithm needs to converge to a reasonable estimation. According to the fact that each power cycle is equivalent to 40 samples, average number of samples that is needed for each algorithm to have estimation with less than one percent error is represented in Table I.

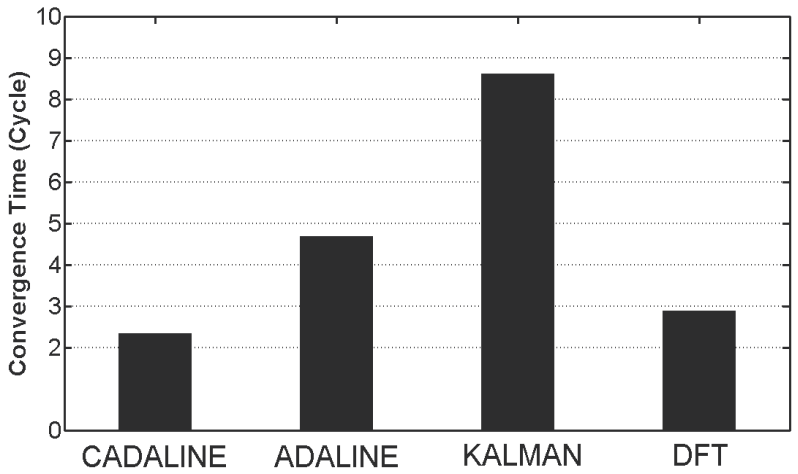


Fig. 8. Average convergence time (cycles) to track static frequency changes

6.2 Simulation in PSCAD/EMTDC software

In this case, a three-machine system controlled by governors is simulated in PSCAD/EMTDC software, shown in Fig. 9. Information of the simulated system is given in Appendix I. A three-phase fault occurs at 1 sec. Real frequency changes, estimation by use of ADALINE, CADALINE and Kalman approaches are shown in Fig. 10. Instead of DFT method, the frequency measurement module (FMM) performance which exists in PSCAD library is compared with the presented methods. Phase-A voltage signal is shown in Fig. 11.

	Approaches			
f_x(Hz)	**CADALINE**	**KALMAN**	**ADALINE**	**DFT**
70	95	360	202	111
69	97	421	188	114
68	93	358	186	118
67	90	384	187	114
66	95	385	178	114
65	97	305	138	139
64	92	361	211	114
63	93	328	193	116
62	98	430	206	115
61	96	360	231	116
60	92	385	220	112
59	83	234	155	97
58	81	281	181	116
57	88	313	197	117
56	98	216	178	123
55	97	377	192	117
54	96	336	206	122
53	90	331	195	114
52	96	290	190	108
51	96	374	184	120
50	105	405	113	112

Table I Samples needed to estimate with 1 percent error for 50-70 frequency range

The complex normalized rotating state vector ($An_1(kT_s)$) is shown in Fig. 12. The best transient response and accuracy belongs to ADALINE and CADALINE, but CADALINE has faster response with a considerable lower overshoot, as can be seen in Fig. 10. Kalman

approach has a suitable response in this case, but its error and overshoot in estimating frequency are bigger than that in CADALINE. The PSCAD FMM shows drastic fluctuations in comparison with other methods proposed and reviewed here.

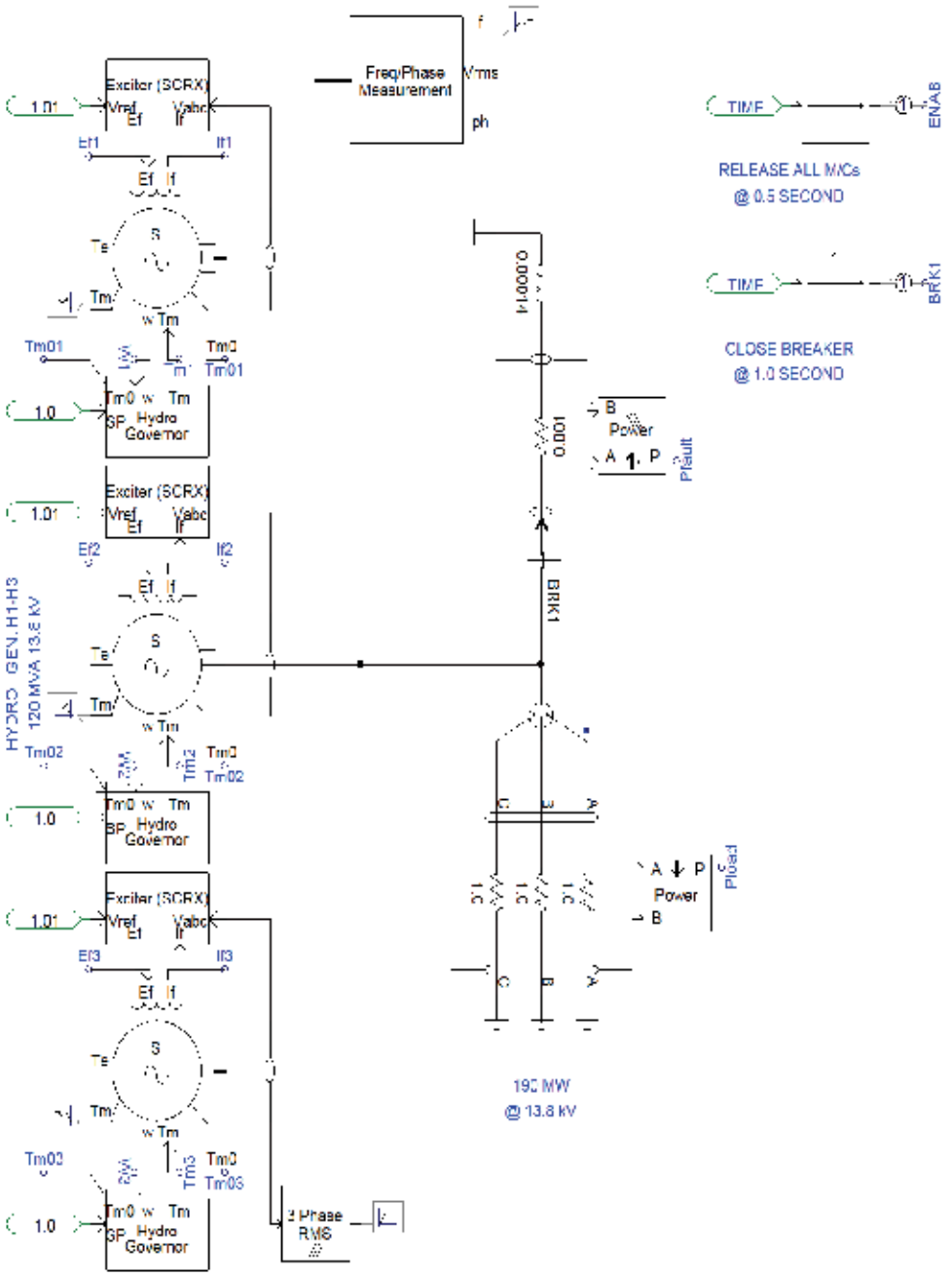


Fig. 9. A three-machine connected system simulated in PSCAD/EMTDC software

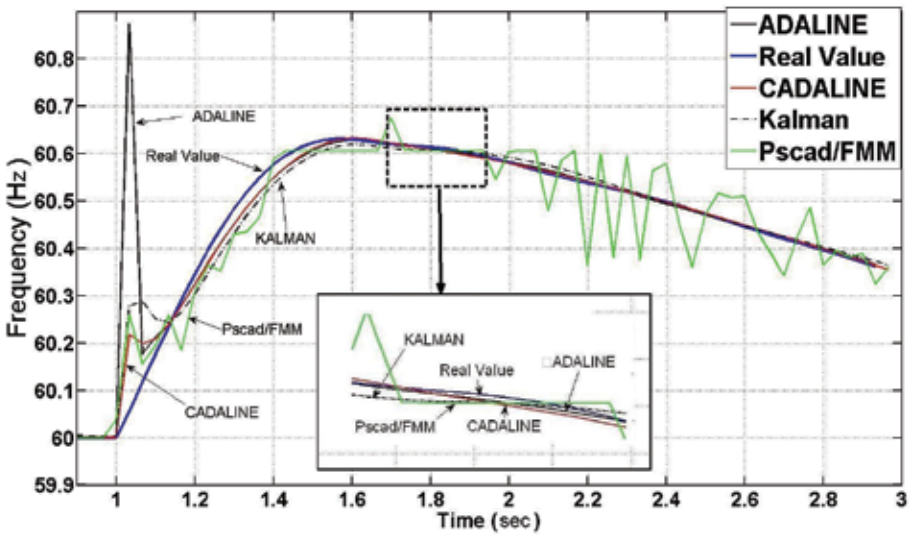


Fig. 10. Tracked frequency (Hz)

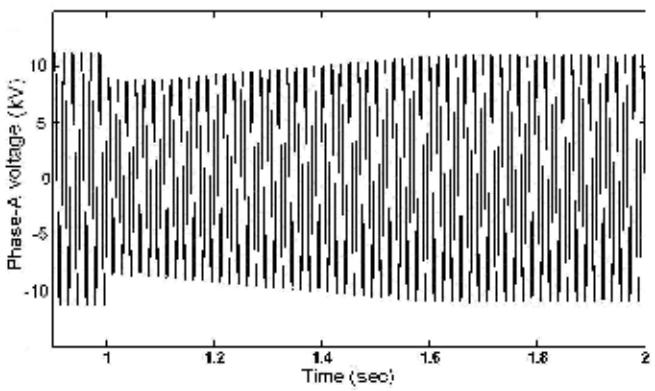


Fig. 11. Phase-A voltage (kV)

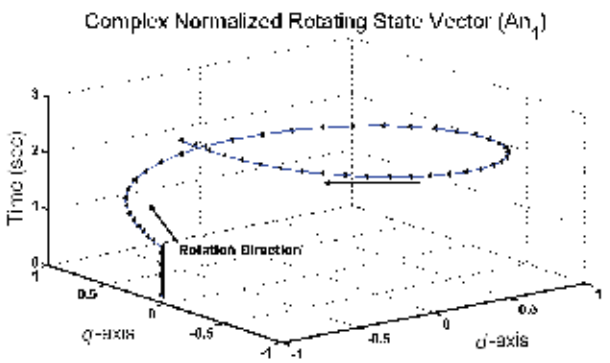


Fig. 12. Complex normalized rotating state vector (An_1)

6.3 Practical study

In this case, a practical example is represented. Voltage signal measurements are applied from the Marvdasht power station in Fars province, Iran. The recorder's sampling frequency (f_s) is 6.39 kHz and fundamental frequency of power system is 50 Hz. A fault between pahse-C and groung occurred on 4 March 2006. The fault location was 46.557 km from Arsanjan substation. Main information on the Marvdasht 230/66 kV station and other substation supplied by this station is given in Tables II and III, presented in Appendix II. Fig. 13 shows the performance of CADALINE, ADALINE, Kaman and DFT approaches. Besides, phase-C voltage and residual voltage are revealed in Fig. 14 (A) and Fig. 14 (B) respectively. Complex normalized rotating state vector (An_1) is shown in Fig. 15.

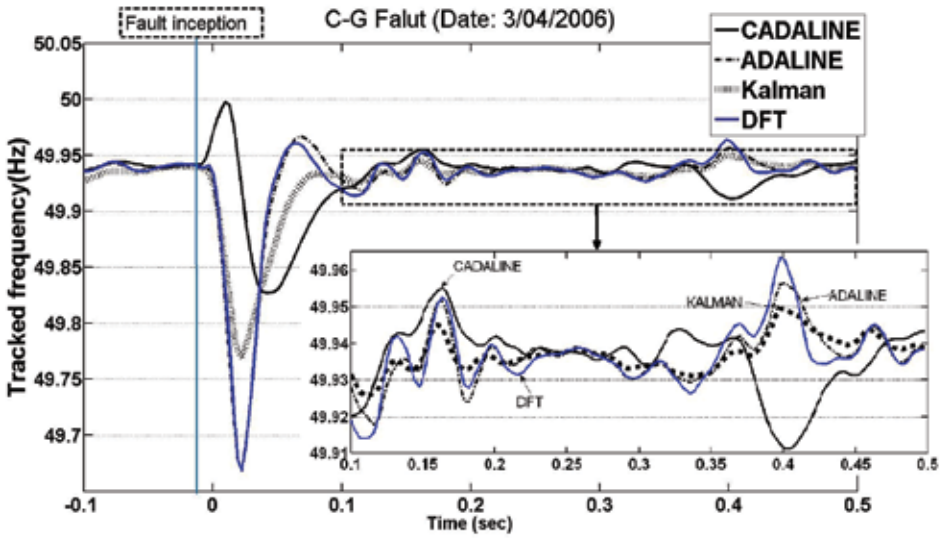


Fig. 13. Tracked frequency (Hz), case V.C.

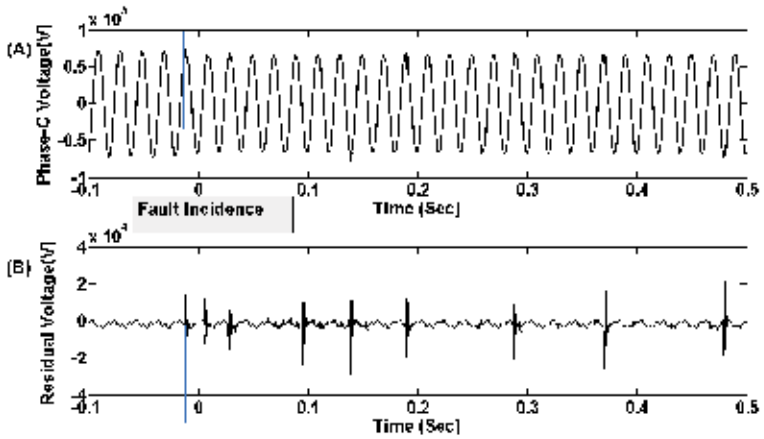


Fig. 14. (A): phase-C voltage and (B): residual voltage, case V.C.

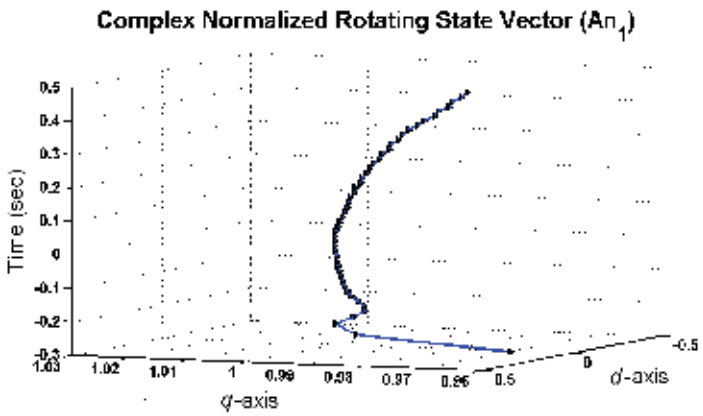


Fig. 15. Complex normalized rotating state vector (An_1), case V.C.

7. Conclusion

This section proposes an adaptive approach for frequency estimation in electrical power systems by introducing a novel complex ADALINE (CADALINE) structure. The proposed technique is based on tracking and analyzing a complex rotation state vector in *d*-*q* frame that appears when a frequency drift occurs. This method improves the convergence speed both in steady states and dynamic disturbances which include changes in base frequency of power system. Furthermore, the proposed method reduces the size of the state observer vector that has been used by simple ADALINE structure in other references. The numerical and simulation examples have verified that the proposed technique is far more robust and accurate in estimating the instantaneous frequency under various conditions compared with methods that have been reviewed in this section.

8. Appendices

8.1 Appendix I. multi-machine system information simulated in PSCAD/EMTDC software

1. Basic data of all generators are:
   Number of machines: 3
   Rated line-to-neutral voltage (RMS): 7.967 [kV]
   Rated line current (RMS): 5.02 [kA]
   Base angular frequency: 376.991118 [rad/sec]
   Inertia constant: 3.117 [s]
   Mechanical friction and windage: 0.04 [p.u.]
   Neutral series resistance: 1.0E5 [p.u.]
   Neutral series reactance: 0 [p.u.]
   Iron loss resistance: 300.0 [p.u.]
2. Fault characteristics:
   Fault inductance: 0.00014 [H]
   Fault resistance: 0.0001 [Ω]
3. Load characteristics:

Load active power: 190 [MW]
Load nominal line-to-line voltage: 13.8 [kV]

8.2 Appendix. II

Main information on the Marvdasht 230/66 (kV) station and other substation supplied by this station is given in Tables II and III.

NO.	SUBSTATION NAME	FEEDER NO. TAG	3-PHASE SHORT CIRCUIT CAPACITY (MVA)	1-PHASE SHORT CIRCUIT CAPACITY (MVA)
1	Marvdasht 230/66 (kV)	-	1460	1184
2	Marvdasht City	602	896	640
3	Mojtama	601	631	423
4	Kenare	607	1005	718
5	Sahl Abad	603	751	500
6	Dinarloo	604	203	121
7	Seydan	608	381	237
8	Arsanjan	605	145	84

Table II Marvdasht substation capacities

NO.	SUBSTATION NAME	FEEDER NO. TAG	3-PHASE SHORT CIRCUIT CURRENT (kA)	1-PHASE SHORT CIRCUIT CURRENT (kA)	$\lvert Z \rvert$ (Ω)
1	Marvdasht 230/66 (kV)	-	12.77169	10.35731	2.983562
2	Marvdasht City	602	7.837967	5.598548	4.861607
3	Mojtama	601	5.519818	3.70029	6.903328
4	Kenare	607	8.79147	6.280871	4.334328
5	Sahl Abad	603	6.569546	4.373866	5.800266
6	Dinarloo	604	1.775789	1.058475	21.45813
7	Seydan	608	3.332886	2.073212	11.43307
8	Arsanjan	605	1.268421	0.734809	30.04138

Table III Marvdasht substation three-phase and single-phase short circuit capacities and impedances ($\lvert Z \rvert$)

9. References

J.K. Wu, Frequency tracking techniques of power systems including higher order harmonics devices, Proceedings of the Fifth IEEE International Caracas Conference, vol. 1, (3–5 Nov. 2004), pp. 298–303.

M. Akke, Frequency estimation by demodulation of two complex signals, IEEE Trans. Power Del., vol. 12, no. 1, (Jan. 1997), pp. 157–163.

P.J. Moore, R.D. Carranza and A.T. Johns, Model system tests on a new numeric method of power system frequency measurement, IEEE Trans. Power Del., vol. 11, no. 2, (Apr. 1996), pp. 696–701.

M.M. Begovic, P.M. Djuric, S. Dunlap and A.G. Phadke, Frequency tracking in power networks in the presence of harmonics, IEEE Trans. Power Del., vol. 8, no. 2, (April 1993), pp. 480–486.

C.T. Nguyen and K.A. Srinivasan, A new technique for rapid tracking of frequency deviations based on level crossings, IEEE Trans. Power App. Syst., vol. 103, no. 8, (April 1984), pp. 2230–2236.

I. Kamwa and R. Grondin, Fast adaptive schemes for tracking voltage phasor and local frequency in power transmission and distribution systems, IEEE Trans. Power Del., vol. 7, no. 2, (April 1992), pp.789–795.

M.S. Sachdev and M.M. Giray, A least error square technique for determining power system frequency, IEEE Trans. Power App. Syst., vol. 104, no. 2, (Feb. 1985), pp. 437–443.

M.M. Giray and M.S. Sachdev, Off-nominal frequency measurements in electric power systems, IEEE Trans. Power Del., vol. 4, no. 3, (July 1989), pp. 1573–1578.

V.V. Terzija, M.B. Djuric and B.D. Kovacevic, Voltage phasor and local system frequency estimation using Newton-type algorithm, IEEE Trans. Power Del., vol. 9, no. 3, (Jul. 1994), pp. 1368–1374.

M.S. Sachdev, H. C. Wood and N. G. Johnson, Kalman filtering applied to power system measurements for relaying, IEEE Trans. Power App. Syst., vol. 104, no. 12, (Dec. 1985), pp. 3565–3573.

A.A. Girgis and T.L.D. Hwang, Optimal estimation of voltage phasors and frequency deviation using linear and nonlinear Kalman filter: Theory and limitations, IEEE Tran. Power App. Syst., vol. 103, no. 10, (1984), pp. 2943–2949.

A.A. Girgis and W.L. Peterson, Adaptive estimation of power system frequency deviation and its rate of change for calculating sudden power system overloads, IEEE Trans. Power Del., vol. 5, no. 2, (Apr. 1990), pp. 585–594.

T. Lobos and J. Rezmer, Real time determination of power system frequency, IEEE Trans. Instrum. Meas., vol. 46, no. 4, (Aug. 1997), pp. 877–881.

A.G. Phadke, J.S. Thorp and M.G. Adamiak, A new measurement technique for tracking voltage phasors, local system frequency, and rate of change of frequency, IEEE Trans. Power App. and Systems, vol. 102, no. 5, (May 1983), pp. 1025–1038.

J.Z. Yang and C.W. Liu, A precise calculation of power system frequency, IEEE Trans. Power Del., vol. 16, no. 3, (July 2001), pp. 361–366.

J.Z. Yang and C.W. Liu, A precise calculation of power system frequency and phasor, IEEE Trans. Power Del., vol. 15, no. 2, (Apr. 2000), pp. 494–499.

S.L. Lu, C.E. Lin and C.L. Huang, Power frequency harmonic measurement using integer periodic extension method, Elect. Power Syst. Res., vol. 44, no. 2, (1998), pp. 107–115.

P.J. Moore, R.D. Carranza and A.T. Johns, A new numeric technique for high speed evaluation of power system frequency, IEE Gen. Trans. Dist. Proc., vol. 141, no. 5 (Sept. 1994), pp. 529–536.

J. Szafran and W. Rebizant, Power system frequency estimation, IEE Gen. Trans. Dist. Proc., vol. 145, no. 5, (Sep. 1998), pp. 578–582.

A.A. Girgis and F.M. Ham, A new FFT-based digital frequency relay for load shedding, IEEE Trans. Power App. Syst., vol. 101, no. 2, (Feb. 1982), pp. 433–439.

H.C. Lin and C. S. Lee, Enhanced FFT-based parameter algorithm for simultaneous multiple harmonics analysis, IEE Gen. Trans. Dist. Proc., vol. 148, no. 3, (May 2001), pp. 209–214.

W.T. Kuang and A.S. Morris, Using short-time Fourier transform and wavelet packet filter banks for improved frequency measurement in a Doppler robot tracking system, IEEE Trans. Instrum. Meas., vol. 51, no. 3, (June 2002), pp. 440–444.

V.L. Pham and K. P. Wong, Wavelet-transform-based algorithm for harmonic analysis of power system waveforms, IEE Gen. Trans. Dist. Proc., vol. 146, no. 3, (May 1999), pp. 249–254.

V.L. Pham and K.P. Wong, Antidistortion method for wavelet transform filter banks and nonstationary power system waveform harmonic analysis, IEE Gen. Trans. Dist. Proc., vol. 148, no. 2, (March 2001), pp. 117–122.

M. Wang and Y. Sun, A practical, precise method for frequency tracking and phasor estimation, IEEE Trans. Power Del., vol. 19, no. 4, (Oct. 2004), pp. 1547–1552.

D.W.P. Thomas and M.S. Woolfson, Evaluation of a novel frequency tracking method, Transmission and Distribution Conference, vol. 1, (11–16 April 1999), pp. 248–253.

M.I. Marei, E.F. El-Saadany and M.M.A. Salama, A processing unit for symmetrical components and harmonics estimation based on a new adaptive linear combiner structure, IEEE Trans. Power Del., vol. 19, no. 3, (July 2004), pp. 1245–1252.

P.J. Moore, J.H. Allmeling and A.T. Johns, Frequency relaying based on instantaneous frequency measurement, IEEE Trans. Power Del., vol. 11, no. 4, (Oct. 1996), pp. 1737–1742.

D.W.P. Thomas and M.S. Woolfson, Evaluation of frequency tracking methods, IEEE Trans. Power Del., vol. 16, no. 3, (July 2001), pp. 367–371.

G. J. Retter, Matrix and Space-Phasor Theory of Electrical Machines, Akademiai Kiado, Budapest, Rumania, (1987).

G.H. Hostetter, Recursive discrete Fourier transformation, IEEE Trans. Speech Audio Process., vol. 28, no, 2, (1980), pp. 184–190.

R.R. Bitmead, A.C. Tsoi and P.J. Parker, A Kalman filtering approach to short-time Fourier analysis, IEEE Trans. Speech Audio Process., vol. 34, no. 6, (1986), pp. 1493–1501

T. Kailath, Linear Systems, Prentice-Hall, New Jersey, (1980).

14

Application of ANN to Real and Reactive Power Allocation Scheme

S.N. Khalid, M.W. Mustafa, H. Shareef and A. Khairuddin
Universiti Teknologi Malaysia
Malaysia

1. Introduction

This chapter describes the implementation of ANN for real and reactive power transfer allocation. The 25 bus equivalent power system of south Malaysia region and IEEE 118 bus system are used to demonstrate the applicability of the ANN output compared to that of the Modified Nodal Equations (MNE) which is used as trainers for real and reactive power allocation. The basic idea is to use supervised learning paradigm to train the ANN. Then the descriptions of inputs and outputs of the training data for the ANN are easily obtained from the load flow results and each method used as teachers respectively. The proposed ANN based method provides promising results in terms of accuracy and computation time. Artificial intelligence has been proven to be able to solve complex processes in deregulated power system such as loss allocation. So, it can be expected that the developed methodology will further contribute in improving the computation time of transmission usage allocation for deregulated system.

2. Importance of deregulation

Deregulated power systems unbundles the generation, transmission, distribution and retail activities, which are traditionally performed by vertically integrated utilities. Consequently different pricing policies will exist between different companies. With the separate pricing of generation, transmission and distribution, it is necessary to find the capacity usage of different transaction happening at the same time so that a fair use-of-transmission-system charge can be given to individual customer separately. Then the transparency in the operation of deregulated power systems can be achieved. In addition, the capacity usage is another application for transmission congestion management. For that reason the power produced by each generator and consumed by each load through the network should be trace in order to have acceptable solution in a fair deregulated power system. In Malaysian scenario the future electricity sector will be highly motivated to be liberalized, i.e. deregulated. Thus the proposed methodology is expected to contribute significantly to the development of the local deregulated power system. Promising test results were obtained from the extensive case studies conducted for several systems. These results shall bring about some differences from those based on other methods as different view-points and approaches may end up with different results. This chapter is offering the solution by an alternative method with better computational time and acceptable accuracy. These findings bring a new perspective on the

subject of how to improve the conventional real power allocation methods. A technically sound approach, to determine the real power output of individual generators, is proposed. This method is based on current operating point computed by the usual laod flow code and basic equations governing the load flow in the network. The proposed MNE method has also been extended to reactive power allocation. The simulation results have also shown that of reactive power supply and reception in a power system is in conformity with a given operating point. The study results and analysis suggest that, the proposed MNE Method overcome problems arising in the conventional reactive allocation algorithms. From these two methods, the calculations results might bring about some differences because of the deviation in the concept applied by the proposed method. For example the proposed methods use each load current as a function of individual generators' current and voltage. This is different from the Chu's Method (Chu & Liao, 2004), where each load voltage is represented as a function of individual generators' voltage only. The proposed MNE Method for reactive power allocation is enhanced by utilizing ANN. When the performances of the developed ANN are investigated, it can be concluded that the developed ANN is more reliable and computationally faster than that of the MNE Method. Furthermore, the developed algorithms and tools for the proposed techniques have been used to investigate the actual 25 bus system of South Malaysia. The proposed methods have so far been focused on the viewpoint of suppliers. It is also very useful to develop and test the allocation procedures from the perspective of consumers. Both MNE Method and Chu's Method are equally suitable for modification in this respect. Additionally, this technique requires handling of future expansions into an ANN structure to make it a universal structure. Moreover adaptation of appropriate ANN architecture for the large real life test system is expected to deliver a considerable efficiency in computation time, especially during training processes. It may be a future work to analyze the performance of the algorithm for every change in the network topology.

3. Modified nodal equations method

The derivation, to decompose the load real powers into components contributed by specific generators starts with basic equations of load flow. Applying Kirchhoff's law to each node of the power network leads to the equations, which can be written in a matrix form as in equation (1) (Reta & Vargas, 2001):

$$\boldsymbol{I} = \boldsymbol{YV} \tag{1}$$

where:

V: is a vector of all node voltages in the system

I: is a vector of all node currents in the system

Y: is the Y-bus admittance matrix

The nodal admittance matrix of the typical power system is large and sparse, therefore it can be partitioned in a systematic way. Considering a system in which there are G generator nodes that participate in selling power and remaining $L= n\text{-}G$ nodes as loads, then it is possible to re-write equation (1) into its matrix form as shown in equation (2):

$$\begin{bmatrix} \boldsymbol{I}_G \\ \boldsymbol{I}_L \end{bmatrix} = \begin{bmatrix} \boldsymbol{Y}_{GG} & \boldsymbol{Y}_{GL} \\ \boldsymbol{Y}_{LG} & \boldsymbol{Y}_{LL} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_G \\ \boldsymbol{V}_L \end{bmatrix} \tag{2}$$

Solving for $\boldsymbol{I}_G$ and $\boldsymbol{I}_L$ using equation (2), the relationship can be obtained as shown in equations (3) and (4).

$$\boldsymbol{I}_G = \boldsymbol{Y}_{GG}\boldsymbol{V}_G + \boldsymbol{Y}_{GL}\boldsymbol{V}_L \tag{3}$$

$$\boldsymbol{I}_L = \boldsymbol{Y}_{LG}\boldsymbol{V}_G + \boldsymbol{Y}_{LL}\boldsymbol{V}_L \tag{4}$$

From equation (3), $\boldsymbol{V}_G$ can be solved as depicted in equation (5):

$$\boldsymbol{V}_G = \boldsymbol{Y}_{GG}^{-1}\left(\boldsymbol{I}_G - \boldsymbol{Y}_{GL}\boldsymbol{V}_L\right) \tag{5}$$

Now, on substituting equation (5) in equation (4) and rearranging it, the load currents can be presented as a function of generators' current and load voltages as shown in equation (6):

$$\boldsymbol{I}_L = \boldsymbol{Y}_{LG}\boldsymbol{Y}_{GG}^{-1}\boldsymbol{I}_G + \left(\boldsymbol{Y}_{LL} - \boldsymbol{Y}_{LG}\boldsymbol{Y}_{GG}^{-1}\boldsymbol{Y}_{GL}\right)\boldsymbol{V}_L \tag{6}$$

Then, the total real and reactive power $\boldsymbol{S}_L$ of all loads can be expressed as shown in equation (7):

$$\boldsymbol{S}_L = \boldsymbol{V}_L\boldsymbol{I}_L^{*} \tag{7}$$

where $(*)$ stands for conjugate,
Substituting equation (6) into equation (7) and solving for $\boldsymbol{S}_L$ the relationship as shown in equation (8) can be found;

$$\begin{aligned}\boldsymbol{S}_L &= \boldsymbol{V}_L\left(\boldsymbol{Y}_{LG}\boldsymbol{Y}_{GG}^{-1}\right)^{*}\boldsymbol{I}_G^{*} + \boldsymbol{V}_L\left(\left(\boldsymbol{Y}_{LL} - \boldsymbol{Y}_{LG}\boldsymbol{Y}_{GG}^{-1}\boldsymbol{Y}_{GL}\right)\boldsymbol{V}_L\right)^{*} \\ &= \mathrm{Re}\left\{\boldsymbol{V}_L\sum_{i=1}^{nG}\Delta\boldsymbol{I}_L^{*I_{Gi}} + \boldsymbol{V}_L\left(\left(\boldsymbol{Y}_{LL} - \boldsymbol{Y}_{LG}\boldsymbol{Y}_{GG}^{-1}\boldsymbol{Y}_{GL}\right)\boldsymbol{V}_L\right)^{*}\right\}\end{aligned} \tag{8}$$

where

$$\left(\boldsymbol{Y}_{LG}\boldsymbol{Y}_{GG}^{-1}\right)^{*}\boldsymbol{I}_G^{*} = \sum_{i=1}^{nG}\Delta\boldsymbol{I}_L^{*I_{Gi}}$$

nG : number of generators
Now, in order to decompose the load voltage dependent term further in equation (8), into components of generator dependent terms, the equation (10) derivations are used. A possible way to deduce load node voltages as a function of generator bus voltages is to apply superposition theorem. However, it requires replacing all load bus current injections into equivalent admittances in the circuit. Using a readily available load flow results, the equivalent shunt admittance Y_{Lj} of load node j can be calculated using the equation (9):

$$Y_{Lj} = \frac{1}{V_{Lj}}\left(\frac{S_{Lj}}{V_{Lj}}\right)^{*} \tag{9}$$

S_{Lj} is the load complex power on node j and V_{Lj} is the bus load voltage on node j. After adding these equivalences to the diagonal entries of Y-bus matrix, equation (1) can be rewritten as in equation (10):

$$\boldsymbol{V} = \boldsymbol{Y}^{'-1}\boldsymbol{I}_G \tag{10}$$

where $\boldsymbol{Y}'$ is the modified $\boldsymbol{Y}$.
Next, adopting equation (10) and taking into account each generator one by one, the load bus voltages contributed by all generators can be expressed as in equation (11):

$$\boldsymbol{V}_L = \sum_{i=1}^{nG} \Delta \boldsymbol{V}_L^* \boldsymbol{I}_{Gi} \tag{11}$$

It is now, simple mathematical manipulation to obtain required relationship as a function of generators dependent terms. By substituting equation (11) into equation (8), the decomposed load real and reactive powers can be expressed as depicted in equation (12):

$$\boldsymbol{S}_L = \boldsymbol{V}_L \sum_{i=1}^{nG} \Delta \boldsymbol{I}_L^* \boldsymbol{I}_{Gi} + \sum_{i=1}^{nG} \Delta \boldsymbol{V}_L^* \boldsymbol{I}_{Gi} \left(\left(\boldsymbol{Y}_{LL} - \boldsymbol{Y}_{LG} \boldsymbol{Y}_{GG}^{-1} \boldsymbol{Y}_{GL} \right) \boldsymbol{V}_L \right)^* \tag{12}$$

This equation shows that the real and reactive power of each load bus consists of two terms by individual generators. The first term relates directly to the generator's currents and the second term corresponds to their contribution to load voltages. With further simplification of equation (12), the real and reactive power contribution that load *j* acquires from generator *i* is as shown in equation (13):

$$S_{Lj} = \sum_{i=1}^{nG} S_{Lji}^{\Delta I_L} + \sum_{i=1}^{nG} S_{Lji}^{\Delta V_L} \tag{13}$$

where:

$S_{Lji}^{\Delta I_L}$: current dependent term of generator *i* to S_{Lj}

$S_{Lji}^{\Delta V_L}$: voltage dependent term of generator *i* to S_{Lj}

All procedures of the computation mentioned above can be demonstrated as a flowchart illustrated in Figure 1. Vector S_{Lj} is used as a target in the training process of the proposed ANN.

3. Test conducted on the practical 25-bus equivalent power system of south Malaysia region

3.1 Application of ANN to real and reactive power allocation method

This section presents test conducted on the practical 25-bus equivalent power system of south Malaysia region. An ANN can be defined as a data processing system consisting of a large number of simple, highly interconnected processing elements (artificial neurons) in an architecture inspired by the structure of the cerebral cortex of the brain (Tsoukalas & Uhrig, 1997). The processing elements consist of two parts. The first part simply sums the weighted inputs; the second part is effectively a nonlinear filter, usually called the activation function, through which the combined signal flow. These processing elements are usually organized into a sequence of layers or slabs with full or random connections between the layers. Neural network perform two major functions which are training (learning) and testing (recall). Testing occurs when a neural network globally processes the stimulus presented at its input buffer and creates a response at the output buffer. Testing is an integral part of the training process since a desired response to the network must be compared to the actual output to create an error function.

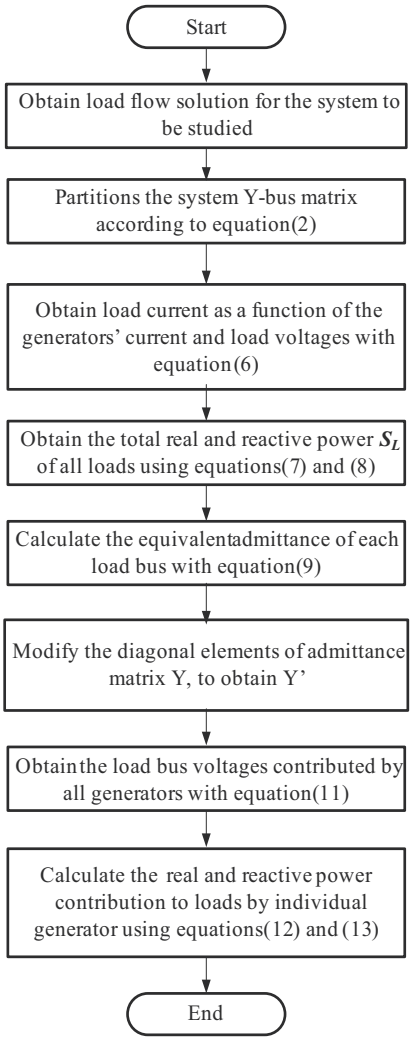


Fig. 1. Flow chart of the proposed real and reactive power allocation method

3.1.1 Structure of the proposed neural network in real and reactive power allocation method

In this work, 3 fully connected feedforward neural networks under MATLAB platform are utilized to obtain both real as well as reactive power transfer allocation results for the practical 25-bus equivalent power system of south Malaysia region as shown in Figure 2. This system consists of 12 generators located at buses 14 to 25 respectively. They deliver power to 5 loads, through 37 lines located at buses 1, 2, 4, 5, and 6 respectively. All discussions on designing of each of these ANN below are for this 25-bus equivalent system. Each network corresponds to four numbers of generators in the test system and each consists of two hidden layers and a single output layer. This means that in the first network is associated with four numbers of generator located at buses 14 to 17. This realization is adopted for simplicity and to reduce the training time of the neural networks.

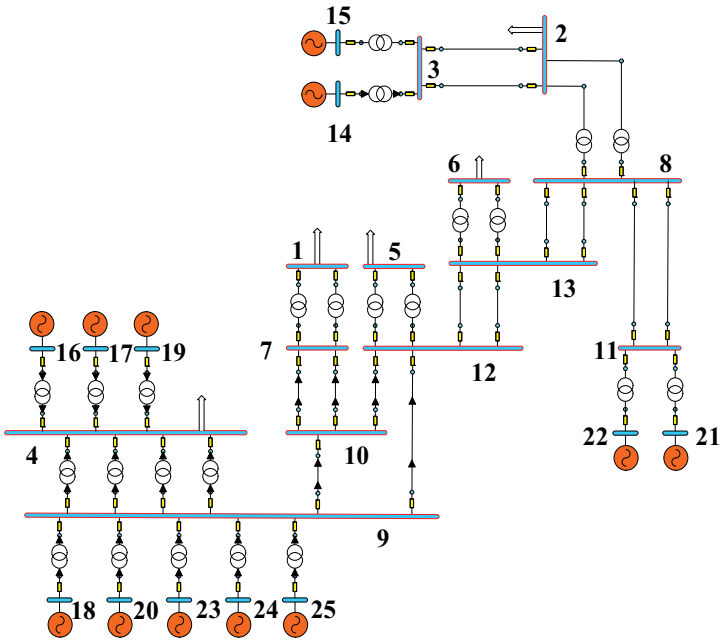


Fig. 2. Single line diagram for the 25-bus equivalent system of south Malaysia

The input samples for training is assembled using the daily load curve and performing load flow analysis for every hour of load demand. Again the load profile on hourly basis (Cheng, 1998) is utilized to produce 24 hours loads here also. Similarly the target vector for the training is obtained from the proposed method using MNE. Input data (D) for developed ANN contains independent variables such as real loads (P_1, P_2, P_4 to P_6) or reactive loads (Q_1, Q_2, Q_4 to Q_6) for real and reactive power transfer allocation respectively, bus voltage magnitude (V_1 to V_{13}) for both real as well as reactive power, real power (P_{line1} to P_{line37}) or reactive power (Q_{line1} to Q_{line37}) for line flows of real and reactive power transfer allocation respectively, and the target/output parameter (T) which is real or reactive power transfer between generators and loads placed at buses 1, 2, 4 to 6. This is considered as 20 outputs for both real as well as reactive power transfer allocation. Hence the networks have twenty output neurons. For the neural network 1, the first five neurons represent the contribution from generator 14 to the loads and the remaining outputs neurons correspond to the other three generators located at buses 15 to 17 respectively. Tables 1 and 2 summarize the description of inputs and outputs of the training data for each ANN for real and reactive power allocation respectively.

Input and Output (layer)	Neurons	Description (in p.u)
I_1 to I_5	5	Real loads
I_6 to I_{18}	13	Bus voltage magnitude
I_{19} to I_{55}	37	Real power for line flows
O_1 to O_{20}	20	Real power transfer between generators and loads

Table 1. Description of inputs and outputs of the training data for each ANN for real power

Input and Output (layer)	Neurons	Description (in p.u)
I_1 to I_5	5	Reactive loads
I_6 to I_{18}	13	Bus voltage magnitude
I_{19} to I_{55}	37	Reactive power for line flows
O_1 to O_{20}	20	Reactive power transfer between generators and loads

Table 2. Description of inputs and outputs of the training data for each ANN for reactive power

3.1.2 Training

Neural networks are sensitive to the number of neurons in their hidden layer. Too few neurons in the hidden layer prevent it from correctly mapping inputs to outputs, while too many may impede generalization and increasing training time. Therefore number of hidden neurons is selected through experimentation to find the optimum number of neurons for a predefined minimum of mean square error in each training process. To take into account the nonlinear characteristic of input (D) and noting that the target values are either positive or negative, the suitable transfer function to be used in the hidden layer is a tan-sigmoid function. Non linear activation functions allow the network to learn nonlinear relationships between input and output vectors. Levenberg-Marquardt algorithm has been used for training the network. After the input and target for training data is created, next step is to divide the data (D and T) up into training, validation and test subsets. In this case 100 samples (60%) of data are used for the training and 34 samples (20%) of each data for validation and testing. Table 3 shows the numbers of samples for training, validation and test data for real and reactive power allocation respectively.

Data Types	Number of Samples (Hour)
Training	100
Validation	34
Testing	34

Table 3. The number of samples for training, validation and test set

The error on the training set is driven to a very small value i.e. 3.5×10^{-8} . If the calculated output error becomes much larger than acceptable, when a new data is presented to the trained network, then it can be said that the network has memorized the training samples, but it has not learned to generalize to new situations. Validation sets is used to avoid this overfitting problem. The test set provides an independent measure of how well the network can perform on data not used to train it. In real power allocation scheme, the performance of the training for the ANN with two hidden layers having different number of neurons i.e. 15 and 10 respectively is as shown in Figure 3. From Figure 3, it can also be seen that the training goal is achieved in 12 epochs with a mean square error of 8.897×10^{-9}. For reactive power allocation scheme, the performance of the training for the ANN is also made with two hidden layers having different number of neurons i.e. 10 and 15 respectively as shown in Figure 4.
In this Figure 4 the training goal is achieved in 13 epochs with a mean square error of 9.50128×10^{-9}. Note that the mean square error is not much different for both real as well as reactive power transfer allocation. This indicates that the developed ANN can allocate both real as well as reactive power transfer between generators and loads with almost similar accuracy.

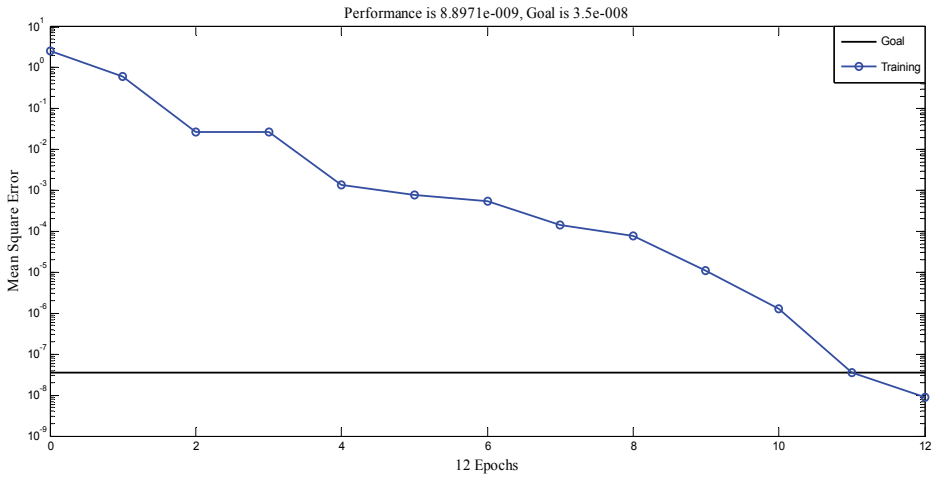


Fig. 3. Training curve with two hidden layers having different number of neurons i.e. 15 and 10 respectively for real power allocations

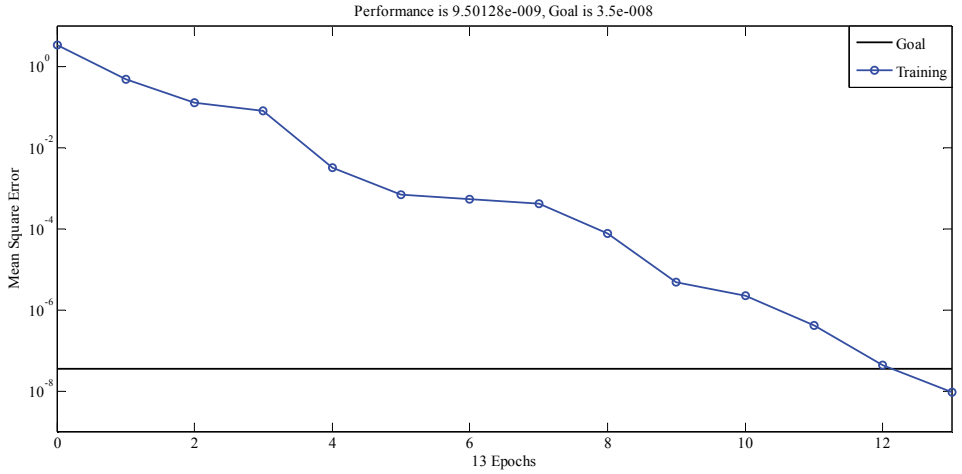


Fig. 4. Training curve with two hidden layers having different number of neurons i.e. 10 and 15 respectively for reactive power allocations

The result is reasonable, since the test set error and the validation set error have similar characteristics with the training set, and it doesn't appear that any significant overfitting has occurred. The same network setting parameters is used for training the other 2 networks.

3.1.3 Pre-testing and simulation

After the networks have been trained, next step is to simulate the network. The entire sample data is used in pre testing. After simulation, the obtained result from the trained network is evaluated with a linear regression analysis. In real power allocation scheme, the regression analysis for the trained network that referred to contribution of generator at bus 15 to load at bus 1 is shown in Figure 5.

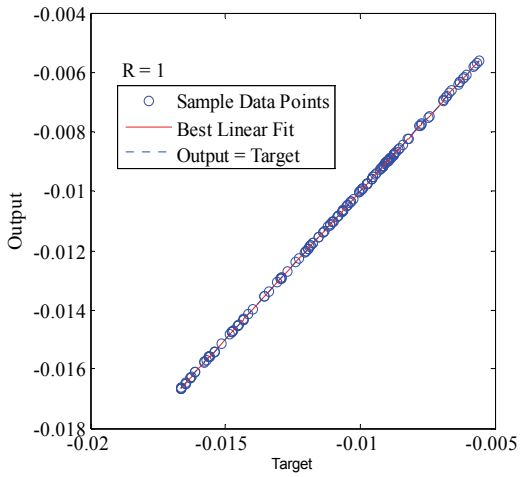


Fig. 5. Regression analysis between the network output and the corresponding target for real power allocation

The correlation coefficient, (R) in this case is equal to one which indicates perfect correlation between MNE Method and output of the neural network. The best linear fit is indicated by a solid line whereas the perfect fit is indicated by the dashed line. Subsequently, similar results is obtained on regression analysis for reactive power allocation method for the trained network that referred to contribution of generator at bus 14 to load at bus 2 as shown in Figure 6.

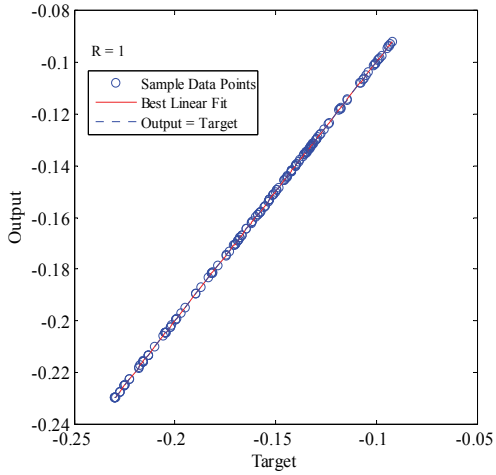


Fig. 6. Regression analysis between the network output and the corresponding target for reactive power allocation

Finally, both real as well as reactive power contribution to loads is determined and compared with the MNE Method's output. Daily load curves for every load bus are shown in Figures 7 to 8 and the target patterns for generator located at buses 14 and 22 are given in Figures 9 to 12.

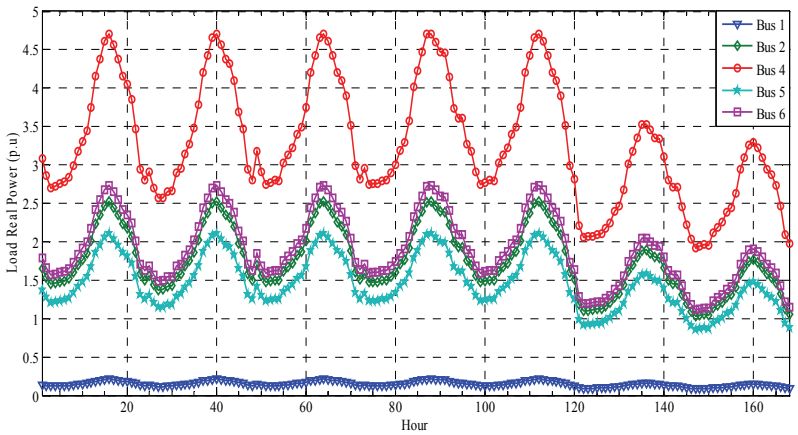


Fig. 7. Real power allocation method daily load curves for different buses

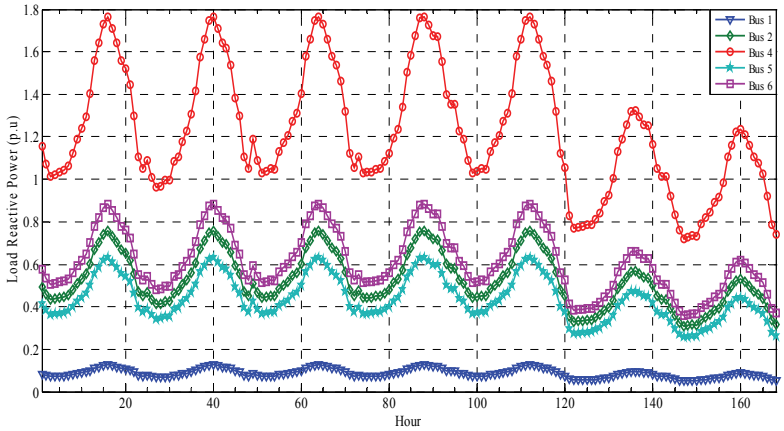


Fig. 8. Reactive power allocation method daily load curves for different buses

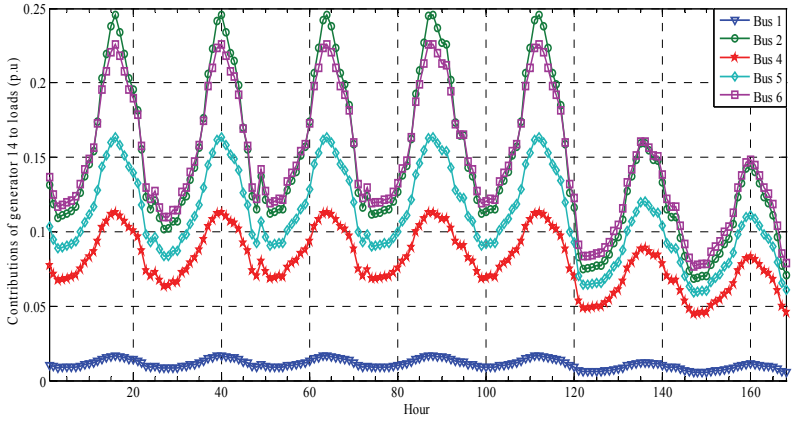


Fig. 9. Selected target patterns of generator at bus 14 of real power allocation scheme within 168 hours

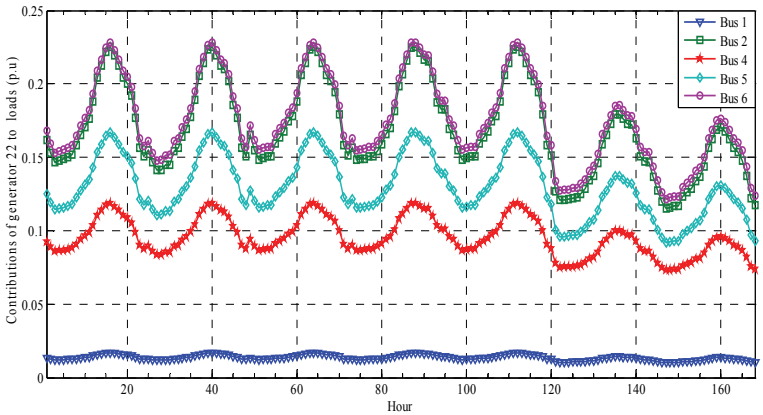


Fig. 10. Selected target patterns of generator at bus 22 of real power allocation scheme within 168 hours

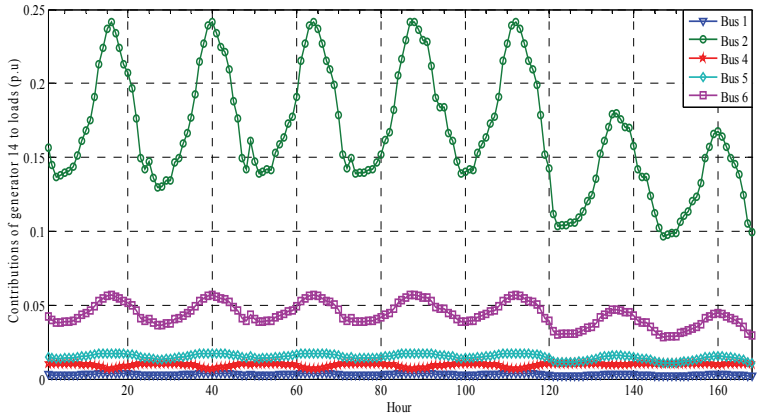


Fig. 11. Selected target patterns of generator at bus 14 of reactive power allocation scheme within 168 hours

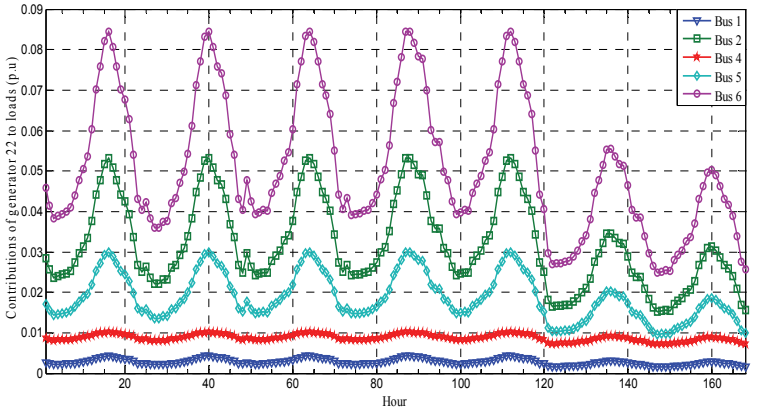


Fig. 12. Selected target patterns of generator at bus 22 of reactive power allocation scheme within 168 hours

4. Real power allocation results for 25-bus test system

At different loads, comparison of results of (Bialek, 1996) Method with the proposed method is as shown in Table 4. It is observed that, the results of proposed method is very much comparable with (Bialek, 1996) Method. Due to the different approach the difference of allocation factor lies between the results of proposed method and (Bialek, 1996) Method occurred at each load buses 1, 2, and 4 to 6. This difference does not exist i.e. zero contribution in the (Bialek, 1996). Method for about half count buses while the proposed method distribute allocation factor to all load buses. The other difference of the proposed method is due to the use of basic system nodal equations which minimize the simplifying assumptions such as the proportional sharing and lossless network as considered in Bialek's Method. From Table 4, it can also be observed that the sum of the real power contributed by each generator is in conformity with the solved load flow. In this system, (Bialek, 1996) Method and the proposed method can compute the required relationship with similar computation time i.e. within 46 msec. Hence, it is proven that the proposed methodology provides reasonable and acceptable results to real power transfer allocation as compared to (Bialek, 1996) Method.

Supplied by	**Load bus no.**									
	Modified Nodal Equations Method					**Bialek's Method**				
(MW)	**1**	**2**	**4**	**5**	**6**	**1**	**2**	**4**	**5**	**6**
Gen-14	1.150	15.041	8.519	11.475	15.318	0	71.274	0	0	0
Gen-15	1.150	15.041	8.519	11.475	15.318	0	71.274	0	0	0
Gen-16	1.489	16.741	96.602	14.772	18.816	0	0	85.144	0	0
Gen-17	1.456	16.257	93.268	14.388	18.307	0	0	82.090	0	0
Gen-18	0.93393	10.786	7.210	9.402	12.027	2.181	0	16.593	21.805	13.444
Gen-19	1.064	11.538	64.478	10.35	13.108	0	0	56.392	0	0
Gen-20	0.97752	11.451	7.619	9.919	12.717	2.353	0	17.903	23.527	14.505
Gen-21	1.343	17.026	9.602	13.087	17.626	0	19.446	0	0	51.670
Gen-22	1.376	17.389	9.759	13.337	17.997	0	19.446	0	0	51.670
Gen-23	1.376	16.756	11.011	14.275	18.408	3.586	0	27.292	35.863	22.111
Gen-24	1.248	14.774	9.796	12.739	16.358	3.070	0	23.362	30.699	18.927
Gen-25	1.554	18.643	12.308	15.982	20.564	3.931	0	29.912	39.306	24.234
Total Load	15.120	181.443	338.691	151.202	196.564	15.121	181.440	338.688	151.200	196.561
Actual Load	15.12	181.44	338.69	151.2	196.56	15.12	181.44	338.69	151.2	196.56

Table 4. Comparison of the real power distribution by each generator to load at buses 1, 2, 4 to 6 for the practical 25-bus equivalent power system

The proposed MNE Method has been simulated to reveal the accuracy of the developed ANN. The case scenario is that the real and reactive load is decreasing in 10% from the nominal trained pattern. Furthermore, it is also assumed that all generation is divided proportionally according to the load demands, to ensure that all real power generation of generator at buses 14 to 25 varies in respond to the daily load pattern of the loads at least by a small amount rather than to give the unbalance load only to the slack generator. Figure 13 shows the real power transfer allocation results due to generator located at bus 14 by the ANN output along with the result obtained through to proposed method for loads at buses 1, 2, and 4 to 6 within 168 hours.

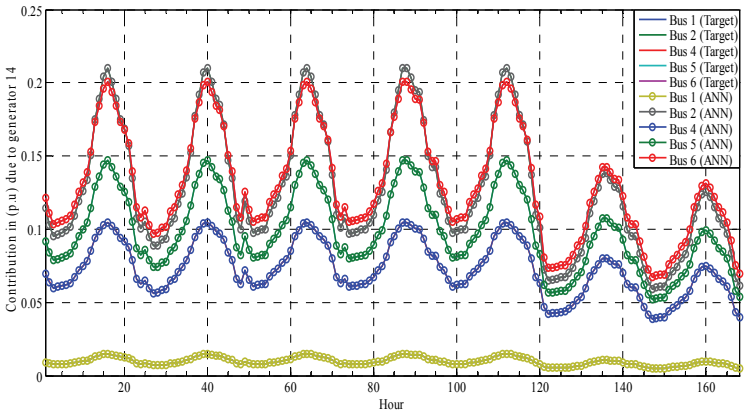


Fig. 13. Distribution of real power from generator at bus 14 to loads within 168 hours

Results obtained from the ANN output are indicated with lines having circles and the solid lines represent the output of the MNE Method. From Figure 13, it can be observed that the developed ANN can allocate real power transfer between generators and load with very good accuracy, almost 100 %. In this simulation, ANN computes within 45 msec whereas the MNE Method takes 1314 msec for the same real power transfer allocation. Consequently, it can be concluded that the ANN is more efficient in terms of computation time. Moreover, the final allocation of real power to loads on hours twelve out of 168 hours using developed ANN is presented in Table 5 along with the result obtained through MNE Method. Note that the result obtained by the ANN output is comparable with the result of MNE Method. The difference of real power between generators in both methods is very small which is less than 0.0053 MW.

Supplied by	**Load bus no.**									
	ANN Output					**Modified Nodal Equations Method**				
(MW)	**1**	**2**	**4**	**5**	**6**	**1**	**2**	**4**	**5**	**6**
Gen-14	1.150	15.042	8.519	11.476	15.319	1.150	15.041	8.519	11.475	15.318
Gen-15	1.150	15.043	8.519	11.477	15.32	1.150	15.041	8.519	11.475	15.318
Gen-16	1.489	16.744	96.603	14.773	18.816	1.489	16.741	96.602	14.772	18.816
Gen-17	1.456	16.258	93.273	14.388	18.308	1.456	16.257	93.268	14.388	18.307
Gen-18	0.93393	10.786	7.210	9.402	12.027	0.93393	10.786	7.210	9.402	12.027
Gen-19	1.064	11.538	64.477	10.35	13.108	1.064	11.538	64.478	10.35	13.108
Gen-20	0.97752	11.451	7.619	9.919	12.717	0.97752	11.451	7.619	9.919	12.717
Gen-21	1.343	17.026	9.602	13.087	17.626	1.343	17.026	9.602	13.087	17.626
Gen-22	1.375	17.389	9.759	13.336	17.996	1.376	17.389	9.759	13.337	17.997
Gen-23	1.376	16.755	11.01	14.275	18.407	1.376	16.756	11.011	14.275	18.408
Gen-24	1.248	14.773	9.795	12.739	16.357	1.248	14.774	9.796	12.739	16.358
Gen-25	1.553	18.642	12.307	15.981	20.563	1.554	18.643	12.308	15.982	20.564
Total Load	15.120	181.446	338.697	151.202	196.564	15.120	181.443	338.691	151.202	196.564
Actual Load	15.12	181.44	338.69	151.2	196.56	15.12	181.44	338.69	151.2	196.56

Table 5. Analysis of real power allocation for the practical 25-bus equivalent power system of south Malaysia

5. Reactive power allocation results for 25-bus test system

Table 6 shows a comparison of reactive power distribution of generators at buses 14 to 25 obtained through the Chu's Method (Chu & Liao,2004) and proposed MNE Method. By comparing the values depicted in Table 6, it is obvious that the reactive power allocation made by the proposed method is slightly difference from that of Chu's Method. The difference in the result between both methods is only noticeable for load at bus 4 while the results of others load buses are almost similar. This may due to the concept applied by the proposed method which represents each load current as a function of individual generators' current and voltage. On the other hand the Chu's Method represents each load voltage as a function of individual generators' voltage.

Supplied by	**Load bus no.**									
	Modified Nodal Equations Method					**Chu's Method**				
(MVAr)	**1**	**2**	**4**	**5**	**6**	**1**	**2**	**4**	**5**	**6**
Gen-14	0.31492	17.18	0.96389	1.5687	4.5436	0.31492	17.18	2.5279	1.5687	4.5436
Gen-15	0.31492	17.18	0.96389	1.5687	4.5436	0.31492	17.18	2.5279	1.5687	4.5436
Gen-16	0.74182	1.2167	36.688	3.4787	4.0287	0.74182	1.2167	23.467	3.4787	4.0287
Gen-17	0.73775	1.2058	36.835	3.4491	3.9978	0.73775	1.2058	23.325	3.4491	3.9978
Gen-18	0.97819	1.6864	3.2764	4.7926	5.484	0.97819	1.6864	8.6761	4.7926	5.484
Gen-19	0.57913	0.93221	30.051	2.6715	3.1082	0.57913	0.93221	18.266	2.6715	3.1082
Gen-20	0.99247	1.7194	3.3289	4.8834	5.5814	0.99247	1.7194	8.8266	4.8834	5.5814
Gen-21	0.28846	3.2488	0.89633	1.9222	5.2149	0.28846	3.2488	2.4623	1.9222	5.2149
Gen-22	0.28846	3.2488	0.89633	1.9222	5.2149	0.28846	3.2488	2.4623	1.9222	5.2149
Gen-23	1.2757	2.2432	4.2971	6.3601	7.2436	1.2757	2.2432	11.44	6.3601	7.2436
Gen-24	1.248	2.1686	4.1895	6.1571	7.0321	1.248	2.1686	11.118	6.1571	7.0321
Gen-25	1.2941	2.2928	4.3687	6.4951	7.3842	1.2941	2.2928	11.655	6.4951	7.3842
Total Load	9.05392	54.3227	126.755	45.2694	63.377	9.05392	54.32271	126.7541	45.2694	63.377
Actual Load	9.0539	54.323	126.75	45.269	63.377	9.0539	54.323	126.75	45.269	63.377

Table 6. Reactive power distribution of generators to loads for the 25-bus equivalent system

A number of simulations have been carried out to demonstrate the accuracy of the developed ANN with the same 25-bus equivalent system of south Malaysia. The scenario is a decrement by 10% of the real and reactive load demand from the nominal trained pattern. Besides it also assumed that all generators also decrease their production proportionally according to this variation in the load demands. Figure 14 shows the reactive power transfer allocation result for generator located at bus 14 calculated by the ANN along with the result obtained through MNE Method for loads at buses 1, 2, and 4 to 6 within 168 hours.

The pattern used for results is same as of real power allocation. From Figure 14, it can be observed that the developed ANN can allocate reactive power transfer between generators and load with very good accuracy, almost 100%. In this simulation, ANN computes within 45 msec whereas the MNE Method took 908 msec for the calculation of same reactive power transfer allocation. Therefore it can be concluded that the ANN is more efficient in terms of computation time. From Table 7, it can be noted that the result obtained by the ANN output in this thesis is compared well with the result of MNE Method.

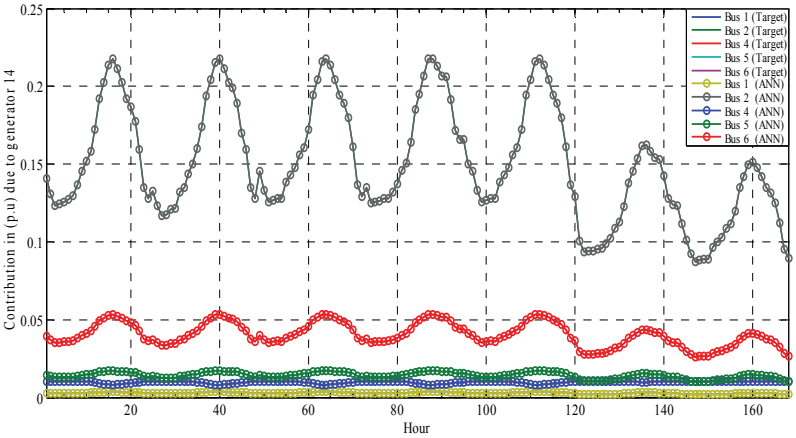


Fig. 14. Distribution of reactive power from generator at bus 14 to loads within 168 hours

Supplied by	Load bus no.									
	ANN Output					Modified Nodal Equations Method				
(MVAr)	1	2	4	5	6	1	2	4	5	6
Gen-14	0.31492	17.18	0.96386	1.5686	4.5436	0.31492	17.18	0.96389	1.5687	4.5436
Gen-15	0.31492	17.18	0.96386	1.5687	4.5435	0.31492	17.18	0.96389	1.5687	4.5436
Gen-16	0.74181	1.2167	36.689	3.4786	4.0286	0.74182	1.2167	36.688	3.4787	4.0287
Gen-17	0.73775	1.2058	36.835	3.449	3.9978	0.73775	1.2058	36.835	3.4491	3.9978
Gen-18	0.97821	1.6865	3.2764	4.7927	5.4841	0.97819	1.6864	3.2764	4.7926	5.484
Gen-19	0.57914	0.9322	30.05	2.6715	3.1082	0.57913	0.93221	30.051	2.6715	3.1082
Gen-20	0.99249	1.7194	3.3288	4.8834	5.5815	0.99247	1.7194	3.3289	4.8834	5.5814
Gen-21	0.28846	3.2489	0.89634	1.9223	5.2152	0.28846	3.2488	0.89633	1.9222	5.2149
Gen-22	0.28845	3.2487	0.89632	1.9222	5.2147	0.28846	3.2488	0.89633	1.9222	5.2149
Gen-23	1.2756	2.2431	4.2971	6.3599	7.2433	1.2757	2.2432	4.2971	6.3601	7.2436
Gen-24	1.2479	2.1685	4.1894	6.1569	7.0319	1.248	2.1686	4.1895	6.1571	7.0321
Gen-25	1.2941	2.2928	4.3687	6.4949	7.3839	1.2941	2.2928	4.3687	6.4951	7.3842
Total Load	9.05375	54.3226	126.755	45.2687	63.3763	9.05392	54.32271	126.755	45.2694	63.377
Actual Load	9.0539	54.323	126.75	45.269	63.377	9.0539	54.323	126.75	45.269	63.377

Table 7. Analysis of reactive power allocation for the 25-bus equivalent system

The difference of reactive power between generators in both methods is very small, which are less than 10^{-3} MVAr. The consumer located at bus 4 consumed the highest demand compared to other consumers in this hour. Consequently, the contribution of reactive power due to generators 16, 17 and 19 located at the same bus provides more reactive power to load at bus 4 by both methods as well. For this reason the acquired result illustrates that the contribution of individual generators are mostly confined in their neighborhood.

6. Test conducted on the IEEE 118 bus system

The proposed methods have also been tested on IEEE 118 bus system. This system consists of 186 lines, 33 physical reactive power sources and 54 real power generators.

6.1.1 Application of RBFN in real and reactive power allocation method

One of the main purposes of this work is to incorporate RBFN into real and reactive power allocation method between generators and load. The structure of the proposed RBFN for each allocation scheme is discussed in the following sub-sections.

6.1.2 Real power allocation method

In this case study, RBFN with one hidden layer and one output layer has been chosen. The proposed allocation method is elaborated by designing an appropriate RBFN for the IEEE 118 bus system as shown in Figure 15. This system consists of 54 generators located at selected buses which lies in between buses numbered as 1 to 118. They deliver power to 64 loads, through 186 branches located at selected buses which lies in between buses numbered as 1 to 118.

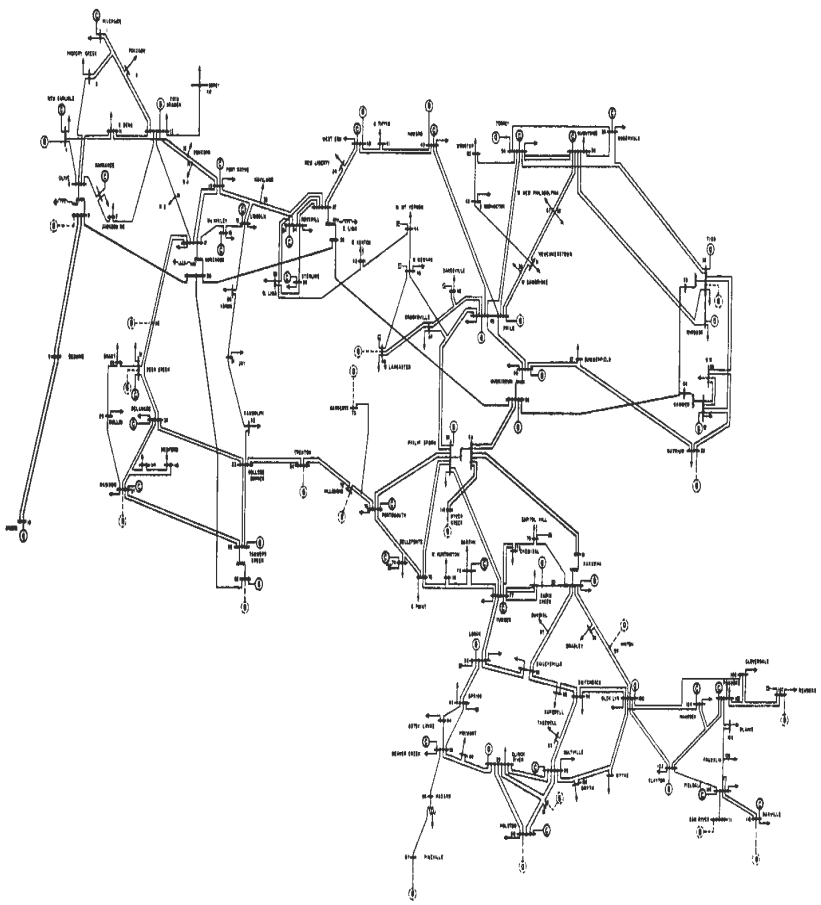


Fig. 15. Single line diagram for the IEEE 118 bus system

The input samples for training is assembled using the daily load curve and performing load flow analysis for every hour of load demand. Again the load profile on hourly basis (Cheng, 1998) is utilized to produce 24 hours loads here also. Similarly the target vector for the training is obtained from the proposed method using nodal equations. Input data (D) for

developed ANN contains independent variables such as real power generation located at selected buses which lies in between buses numbered as *(P_{g1}, to P_{g118})*, real loads located at selected buses which lies in between buses numbered as *(P_2 to P_{118})*, reactive loads located at selected buses which lies in *(Q_2 to Q_{118})*, average power for line flows *(P_{line1} to $P_{line186}$)* and the target/output parameter, (T) which is real power transfer between generators and loads placed at selected buses which lies in between buses numbered as 2 to 118. This is considered as 3456 outputs and therefore the networks have three thousand, four hundred and fifty six output neurons. Each generator allocates to the sixty four output neurons which correspond to the loads located at selected buses which lies in between buses numbered as 2 to 118. For example, the first sixty four neurons (1-64) represent the contribution from generator at bus 1 to the sixty four loads, the second sixty four neurons (65-128) represent the contribution from generator at bus 4 to the sixty four loads and so on for generators located at selected buses which lies in between buses numbered as 1 to 118. Table 8 summarizes the description of inputs and outputs of the training data for the RBFN.

Input and Output (layer)	Neurons	Description (in p.u)
I_1 to I_{54}	54	Real power generations
I_{55} to I_{118}	64	Real loads
I_{119} to I_{182}	64	Reactive loads
I_{183} to I_{368}	186	Average power for line flows
O_1 to O_{3456}	3456	Real power transfer between gen. and loads

Table 8. Description of inputs and outputs of the training data for the RBFN

6.1.3 Reactive power allocation scheme

In this case study, structure and description of input and output of each RBFN is similar to those of the real power allocation scheme. Table 9 shows the details of inputs and outputs of the training data for the RBFN.

Input and Output (layer)	Neurons	Description (in p.u)
I_1 to I_{54}	54	Real power generations
I_{55} to I_{118}	64	Real loads
I_{119} to I_{182}	64	Reactive loads
I_{183} to I_{368}	186	Average power for line flows
O_1 to O_{3456}	3456	Reactive power transfer between gen. and loads

Table 9. Description of inputs and outputs of the training data for the RBFN

6.1.4 Unsupervised learning to choose the centers of training samples

The well-known *k*-means clustering algorithm is used to find a set of centers for the training samples. In *k*-means clustering, the number of desired centers (*k*), must be decided in advance. One simple way of choosing the value of *k* is to set it equal to a fraction of total training data samples. The *k*-means algorithm is as follows (Abdullah, 2008):

Step 1: Assign the input data to random k sets.
Step 2: Compute the mean of each set.
Step 3: Reassign each point to a new set according to which is the nearest mean vector.
Step 4: Recomputed the mean of each set.

Step 5: Repeat steps 3 and 4 until there is no further change in the grouping of data points.
Step 6: The mean of the sets will be the RBFN center.

6.1.5 Training

After the input and target for training data is created, it can be made more efficient to scale (preprocess) the network inputs and targets so that they always fall within a specified range. In this case the minimum and maximum value of input and output vectors is used to scale them in the range of -1 and +1. Next step is to divide the input data and target data up into training. In this case 14 samples (60%) of data are used for the training as shown in Table 10.

Data Types	Samples (Hour)
Training	1,6,11,16,21,3,8,13,18,23,5,10,15,20

Table 10. The Numbers of Samples for Training

The training of the RBFN consists of two separate stages. First step is to find the centers parameter by using the k-means clustering algorithm. Initially, the number of trials with different number of k keeping the β constant and vice versa is set. In both real and reactive power allocation scheme, the k is taken as 14 samples equal to number of hours and the β as 10, resulting in reasonable accuracy of the output of the RBFN with the target. For this k=14 and β =10, the computed training time i.e. 187 msec taken by the RBFN is same for both of the real and reactive power allocation scheme. Total number of the second layer weights influencing the individual output is, 14. Therefore, the minimum number of data set required to train the network is 14. In the second training stage, the second layer weights in connections between the hidden layer and the output layer are determined using the least squares based on minimization of quadratic errors of RBFN network output values over the set of training input-output vector pairs. At that stage, the weights in connections between the input layer and the hidden layer and the parameters of the radial basis functions of the hidden layer are already set as determined in the first training stage and are not subject to any further changes. During this training, the RBFN network is presented with individual input vectors from the set of training samples and responds with certain output vectors. These output vectors are compared with the target output vectors also given in the training set, and the individual weights are updated in a way ensuring a decrease of the difference between the actual and target output vectors. The individual input-output training pairs are presented to the RBFN network repeatedly until the error decreases to an acceptable level.

6.1.6 Pre-testing and simulation

In first step using MATLAB, the network is to be trained. In the second step involves simulating the network. The entire sample data is used in pre testing. After simulation, the obtained result from the trained network is evaluated with a linear regression analysis. The regression analysis for the trained network that referred to contribution of generator at bus 1 to load at bus 2 is shown in Figure 16.
The correlation coefficient, (R) in this case is equal to one which indicates perfect correlation between MNE Method and output of the neural network. The best linear fit is indicated by a solid line whereas the perfect fit is indicated by the dashed line. Moreover, performing regression analysis of reactive power allocation scheme for the trained network, similar results is obtained which refers to contribution of generator at bus 1 to load at bus 16 as shown in Figure 17.

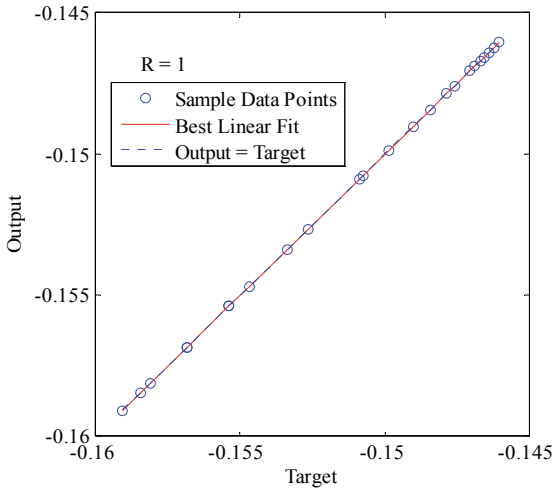


Fig. 16. Regression analysis between the network output and the corresponding target keeping k =14 and β =10 for real power allocation

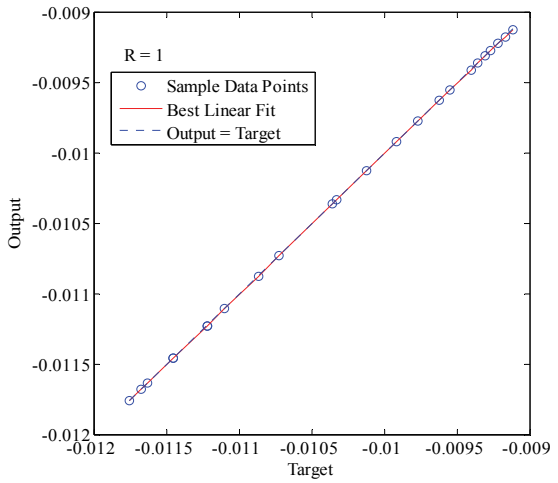


Fig. 17. Regression analysis between the network output and the corresponding target keeping k =14 and β =10 for reactive power allocation

6.1.7 Real power allocation results for IEEE 118 bus system

The case scenario is that increment by 10% of the real and reactive load demand from the nominal trained pattern. In addition it is also assumed that all generation is divided linearly according to the load demands. Figure 18 shows the real power transfer allocation result for generator located at bus 69 calculated by the RBFN along with the result obtained through to the MNE Method for loads at buses 41, 43, 44, 45, 47, 48, 53, 57, 58 and 79 within 24 hours. Results obtained from the RBFN are indicated with lines having circles, and the solid lines represent the output of the MNE Method. From Figure 18, it can be observed that the developed RBFN can allocate real power transfer between generators and load with very

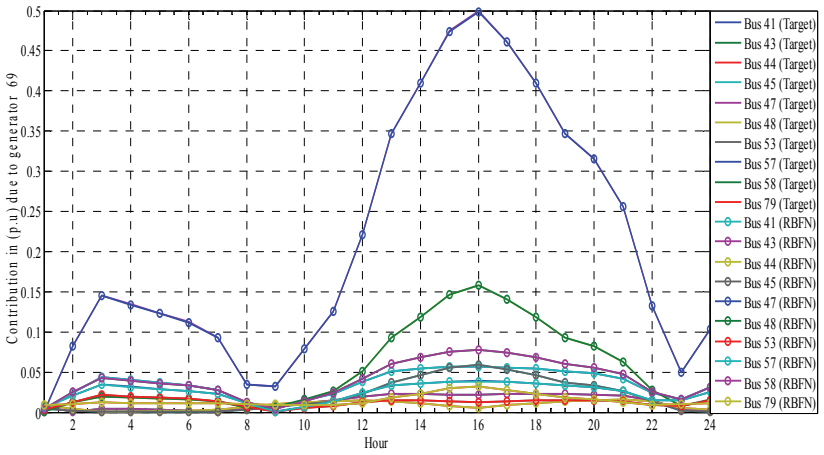


Fig. 18. Distribution of real power from generator at bus 69 to loads within 24 hours

good accuracy, almost 100%. In this simulation, RBFN computes within 15 ms, whereas the MNE Method took 3000 ms for the calculation of same real power transfer allocation. For that reason it can be concluded that the RBFN is more efficient in terms of computation time. Moreover, the final allocation of real power to loads using proposed RBFN on hours 12 out of 24 hours is presented in Table 11 along with the result obtained through MNE Method. It can be noted that the result obtained by the proposed RBFN compares well with the result of MNE Method. The difference of real power between generators in both methods is too small i.e. less than 7.687×10^{-4}MW. It is worth noting that the total contributions of each generator to loads are reasonable since it is less than its total production. For example, the total contribution of generator at bus 107 to all loads is 56.609 MW and this value does not exceed its generation i.e. 60MW.

Bus	Actual	RBFN Output						Modified Nodal Equations Method					
no.	load	Gen-107	Gen-110	Gen-111	Gen-112	Gen-113	Gen-116	Gen-107	Gen-110	Gen-111	Gen-112	Gen-113	Gen-116
	(MW)	(MW)	(MW)	(MW)	(MW)	(MW)	(MW)	(MW)	(MW)	(MW)	(MW)	(MW)	(MW)
2	33.742	0.17641	0.07029	0.074479	0.084484	0.26015	0.54608	0.17642	0.070304	0.074488	0.084469	0.26018	0.54609
3	34.936	0.0011481	0.02429	-0.00184	-0.00487	-0.12908	0.08572	0.0011802	0.024295	-0.00184	-0.00487	-0.12913	0.08577
7	20.532	-0.50109	-0.08008	-0.22335	-0.26714	-1.397	-1.139	-0.50109	-0.08014	-0.22334	-0.26723	-1.397	-1.139
11	22.044	-0.079927	-0.01767	-0.03514	-0.04149	-0.19599	-0.19864	-0.079927	-0.01767	-0.03514	-0.04150	-0.19607	-0.19855
13	21.964	0.3324	0.16024	0.13762	0.15281	0.33545	1.124	0.33242	0.16026	0.13762	0.15282	0.33547	1.124
14	20.691	0.16734	0.085524	0.068802	0.075821	0.1413	0.58286	0.16735	0.085535	0.068803	0.075822	0.14129	0.58291
16	20.85	0.27669	0.12888	0.115	0.12823	0.30286	0.92001	0.2767	0.12889	0.115	0.12823	0.30278	0.92001
17	21.089	-1.328	-0.44117	-0.5697	-0.65629	32.756	-3.794	-1.328	-0.44119	-0.5698	-0.65648	32.755	-3.795
20	21.168	0.20442	0.13588	0.080957	0.085462	0.015533	0.815	0.20441	0.13587	0.080954	0.085448	0.015482	0.81502
21	20.85	0.41409	0.21672	0.16975	0.18646	0.34599	1.435	0.41409	0.21673	0.16975	0.18646	0.34593	1.435
22	21.487	0.42325	0.22155	0.1735	0.19058	0.36393	1.456	0.42326	0.22155	0.17351	0.19058	0.3639	1.456
23	22.839	-0.66814	-0.25275	-0.28345	-0.323	0.080557	-1.915	-0.66823	-0.25273	-0.28347	-0.32303	0.080539	-1.916
28	18.463	0.13753	0.078981	0.055688	0.060326	0.46697	0.50158	0.13752	0.078984	0.055685	0.060323	0.46693	0.50158
29	35.015	0.08196	0.056481	0.032272	0.033801	0.71278	0.3329	0.081951	0.056484	0.032257	0.033798	0.7128	0.33289
33	42.177	0.59022	0.2685	0.24595	0.275	0.66717	1.956	0.59024	0.26853	0.24594	0.275	0.66711	1.956
35	26.261	-0.22153	-0.11404	-0.09094	-0.1002	-0.19487	-0.78481	-0.22151	-0.11404	-0.09099	-0.10018	-0.19497	-0.78503
39	29.445	0.15404	0.18401	0.052954	0.045775	-0.16485	0.90055	0.15401	0.18401	0.05296	0.045736	-0.16498	0.90052

Table 11. Analysis of real power allocation for selected generators in the IEEE 118 bus system

41	29.445	-0.38925	0.034202	-0.18297	-0.22959	-0.89799	-0.58842	-0.38924	0.034191	-0.18297	-0.2296	-0.89814	-0.58843
43	30.24	0.50245	0.24314	0.20791	0.23076	0.47352	1.729	0.50243	0.24313	0.20792	0.23077	0.4735	1.729
44	28.649	0.57406	0.30863	0.23452	0.25662	0.45607	2.082	0.57406	0.30864	0.23453	0.25663	0.45605	2.082
45	42.177	0.28215	0.1641	0.11407	0.12332	0.20542	1.066	0.28215	0.16412	0.11406	0.12331	0.20536	1.066
47	58.889	0.23569	0.15323	0.093944	0.099664	0.152	0.93871	0.23577	0.15319	0.093967	0.099666	0.15199	0.93889
48	47.748	0.055997	0.023694	0.023554	0.026538	0.048013	0.18169	0.056017	0.023724	0.023556	0.026553	0.04797	0.18162
50	29.445	0.21457	0.12709	0.086539	0.093283	0.1485	0.82321	0.21457	0.12708	0.086535	0.093279	0.14849	0.82322
51	29.445	0.40261	0.22604	0.16355	0.17781	0.28905	1.506	0.40262	0.22604	0.16356	0.17781	0.28903	1.506
52	30.24	0.63096	0.34321	0.25738	0.28115	0.46243	2.325	0.63096	0.34321	0.25738	0.28115	0.46242	2.325
53	26.261	-0.10263	0.055608	-0.05283	-0.07115	-0.17384	-0.00468	-0.10269	0.055604	-0.05285	-0.07121	-0.17391	-0.00487
57	33.424	-0.08542	0.13265	-0.05248	-0.07895	-0.22284	0.28771	-0.085465	0.13258	-0.05248	-0.07897	-0.22301	0.28766
58	19.895	-0.31289	0.063793	-0.15073	-0.19292	-0.4379	-0.36759	-0.31304	0.063805	-0.15073	-0.19294	-0.43799	-0.36768
60	62.072	0.012021	0.17156	-0.01134	-0.03242	-0.12428	0.58579	0.011923	0.17159	-0.01138	-0.03239	-0.12441	0.58567
67	22.282	0.13199	0.10379	0.050696	0.051522	0.068946	0.59666	0.13198	0.10379	0.050696	0.051512	0.068944	0.59664
75	37.403	0.18798	-0.02875	0.092096	0.11654	0.17113	0.048409	0.18811	-0.02816	0.091987	0.1166	0.17089	0.049178
78	56.502	0.54436	0.46	0.23461	0.24055	0.25415	1.780	0.54457	0.46035	0.23476	0.2407	0.25441	1.780
79	31.036	0.37995	0.40042	0.15918	0.15374	0.17367	1.528	0.37987	0.40043	0.15914	0.15373	0.17364	1.528
82	42.973	0.8622	0.77475	0.32579	0.31954	0.41579	2.577	0.86217	0.77462	0.3258	0.31954	0.41581	2.577
83	15.916	0.31408	0.27243	0.11957	0.11858	0.14553	0.86389	0.31408	0.27244	0.11958	0.11856	0.14552	0.86388
84	87.538	0.018614	-0.17347	0.029182	0.054584	-0.03262	-0.56264	0.018548	-0.17346	0.029177	0.054564	-0.03262	-0.56274
86	16.712	-0.1309	-0.23078	-0.03578	-0.01950	-0.08321	-0.6854	-0.13092	-0.2308	-0.03580	-0.01955	-0.08323	-0.6857
88	38.198	0.87967	-0.05461	0.43983	0.54895	0.14802	-0.5368	0.87941	-0.05460	0.43979	0.54891	0.14805	-0.53752
93	95.496	-0.25741	-0.11605	-0.07175	-0.07622	-0.10564	-0.3485	-0.25755	-0.11608	-0.07176	-0.07621	-0.10566	-0.34858
94	23.874	1.668	0.63315	1.507	1.794	-0.19487	-1.373	1.6685	0.63318	1.507	1.794	-0.19489	-1.374
95	33.424	0.6563	0.80719	0.22336	0.18961	0.34821	2.309	0.65622	0.80733	0.22336	0.18955	0.34821	2.31
96	30.24	0.4655	0.60166	0.1647	0.13912	0.25282	1.886	0.46548	0.60158	0.16469	0.13909	0.25282	1.886
97	11.937	-0.19377	0.23982	-0.09554	-0.1436	-0.05865	0.77291	-0.1938	0.23974	-0.09558	-0.14365	-0.05867	0.77298
98	27.057	0.8291	0.49602	0.59536	0.68224	0.08842	0.61209	0.82889	0.49604	0.59535	0.68216	0.088447	0.6121
101	97.088	3.857	2.371	1.911	2.107	0.9018	3.569	3.857	2.371	1.911	2.107	0.90182	3.569
102	11.937	0.007438	-0.34445	0.09218	0.15069	-0.13734	-1.125	0.007336	-0.34441	0.092196	0.15065	-0.13734	-1.125
106	34.219	42.439	-0.57113	1.485	1.895	-0.54923	-2.923	42.438	-0.57107	1.485	1.895	-0.5493	-2.923
108	89.13	12.297	3.114	2.377	2.603	0.57067	1.915	12.296	3.114	2.377	2.602	0.57067	1.915
109	14.324	-1.378	12.968	17.664	20.427	-0.53203	-2.062	-1.378	12.968	17.664	20.427	-0.53203	-2.062
114	14.324	0.050762	0.069213	0.01661	0.013121	1.568	0.32816	0.050775	0.069204	0.016619	0.013128	1.567	0.32813
115	17.508	0.081997	0.050169	0.0329	0.035261	0.48282	0.30947	0.081973	0.050169	0.032889	0.035252	0.48282	0.30945
117	15.916	0.26975	0.14147	0.11056	0.1214	0.2092	0.95213	0.26977	0.14147	0.11056	0.12141	0.20917	0.95212
118	26.261	0.20535	0.22663	0.074181	0.067384	0.093782	1.021	0.20527	0.22663	0.074156	0.067356	0.093762	1.021
Total:		56.609	18.605	24.406	28.148	29.977	26.348	56.609	18.607	24.405	28.148	29.977	26.354

Table 11. Analysis of real power allocation for selected generators in the IEEE 118 bus system (cont.)

6.1.8 Reactive power allocation results for IEEE 118 bus system

For case scenario, the real and reactive load demand from the nominal trained pattern is increased by 10%. Figure 19 shows the reactive power transfer allocation result for generator located at bus 69 calculated by the RBFN along with the result obtained through to the MNE Method for loads at buses 2, 3, 11, 13, 14, 16, 17, 20, 21 and 22 within 24 hours. The pattern used for results is same as of real power allocation. From Figure 6.7, it can be observed that the developed RBFN can allocate reactive power transfer between generators and load with very good accuracy, almost 100%. In this simulation, RBFN computes within 15ms, whereas the MNE Method took 2911ms for the calculation of same reactive power transfer allocation. As a result it can be concluded that the RBFN is more efficient in terms of computation time.

Furthermore, the final allocation of reactive power to loads at hour 12 using developed RBFN is presented in Table 12 along with the result obtained through MNE and found close match between their results. The difference of reactive power between generators in both methods is very small i.e. <0.0067Mvar.

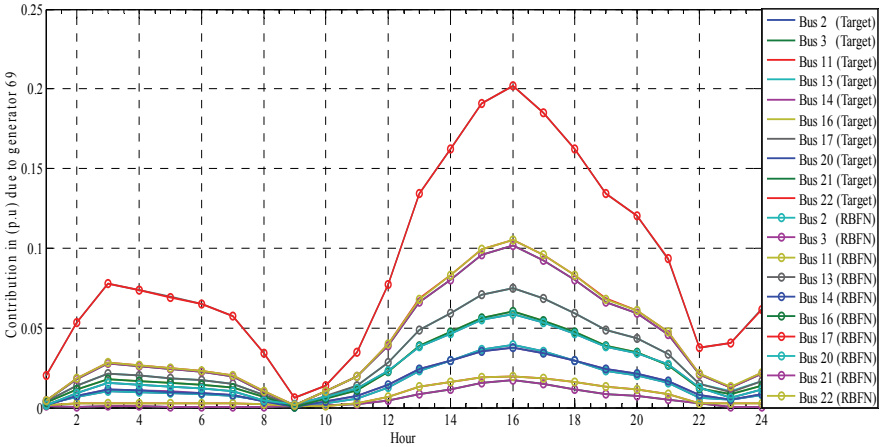


Fig. 19. Distribution of reactive power from generator at bus 69 to loads within 24 hours

Bus	Actual	RBFN Output						Modified Nodal Equations Method					
no.	load	Gen-107	Gen-110	Gen-111	Gen-112	Gen-113	Gen-116	Gen-107	Gen-110	Gen-111	Gen-112	Gen-113	Gen-116
	(MVAr)	(MVAr)	(MVAr)	(MVAr)	(MVAr)	(MVAr)	(MVAr)	(MVAr)	(MVAr)	(MVAr)	(MVAr)	(MVAr)	(MVAr)
2	22.442	0.006014	-0.06158	0.00882	0.017496	0.36319	-0.2018	0.00603	-0.06151	0.00884	0.017491	0.36318	-0.2017
3	22.282	0.070593	0.02632	0.02990	0.034195	0.11366	0.21179	0.07059	0.02634	0.02998	0.034208	0.11366	0.21241
7	22.282	0.3418	0.30944	0.1272	0.12409	-0.45274	1.656	0.34178	0.30957	0.12725	0.12405	-0.45274	1.655
11	22.839	0.03888	0.04396	0.01367	0.01214	-0.09954	0.21848	0.03895	0.04395	0.01364	0.012157	-0.09952	0.21858
13	22.76	0.083069	-0.08600	0.04681	0.066985	0.78692	-0.1534	0.08310	-0.08600	0.04681	0.066996	0.78688	-0.1536
14	21.487	0.057507	-0.03771	0.03022	0.041412	0.42551	-0.0309	0.05756	-0.03767	0.03027	0.041406	0.42552	-0.0309
16	21.487	0.058085	-0.07633	0.03432	0.050523	0.67031	-0.1639	0.05811	-0.07632	0.03432	0.050547	0.67026	-0.1639
17	21.884	0.23004	0.56429	0.05054	0.002111	18.211	2.346	0.22997	0.56432	0.05057	0.002159	18.212	2.348
20	22.919	0.1559	-0.01169	0.07312	0.091547	0.6119	0.2164	0.15603	-0.01170	0.07315	0.091584	0.61188	0.21644
21	22.76	0.15113	-0.08834	0.07843	0.10628	0.94974	-0.0719	0.15116	-0.08837	0.07845	0.10632	0.9498	-0.0719
22	22.362	0.15457	-0.09030	0.08020	0.10867	0.92085	-0.0853	0.15456	-0.09031	0.08021	0.1087	0.92084	-0.0854
23	21.248	0.028327	0.24962	-0.01151	-0.04082	-0.38525	0.95643	0.02828	0.24965	-0.01150	-0.04088	-0.3852	0.95638
28	22.68	0.069909	-0.02159	0.03437	0.044734	0.59457	0.03467	0.06986	-0.02162	0.03436	0.04475	0.59454	0.03472
29	22.282	0.068016	-0.00258	0.03164	0.039312	0.64619	0.10095	0.06805	-0.00250	0.03164	0.039349	0.64619	0.10104
33	31.036	0.10281	-0.17039	0.06418	0.097457	1.096	-0.4001	0.10277	-0.17036	0.06419	0.097468	1.096	-0.4
35	23.078	-0.063832	0.05199	-0.03457	-0.04824	-0.3362	0.05991	-0.0638	0.05203	-0.03457	-0.04825	-0.33624	0.06035
39	22.282	0.31646	0.07433	0.13858	0.16323	0.52581	0.82355	0.3163	0.07434	0.13864	0.16325	0.52581	0.82333
41	22.282	0.50263	0.3398	0.19844	0.20875	0.23008	2.061	0.50257	0.33997	0.19845	0.20874	0.23008	2.060
43	22.282	0.12944	-0.12873	0.07239	0.10316	0.61613	-0.2036	0.12945	-0.12871	0.07240	0.10316	0.61614	-0.2039
44	22.282	0.23245	-0.11355	0.11843	0.15832	0.55532	0.02799	0.23244	-0.11352	0.11843	0.15837	0.55533	0.02814
45	30.24	0.14994	-0.04177	0.07327	0.094882	0.26658	0.12454	0.14991	-0.04186	0.073299	0.094987	0.26655	0.12442
47	46.156	0.17256	-0.01586	0.08128	0.10203	0.22298	0.23989	0.17253	-0.01587	0.081271	0.10205	0.22294	0.23954
48	30.24	0.0083913	-0.01704	0.00552	0.008596	0.026759	-0.0438	0.00826	-0.01704	0.00550	0.008619	0.026778	-0.0437
50	22.282	0.1219	-0.02881	0.05915	0.076116	0.17817	0.12473	0.12205	-0.02880	0.059164	0.076136	0.17818	0.12482

Table 12. Analysis of reactive power allocation for selected generators in the IEEE 118 bus system

51	22.282	0.19109	-0.06871	0.09492	0.12455	0.29682	0.1205	0.19108	-0.06869	0.09494	0.12458	0.2968	0.12057
52	22.282	0.26658	-0.12054	0.13482	0.17936	0.43472	0.08783	0.26662	-0.12039	0.13488	0.17942	0.43477	0.08748
53	22.282	0.26907	0.1425	0.11017	0.12086	0.20019	0.98043	0.26922	0.14249	0.11021	0.12087	0.20023	0.98073
57	30.24	0.46583	0.21367	0.19396	0.2167	0.37681	1.584	0.46596	0.21368	0.19399	0.21671	0.37688	1.584
58	30.24	0.52403	0.31769	0.21048	0.22596	0.3542	2.042	0.52395	0.31771	0.21051	0.226	0.35413	2.043
60	38.198	0.46826	0.17973	0.19827	0.22563	0.39709	1.499	0.46806	0.17974	0.1983	0.22566	0.39709	1.500
67	22.282	0.14639	0.0104	0.066526	0.08103	0.16252	0.31679	0.14638	0.0104	0.066528	0.08104	0.16252	0.31678
75	22.282	-0.28563	-0.17173	-0.11374	-0.12204	-0.18192	-0.9441	-0.2856	-0.17227	-0.11379	-0.12211	-0.18193	-0.9455
78	36.607	0.68374	0.15851	0.3118	0.36798	0.36303	0.61565	0.68472	0.15851	0.31193	0.36825	0.36319	0.61571
79	25.466	0.71643	0.21628	0.32115	0.37354	0.37132	0.8582	0.71669	0.21646	0.32128	0.3736	0.37138	0.85835
82	29.445	1.215	0.18154	0.54397	0.65232	0.45164	0.53679	1.215	0.18158	0.54398	0.65233	0.45166	0.53681
83	30.24	0.41881	0.057094	0.18851	0.22664	0.14244	0.13402	0.41883	0.057092	0.18848	0.22665	0.14244	0.13402
84	29.445	-0.50926	-0.19681	-0.21449	-0.24384	-0.1796	-0.5477	-0.5092	-0.19685	-0.2145	-0.24384	-0.1796	-0.5476
86	30.24	-0.50221	-0.14519	-0.21664	-0.25197	-0.15772	-0.34537	-0.5023	-0.1451	-0.21667	-0.25204	-0.15773	-0.3453
88	30.24	-1.130	-0.67583	-0.4443	-0.47638	-0.42418	-1.861	-1.130	-0.6758	-0.44431	-0.47644	-0.42418	-1.861
93	29.445	-0.06821	0.16569	-0.03630	-0.06254	0.0097741	0.29743	-0.0682	0.16568	-0.03632	-0.06256	0.0097718	0.29774
94	28.649	-0.71453	1.372	-0.21517	-0.41236	-0.27547	-0.48801	-0.7145	1.372	-0.21517	-0.41237	-0.27545	-0.4876
95	30.24	1.539	0.37092	0.67386	0.79238	0.43411	0.66494	1.539	0.37096	0.67379	0.79239	0.43415	0.66502
96	27.853	1.172	0.32145	0.51324	0.59942	0.39961	0.77962	1.172	0.32148	0.51324	0.59942	0.39965	0.77957
97	31.036	0.89174	0.47193	0.37349	0.41072	0.39477	1.548	0.89179	0.47195	0.37348	0.41077	0.39481	1.548
98	22.282	0.59015	0.72166	0.32989	0.33026	0.11952	0.20614	0.59018	0.72167	0.32991	0.33027	0.11959	0.20652
101	19.895	2.097	0.47469	1.034	1.226	0.15317	-1.702	2.096	0.47475	1.034	1.226	0.15325	-1.703
102	22.282	-0.99718	-0.23722	-0.4167	-0.48922	-0.2573	-0.56698	-0.9971	-0.2371	-0.41669	-0.48921	-0.25731	-0.5664
106	20.691	3.952	1.101	-1.209	-1.611	-0.39309	-0.02700	3.953	1.101	-1.209	-1.610	-0.39313	-0.0261
108	16.712	2.931	0.77924	1.349	1.583	-0.042595	-1.570	2.931	0.77967	1.349	1.583	-0.042592	-1.570
109	18.303	-2.189	17.388	-0.87222	-2.934	-0.071792	1.069	-2.189	17.381	-0.87225	-2.934	-0.071785	1.069
114	22.282	0.13799	0.036191	0.060094	0.070396	1.404	0.34794	0.13803	0.03614	0.060138	0.070399	1.404	0.34807
115	21.487	0.05025	-0.00954	0.02414	0.030845	0.48589	0.045807	0.05031	-0.0094	0.024152	0.030837	0.48589	0.04580
117	22.282	0.10023	-0.05713	0.05187	0.070141	0.68789	-0.02434	0.10024	-0.0571	0.051878	0.070164	0.68786	-0.0244
118	19.895	0.40966	0.092916	0.18067	0.21309	0.30092	0.6676	0.40954	0.09295	0.18067	0.21317	0.30094	0.66786

Table 12. Analysis of reactive power allocation for selected generators in the IEEE 118 bus system (cont.)

7. Conclusion

The proposed real and reactive power allocation methods have been tested in this chapter for 25 bus and IEEE 118 bus systems. Table 13 shows the advantages and improvement in the computation time of the developed ANN and RBFN vs. MNE Method. In the 25 bus system, the developed ANN is compared with the MNE Method while for large system like IEEE 118, RBFN is compared with MNE because for large bus system ANN requires large number of networks and hence large computational time for training. It is observed that, as the number of buses increase (i.e. IEEE 118) the computational time in the MNE Method increases proportionally (i.e. for real power allocation is 3,000 msec and for reactive power is 2,911 msec) while for developed RBFN it remain almost same (i.e. for real power allocation is 15 msec and for reactive power is 15 msec) as shown in Table 13.

Test System	Computational time in msec					
	MNE		ANN		RBFN	
	Real Power Allocation	Reactive Power Allocation	Real Power Allocation	Reactive Power Allocation	Real Power Allocation	Reactive Power Allocation
25 bus	1314	908	45	45	---	---
IEEE 118 bus	3000	2911	---	---	15	15

Table 13. Comparative computational time for MNE, ANN, and RBFN methods for different bus system

8. References

Abdullah, S.S (2008). *A Short Course in Artificial Neural Network,* Desktop, ISBN, Malaysia

Bialek, J. ; (1996). Tracing the flow of electricity, *IEE Proceedings Generation, Transmission & Distribution,* Vol.,143 No., 4 (313-320)

Chu, W.; Chen, B. & Liao, C. (2004). Allocating the Costs of Reactive Power Purchased in an Ancillary Service Market by Modified Y-Bus Matrix Method, *IEEE Transaction on Power system,* Vol.,19 No., 1 (174-178)

Cheng, J.W.M. (1998). Studies of Bilateral Contracts with Respects to Steady-State Security in a Deregulated Environment, *IEEE Transaction on Power system,* Vol.,13 No.,3 (1020-1025)

Haque, R .; & Chowdhury, N. (2005). An Artificial Neural Network Based Transmision Loss Allocation For Bilateral Contracts, *Proceedings of the 18th Annual Canadian Conference on Electrical and Computer Engineering,* pp.2197-2201, Canada, May 2005

Tsoukalas, LH.; & Uhrig, RE. (1997). *Fuzzy and Neural Approaches in Engineering,* Wiley, ISBN, New York

Reta, R. ; & Vargas,A . (2001). Electricity Tracing and Loss Allocation Methods Based on Electric Concepts, *IEE Proceedings Generation, Transmission & Distribution,* Vol.,148 No., 6 (518-522)

Part 5

Mechanical Engineering

15

The Applications of Artificial Neural Networks to Engines

Deng, Jiamei, Stobart, Richard and Maass, Bastian
Loughborough University
UK

1. Introduction

Artificial Neural Networks (ANN) provide a broad spectrum of functions which are required in the field of engine applications (modelling, especially for controller design, on-board testing and diagnostics). Exhaust emissions laws are becoming progressively more stringent, while the pressure on fuel economy has been intensifying significantly in the last few years. For diesel engines, a large number of technologies, such as, multi-pulse injection and variable valve actuation, show significant promise to both improve fuel economy and reduce exhaust emissions.

Such technologies lead to high degree of freedom systems. Therefore, the engine management system has to handle this increased complexity. The traditional orthogonal grid look up tables will increase exponentially as the degrees of freedom increase. This will increase the complexity and cost of the mapping and calibration. The electronic control unit (ECU) memory consumption will increase in parallel. Use of non-linear functions and in particular neural networks is offering one important route to managing the data tables and achieving the overall goal of reducing the emissions and improving fuel economy. The need for speed and accuracy in the modelling process tends to militate against phenomenological methods

Moreover, in the general control system design, variables, such as exhaust temperature and exhaust manifold pressure, are the usual feedback signals. The brake specific fuel-consumption (BSFC) and emissions (concentration or specific) are the objective variables to which the controller set points are set in order to achieve minimum values. All of these variables can potentially be represented by black-box models. Brahma et al. proposes a dynamic model as the basis for a fuel path control system (Brahma et al., 2004). Wu et al. demonstrated a neural network approach to represent air flow rate (Wu et al., 2004), Maass et al presented a NO_x prediction neural network model (Maass et al., June 2009) and Maass et al presented a smoke prediction neural network model (Maass et al., November 2009].

Real-time operation and the mapping of complex, highly non-linear and dynamic patterns in engine behaviour are challenges that have to be met in modern combustion engines. Neural networks can handle single-input single-output up to multiple-input multiple-output problems, classification tasks and also function approximation. Their generalisation to unforeseen situations enables a wide application if the design of input data captures all the dynamics of the system. In addition, architectures and combinations of networks have a considerable impact on the performance level. We will address these challenging areas.

Firstly, this chapter will address some data collection procedures, from the design of the experiment to neural network identification. The data acquisition for network development

is crucial and the design of experiments has a significant impact on the model performance and data collection length, especially for engine systems. We will explain how to choose data perturbation signal, design of experiment to achieve minimum data. We will use practical engine examples to demonstrate these issues. For the application to engines, the relation should be explainable through the chosen inputs and the choice is influenced by the understanding of relations between inputs and outputs. Acquisition of data needs to be done accurately. It needs to be determined if transient behaviour or steady-state operation provide sufficient features for training and validation. The more features the training data covers, the better the network is trained for generalisation of engine behaviour.
Secondly, this chapter addresses architectures and combinations of networks, the application of ANN and combination of those in engine diagnostics and controller development. Combinations of ANN into groups are described achieving improved overall model behaviour. Here, task distribution into special subtask or error reduction through model redundancy can lead to the best possible result. The combination of ANN includes specialised networks trained for subtasks combined with others resulting in a superior task solution. Task distribution helps in overcoming generalisation problems by including redundant networks whose best result is chosen for solution of a specific task.
Thirdly, practical application examples are shown in the domain of emission modelling and estimation of on-board diagnostics of NO_x and PM for heavy- and medium-duty diesel engines (Maass et al., 2009; Maass et al., 2009). It will also cover Non-linear autoregressive exogenous input (NLARX) neural networks to represent intake manifold pressure, exhaust manifold temperature, exhaust manifold pressure to support control system development (Deng et al., 2010). Neural networks are chosen due to their capability to represent complex and highly nonlinear input/output relationships and can be used to represent the plant during control simulation, and the behaviour of nonlinear control methods.

2. Architecture choices of neural networks

2.1 Introduction of architectures

The choice of network architecture is dependent on the problem. Classification, linear or non-linear problems, with or without underlying system dynamics guides the choices of network composition and the topology. In general it can be distinguished between three types of networks:

- Single-Feedforward Networks (SLFN)
- Multi-Layer Feedforward Networks (MLFN)
- Recurrent Networks (RNN).

Where the single feedforward network describes a simple mapping network it can be used in classification or for mapping of simple input output functionality. It is defined through a single layer of neurons. Hence, the knowledge storage capacity is restricted and only simple logic relations can be mapped. An extension of this is the multi-layer feedforward network, also found as multi-layer perceptron. This network architecture is defined through a minimum of one hidden layer of neurons. The number of hidden layers can be increased dependent on the problem. However, literature states (reference) that a multi-layer perceptron with three hidden layers is sufficient to map every continuous function by adding a certain number of neurons to meet required complexity. However, big growing networks can be ill-posed for overtraining and be difficult to implement in real-time applications. Therefore, recurrent structures of networks are in place that will accommodate

the underlying output dynamics, a feature that is of particular interest with engine applications. In turbocharged combustion engines intake and exhaust shows related dynamics through the turbine and compressor connection. Those dynamics can be taken into consideration with output recurrent network structures.
The automotive sector has applied neural networks models in several different cases. Their main implementation is seen in control design in the area of engine operation. Hence, in engine development neural networks are used for control problems such as fuel injection, output performance or speed (Hafner et al., 2000; Ouladsine et al., 2004). In addition, advanced control strategies as variable turbine geometry (VGT), exhaust gas recirculation (EGR) or variable valve timing (VVT) have been in the focus of ANN modelling (Thompson et al., 2000). Nevertheless, the application is also used for virtual sensing such as emissions (Hanzevack, 1997; Atkinson, 2002) or as described in Prokhorov (Prokhorov, 2005) for misfire detection, torque monitoring or tyre pressure change detection.
The combustion process itself has been investigated and parameters been modelled with neural networks by different authors (Potenza et al., 2007; He et al., 2004). Potenza et al. developed a model estimating Air-to-Fuel Ratio (AFR) or in-cylinder pressure and temperature on the basis of crankshaft kinematics and its vibrations. In the work of He et al. combustion parameters and emissions are modelled under the consideration of boost pressure and EGR.
Typical network structures in these investigations have been the NLARX as has also presented in the example application in the previous section. The NLARX structure can accommodate the dynamics of the system by feeding previous network outputs back into the input layer. It also enables the user to define how many previous output and input time steps are required for representing the systems dynamics best. Other network structures include the radial-basis function networks or single layer feedforward networks for classification problems such as misfire indication or component failure detection.
This section describes the commonly applied architecture of the NLARX model. In addition recent investigations on combinations of artificial neural networks for more efficient applications are presented in a practical example for smoke emission output prediction.

2.2 The NLARX architecture

Amongst several architecture styles the NLARX model structure is a commonly used structure and is presented here. For further topologies the literature shows many examples as can be found in Haykin or Hagan (Haykin, 2001; Hagan, 1999).
A typical structure of a NLARX model is illustrated in Figure 1. The inputs are represented by $u(n)$ and the outputs are described by $y(n)$. The inputs are represented by $u(n)$ and the outputs are described by $y(n)$. The formulation of this NLARX model can be described as:

$$y(n) = F(y(n-1), ..., y(n-ny), u(n), ..., u(n-nu+1)) \tag{1}$$

where ny is number of past output terms used to predict the current output, nu is the number of input terms used to predict the current output.
Each output of an NLARX model is a function of regressors that are transformations of past inputs and past outputs. Usually this function has a linear block and a nonlinear block. The model output is the sum of the outputs of the two blocks. Typical regressors are simply delayed input or output variables. More advanced regressors are in the form of arbitrary user-defined functions of delayed input and output variables.

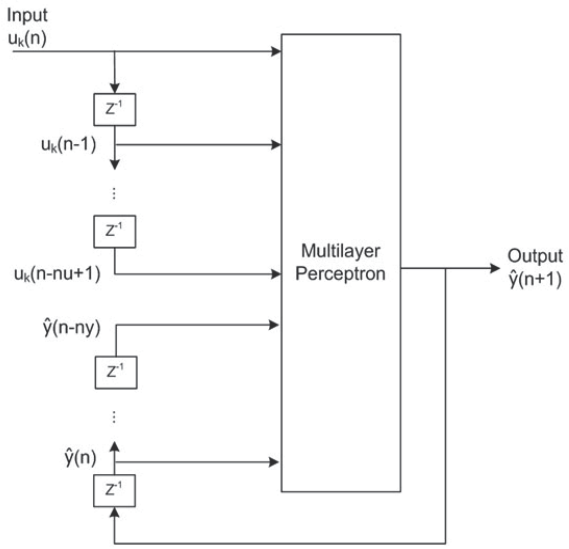


Fig. 1. Canonical representation of a NLARX model structure

The NLARX model training can be cast as a non-linear unconstrained optimization problem:

$$\min_{\theta} F_M(\theta, Z_M) = \frac{1}{2M} \sum_{t=1}^{M} \|y(k) - \hat{y}(k|\theta)\|^2 \tag{2}$$

where $Z_M = [y(k), u(k)]_{k=1,...,M}$ is a training data set, $y(k)$ represents the measured output which is the measured soot in the training set, $\hat{y}(k|\theta)$ is the NLARX output, $||.||^2$ is a 2-norm operation, and θ is a parameter vector, where $\theta = [\theta_1, ..., \theta_i, ..., \theta_p]$ and p is the number of parameters. The training process can be described as follows: Given a neural network described by equation 1, there is an error metric, that is referred to as performance index of equation 2. This index is to be minimised and represents the approximation of the network to some given training patterns. The task will be to modify the network parameters θ to reduce the index $F_M(\theta, Z_M)$ over the complete trajectory to achieve the minimal value.

3. Data collection

Data collection should capture as much information possible from the engine application, either through design of experiment or using perturbation signals. This section will discuss the definition of the engine test where the target of the modelling exercise is to represent gaseous emissions, using random signals as perturbation signals and design of experiment method to decide the data requirements. .

Data acquisition is a key element for successful modelling of systems behaviour. In the field of neural network modelling the training data is crucial for creating a good generalising network covering a broad range of the systems behaviour. Hence, a sufficient design of experiments is a key for a successful neural network design.

An efficient and sufficient training requires a data generation strategy that defines the least required data covering the broadest engine operation range. This data set does not necessarily need to contain all different operation states. If it contains the main system

dynamics represented in characteristic features the network would be able to generalise engine states in between recorded data. However, missing out extreme states in the operation may result in a lack of training information. Neural networks cannot extrapolate states that are not covered by the training data as shown in the subsection.
Data collection can be divided into the following categories for diesel engine applications:

1. Predefined engine tests that are used for engine calibration or meeting legislation requirements.
2. Pseudo-random signal generation for engine parameters such as fuel-rail pressure or start of injection that explore a broader range of engine performance.
3. Design of experiment, such as classical, space-filling or optimal design experiments.

This section will use the examples to cover these three aspects of the data collection.

3.1 Predefined engine tests

New emission regulations are going to take effect within the next years in Europe and North America. These implementations bring more and more stringent Emission standards. Different regions have different engine requirement tests. The Non-Road Transient Cycle (NRTC) is an engine dynamometer transient driving schedule of total duration of about 1200 seconds. The speed and torque during the NRTC test is shown in Figure 2. It is a cycle that was devised by the Environmental-Protection Agency (EPA) of the United States of America to represent the range of operating conditions of off-highway machinery. It is the standard test cycle for Tier 4 emissions standards. Normally, the motivation for this choice of cycle is twofold. Firstly, experience has shown that this is one of the most challenging cycles in terms of emissions modelling. Secondly, engine manufacturers must conform the emissions legislation of which the NRTC cycle is an integral part. The current trend is to design engines that pass legislative emission tests by a small margin, but where that margin must be provably robust against deterioration in engine systems. For this the data generated by this cycle is of critical importance.

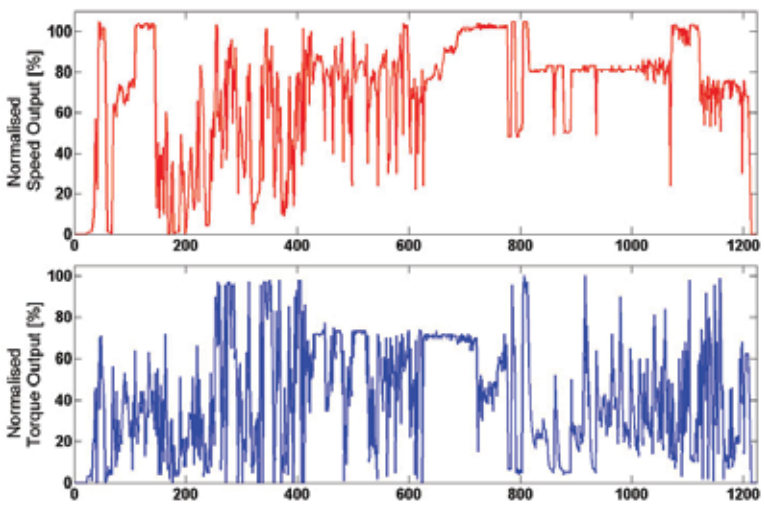


Fig. 2. Non-Road-Transient-Cycle (NRTC) displayed in normalized speed and torque characteristics – used for generation of Data set I [Dieselnet, 2009]

The data used in this section originates from two independent experiments to show the general applicability of the proposed method of prediction. The first data set is created with a NRTC as it is used for certification of non-road engines meeting EPA and EU standards. In the second test a composition of test cycles is operated also including the NRTC.
DATA SET I – The first data set consists of 12 inputs and the NO_x emission output displayed in Figure 3. It is predicted on the foundation of the inputs such as: torque, boost pressure, engine speed, liquid pilot fuel quantity, final fuel injection, back pressure, intake manifold temperature, exhaust temperature, intake depression and coolant temperatures in and out. The data is sampled at a rate of 1Hz and recorded over the whole NRTC cycle range of 1200 seconds.

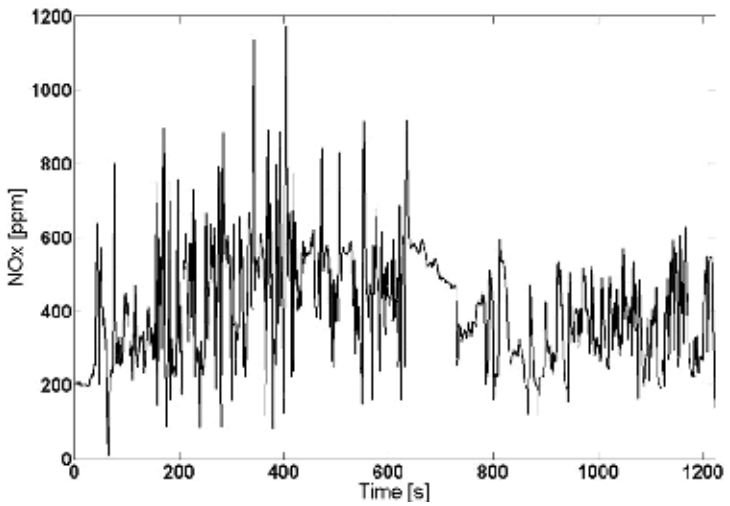


Fig. 3. Data set I - NO_x emission output generated in NRTC mode

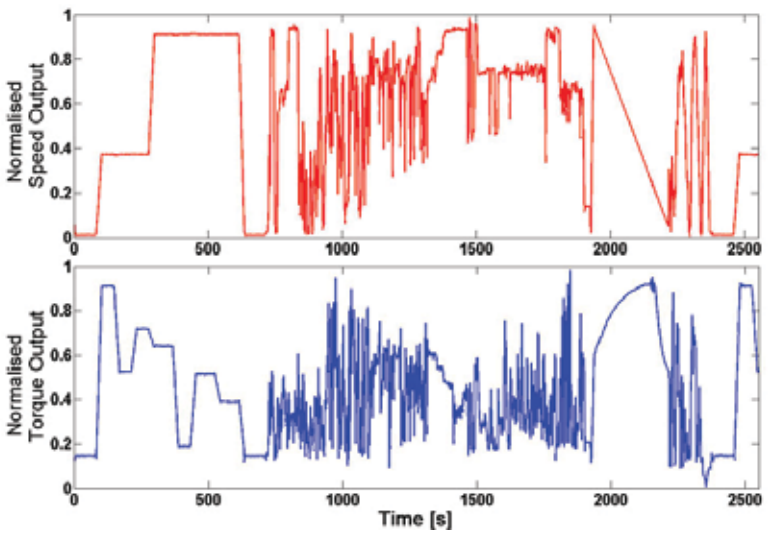


Fig. 4. Test cycle composition of NRTC, ramped modal (8 points), full load and key steady state points

DATA SET II – The second data set consists of 16 inputs to predict the NO_x emission output. The data is also sampled at 1Hz sampling frequency. The operated cycle is a composition of a NRTC, a ramped modal cycle, a full load and some key steady state points as it can be seen in Figure 4. This cycle is repeated 28 times and varied in the engine calibration maps for start of injection (SOI), fuel rail pressure (FRP) and fuel quantity.

3.1.1 Data pre-processing

Both data sets require prior processing in order to ease the training process of the NLARX model. In view of the data variability the sets are normalized to reduce the range of the inputs data. Then a further step of processing is done as follows.

DATA SET I – The initial data set provides limited data in terms of different runs and variation in signal features. Consequently, the data set is re-arranged to spread features into sets of training and validation. The signal is first divided into quarters and then arranged into training sets of the first quarter & third quarter and second quarter & fourth quarter. The result can be seen in Figure 5.

The figure shows a better distribution of signal characteristics. Each set contains a part with high frequent, high amplitudes and a lower frequency section with lower amplitudes.

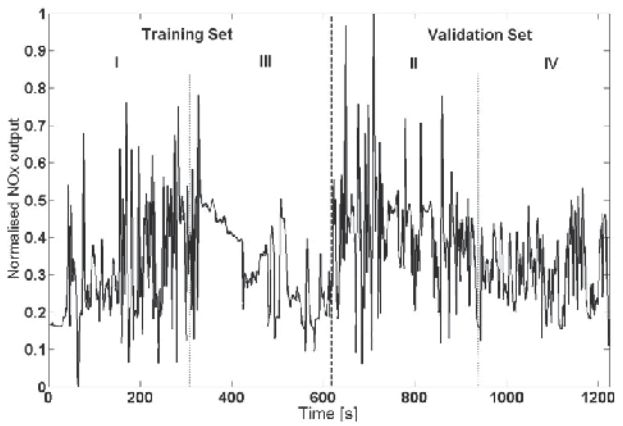


Fig. 5. Pre-processed NO_x output signal. Rearranged and composed training and validation set

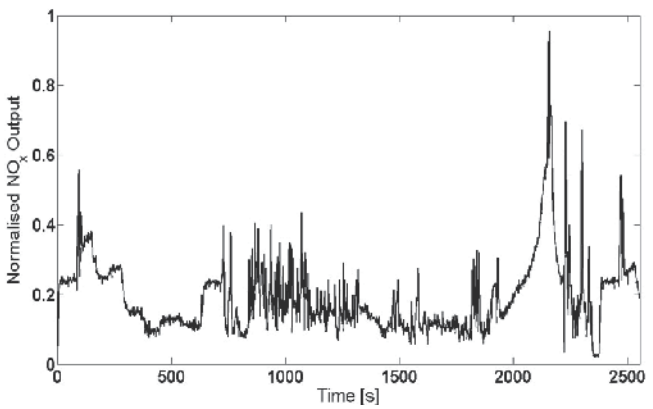


Fig. 6. Data set II training cycle of NO_x target output

DATA SET II – The second data set is split into a training set represented by the first cycle and the residual 27 cycles serve as validation sets individually. Each cycle varies slightly in its range due to the fact of cyclic variations but more importantly that different engine calibration maps are used. Start of injection (SOI), fuel-rail pressure (FRP) and fuel quantity are changed over all 28 cycles systematically. A training output can be seen in Figure 6.

3.1.2 Results

The NLARX models are "teacher forced" trained with an output target as shown before in Figure 4. and Figure 5.

DATA SET I RESULTS - The neural network is fed with the training data and trained manually. The results are promising with R^2=0.96 for the training set and R^2=0.94 for the validation set. The correlation of predicted results with the output target is realized with the correlation method coefficient of determination R^2 that is expressed through:

$$R^2 = 1 - \frac{\sum_{t=1}^{M}(y_t - \hat{y}_t)^2}{\sum_{t=1}^{M}(y_t - \bar{y}_t)^2} \tag{3}$$

Where y_t describes the measured data, $\hat{y}_t$ the prediction and $\bar{y}_t$ the mean value of the output data. The coefficient of determination shows the explained variability of the systems output by the regression model. A result of R^2=1 means an accurate model has been found whereas with a R^2 value of 0 there is no correlation between the system and the model output.

The predicted signal shows a good correspondence with the measured signal as it can be seen in Figure 7.

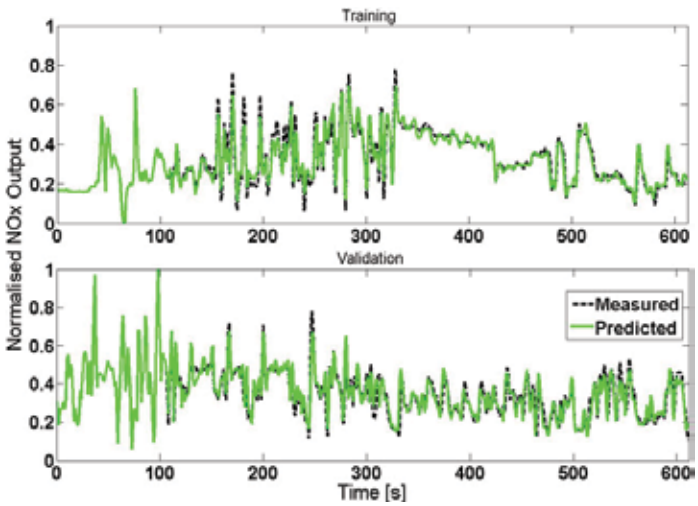


Fig. 7. Correlation of measured NO_x output with predicted neural network signal

However, the model introduces some noise in the second half of the signal. Here, the measured signal fluctuates less but the prediction is characterized with an overreaction. This is assumed to be a side effect of the good correspondence in the more oscillatory region of the test. The model is trained for a more frequent change in the signal and tends to react "nervously" on less varying patterns.

DATA SET II RESULTS – This second data set is to investigate the flexibility of the chosen network architecture. The data set stretches the signal spectrum not only by cycle variances but also with different calibration maps. For the training set a correlation of R^2 =0.95 is achieved as displayed in Figure 8.
Subsequently, this model is individually applied to the residual 27 cycles with the result displayed in Figure 9. It shows the R^2 values over the 27 validation test cycles (black line). A

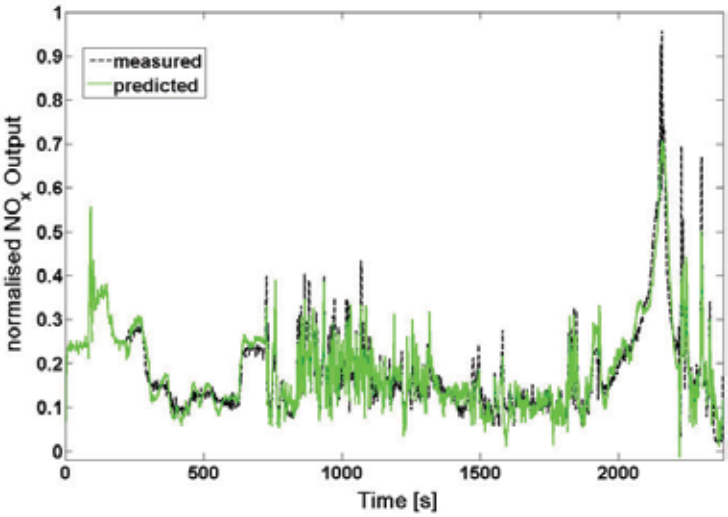


Fig. 8. Correlation between measured target output and predicted output with an R^2 = 0.95

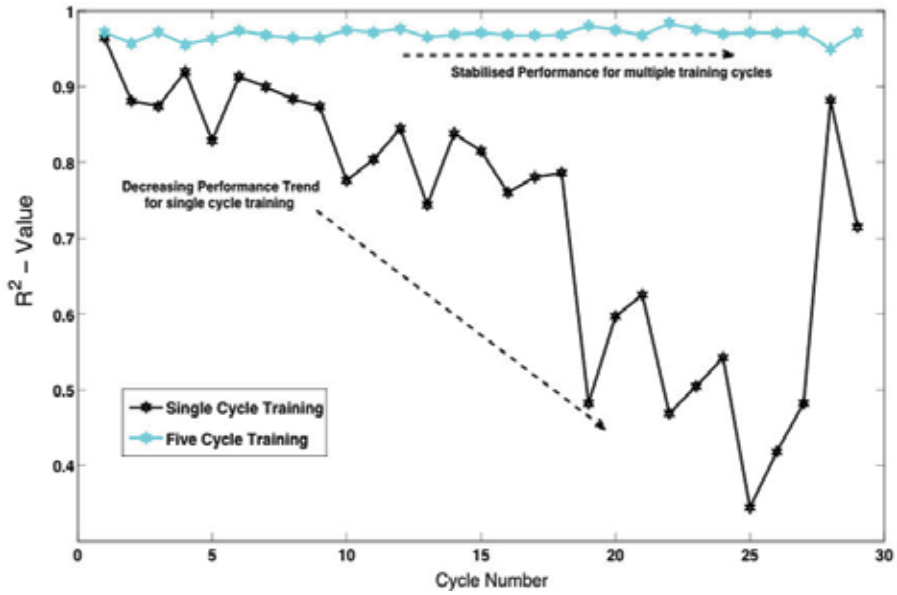


Fig. 9. Trend of prediction for 28 validation test cycles - decreasing correlation with increasing SOI timing (black line) and overcoming calibration variation with multiple training cycles (blue line)

general decreasing trend is recognized whose characteristic seems to result from the increase of SOI timing. With more advanced SOI the NO_x output increases and the signal amplitudes rise. This variance introduces an offset to the signal that cannot be handled by the present model. Hence, the calibration variance has a significant impact on the model performance. The other two calibration variables, FRP and fuel quantity show less impact on the model performance. In order to overcome this performance variance with changing engine calibration settings additional training data is required. Additional features teach the network for a broader application spectrum. The result in performance can also be seen in Figure 8. The R^2 output over all 28 cycles settles above 0.95 that is an acceptable and sufficient result (blue line). This shows that an increase of teaching features improves the knowledge area of the network and underlines the importance of sufficient engine characteristics within a predefined test cycle.

3.1.3 Conclusion

This section shows the data collection for neural network training with a predefined engine test. It is used to create a broad spectrum of engine NO_x output response of two independent heavy-duty diesel engines.

Due to a limited stock of data in the first set the training and validation set is built from a single set of data consisting of 13 channels – 12 inputs and 1 output. As a consequence of this lack of data the available set is recomposed for a better distribution of signal characteristics. This leads through manual training of the NLARX model towards a R^2 value of 0.96 and 0.94 for training and validation set respectively.

The second data set provides a broader validation spectrum because of calibration variances in SOI, FRP and fuel quantity over 28 test cycles. The training results achieve an R^2 value of 0.97 whereas the validation value ranges between R^2=0.88 down to R^2=0.76. An increase in SOI timing causes an offset in the signal that cannot be handled by the trained model. This problem requires a broader featured training set that actually includes the peaks caused from particular input characteristics such as, for example, an increasing load demand. Hence, a training set of five cycles from data set II is created that covers different calibration settings. The correlation result improves significantly over the whole set of data with the R^2 value settling above 0.95.

3.2 Random signal for data generation

In order to capture as much dynamic information as possible, random steps are used as input signals. They are discrete time signals where steps of random magnitude may occur at sampling instants with a certain probability *p*. The input signal *r* can be expressed as follows:

$$r(k) = \begin{cases} r(k-1) & with \quad probability \quad 1-p \\ e(k) & with \quad probability \quad p \end{cases} \tag{4}$$

where k is an integer, e is a discrete time white noise process with zero mean and standard deviation. In the following a modelling approach is presented with following input signals:

- Start of injection timing
- Rail pressure

- Dwell time
- Fuel ratio (quantity ratio between two pulses).

These signals are used to predict exhaust temperature and pressure, compressor mass-air flow and the NO_x output of an engine. Figure 10 and Figure 11 show the random input signals of start of injection timing and fuel-rail pressure for both training and validation purposes. They are representative for the four generated input signals. These figures show the random frequency and amplitude changes of SOI and FRP.

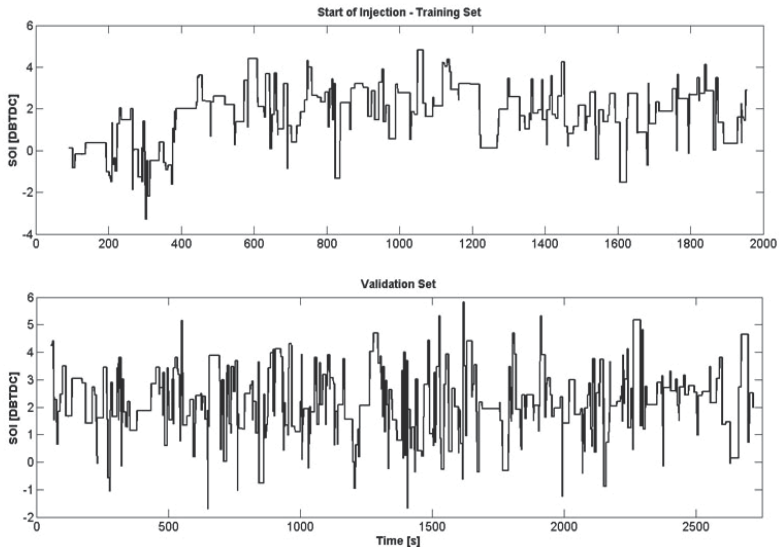


Fig. 10. Random signal of SOI for training and validation

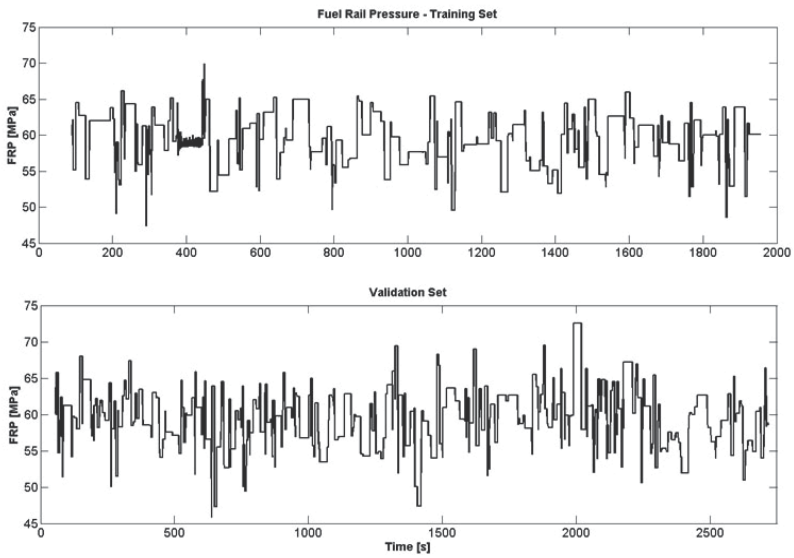


Fig. 11. Random signal of FRP for training and validation

The experiment plan was designed to cover the whole range of fuel injection space as models are effective in interpolating within the range of the training data, but not extrapolating beyond the range. With the engine running at speed of 1440 rpm and torque of 466 Nm, the injection timing spanned a range from -3 degree to 6 degree before top dead center (BTDC), rail pressure from 45 MPa to 75 MPa, dwell from 0.4 ms to 0.5 ms, fuel ratio from 0.5 to 1. Data logged for 2000 seconds was used for training purpose and data logged for a period of 2500 seconds data was used for validation.

3.2.1 Results

The results are summarized in Table 1. Four combinations of input and output are tested. Each output is predicted on the basis of all four inputs. Hence, four different models are created and trained. The correlation of the predicted results with the actual measured results is quantified using the correlation coefficient, R^2 (see (1)).

Test	**Output**	**R^2 Validation**		
		Training	**Validation**	**Fig.**
1	Exhaust manifold temperature	**0.9998**	**0.9997**	**11**
2	Compressor mass flow	**0.9998**	**0.9997**	**12**
3	Exhaust manifold pressure	**0.9957**	**0.9936**	**13**
4	NO_x	**0.9999**	**0.9999**	**14**

Table 1. Results for NLARX models for random signal training

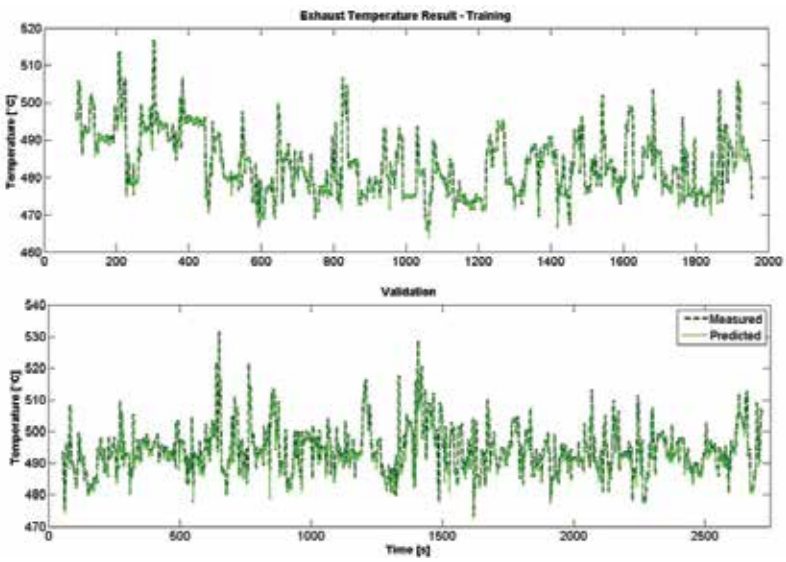


Fig. 12. Correlation of engine exhaust temperature with predicted neural network signal

The results show that the NLARX network is well able to represent the fuel path behaviour. The NLARX model has shown itself useful as a way of representing engine behaviour and that could be used as the basis for a diagnosis algorithm or as a fast measurement.

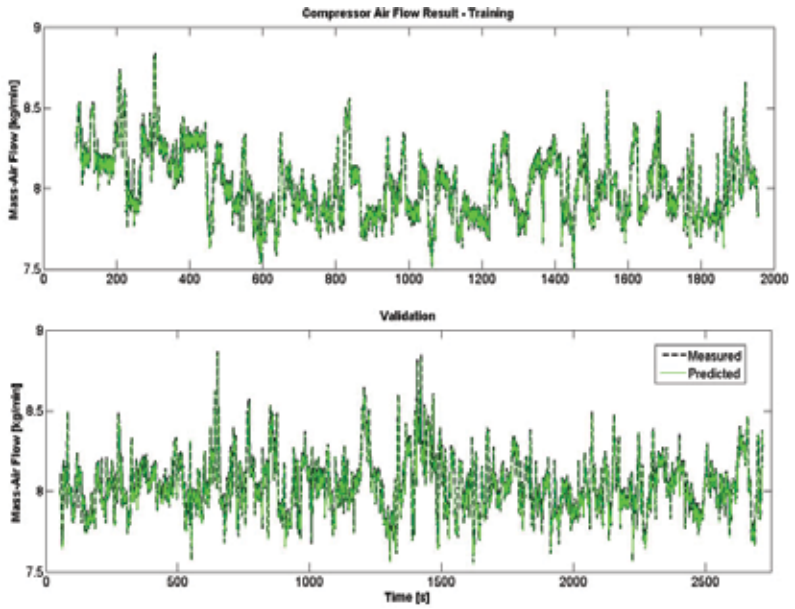


Fig. 13. Correlation of engine compressor mass-air flow with predicted neural network signal

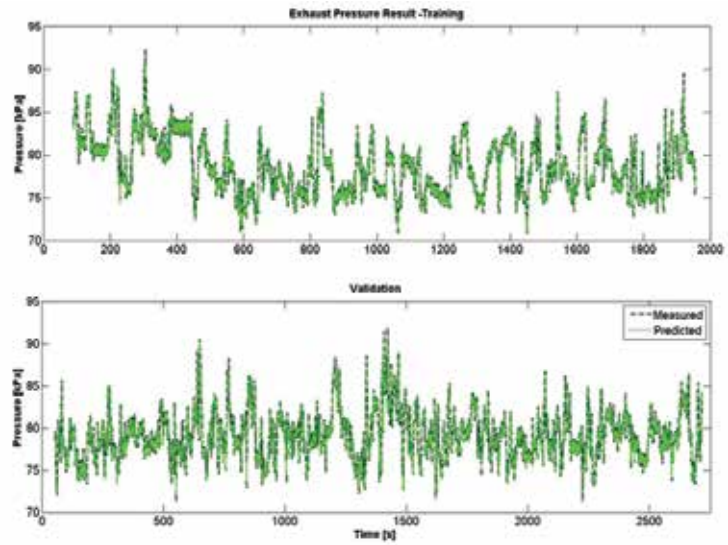


Fig. 14. Correlation of engine exhaust pressure with predicted neural network signal

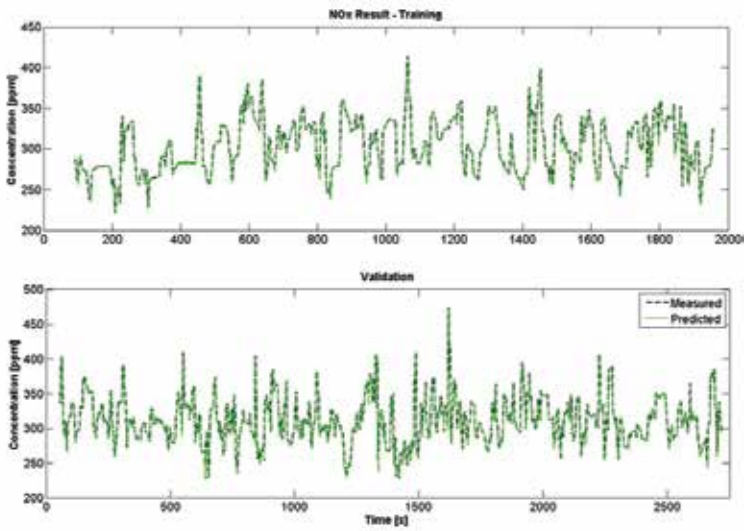


Fig. 15. Correlation of engine NOx with predicted neural network signal

3.2.2 Conclusions

The investigation of fuel path dynamics in regard to the development of a fuel path control algorithm is a novel field of study. This section has shown some initial results intended to support control systems development. The data generated for network training is created with a random signal that is used to perturb engine operation and create a variance in the engine response for exhaust manifold temperature and pressure, engine compressor mass-air flow and the NO_x output. The inputs SOI, fuel rail pressure, dwell timing between injection events and the fuel ratio are varied over a reasonable range at a fixed operation point. This can be applied for several operation points in order to create wider engine behaviour characteristic. Those points can then be used for teaching a single neural network or a combination of networks applied for specific tasks.

A single NLARX model is used for each output parameter measured: exhaust temperature, compressor mass air-flow, exhaust pressure and NO_x. The models demonstrate excellent performance at the operating conditions judged by correlation coefficients close to unity. Further work is required to evaluate the potential for the NLARX model to represent behaviour across a number of operating points. Such a non-linear model is capable of supporting diagnosis processes as well as being a fast model for controls design and evaluation.

3.3 Design of experiment for data generation

This section shows using a design of experiment method to minimise the test and collect informative data for neural networks training and validation.

Figure 16 shows the schematic diagram of a diesel engine. The original engine used for generation of neural network training and validation data is a Caterpillar C6.6 heavy-duty diesel engine with EGR, VGT and VVT function. This engine is modelled in Dynasty 9.4.1 in order to simulate cost effective the engines behaviour. Dynasty is a dynamic simulation tool designed for modelling, simulation and analysis of physical systems in both transient and steady state conditions. During the simulation study, the fuel injection timing and quantity

are held constant. The data for both neural network training and validation are extracted using the Dynasty simulation software. Figure 17 shows the intake and exhaust valve lift. Both inlet and exhaust valve profiles can be changed freely either in the transient or steady state during the simulation.

The experiment plan is designed to cover the whole operating range of the engine. The engine speed spanned a range from 660 RPM to 2000 RPM, torque from 45 Nm to 1000 Nm, EGR from 0.1 to 0.9, VGT from 0 to 1, inlet valve phase shift from 330 degrees to 360 degrees and exhaust valve phase shift from 100 degrees to 140 degrees. The experiment was designed by using the stratified Latin hypercube design method available within the Matlab R2009b Model Based Calibration Toolbox. This design method belongs to the space-filling design style that is used for modelling processes where the system understanding is rudimentary. The purpose is to cover most of the operating range. This design created a total of 196 test points for all parameters. 168 of these test points were used for training purpose and 28 test points were used for validation purpose.

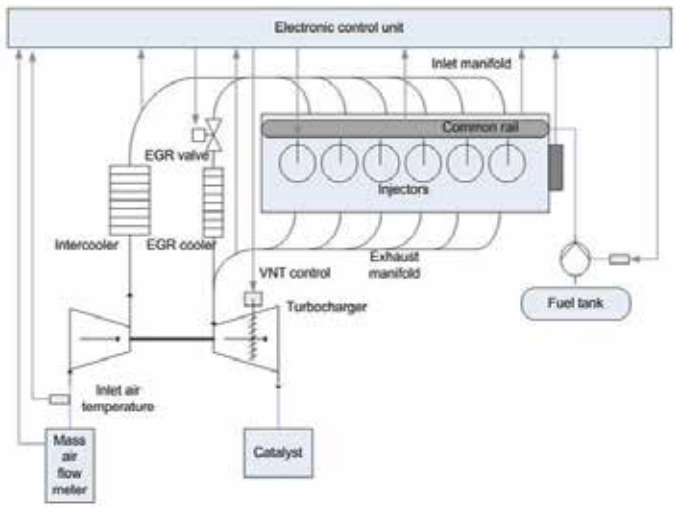


Fig. 16. Schematic drawing of a diesel engine and auxiliaries

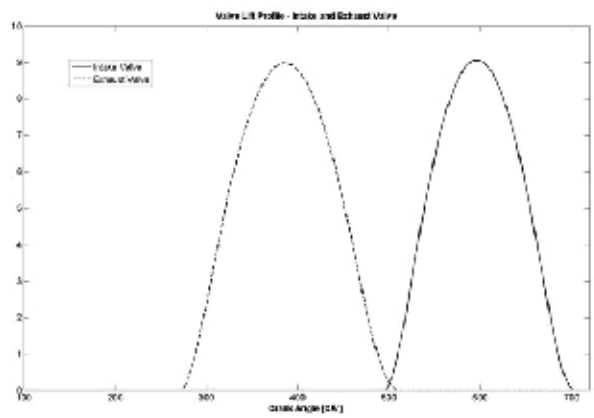


Fig. 17. Valve-Lift profile for inlet and exhaust valve

Additional designs of experiments styles are the classical approach and an optimal approach. The classical approach has been used for simple operation areas with a small number of parameters. . In case of an optimal design of experiments the system knowledge is high and the desired model type is already known. The stratified Latin hypercube design enables the definition of how many operation points per parameter are of interest and leads to an even representation of the multidimensional operation hypercube created by the six parameters in this case.

3.3.1 Results

The first neural network has one output: intake manifold pressure; and six inputs: engine speed, torque, EGR, inlet valve phase and exhaust valve phase. The results are promising with R^2=1 for the training set and R^2=0.9925 for validation set.

Figure 18 shows that the intake manifold pressure predicted from the neural network correlates closely with the generated signal of the Dynasty simulation.

The second neural network is designed to predict BSFC based on six inputs: engine speed, torque, EGR, inlet valve phase and exhaust valve phase. The results are promising with R^2=1 for the training set and R^2=0.9975 for the validation set. It can be seen in Figure 19 the predicted BSFC output of the neural network shows a good correspondence with the measured BSFC from the Dynasty model.

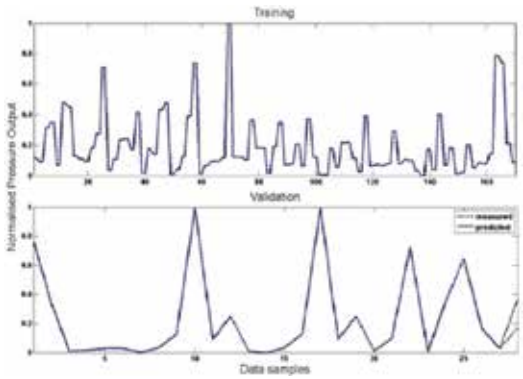


Fig. 18. Correlation of engine intake manifold pressure with predicted neural network signal

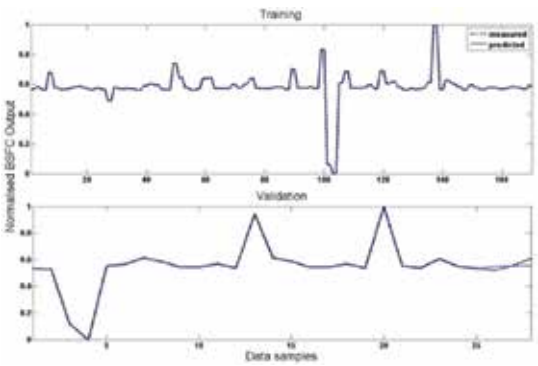


Fig. 19. Correlation of engine BSFC with predicted neural network signal

3.3.2 Conclusion

The design of experiments is a powerful tool in the optimisation of the modelling process. Progressively more complex system architectures make it difficult and eventually impossible to cover each operating point. Depending on the system knowledge different strategies for designing an experiment dictate the sampling coverage for successful modelling processes. The often small knowledge base of parameters effects on the systems response makes the space-filling design style in particular useful for neural network design. This approach allows an even distribution across the operating window and hence covering the main response characteristic for all parameters.
In this particular case for both training and validation data, the sampling points need to be increased significantly. The test points cover a minor operating range of the engine and in order to use neural networks for prediction their generalisation capability has to be increased by additional engine operation characteristics. The approach is presented for the demonstration of a design of experiment and how to use the data in teaching a NLARX that can predict intake manifold pressure and BSFC.

4. Combining Neural Networks

The complexity of today's systems makes it occasionally impossible to find a sufficiently performing single network composition, even in the case of a highly complex recurrent structure. Hence, the combination of networks has become popular where tasks are either distributed across separate networks or competitive structures with redundant networks are created (Sharkey, 1999). The literature distinguishes between modular and ensemble structures. Modular applications are defined by the fact that each network is trained for a subtask and all networks together form a superior solution. In an ensemble networks are trained differently or show different topological features but are predicting all the same output. A superior decision instance compares the results and votes for the best performance. This approach can create a more reliable performance since the optimum can be chosen from a variety of results. A third approach is the combination of modular structures and ensembles. In the following example a parallel neural network structure is composed where three individual NLARX networks are used in order to predict a superior signal that is a combination of all three. Similar to the previous NO_x example, here the smoke emissions are investigated and the behaviour is modelled by a neural network structure. The smoke signal represents in this case the solid component of particulate emissions. Smoke is assumed to be a good proxy for this emission formation.
The experiment for data generation was conducted on a heavy-duty diesel engine that is run under the conditions of an NRTC. It is applied to generate emission data for training and validating the neural network which is presented in the graph in Figure 20. The smoke output signal is predicted on the basis of 12 inputs such as torque, boost pressure, engine speed, liquid pilot fuel quantity, final fuel injection, back pressure, intake manifold temperature, exhaust temperature, intake depression and coolant temperatures for flows in and out. All parameters were used from the beginning and investigated and revised for their impact on the model.
The initial output signal shows two characteristic halves. In the first half strong fluctuations and high peaks are present, whereas the second half displays a much flatter characteristic with small oscillations. The approach of modelling and estimating the signal requires a training and validation data set. Therefore the signal is bisected. However, a training set requires preferably a broad spectrum of features provided by the signal. The signal is first

divided into quarters accordingly and then newly-arranged. As a result the training and validation set cover a high oscillating part with high peaks and a flat, low oscillating part – Figure 20. Every set consists of a correspondingly split smoke output and twelve inputs. As well as the data partitioning a normalisation process is applied to the inputs and output.

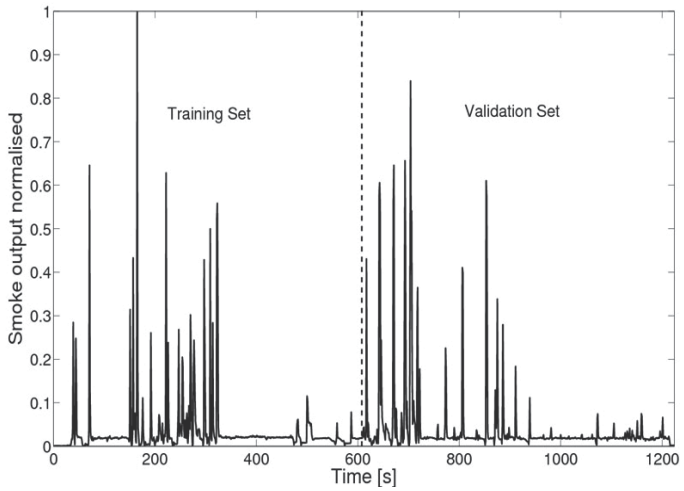


Fig. 20. Processed smoke output signal

In an initial approach of modelling the signal with an single NLARX network it was recognised that noise is introduced by the model. This occurs especially then, if the signal contains large amplitudes and high-frequencies. In Figure 21 the modelling results of a single NLARX model are plotted over the measured signal. The early phase of the signal is well predicted. However, in the second phase of the characteristic the prediction data starts oscillating in high-frequencies as well as an underlying lower frequency. The model becomes unstable. This is assumed to be forced by the training on high amplitudes in the first stage and hence the development of a hypersensitive behaviour. Other approaches are known to overcome those issues such as fuzzy logic and wavelet networks (Parasuraman & Elshorbagy, 2005). They offer a much better response to highly fluctuating signals.

Among those approaches, Guoyin et al. (Guoyin & Hongbao, 1995) introduced three classes of parallel network systems. Here, a parallel network system with multiple tasks is chosen. Lee (Lee, 1997) states that due to the approach of more than one network the risk of settling in a local minimum decreases. Additionally, the performance increases due to the fact that particular networks handle a specific subspace instead of dealing with the whole problem.

In the current work the signal is divided into different vertical layers. Consequently the amplitudes are cut and the frequency of the residual signal part is decreased. With lower frequencies the NLARX model promises satisfying results regarding performance and cost.

By trial and error three layers are determined as a reasonable degree of divisions. The first layer called lower layer (LL) contains the signal noise and low frequencies. The remaining part is split into a mid layer (ML) and a top layer (TL). The ML covers a part of the signal with a medium density of oscillations and peaks in the smoke value up to y=0.3. The residual signal peaks are covered by the TL. Its characteristic is marked by only a few single peaks, the occurrence of which is not distracted by noise or smaller peaks. The division borders in this approach are chosen as outlined in Table 2 and illustrated in Figure 22.

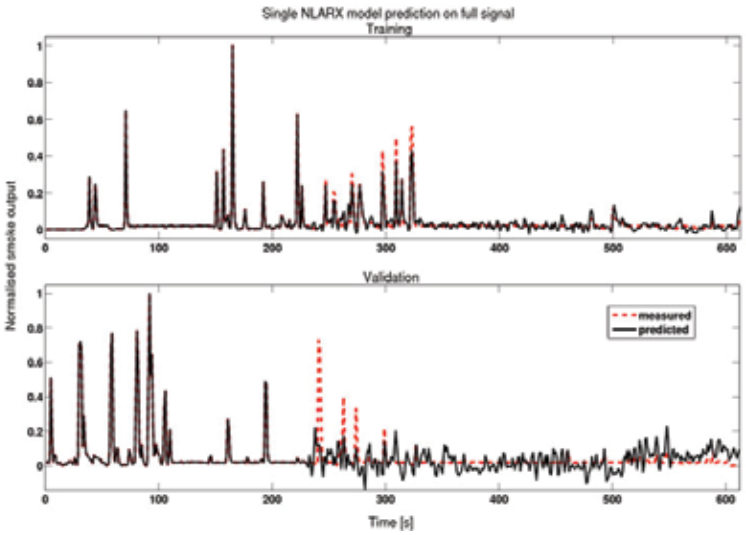


Fig. 21. Single NLARX model: measured output signal correlated versus predicted output signal

0 LL 0.035	=>	$\Delta y_{LL} = 0.035$
0.035 < ML < 0.3	=>	$\Delta y_{ML} = 0.265$
0.3 < TL < 1	=>	$\Delta y_{TL} = 0.7$

Table 2. Division borders of layer approach

Each division is processed and estimated independently. This leads to a parallel processing model structure as shown in Figure 23. The input vector *U* with its twelve input signals is used for all three independent layers whereas the predicted output is split into the three divisions, *top*, *mid* and *low*. After estimation they are combined to $\hat{y}_{overall}$ and compared against the overall measured output.

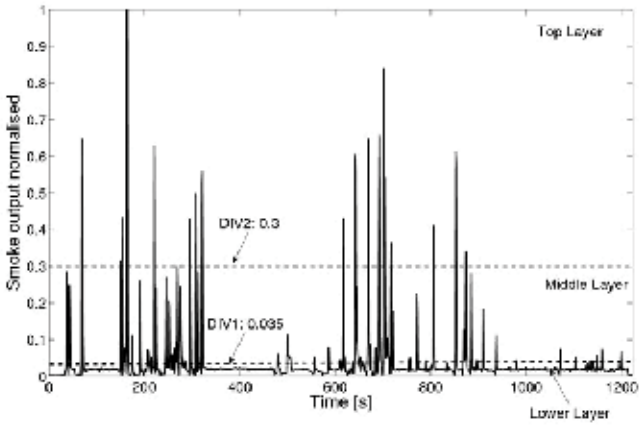


Fig. 22. Layer approach with correlating divisions

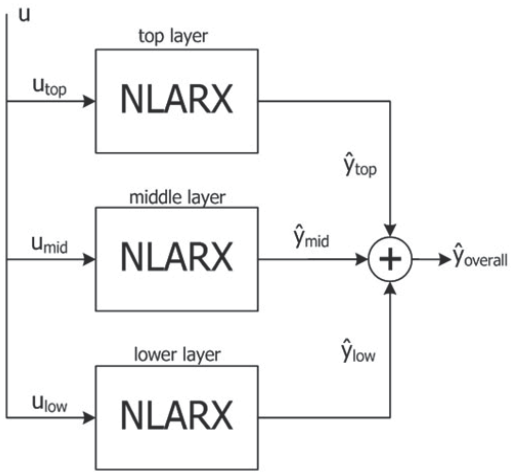


Fig. 23. Scheme of applied parallel model structure

Results - An estimation is processed by initially training and then validating an artificial neural network with the corresponding signals. Every layer is estimated independently. The NLARX-model are initialised with an arbitrarily state and taught with the corresponding training data set. Based on this data the NLARX-model is designed to estimate the desired output signal. The designing process consists of changing the design parameters in Matlab R2009b by teacher-forced learning until a satisfactory result is achieved. The design parameters are the input/output delays.

The lower part is marked by 1) the lowest values of the higher oscillations of the signal and 2) small oscillations that are introduced by noise. By cutting off a lower part of the signal a more homogeneous distribution of the height of oscillations is created. This enables a better estimation with the chosen NLARX approach.

The training of the network generates a correlation between the measured and estimated signal of R^2=0.97. Validating the network leads to a performance of R^2=0.95 which demonstrates the practicability of the chosen design. However, the model introduces additional noise to the signal. This effect is discussed in more detail in the following sections.

The middle layer represents the central section of the high peaks and the medium peaks. The lowest values of the large signal excursions are included in the lower layer. Through training the NLARX model achieves a correlation of R^2=0.93 with the measured signal. The model's quality is confirmed by the validation set, which achieves a performance of R^2=0.9. The performance is predictably lower than in the first layer due to the higher frequencies. Higher frequencies occur because of an expanded range of y-values.

The characteristic of the graph is marked by noise in the second, low oscillating part of the signal. It is assumed that this noise is introduced as a result of the network design. There is a fast response identified by the network when managing high oscillating signals. In consequence, this leads to an oscillating estimation signal.

The top layer covers the high peaks of the signal. Consequently high frequencies are introduced and a lower correlation performance is expected. The design process achieves a result of R^2=0.83 compared to a R^2=0.92 for the validation data. Validation shows a better result because the main peaks of the validation signal are predicted well, whereas the training signal shows some missing details in the middle part for three consecutive spikes.

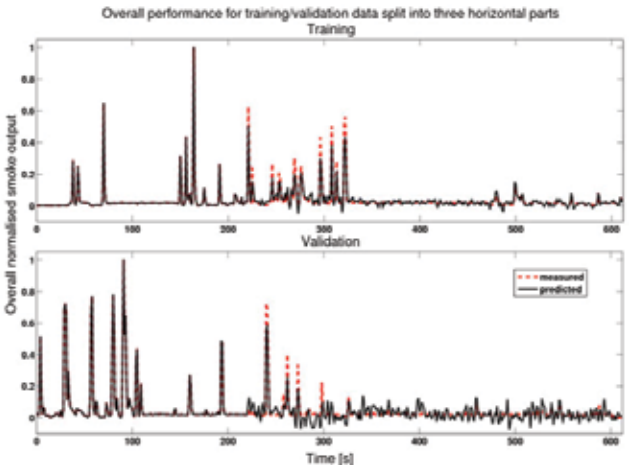


Fig. 24. Overall Performance: Training and validation sets estimation of measured and predicted signal

The overall estimation is achieved by adding the three estimated signals together and correlating it with the measured output – see Table 3. The comparison of the measured and predicted signal shows a distribution around the linear correlation in Figure 25. The reason that a cluster of points forms close to the origin is due to the fact that the most of the data samples are measured in the lower data scope. However, the results of overall correlations of the smoke output signal are R^2=0.97 and R^2=0.96 for training and validation set respectively as illustrated in Figure 24. It can be seen that parts intially classified as difficult due to big amplitude differences and high frequencies are modelled well. The patterns of high peaks and high density of oscillations show appropriate correlations. However, the flatter parts are marked by the introduction of noise through the model design as mentioned earlier.

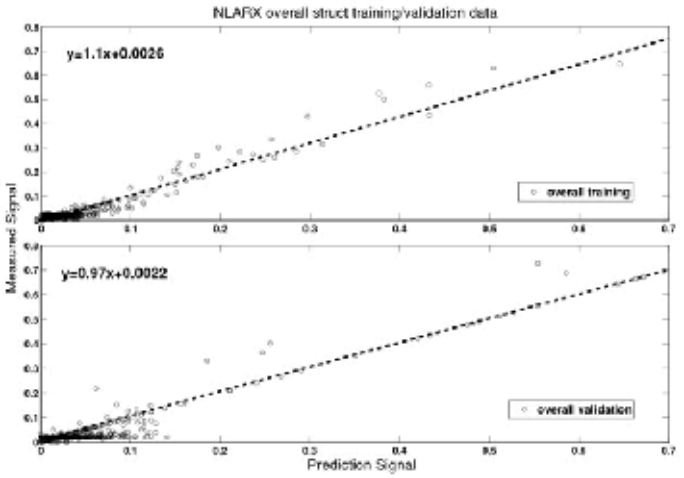


Fig. 25. Overall Performance: Training and validation sets estimation in correlation to measured data

Layer	NLARX-Model	
	train	valid
TL	0.8330	0.9163
ML	0.9328	0.9006
LL	0.9743	0.9164
Total	0.9706	0.9616

Table 3. RMS performance indication

Conclusion – Here a parallel neural network structure is presented to predict the smoke output of a diesel engine based on NLARX-models. The model is chosen due to its good generalisation. Its weakness of not being capable of high-frequency signals which is shown with a single NLARX model for comparison, is overcome by an approach of frequency filtering.
The NLARX architecture is cut into different layers to reduce the frequency bands resulting in a better overall performance. A lower layer for the signal scope that covers noise and the base of higher peaks, followed by a middle layer for medium density of oscillations and a top layer for the peak tops. This approach demonstrates the application of network combination. Three independent networks trained for different tasks can predict if combined the overall signal at a sufficient performance.
Another approach is the training of three networks with the same task just slightly different training data and in case of the prediction performance the network with the best performance wins. Here, the networks are redundant and a competitive approach is used to find the optimum output.

5. Summary

The chapter presents the application of artificial neural networks on engine applications. Several practical examples show the applicability of artificial neural networks in the domain of virtual sensing and control development support.
A critical part of for a successful modelling of engine behaviour is the generation of comprehensive system characteristic. The better the training data describes the system dynamics the better the generalisation capability of the model. A crucial part is the planning of data generation. Here the chapter presents three differen approaches:

- Predefined engine tests such as the NRTC for off-highway diesel applications
- Random control signal generation for engine response measurements
- Systematic design of experiment approach.

Another crucial part is the choice of the right model structure for the problem at hand. The chapter presents a recurrent network structure that is applicable for highly non-linear and dynamic systems. The NLARX network is presented in several different applications. A successful implementation can be seen in the virtual sensing of diesel engine emissions. However, the network has also been implemented for combustion modelling.
A last part describes the investigation of combinations of networks. Increasing complexity of systems leads to difficulties of finding cost effective network structures in view of training and operation costs. An approach is presented where a superior task, the prediciton of smoke emissions of a diesel engine is split into three individual tasks solved by independent network compounds. Other approaches can be implemented with competetive structures of redundant networks whose results are competing against each other.

Artificial neural networks can be a powerful tool for monitoring of engine operation or the design of controller applications. However, the correct training data and the optimal design are crucial for a successful modelling process.

6. References

Atkinson, C.; Traver, M.; Long, T. W. and Hanzevack, E. (2002), Predicting Smoke, *InTech*, 2002, pp. 32-35

Brahma, A.; Upadhyay, D.; Serrani, A. And Rizzoni, G. (2004). Modeling, identification and state estimation of diesel engine torque and NO_x dynamics in response to fuel quantity and timing excitations, *Proceedings of the 2004 American Control Conference*, Vol. 3, 2004, pp. 2166-2171

Deng, J.; Maass, B.; Stobart, R. K. (2009), Using Artificial Neural Networks for Representing the Brake Specific-Fuel Consumption and Intake Manifold pressure of a Diesel Engine, *Asia Pacific Power and Energy Engineering Conference 2009*, pp. 1-4

Deng, J.; Winward, E.; Stobart, R. and Desai, P. R. (2010), Modeling Techniques to Support Fuel Path Control in Medium Duty Diesel Engines, *SAE World Congress 2010*, SAE paper no. 2010-01-0332, April 2010

Guoyin, W. (1995). Parallel Neural Network Architectures, *Proceedings of the IEEE International Conference on Neural Networks 1995*, Vol. 3, 1995, pp. 1234-1239

Hafner, M.; Schüler, M.; Nelles, O. and Isermann, R. (2000). Fast Neural Networks for Diesel Engine Control Design, *Control Engineering Practice*, Vol. 8, 2000, pp. 1211-1221

Hagan, M. T.; Demuth, H. B. and Beale, M. (1999), *Neural Network Design*, PWS Publishing Company

Hanzevack, E. (1997). Virtual Sensors for Spark Ignition Engines using Neural Networks, *Proceedings of the 1997 American Control Conference*, Vol. 1, 1997, pp. 669-673

Haykin, S. (1999), Neural Networks – A Comprehensive Foundation, Prentice Hall

Lee, B. (1997). Parallel Neural Networks for Speech Recognition, *Proceedings of the IEEE International Conference on Neural Networks 1997*, pp. 2093-2097

Maass, B.; Stobart, R. K. and Deng, J. (2009). Diesel Engine Emissions Prediction Using Parallel Neural Networks, *American Control Conference*, June 2009.

Maass, B.; Stobart, R. K. and Deng, J. (2009). Prediction of NOx emissions of a heavy duty diesel engine with a NLARX model, *SAE Powertrains, Fuels and Lubricants Meeting*, SAE paper no. 2009-01-2796, November 2009

Ouladsine, M.; Bloch, G. and Dovifaaz, X. (2004). Neural Modelling and Control of Diesel Engine Pollution Constraints, *Journal of Intelligent and Robotic Systems*, Vol. 41, 2004, pp. 157-171

Parasuraman, K. and Elshorbagy, A. (2005). Wavelet Networks: An Alternative to Classical Neural Networks, *Proceedings of the IEEE International Conference on Neural Networks 2005*, Vol. 5, 2005, pp. 2674-2679

Potenza, R.; Dunne, J. F.; Vulli, S.; Richardson, D. And King, P. (2007). Multicylinder Engine Pressure Reconstruction using NARX Neural Networks and Crank Kinematics, *International Journal of Engine Research*, Vol. 8, 2007, pp. 499-518

Prokhorov, D. (2005), Virtual Sensors and Their Automotive Applications, Intelligent Sensors, *Proceedings of the 2005 International Conference on Sensor Networks and Information Processing*, 2005, pp. 411-416

Sharkey, A. J. C. (1999), Combining Artificial Neural Nets, ensemble and modular multi net systems, Springer, London

Thompson, G.; Atkinson, C.; Clark, N.; Long, T. and Hanzevack, E. (2000), Neural Network Modelling of the Emissions and Performance of a Heavy-Duty Diesel Engine, *Proceedings of the Institute of Mechanical Engineers*, Vol. 214, Part D, No. D04499, 2000

Wu, B.; Filipi, Z.; Assanis, D.; Kramer, D. M.; Ohl, G. L.; Prucka, M. J. and Divalentin, E. (2004). Using artificial neural networks for representing the air flow rate through a 2.4 litre VVT engine, SAE paper no. 2004-01-3054

16

A Comparison of Speed-Feed Fuzzy Intelligent System and ANN for Machinability Data Selection of CNC Machines

Zahari Taha[1] and Sarkawt Rostam[2]
[1]*Department of Manufacturing Engineering, University Malaysia Pahang*
26300 Gambang, Pahang,
[2]*Centre for Product Design and Manufacturing Department of Engineering Design and Manufacture Faculty of Engineering- University of Malaya*
50603 Kuala Lumpur,
Malaysia

1. Introduction

The machining process exhibits piecewise behaviour and cannot be linearly extrapolated in a wide range. It cannot be modelled effectively using theories and equations. Thus, expert systems have emerged as a major tool for decision-making in such complicated situations (Singh & Raman, 1992).

The conventional method for selecting machining parameters such as cutting speed and feed rate is based on data from machining hand books and/or on the experience and knowledge of the operator or CNC programmer. The parameters chosen in most cases are extremely conservative to protect over- matching errors from tool failures such as deflection, wear, breakage, etc. Accordingly, the metal removal rate is low due to the use of such conservative machining parameters (Park & Kim, 1998).

Guidelines on machinability data selection is normally made on the basis of the manufacturer's machinability hand book (Hashmi et al., 2003). Using machining data handbook for the choice of cutting conditions for material hardness that lies in the middle of a group is simple and straight forward. But there exists a degree of vagueness in boundary cases, where two choices of cutting speeds are applicable for one choice of material hardness. In this situation, the skilled operator makes a decision on the appropriate cutting speed, based on his experience. However, this method of data selection by individual operators is not very desirable, because it may vary from operator to operator. Therefore, it is desirable to have an operator independent data selection system for choosing machining operation (Hashmi et al., 1998).

While the output variables of the machining process depend on the cutting conditions, the decision concerning the selection of the cutting parameters have an important influence on the extent, cost and quality of the production. Due to the increased use of CNC machines and severe competition between the makers, the importance of precise optimization cutting conditions has increased (Cus & Zuperl, 2006).

Fuzzy logic can be applied to any process in which a human being plays an important role which depends on his subjective assessment (EL Baradie, 1997).
For the selection of machining parameters different methods have been proposed. Hashmi et al. (1998, 1999) applied fuzzy logic with triangular shape for selecting cutting conditions in machining operations using single input (material hardness) and single output (cutting speed) model. El Baradie (1997) presented the development of a fuzzy logic model for machining data selection using material hardness (input) and cutting speed (output) with triangular shape. A study was made by Wong et al (1999) to obtain a generalized model for metal cutting data selection. Wong and Hamouda (2003a) developed an online knowledge-based expert system for machinability data selection using two input-one output model for cutting speed and one input-one output model for feed rate.
Cus and Zuperl (2006) proposed a neural network approach for the optimization of cutting conditions. Neural networks were used by Wong and Hamouda (2003b) in the representation of machinability data to predict optimum machining parameters. Zuperl and Cus (2003) proposed a neural based approach to optimization of cutting parameters to represent the manufacturer's preference structure.
Lee and Tarng (2000) used a polynomial network to construct the relationship between the machining parameters and cutting performance. An optimization algorithm of sequential quadratic programming method was used to solve the optimal machining parameters. A gradient based multi criteria decision making approach was applied by Malakooti and Deviprasad (1989) to aid the decision-maker in setting up machining parameters in metal cutting. The optimal machining parameters for continuous profile machining for turning cylindrical stock were determined by Saravanan et al. (2003) using simulated annealing and genetic algorithm. Vitanov et al. (1995) introduced a knowledge-based interactive approach for optimum machining parameter selection in metal cutting using multi-objective probabilistic geometric programming and artificial techniques. The machining parameters were optimized based on the Taguchi method in a proposed model by Nian et al. (1999) considering the multiple performance characteristics including tool life, cutting force and surface finish.
The fuzzy logic approach is used in different applications. For example, Hashmi et al. (2000) have used the fuzzy logic model to select drilling speeds for different materials in a drilling operation. A fuzzy logic based expert system was developed by Liu at el. (1996) for diagnosing defects in rolling element bearing and offering instructions and guidelines for the detection of these defects. A user friendly fuzzy-expert system was introduced by Yilmaz et al. (2006) for the selection of electro discharge machining process parameters using triangular membership function and expert rules. Arghavani et al. (2001) presented the application of a fuzzy decision support system by applying fuzzy logic theory for gasket selection and gasket sealing performance.
Researchers have applied ANN methods in a wide variety of fields, ranging from science to business and engineering (Ghiassi & Saidene, 2005). Neural networks have the potential to provide some of the human characteristics of problem solving that are difficult to simulate using the logical, analytical techniques of expert system and standard software technologies. The immediate practical implication of neural computing is its emergence as an alternative or supplement to conventional computing systems and AI techniques. As an alternative, neural computing can offer the advantage of execution speed once the network has been

trained. The ability to train the system with data sets, rather than having to write programs, may be more cost effective (Medsker, 1996).
In this chapter the performance of Speed-Feed Fuzzy (SFF) intelligent system is compared with Artificial Neural Networks in finding the selection of machining parameters which can result in a longer tool life, a lower cutting force and better surface finish. The proposed system is expected to contribute in the selection of optimal parameters that will assist process planners, CNC programmers, production engineers and machinists with easy access to data necessary for effective machining process.

2. Fuzzy model for machinability data selection

2.1 Fuzzy logic concept

The fuzzy logic first proposed at 1965 by Lotfi Zadeh(Zadeh, 1965). The fuzzy set theory provides means for representing uncertanity. It is used to model the kind of uncertainity associated with imprecision. It offers the concept to compute a model with words using human expertise used in daily language. The fuzzy set theory provides a mechanism to represent linguistic constructions. The fuzzy inference engine build on a set of rules, so, it is called fuzzy- rule based system. These rules are supplied by an expert or a decsion-maker to formulate the mapping of the system which can perform approximate reasoning similar to but much more primal than that of the human brain (Sivanandam, 2007).

2.2 Fuzziness and fuzzification

In a fuzzy set, the fuzziness is characterized by its membership function. It classifies the element in the set, whether it is discrete or continuous. The membership functions can also be formed by graphical representations. The fuzzification procedure is used to control the fuzziness of the fuzzy set and it is an important concept in the fuzzy logic theory where the crisp quantities are converted to fuzzy quantities (Arghavani et al., 2001; Sivanandam et al., 2007).

2.3 Membership functions for fuzzy variables

The SFF model use multi input- multi output fuzzy variables for the selection of machining parameters (Fig. 1). The multi inputs are material hardness (BHN) and depth of cut (DOC) and the multi outputs are cutting speed and feed rate. The fuzzy expressions for the inputs and outputs are shown in Table 1.
The model is applied for turning operation for wrought carbon steels using different types of tools. The extracted data from Machining Data Handbook (Metcut Research Associates Inc., 1980) are tabulated as shown in Table 2.
Different applications of the fuzzy control technique use a specific shape of the fuzzy set. There is no standard method of choosing the proper shape of the fuzzy sets of the control variables. Trial and error methods are usually exercised (Hashmi et al., 2003). In this model an equal sided triangular shape membership function is selected for both inputs BHN and DOC and for the cutting speed as shown in Figures 2-5. As for the feed rate, an unequal sided triangular shape (Figures 6 - 8) was chosen because of the variation of the feed rate for different values of depth of cut (1-8) mm and 16 mm with their corresponding hardness 85-175 and 175-275 respectively, for the types of cutting tools listed in Table 2.

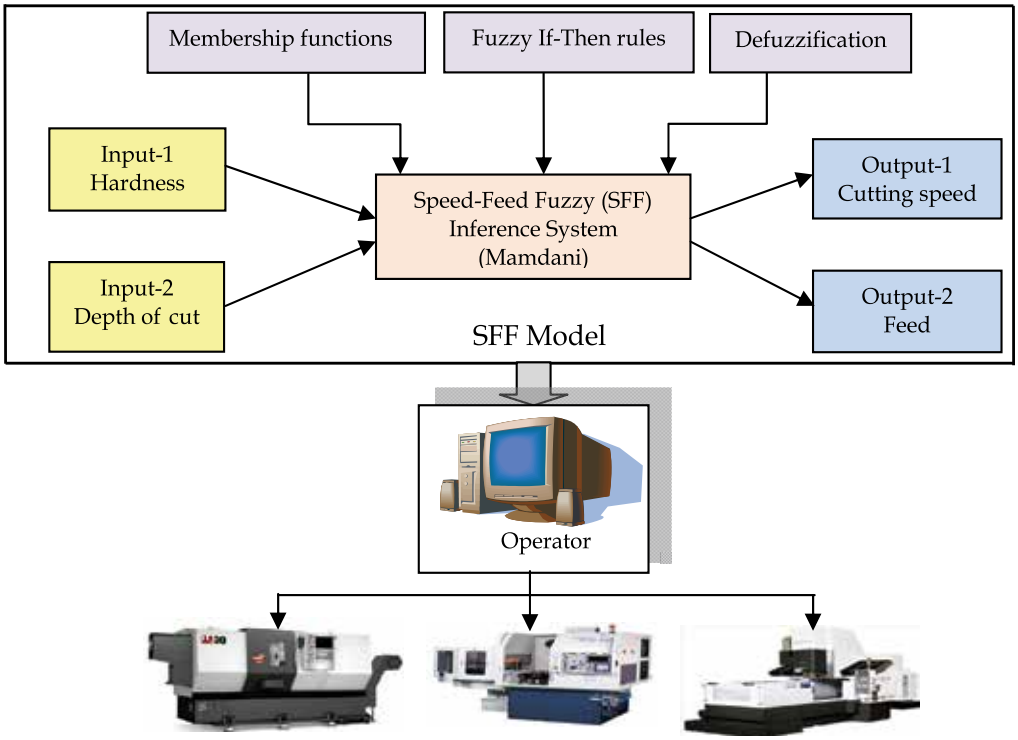


Fig. 1a. Structure of SFF model

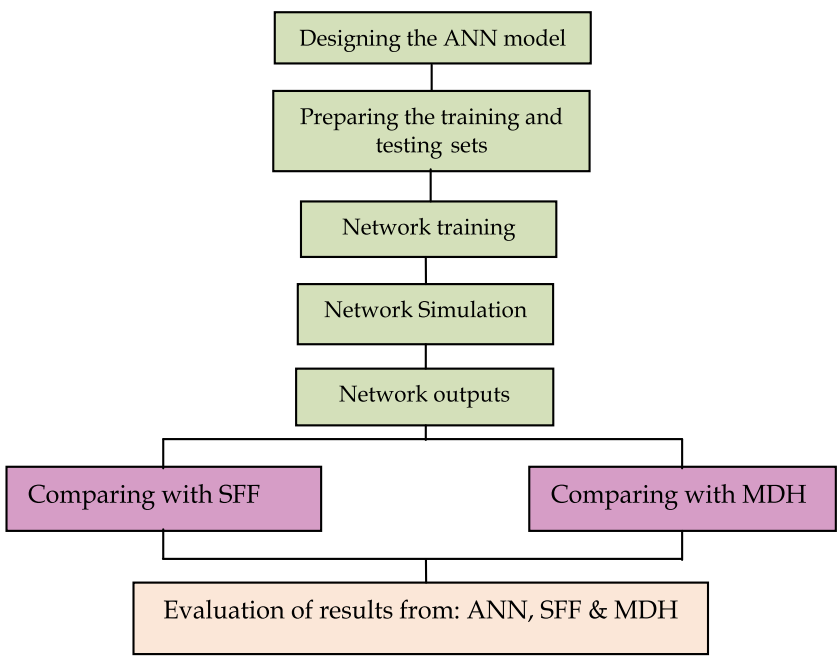


Fig. 1b. Structure of ANN model

Inputs		Outputs	
Material hardness (BHN)	Depth of cut (mm)	Cutting speed (m/min)	Feed rate (mm/rev)
Very soft (VS)	Very shallow (VSH)	Very slow (VSL)	Very slow (VSLO)
Soft (S)	Shallow (SH)	Slow (SL)	Slow (SLO)
Medium (M)	Medium (M)	Medium slow (MSL)	Medium (M)
Medium hard (MH)	Medium deep (MD)	Medium high (MHI)	Medium fast (MFA)
Hard (H)	Deep (D)	High (HI)	Fast (FA)
Very hard (VH)	Very deep (VD)	Very high (VHI)	Very fast (VFA)

Table 1. Fuzzy expressions for inputs and outputs

Condition	Material hardness BHN	Depth of cut mm	High speed steel tool Brazed (S4, S5)		Carbide tool Uncoated Indexible (ISO P10-P40)		Carbide tool Uncoated (ISO P10-P40)		Carbide tool Coated (ISO CP10-CP30)	
			Speed m/min	Feed mm/rev	Speed m/min	Feed mm/rev	Speed m/min	Feed mm/rev	Speed m/min	Feed mm/rev
-Material: Wrought carbon steels.										
Hot rolled	85-125	1	56	0.18	165	0.18	215	0.18	320	0.18
Normalized		4	44	0.40	135	0.50	165	0.50	215	0.40
Annealed		8	35	0.50	105	0.75	130	0.75	170	0.50
Cold drawn		16	27	0.75	81	1.00	100	1.00	-	-
Hot rolled	125-175	1	46	0.18	150	0.18	195	0.18	290	0.18
Normalized		4	38	0.40	125	0.50	150	0.50	190	0.40
Annealed		8	30	0.50	100	0.75	120	0.75	150	0.50
Cold drawn		16	24	0.75	75	1.00	95	1.00	-	-
Hot rolled	175-225	1	44	0.18	140	0.18	175	0.18	260	0.18
Normalized		4	35	0.40	115	0.50	135	0.50	170	0.40
Annealed		8	29	0.50	90	0.75	100	0.75	135	0.50
Cold drawn		16	23	1.00	72	1.00	81	1.00	-	-
Hot rolled	225-275	1	38	0.18	125	0.18	155	0.18	230	0.18
Normalized		4	29	0.40	110	0.50	120	0.50	150	0.40
Annealed		8	23	0.50	87	0.75	95	0.75	120	0.50
Cold drawn		16	18	1.00	67	1.00	73	1.00	-	-

Table 2. Machining parameters for workpiece-tool combination, turning process.

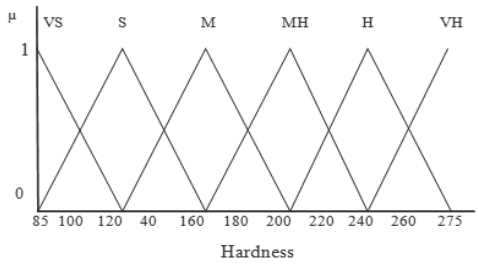


Fig. 2. Hardness membership function

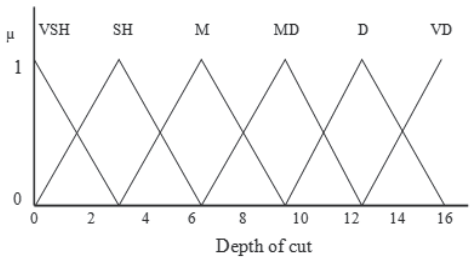


Fig. 3. Depth of cut membership function

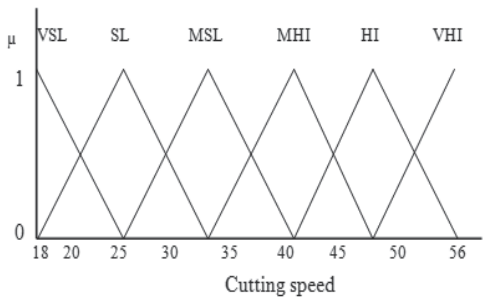


Fig. 4. Speed membership function for HSS tool

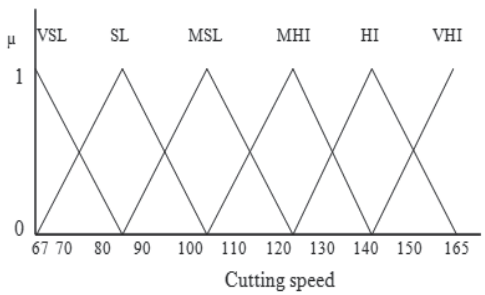


Fig. 5. Speed membership function for carbide tool

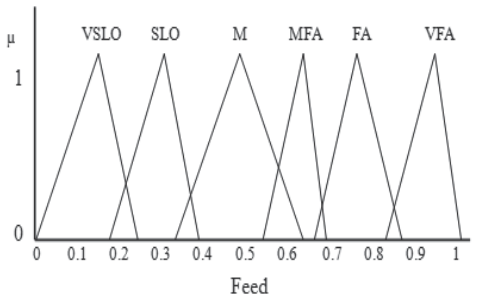


Fig. 6. Feed membership function (BHN=85-175,HSS)

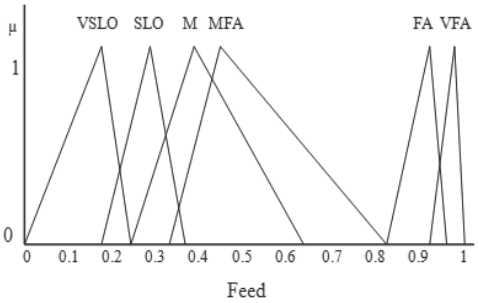


Fig. 7. Feed membership function (BHN=175-275, HSS)

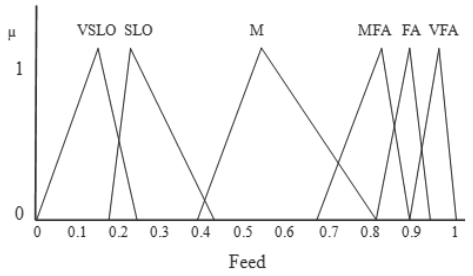


Fig. 8. Feed membership function for carbide tool

2.4 Fuzzy rules

The point of fuzzy logic is to map an input space to an output space and the primary mechanism for doing this is a set of IF-THEN rules with the application of fuzzy operator AND or OR. These if-then rules are used to formulate the conditional statements that comprise fuzzy logic. By using the rules, then the fuzzy inference system (FIS) formulates the mapping form. Mamdani's fuzzy inference system, which is used in this work, is the most commonly seen fuzzy methodology (The MathWorks, Inc., 2009). The relationship between the input variables and output variables is characterized by if-then rules defined based on experimental, expert and engineering knowledge (Yilmaz et al., 2006). The two common methods for the FIS engine are Max-Min method and Max-Product method. The difference between them is the aggregation of the rules. The first use truncation and the last use multiplication of the output

fuzzy set. Both methods are tested and the Max-Min method gives more accurate results, therefore, it is used in all calculations in the fuzzy system.
In this study, there are two input variables hardness and depth of cut each of six fuzzy sets, and then the fuzzy system of a minimum of 6 x 6 = 36 rules can be defined. Table 3 shows a part of the rules in linguistic form. By using these rules the input-output variables in a network representation can be drawn as in Figs. 9 and 10.

Rule 1: IF hardness is very soft AND depth of cut is very shallow THEN speed is very high and feed is very slow.
Rule 2: IF hardness is very soft AND depth of cut is shallow THEN speed is very high and feed is slow.
Rule 3: IF hardness is very soft AND depth of cut is medium THEN speed is medium high and feed is medium.
Rule 4: IF hardness is very soft AND depth of cut is medium deep THEN speed is medium slow and feed is medium.
.
.
.
.
.
.
.
.
Rule 35: IF hardness is very hard AND depth of cut is deep THEN speed is very slow and feed is very fast.
Rule 36: IF hardness is very hard AND depth of cut is very deep THEN speed is very slow and very fast.

Table 3. Part of fuzzy rules in linguistic form.

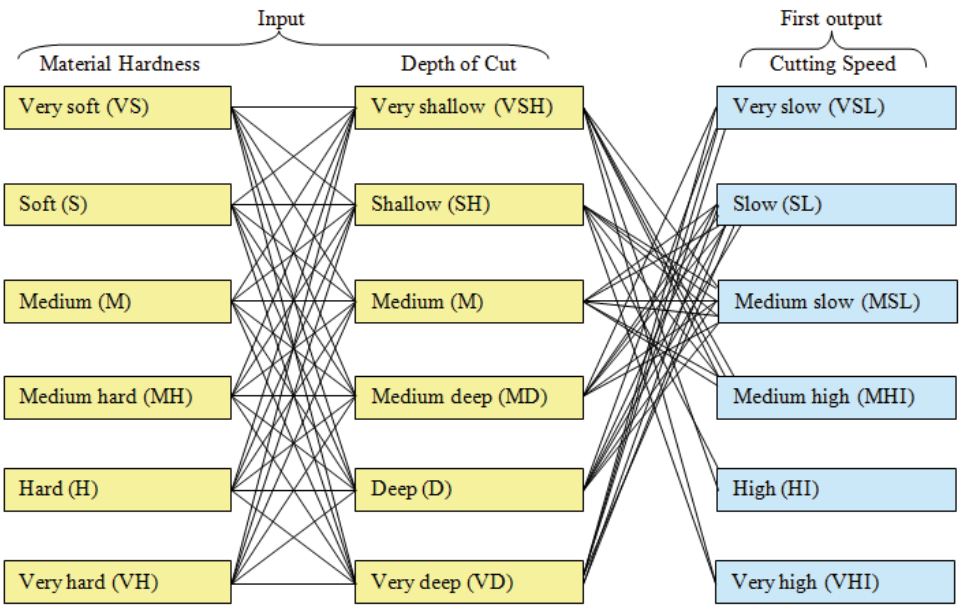


Fig. 9. Network representation for the first output- cutting speed.

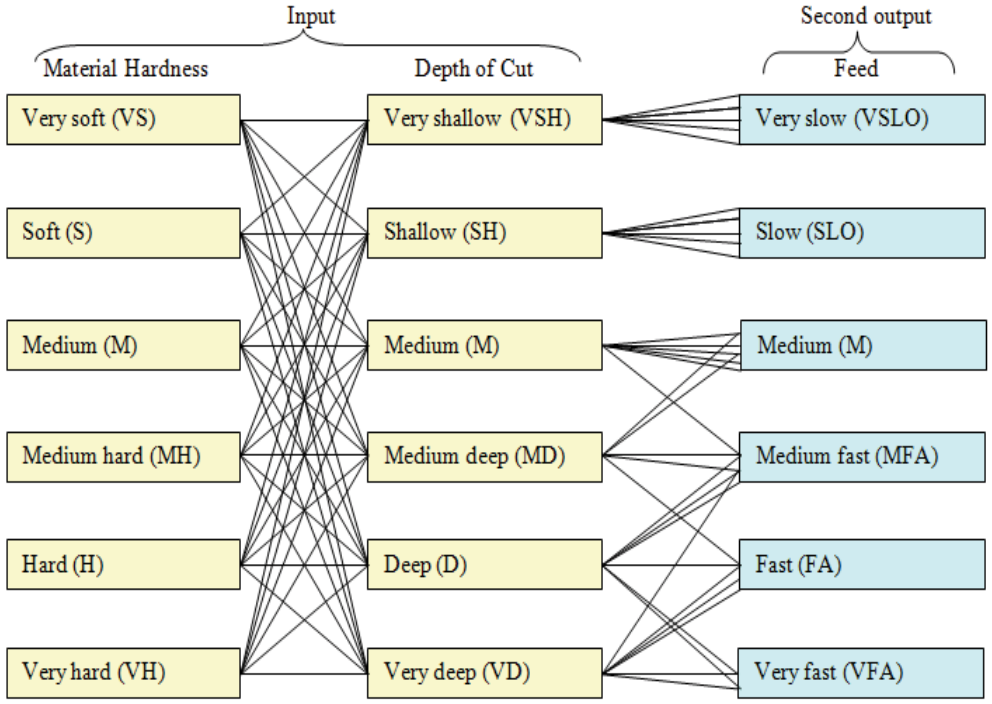


Fig. 10. Network representation for the second output- feed.

2.5 Defuzzification

Defuzzification is the process of converting the fuzzy quantities to crisp quantities. There are several methods used for defuzzifying the fuzzy output functions: the centroid method, the centre of sums, the centre of largest area, the max-membership function, the mean-max membership function, the weighted average method, and the first of maxima or the last of maxima. The selected defuzzification method is significantly affecting the accuracy and speed of the fuzzy algorithm. The centroid method provides more linear results by taking the union of the output of each fuzzy rule (Arghavani et al., 2001; Sivanandam et al., 2007) and this method is adopted in this study.

3. Artificial Neural Network (ANN) model

Neural networks attempt to model human intuition by simulating the physical process upon which intuition is based, that is, by simulating the process of adaptive biological learning. It learns through experience, and is able to continue learning as the problem environment changes (Kim & Park, 1997).

A typical ANN is comprised of several layers of interconnected neurons, each of which is connected to other neurons in the ensuing layer. Data is presented to the neural network via an input layer, while an output layer holds the response of the network to the input. One or more hidden layers may exist between the input layer and the output layer. All hidden and output neurons process their inputs by multiplying each input by its weight, summing the

product, and then processing the sum using a non-linear transfer function to generate a result (Chau, 2006).

The most commonly used approach to ANN learning is the feed-forward back propagation algorithm. The parameters of the model such as the choice of input nodes, number of hidden layers, number of hidden nodes (in each hidden layer), and the form of transfer functions, are problem dependent and often require trial and error to find the best model for a particular application (Ghiassi & Saidene, 2005).

There is no exact rule to decide the number of the hidden layers. There are four methods of selecting the number of hidden nodes (NHN) (Kuo et al., 2002; Yazgan et al., 2009). The four methods are dependent on: the number of input nodes (IN), the number of output nodes (ON), and the number of samples (SN):

$$NHN\ 1 = (IN \times ON)^{1/2} \tag{1}$$

$$NHN\ 2 = \tfrac{1}{2}(IN + ON) \tag{2}$$

$$NHN\ 3 = \tfrac{1}{2}(IN + ON) + (SN)^{1/2} \tag{3}$$

$$NHN\ 4 = 2(IN) \tag{4}$$

The ANN in this study (Fig.11) uses feed-forward back-propagation algorithm. It is composed of two neurons for the two inputs material hardness and depth of cut. The outputs from the neural network are speed and feed; therefore two output neurons are required.

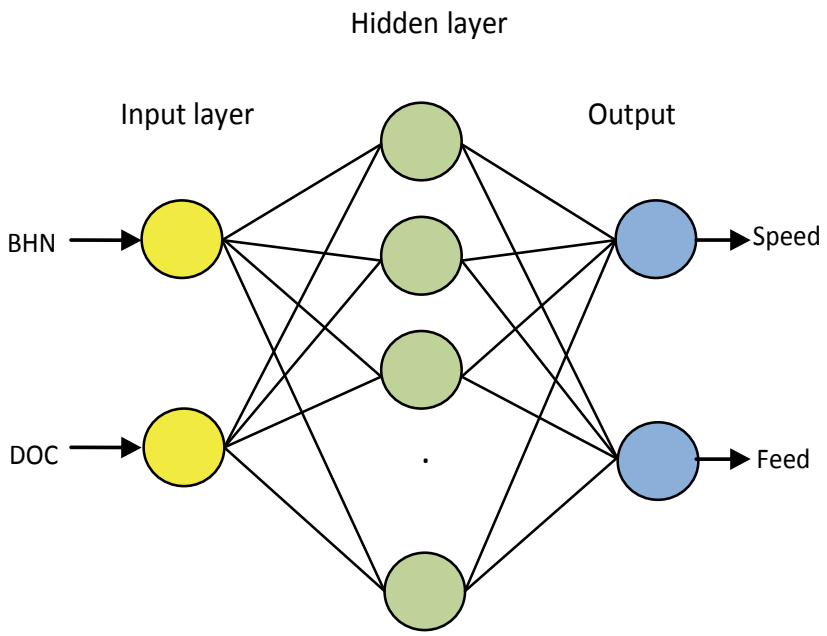


Fig. 11. Neural network structure for machining parameters

4. Results and discussion

Both SFF-ANN are used to predict optimum machining parameters using data extracted from the Machining Data Handbook (MDH) (Table 2).
A user-friendly viewer of the SFF model is shown in Fig. 12 enabling an easy and time saving way for operator for interring the inputs and getting the outputs.

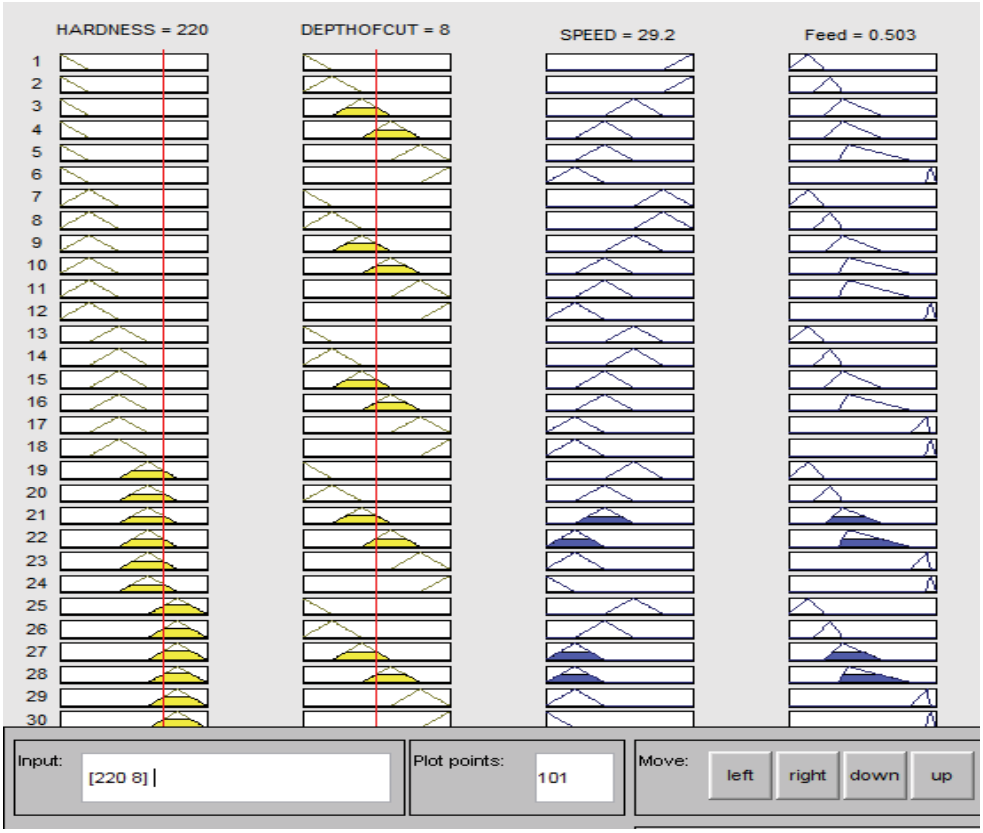


Fig. 12. User-friendly viewer for the SFF model (from MATLAB)

The viewer shown in Fig.12 is used to generate the input-output samples. The values are tabulated in Tables 4 and 5. The tables show the validation of the predicted values of cutting speed and feed found by the SFF model with the Machining Data Handbook. Seventy two different values of wrought carbon steel hardness from (85-275) BHN and depth of cut from (1-16) mm were selected for this comparison. For demonstration purpose two tool types are used: high speed steel (HSS) tool and uncoated brazed carbide (Carbide) tool. The SFF model is applied to obtain the outputs speed and feed and the values are then compared. The absolute error percentage is calculated for each value and the mean absolute error percentages are obtained for the 36 samples. The mean error percentage is almost 7% for speed and 4% for feed when using high speed steel tool and for carbide tool is almost 8% for speed and 7% for feed (Table 6). In order to get better results, the density of the selected samples can be increased.

No.	Material hardness (BHN)	Depth of cut (mm)	Cutting speed (m/min) MDH (Table 2)	SFF model	Abs. error (%)	Feed (mm/rev) MDH (Table 2)	SFF model	Abs. error (%)
1	85	1	56	53.4	4.6429	0.18	0.171	5.0000
2	85	4	44	47.5	7.9545	0.4	0.361	9.7500
3	85	8	35	37	5.7143	0.5	0.4680	6.4000
4	85	16	27	25.6	5.1852	0.75	0.7540	0.5333
5	105	1	56	49.3	11.9643	0.18	0.1760	2.2222
6	105	4	44	46.8	6.3636	0.4	0.37	7.5000
7	105	8	35	37	5.7143	0.5	0.5050	1.0000
8	105	16	27	25.6	5.1852	0.75	0.7550	0.6667
9	120	1	56	48.4	13.5714	0.18	0.171	5.0000
10	120	4	44	46.2	5.0000	0.4	0.3610	9.7500
11	120	8	35	37	5.7143	0.5	0.5050	1.0000
12	120	16	27	25.6	5.1852	0.75	0.7540	0.5333
13	145	1	46	44.1	4.1304	0.18	0.1740	3.3333
14	145	4	38	41.8	10.0000	0.4	0.3670	8.2500
15	145	8	30	32.8	9.3333	0.5	0.5010	0.2000
16	145	16	24	25.6	6.6667	0.75	0.7550	0.6667
17	180	1	44	37.8	14.0909	0.18	0.1770	1.6667
18	180	4	35	37	5.7143	0.4	0.3680	8.0000
19	180	8	29	29.4	1.3793	0.5	0.5030	0.6000
20	180	16	23	24.6	6.9565	1	0.9630	3.7000
21	190	1	44	38.2	13.1818	0.18	0.1710	5.0000
22	190	4	35	35.3	0.8571	0.4	0.3580	10.5000
23	190	8	29	29.4	1.3793	0.5	0.5030	0.6000
24	190	16	23	23.1	0.4348	1	0.9630	3.7000
25	220	1	44	37.9	13.8636	0.18	0.1750	2.7778
26	220	4	35	30.7	12.2857	0.4	0.3650	8.7500
27	220	8	29	29.2	0.6897	0.5	0.5030	0.6000
28	220	16	23	20.8	9.5652	1	0.9630	3.7000
29	245	1	38	38.2	0.5263	0.18	0.1710	5.0000
30	245	4	29	31	6.8966	0.4	0.3580	10.5000
31	245	8	23	25.3	10.0000	0.5	0.5030	0.6000
32	245	16	18	20.5	13.8889	1	0.9630	3.7000
33	265	1	38	35.5	6.5789	0.18	0.1710	5.0000
34	265	4	29	31	6.8966	0.4	0.3580	10.5000
35	265	8	23	24.6	6.9565	0.5	0.5030	0.6000
36	265	16	18	20.6	14.4444	1	0.9630	3.7000

Table 4. Comparison of the results from SFF model with MDH for high speed steel tool

No.	Material hardness (BHN)	Depth of cut (mm)	Cutting speed (m/min) MDH (Table 2)	SFF model	Abs. error (%)	Feed (mm/rev) MDH (Table 2)	SFF model	Abs. error (%)
1	95	1	165	151	8.4848	0.18	0.1700	5.5556
2	95	4	135	143	5.9259	0.5	0.4200	16.000
3	95	8	105	116	10.4762	0.75	0.6750	10.000
4	95	16	81	86.6	6.9136	1	0.9510	4.9000
5	110	1	165	147	10.9091	0.18	0.1710	5.0000
6	110	4	135	141	4.4444	0.5	0.4260	14.800
7	110	8	105	118	12.3810	0.75	0.6750	10.000
8	110	16	81	86.6	6.9136	1	0.9500	5.0000
9	140	1	150	136	9.3333	0.18	0.1760	2.2222
10	140	4	125	130	4.0000	0.5	0.4370	12.600
11	140	8	100	116	16.000	0.75	0.6750	10.000
12	140	16	75	86.6	15.4667	1	0.9490	5.1000
13	195	1	140	119	15.000	0.18	0.1700	5.5556
14	195	4	115	109	5.2174	0.5	0.4200	16.000
15	195	8	90	96.4	7.1111	0.75	0.6750	10.000
16	195	16	72	77	6.9444	1	0.9520	4.8000
17	210	1	140	119	15.000	0.18	0.1700	5.5556
18	210	4	115	100	13.0435	0.5	0.4650	7.0000
19	210	8	90	96.4	7.1111	0.75	0.6980	6.9333
20	210	16	72	73.7	2.3611	1	0.9510	4.9000
21	230	1	125	119	4.8000	0.18	0.17	5.5556
22	230	4	110	101	8.1818	0.5	0.4510	9.8000
23	230	8	87	92	5.7471	0.75	0.7510	0.1333
24	230	16	67	73.4	9.5522	1	0.9510	4.9000
25	240	1	125	119	4.8000	0.18	0.1700	5.5556
26	240	4	110	101	8.1818	0.5	0.4320	13.600
27	240	8	87	86.5	0.5747	0.75	0.7870	4.9333
28	240	16	67	73.3	9.4030	1	0.9520	4.8000
29	255	1	125	116	7.2000	0.18	0.1770	1.6667
30	255	4	110	99.6	9.4545	0.5	0.4840	3.2000
31	255	8	87	84.2	3.2184	0.75	0.7120	5.0667
32	255	16	67	74.3	10.8955	1	0.9480	5.2000
33	270	1	125	110	12.000	0.18	0.1700	5.5556
34	270	4	110	101	8.1818	0.5	0.4420	11.600
35	270	8	87	84	3.4483	0.75	0.6870	8.4000
36	270	16	67	73.3	9.4030	1	0.9520	4.8000

Table 5. Comparison of the results from SFF model with MDH for carbide tool

Mean absolute error percentage (Using HSS tool)
-Speed= 7.19%
-Feed= 4.19%
Mean absolute error percentage (Using carbide tool)
-Speed= 8.29%
-Feed= 7.13%

Table 6. Mean absolute error using 36 samples

Figures 13-16 show the results from Tables 4 and 5 in graphical representation. From these figures it can be seen that the fuzzy cutting speed and feed obtained by the SFF model lie close to the recommended values from the Machining Data Handbook.

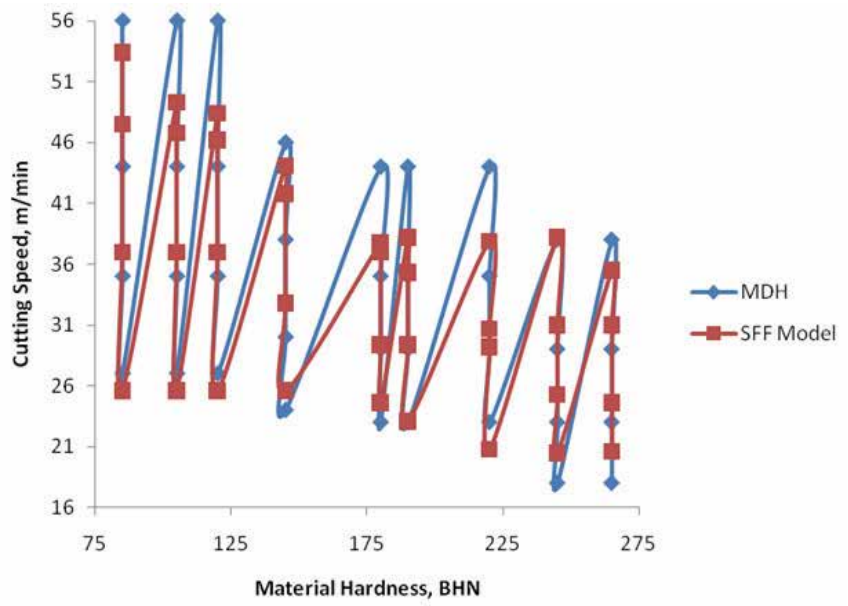


Fig. 13. Cutting speed for high speed steel

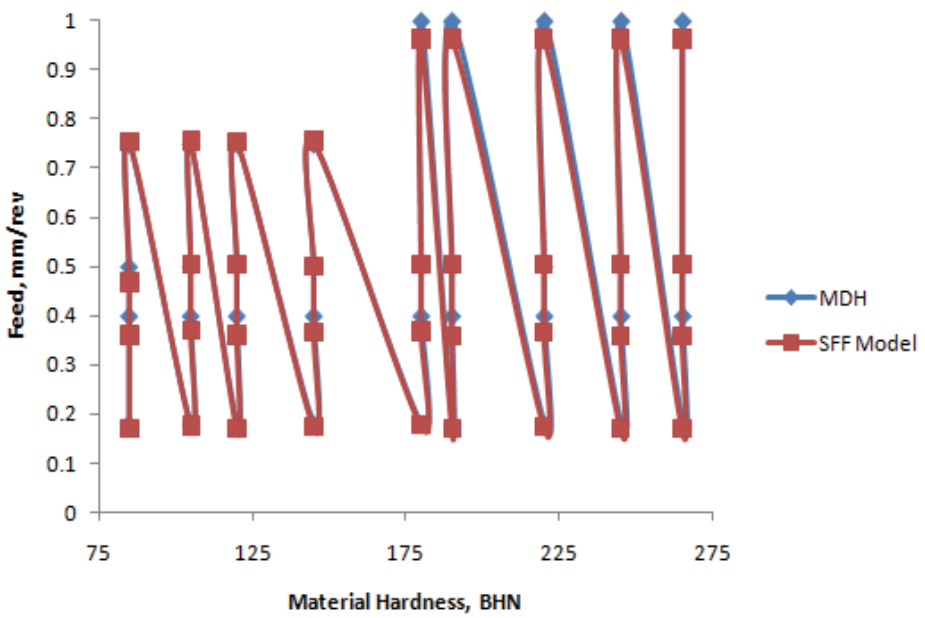


Fig. 14. Feed for high speed steel

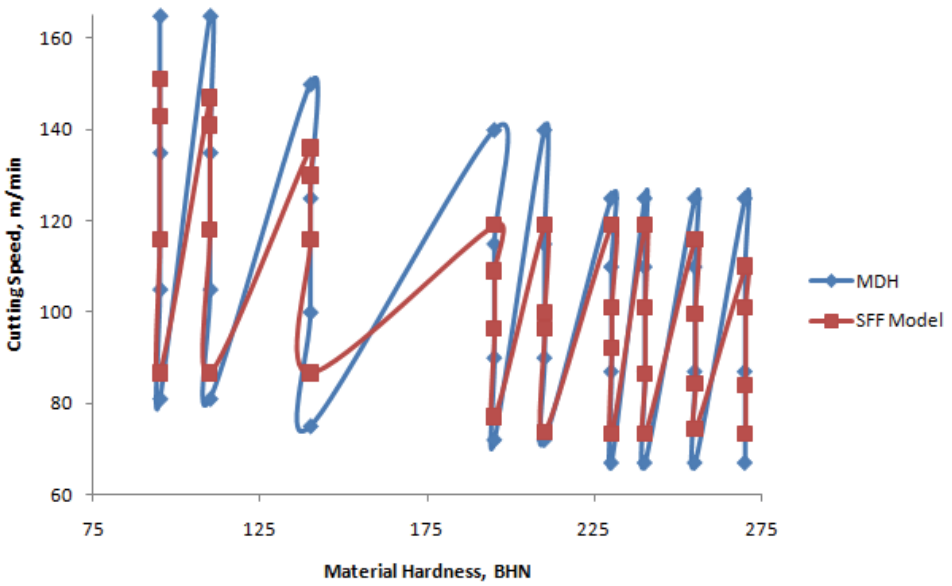


Fig. 15. Cutting speed for carbide tool

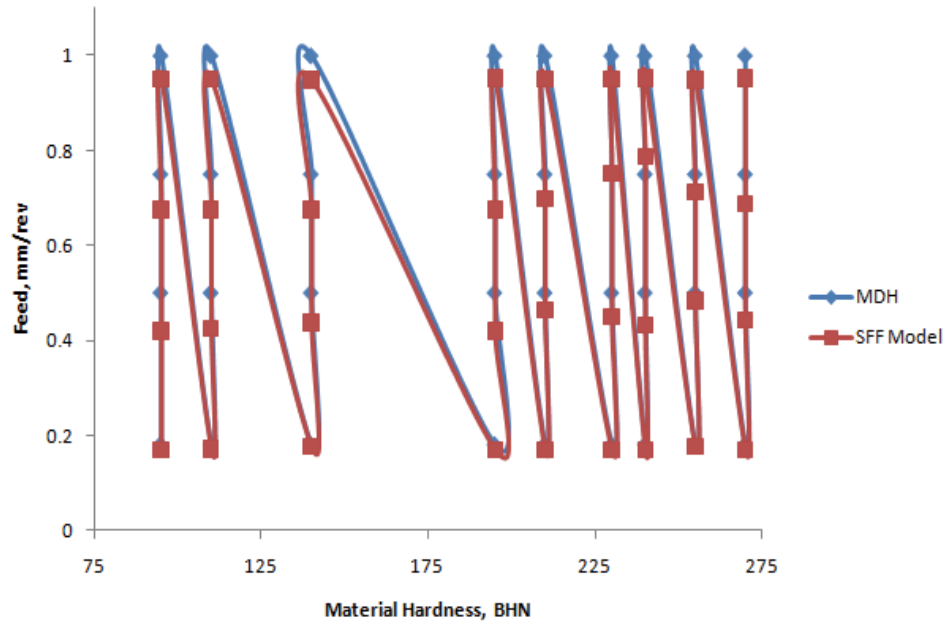


Fig. 16. Feed for carbide tool

The ANN model is composed of two input neurons, material hardness and depth of cut, and two output neurons speed and feed. The values of inputs and outputs are not of the same scale. So, all data are normalized. Tables 7 and 8 contain a set of 18 training and 18 testing samples in normalized form for HSS tool and Carbide tool respectively.

No.	Input 1 Hardness	Input 2 Depth of cut	Output 1 Speed	Output 2 Feed	No.	Input 1 Hardness	Input 2 Depth of cut	Output 1 Speed	Output 2 Feed
Training set					Testing set				
1	0.0137	0.0038	0.0436	0.0100	19	0.0289	0.0307	0.0240	0.0293
2	0.0137	0.0153	0.0388	0.0210	20	0.0289	0.0613	0.0201	0.0562
3	0.0137	0.0307	0.0302	0.0273	21	0.0305	0.0038	0.0312	0.0100
4	0.0137	0.0613	0.0209	0.0440	22	0.0305	0.0153	0.0288	0.0209
5	0.0169	0.0038	0.0403	0.0103	23	0.0305	0.0307	0.0240	0.0293
6	0.0169	0.0153	0.0382	0.0216	24	0.0305	0.0613	0.0189	0.0562
7	0.0169	0.0307	0.0302	0.0294	25	0.0354	0.0038	0.0310	0.0102
8	0.0169	0.0613	0.0209	0.0440	26	0.0354	0.0153	0.0251	0.0213
9	0.0193	0.0038	0.0395	0.0100	27	0.0354	0.0307	0.0239	0.0293
10	0.0193	0.0153	0.0378	0.0210	28	0.0354	0.0613	0.0170	0.0562
11	0.0193	0.0307	0.0302	0.0294	29	0.0394	0.0038	0.0312	0.0100
12	0.0193	0.0613	0.0209	0.0440	30	0.0394	0.0153	0.0253	0.0209
13	0.0233	0.0038	0.0360	0.0101	31	0.0394	0.0307	0.0207	0.0293
14	0.0233	0.0153	0.0342	0.0214	32	0.0394	0.0613	0.0168	0.0562
15	0.0233	0.0307	0.0268	0.0292	33	0.0426	0.0038	0.0290	0.0100
16	0.0233	0.0613	0.0209	0.0440	34	0.0426	0.0153	0.0253	0.0209
17	0.0289	0.0038	0.0309	0.0103	35	0.0426	0.0307	0.0201	0.0293
18	0.0289	0.0153	0.0302	0.0215	36	0.0426	0.0613	0.0168	0.0562

Table 7. Training-testing data for high speed steel tool

No.	Input 1 Hardness	Input 2 Depth of cut	Output 1 Speed	Output 2 Feed	No.	Input 1 Hardness	Input 2 Depth of cut	Output 1 Speed	Output 2 Feed
Training set					Testing set				
1	0.0136	0.0038	0.0402	0.0083	19	0.0301	0.0307	0.0257	0.0342
2	0.0136	0.0153	0.0381	0.0206	20	0.0301	0.0613	0.0196	0.0466
3	0.0136	0.0307	0.0309	0.0331	21	0.0330	0.0038	0.0317	0.0083
4	0.0136	0.0613	0.0231	0.0466	22	0.0330	0.0153	0.0269	0.0221
5	0.0158	0.0038	0.0391	0.0084	23	0.0330	0.0307	0.0245	0.0368
6	0.0158	0.0153	0.0375	0.0209	24	0.0330	0.0613	0.0195	0.0466
7	0.0158	0.0307	0.0314	0.0331	25	0.0344	0.0038	0.0317	0.0083
8	0.0158	0.0613	0.0231	0.0465	26	0.0344	0.0153	0.0269	0.0212
9	0.0201	0.0038	0.0362	0.0086	27	0.0344	0.0307	0.0230	0.0386
10	0.0201	0.0153	0.0346	0.0214	28	0.0344	0.0613	0.0195	0.0466
11	0.0201	0.0307	0.0309	0.0331	29	0.0365	0.0038	0.0309	0.0087
12	0.0201	0.0613	0.0231	0.0465	30	0.0365	0.0153	0.0265	0.0237
13	0.0279	0.0038	0.0317	0.0083	31	0.0365	0.0307	0.0224	0.0349
14	0.0279	0.0153	0.0290	0.0206	32	0.0365	0.0613	0.0198	0.0464
15	0.0279	0.0307	0.0257	0.0331	33	0.0387	0.0038	0.0293	0.0083
16	0.0279	0.0613	0.0205	0.0466	34	0.0387	0.0153	0.0269	0.0217
17	0.0301	0.0038	0.0317	0.0083	35	0.0387	0.0307	0.0224	0.0337
18	0.0301	0.0153	0.0266	0.0228	36	0.0387	0.0613	0.0195	0.0466

Table 8. Training-testing data for carbide tool

The first half of the data in each table is used for training the network with different number of hidden nodes: two, four, and eight, extracted using the equations (1-4). The models are trained with different training parameters and different activation functions as shown in Tables 9 and 10. The mean square error (MSE) value is used as the stop criteria.

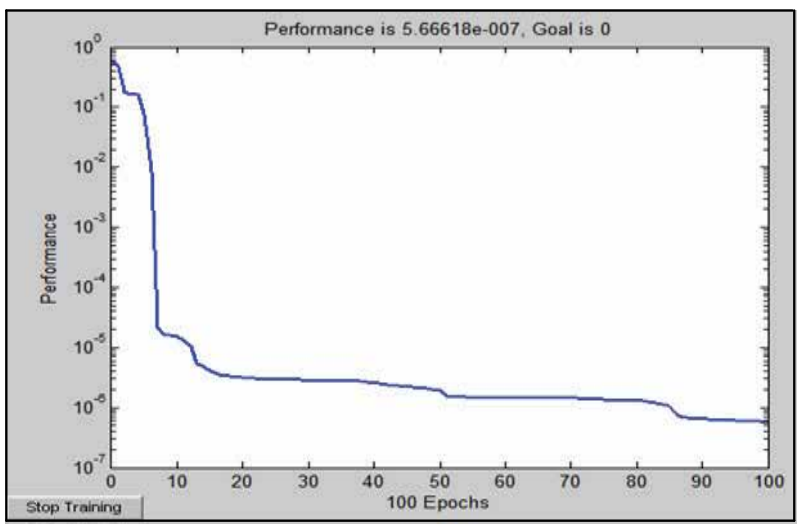
Input Nodes	Hidden Nodes	Output Nodes	Training Function	Transfer Function	Epochs	Performance
2	2	2	TRAINLIM	TANSIG	150	3.61807e-006
2	4	2	TRAINLIM	TANSIG	150	3.43611e-006
2	8	2	TRAINLIM	TANSIG	100	5.66618e-007
2	2	2	TRAINLIM	SIGMOID	200	3.23253e-006
2	4	2	TRAINLIM	SIGMOID	350	3.78049e-007
2	8	2	TRAINLIM	SIGMOID	350	3.117 65e-007

Table 9. ANN model parameters for HSS tool

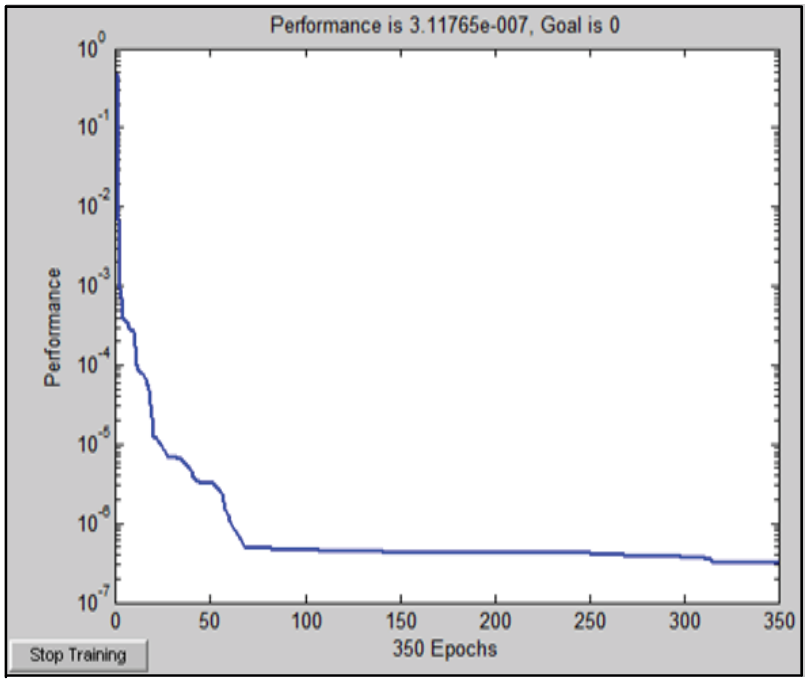
Input Nodes	Hidden Nodes	Output Nodes	Training Function	Transfer Function	Epochs	Performance
2	2	2	TRAINLIM	TANSIG	350	9.96923e-007
2	4	2	TRAINLIM	TANSIG	250	8.26549e-007
2	8	2	TRAINLIM	TANSIG	190	2.19803e-007
2	2	2	TRAINLIM	SIGMOID	250	9.87903e-007
2	4	2	TRAINLIM	SIGMOID	236	5.12694e-007
2	8	2	TRAINLIM	SIGMOID	145	1.325 60e-007

Table 10. ANN model parameters for carbide tool.

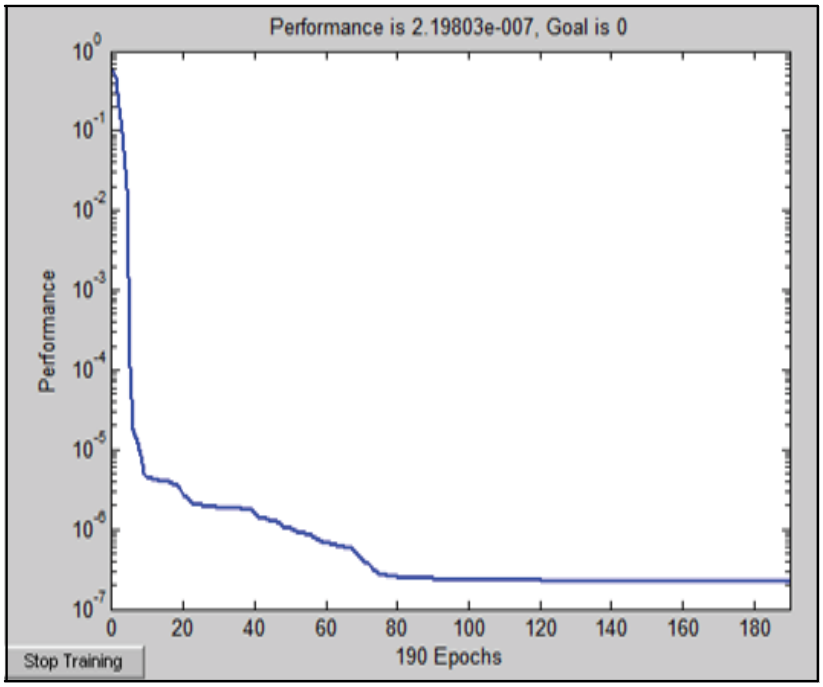
The trained neural network was tested based on the second half of the input-output samples in Tables 7 and 8. The performance of the best training processes is shown in Fig.17. Fig.18 shows the architecture of the best feed forward neural network (2-8-2) model.



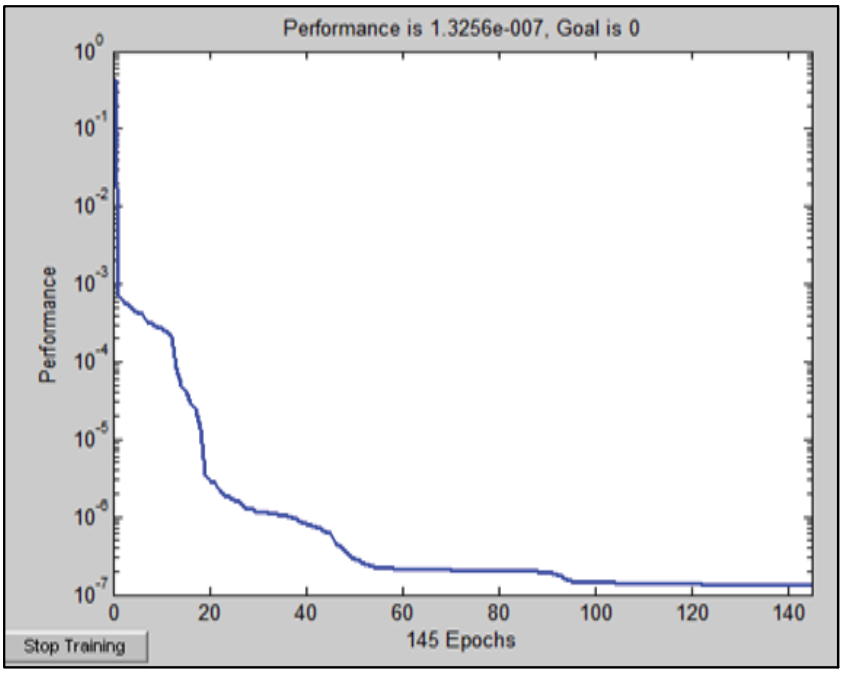
(a) 2-8-2 ANN model using Tansig function for HSS tool



(b) 2-8-2 ANN model using Sigmoid function for HSS tool



(c) 2-8-2 ANN model using Tansig function for carbide tool



(d) 2-8-2 ANN model using Sigmoid function for carbide tool

Fig. 17. Performance curves for best tested ANN models

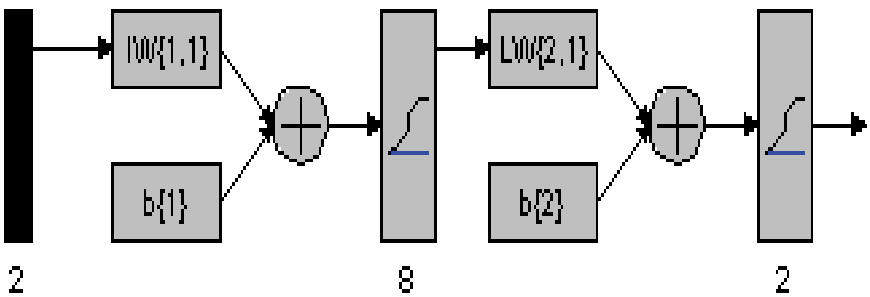


Fig. 18. Architecture of 2-8-2 ANN model (from MATLAB)

From Tables 9 and 10 and Fig.17 (b) and (d), it can be seen that the 2-8-2 ANN model gives a small error. The error is 3.11765e-007 for high speed steel and 1.3256e-007 for carbide tool and the trained network is considered valid.

The ANN model is simulated based on the test data set (19-36) from Tables 7 and 8. The outputs from the network simulation are shown in Tables 11 and 12. These tables show the comparison between the values obtained by SFF and the values predicted by ANN for the two types of the tools used in the demonstration. From the tables it can be seen that the obtained values closely matches the predicted values of the ANN model.

	Output- Speed			Output- Feed		
No.	SFF model	ANN model	Difference	SFF model	ANN model	Difference
19	0.0239	0.0240	0.0001	0.0293	0.0293	0
20	0.0199	0.0201	0.0002	0.0561	0.0562	0.0001
21	0.0310	0.0312	0.0002	0.0100	0.0100	0
22	0.0287	0.0288	0.0001	0.0212	0.0209	-0.0003
23	0.0233	0.0240	0.0007	0.0294	0.0293	-0.0001
24	0.0192	0.0189	-0.0003	0.0562	0.0562	0
25	0.0302	0.0310	0.0008	0.0101	0.0102	0.0001
26	0.0277	0.0251	-0.0026	0.0207	0.0213	0.0006
27	0.0225	0.0239	0.0014	0.0293	0.0293	0
28	0.0169	0.0170	0.0001	0.0564	0.0562	-0.0002
29	0.0297	0.0312	0.0015	0.0102	0.0100	-0.0002
30	0.0268	0.0253	0.0015	0.0204	0.0209	0.0005
31	0.0210	0.0207	-0.0003	0.0293	0.0293	0
32	0.0166	0.0168	0.0002	0.0560	0.0562	0.0002
33	0.0291	0.0290	-0.0001	0.0104	0.0100	-0.0004
34	0.0248	0.0253	0.0005	0.0209	0.0209	0
35	0.0202	0.0201	-0.0001	0.0293	0.0293	0
36	0.0165	0.0168	0.0003	0.0561	0.0562	0.0001

Table 11. Comparison of outputs for HSS tool

	Output- Speed			Output- Feed		
No.	SFF model	ANN model	Difference	SFF model	ANN model	Difference
19	0.0257	0.0255	0.0002	0.0342	0.0340	0.0002
20	0.0196	0.0196	0	0.0466	0.0466	0
21	0.0317	0.0314	0.0003	0.0083	0.0084	-0.0001
22	0.0269	0.0269	0	0.0221	0.0221	0
23	0.0245	0.0239	0.0006	0.0368	0.0365	0.0003
24	0.0195	0.0196	-0.0001	0.0466	0.0466	0
25	0.0317	0.0310	0.0007	0.0083	0.0084	-0.0001
26	0.0269	0.0267	0.0002	0.0212	0.0224	-0.0012
27	0.0230	0.0228	0.0002	0.0386	0.0385	0.0001
28	0.0195	0.0196	-0.0001	0.0466	0.0465	0.0001
29	0.0309	0.0305	0.0004	0.0087	0.0083	0.0004
30	0.0265	0.0264	0.0001	0.0237	0.0227	0.001
31	0.0224	0.0226	-0.0002	0.0349	0.0349	0
32	0.0198	0.0196	0.0002	0.0464	0.0465	-0.0001
33	0.0293	0.0299	-0.0006	0.0083	0.0082	0.0001
34	0.0269	0.0267	0.0002	0.0217	0.0216	0.0001
35	0.0224	0.0222	0.0002	0.0337	0.0336	0.0001
36	0.0195	0.0195	0	0.0466	0.0465	0.0001

Table 12. Comparison of outputs for carbide tool

No.	Input 1 Hardness	Input 2 Depth of cut	Output 1 Speed	Output 2 Feed	No.	Input 1 Hardness	Input 2 Depth of cut	Output 1 Speed	Output 2 Feed
Training set					Testing set				
1	0.0136	0.0038	0.0454	0.0102	19	0.0301	0.0307	0.0235	0.0282
2	0.0136	0.0153	0.0357	0.0226	20	0.0301	0.0613	0.0187	0.0564
3	0.0136	0.0307	0.0284	0.0282	21	0.0330	0.0038	0.0357	0.0102
4	0.0136	0.0613	0.0219	0.0423	22	0.0330	0.0153	0.0284	0.0226
5	0.0158	0.0038	0.0454	0.0102	23	0.0330	0.0307	0.0235	0.0282
6	0.0158	0.0153	0.0357	0.0226	24	0.0330	0.0613	0.0187	0.0564
7	0.0158	0.0307	0.0284	0.0282	25	0.0344	0.0038	0.0357	0.0102
8	0.0158	0.0613	0.0219	0.0423	26	0.0344	0.0153	0.0284	0.0226
9	0.0201	0.0038	0.0454	0.0102	27	0.0344	0.0307	0.0235	0.0282
10	0.0201	0.0153	0.0357	0.0226	28	0.0344	0.0613	0.0187	0.0564
11	0.0201	0.0307	0.0284	0.0282	29	0.0365	0.0038	0.0308	0.0102
12	0.0201	0.0613	0.0219	0.0423	30	0.0365	0.0153	0.0235	0.0226
13	0.0279	0.0038	0.0373	0.0102	31	0.0365	0.0307	0.0187	0.0282
14	0.0279	0.0153	0.0308	0.0226	32	0.0365	0.0613	0.0146	0.0564
15	0.0279	0.0307	0.0243	0.0282	33	0.0387	0.0038	0.0308	0.0102
16	0.0279	0.0613	0.0195	0.0423	34	0.0387	0.0153	0.0235	0.0226
17	0.0301	0.0038	0.0357	0.0102	35	0.0387	0.0307	0.0187	0.0282
18	0.0301	0.0153	0.0284	0.0226	36	0.0387	0.0613	0.0146	0.0564

Table 13. Training-testing data from MDH for high speed steel tool

The performance of the SFF is compared with ANN and MDH using high speed steel tool as a demonstration example (Table 13).
The performance of the best training process using network architecture 2-8-2 with 950 epochs is shown in Fig. 19 where the value is 3.92694e-007.

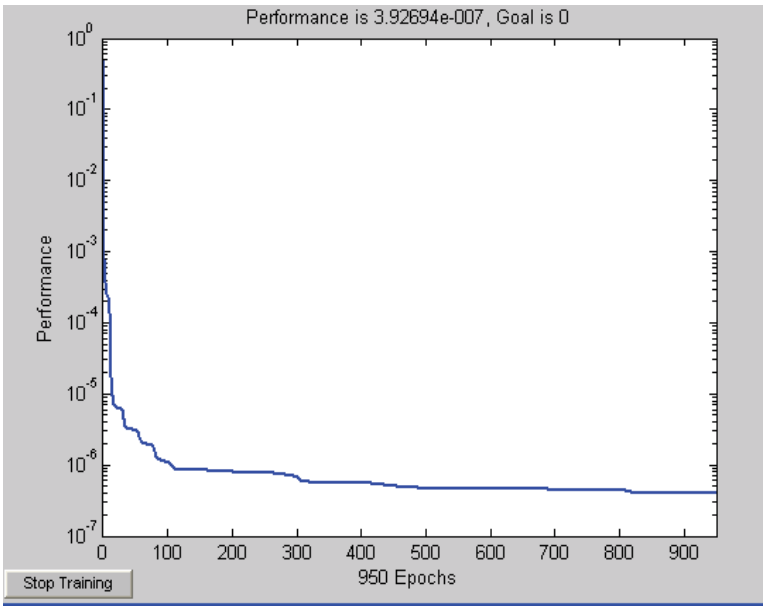


Fig. 19. Performance curve for best tested ANN model

The output from the simulated network using test data set (19-36) from Table 13 is shown in Figures 20 and 21. The Figures show the comparison between the values obtained by SFF model and the predicted values by ANN model and values from MDH.

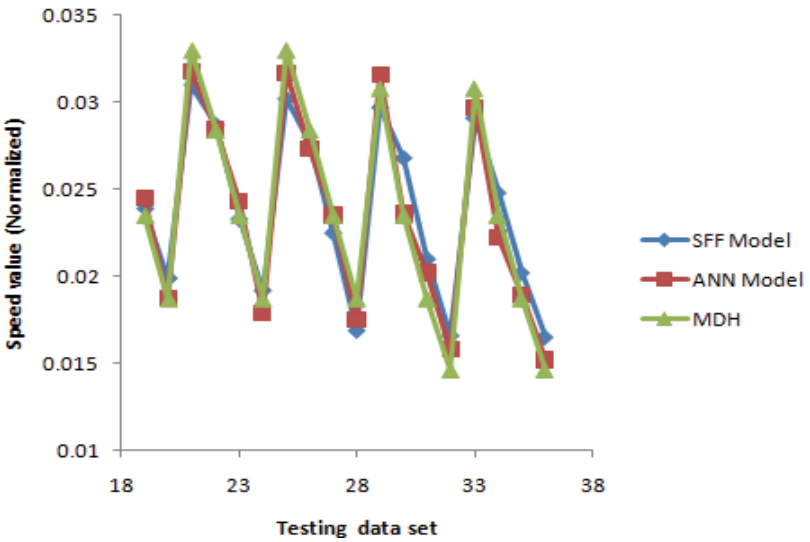


Fig. 20. Comparison of speed values between SFF, ANN and MDH

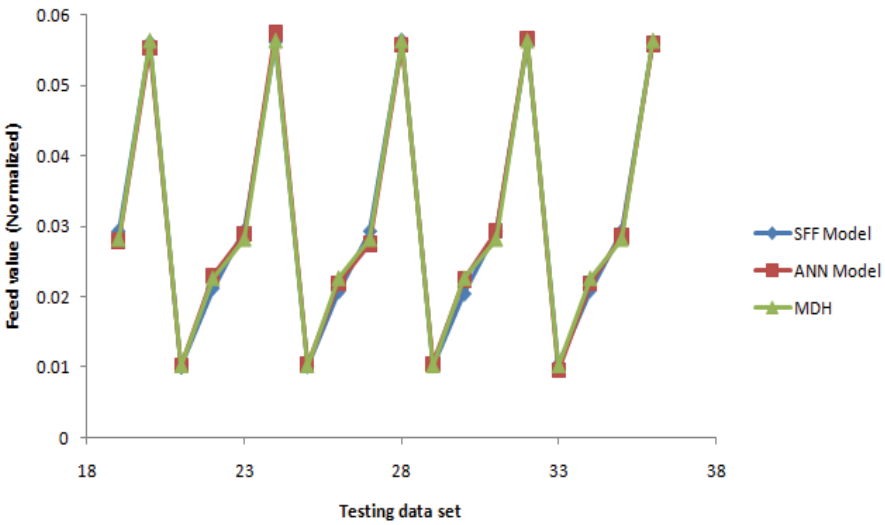


Fig. 21. Comparison of feed values between SFF, ANN and MDH

5. Conclusion

In this study, a fuzzy logic using expert rules and ANN model are used to predict machining parameters.

The fuzzy inference engine used in the model has successfully formulated the input-output mapping enabling an easy and effective approach for selecting optimal machining parameters. ANN was also found to be accurate in predicting the optimal parameters.

Both approaches can be easily expanded to handle more tool-workpiece materials combinations and it is not limited to turning process only and can be used for other machining processes like: milling, drilling, grinding, etc.

However the SFF is more user-friendly and compatible with the automation concept of a flexible and computer integrated manufacturing systems. It allows the operator, even unskilled to find the optimal machining parameters for an efficient machining process that can lead to an improvement of product quality, increase production rates and thus reducing product cost and total manufacturing costs.

6. References

Arghavani, J.; Derenne, M. & Marchand, L. (2001). Fuzzy logic application in gasket selection and sealing performance. *International Journal of Advanced Manufacturing Technology*, Vol. 18, (67-78).

Chau, Kwokwing (2006). A review of the integration of artificial intelligence into coastal modelling. *Journal of Environmental Management*, Vol. 80, (47-57).

Cus, Franci & Zuperl, Uros (2006). Approach to optimization of cutting conditions by using artificial neural networks. *Journal of Materials Processing Technology*, Vol. 173, (281-290)

El Baradie, M.A. (1997). A fuzzy logic model for machining data selection. International Journal of Machine Tools and Manufacture, Vol. 37, No. 9, (1353-1372).

Ghiassi, M. & Saidene, H. (2005). A dynamic architecture for artificial neural networks. *Neurocomputing*, Vol. 63, (397-413).

Hashmi, K.; Graham, I.D. & Mills, B. (2003). Data selection for turning carbon steel using a fuzzy logic approach. *Journal of Materials Processing Technology*, Vol. 135, (44-58).

Hashmi, K.; Graham, I.D. & Mills, B. (2000). Fuzzy logic based Data selection for the drilling process. *Journal of Materials Processing Technology*, Vol. 108, (55-61).

Hashmi, K.; El Baradie, M.A. & Ryan, M. (1999). Fuzzy logic based intelligent selection of machining parameters. *Journal of Materials Processing Technology*, Vol. 94, (94-111).

Hashmi, K.; El Baradie, M.A. & Ryan, M. (1998). Fuzzy logic based intelligent selection of machining parameters. *Computers and Industrial Engineering*, Vol. 35, No. (3-4), (571-574).

Kim, Jae Kyeong & Park, Kyung Sam (1997). Modelling a class of decision problems using artificial neural networks. *Expert Systems with Applications*, Vol. 12, (195-208).

Kuo, R.J.; Chi, S.C. & Kao, S.S. (2002). A decision support system for selecting convenience store location through integration of fuzzy AHP and artificial neural network. *Computers in Industry*, Vol. 47, (199-214).

Lee, B.Y. & Tarng, Y.S. (2000). Cutting parameter selection for maximizing production rate or minimizing production cost in multistage turning operations. *Journal of Materials Processing Technology*, Vol. 105, (61-66).

Liu, T.I.; Singonahalli, J.H. & Iyer, N.R. (1996). Detection of roller bearing defects using expert system and fuzzy logic. *Mechanical Systems and Signal Processing*, Vol. 10, No. 5, (595-614).

Malakooti, B. & Deviprasad, J. (1989). An interactive multiple criteria approach for parameter selection in metal cutting. *Operations Research*, Vol. 37, No. 5, (805-818).

Medsker, Larry R. (1996). Microcomputer applications of hybrid intelligent systems. *Journal of Network and Computer applications*, Vol. 63, (213-234).

Metcut Research Associates Inc. (1980). *Machining Data Handbook*, 3rd edition, Vol. 1 & 2, Cincinnati.

Nian, C.Y.; Yang, W.H. & Tarng, Y.S. (1999). Optimization of turning operations with multiple performance characteristics. *Journal of Materials Processing Technology*, Vol. 95, (90-96).

Park, Kyung Sam & Kim, Soung Hie (1998). Artificial intelligence approaches to determination of CNC machining parameters in manufacturing: a review. *Artificial Intelligence in Engineering*, Vol. 12, (127-134).

Saravanan, R.; Asokan, P. & Vijayakumar, K. (2003). Machining parameters optimization for turning cylindrical stock into a continuous finished profile using genetic algorithm (GA) and simulated annealing (SA). *International Journal of Advanced Manufacturing Technology*, Vol. 21, (1-9).

Singh, Rajiv & Raman, Shivakumar (1992). METEX- an expert system for machining planning. *International Journal of Production Research,* Vol. 30, No. 7, (1501-1516).

Sivanandam, S.N.; Sumathi, S. & Deepa, S.N. (2007). *Introduction to Fuzzy Logic using MATLAB*, Springer, Springer-Verlag Berlin Heidelberg.

The MathWorks, Inc. (2009). *MATLAB Fuzzy Logic Toolbox*, User's guide.

Vitanov, V.I.; Harrison, D.K.; Mincoff, N.H. & Vladimirova, T.V. (1995). An expert system shell for the selection of metal cutting parameters. *Journal of Materials Processing Technology*, Vol. 15, (111-116).

Wong, S.V. & Hamouda, A.M.S. (2003a). The development of an online knowledge-based expert system for machinability data selection. *Knowledge-based Systems*, Vol. 16, (215-229).

Wong, S.V. & Hamouda, A.M.S. (2003b). Machinability data representation with artificial neural network. *Journal of Materials Processing Technology*, Vol. 138, (538-544).

Wong, S.V.; Hamouda, A.M.S. & El Baradie, M.A. (1999). Generalized fuzzy model for metal cutting data selection. *Journal of Materials Processing Technology*, Vol. 89-90, (310-317).

Yazgan, Harun Resit; Boran, Semra & Goztepe, Kerim (2009). An ERP software selection process with artificial neural network based on analytic network process approach. *Expert Systems with Applications*, Vol. 36, (9214-9222).

Yilmaz, Oguzhan; Eyercioglu, Omer & Gindy, Nabil N.Z. (2006). A user-friendly fuzzy based system for the selection of electro discharge machining process parameters. *Journal of Materials Processing Technology*, Vol. 172, (363-371).

Zadeh, L.A. (1965). Fuzzy sets. *Information and Control*, Vol. (8), (338-353).

Zuperl, Uros & Cus, Franci (2003). Optimization of cutting conditions during cutting by using neural networks. *Robotics and Computer Integrated Manufacturing*, Vol. 19, (189-199).

Part 6

Control and Robotic Engineering

17

Artificial Neural Network – Possible Approach to Nonlinear System Control

Jan Mareš, Petr Doležel and Pavel Hrnčiřík
Institute of Chemical Technology, Prague
&University of Pardubice, Pardubice
Czech Republic

1. Introduction

Artificial Neural Networks (ANN) have traditionally enjoyed considerable attention in process control applications. Thus, the paper is focused on real system control design using neural networks. The point is to show whether neural networks bring better performances to nonlinear process control or not.
Artificial Neural Network is nowadays a popular methodology with lots of practical and industrial applications. As introduction, some concrete examples of successful application of ANN can be mentioned, *e.g.* mathematical modeling of bioprocesses [Montague et al., 1994], [Teixeira et al., 2005], prediction models and control of boilers, furnaces and turbines [Lichota et al., 2010] or industrial ANN control of calcinations processes, or iron ore process [Dwarapudi, et al., 2007].
Specifically in our paper, the aim is to explain and describe usage of neural network in the case of nonlinear reactor furnace control.

2. Controlled system

Real system (controlled plant) is a reactor furnace, which is significantly nonlinear system. Furnace is an equipment of the research laboratory of the Department of Physical Chemistry at the University of Pardubice, Czech Republic.
Reactor furnace is used for research of oxidation and reduction qualities of catalyzers under different temperatures by controlled heating of the reactor (where the chemical substance is placed). The temperature profile of the reactor is strictly defined. It is linear increasing up to 800 °C, then keeping the constant value of 800 °C till the end of the experiment. The difference between the setpoint and controlled variable (furnace temperature) has to be less than 10 °C.
The basic premise is so strict, that it is not possible to use standard control techniques as PID controller. Thus, an artificial neural network represents one of the available techniques for overcoming this obstacle.

2.1 System description

Reactor furnace base is a cored cylinder made of insulative material, described in [Mareš et al., 2010a]. On the inner surface there are two heating spirals (powered by voltage 230 V). In

the middle of the cylinder there is a reactor. The reactor temperature is measured by one platinum thermometer (see Figure 1).

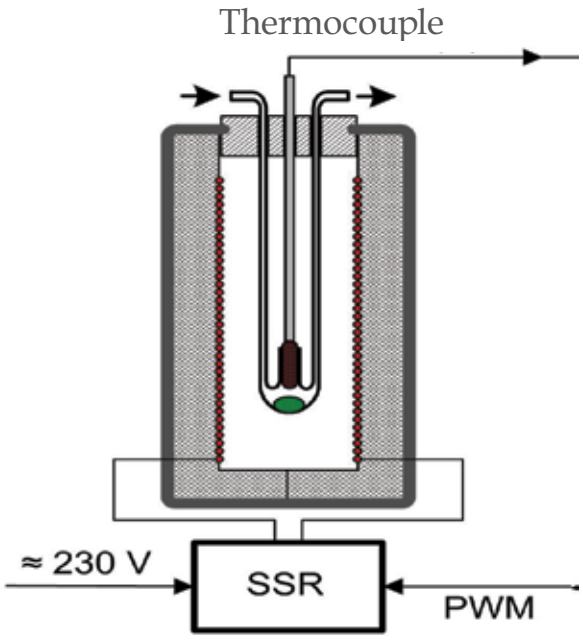


Fig. 1. Reactor furnace chart

The system is a thermal process with two inputs (spiral power and ambient temperature) and one output (reactor temperature). Thus, the controlled variable is the reactor temperature and the manipulated variable is the spiral power with the ambient temperature as measured error. The plant is significantly nonlinear system. Nonlinearity is caused by heat transfer mechanism. When the temperature is low, heat transfer is provided only by conduction. However, when the temperature is high, radiation presents an important transfer principle.

2.2 Nonlinear model

Nonlinear mathematical model of reactor furnace consists of four parts. Differential equations describing isolation, heating spiral, inner space and reactor were derived.
Because variables changes along devices dimension are irrelevant, process behavior can be considered as a lumped system.
Nonlinear mathematical model is possible to describe by equations (1) to (4), more in [Mares et al., 2010a].
Isolation

$$\begin{aligned} &\underbrace{\alpha_{AB}.S_{AB}}_{aAB}.(T_B - T_A) + \underbrace{\alpha_{AC}.S_{AC}}_{aAC}.(T_C - T_A) + S_2\sigma(T_B^4 - T_A^4) = \\ &= \underbrace{\alpha_{AO}.S_{AO}}_{aAO}.(T_A - T_{OK}) + S_3\sigma(T_A^4 - T_{OK}^4) + S_4\sigma(T_A^4 - T_D^4) + \underbrace{m_A.c_A}_{PA}.\frac{dT_A}{dt} \end{aligned} \tag{1}$$

Spiral

$$\frac{E}{1+\beta T_B} = \underbrace{\alpha_{AB}.S_{AB}}_{aAB}.(T_B - T_A) + \underbrace{\alpha_{BC}.S_{BC}}_{aBC}.(T_B - T_C) + S_1.\sigma.(T_B^4 - T_D^4) + \\ + S_2.\sigma.(T_B^4 - T_A^4) + \underbrace{m_B.c_B}_{PB}.\frac{dT_B}{dt} \tag{2}$$

Inner space

$$\underbrace{\alpha_{BC}.S_{BC}}_{aBC}.(T_B - T_C) = \underbrace{\alpha_{CO}.S_{CO}}_{aCO}.(T_C - T_{OK}) + \underbrace{\alpha_{CD}.S_{CD}}_{aCD}.(T_C - T_D) + \\ + \underbrace{\alpha_{AC}.S_{AC}}_{aAC}.(T_C - T_A) + \underbrace{m_C.c_C}_{PC}.\frac{dT_C}{dt} \tag{3}$$

Reactor

$$\underbrace{\alpha_{CD}.S_{CD}}_{aCD}.(T_C - T_D) + S_1.\sigma.(T_B^4 - T_D^4) + S_4.\sigma.(T_A^4 - T_D^4) = \underbrace{m_D.c_D}_{PD}.\frac{dT_D}{dt} \tag{4}$$

where
A is isolation
B is spiral
C is inner space
D is reactor
α_{ij} ,J.K^{-1}.m^{-2}.s^{-1}, is transfer coefficient between i and j
S_{ij} , m^2, is surface of contact between i and j
S_1,S ,S_3,S_4 are surfaces of reactor, isolation inside and outside surface of the furnace
m_i , kg, is weight of i
β , K^{-1}, is spiral temperature coefficient
c_i , J.K^{-1}.kg^{-1}, is capacity of i
σ , J.K^{-4}.m^{-2}.s^{-1}, is Stefan-Bolzmann constant
From the model it is evident that the system is strongly nonlinear and very difficult to control. Thus complex techniques are necessary to use.

3. Control techniques

Several control techniques with neural network were chosen, applied and compared to classical ones. One of the objectives is to find out whether control techniques with neural networks bring any improvement to control performances at all. Brief description of the applied techniques is given below.

3.1 Internal model control

Standard internal model control (IMC) is technique closely connected to direct inverse control which brings some limitations to system to be controlled. On the other side, IMC has some convenient features, e.g. it is able to cope well with output disturbances. The concept of IMC is

presented in [Rivera et al., 1986]. IMC for nonlinear systems is introduced in [Economou et al., 1986] and IMC with neural networks is described e.g. in [Norgaard et al., 2000].

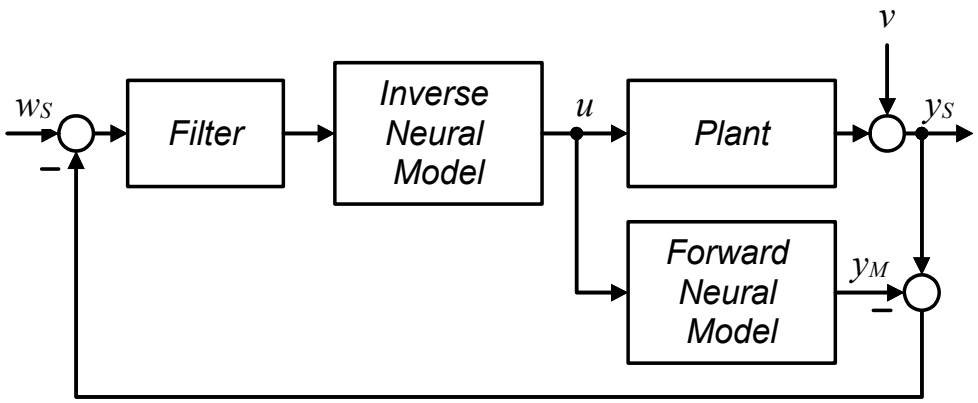


Fig. 2. Internal model control scheme

Internal model controller requires a forward model as well as an inverse model of the system to be controlled. Both models are replaced with adequate neural network model - design of both models is described in [Nguyen et al., 2003]. Then, control loop can be put together – see Fig. 2, where w_S, u, v, y_S, and y_M are reference variable, control signal, output disturbance, control variable and forward model output. It can be shown, that equation (5) is valid in case of ideal inverse and forward neural model. In some cases, filtering can be applied ahead of inverse controller to smooth reference variable to eliminate negative influence of sudden changes. In the case of linear continuous-time IMC, filter usage is essential.

$$y_S = 1w_S + 0v \tag{5}$$

The equation above is unattainable in real processes but can be approximately approached if discrete neural models are used.
In section 4.3, control experiments with neural models of linear IMC as well as IMC with neural models are demonstrated

3.2 Predictive control

Predictive control is used in two variants. The first one is typical Model Predictive Control and the second one is Neural Network Predictive Control.

3.2.1 Model predictive control

Model predictive control (MPC) is widely used technique for process control in industry, where better control performance is necessary. MPC is a general strategy which comes from the process model, therefore MPC controllers are truly-tailor-made. The working principle is briefly described in this chapter (the description is not in general, but only for SISO systems), more in [Camacho, 2007].
The mathematical model of the controlled system is assumed in the form of equation (6).

$$A(z^{-1})y(k) = z^{-d}.B(z^{-1})u(k-1) + C(z^{-1})e(k) \tag{6}$$

where A, B, C are polynomials, $y(k)$ is model output, $u(k)$ is model input $e(k)$ is output error. The model without errors and without output delay is supposed, therefore $C(z^{-1}) = 0$ a $d = 0$. Then it is possible to rewrite (6) to the form of (7).

$$A(z^{-1})y(k) = B(z^{-1})u(k-1) \tag{7}$$

The model is used for the calculation of future output prediction. There are several different methods how to calculate it. One of the simplest ways (using the inverse matrix) is described in this chapter.
The prediction of N steps is possible to write by the set of equations (8).

$$\begin{aligned}
&y(k+1) = b_1u(k) + b_2u(k-1) + ... \\
&... + b_{n+1}u(k-n) - a_1y(k) - a_2y(k-1) - ... - a_{m+1}y(k-m) \\
&\\
&y(k+2) = b_1u(k+1) + b_2u(k) + ... \\
&... + b_{n+1}u(k-n+1) - a_1y(k+1) - a_2y(k) - ... - a_{m+1}y(k-m+1) \\
&\\
&y(k+3) = b_1u(k+2) + b_2u(k+1) + ... \\
&... + b_{n+1}u(k-n+2) - a_1y(k+2) - a_2y(k+1) - \cdots \\
&\qquad .. - a_{m+1}y(k-m+2) \\
&\vdots \\
&y(k+N) = b_1u(k+N-1) + b_2u(k+N) + ... \\
&... + b_{n+1}u(k+N-n+1) - a_1y(k+N-1) - \\
&\qquad - a_2y(k+N) - ... - a_{m+1}y(k+N-m+1)
\end{aligned} \tag{8}$$

In matrix form it is possible to write

$$\mathbf{A}\begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ y(k+N) \end{bmatrix} = \mathbf{B}\begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N-1) \end{bmatrix} + \tilde{\mathbf{B}}\begin{bmatrix} u(k-1) \\ u(k-2) \\ \vdots \\ u(k-n) \end{bmatrix} + \tilde{\mathbf{A}}\begin{bmatrix} y(k) \\ y(k-1) \\ \vdots \\ y(k-m) \end{bmatrix} \tag{9}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -a_1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ -a_N & -a_{N-1} & \cdots & 1 \end{bmatrix}; \quad \dim(\mathbf{A}) = N \,\mathrm{x}\, N$$

$$\mathbf{B} = \begin{bmatrix} b_1 & 0 & \cdots & 0 \\ b_2 & b_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ b_N & b_{N-1} & \cdots & b_1 \end{bmatrix}; \quad \dim(\mathbf{B}) = N \,\mathrm{x}\, N$$

$$\tilde{\mathbf{A}} = \begin{bmatrix} a_1 & a_2 & \cdots & a_{m+1} \\ a_2 & \cdots & a_{m+1} & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{bmatrix}; \quad \dim(\tilde{\mathbf{A}}) = N \text{ x} (m+1);$$

$$\tilde{\mathbf{B}} = \begin{bmatrix} b_2 & b_3 & \cdots & b_{n+1} \\ b_3 & \cdots & b_{n+1} & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{bmatrix}; \quad \dim(\tilde{\mathbf{B}}) = N \text{ x} n$$

Output prediction $y(k+i)$ is possible to calculate by multiplying the equation (9) by the inverse matrix $\mathbf{A}^{-1}$, equation (10).

$$\begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ y(k+N) \end{bmatrix} = \mathbf{A}^{-1}\mathbf{B} \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N-1) \end{bmatrix} + \mathbf{A}^{-1}\tilde{\mathbf{B}} \begin{bmatrix} u(k-1) \\ u(k-2) \\ \vdots \\ u(k-n) \end{bmatrix} + \mathbf{A}^{-1}\tilde{\mathbf{A}} \begin{bmatrix} y(k) \\ y(k-1) \\ \vdots \\ y(k-m) \end{bmatrix} \tag{10}$$

Because the last two terms describe only the system history, it is possible to put them together to the matrix **F** and the vector of historical output and input $\mathbf{h} = \begin{bmatrix} \mathbf{y} & \mathbf{u} \end{bmatrix}^T$. Thus, it is possible to rewrite the equation of prediction to the form of equation (11).

$$\mathbf{y} = \mathbf{G}\mathbf{u} + \mathbf{F}\mathbf{h} \tag{11}$$

The aim of MPC is to calculate the vector of manipulated variable by minimizing the cost function (12), described in [Baotic, 2006].

$$J = \mathbf{e}_N^T . \mathbf{e}_N + \lambda . \mathbf{u}^T . \mathbf{u} \tag{12}$$

where **e** is vector of control errors (length N), **u** is vector of manipulated variables (length N) and λ is weighting coefficient.
The cost function can be modified using output prediction (10) and set point vector **w.**

$$J = (\mathbf{w} - \mathbf{G}.\mathbf{u} - \mathbf{F}.\mathbf{h})^T (\mathbf{w} - \mathbf{G}.\mathbf{u} - \mathbf{F}.\mathbf{h}) + \lambda . \mathbf{u}^\mathbf{T} . \mathbf{u} \tag{13}$$

It is possible to calculate the vector of manipulated variable **u** analytically using the square norm, equation (14).

$$\mathbf{u} = (\mathbf{G}^\mathbf{T}.\mathbf{G} + \lambda.\mathbf{I})^{-1}.\mathbf{G}^\mathbf{T}.(\mathbf{w} - \mathbf{F}.\mathbf{h}) \tag{14}$$

Only one actual value of the manipulated variable (the first element of the vector) is needed, therefore the final form of the control law is equation (15).

$$u = \mathbf{K}\left\{ \begin{bmatrix} w(t) \\ w(t+1) \\ \vdots \\ w(t+N) \end{bmatrix} - \mathbf{F}. \begin{bmatrix} u(t-1) \\ u(t-2) \\ \vdots \\ u(t-n) \\ y(t) \\ y(t-1) \\ \vdots \\ y(t-m) \end{bmatrix} \right\} \tag{15}$$

3.2.2 Method modification

GPC theory is formulated in detail for the group of linear systems but in the case of nonlinear systems it is not possible to use it because linear models are not able to describe nonlinear processes well enough. Nonlinear process control needs better description using piecewise linearized model.

In the case of linearized MPC several points where the linearization is done are chosen and for each point controller setting (matrices **G** and **F) is** pre-calculated**.** Then the controller switches between pre-calculated settings during control experiment (according to actual reactor temperature) and it is possible to interpolate between two adjoining settings. Thus, nonlinear behavior of the system is substituted by piecewise linearized model, more in [Mares et al., 2010b].

Control law can be transformed to equation (16).

$$u = \mathbf{K}. \begin{bmatrix} w(t) \\ w(t+1) \\ \vdots \\ w(t+N) \end{bmatrix} - \bar{\mathbf{F}}. \begin{bmatrix} u(t-1) \\ u(t-2) \\ \vdots \\ u(t-n) \\ y(t) \\ y(t-1) \\ \vdots \\ y(t-m) \end{bmatrix} \tag{16}$$

where vector **K** is the same as in equation (14) and vector $\bar{\mathbf{F}}$ is product of matrices **K** and **F** multiplying**.** Interpolation is the main reason of multiplying (it is simpler to interpolate between vectors than matrices).

The whole algorithm can be written as:

1. Pre-control
   fill the data history
   calculate vectors T_{LIN}, K_{LIN} and $\bar{\mathbf{F}}_{\mathbf{LIN}}$
2. Control

a. *measure actual temperature*

b. *choose the interval* $\mathbf{K}_{Ti}$ $\mathbf{K}_{Ti+1}$ and $\bar{\mathbf{F}}_{Ti}$ a $\bar{\mathbf{F}}_{Ti+1}$

c. *using interpolation calculate vectors* **K** a $\bar{\mathbf{F}}$ for the control law

d. *calculate the actual value of manipulated variable u*

e. *actualize the data history*

Vectors T_{LIN}, K_{LIN} and $\bar{\mathbf{F}}_{\mathbf{LIN}}$ are defined as

$$\mathbf{T_{LIN}} = \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_N \end{bmatrix}; \quad \mathbf{K_{LIN}} = \begin{bmatrix} \mathbf{K}_{\mathbf{T_1}} \\ \mathbf{K}_{\mathbf{T_2}} \\ \vdots \\ \mathbf{K}_{\mathbf{T_N}} \end{bmatrix}; \quad \bar{\mathbf{F}}_{\mathbf{LIN}} = \begin{bmatrix} \bar{\mathbf{F}}_{\mathbf{T_1}} \\ \bar{\mathbf{F}}_{\mathbf{T_2}} \\ \vdots \\ \bar{\mathbf{F}}_{\mathbf{T_N}} \end{bmatrix}$$

3.2.3 Neural network predictive control

There are several variations of neural network predictive controller. This approach uses a neural network model of nonlinear plant to predict future plant performance. The controller then calculates the control input that will optimize plant performance over a specified future time horizon.

The first stage of neural network predictive control is to design a neural network which represents the dynamics of the plant. The prediction error between the plant output and neural network output is used as the neural network training signal [Nguyen et al., 2003].

Obtained neural network predicts the plant response over a specified time horizon. The predictions are used by some search technique to determine the control signal that minimizes the following performance criterion over the specified horizon N

$$J = \mathbf{e}_N^T.\mathbf{e}_N + \lambda.\Delta\mathbf{u}^T.\Delta\mathbf{u} \tag{17}$$

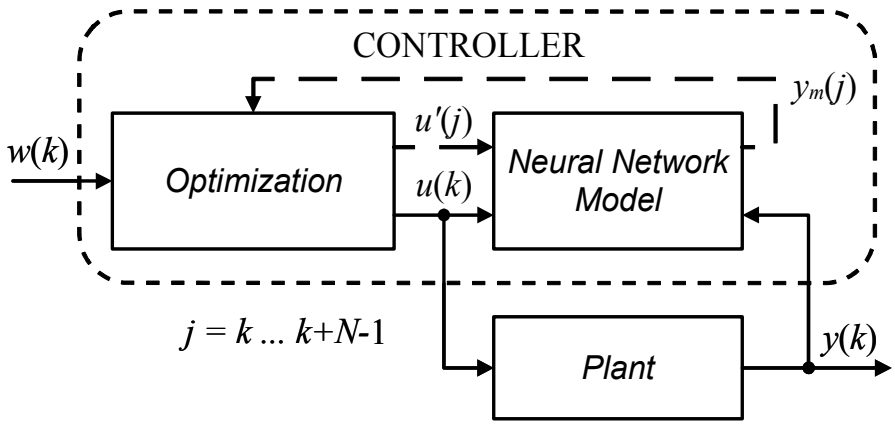


Fig. 3. Neural network predictive control

where **e** is vector of control errors and **Δu** is vector of controller outputs differences in time.

The following figure illustrates the neural network predictive control process. The controller consists of the neural network plant model and the optimization block. The optimization block determines the values of u' that minimize the criterion J and the optimal u' is input to the plant.

In section 4.4, experimental results of typical Model Predictive Control performance are compared to Neural Network Predictive Control ones.

3.3 Discrete controller tuning online

This technique amplifies the basic feedback control loop. It aims to tune any discrete controller online. For this purpose the knowledge of the controlled system model (e.g. neural model) and

reference variable course over known future finite horizon is necessary. Based on this, the parameters of any chosen discrete controller are determined repeatedly every discrete time instant so that the control response computed via the neural model over future horizon is optimal (according to chosen performance criterion). Simplified scheme is depictured in Fig. 4. The search of discrete controller parameters has to run repeatedly in every single step of the sampling interval, which puts great demands on computing time of the search algorithm. Naturally, usage of some iterative optimization algorithm with only one iteration realization every time instant is suggested. Gradient descent techniques seem inconvenient because of neural model usage. Neural model is black-box-like model so it is not possible to determine gradient descent analytically. On the other hand, evolutionary search techniques (genetic algorithm, differential evolution, … see [Coello et al., 2002]) appear to be suitable because these techniques do not require any particular information about search problem. The other indisputable advantage is its operating principle. In each iteration, evolutionary search techniques explore not only one value of input variables but whole set of them (one generation of individual solutions), which lowers significantly troubles with initial parameters random choice. In this particular case, differential evolution is chosen. The reasons are, among others, that differential evolution works with decimal input values (contrary to genetic algorithm) and population of possible solutions is kept more diversified.

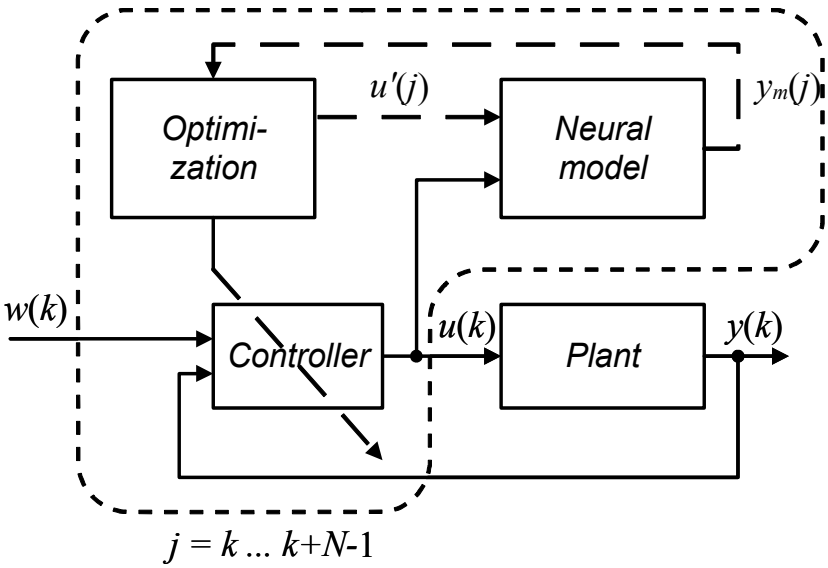


Fig. 4. Controller tuning online using neural network

The control method which is described here does not require any special form of discrete controller. According to some experiments [Dolezel et al., 2009a], [Dolezel et al., 2009b], controller form

$$u(k) = p_0 w(k) - p_1 y(k) - p_2 y(k-1) - p_3 y(k-2) + u(k-1) \tag{18}$$

was considered to be convenient. For some p_0 … p_3 parameters combinations, controller (18) acts like discrete PID controller. In general, however, it has one additional independent parameter. Suitable control performances can be obtained by well-tuned controller (18).
Because of the evolutionary algorithm, cost function can be selected from huge number of possibilities. One of suitable definitions is

$$J = \frac{1}{N} \cdot \sum_{i=k}^{k+N-1} |e(i)| + \frac{h_1}{N-1} \cdot \sum_{i=k+1}^{k+N-1} |\Delta u(i)| + h_2 \cdot |e(k+N-1)| \tag{19}$$

where $\Delta u(i) = u(i) - u(i\text{-}1)$, $e(i)$ is control error, h_1 is function parameter influencing manipulated variable differences, h_2 is function parameter influencing the state on the end of future horizon, N is future horizon length and $w(i)$ is reference variable.
Note that definition (19) can be changed in order to get any particular control performance.
The whole algorithm of the above described control method is compiled in the following points:

1. Create dynamical neural model of controlled system – see [Nguyen, 2003]
2. Choose future horizon length N
3. Choose differential evolution parameters (number of individual solutions in one generation NP – any solution represents one particular quaternion of controller parameters $p_0 \ldots p_3$, crossover constant CR, mutation constant F) and their initial values
4. Measure controlled variable $y(k)$
5. Perform one iteration of differential evolution (based on the knowledge of controlled variable $y(k)$, course of its reference $w(k)$ till $w(k+N\text{-}1)$ and neural model of controlled system)

a. *perform control simulation with discrete controller and the neural model over future horizon N and evaluate cost function for all the individual solutions from current generation*
b. *Apply cross-over and mutation (see* [Coello et al., 2002]) *so that offspring generation of solutions is bred*
c. *Evaluate cost functions of offspring (see step a))*
d. *Choose the best individual solution from the offspring generation*

6. Evaluate manipulated variable $u(k)$ with discrete controller determined by the best individual solution obtained in step 5d)
7. $k = k + 1$, go to step 4

4. Experimental results

4.1 Dynamical neural model of the plant

Control techniques described above need neural plant model to be designed. In [Nguyen, 2003], whole algorithm of neural model design is presented in detail. First, a training set of process data is to be measure. For this purpose, a simple control experiment with reactor furnace and PI controller is performed – see Fig. 5.
Data (sampling interval 3s) are slotted according to Table 1 so that neural network corresponds to difference equation (20)

$$T_D{}^M(k) = \varphi[T_D{}^M(k-1), T_D{}^M(k-2), E(k-1), E(k-2)] \tag{20}$$

Input				Output
$T_D{}^M(k\text{-}1)$	$T_D{}^M(k\text{-}2)$	$E(k\text{-}1)$	$E(k\text{-}2)$	$T_D{}^M(k)$
$T_D{}^M(2)$	$T_D{}^M(1)$	$E(2)$	$E(1)$	$T_D{}^M(3)$
$T_D{}^M(3)$	$T_D{}^M(2)$	$E(3)$	$E(2)$	$T_D{}^M(4)$
…	…	…	…	…
$T_D{}^M(N\text{-}1)$	$T_D{}^M(N\text{-}2)$	$E(N\text{-}1)$	$E(N\text{-}2)$	$T_D{}^M(N)$

Table 1. Training set 1

It is possible to choose higher order of difference equation (20), but after many experiments, second order seems convenient. Formal scheme of the neural model can be found in Fig. 6.

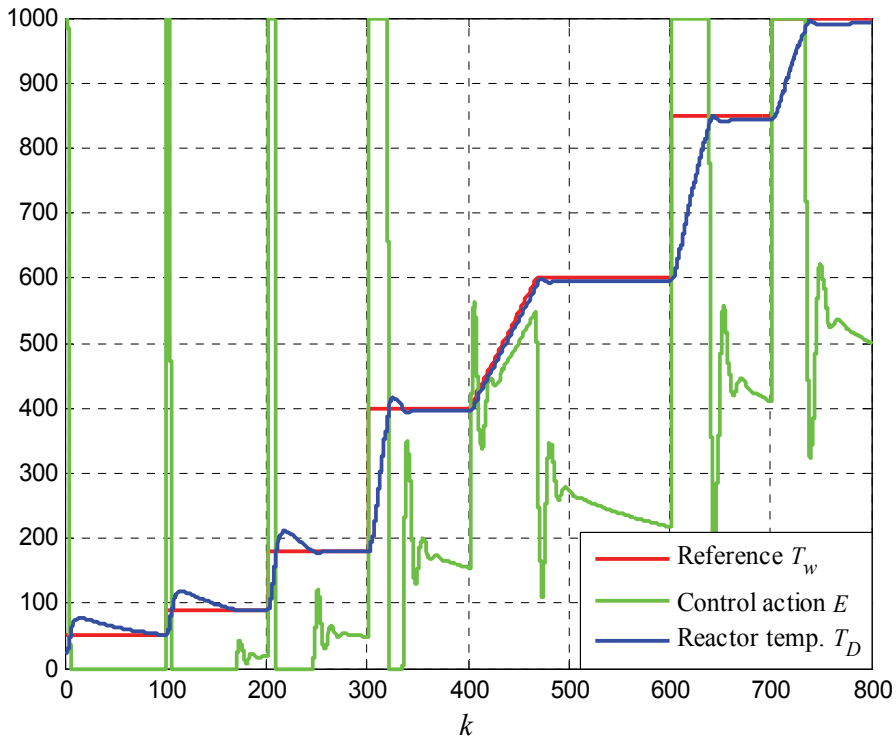


Fig. 5. Control performance with PI controller – Training set experiment

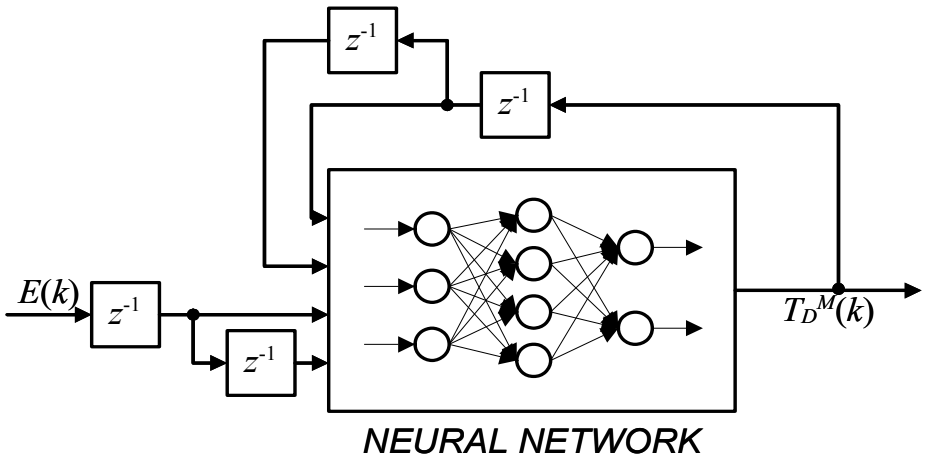


Fig. 6. Formal scheme of the neural model

Now, topology of the neural network has to be optimized. Several neural networks with different number of hidden neurons were trained (Levenberg-Marquardt Algorithm was used) and cost function courses are depictured in Fig. 7.

For control experiments neural model with network of 4-6-1 topology is chosen, because networks with more complex topologies do not bring considerably improved performances.

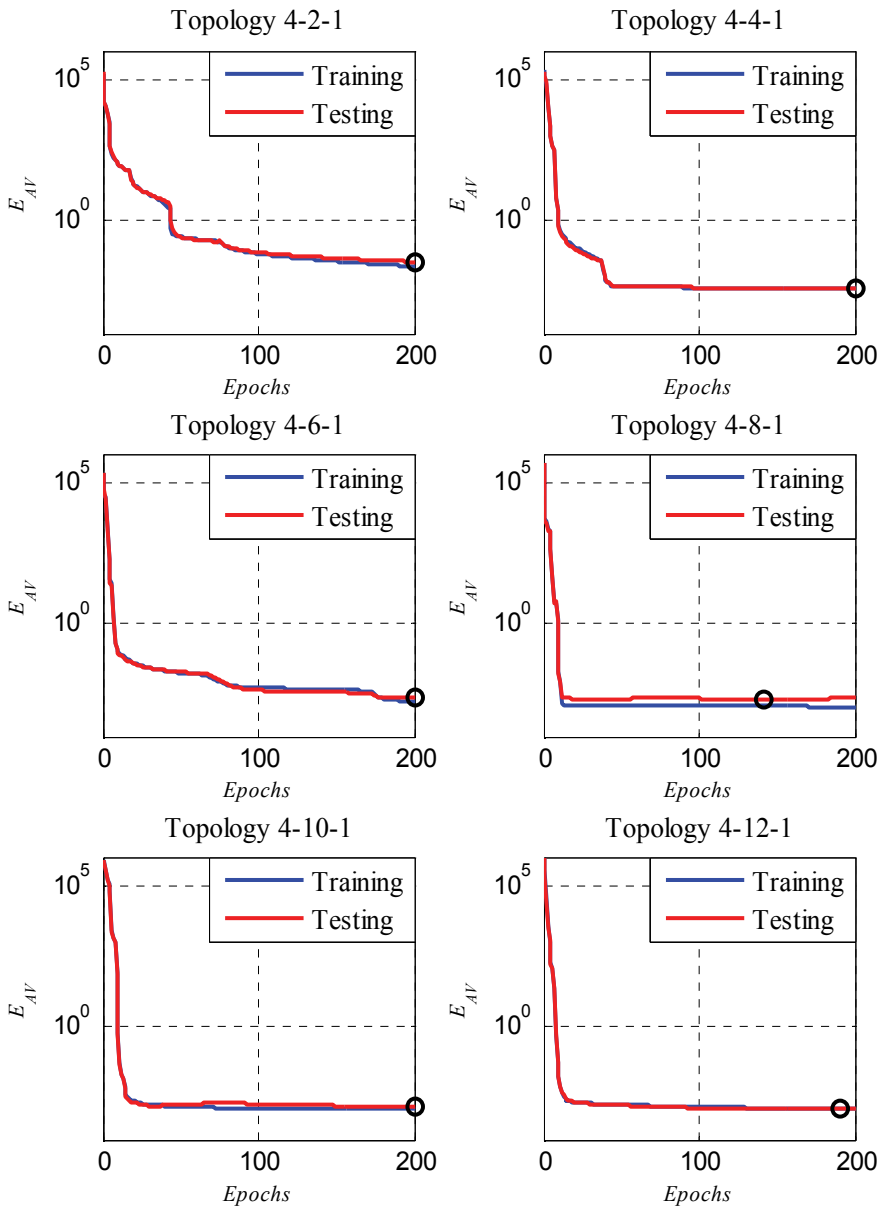


Fig. 7. Topology optimizing I

4.2 Inverse neural model of the plant

For IMC control an inverse plant model is needed. Inverse neural model control design is formally the same as feedforward model, the only difference is, that data of training set has to be slotted in another way. Inverse difference equation of equation (20) can be obtained by actual input-actual output interchanging – equation (21).

$$E(k-1) = \varphi[T_D{}^M(k), T_D{}^M(k-1), T_D{}^M(k-2), E(k-2)] \tag{21}$$

Input				Output
TDM(k)	***TDM(k-1)***	***TDM (k-2)***	***E(k-2)***	***E(k-1)***
$T_D{}^M(3)$	$T_D{}^M(2)$	$T_D{}^M(1)$	$E(1)$	$E(2)$
$T_D{}^M(4)$	$T_D{}^M(3)$	$T_D{}^M(2)$	$E(2)$	$E(3)$
…	…	…	…	…
$T_D{}^M(N)$	$T_D{}^M(N\text{-}1)$	$T_D{}^M(N\text{-}2)$	$E(N\text{-}2)$	$E(N\text{-}1)$

Table 2. Training set 2

Equation above lacks time causality. However, it can be used to training set slotting – see Table 2.

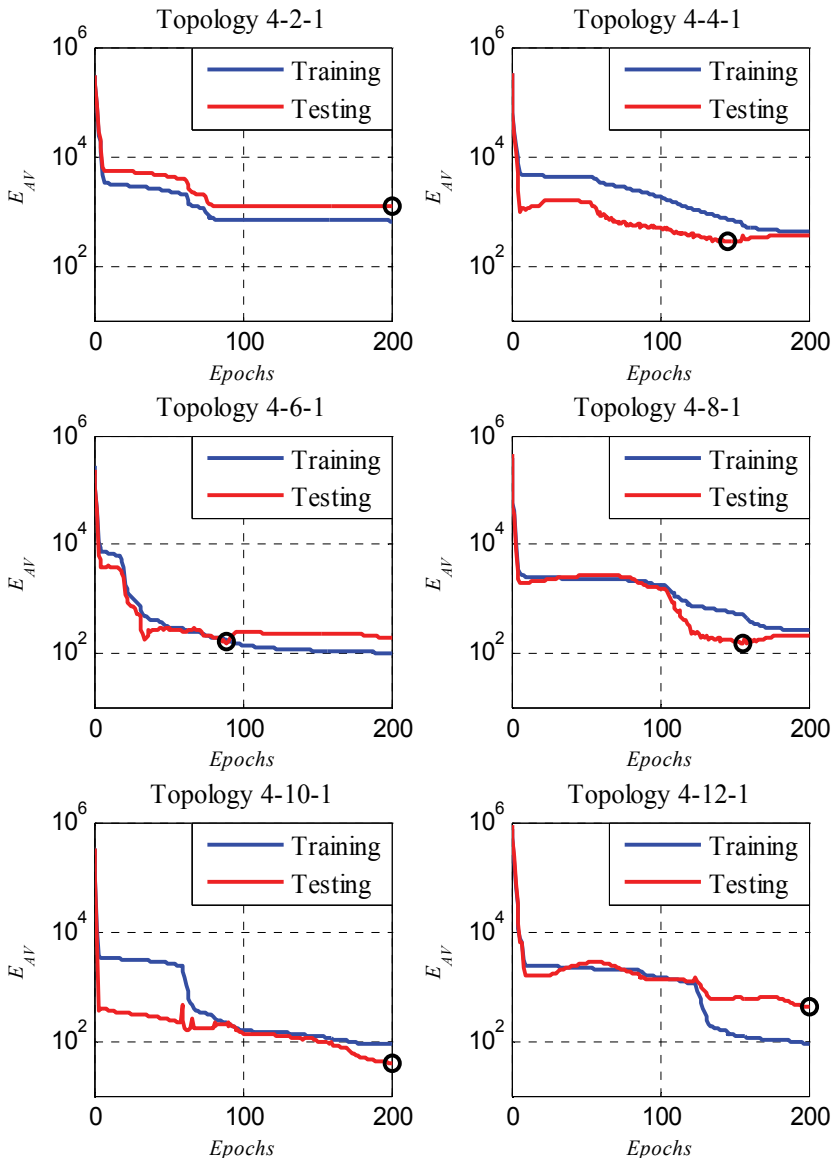


Fig. 8. Topology optimizing II

Topology is optimized as well as in section 4.1. Cost function courses are depictured in Fig. 8. Now, inverse neural model with 4-10-1 topology is chosen.

4.3 Neural internal model control

If both feedforward and inverse neural models are designed, control loop can be put together – see Fig. 9.

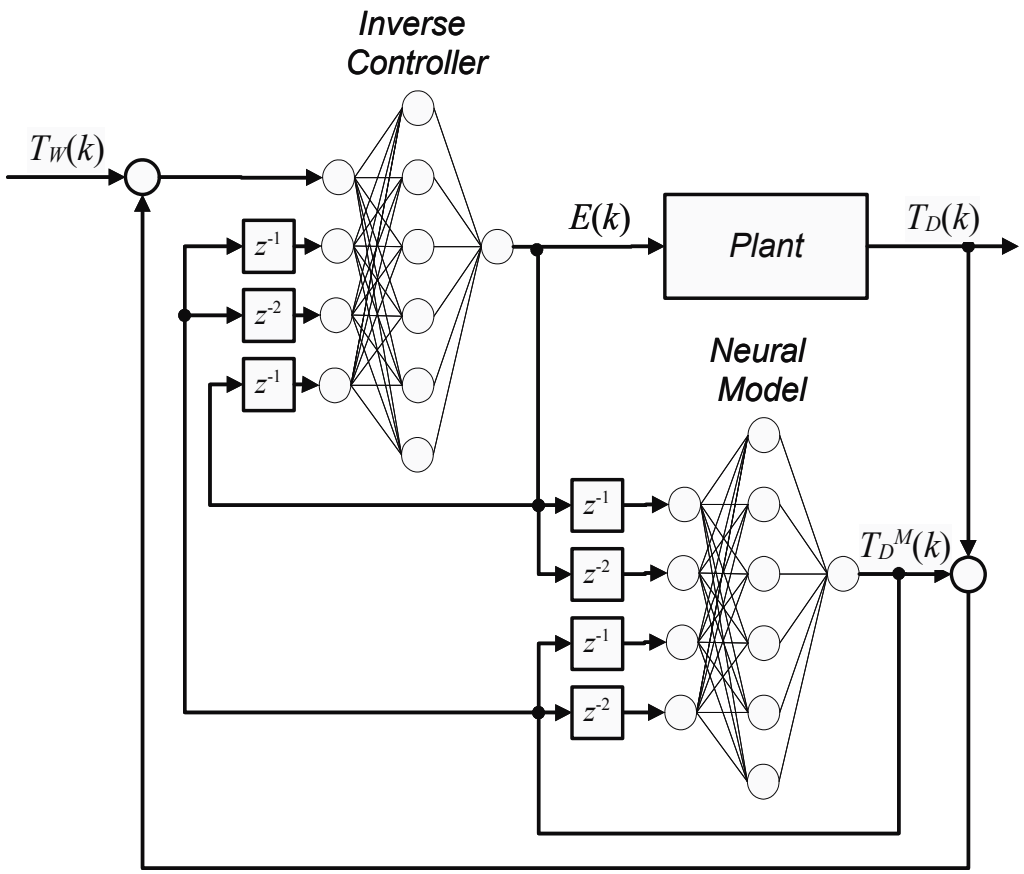


Fig. 9. Reactor Furnace controlled using Neural IMC

Control response for the desired plant output course is shown below compared to response obtained by classical IMC (for linear model design, same data is used).

For ramp as reference, reactor temperature courses are similar for both control techniques (in addition, control response with classical IMC is smoother), so for reactor furnace control, it is no need to extend classical IMC technique with neural networks.

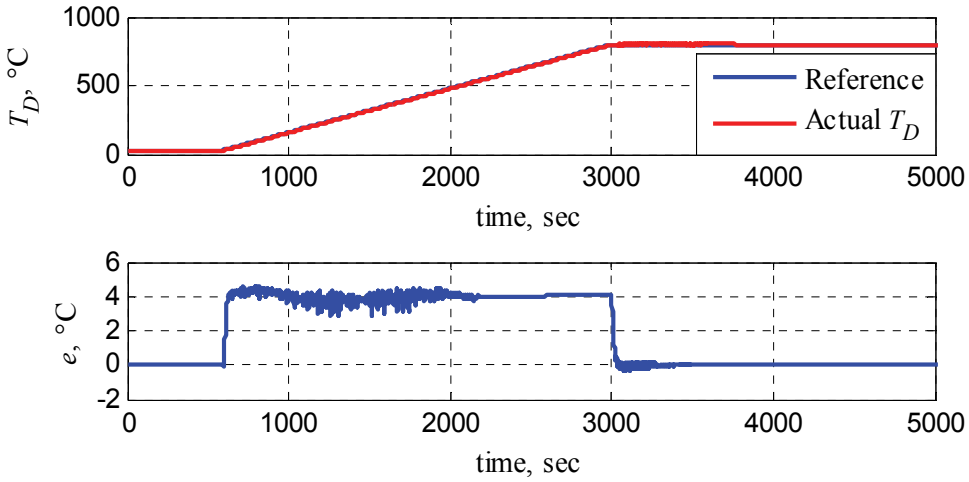


Fig. 10. Neural IMC control response

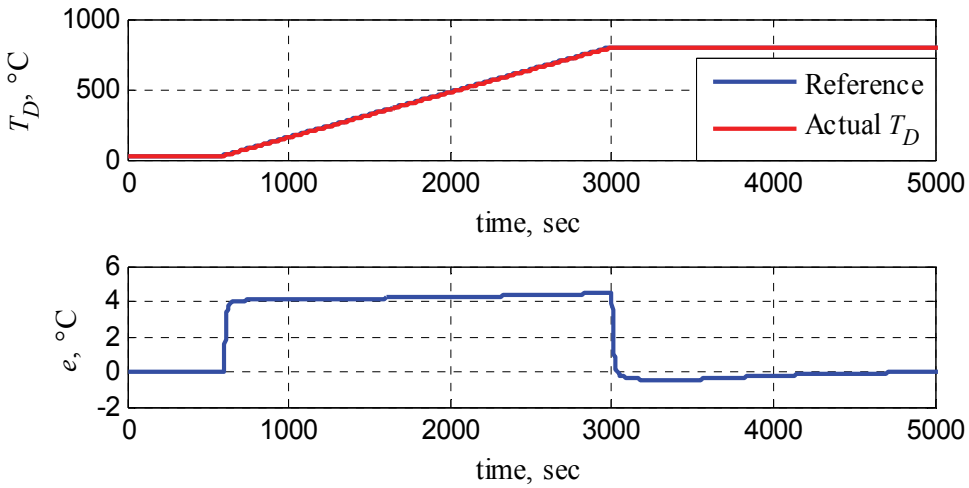


Fig. 11. Classical IMC control response

4.4 Neural network predictive control

As shown in section 3.2.3, only feedforward neural model of the plant is needed for neural network predictive control. Control experiment is performed according to section 3.2.3 with golden section search routine [Fletcher, 1987], prediction horizon N = 20 and weighting coefficient λ = 0.1. Control response is shown in Fig. 12.

Alternatively, control response gained by piecewise linearized model predictive controller (the same prediction horizon and weighting coefficient – see section 3.2.1 and 3.2.2) is plotted in Fig. 13. It is obvious that neural network predictive controller provides less suitable controlperformance. However, it has to be mentioned, that neural network predictive controller is much simpler to design than piecewise linear model predictive controller.

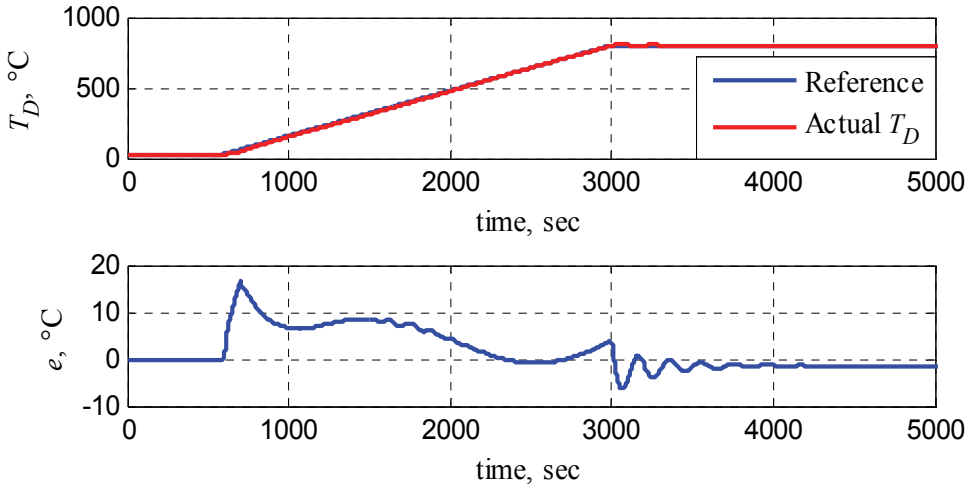


Fig. 12. Neural Network Predictive control response

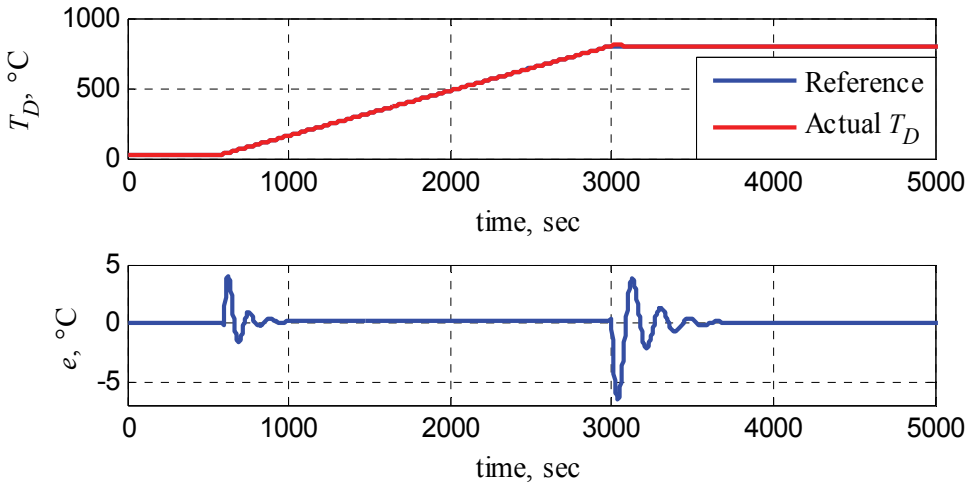


Fig. 13. Piecewise Linearized Model Predictive control response

4.5 Discrete controller tuning online

Control loop of this technique is connected in a way introduced briefly in section 3.3. Differential evolution is chosen as search technique. After some experiments, eligible parameters are chosen this way: $NP = 30$; $CR = 0.85$; $F = 0.6$; $N = 20$. Cost function is selected according to Eq. (19), where $h_1 = 0.1$, $h_2 = 0.01$. Control response is depicted in Fig. 14.

There is no exact alternative in classical control theory to this technique. However, in a certain way it is close to predictive control, therefore it can be compared to Fig. 13.

It is remarkable, that control response shown in Fig. 14 provides the most suitable performance of all experiments. But, on the other hand, it is highly computationally demanding technique.

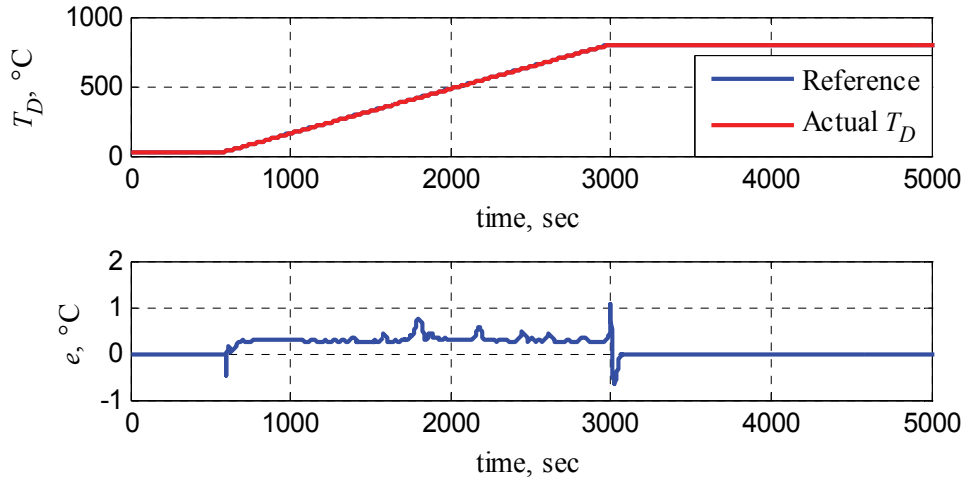


Fig. 14. Discrete controller tuned online

5. Conclusion

The aim of this work was to design a controller, which provides control performance with control error less than 10°C. Because of the nonlinearity of the plant, two groups of advanced control techniques were used. The first group is based on artificial neural networks usage while the second one combines their alternatives in modern control theory. Generally speaking, neural networks are recommended to use when plant is strongly nonlinear and/or stochastic. Although reactor furnace is indispensably nonlinear, it is evident that control techniques without neural networks can control the plant sufficiently and in some cases (especially predictive control and internal model control) even better. Thus, neural network usage is not strictly necessary here, although especially Discrete Controller Tuning Online brings extra good performance.

6. Acknowledgement

This work was supported by the 6th Framework Programme of the European Community under contract No. NMP2-CT-2007-026515 "Bioproduction project - Sustainable Microbial and Biocatalytic Production of Advanced Functional Materials" and by the funds No. MSM 6046137306 and No. MSM 0021627505 of the Czech Ministry of Education. This support is very gratefully acknowledged.

7. References

Baotic, M.; Christophensen, F.; Morari , M. (2006). Constrained Optimal Control of Hybrid Systems With a Linear Performance Index. *IEEETrans. on Automatic Control,* Vol.51, No 12., ISSN 1903-1919.

Camacho, E.F.; Bordons, C. (2007). *Model Predictive Control*, Springer-Verlag, ISBN 1-85233-694-3, London

Coello, C. A. C.; Lamont, G. B. (2002). *Evolutionary Algorithms for Solving Multi-Objective problems*, Springer, ISBN 978-0-387-33254-3, Boston

Dolezel, P.; Taufer, I. (2009a). PSD controller tuning using artificial intelligence techniques, *Proceedings of the 17th International Conference on Process Control '09*, pp. 120-124, ISBN 978-80-227-3081-5, Strbske Pleso, june 2009, STU, Bratislava.

Dolezel, P.; Mares, J. (2009b). Reactor Furnace Control using Artificial Neural Networks and Genetic Algoritm, *Proceedings of the International Conference on Applied Electronics*, pp. 99-102, ISBN 978-80-7043-781-0, Plzen, september 2009, ZCU, Plzen.

Dwarapudi, S.; Gupta, P. K.; Rao, S. M. (2007). Prediction of iron ore pellet strength using artificial neural network model, *ISIJ International,* Vol. 47, No 1., ISSN 0915-1559.

Economou, C.; Morari, M.; Palsson, B. (1986). Internal Model Control: extension to nonlinear system, *Industrial & Engineering Chemistry Process Design and Development,* Vol. 26, No 1, pp. 403-411, ISSN 0196-4305.

Fletcher, R. (1987). *Practical Methods of Optimization*, Wiley, ISBN 978-0-471-91547-8, Chichester, UK.

Lichota, J. ; Grabovski, M. (2010). Application of artificial neural network to boiler and turbine control, *Rynek Energii*, ISSN 1425-5960.

Mares, J., Dusek, F., Dolezel, P.(2010a) Nelinearni a linearizovany model reaktorové pece. In *Proceedings of Conference ARTEP'10*", 24.-26. 2 2010.Technicka univerzita Kosice, 2010. Pp. 27-1 – 27-14. ISBN 978-80-553-0347-5.

Mares, J., Dusek, F., Dolezel, P.(2010b). Prediktivni rizeni reaktorove pece. *In Proceedings of XXXVth Seminary ASR'10 „Instruments and Control*", VSB- Technical University Ostrava, 2010. Pp. 269 – 279. *ISBN 978-80-248-2191-7.*

Montague, G.; Morris, J. (1994). Neural network contributions in biotechnology, *Trends in biotechnology,* Vol. 12, No 8., ISSN 0167-7799.

Nguyen, H.; Prasad, N.; Walker, C. (2003). *A First Course in Fuzzy and Neural Control*, Chapman & Hall/CRC, ISBN 1-58488-244-1, Boca Raton.

Norgaard, M.; Ravn, O.; Poulsen, N. (2000). *A Neural Networks for Modelling and Control of Dynamic Systems*, Springer-Verlag, ISBN 978-1-85233-227-3, London.

Rivera, D.; Morari, M.; Skogestad, S. (1986). Internal Model Control: PID Controller Design, *Industrial & Engineering Chemistry Process Design and Development,* Vol. 25, No 1., pp. 252-265, ISSN 0196-4305.

Teixeira, A.; Alves, C.; Alves, P. M. (2005). Hybrid metabolic flux analysis/artificial neural network modelling of bioprocesses, *Proceedings of the 5th International Conference on Hybrid Intelligent Systems*, ISBN 0-7695-2457-5, Rio de Janeiro.

18

Direct Neural Network Control via Inverse Modelling: Application on Induction Motors

Haider A. F. Almurib[1], Ahmad A. Mat Isa[2] and Hayder M.A.A. Al-Assadi[2]
[1]*Department of Electrical & Electronic Engineering, The University of Nottingham Malaysia Campus Semenyih, 43500*
[2]*Faculty of Mechanical Engineering, University Technology MARA (UiTM) Shah Alam, 40450*
Malaysia

1. Introduction

Applications of Artificial Neural Networks (ANNs) attract the attention of many scientists from all over the world. They have many advantages over traditional algorithmic methods. Some of these advantages are, but not limited to; ease of training and generalization, simplicity of their architecture, possibility of approximating nonlinear functions, insensitivity to the distortion of the network and inexact input data (Wlas et al., 2005). As for their applications to Induction Motors (IMs), several research articles have been published on system identification (Karanayil et al., 2003; Ma & Na, 2000; Toqeer & Bayindir, 2000; Sjöberg et al. 1995; Yabuta & Yamada, 1991), on control (Kulawski & Brys, 2000; Kung et al., 1995; Henneberger & Otto, 1995), on breakdown detection (Raison, 2000), and on estimation of their state variables (Simoes & Bose, 1995; Orłowska-Kowalska & Kowalski, 1996).

The strong identification capabilities of artificial neural networks can be extended and utilized to design simple yet good performance nonlinear controllers. This chapter contemplates this property of ANNs and illustrates the identification and control design processes in general and then for a given system as a case study.

To demonstrate its capabilities and performance, induction motors which are highly nonlinear systems are considered here. The induction machine, especially the squirrel-cage induction motor, enjoys several inherent advantages like simplicity, ruggedness, efficiency and low cost, reliability and compactness that makes it the preferred choice of the industry (Vas, 1990; Mehrotra et al., 1996; Wishart & Harley, 1995; Merabet et al., 2006; Sharma, 2007). On the other hand, advances in power switching devices and digital signal processors have significantly matured voltage-source inverters (VSIs) with the associated pulse width modulation (PWM) techniques to drive these machines (Ebrahim at el., 2010). However, IMs comprise a theoretically challenging problem in control, since they are nonlinear multivariable time-varying systems, highly coupled, nonlinear dynamic plants, and in addition, many of their parameters vary with time and operating condition (Mehrotra et al., 1996a; 1996b; Merabet et al., 2006).

2. System identification

This chapter will carry out the system identification of an induction motor using the artificial neural network and precisely the Back Propagation Algorithm. The procedure used to identify the system is as described in Fig.1.

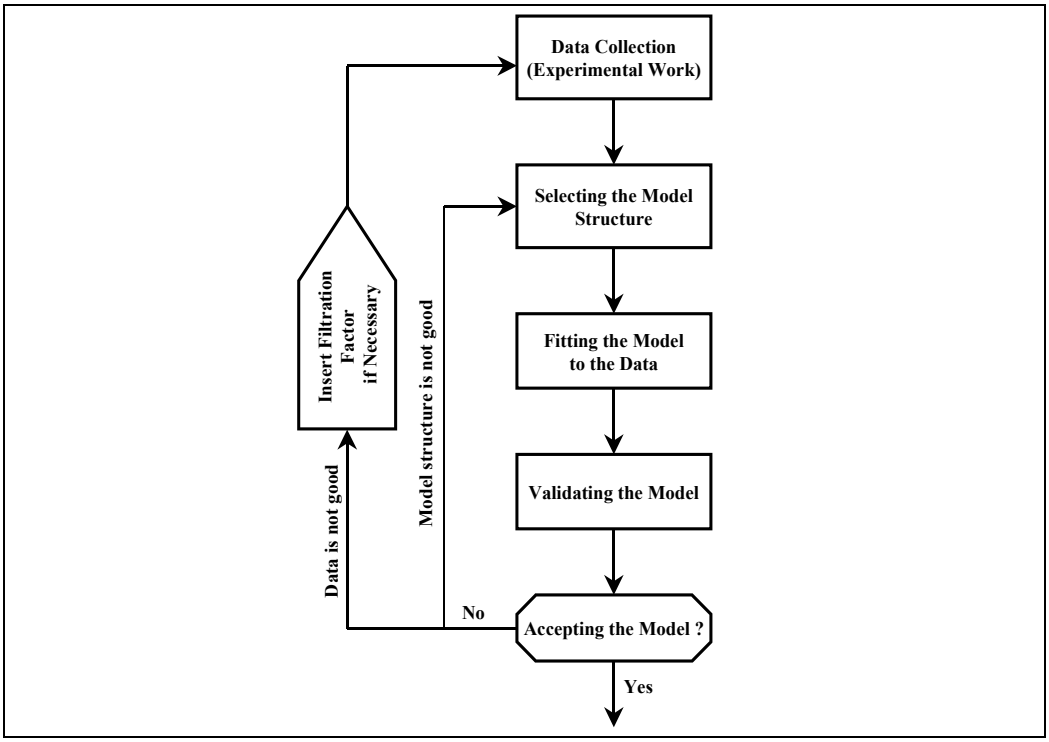


Fig. 1. System identification loop

Now, the system identification problem would be as follows: We have observed inputs, *u(t)*, and outputs, *y(t)*, from the plant under consideration (induction motor):

$$u^t = \left[u(1), u(2), \cdots, u(t)\right] \tag{1}$$

$$y^t = \left[y(1), y(2), \cdots, y(t)\right] \tag{2}$$

where u^t is the input signal to the plant (input to the frequency inverter) and y^t is the output signal (measured by the tacho-meter representing the motor's speed). We are looking for a relationship between past $\left[u^{t-1}, y^{t-1}\right]$ and future output, *y(t)*:

$$\hat{y}(t \mid \theta) = g\left[\varphi(t), \theta\right] \tag{3}$$

where $\hat{y}$ denotes the model output which approximates the actual output $y(t)$, g is a nonlinear mapping that represents the model, $\varphi(t)$ is the regression vector given by

$$\varphi(t) = \varphi\left(u^{t-1}, y^{t-1}\right) \tag{4}$$

and its components are referred to as regressors. Here, θ is a finite dimensional parameter vector, which is the weights of the network in our case (Bavarian, 1988; Ljung & Sjöberg, 1992; Sjöberg et al. 1995).
The objective in model fitting is to construct a suitable identification model (Fig. 2) which when subjected to the same input $u(t)$ to the plant, produces an output $\hat{y}(t)$ which approximates $y(t)$. However, in practice, it is not possible to obtain a perfect model. The solution then is to select θ in Eq. (3) so as to make the calculated values of $\hat{y}(t|\theta)$ fit to the measured outputs $y(t)$ as close as possible. The fit criterion will be based on the least square method given by

$$\min_{\theta} V_N\left[\theta, \varphi(t)\right] \tag{5}$$

where

$$V_N\left[\theta, \varphi(t)\right] = \frac{1}{N}\sum_{t=1}^{N}\left[y(t) - \hat{y}(t|\theta)\right]^2 \tag{6}$$

Hence, the error ε is given by

$$\varepsilon(t) = y(t) - \hat{y}(t|\theta) \tag{7}$$

This is illustrated in Fig. 2.

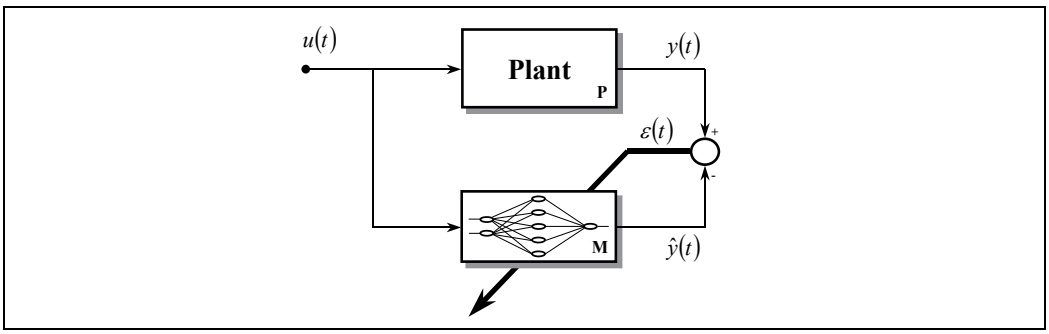


Fig. 2. Forward plant modelling

3. Artificial Neural Networks

Strong non-linearities and model uncertainty still pose a major problem for control engineering. Adaptive control techniques can provide solutions in some situations however in the presence of strongly non-linear behaviour of the system traditional adaptive control algorithms do not yield satisfactory performance. Their inherent limitations lie in the linearization based approach. A linear model being a good approximation of the non-linear plant for a given operation point cannot catch up with a fast change of the state of the plant and poor performance is observed until new local linear approximation is built.
Artificial neural networks offer the advantage of performance improvement through learning using parallel and distributed processing. These networks are implemented using massive connections among processing units with variable strengths, and they are attractive for applications in system identification and control.

3.1 The network architecture

Figure 3 shows a typical two-layer artificial neural network. It consists of two layers of simple processing units (termed neurons).
The outputs computed by unit j of the hidden-layer and unit k of the output-layer are given by:

$$x_j = f_h\left(H_j\right) \qquad j = 1,\ 2,\ ...,\ h \tag{8}$$

$$y_k = f_o\left(I_k\right) \qquad k = 1,\ 2,\ ...,\ m \tag{9}$$

respectively, where f_h and f_o are the bounded and differentiable activation functions. Thus, the output unit k will result in the following:

$$y_k = f\left[\sum_j w_{kj} f\left(\sum_i v_{ji} u_i\right)\right] \tag{10}$$

where y_k here is the vector representing the network output.
It has been formally shown (Lippman, 1987; Fukuda & Shibata, 1992) that Artificial Neural Networks with at least one hidden layer with a sufficient number of neurons are able to approximate a wide class continuous non-linear functions to within an arbitrarily small error margin.

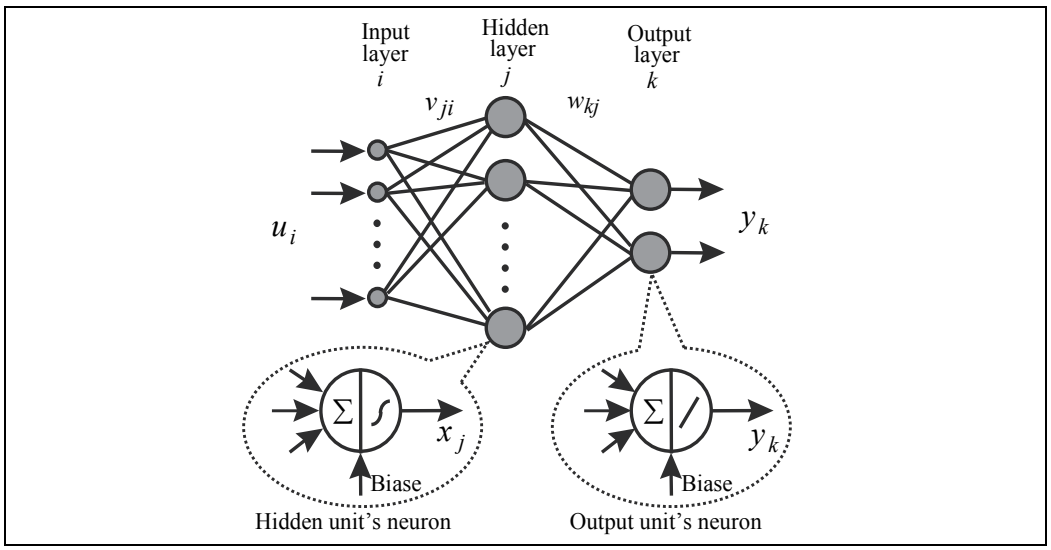


Fig. 3. A two layer artificial neural network

3.2 The training agorithm

In developing a training algorithm for this network, we want a method that specifies how to reduce the total system error for all patterns through an adjustment of the weights. This chapter uses the Back-Propagation training algorithm which is an iterative gradient algorithm designed to minimize the mean square error between the actual output of a feed-forward network and the desired output (Lippman, 1987; Weber et al., 1991; Fukuda & Shibata, 1992).

The back-propagation training is carried out as follows: the hidden layer weights are adjusted using the errors from the subsequent layer. Thus, the errors computed at the output layer are used to adjust the weights between the last hidden layer and the output layer. Likewise, an error value computed from the last hidden layer output is used to adjust the weights in the next to the last hidden layer and so on until the weight connections to the first hidden layer are adjusted. In this way, errors are propagated backwards layer by layer with corrections being made to the corresponding layer weights in an iterative manner. The process is repeated a number of times for each pattern in the training set until the criterion minimization is reached. This is illustrated in Fig. 4. Therefore, we first calculate the predicted error at each time step s (we refer to s here to introduce the discrete time factor). Then, an equivalent error is calculated for each neuron in the network. For example the equivalent error δ_k of the neuron k in the output layer is given by (taking into account that the derivative of the output layer's activation function is unity because it is a linear activation function):

$$\delta_k(s) = \varepsilon_k(s) = y_k(s) - \hat{y}_k(s) \tag{11}$$

The equivalent error δ_j of neuron j in the hidden layer is given by:

$$\delta_j(s) = \frac{df\left(H_j(s)\right)}{dH_j(s)} \sum_k \delta_k(s) w_{kj} \tag{12}$$

Weights connecting the hidden and output layers are adjusted according to:

$$\begin{aligned} w_{kj}(s) &= w_{kj}(s-1) + \Delta w_{kj}(s) \\ \Delta w_{kj}(s) &= \alpha \delta_k(s) x_j(s) + \beta \Delta w_{kj}(s-1) \end{aligned} \tag{13}$$

where: α and β are the learning rate and the momentum parameters respectively. Weights connecting the input and hidden layer are adjusted according to:

$$\begin{aligned} v_{ji}(s) &= v_{ji}(s-1) + \Delta v_{ji}(s) \\ \Delta v_{ji}(s) &= \alpha \delta_j(s) u_i(s) + \beta \Delta v_{ji}(s-1) \end{aligned} \tag{14}$$

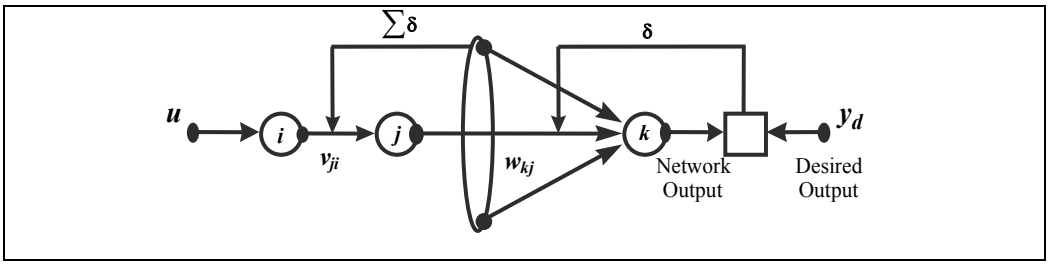


Fig. 4. Back-propagation algorithm

In summary, the training algorithm is as follow: the output layer error is calculated first using Eq. (11) and then backpropagated through the network using Eq. (12) to calculate the

equivalent errors of the hidden neurons. The network weights are then adjusted using Eq. (13) and Eq. (14).

3.3 Model validation

In order to check if the identified model agrees with the real process behavior, model validation is necessary. This is imperative as to taken into account the limitations of any identification method and its final goal of model application. This includes a check to determine if the priori assumptions of the identification method used are true and to compare the input-output behaviour of the model and the plant (Ljung & Guo, 1997).

To validate the model, a new input will be applied to the model under validation tests. The new outputs will be compared with the real time outputs and validation statistics is calculated. These statistics will decide whether the model is valid or not.

To carry out the validation task, we use the following statistics for the model residuals:

The maximal absolute value of the residuals

$$M_N^{\varepsilon} = \max_{1\le t\le N} \left|\varepsilon(t)\right| \tag{15}$$

Mean, Variance and Mean Square of the residuals

$$m_N^{\varepsilon} = \frac{1}{N}\sum_{t=1}^{N} \varepsilon(t) \tag{16}$$

$$V_N^{\varepsilon} = \frac{1}{N}\sum_{t=1}^{N} \left[\varepsilon(t) - m_N^{\varepsilon}\right]^2 \tag{17}$$

$$S_N^{\varepsilon} = \frac{1}{N}\sum_{t=1}^{N} \varepsilon(t)^2 = \left(m_N^{\varepsilon}\right)^2 + V_N^{\varepsilon} \tag{18}$$

In particular we stress that the model errors must be separated from any disturbances that can occur in the modelling. As this can correlates the model residuals and the past inputs. This plays a crucial role. Thus, it is very useful to consider two sources of model residuals or model errors ε. The first error originates from the input $u(t)$ while the other one originates from the identified model itself. If these two sources of error are additive and the one that originates from the input is linear, we can write

$$\varepsilon(t) = \Delta(q)u(t) + v(t) \tag{19}$$

Equation (19) is referred to as the separation of the model residuals and the disturbances. Here, *v(t)* would not change, if we changed the input *u(t)*. To check the part of the residuals that might originate from the input, the following statistics are frequently used:

If past inputs are $\phi(t) = \left[u(t), u(t-1), \cdots, u(t-M+1)\right]^T$ and $R_N = \frac{1}{N}\sum_{t=1}^{N} \phi(t)\phi(t)^T$, then the scalar measure of the correlation between past inputs $\phi(t)$ and the residuals $\varepsilon(t)$ is given by:

$$\xi_N^M = r_{\varepsilon u}^T R_N^{-1} r_{\varepsilon u} \tag{20}$$

where $r_{\varepsilon u} = \left[r_{\varepsilon u}(0), \cdots, r_{\varepsilon u}(M-1) \right]^T$

with $r_{\varepsilon u}(\tau) = \frac{1}{\sqrt{N}} \sum_{t=1}^{N} \varepsilon(t) u(t-\tau)$.

The obtained model should pass the validation tests of a given data set. Then we can say that our model is unfalsified. Here, we shall examine our model when the validation test is based on some of the statistics given previously in Eqs. (15-20).

Let us first assume that the model validation criterion be a positive constant $\mu > 0$ for the maximal absolute value of the residuals M_N^{ε} stated in Eq. (15)

$$g\left[\varphi(t), \theta\right] \text{ is not validated iff } M_N^{\varepsilon}(\theta) \le \mu \tag{21}$$

The problem of determining which models satisfy the inequality of Eq. (21) is the same problem that deals with set membership identification (Ninness & Goodwin, 1994). Typically this set is quite complicated and it is customary to outerbound it either by an ellipsoid or a hypercube. Therefore, it is agreed that a reasonable candidate model for the true dynamics should make the sample correlation between residuals $\varepsilon(t,\theta) = y(t) - \hat{y}(t \mid \theta)$ and past inputs $u(t-1), \cdots, u(t-m)$ small within certain criterion. One possible validation criterion is to require this correlation to be small in comparison with the Mean Square of the Model Residuals S_N^{ε} stated in Eq. (18). This is given by:

$$g\left[\varphi(t), \theta\right] \text{ is not validated iff } \xi_N^M(\theta) \le \gamma S_N^{\varepsilon}(\theta) \tag{22}$$

where γ is a subjective threshold that will be selected according to the application.

4. The neurocontroller

Conceptually, the most fundamental neural network based controllers are probably those using the inverse of the plant as the controller. The simplest concept is called direct inverse control, which is used in this chapter. Before considering the actual control system, an inverse model must be trained. There are tow ways of training the model; generalized training and the specialized training. This chapter uses the generalized training method. Figure 5 shows the off-line diagram of the inverse plant modelling.

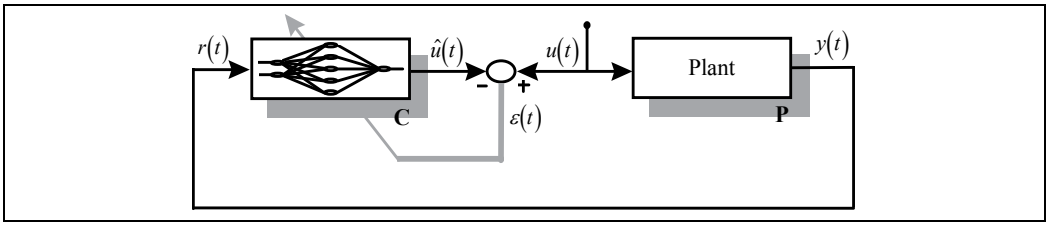


Fig. 5. Inverse plant modelling

Given the input-output data set which will be referred to as Z^N over the period of time $1 \le t \le N$

$$Z^N = \left\{ u(1), y(1), ..., u(N), y(N) \right\} \tag{23}$$

where *u(t)* is the input signal and $y(t)$ is the output signal, the system identification task is basically to obtain the model $\hat{y}(t \mid \theta)$ that represent our plant;

$$\hat{y}(t \mid \theta) = g\left(\theta, Z^N\right) \tag{24}$$

where $\hat{y}$ denotes the model output and g is some non-linear function parameterized by θ which is the finite dimensional parameter vector, the weights of the network in our case (Ljung & Sjöberg 1992; Ljung, 1995; Sjöberg, 1995).
The objective with inverse plant modelling is to formulate a controller, such that the overall controller-plant architecture has a unity transfer function, i.e., if the plant can be described as in Eq. (24), a network is trained as the inverse of the process:

$$\hat{u}(t \mid \theta) = g^{-1}\left(\theta, Z^N\right) \tag{25}$$

However, modelling errors perturb the transfer function away from unity. Therefore, $\hat{g}^{-1}\left(\theta, Z^N\right)$ will be used instead of $g^{-1}\left(\theta, Z^N\right)$.
To obtain the inverse model in the generalized training method, a network is trained off-line to minimize the following criterion instead:

$$W_N\left(\theta, Z^N\right) = \frac{1}{N}\sum_{t=1}^{N}\left(u(t) - \hat{u}(t \mid \theta)\right)^2 \tag{26}$$

In other words, our aim is to reduce the error ε where:

$$\varepsilon(t) = u(t) - \hat{u}(t \mid \theta) \tag{27}$$

Once we carry out that, the inverse model is subsequently applied as the controller for the system by inserting the desired output (the reference) instead of the system output. This is illustrated in Fig. 6.

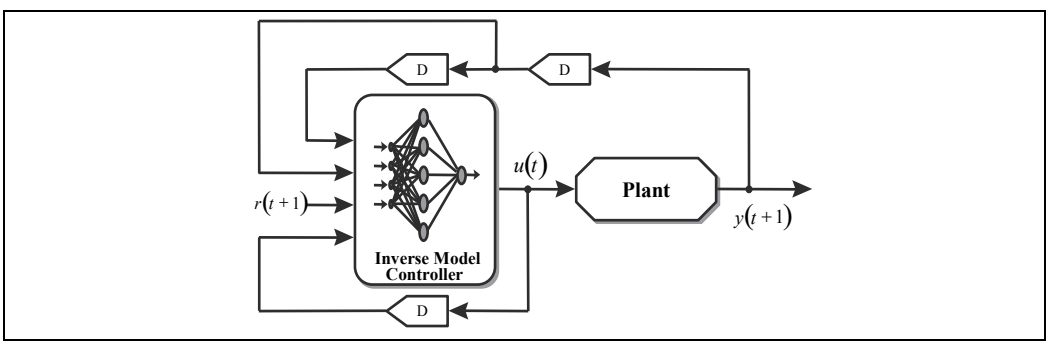


Fig. 6. Direct inverse control

5. Simulation and results

The first step is to collect training data from the real plant, which is a three phase squirrel-cage induction motor with the following ratings: 380V, 50Hz, 4-pole, 0.1kW, 1390rpm, and is

Y-connected. That was carried out by using a data acquisition card to interface the induction motor and the inverter and its inputs and outputs to the computer. A voltage signal is to be sent to the frequency inverter which changes the three phase lines frequency into a new signal with different frequency to drive the induction machine speed. That was the input signal. The output signal is taken from a tachometer connected directly to the rotor shaft and back to the interfacing data acquisition card as the speed signal. Figure 7 shows the overall experimental system setup.

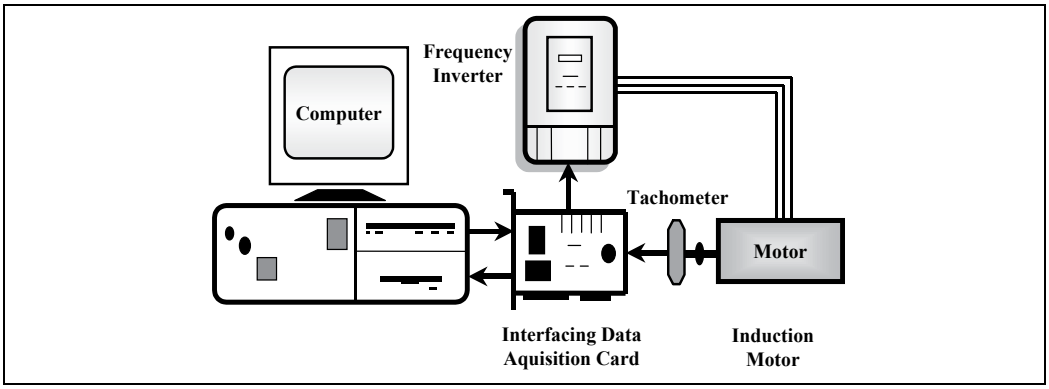


Fig. 7. The experimental work

5.1 Results of system identification

The input data set is designed to be a PRBS signal chosen randomly, both in amplitude and frequency, to fully excite the whole speed range which allows the network to recognize the overall system's behaviour. In addition, the sampling time is made to be 40 times smaller than the settling time of the system to obtain more accurate model and avoid aliasing problems. The input-output data set is shown in Fig. 8. The data set will be divided into two sets; a network training set and a model validation set.

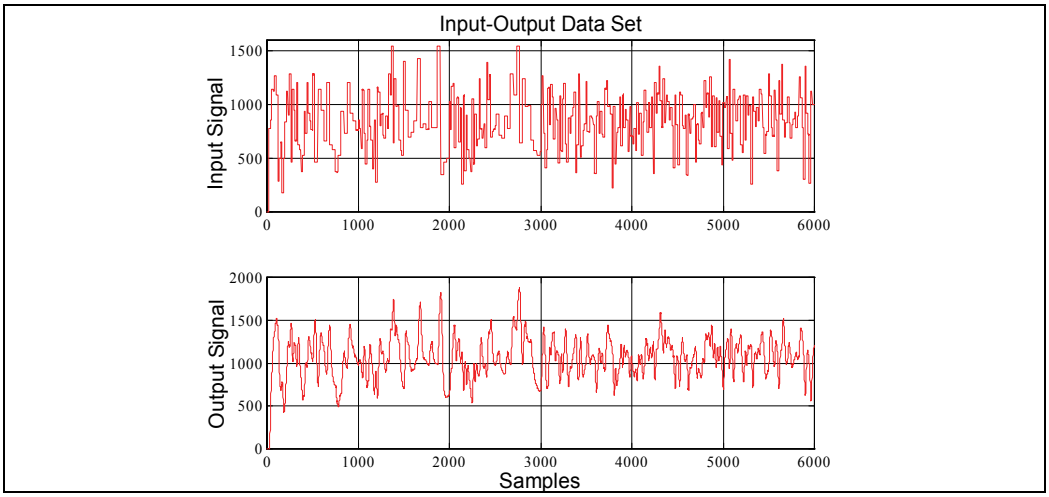


Fig. 8. The input-output data set

Since the system is a single-input single-output nonlinear system, this work uses a second order NARX model. This means that the regressor vector is as follows:

$$\varphi(t)=\left[y(t-1),y(t-2),u(t-1),u(t-2)\right] \tag{28}$$

The network structure is a two-layer hyperbolic tangent sigmoidal feed-forward architecture (one hidden layer with a tanh activation function and one output layer with a linear activation function). The weights for both hidden layer and output layers are initially randomized around the values of -0.5 and +0.5 before the training. This is useful so that the training would fall in a global minima rather than a local minima (Patterson, 1996).
Too many hidden neurons can cause the over-fitting, while too few neurons cause the under-fitting (Patterson, 1996). Moreover, a big network (many neurons) causes the training process to become very slow. The training showed good results when a five hidden neurons is used and 3000 samples are used as a training set. During each back propagation iteration the Sum of Squared Errors (SSE) are computed and compared to an error criteria α , i.e.

$$SSE=\sum_{i=1}^{N}\left[y(t)-\hat{y}(t)\right]^2<\alpha \tag{29}$$

The SSE decreased gradually during the training process until it is within the criteria threshold after approximately 370 iterations. To test whether the network can produce the same output as the plant or not, and considering the over-fitting problem, the output $\hat{y}(t\,|\,\theta)$ of the model will be compared with the plant output $y(t)$ to calculate the residuals $\varepsilon(t)=y(t)-\hat{y}(t\,|\,\theta)$. The results of applying both training and validation data sets are shown in Table 1.

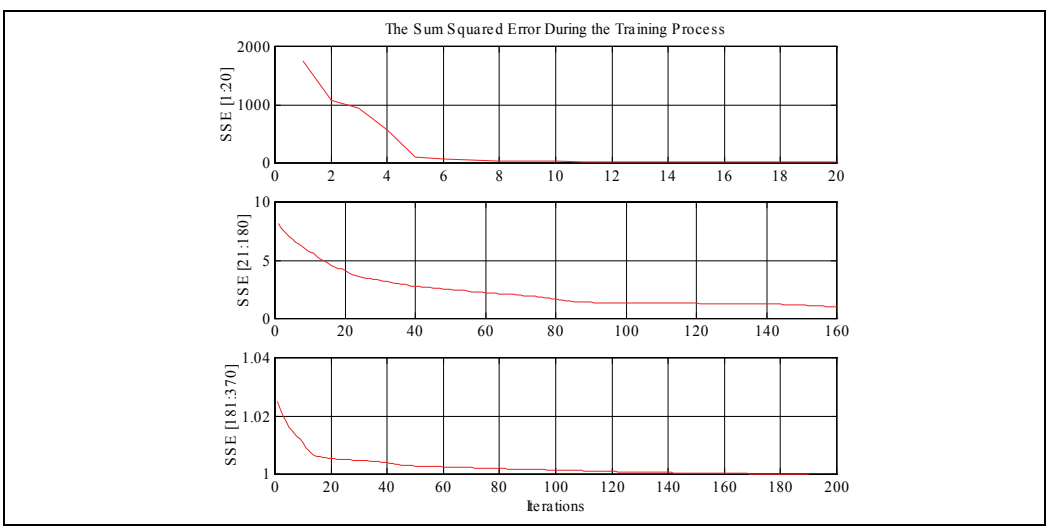


Fig. 9. The Sum Squared Error during the training process

Residual Statistics	Training data set	Validation data set
Mean Square S_N^{ε}	$2.869 \cdot 10^{-4}$	$2.936 \cdot 10^{-4}$
Maximal Absolute Value M_N^{ε}	2.9692%	3.0286%

Table 1. Residual Analysis

From the table we can see that there are only small differences in the residual statistics between the training data set and the validation data set. Thus the inequality of Eq. (21) is satisfied. However, one should check the correlation ξ_N^M between the residuals $\varepsilon(t,\theta)$ and past inputs $u(t-1),\cdots,u(t-m)$ because the residual statistics are not enough to judge the quality of the network model. This is done by constructing the past input vector and then calculating the correlation function.

The correlation results are shown in Fig. 9, where it can be seen that the auto-correlation of the residuals lies within the 99% confidence limits which gives a strong indication that the model is acceptable. Furthermore, we can see that the cross correlation between the past inputs and the residuals lies between the 99% confidence limits also.

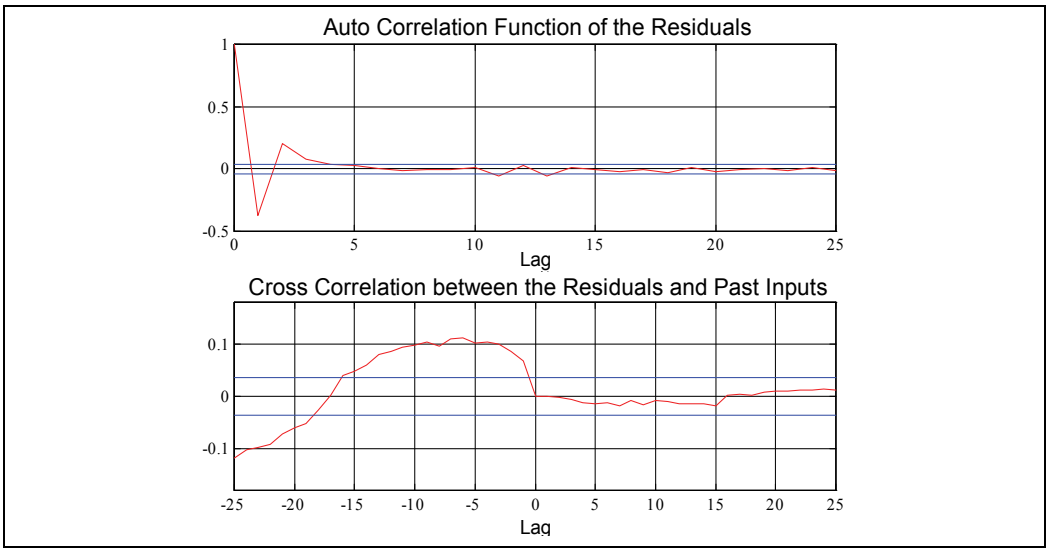


Fig. 9. Correlation Analysis of the Validation Data Set

5.2 Results of inverse training and control

As mentioned earlier, it is clear that the plant is a single-input single-output (SISO) system. First the regressors are chosen based on inspiration from linear system identification. The model order was chosen as a second order which gave us good results. Clearly, the input vector to the network contains two past plant outputs and two past plant inputs.

$$Z^N = \left[y(t-1), y(t-2), u(t-1), u(t-2) \right] \tag{30}$$

The network structure is a two layer hyperbolic tangent sigmoidal feed-forward architecture (one hidden layer with a tanh activation function and one output layer with a linear

activation function). The network weights are initially randomised around the values -0.5 and +0.5 before the training.

The back-propagation training showed good results when using a network structure with two layer feed forward architecture neuron and 3000 samples as a training set. The network architecture contains one hidden layer with a hyperbolic tangent (tanh) activation function and one output layer with a linear activation function. The hidden layer consists of six hidden neurons while the output layer consists of one neuron. The results of the inverse plant model training algorithm is shown in Fig. 10 where α is chosen as 1.

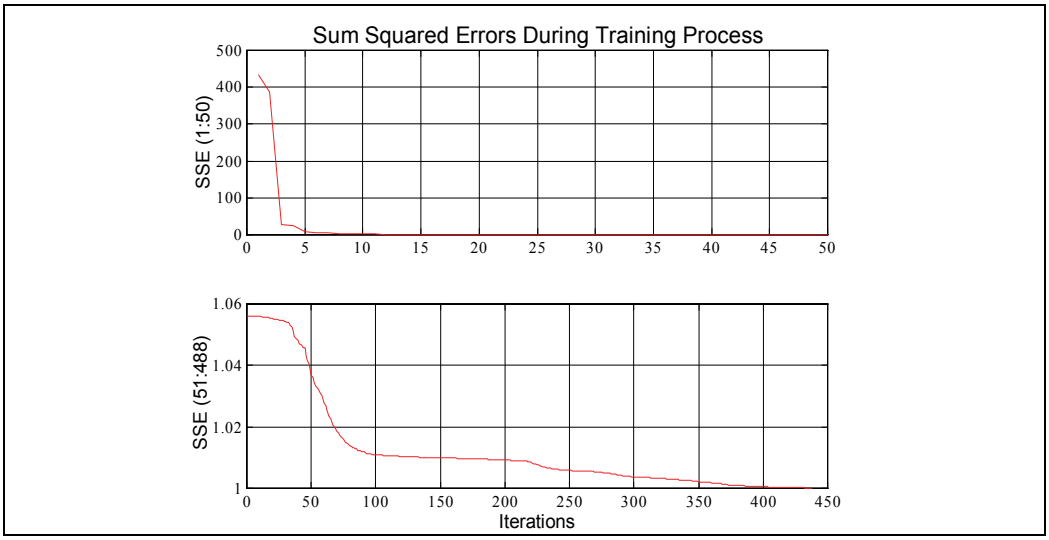


Fig. 10. SSE of inverse plant modelling

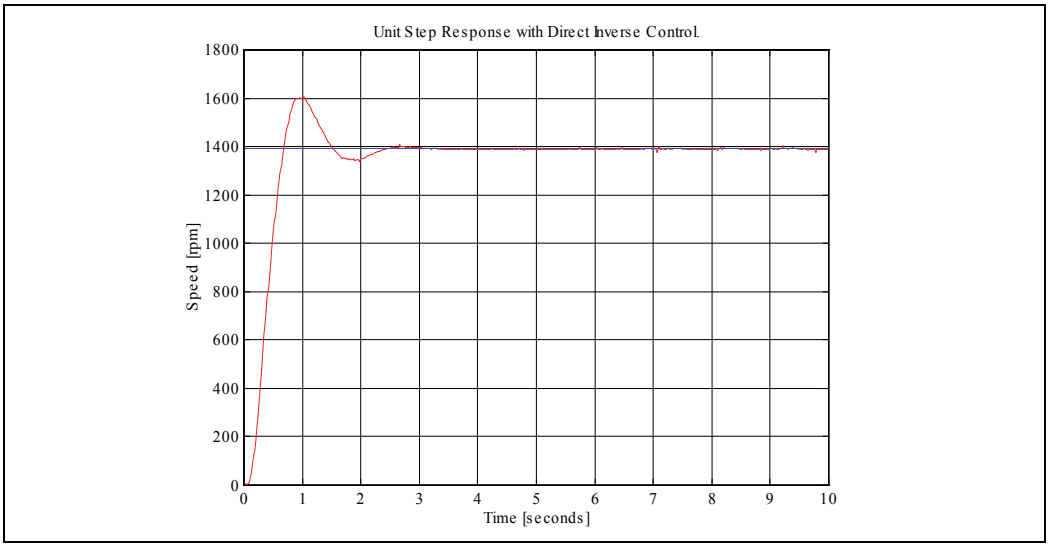


Fig. 11. Speed error due to a step reference signal

The final step after obtaining the inverse model is to implement the controller. The same setup of Fig. 7 is used to control the speed of the motor. First, to check the controller performance, a step input signal with the value of 1390rpm is fed to the system. The resulting response and error between the reference signal and the measured output speed

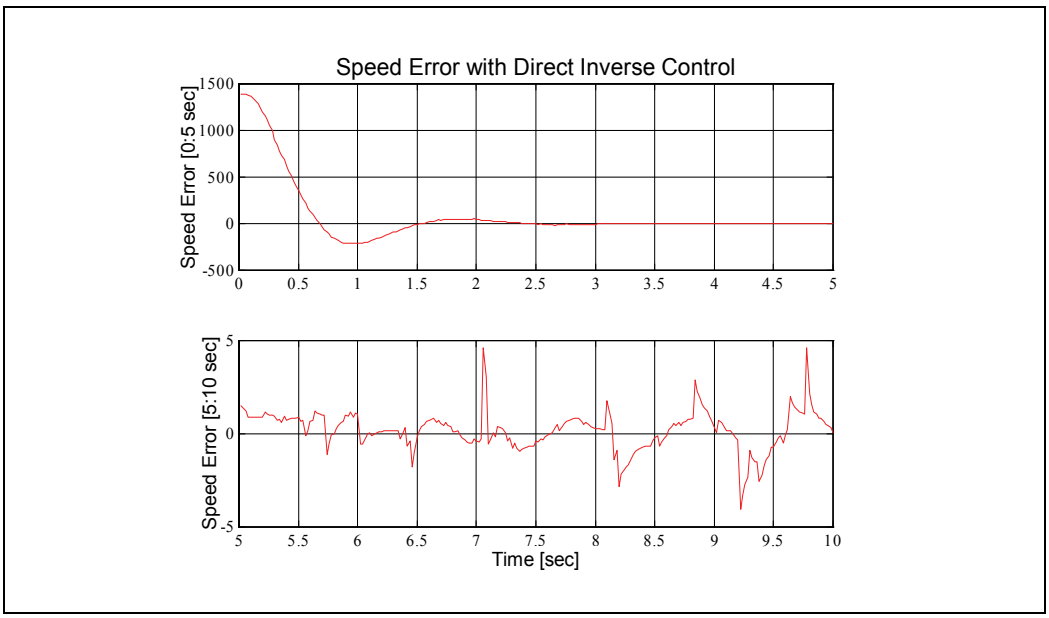


Fig. 12. Speed error due to a step reference signal

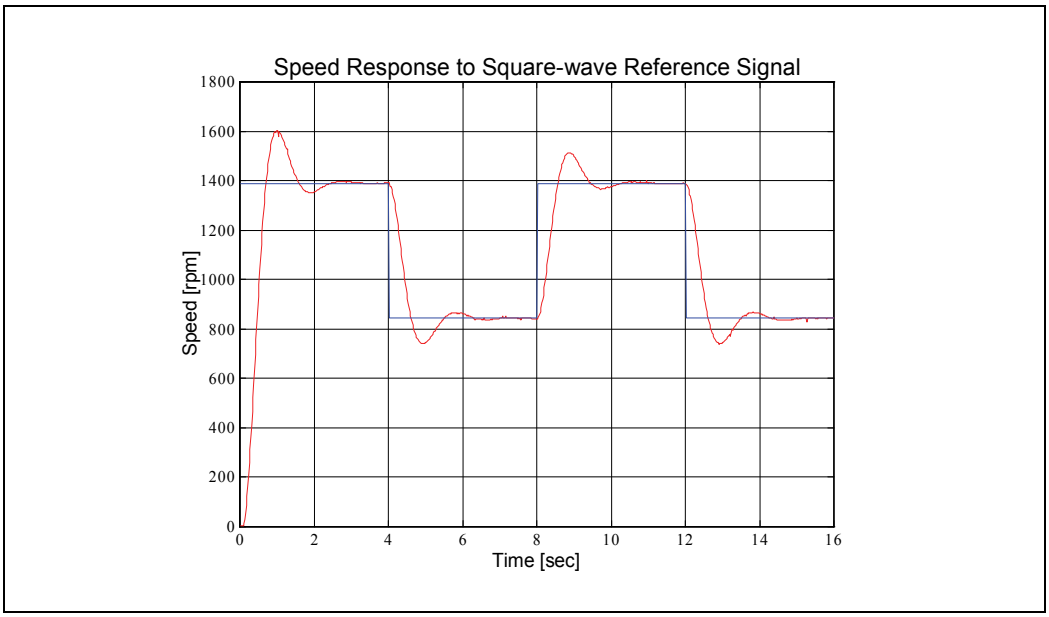


Fig. 13. System response to a square wave reference signal

are illustrated in Figs. 11 and 12 respectively. It can be seen from Fig. 12 that the speed of the induction motor followed the reference signal with an acceptable steady state error equals to 0.2878%. The results of Figs. 11 and 12 also show a maximum overshoot of less than 13%.
To investigate the tracking capabilities of the system, different reference signals were fed to the controller and its performance is examined. The following real time tests will explore

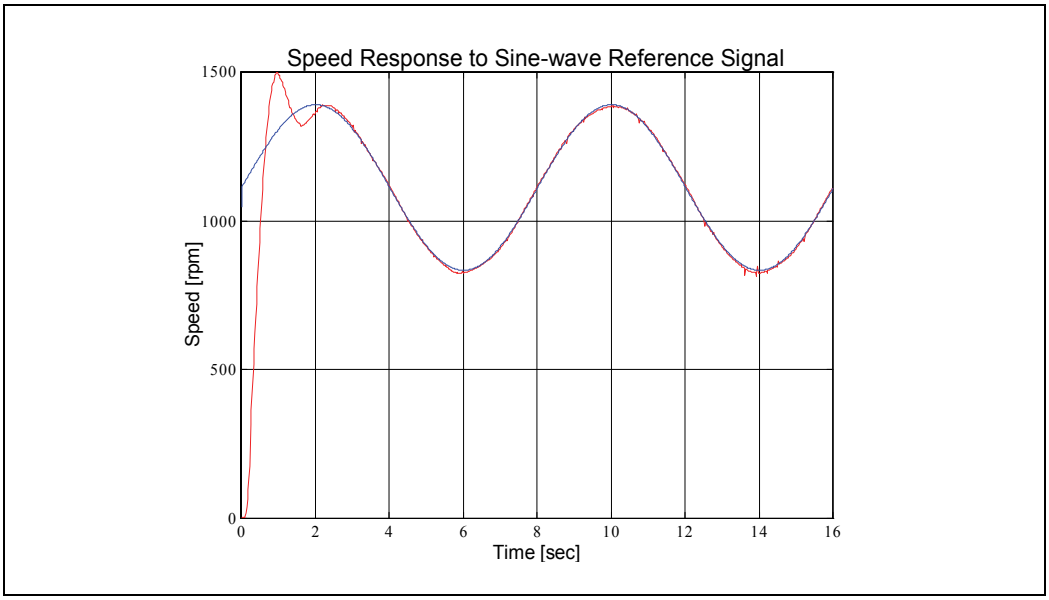


Fig. 14. System response to a sine wave reference signal

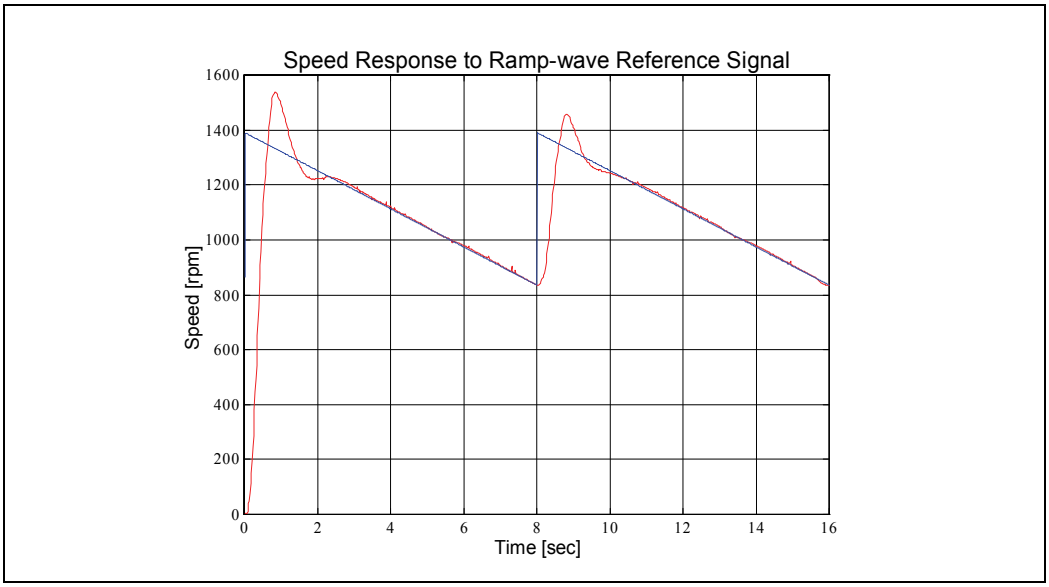


Fig. 15. System response to a saw-tooth wave reference signal

the response to three different types of speed reference signals; square wave (Fig. 13), sine wave (Fig. 14), and saw-tooth wave (Fig. 15) reference signals. In addition, the steady state errors are recorded in Table 2.

Reference Signal	Min. Error	Max. Error
Square wave	–0.31%	+0.56%
Sine wave	–0.43%	+1.00%
Saw-tooth wave	–0.65%	+0.29%

Table 2. Steady state errors analysis for different reference signal types

The previous figures suggest that the direct inverse model control scheme can track changes in the reference signal while maintaining good performance.
Next, to test the system under disturbances in the form of load torque conditions, a step reference signal representing 1390 rpm is fed to the system while a load torque step signal of 2 N.m (which is the full load) is applied to the shaft during the period of 4 to 8 seconds. The results are shown in Fig. 16. It can be seen from the figure that the direct inverse controller could recover the disturbance caused by the applied load torque. The induction motor speed followed the reference signal in a short time.

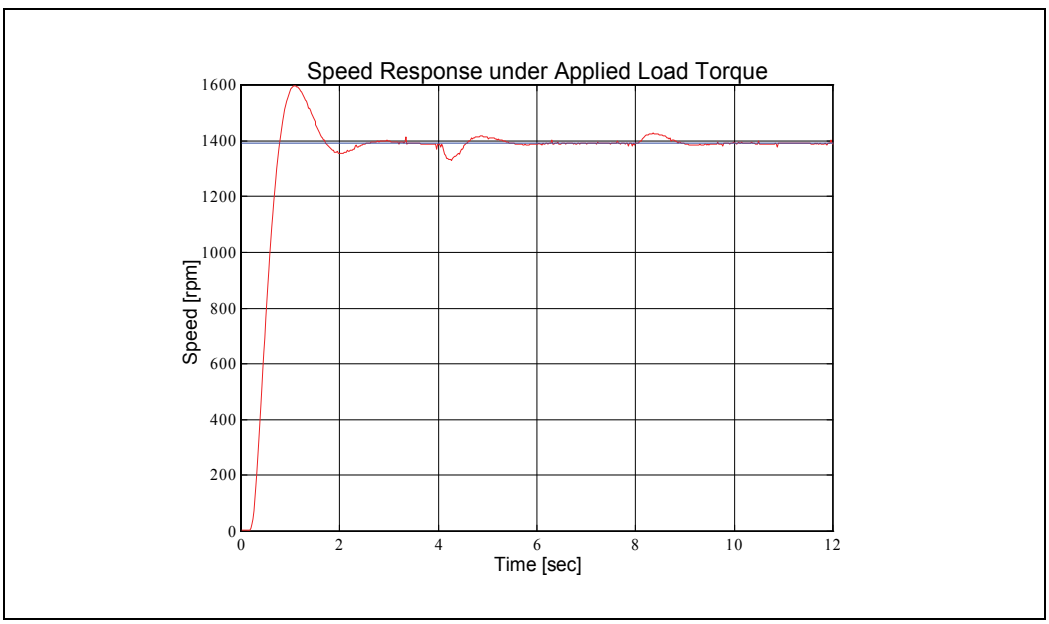


Fig. 16. Speed response under load torque condition

6. Conclusion

In this chapter, the nonlinear black box modelling for an induction motor is carried out using the back propagation training algorithm. Half of the experimentally collected data was employed for ANN training and the other half was used for model validation. Applying the validation tests, the network model could pass the residual tests and the cross correlation tests resulting into a simple yet a highly accurate model of the induction motor. The same method was then used to model the inverse model of the system. The real time implementation for the direct inverse neural network based control scheme has been presented and its performance has been tested over different types of reference signals and applied load torque. The controller tracked the given reference speed signals and overcame the applied load torque disturbance demonstrating the strong capabilities of artificial neural networks in nonlinear control applications.

7. References

Bavarian B. (1988). Introduction to Neural Networks for Intelligent Control, *IEEE Control Systems Magazine*, Vol. 8, No. 2, pp. 3-7.

Ebrahim Osama S., Mohamed A. Badr, Ali S. Elgendy, and Praveen K. Jain,(2010). ANN-Based Optimal Energy Control of Induction Motor Drive in Pumping Applications. *IEEE Transactions On Energy Conversion*, Vol. 25, No. 3, (Sept. 2010), pp. 652-660.

Fukuda T. & Shibata T. (1992). Theory and Application of Neural Networks for Industrial Control Systems, *IEEE Trans. on Industrial Electronics*, Vol. 39, No. 6, pp. 472-489.

Henneberger G. and B. Otto (1995). Neural network application to the control of electrical drives. *Proceeding of Confrence of Power Electronics Intelligent Motion*, pp. 103–123, Nuremberg, Germany.

Karanayil B.; Rahman M. F. & Grantham C. (2003). Implementation of an on-line resistance estimation using artificial neural networks for vector controlled induction motor drive, Proceeding of *IECON '03 29th Annual Conference of the IEEE Industrial Electronics Society*, Vol. 2, pp. 1703-1708.

Kulawski G. and Mietek A. Brdyś (2000). Stable adaptive control with recurrent networks. *Automatica A Journal of IFAC, the International Federation of Automatic Control*, vol. 36, No. 1, (Jan. 2000), pp. 5–22.

Kung Y. S., C. M. Liaw, and M. S. Ouyang (1995). Adaptive speed control for induction motor drive using neural network. *IEEE Transactions on Industrial Electronics*, vol. 42, no. 1, (Feb. 1995), pp. 9–16.

Lippman R. P. (1987). An Introduction to Computing with Neural Nets, *IEEE ASSP Magazine*, Vol. 4, pp. 4-22.

Ljung L. & Guo L. (1997). Classical model validation for control design purposes, *Mathematical Modelling of Systems*, 3, 27–42.

Ljung L. & Sjöberg J. (1992). A System Identification Perspective on Neural Nets, Technical Report, Report No. LiTH-ISY-R-1373, At the location: www.control.isy.liu.se, May 27.

Ljung L. (1995). System Identification, Technical Report, Report No. LiTH-ISY-R-1763, At the location: www.control.isy.liu.se.

Ma X. & Na Z. (2000). Neural network speed identification scheme for speed sensor-less DTC induction motor drive system, *PIEMC 2000 Proceeding. 3rd Int. Conference on Power Electronics and Motion Control*, Vol. 3, pp. 1242-1245.

Mehrotra P. ; Quaicoe J. E. & Venkatesan R. (1996a). Development of an Artificial Neural Network Based Induction Motor Speed Estimator, *PESC '96 IEEE Power Electronics Specialists Conference*, Vol. 1, pp. 682-688.

Mehrotra P.; Quaicoe J. E. & Venkatesan R. (1996b). Induction Motor Speed Estimation Using Artificial Neural Networks, *IEEE Canadian Conference on Electrical and Computer Engineering*, Vol. 2, pp. 607-610,

Merabet Adel, Mohand Ouhrouche and Rung-Tien Bui (2006). Neural Generalized Predictive Controller for Induction Motor, *International Journal of Theoretical and Applied Computer Sciences*, Vol. 1, 1 (2006), pp. 83–100.

Mohamed H. A. F.; Yaacob S. & Taib M. N. (1997). Induction Motor Identification Using Artificial Neural Networks, *APEC 97 Electric Energy Conference,* pp. 217-221, 29-30th Sep..

Ninness B. & Goodwin G. C. (1994). Estimation of Model Quality, *10th IFAC Symposium Proceeding. on System Identification*, 1, pp. 25-44.

Orłowska-Kowalska T. and C. T. Kowalski (1996). Neural network based flux observer for the induction motor drive. *Proceedingceding of International Confrence of Power Electronics Motion Control,* pp. 187–191, Budapest, Hungary, 1996.

Patterson D. W. (1996). Artificial Neural Networks: Theory and Applications, Simon and Schuster (Asia) Pte. Ltd., Singapore: Prentice Hall.

Raison B., F. Francois, G. Rostaing, and J. Rogon (2000). Induction drive monitoring by neural networks. *Proceeding of IEEE International Conference of Industrial Electronics,Control Instrumentation*, pp. 859–863, Nagoya, Japan, 2000.

Sharma A. K., R. A. Gupta, Laxmi Srivastava (2007). Performance of Ann Based Indirect Vector Control Induction Motor Drive, *Journal of Theoretical and Applied Information Technology*, Vol. 3, No. 3, (2007), pp 50-57.

Simoes M. G. and B. K. Bose (1995).Neural network based estimation of feedback signals for a vector controlled induction motor drive. *IEEE Transactions on Industry Applications.,* vol. 31, no. 3, (May/Jun. 1995), pp. 620–629.

Sjöberg J.; Zhang Q., Ljung L., Benveniste A., Deylon B., Glorennec P. Y., Hjalmarsson H., & Juditsky A. (1995). Nonlinear Black-Box Models In System Identification: A Unified Overview, *Automatica*, Vol. 31, No. 12, pp. 1691-1724.

Toqeer R. S. & Bayindir N. S. (2000). Neurocontroller for induction motors, ICM 2000. Proceeding. *12th Int. Conference on Microelectronics*, pp. 227-230.

Vas P. (1990). Vector Control of AC Machines, Clarendon Press, Oxford.

Weber M.; Crilly P. B. & Blass W. E. (1991). Adaptive Noise Filtering Using an Error-Backpropagation Neural Network, *IEEE Trans. Instrum. Meas.*, Vol. 40(5), pp. 820-825.

Wishart M. T. & Harley R. G. (1995). Identification And Control Of Induction Machines Using Artificial Neural Networks, *IEEE Transactions on Industry Applications*, Vol. 31(3), pp. 612-619.

Wlas M.; Krzeminski, Z.; Guzinski, J.; Abu-Rub, H.; Toliyat, H.A. (2005). Artificial-Neural-Network-Based Sensorless Nonlinear Control of Induction Motors, *IEEE Transection of Energy conversion*, Vol. 20, 3 (Sept 2005), pp. 520-528.

Yabuta T. & Yamada T. (1991). Learning Control Using Neural Networks. *Proceeding of the 1991 IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 740-745.

19

System Identification of NN-based Model Reference Control of RUAV during Hover

Bhaskar Prasad Rimal[1], Idris E. Putro[2], Agus Budiyono[2],
Dugki Min[3] and Eunmi Choi[1]
[1]*Graduate School of Business IT, Kookmin University*
Jeongneung-Dong, Seongbuk-Gu, Seoul, 136-702, Korea
[2]*Department of Aerospace Information Engineering, Konkuk University*
[3]*School of Computer Science and Engineering, Konkuk University*
Hwayang-dong, Gwangjin-gu, Seoul 13-701,
Korea

1. Introduction

Unmanned aerial vehicles (UAVs) are becoming more and more popular in a wide field of applications nowadays. UAVs are used in number of military application for gathering information and military attacks. In the future will likely see unmanned aircraft employed, offensively, for bombing and ground attack. As a tool for research and rescue, UAVs can help find humans lost in the wilderness, trapped in collapsed buildings, or drift at sea. It is also used in civil application in fire station, police observation of crime disturbance and natural disaster prevention, where the human observer will be risky to fight the fire. There is wide variety of UAV shapes, sizes, configuration and characteristics. Therefore, there is a growing demand for UAV control systems, and many projects either commercial or academic destined to design a UAV autopilot were held recently. A lot of impressive results had already been achieved, and many UAVs, more or less autonomous, are used by various organizations.

An Artificial Neural Network (ANN) [3] is an information processing paradigm that is stimulated by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. Basically, a neural network (NN) is composed of a set of nodes (Fig. 1). Each node is connected to the others via a set of links. Information is transmitted from the input to the output cells depending of the strength of the links. Usually, neural networks operate in two phases. The first phase is a learning phase where each of the nodes and links adjust their strength in order to match with the desired output. A learning algorithm is in charge of this process. When the learning phase is complete, the NN is ready to recognize the incoming information and to work as a pattern recognition system.

ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons.

In recent years, there is a wide momentum of ANNs in the control system arena, to design the UAVs. Any system in which input is not proportional to output is known as non-linear systems. The main advantages of ANNs are having the processing ability to model nonlinear systems. ANNs are very suitable for identification of non-linear dynamic systems. Multilayer Perceptron model have been used to model a large number of nonlinear plants. We can vary the number of hidden layers to minimize the mean square error. ANNs has been used to formulate a variety of control strategies [1] [2]. The NN approach is a good alternative for physical modeling techniques for nonlinear systems.

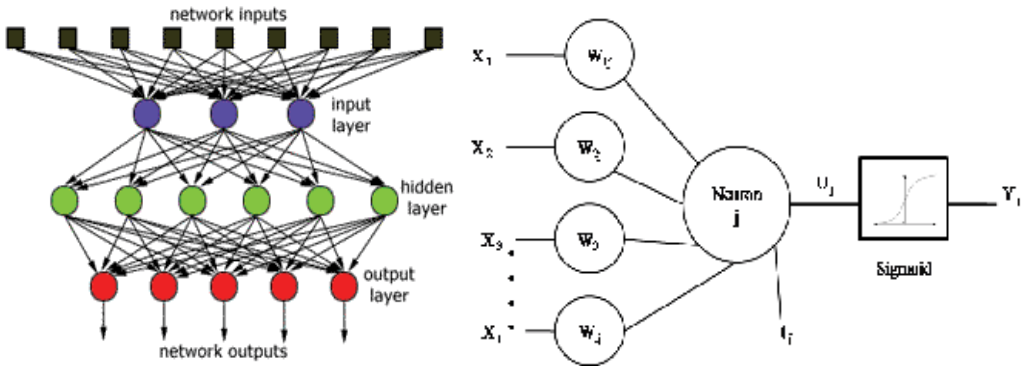


Fig. 1. General Neural Network Architecture

A fundamental difficulty of many non-linear control systems, which potentially could deliver better performance, is extremely difficult to theoretically predict the behavior of a system under all possible circumstances. In fact, even design envelope of a controller often remains largely uncertain. Therefore, it becomes a challenging task to verify and validate the designed controller under all possible flight conditions. A practical solution to this problem is extensive testing of the system. Possibly the most expensive design items are the control and navigation systems. Therefore, one of main questions that each system designer has to face is the selection of appropriate hardware for UAV system. Such hardware should satisfy the main requirements without contravening their boundaries in terms of quality and cost. In UAV design this kind of consideration is especially important due to the safety requirements expressed in airworthiness standards. Therefore question is how to find the optimal solution. Thus, simulation is necessary. Basically there are two type of simulation is needed while designing UAVs systems, they are Software-In-the-Loop (SIL) [5] simulation and Hardware-In-the-Loop (HIL) simulation [4].

To utilize the SIL configuration, the un-compiled software source code, which normally runs on the onboard computer, is compiled into the simulation tool itself, allowing this software to be tested on the simulation host computer. This allows the flight software to be tested without the need to tie-up the flight hardware, and was also used in selection of hardware.

HILS simulates (Fig. 2) a process such that input and output signals show the time-dependent values as real-time operating components. It is possible to test embedded system under real time with various test conditions. It provides the UAV developer to test many aspects of autopilot hardware, finding the real time problems, test the reliability, and many more.

The simulation can be done with the help of Matlab Simulink program environment. This program can be considered as a facility fully competent for this task. Simulink is the most

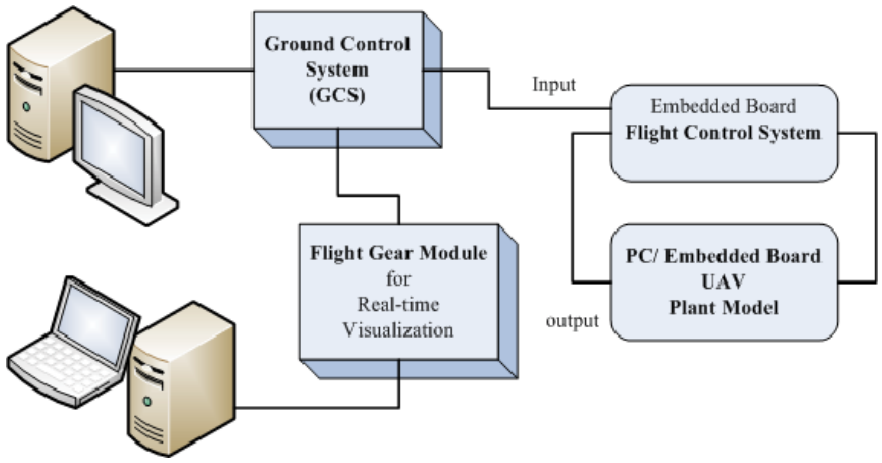


Fig. 2. UAV Architecture: Hardware-in-the-loop Simulation

popular tool, it was not only used for a SIL Simulation of the complete UAV system but also to create the simulation code of a HIL Simulator that runs in real time.

The system identification is the first and crucial step for the design of the controller, simulation of the system and so on. Frequently it is necessary to analyze the flight data in the frequency domain to identify the UAV system. This paper demonstrates how ANN can be used for non linear identification and controller design. The simulation processes consists of designing a simple system, and simulates that system with the help of model reference control block in Matlab/Simulink [6].

The paper is organized as follows: Section 2 describes some related work. Section 3 deals with system identification and control on the basis of NNs. Details of design and control system with NNs approaches is describes in section 4. In section 5, simulations are performed on RUAVs system and finally, conclusions are drawn in section 6.

2. Related work

Robust control techniques are capable for adapting themselves for changing the dynamics which are necessary for autonomous flight. This kind of controller can be designed with the help of system identification.

There are lots of work had already done in UAV area in the context of ANNs. Mettler B. et al., [12] describe the process and result of the dynamic modeling of a model-scale unmanned helicopter using system identification. E. D. Beckmann et al., [13] explained the nonlinear modeling of a small-scale helicopter and the identification of its dynamic parameters using prediction error minimization methods. NN approaches have excellent performance than classical technique for modeling and identifying nonlinear dynamic systems [15] [16].

There is also numerous system identification techniques had been developed to model nonlinear systems. Some of them are Fuzzy identification [20] [27], state-space identification [21], frequency domain analysis [22], NN based identification [23] [26]. The exception is given by LPV identification [25] which is applicable for the entire flight envelope. The learning ability is the beauty of NN that has been utilized widely for system identification and control applications. Shim D. H. et al. [28] described time-domain system identification approaches to design the control system for RUAVs.

3. System identification and control

The main idea of system identification is often to get a model that can be used for controller design. System identification (SI) [7] provides the idea of making mathematical models of dynamics systems, starting from experimental data, measurements, and observations.
It is widely used for applications ranging from control system design and signal processing to time series analysis. The system identification is used to verify and test the control system parameters that are associated with the six-degree-of-freedom system using the test flight data. The simulation results and the statistical error analysis are provided for both the cases. Fig. 5 shows the flow of control system design with the system identification model. Basically System identification is the experimental approach to process modeling and it includes the following five steps as shown in Fig. 3
The system considered as a black box (Fig. 4) which receives some inputs that lead to some output. The concern here is: what kind of parameters for a particular black box can correlate the observed inputs and outputs?

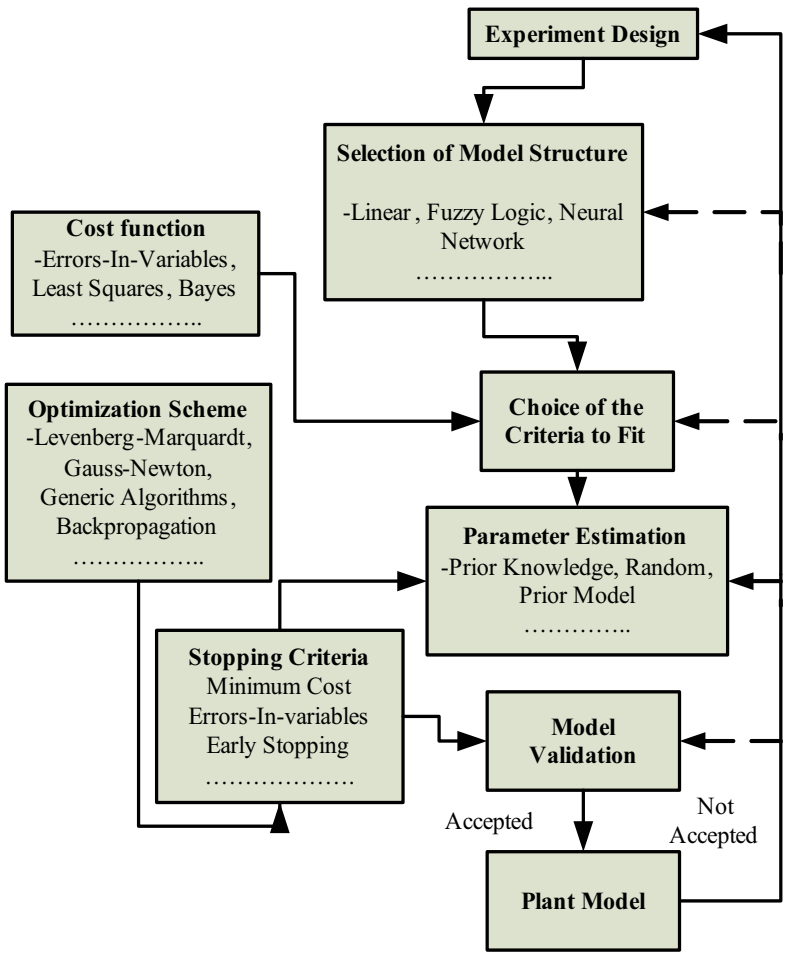


Fig. 4. System Identification Modeling Procedure

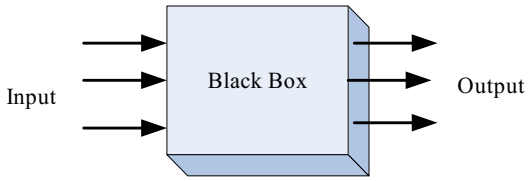


Fig. 5. Black box view of System Identification

Can these parameters help designer to predict the outputs for a new given set of inputs? This is the basic problem of system identification. Neural networks have been applied successfully in the identification and control of dynamic systems. Popular neural network architectures for prediction and control that have been implemented in the NN Toolbox™ software are:

- Model Predictive Control
- NARMA-L2 (Feedback Linearization) Control
- Model Reference Control

There are typically two steps involved when using neural networks for control systems:

- System identification
- Control design

In the system identification stage, we develop a neural network model of the plant that we want to control. The flow of control system design with system identification is shown in Fig 5. In the control design stage, we use the neural network plant model to design (or train) the controller using the propagation of the controlling error through the NN model. Training produces the optimal connection weights for the networks by minimizing the errors between NN output and the plant output over the entire set of samples. Among many network training algorithms Levenberg-Marquardt (LM) algorithm [14] is performed. This approach provides a gradient based technique allowing fast error minimization. The major aim of training is to get the appropriate values of the weights for closest possible prediction through repetitive iterations. The LM method works on the principle of minimizing the mean squared error between actual output of the system and predicted output of the network and can be calculated with the following formula.

$$V_N(\theta) = \frac{1}{N}\sum_{t=1}^{N}\left(y(t) - \hat{y}(t)\right)^2 \tag{1}$$

Where

$$\hat{y}(t) = g(\theta, \phi(t)) \tag{2}$$

$$\theta = (a_1, a_2, \ldots a_{na}, b_1, b_2, b_{nb}) \tag{3}$$

$$\phi(t) = (y(t-1), \ldots y(t-na), u(t-nk), \ldots u(t-nk-nb+1)) \tag{4}$$

Here ϕ is the matrix of past inputs and outputs. To find the coefficient θ, there are many assumptions and detailed knowledge of the plant is required.

In each of the three control architectures mention above, system identification stage is identical but control design stage is different. For model predictive control, the plant model is used to predict future behavior of the plant, and an optimization algorithm is used to select the control input that optimizes future performance.

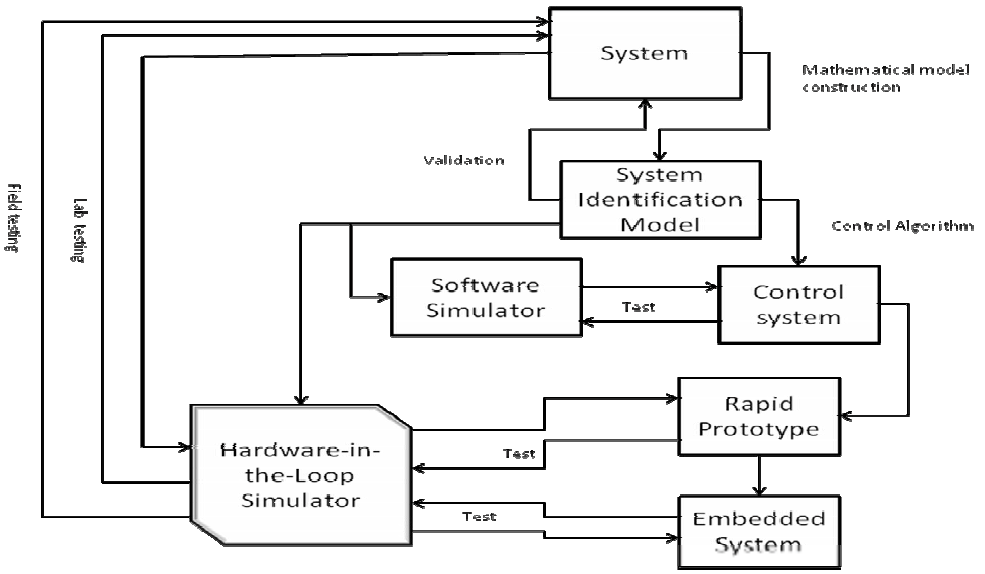


Fig. 5. Flow of Control System Design with System Identification

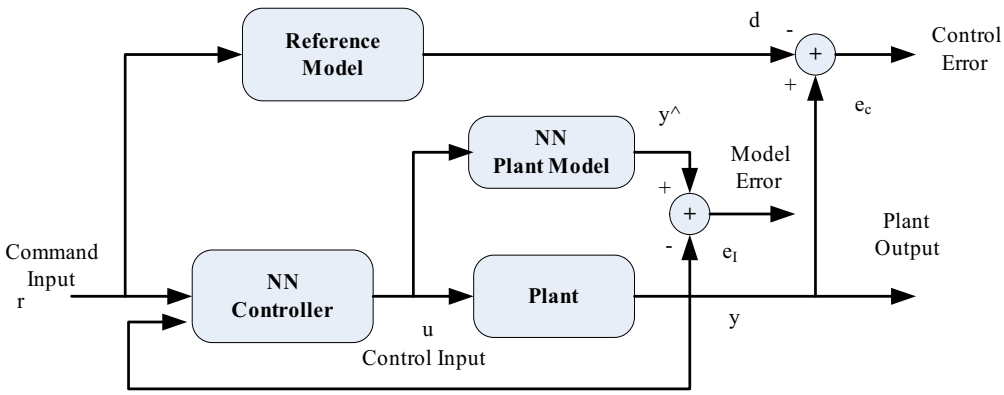


Fig. 6. Neural Network MRC architecture

For NARMA-L2 control, the controller is simply a rearrangement of the plant model. We used model reference control to simulate the nonlinear identification and control of UAV. For model reference control, the controller is a neural network that is trained to control a plant so that it follows a reference model. The neural network plant model is used to assist in the controller training.

The neural model reference control architecture uses two neural networks: a controller network and a plant model network, as shown in the Fig. 6. The plant model is identified first, and then the controller is trained so that the plant output follows the reference model output. The system identification error can be defined by

$$e_I = y - \hat{y} \tag{5}$$

and the tracking error

$$e_c = y - d \tag{6}$$

Controller parameters are updated based on error computed from the system output and the NN model of the plant.
They describe the input /output behavior of the system using a set of weights. Such models can be interpreted as a weighted combination of several local models resulting in a nonlinear global model. Hence the mismatch between the nonlinearities of local models and process is less significant compared with single nonlinear model. Therefore neural network modeling has been applied especially to modeling tasks with uncertain nonlinearities, uncertain parameters and or high complexity.
In the case of multi-input multi-output (MIMO) plants, the plant identification stage is same as that for single-input single-output (SISO) except that the NN model has many neurons in the output layer as the number of outputs of actual plant. Fig. 7 shows the control of a MIMO plant using NN controller.

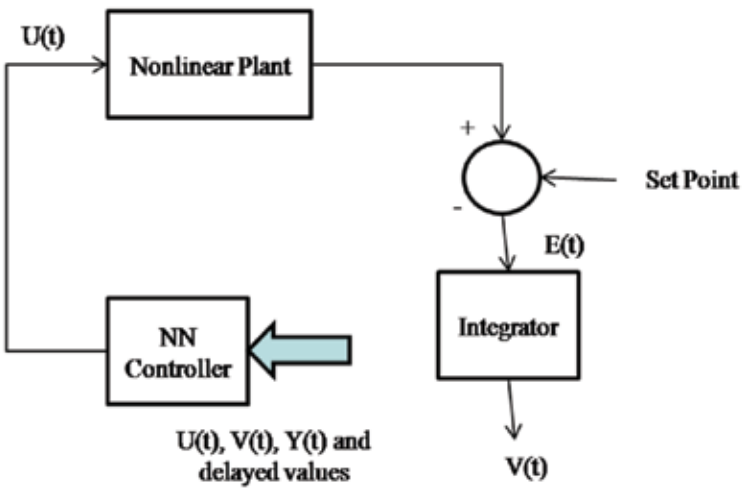


Fig. 7. Control of a MIMO plant using NN controller

3.1 Training of model using NN

Iterative training is conducted to minimize mean square error (MSE) using Levenberg Marquardt (LM) algorithm. The LM is gradient based approach that allows fast error minimization. The mission of training is to obtain the most suitable and optimized values of the weights for closest prediction through iterations.
The training process (Fig 8) is an iterative and can be stopped either when total training error reaches a bottom threshold or when training error ceases to decrease any further. There is flexibility for varying number of neurons in the hidden layers to optimize error. Starting from a small number of neurons, the number can be gradually increased or decrease until an accepted training error is achieved. Once the NN is successfully trained, it can be used to obtain a linear model of the plant from the available input and output values. Figure shows the behavior of plant input and output. It can be seen that the output response is different in each time slot with the variation of input weights. As a typical case, 1000 different sets of initial weights are considered for the network with 10 hidden neurons.

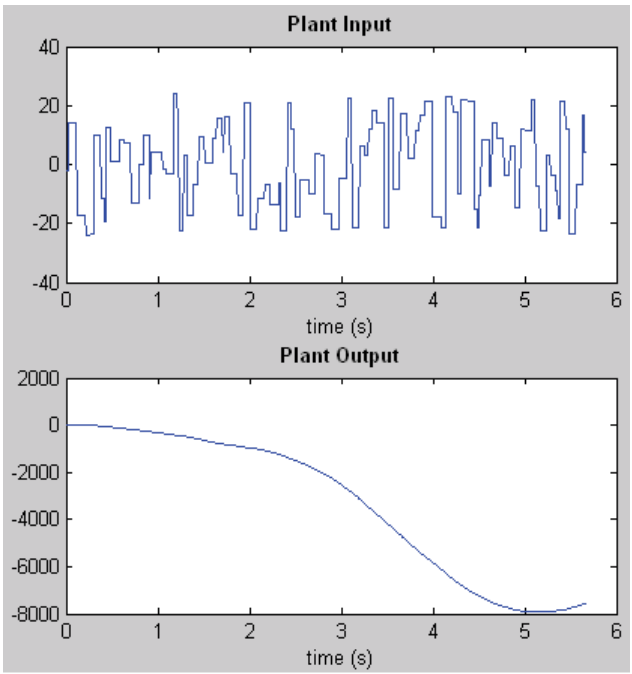


Fig. 8. Plant Input/output behavior during training

The network architecture for MIMO model of NN is shown in figure 9. In a particular equilibrium condition, no inertial and aerodynamic coupling, the behavior of RUAV can be divided into lateral and longitudinal dynamics mode and train with MIMO. The longitudinal cyclic deflection and collective control input is used to control the longitudinal dynamics mode whereas lateral cyclic deflection and pedal control input is used to control the lateral dynamics mode. The NN has trained with four system inputs and six outputs. The number of hidden layer is considered. The network is trained with different sets of data collected from the real flight tests of the RUAV.

Figure 9 describes the longitudinal dynamics mode as longitudinal cyclic deflection and collective control are provided as inputs (U) to the system and pitch rate and forward velocity (u) are considered as the outputs (Y) of the system that results four outputs pitch angle (θ), forward velocity (u),vertical velocity (w) and pitch angular rate (q). Similarly, lateral cyclic deflection and pedal control are provided as inputs to the system and vertical velocity and roll rate are considered as the outputs of the system that results roll angle (φ), lateral velocity (v), roll angular rate (p), yaw angular rate (r).

Figure 10 illustrates a scenario of getting performance behavior of the identification with plant process during training (21 Epochs), where the training data and testing data are following almost same behavior.

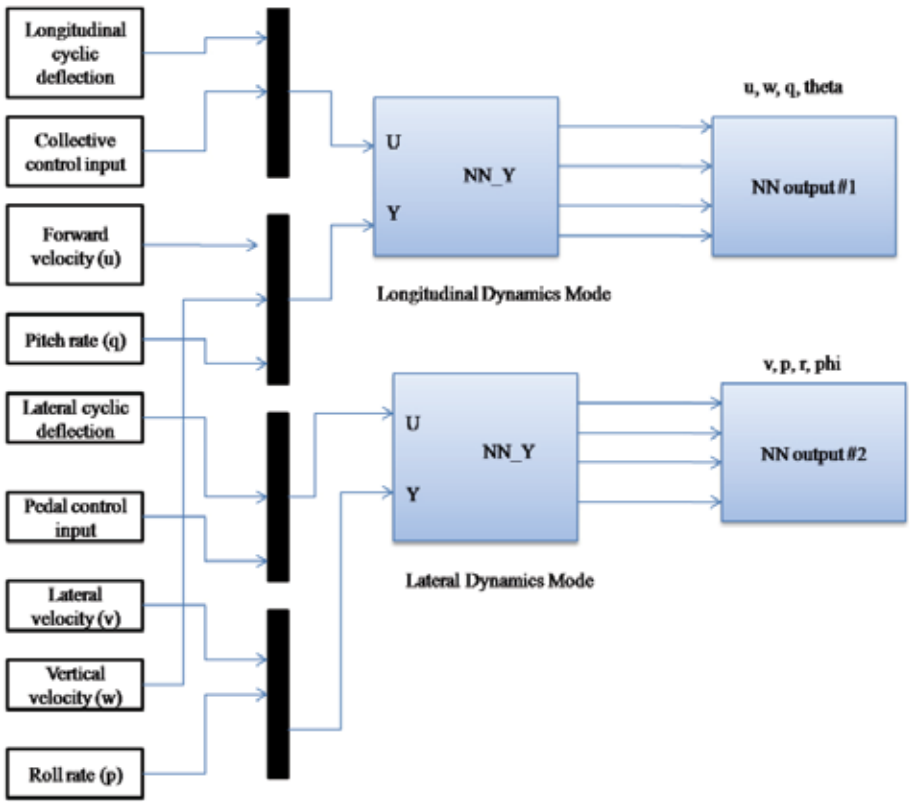


Fig. 9. MIMO model of NN

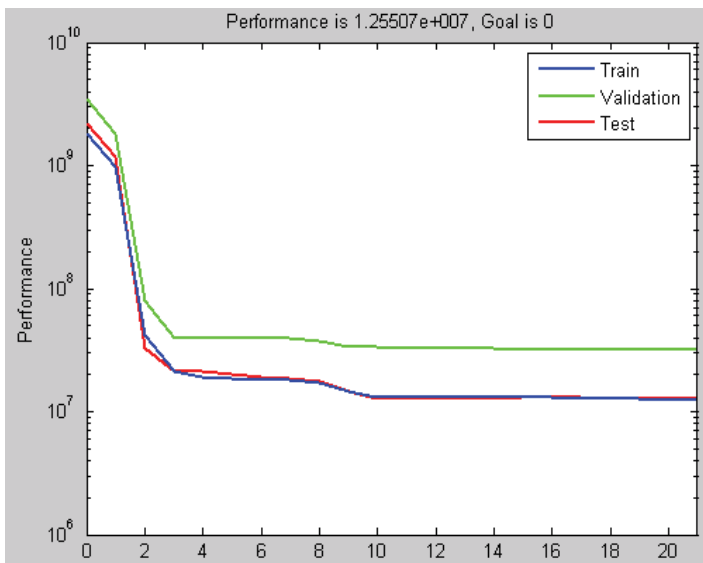


Fig. 10. A scenario of getting performance behavior of the identification with plant process during training (Trainlm at 21 Epochs), where the training data and testing data are following almost same behavior

4. Design of control system

The key challenge to deploy the designed control system of UAV is the potential risk and cost. So, to minimize the cost and the potential risk, we have to test and simulate the control system rigorously to get high degree of precision of safeness. Design, testing and simulation are the iterative process and this can be accomplished by executing several pair of design, test, and simulation. The simulation of UAV control system consist a group of nonlinear Simulink models which is used to estimate the capabilities of controllers. Generally, these models are employed both in evolutionary search to estimate the robustness of a particular controller, and later to verify and validate the designed controllers through extensive simulation with several test cases in different conditions. An UAV represents a complex non-linear system with 6 Degree of Freedom (6-DOF), and having high degree of coupling. It is anticipated that the most effective control on such a system can be gained with an appropriate non-linear controller.

NNs have attracted a great deal of attention owing to their ability to learn non-linear functions from input-output data examples [8]. Applied to control field, NNs are essentially nonlinear models that can be useful to solve non-linear control problems [9].

4.1 Mathematical model of RUAV dynamics

Basic starting point for UAV control design is to find out the state space matrix from 6-DOF equations of motion by linearizing with proper assumptions. The state of a system is a set of variables (Fig. 11) such that the knowledge of these variables and the input functions will, with the equations describing the dynamics, provide the future state and output of the system. The state of the system is described by the set of the first-order differential equations written in terms of state variables [x1 x2 ……xn].

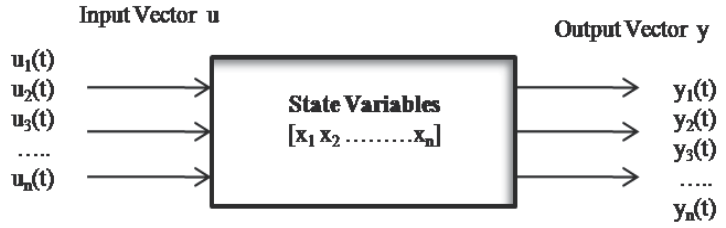


Fig. 11. System inputs and outputs

The state space is defined as the n-dimensional space in which the components of the n state vector represent its coordinate axes. The state equations of a system are a set of n first-order differential equations, where n is the number of independent states. Many control problems, however, that require multiple outputs be controlled simultaneously, to do control of such system, multiple inputs must be manipulated, usually they are orchestrated as MIMO. The helicopter is a complex MIMO system with high correlation. UAV autopilot is an example of MIMO where speed, altitude, pitch, roll, and yaw angles must be maintained and throttle, several rudders, and flaps are available as control variables.

The UAV systems consist of a six degree of freedom, nonlinear complex systems. Budiyono A. et al., [11] illustrate the nonlinear rigid body equations of motion of helicopter (Eq.7-15) that describes the vehicle's translational motion and angular motion about three reference axes.

$$\sum X = m(\vec{u} - rv + qw) + mg\sin\theta \tag{7}$$

$$\sum Y = m(ru + \vec{v} - pw) - mg\sin\phi\cos\theta \tag{8}$$

$$\sum Z = m(-qu + pv - \vec{w}) - mg\cos\phi\cos\theta \tag{9}$$

$$\sum L = I_{xx}\vec{p} - (I_{yy} - I_{zz})qr \tag{10}$$

$$\sum M = I_{yy}\vec{q} - (I_{zz} - I_{xx})pr \tag{11}$$

$$\sum N = I_{zz}\vec{r} - (I_{xx} - I_{yy})pq \tag{12}$$

$$\vec{\phi} = p + (q\sin\phi + r\cos\phi)\tan\theta \tag{13}$$

$$\vec{\theta} = q\cos\phi - r\sin\phi \tag{14}$$

$$\vec{\varphi} = (q\sin\phi + r\cos\phi)\sec\theta \tag{15}$$

Where the vector u, v, w and p, q, r are the fuselage velocities and angular rates in the body coordinate system, respectively. X, Y, Z are the external forces acting on the helicopter center of gravity and L, M, N are the external moments. State space and transfer-function models can be generalized to MIMO models. These first-order differential equations can be written in a general form that can be represented in matrix notation.

$$\begin{aligned} \vec{x} &= A\vec{x} + B\vec{u} \\ \vec{y} &= C\vec{x} + D\vec{u} \end{aligned} \tag{16}$$

Where

$$\vec{x} = \begin{bmatrix} u & w & q & \theta & a_{1s} & v & p & r & \phi & b_{1s} \end{bmatrix}' \tag{17}$$

And

$$\vec{u} = \begin{bmatrix} \delta_{long} & \delta_{coll} & \delta_{lat} & \delta_{ped} \end{bmatrix} \tag{18}$$

The MIMO transfer-function matrix can be obtained from state space model by $G(s) = C(sI - A)^{-1}B + D$ where $A \in \mathbb{R}^{n*n}, B \in \mathbb{R}^{n*m}, C \in \mathbb{R}^{l*n}, D \in \mathbb{R}^{l*m}$. The descriptions of all parameters are shown in Table 1 and Table 2.

Where A, B and C are the representation of the system matrix, input matrix and output matrix respectively. A, B, C and D depends on the flight regime with nominal parameter values for hovering and cruising. Then, u is a vector of the inputs, x is the element state vector, and y is a vector containing outputs. It is easy to see that each linear state space system of Equation (16) can be expressed as a linear time invariant (LTI) transfer functions. The procedure is to take Laplace transformation of the both sides of Equation (16) and use an algorithm is given by Leverrier-Fadeeva-Frame formula [10]. Another approach is to use Matlab functions directly. Let us take a transfer function of UAV (Eq. 19-20) to model and simulate.

Parameter	Symbol	Description
Fuselage Linear Motion	u	Forward velocity
	v	Lateral velocity
	w	Vertical velocity
Fuselage Angular Motion	p	Roll Angular Rate
	q	Pitch Angular Rate
	r	Yaw Angular Rate
Rotor Tip-Path-Plane	a1s	Longitudinal Flapping Angle
	b1s	Lateral Flapping Angle
Pitch	θ	Pitch Angle
Roll	φ	Roll Angle

Table 1. Model states.

Control	Description	Units
δ_{long}	Longitudinal Cyclic Deflection	Dimensionless [-1, 1]
δ_{lat}	Lateral Cyclic Deflection	Dimensionless [-1, 1]
δ_{ped}	Pedal control Input	Dimensionless [-1, 1]
δ_{coll}	Collective Control Input	Dimensionless [0, 1]

Table 2. Control input variables.

$$A = \left[\begin{array}{ccccc|ccccc} X_u & 0 & 0 & -g & -g & 0 & 0 & 0 & 0 & 0 \\ 0 & Z_w & 0 & 0 & Z_{a1s} & 0 & 0 & Z_r & 0 & Z_{b1s} \\ M_u & 0 & 0 & 0 & M_{a1s} & M_v & 0 & 0 & 0 & M_{b1s} \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & A_{a1s} & 0 & 0 & 0 & 0 & A_{b1s} \\ \hline 0 & 0 & 0 & 0 & 0 & Y_v & 0 & 0 & g & g \\ L_u & 0 & 0 & 0 & L_{a1s} & L_v & 0 & 0 & 0 & L_{b1s} \\ 0 & N_w & 0 & 0 & 0 & 0 & N_p & N_r & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & B_{a1s} & 0 & -1 & 0 & 0 & B_{b1s} \end{array}\right] \tag{19}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ Z_{col} & 0 & 0 & 0 \\ M_{col} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & A_{lon} & 0 & A_{lat} \\ 0 & 0 & Y_{ped} & 0 \\ 0 & 0 & 0 & 0 \\ N_{col} & 0 & N_{ped} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & B_{lon} & 0 & B_{lat} \end{bmatrix} \tag{20}$$

$$C = \left[\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \tag{21}$$

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{22}$$

The descriptions of all parameters used in Eq. 19-20 are shown in Table 3

Parameter	Description
Zw Zr Zbls Zals Yv Xu Nw Nr Np Mv Mu Mbls Mals Lv Lu Lals Lbls Bals Bbls Aals Abls	Stability derivative
g	Force of gravity
Zcol Yped Ncol Nped Mcol	Control derivative
Blon Blat Alat Alon	Cyclic input sensitivity

Table 3. Parameters of model constants for fuselage linear motion equations, model constants for tip-path-plane and augmented yaw dynamics, and model constants for angular motion.

The objective of training a NN is to minimize the error between the output of NN and the desired output. First, we use models (Eq. 19-20) to generate training data. Then by propagation algorithm, all the weights in NN plant can be adjusted through the training sets until the NN plant outputs are very close to the plant outputs. This completes the system identification. Second, we will choose a reference model which allows the desired behavior. Let us use a flight data from Eq. 23-24 for design and simulation.

$$A = \left[\begin{array}{ccccc|ccccc} -0.78501 & 0 & 0 & -9.8 & -9.8 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.065145 & 0 & 0 & -56.659 & 0 & 0 & -0.79784 & -0.0045036 & 1344.1 \\ 0.35712 & 0 & 0 & 0 & 92.468 & -0.063629 & 0 & 0 & 0 & 56.515 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & -11.842 & 0 & 0 & 0 & 0 & -7.1176 \\ \hline 0 & 0 & 0 & 0 & 0 & 0.11245 & 0 & 0 & 9.8 & 9.8 \\ 0.46624 & 0 & 0 & 0 & -0.6588 & -0.083441 & 0 & 0 & 0 & 131.19 \\ 0 & 1.0349 & 0 & 0 & 0 & 0 & -9.9435 & -0.30115 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.1755 & 0 & -1 & 0 & 0 & -14.687 \end{array}\right] \quad (23)$$

$$B = \left[\begin{array}{cc|cc} 0 & 0 & 0 & 0 \\ 0.71986 & 0 & 0 & 0 \\ 1.4468 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -11.198 & 0 & 4.3523 \\ \hline 0 & 0 & 204.28 & 0 \\ 0 & 0 & 0 & 0 \\ -3.5204 & 0 & -7.5159 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 2.9241 & 0 & 11.712 \end{array}\right] \quad (24)$$

Eigenvalues:

The key dynamics can be seen from the system's Eigen values and Eigen vectors, are listed in Table 4. The system is stable with damping because all the real parts of the eigenvalues are negative. The simulation (Fig. 12) shows clearly that the system is stable but having damping.

Sno	Eigenvalue	Damping	Freq. (rad/s)
1	-2.01e-002 + 8.27e-003i	9.25e-001	2.17e-002
2	-2.01e-002 - 8.27e-003i	9.25e-001	2.17e-002
3	-1.83e-001 + 9.01e-001i	1.99e-001	9.19e-001
4	-1.83e-001 - 9.01e-001i	1.99e-001	9.19e-001
5	-2.82e-001 + 5.79e-001i	4.37e-001	6.44e-001
6	-2.82e-001 - 5.79e-001i	4.37e-001	6.44e-001
7	-5.93e+000 + 6.22e+000i	6.90e-001	8.59e+000
8	-5.93e+000 - 6.22e+000i	6.90e-001	8.59e+000
9	-7.37e+000 + 1.06e+001i	5.73e-001	1.29e+001
10	-7.37e+000 - 1.06e+001i	5.73e-001	1.29e+001

Table 4. Eignevalues of the helicopter system

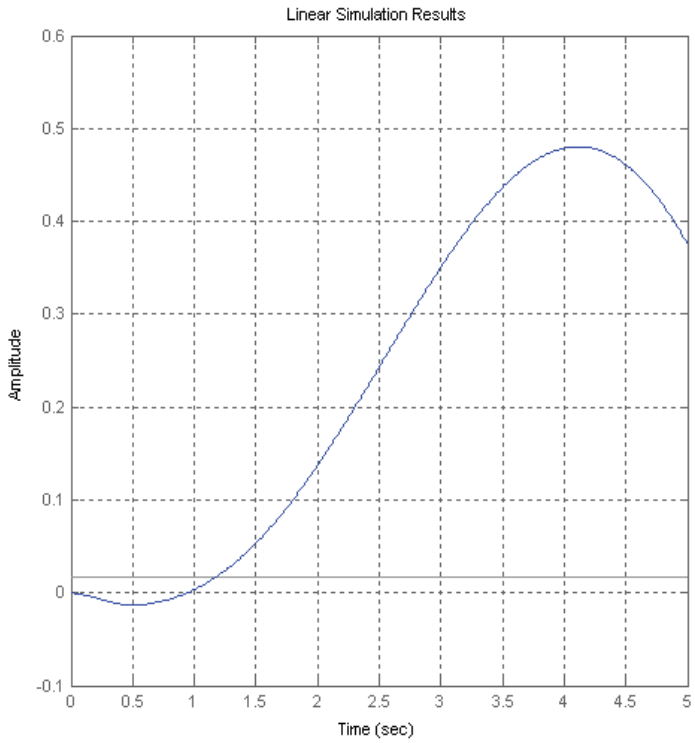


Fig. 12. Transfer function of system

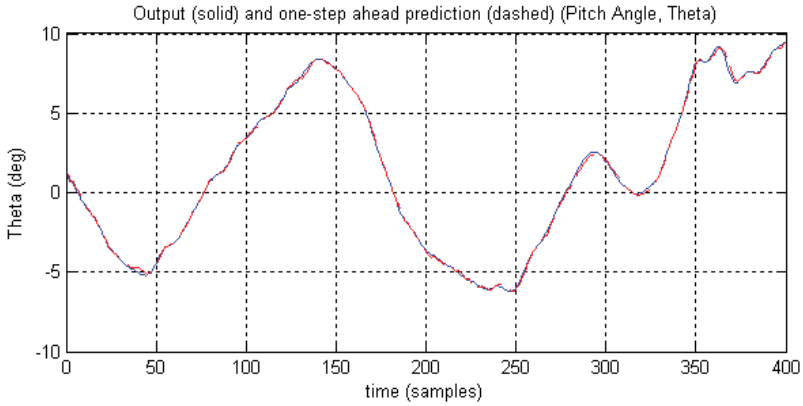
5. Experimental results and analysis

In this experiment we used NN approach to train MIMO model and capture the phenomena of flight dynamics. This simulation is divided into two parts longitudinal mode and Lateral mode. The NN approach considers separate lateral and longitudinal network with inertial coupling between the networks taken into consideration. These networks trained individually by making it MIMO model. Basically system identification process consists of gathering experimental data, estimate model from data and validate model with independent data. NN controller is designed in such a way that makes the plant output to follow the output of a reference model. The main target is to play with fine tuning of controller in order to minimize the state error.

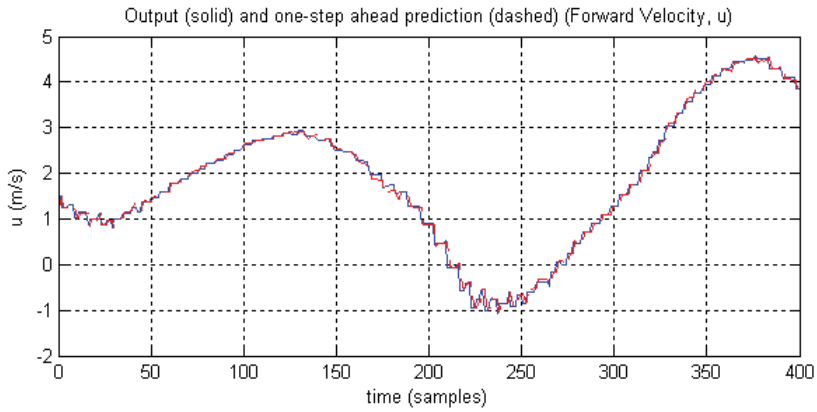
The experiment is carried out with System identification procedures with Prediction Error Method (PEM) algorithm using System Identification Toolbox using Levenberg-Marquardt (LM) algorithm. We observe NN approach to get better result of System identification that shows the perfect matching and shown as RUAV Longitudinal Dynamics and RUAV Lateral Dynamics in the following fig. 13-18

The prediction error of the output responses is described in Fig. 14. The autocorrelation function almost tend to zero and the cross correlation function vary in the range of -0.1to 0.1. This shows the dependency between prediction error and δ_{coll}, δ_{long} but the dependency rate is very less.

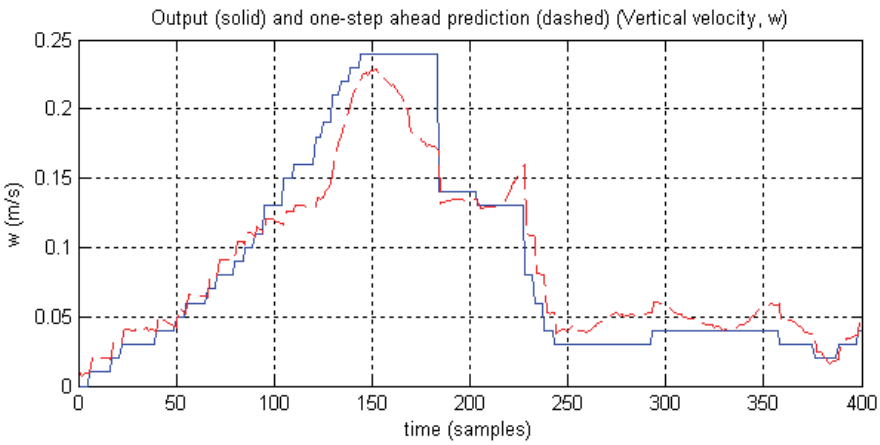
Longitudinal Dynamics Mode Analysis



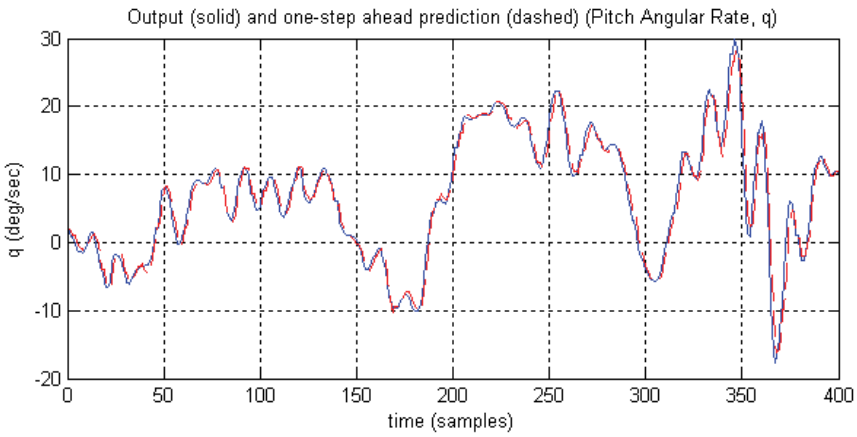
(a) Pitch Angle (θ)



(b) Forward Velocity (u)

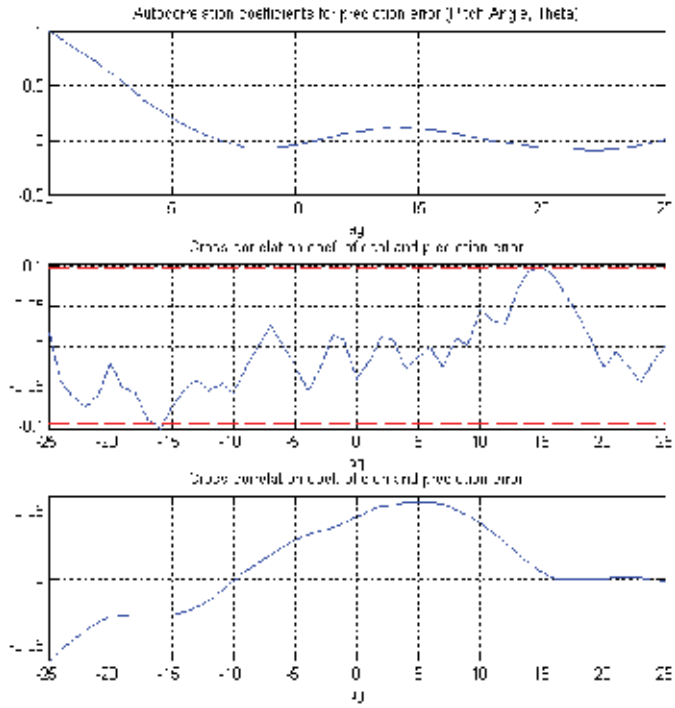


(c) Vertical velocity (w)

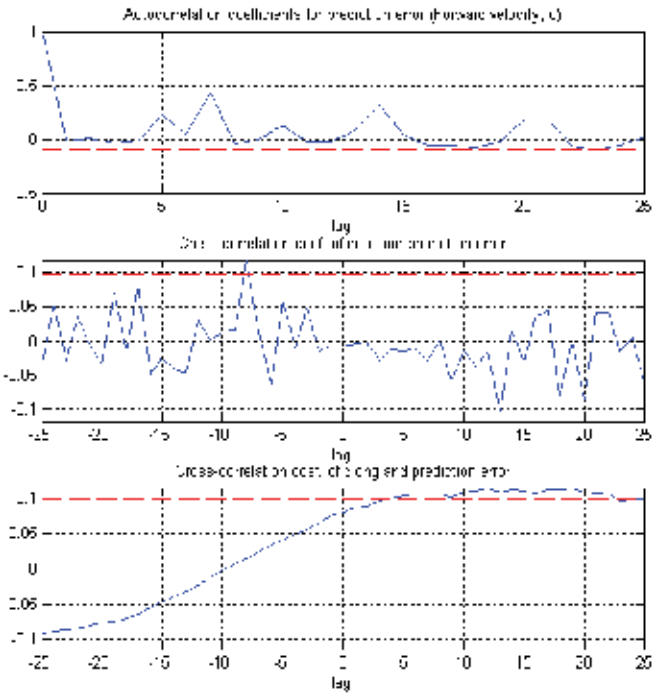


(d) Pitch Angular Rate (q)

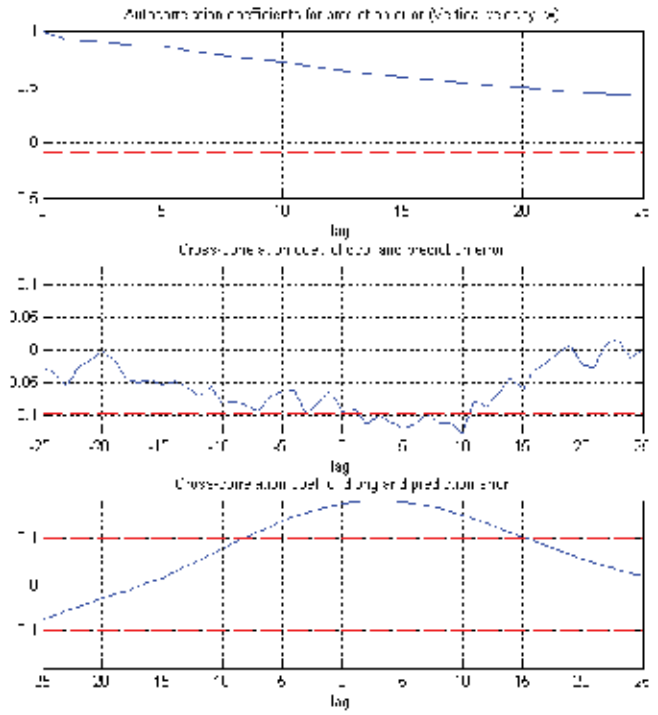
Fig. 13. Output response with network response in Longitudinal dynamics mode



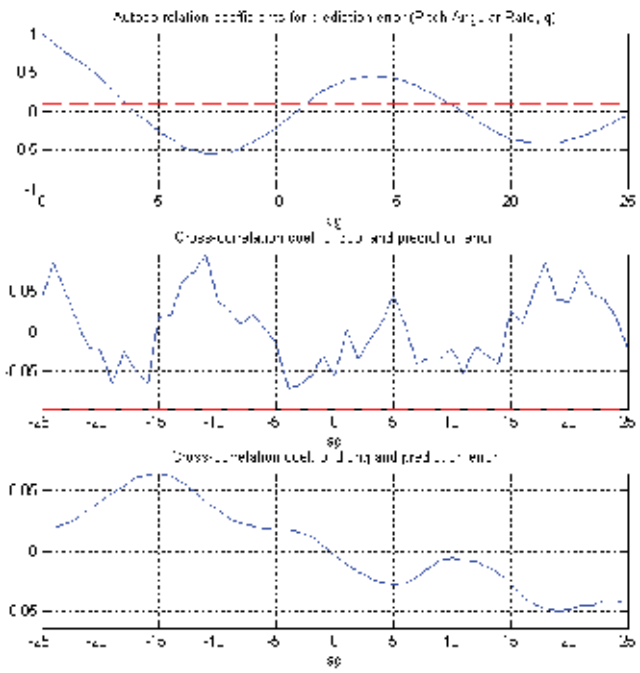
(a) Pitch Angle (θ)



(b) Forward velocity (u)



(c) Vertical velocity (w)



(d) Pitch Angular Rate (q)

Fig. 14. Autocorrelation and Cross-correlation of output response in longitudinal mode

The histogram of prediction error is shown in Fig. 15.

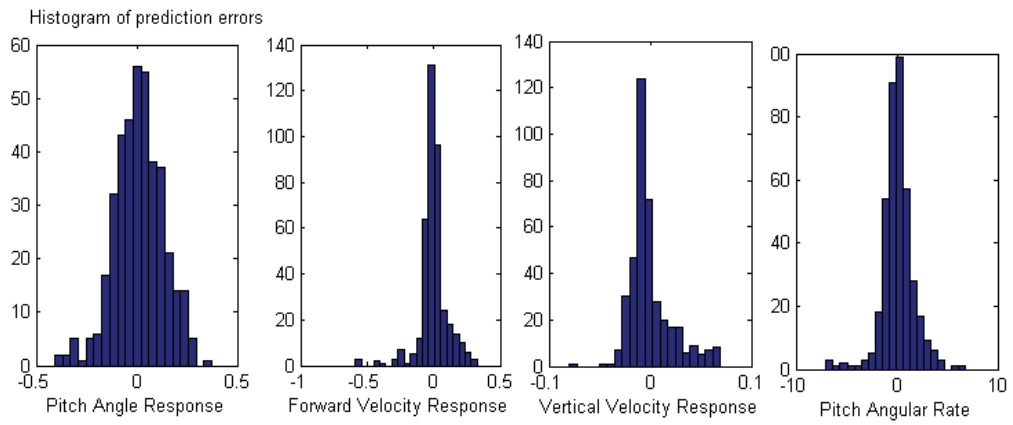
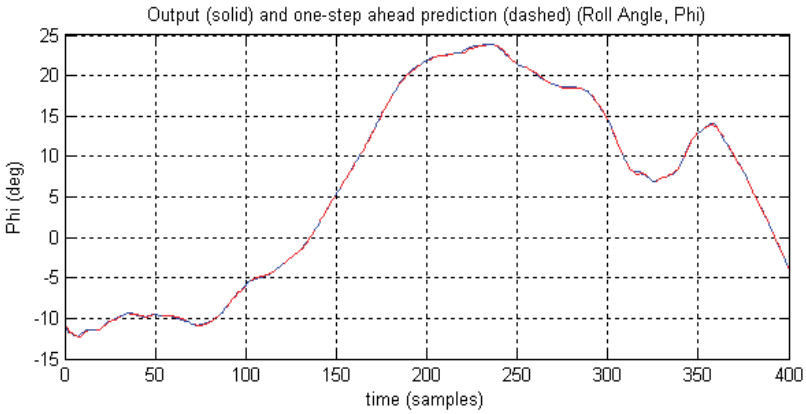
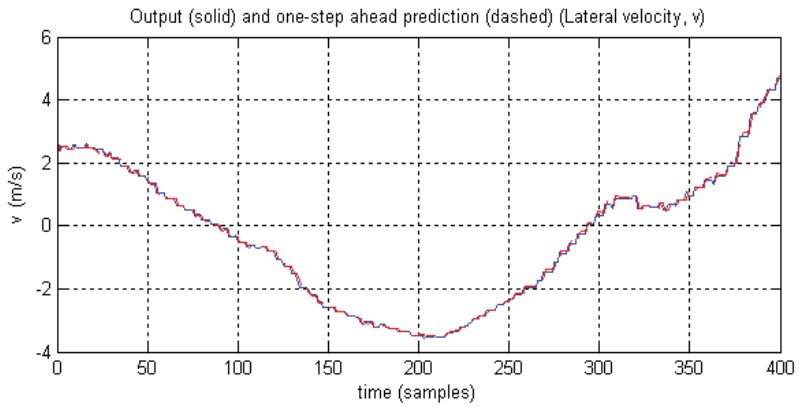


Fig. 15. Histogram of Prediction errors in Longitudinal Mode

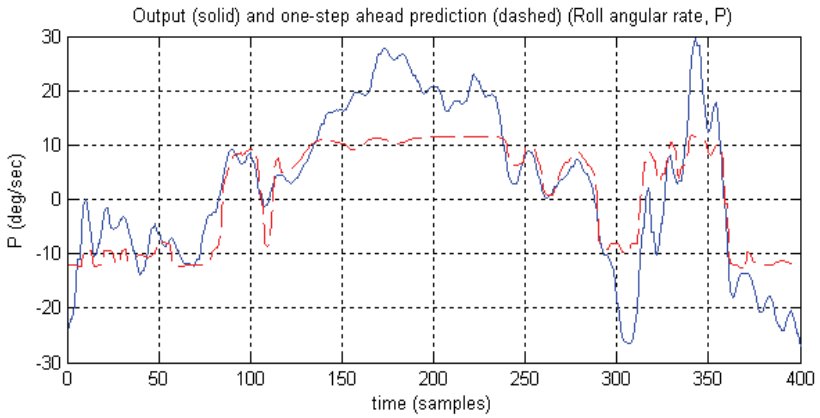
Lateral Dynamics Mode Analysis



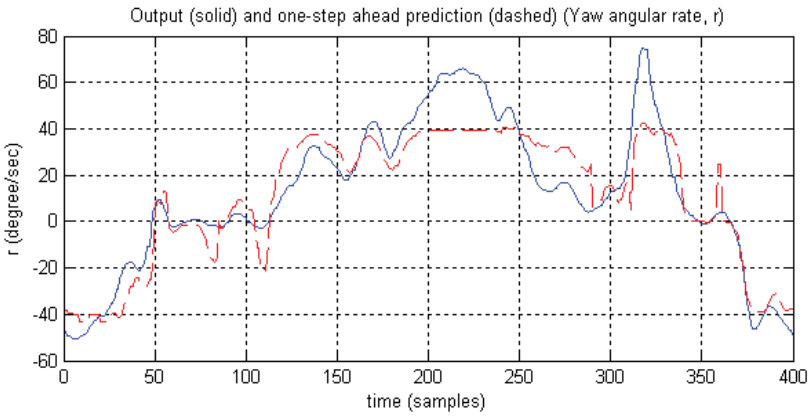
(a) Roll Angle (φ)



(b) Lateral Velocity (v)



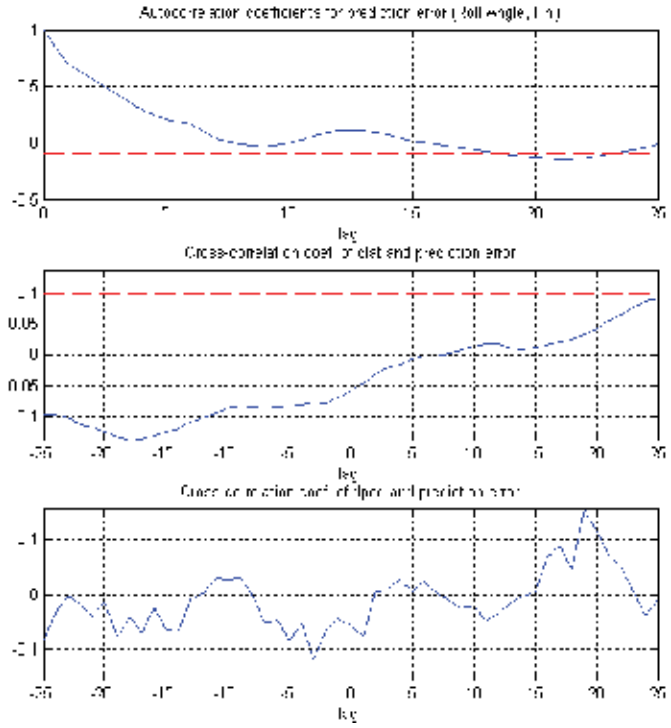
(c) Roll Angular Rate (P)



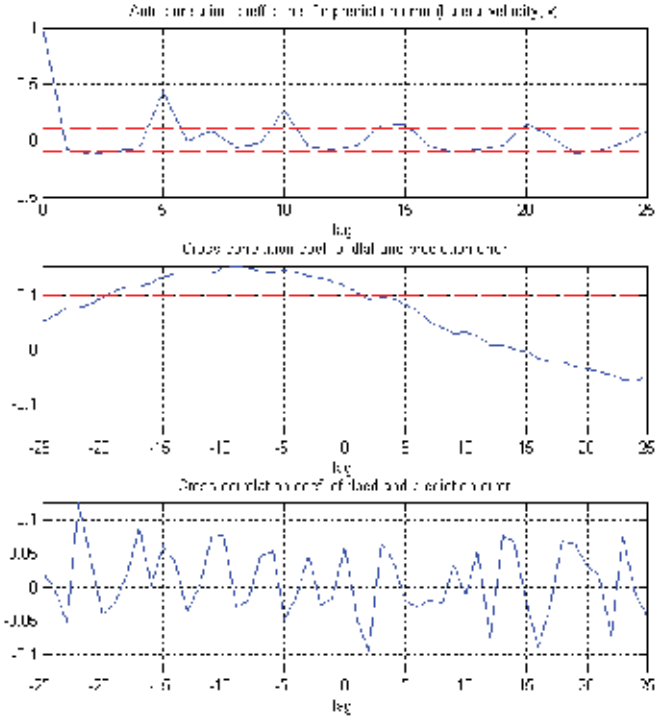
(d) Yaw Angular Rate (r)

Fig. 16. Output response with network response in lateral dynamics mode

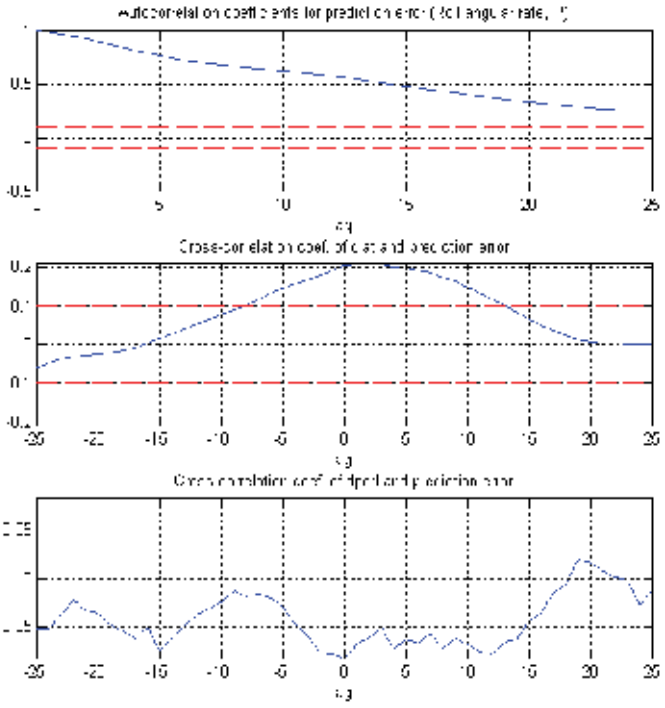
The prediction error of the output responses is described in Fig. 17. Similarly, in lateral mode also, the autocorrelation function almost tend to zero and the cross correlation function vary in the range of -0.1to 0.1. This shows the dependency between prediction error and δ_{lat} , δ_{ped} but the dependency rate is very less.



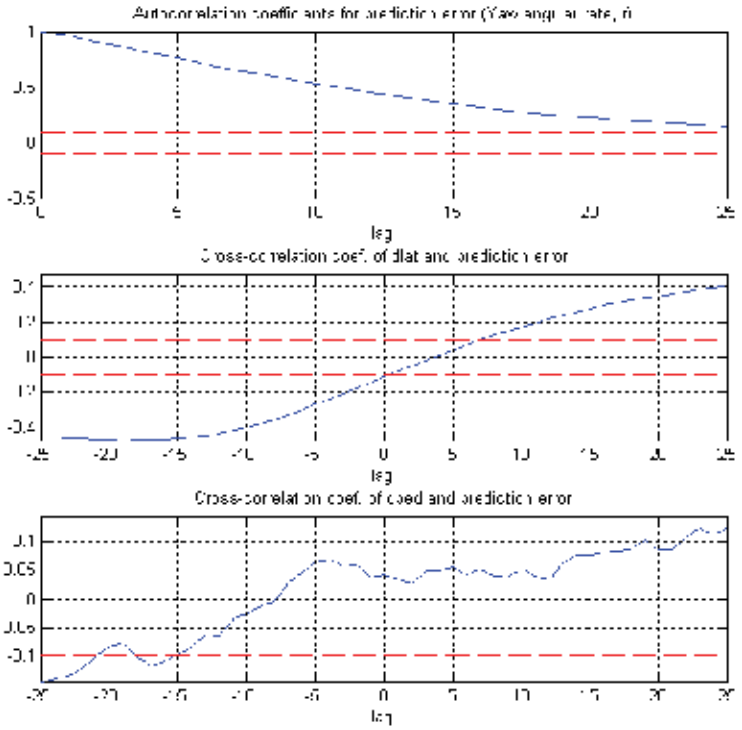
(a) Roll Angle (φ)



(b) Lateral Velocity (v)



(c) Roll Angular Rate (P)



(d) Yaw Angular Rate (r)

Fig. 17. Autocorrelation and Cross-correlation of output response in lateral mode

The histogram of prediction error is shown in Fig. 18.

6. Conclusion

UAV control system is a huge and complex system, and to design and test a UAV control system is time-cost and money-cost. This chapter considered the simulation of identification of a nonlinear system dynamics using artificial neural networks approach. This experiment develops a neural network model of the plant that we want to control. In the control design stage, experiment uses the neural network plant model to design (or train) the controller. We used Matlab to train the network and simulate the behavior.

This chapter provides the mathematical overview of MRC technique and neural network architecture to simulate nonlinear identification of UAV systems. MRC provides a direct and effective method to control a complex system without an equation-driven model. NN approach provides a good framework to implement MEC by identifying complicated models and training a controller for it.

7. Acknowledgment

"This research was supported by the MKE (Ministry of Knowledge and Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency)" (NIPA-2010-C1090-1031-00003)

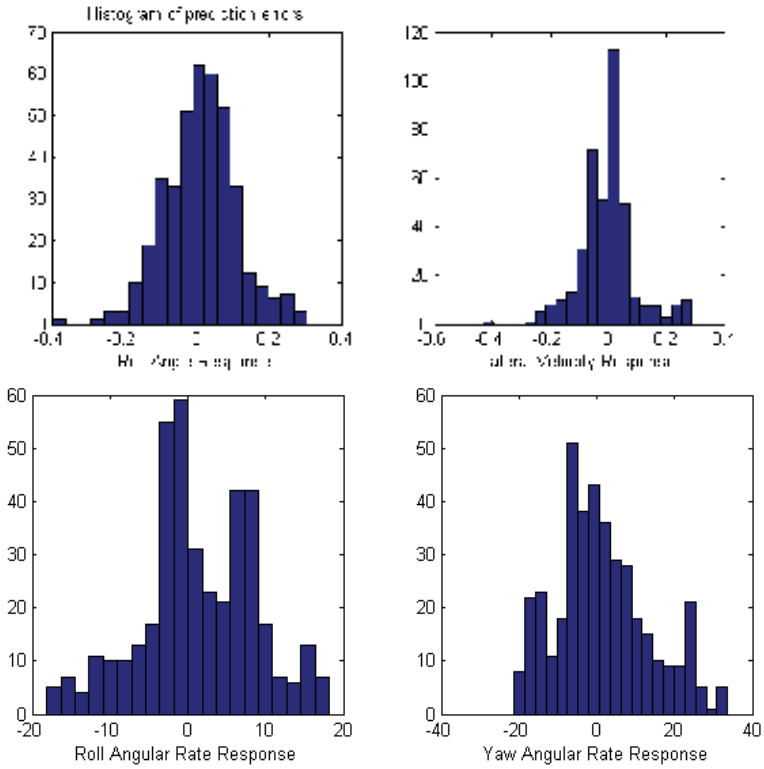


Fig. 18. Histogram of Prediction errors in Longitudinal Mode

8. References

[1] A. U. Levin, k. s Narendra," Control of Nonlinear Dynamical Systems Using Neural Networks: Controllability and Stabilization", *IEEE Transactions on Neural Networks*, 1993, Vol. 4, pp.192-206

[2] A. U. Levin, k. s Narendra," Control of Nonlinear Dynamical Systems Using Neural Networks- Part II: Observability, Identification and Control", *IEEE Transactions on Neural Networks*, 1996, Vol. 7, pp. 30-42

[3] David E. Rumelhart et al., "The basic ideas in neural networks", Communications of the ACM, v.37 n.3, p.87-92, March 1994

[4] E. R. Mueller, "Hardware-in-the-loop Simulation Design for Evaluation of Unmanned Aerial Vehicle Control Systems", *AIAA Modeling and Simulation Technologies Conference and Exhibit* , 20 - 23 August, 2007, Hilton Head, South Carolina

[5] E. N. Johnson and S. Fontaine, "Use of flight simulation to complement flight testing of low-cost UAVs", *AIAA Modeling and Simulation Technologies Conference and Exhibit*, Montreal, Canada, 2001

[6] MATLAB and Simulink for Technical Computing, Available from: http://www.mathworks.com

[7] Oliver Nelles, "Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models, Springer

[8] Cybenko, G., "Approximation by Superposition of a Sigmoidal Function, Mathematics of Control, Signals and Systems, 303-314.
[9] N. K. and K. Parthasarathy, Gradient methods for the optimization of dynamical systems containing neural networks. *IEEE Trans. on Neural Networks*, 252-262.
[10] B.G Martzios and F.L. Lewis, "An algorithm for the computation of the transfer function matrix of generalized two-dimensional systems " *Journal of Circuit, System, and Signal Processing*, Volume 7, Number 4 / December, 1988
[11] Budiyono A, Sudiyanto T, Lesmana H., "First Principle Approach to Modelling of Small Scale Helicopter", International Conference on Intelligent Unmanned System, 2007
[12] B. Mettler, T. Kanade, M.B. Tischler, "System Identification Modeling of a Model-Scale Helicopter", CMU-RI-TR-00-03. 2000.
[13] E. D. Beckmann, G. A. Borges, "Nonlinear Modeling, Identification and Control for a Simulated Miniature Helicopter," Robotic Symposium. LARS'08, pp.53-58, 2008.
[14] D. W. Marquardt. "An algorithm for least-squares estimation of nonlinear parameters". SIAM Journal on Applied Mathematics, Vol. 11 No.2 pp. 431–441, 1963.
[15] S. Haykin, "Neural networks: A comprehensive foundation", IEEE Press, New York, USA, 1994
[16] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," IEEE Transactions on Neural Networks, vol. 1, no. 1, pp. 4–27, 1990.
[17] La Civita, M., G., P., Messner, W. C., and Kanade, T., "Design and Flight Testing of a High-Bandwidth H∞ Loop Shaping Controller for a Robotic Helicopter," Proceedings of the AIAA Guidance, Navigation, and Control Conference, No. AIAA 2002-4836, 2002.
[18] Sahasrabudhe, V., & Celi, R., "Improvement of off-design characteristics in integrated rotor-flight control system optimization". AHS, annual forum 53rd Virginia Beach, VA, April 29–May 1, 1997, Proceedings. Alexandria, VA, American Helicopter Society, 1997. Vol. 1 (A97-29180 07-01).
[19] Smerlas, A., Postlethwaite, I., Walker, D. J., Strange, M. E., Howitt, J., Horton, R. I., Gubbels, A. W., & Baillie, S. W. , "Design and flight testing of an H-infinity controller for the NRC Bell 205 experimental fly-by-wire helicopter. AIAA GNC conference, 1998.
[20] Li-Xin Wang, "Design and analysis of fuzzy identifiers of nonlinear dynamic systems". IEEE Transactions on Automatic Control, 40(1), 1995.
[21] Shaaban A. Salman, Vishwas R. Puttige, and Sreenatha G. Anavatti, "Real-time Validation and Comparison of Fuzzy Identification and State-space Identification for a UAV Platform" Proceeding of the 2006 IEEE International Conference on Control Applications, pages 2138–2143, 2006.
[22] R. Pintelon and J. Schoukens, "System Identification: A Frequency Domain Approach" Wiley-IEEE Press, 1st edition, 2001.
[23] Kumpati S. Narendra and Kannan Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks" IEEE transaction on Neural Networks, 1(1), 1990.
[24] Magnus Norgaard, "Neural Network Based System Identification Tool Box", Version 2, 2000

[25] Budiyono, A. and Sutarto, H.Y., Linear Parameter Varying Model Identification for Control of Rotorcraft-based UAV, Fifth Indonesia-Taiwan Workshop on Aeronautical Science, Technology and Industry, Tainan, Taiwan, November 13-16, 2006

[26] M. M. Korjani, O.Bazzaz, M. B. Menhaj, "Real time identification and control of dynamics systems using recurrent neural network", Journal of Artificial Intelligence Review, Springer, August 2009

[27] W. Yu, X. Li, "Recurrent fuzzy neural networks for nonlinear system identification", 22nd IEEE International Symposium on Intelligent Control Part of IEEE Multi-conference on Systems and Control, Singapore, 1-3 October 2007.

[28] Shim D. H., Kin H. J., Sastry. "Control System Design for Rotorcraft-based Unmanned Aerial Vehicles using Time-domain System Identification". IEEE International Conference on Control Application, 2000. pp. 808-813

20

Intelligent Vibration Signal Diagnostic System Using Artificial Neural Network

Chang-Ching Lin
Tamkang University Tamshui, Taipei County, Taiwan

1. Introduction

In today's sophisticated manufacturing industry maintenance personnel are constantly forced to make important, and often costly, decisions on the use of machinery. Usually, these decisions are based on practical considerations, previous experiences, historical data and common sense. However, the exact determination of machine conditions and accurate prognosis of incipient failures or machine degradation are key elements in maximizing machine availability.

The practice of maintenance includes machine condition monitoring, fault diagnostics, reliability analysis, and maintenance planning. Traditionally, equipment reliability studies depend heavily on statistical analysis of data from experimental life-tests or historical failure data. Tedious data collection procedures usually make this off-line approach unrealistic and inefficient for a fast-changing manufacturing environment (Singh & Kazzaz, 2003). Over the past few decades technologies in machine condition monitoring and fault diagnostics have matured. Many state-of-the-art machine condition monitoring and diagnostic technologies allow monitoring and fault detection to perform in on-line, real-time fashion making maintenance tasks more efficient and effective. Needless to say, new technologies often produce new kinds of information that may not have been directly associated with the traditional maintenance methodologies. Therefore, how to integrate this new information into maintenance planning to take advantages of the new technologies has become a big challenge for the research community.

From the viewpoint of maintenance planning, Condition Based Maintenance (CBM) is an approach that uses the most cost effective methodology for the performance of machinery maintenance. The idea is to ensure maximum operational life and minimum downtime of machinery within predefined cost, safety and availability constraints. When machinery life extension is a major consideration the CBM approach usually involves predictive maintenance. In the term of predictive maintenance, a two-level approach should be addressed: 1) need to develop a condition monitoring for machine fault detection and 2) need to develop a diagnostic system for possible machine maintenance suggestion.

The subject of CBM is charged with developing new technologies to diagnose the machinery problems. Different methods of fault identification have been developed and used effectively to detect the machine faults at an early stage using different machine quantities, such as current, voltage, speed, efficiency, temperature and vibrations. One of the principal tools for diagnosing rotating machinery problems is the vibration analysis. Through the use

of different signal processing techniques, it is possible to obtain vital diagnostic information from vibration profile before the equipment catastrophically fails. A problem with diagnostic techniques is that they require constant human interpretation of the results. The logical progression of the condition monitoring technologies is the automation of the diagnostic process. The research has been underway for a long time to automate the diagnostic process. Recently, artificial intelligent tools, such as expert systems, neural network and fuzzy logic, have been widely used with the monitoring system to support the detection and diagnostic tasks.

In this chapter, artificial neural network (ANN) technologies and analytical models have been investigated and incorporated to present an Intelligent Diagnostic System (IDS), which could increase the effectiveness and efficiency of traditional condition monitoring diagnostic systems.

Several advanced vibration trending methods have been studied and used to quantify machine operating conditions. The different aspects of vibration signal and its processing techniques, including autoregressive (AR) parametric modeling and different vibration trending methods are illustrated. An example of integrated IDS based on real-time, multi-channel and neural network technologies is introduced. It involves intermittent or continuous collection of vibration data related to the operating condition of critical machine components, predicting its fault from a vibration symptom, and identifying the cause of the fault. The IDS contains two major parts: the condition monitoring system (CMS) and the diagnostic system (DS). A neural network architecture based on Adaptive Resonance Theory (ART) is introduced. The fault diagnostic system is incorporated with ARTMAP neural network, which is an enhanced model of the ART neural network. In this chapter, its performance testing on simulated vibration signals is presented. An in-depth testing using lab bearing fault signals has been implemented to validate the performance of the IDS. The objective is to provide a new and practicable solution for CBM.

Essentially, this chapter presents an innovative method to synthesize low level information, such as vibration signals, with high level information, like signal patterns, to form a rigorous theoretical base for condition-based predictive maintenance.

2. Condition monitoring system

The condition monitoring system developed contains four modules (see Fig. 1): data acquisition, Parameters Estimation (PE), Performance Monitoring (PM), and Information Display and Control (IDC). The entire system was coded using C programming language. We have developed a user friendly graphic interface that allows for easy access and control in monitoring an operating machine. The system has been tested and verified on an experimental lab setting. The detailed procedure of ISDS and programming logic is discussed in the following sections.

2.1 Data acquisition module

The data acquisition module is more hardware related than the other modules. Vibration signals were acquired through accelerometers connected to a DASMUX-64 multiplexer board and a HSDAS-16 data acquisition board installed in a PC compatible computer. The multi-channel data acquisition program controlling the hardware equipment has been coded.

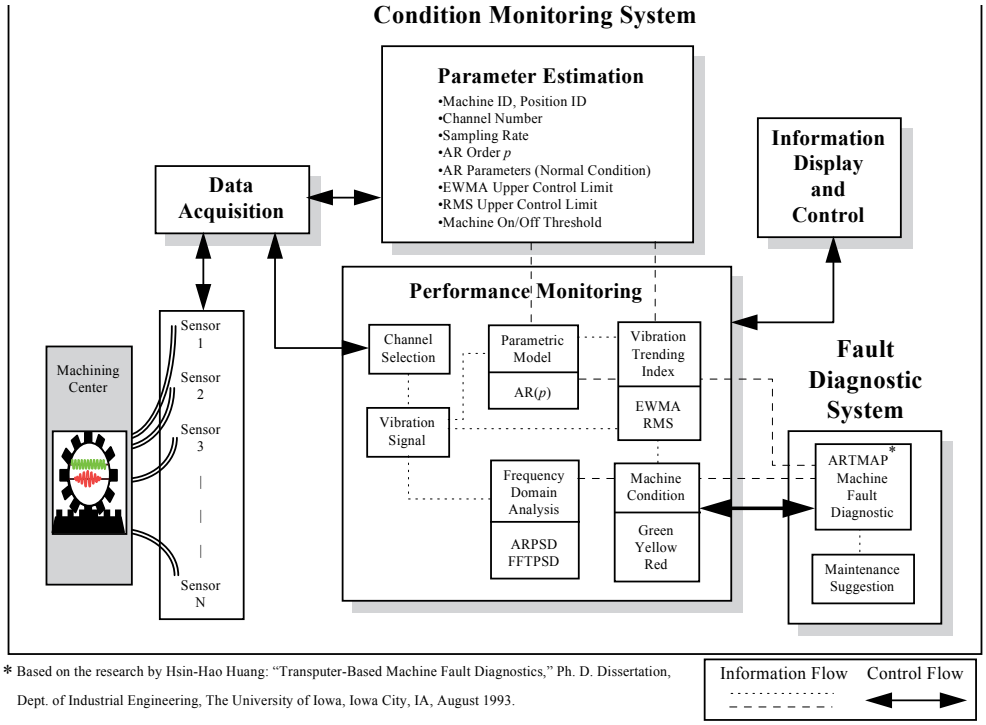


Fig. 1. Overview of intelligent diagnostic system

2.2 Programming logic for Parameter Estimation (PE) module

The parameter estimation module is designed to estimate the parameters of the normal condition of a machine. It provides a procedure to set up the machine positions considered to be critical locations of the machine. The PE module must be executed before running the PM module. The information to be calculated in the PM module needs to be compared to the base-line information generated in the PE module.

The normal operating condition of a machine position is usually defined by experience or from empirical data. Generally speaking, a particular operation mode of a machine is selected and then defined as a "normal condition". However, this normal condition is not unchangeable. Any adjustment to the machine, such as overhaul or other minor repairs, would change its internal mechanisms. In this case, the normal condition must be redefined, and all the base-line data of the monitored positions on the machine need to be reset.

The PE procedure starts with specifying the ID of a machine, its location ID, and several other parameters related to each position, such as channel number and sampling rate. Then the upper control limits of the Exponentially Weighted Moving Average (EWMA) (Spoerre, 1993) and Root Mean Square (RMS) (Monk, 1972; Wheeler 1968) vibration trending indices are determined and an adequate Autoregressive (AR) order is computed. The AR time series modelling method is the most popular parametric spectral estimation method which translates a time signal into both frequency domain and parameter domain (Gersch, 1976). Once the AR order is determined, the AR parameters can be estimated through several normal condition signals collected from the particular position. A major issue with the parametric method is determining the AR order for a given signal. It is usually a trade-off between resolution and unnecessary peaks. Many criteria have been proposed as objective functions for selecting a

"good" AR model order. Akaike has developed two criteria, the Final Prediction Error (FPE) (Akaike, 1969) and Akaike Information Criterion (AIC) (Akaike, 1974). The criteria presented here may be simply used as guidelines for initial order selection, which are known to work well for true AR signals; but may not work well with real data, depending on how well such data set is modelled by an AR model. Therefore, both FPE and AIC have been adapted in this research for the AR order suggestion.

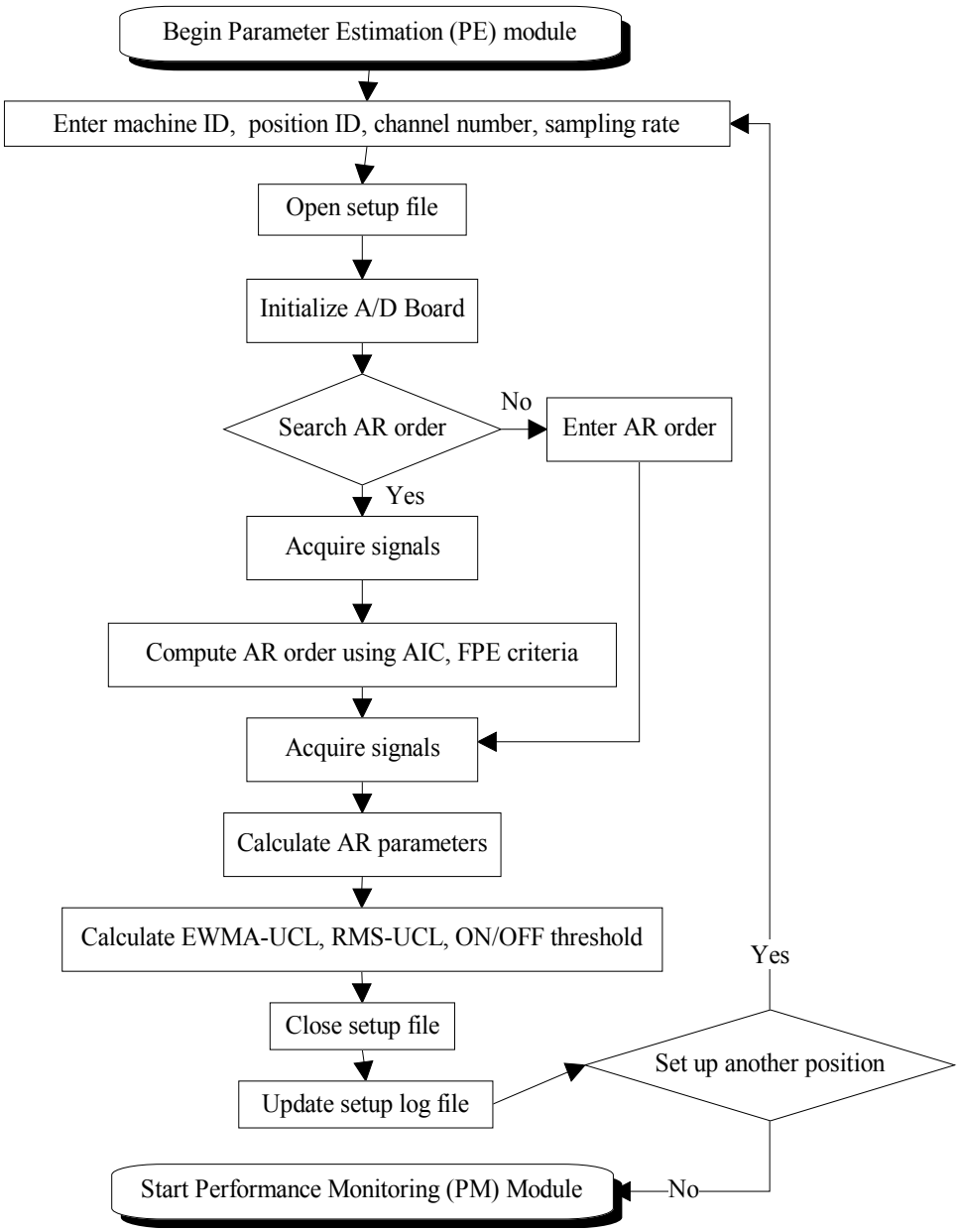


Fig. 2. Flowchart of parameter estimation (PE) module

A setup file is then generated after the PE procedure is completed. This file, given a name that combines the machine ID and the position ID, consists of all the parameters associated with the specific position. The number of setup files created depends on the number of positions to be monitored in the PM mode, that is, each monitored position is accompanied by a setup file.
In order to perform a multi-channel monitoring scheme a setup log file is also generated. This file contains all the names of setup files created in the PE mode. Every time a new position is added its setup file name is appended to the setup log file. The setup log file is very important. It not only determines the channels needing to be scanned when the PM program is executed, it also provides the PM program with paths to locate all the necessary information contained in the setup files. Fig. 2 shows the programming logic of the PE module. In practice, after the PE procedure is completed, on-line performance monitoring of the machine (the PM mode) begins.

2.3 Programming logic for Performance Monitoring (PM) module

In the PM module, vibration data arrive through the data acquisition hardware and are processed by AR, EWMA, ARPSD, RMS, FFT spectrum, and hourly usage calculation subroutines. After each calculation the current result is displayed on the computer screen through the Information Display and Control (IDC) module. Fig. 3 illustrates the flow chart of the PM programming logic.
IDC is in charge of functions such as current information displaying, monitoring control, and machine status reasoning. Details of these functions are given in the following section.

2.4 Information Display and Control (IDC) module

Eight separate, small windows appear on the computer screen when the IDC module is activated. Each window is designed to show the current reading and information related to each calculation subroutine (e.g. AR, EWMA, ARPSD, RMS, and FFT spectrum) for the current position being monitored.
Window 1 is designed to plot the current time domain data collected from the data acquisition equipment. Window 2 displays both the AR parameter pattern of the current signal and the normal condition AR parameter pattern stored in the setup file generated in the PE module. Window 3 plots the current EWMA reading on a EWMA control chart and its upper control limit. Window 4 plots the current RMS value and its upper control limit on a RMS control chart. Both the RMS and EWMA upper control limits are calculated in the PE module. Window 5 displays the hourly usage and other information of the position. The hourly usage of the position is calculated based on the vibration level of that position. It is an estimated running time of the component up to the calculating point from the time this position is set up. Window 6 indicates the current performance status of the position. Three different levels of performance status: normal, abnormal, and stop, are designed. Each status is represented by a different colour: a green light signals a normal condition; a yellow light represents an abnormal condition; and a red light indicates an emergency stop situation. The determination of the status of a position based on the current readings is discussed in the next section. Window 7 gives the current ARPSD spectrum, which is calculated based on the AR parameters from Window 2. Finally, Window 8 displays the current FFT spectrum by using the time domain data from Window 1.
In addition to real-time information display, the IDC module also provides a user-friendly graphic interface for monitoring control. A user can utilize the mouse to navigate around

the computer screen and click on an icon to perform the specified function. For instance, to switch to another channel one can click on the "CH+" or "CH-" icon. Fig. 4 shows the IDC screen layout developed.

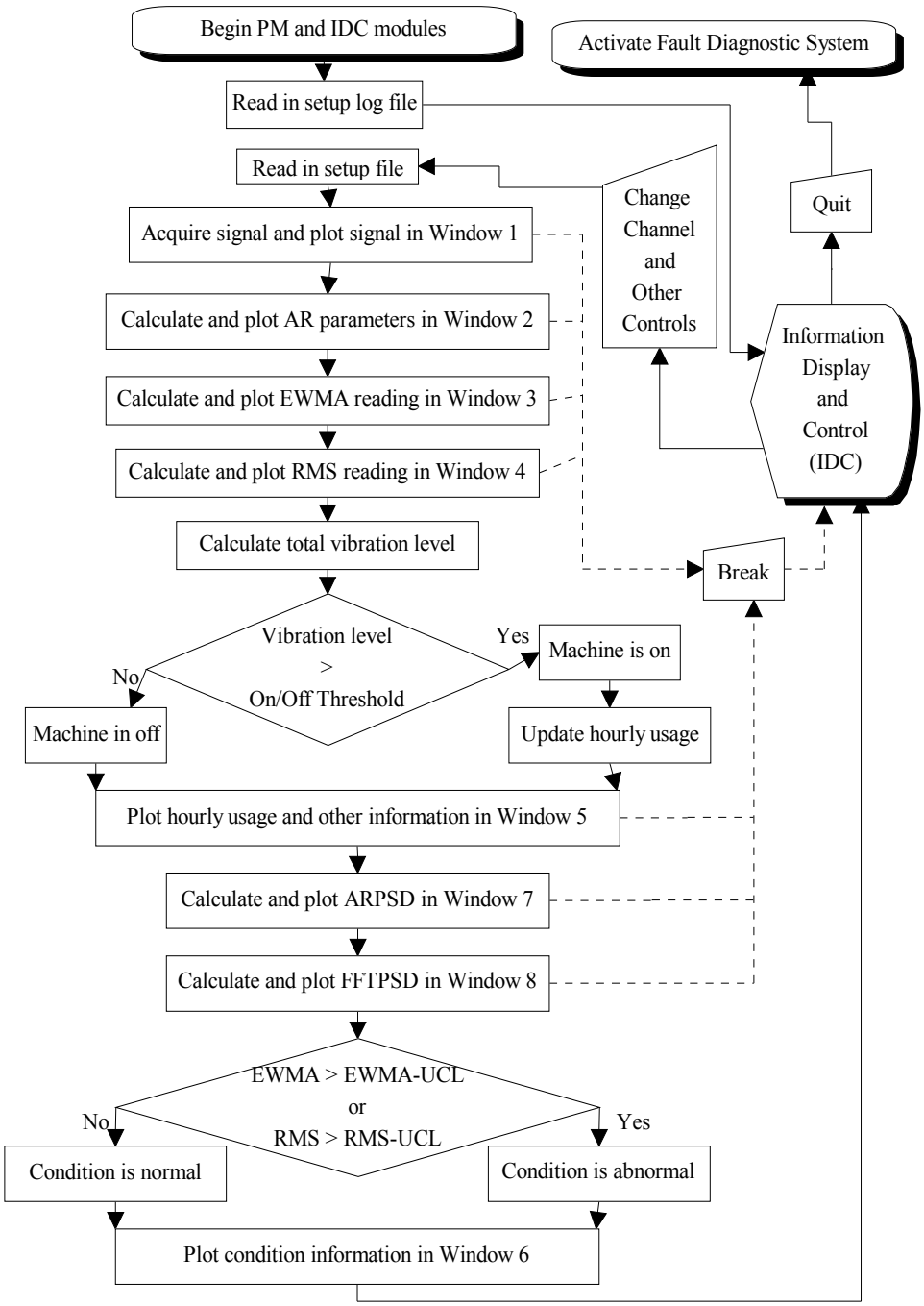


Fig. 3. Flowchart of PM and IDC modules

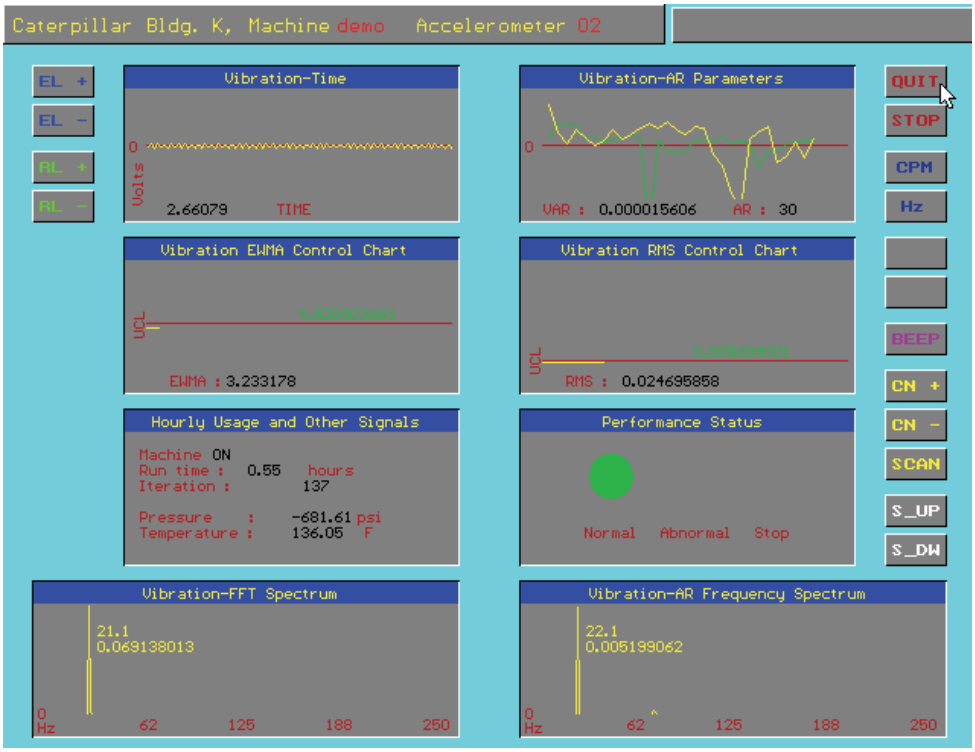


Fig. 4. Condition monitoring information display and control (IDC) Screen layout

2.5 Vibration condition status reasoning

Based on the criteria stored in the setup file and the current readings, the EWMA and RMS control charts show whether the current readings are under or above their respective upper control limit. If both readings are under their corresponding control limits, then the position is in a normal condition. However, if either one of the control readings exceeds its upper control limit, the performance status reasoning program would turn on the yellow light to indicate the abnormality of the position. In this case, the fault diagnostic system is activated.

2.6 Condition monitoring sample session

Data collection, in the form of vibration signals, was conducted using the following test rig (see Fig. 5): a 1/2 hp DC motor connected to a shaft by a drive belt, two sleeve bearings mounted on each end of the shaft and secured to a steel plate, an amplifier to magnify signals, a DASMUX-64 multiplexer board, and a HSDAS-16 data acquisition board installed in a personal computer. Vibration signals were collected from the bearing using 328C04 PCB accelerometers mounted on the bearing housings. Using the test rig, the following sample session was conducted.

Assume that when the motor was turned on initially, it was running in normal condition. Later, a small piece of clay was attached to the rotational element of the test rig to generate an imbalance condition. This was used as an abnormal condition in the experiment. In the beginning, the setup procedure (PE) needed to be performed in order to obtain the base-li information. The sampling rate used was 1000 Hz and the sampling time was one second.

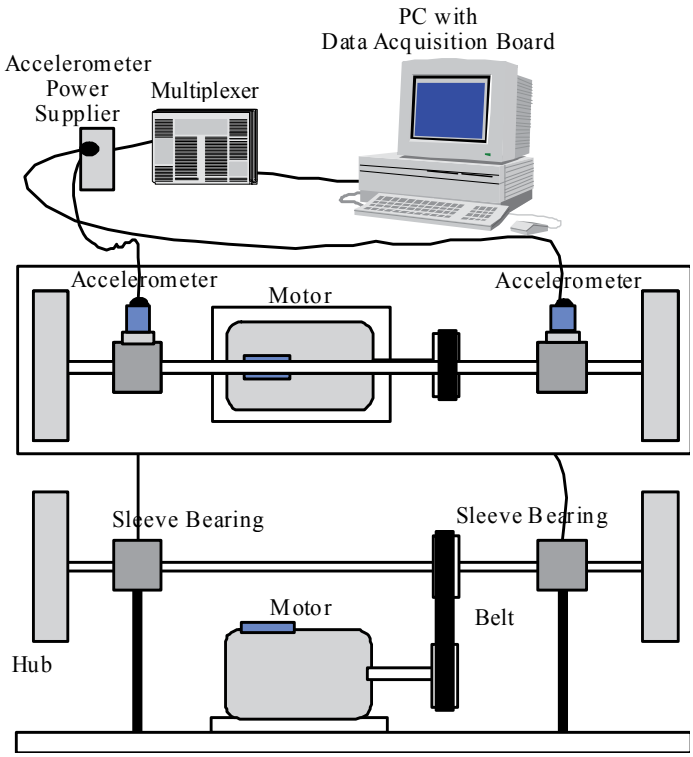


Fig. 5. The test rig for ISDS experiment

The PE program first acquired eight samples and then took their average. Using the average normal signal, the AIC and FPE criteria were calculated. An AR order suggestion for the normal condition of the test rig was made. The AR order was fixed throughout the entire experiment. Once the AR order was known, the program started estimating the AR parameters and upper control limits of RMS and EWMA by collecting another eight data sets, calculating eight sets of AR parameters, and then averaging them. Finally, all parameters were stored in the setup file which would be used in the PM stage. An example of the normal condition parameters from a setup file are listed below:

- Machine ID: TESTRG
- Position ID: CHN1
- Channel number: 1
- Sampling rate: 1000
- AR order: 32
- AR parameters: ....
- EWMAUCL: 0.8912
- RMSUCL: 0.0367

When the machine was running in normal condition the readings of EWMA were approximately -0.486 far below the EWMAUCL of 0.8912. The readings of RMS were about 0.01895, and therefore, they were below the RMSUCL. As soon as an imbalance condition was generated the EWMA and RMS readings jumped to values of 3.3259 and 0.0504, respectively. The EWMA and RMS readings indicated the test rig was in an abnormal condition since both readings exceeded their respective control limits.

The machine condition monitoring mode switches to diagnostic mode when at least one index exceeds its control limit. Once the system is in the diagnostic system, a detailed automatic analysis begins to identify the machine abnormality occurred. The next section explains the fault diagnostic system designed for this research.

3. ARTMAP-based diagnostic system

3.1 Introduction to ARTMAP neural network

The diagnostic system in this paper employs a neural network architecture, called Adaptive Resonance Theory with Map Field (ARTMAP). The fault diagnostic system is based on the ARTMAP fault diagnostic network developed by Knapp and Wang (Knapp & Wang, 1992). The ARTMAP network is an enhanced model of the ART2 neural network (Carpenter, 1987; Carpenter, 1991). The ARTMAP learning system is built from a pair of ART modules (see Fig. 6), which is capable of self-organizing stable recognition categories in response to arbitrary sequences of input patterns. These ART modules (ART_a and ART_b) are linked by Map Field and an internal controller that controls the learning of an associative map from the ART_a recognition categories to the ART_b recognition categories, as well as the matching of the ART_a vigilance parameter (ρ'). This vigilance test differs from the vigilance test inside the ART2 network. It determines the closeness between the recognition categories of ART_a and ART_b (Knapp, 1992).

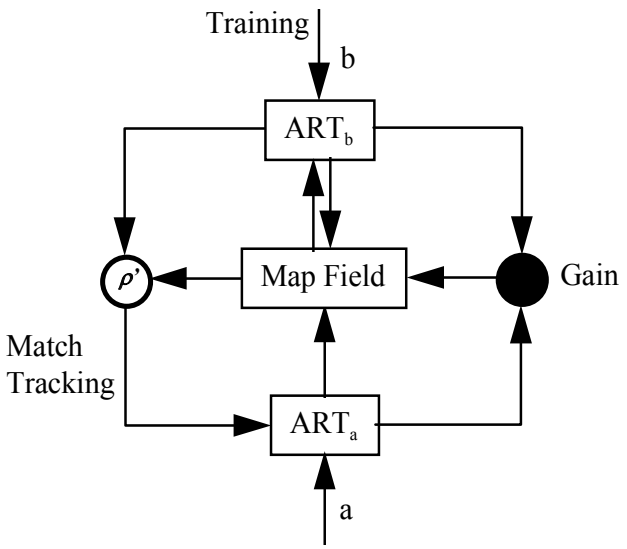


Fig. 6. ARTMAP architecture

A modified ARTMAP architecture has been adopted in this paper in order to perform the supervised learning. The modified ARTMAP architecture is based on the research by Knapp and Huang, which replaces the second ART module (ART_b) by a target output pattern provided by the user (Huang, 1993; Knapp, 1992). The major difference between the modified ARTMAP network and the ART2 network is the modified ARTMAP permits supervised learning while ART2 is an unsupervised neural network classifier. Fig. 7 shows the modified ARTMAP architecture.

3.2 Performance analysis of ARTMAP-based diagnostic system

The performance of the ARTMAP-based diagnostic system was validated by employing vibration signals from test bearings. A small adjustment was made on the experimental test rig shown in Figure 5. The two sleeve bearings were replaced by two ball bearings with steel housings. The new setup allows easy detachment of the ball bearing from the housing for exchanging different bearings. Figure 8 shows the modified experimental setup.

Six bearings with different defect conditions were made. Table 1 describes these defective ball bearings. A two-stage vibration data collection was conducted for each bearing. Five sets of vibration signals were collected in the first batch, three sets in the second batch. A total of eight sets of vibration signals were collected under each defect. Therefore, there were a total of 48 data sets. All time domain vibration signals were transformed and parameterized through the ARPSD algorithm. The AR order used was 30. Thus, the dimension number for each AR parameter pattern was 31 (i.e., 30 AR parameters plus one variance). These 48 AR parameter patterns were used to train and test the ARTMAP-based diagnostic system.

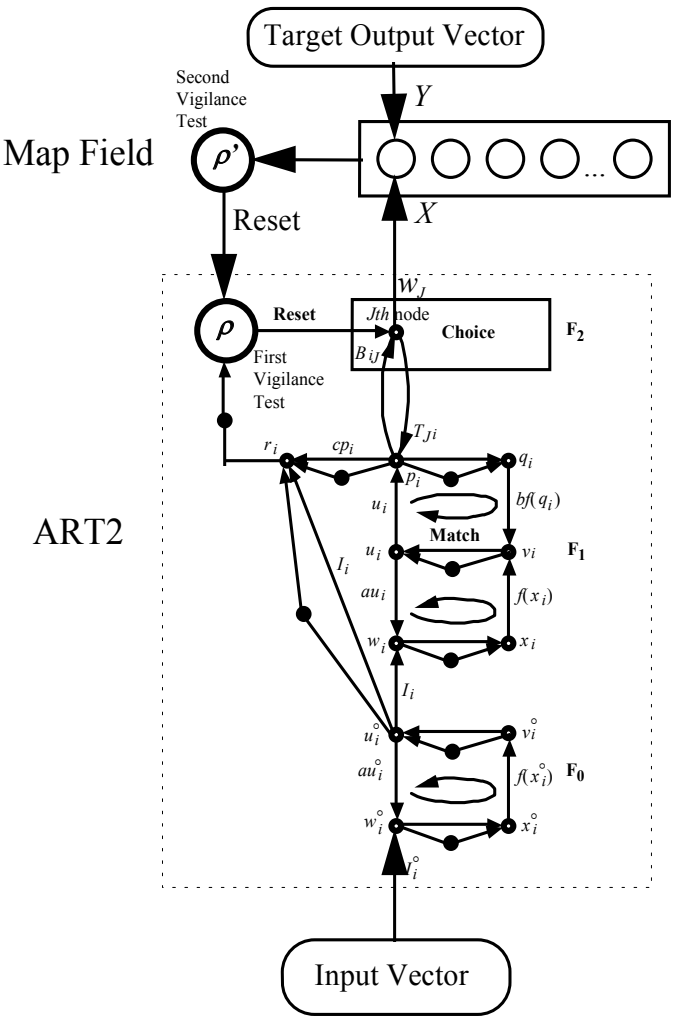


Fig. 7. Modified ARTMAP architecture

Bearing #	Defect
1	Good bearing
2	White sand in bearing
3	Over-greased in raceway
4	One scratch in inner race
5	One scratch in one ball
6	No grease in raceway

Table 1. Test ball bearings

Pattern Number		Bearing Number											
		1		2		3		4		5		6	
Batch 1	1	Train		Train		Train		Train		Train		Train	
	2	1	3	2	6	3	1	4	2	5	6	6	2
	3	1	6	2	6	3	1	4	2	5	4	6	1
	4	1	6	2	6	3	1	4	2	5	4	6	2
	5	1	6	2	6	3	1	4	2	5	6	6	1
Batch 2	1	1	3	2	6	3	1	**5**	**4**	5	4	6	5
	2	1	3	2	6	3	1	**5**	**4**	5	4	6	5
	3	1	3	2	6	3	1	**5**	**4**	5	4	6	5

Table 2. Bearing test results of ARTMAP-based ISDS

Note that the 512 frequency components in each ARPSD spectrum were compressed to only 31 parameters in each AR model indicating the system dealt with a significantly reduced amount of data; this is extremely beneficial in real-time applications.

Fig. 8 shows the plots of AR parameter patterns from the six defective bearings. The first column displays the six training patterns, which is the first one of the eight data sets from each bearing type. The second column illustrates some of the other seven test patterns, where the solid lines represent data from the first collection batch and the dotted lines are from the second batch. As can be seen from Fig. 8, the profiles of the AR parameter patterns within each group are very similar. Only a few deviations can be seen between the first and second batches. The deviations come from the very sensitive but inevitable internal structure changes of the setup during the bearing attachment and detachment operations between the two data collections.

The experimental procedure began with using the first pattern of all the conditions for training and then randomly testing the other seven patterns. In addition, the modified ARTMAP network was designed to provide two suggested fault patterns (i.e., the outputs of the first two activated nodes from the F_2 field). Table 2 summarizes the test results on diagnosing the 42 test patterns. The first column of Table 2 for each bearing type is the first identified fault from the network. It shows only 3 of the 42 test cases were mismatched in the first guess but they were then picked up correctly by the network in the second guess (see bold-face numbers in Table 2). Interestingly, these three mismatched patterns were from

the second batch. If the profiles of Bearings 4 and 5 in the second batch (the dotted profiles in the second column of Fig. 8) were compared, then one could see the test patterns of Bearing 4 from the second batch were much closer to the training pattern of Bearing 5 than that of Bearing 4. This is why the network recognized the test patterns of Bearing 4 as Bearing 5 in its first guess. These test results clearly display the capability and reliability of the ARTMAP-based diagnostic system and the robustness of using AR parameter patterns to represent vibration signals. For the efficiency of the ARTMAP training, the training time of one 31-point AR parameter pattern was less than one second on a PC.

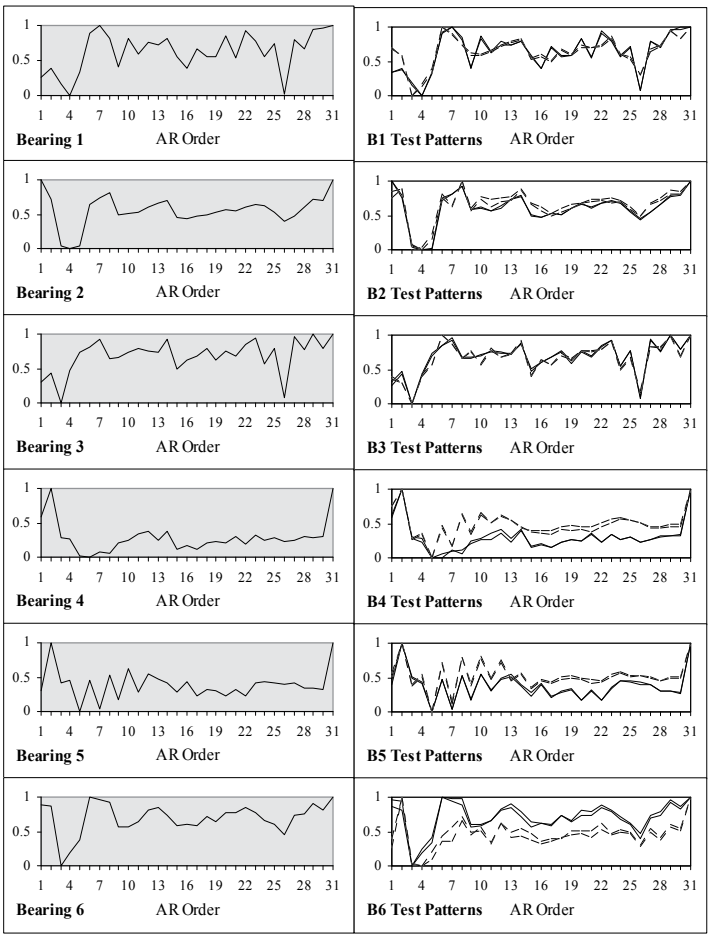


Fig. 8. AR parameters patterns of defective bearings

4. Summary and conclusions

This paper presents an integrated Intelligent Diagnostic System (IDS). Several unique features have been added to ISDS, including the advanced vibration trending techniques, the data reduction and features extraction through AR parametric model, the multi-channel and on-line capabilities, the user-friendly graphical display and control interface, and a unique machine diagnostic scheme through the modified ARTMAP neural network.

Based on the ART2 architecture, a modified ARTMAP network is introduced. The modified ARTMAP network is capable of supervised learning. In order to test the performance and robustness of the modified ARTMAP network in ISDS, an extensive bearing fault experiment has been conducted. The experimental results show ISDS is able to detect and identify several machine faults correctly (e.g., ball bearing defects in our case).

5. Appendix

5.1 Time series autoregressive (AR) parametric model

According to the features representation requirements in signal pattern recognition, if the features shown by raw data are ambiguous, then it is necessary to use a preprocessor or transformation method on the raw data. Such a preprocessor should have feature extraction capability that can invariably transfer raw data from one domain to another. The objective of this preprocessing stage is to reveal the characteristics of a pattern such that the pattern can be more easily identified.

The most important feature provided in vibration signals is frequency. Therefore, the characteristics of vibration signals can be shown clearly in the frequency domain. Traditionally, the Fast Fourier Transform (FFT) based spectral estimators are used to estimate the power spectral density (PSD) of signals. Recently, many parameter estimation methods have been developed. Among them, the autoregressive (AR) modeling method is the most popular (Gersch & Liu, 1976). The major advantage of using the parametric spectral estimation method is its ability to translate a time signal into both frequency (PSD) domain and parameter domain. In addition, parametric spectrum estimation is based on a more realistic assumption and does not need a long data record to get a high resolution spectrum.

5.2 Parametric autoregressive spectral estimation

Vibration signals can be treated as if they were generated from a time series random process. Now consider a time series x_n,

$$x_n, \quad n = -\infty, \ldots, 0, \ldots, \infty \tag{A.1}$$

where the observed interval is from n = 1, ..., N. The autoregressive model of x_n is given in Equation (A.2).

$$x_n = -a_1 x_{n-1} - a_2 x_{n-2} - \ldots - a_p x_{n-p} + e_n \tag{A.2}$$

where e_n is the prediction error, and p is the order of the model. The parametric spectrum may be computed by plugging all p a_k parameters into the theoretical power spectral density (PSD) function defined from Equation (A.3).

$$P_{AR}(f) = \frac{2\,\Delta t\ \sigma^2}{\left(1 + \sum_{k=1}^{P} a_k\ \exp\left(-i2\pi fk\Delta t\right)\right.^2}$$
$$-\frac{1}{2} \le f \le \frac{1}{2} \tag{A.3}$$
$$\Delta t = \frac{1}{S}$$

where S is the sampling rate used in data acquisition, f is the fraction of the sampling rate, p is the prediction lag or order of the AR model, and σ^2 is the variance. Therefore, if the prediction coefficients, a_k, can be estimated accurately, the parametric spectrum, $P_{AR}(f)$, of the random process can be calculated correctly through Equation (A.3).

Several approaches are available for estimating the AR model parameters. It has been observed that if the data consist of sinusoids with white noise, the peak location in the AR spectral estimate critically depends on the phase of the sinusoid (Swingler, 1980). The degree of phase dependence varies with different parameter estimation methods. Of all the AR parameter estimation methods, the modified covariance method appears to yield the best results (Kay, 1988). The modified covariance method appears to yield statistically stable spectral estimates with high resolution (Kay, 1988). For data consisting of sinusoids with white noise, a number of desirable properties have been observed (Kay, 1988; Marple, 1987):

1. The shifting of the peaks from the true frequency locations due to additive noise appears to be less than many other AR spectral estimators.
2. The peak location affected by initial sinusoidal phase is considerably reduced.
3. Spectral line splitting in which a single sinusoidal component gives rise to two distinct spectral peaks has never been observed.

5.3 AR order selection

A major issue with the parametric method is determining the AR order for a given signal. It is usually a trade-off between resolution and unnecessary peaks. Many criteria have been proposed as objective functions for selecting a "good" AR model order. Akaike has developed two criteria, the final prediction error (FPE) (Akaike, 1969) and Akaike information criterion (AIC) (Akaike, 1974). The FPE for an AR process is defined as follows:

$$\mathrm{FPE}(p) = \hat{\sigma}_p^{\,2}\left(\frac{N+(p+1)}{N-(p+1)}\right) \tag{A.4}$$

where N is the number of data samples, p is the order, and $\hat{\sigma}_p^{\,2}$ is the estimated variance at order p. The order p selected is the one for which the FPE value is the minimum. The AIC for an AR process has the following form:

$$\mathrm{AIC}(p) = N\ln\left(\hat{\sigma}_p^2\right) + p\ln(N) \tag{A.5}$$

The criteria presented here may be simply used as guidelines for initial order selection, which are known to work well for true AR signals; but may not work well with real data, depending on how well such data set is modeled by an AR model. Therefore, both FPE and AIC have been adapted in this research for the AR order suggestion.

Figure 3.3 displays an example of FPE and AIC criteria map. The signal used here is the same one shown in Figure 3.2. All rescaled FPE(p) and AIC(p) values at different AR order p are calculated and plotted in Figure A.1. The order searching range is from 1 to 80. The AIC reaches its minimum at p equal to 49. With the FPE, the minimum values are obtained when AR order is 59. Comparing these two orders by looking at their AR spectra, an order of 49 is able to produce a relatively good resolution spectrum while an order of 59 does not improve the resolution by much. Therefore, 49 may be selected as the AR model order for this signal.

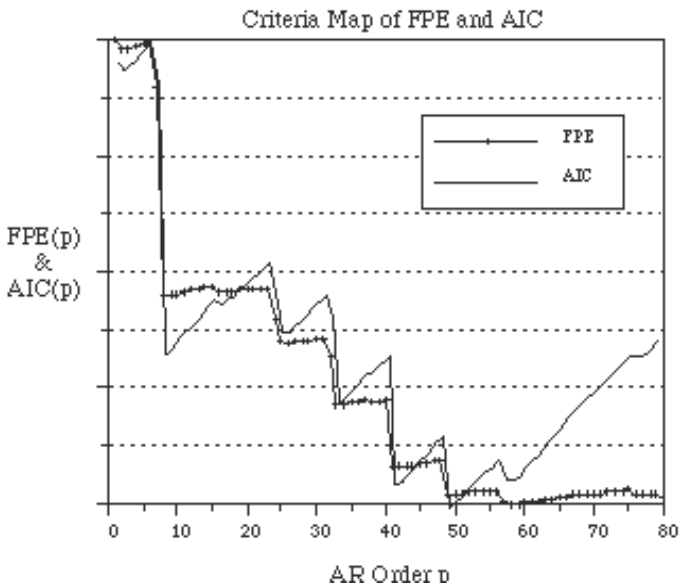


Fig. A.1 Criteria map of FPE and AIC

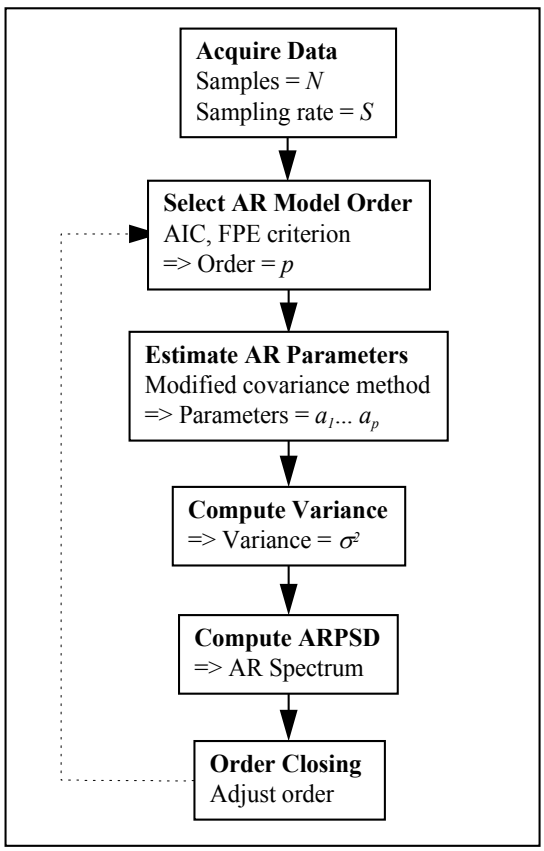


Fig. A.2 AR parameters and PSD estimation flow chart

Fig. A.2 summarizes the flow chart of calculating AR parameters and ARPSD to conclude this section.

5.4 Trending techniques for vibration condition

In order to monitor the condition of a machine throughout its operational life, several vibration trending techniques have been investigated. Vibration trending indices allow the relative machine condition to be plotted with respect to time. From the trending plot most gradual changes relating to the condition of machines can be detected.
Each signal could have more than one trending index associated with it. Furthermore, each trending index, which may be treated as a different aspect of the signal, carries different sensitivities for different machine fault types. In the case of vibration, several trending monitoring techniques have been developed and studied (Mathew, 1989; Dyer & Stewart, 1978; Mathew & Alfredson, 1984; Spoerre, 1993). In this instance, EWMA (Exponential Weighted Moving Average), RMS (Root Mean Square). The mathematical description of each method is given in the following sections.

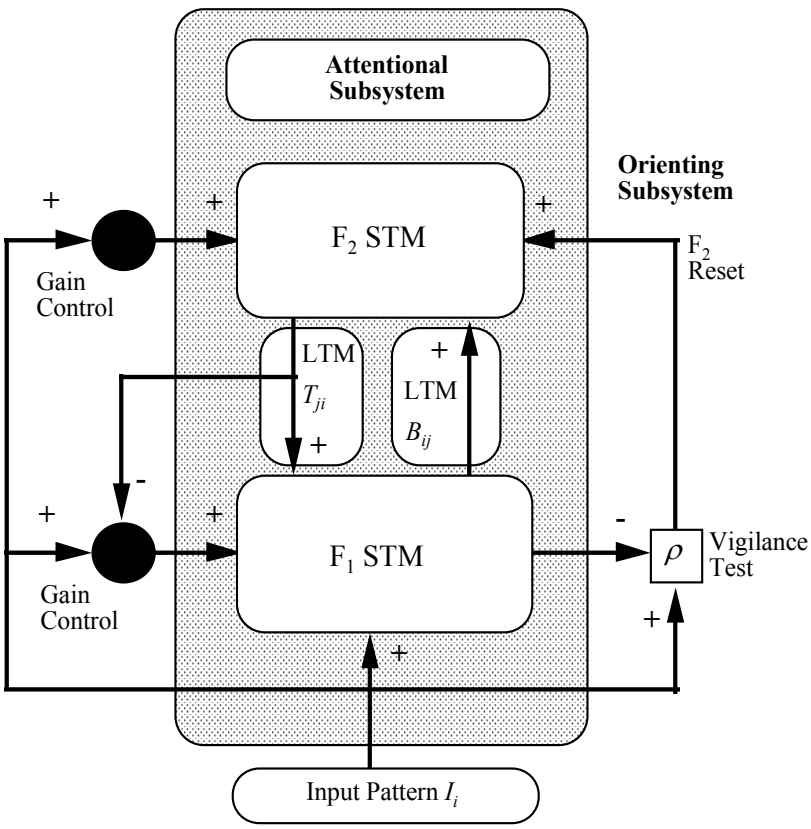
5.5 ART2 neural network

Adaptive Resonance Theory (ART) is first introduced by Grossberg in 1976 (Grossberg, 1976a). This theory emerged through an analysis of neural networking mechanisms, namely, the hypothetical STM (Short-Term Memory) and the LTM (Long-Term Memory) architectures of the human brain. The theory has been claimed to be capable of self-organizing and self-stabilizing learning in real time in an arbitrarily changing complex input environment (Banquet & Grossberg, 1987; Grossberg, 1976a; Grossberg, 1976b). Over the years, ART has steadily developed as a physical theory to explain cognitive information processing and has been applied to many pattern classification problems (Carpenter et al., 1991).
The architecture designed by the ART algorithm performs pattern clustering and is trained without supervision. Analyses showed this type of top-down feedback learning scheme could significantly overcome the problem of unstable learning, such as local minimum problem in the back propagation algorithm (Grossberg, 1987a).

5.6 Basic concept of the adaptive resonance theory

The basic ART architecture includes two subsystems, an attentional subsystem and an orienting subsystem. When learning or classification occurs within the ART architecture these two functionally complementary subsystems are activated to process familiar and unfamiliar patterns. Fig. A.3 illustrates the anatomy of the ART attentional-orienting system. At first, familiar patterns are processed within the attentional subsystem, which is built up from a competitive learning network. The second subsystem, the orienting subsystem, resets the attentional subsystem when a unfamiliar pattern occurs. Interactions between these two subsystems help to express whether a novel pattern is familiar and well represented by an existing category code, or unfamiliar and in need of a new category code.
In the attentional subsystem two successive stages, the feature representation field (F_1) and the category representation field (F_2), encode input patterns into a form of short term memory. Bottom-up and top-down pathways between F_1 and F_2 contain long term memory traces. Those traces are represented as weight vectors B_{ij} and T_{ji} in Fig. A.3 When a new input pattern arrives, it is then transformed into an activating pattern as an STM form in F_1. This STM pattern is then multiplied, or gated, by the pathway's bottom-up LTM traces.

After the LTM gated signal reaches F_2, the signal is quickly transformed by interactions among the nodes in F_2. The resulting pattern is then stored as another STM in F_2. Just like the new pattern gated by the bottom-up adaptive filter, the STM pattern in F_2 is gated by the top-down LTM traces and summed up as an internal pattern which is then called a top-down template, or learned expectation to F_1. As soon as a top-down template is generated, F_1 acts to match the top-down template against the current STM pattern in F_1. If a mismatch occurs in F_1, the orienting subsystem is engaged, thereby leading to deactivate the current STM in F_2. After that, a new active STM pattern in F_2 is produced. This generates a new top-down template pattern through top-down traces again. The search ends when an STM pattern across F_2 reads out a top-down template which matches the current STM in F_1 to the degree of accuracy required by the level of the vigilance parameter. In this case, the bottom-up and top-down LTM traces are adaptively adjusted according to the current internal STM in F_1. Otherwise, a new classification category is then established as a bottom-up code and a new top-down template is learned.



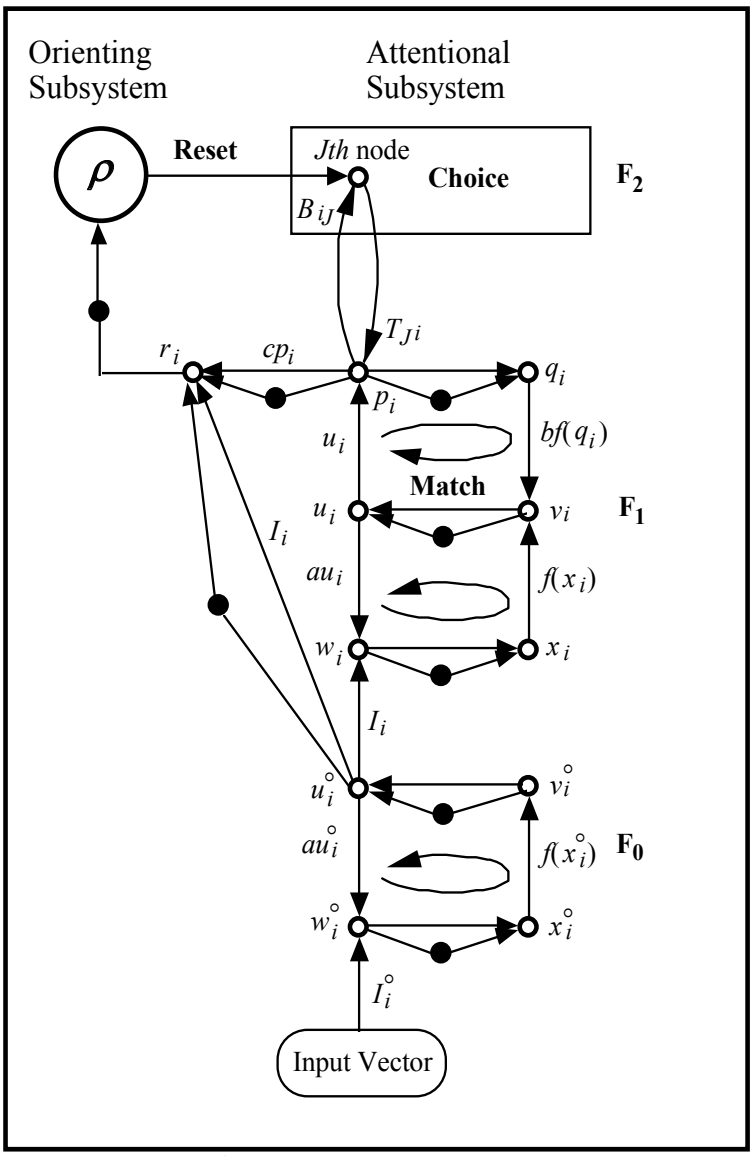
(Adapted from Carpenter and Grossberg, 1987b)

Fig. A.3 typical ART module.

By this fashion, a rapid series of the STM matching and resets may take place. Such an STM matching and reset series controls the system's hypothesis testing and search of the LTM by sequentially engaging the novelty-sensitive orienting subsystem.

5.7 ART2 system dynamics

The mathematical representation of ART2 dynamics is discussed in this section. Fig. A.4 illustrates an ART2 architecture that includes the principal components of ART modules, the attentional subsystem, and the orienting subsystem. In the attentional subsystem, there are three separate fields: an input preprocessing field, F_0, an input representation field, F_1, and a category representation field, F_2. Fig. A.4 also displays the ART2 dynamics by arrows, circles, and filled circles where arrows represent the processing directions and filled circles represent the normalization operations (i.e., Euclidean Normalization).



(Adapted from Carpenter and Grossberg, 1987b)

Fig. A.4 ART2 architecture.

6. References

Singh, G. K. & Kazzaz, S. A. S. A. (2003). Induction Machine Drive Condition Monitoring and Diagnostic Research - a Survey, *Electric Power Systems Research,* Vol. 64, pp. 145-158.

Huang, H. -H. (1993). *Transputer-Based Machine Fault Diagnostic, Ph.D. Dissertation,* Department of Industrial Engineering, The University of Iowa, Iowa City, Iowa.

Knapp, G. M. and Wang, H. -P. (1992). Machine Fault Classification: A Neural Network Approach, *International Journal of Production Research*, Vol. 30, No. 4, pp. 811-823.

Knapp, G. M., (1992). *Hierarchical Integrated Maintenance Planning for Automated Manufacturing Systems, Ph.D. Dissertation*, Department of Industrial Engineering, The University of Iowa, Iowa City, Iowa.

Carpenter, G. A. & Grossberg, S. (1987b). ART2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns, *Applied Optics*, Vol. 26, No. 23, December, pp. 4919-4930.

Carpenter, G. A.; Grossberg, S. & Reynolds, J. H. (1991). ARTMAP: Supervised Real-Time Learning and Classification of Non-stationary Data by a Self-Organizing Neural Network, *Neural Networks*, Vol. 4, pp. 565-588.

Spoerre, J. K. (1993). Machine Performance Monitoring and Fault Classification Using an Exponentially Weighted Moving Average Scheme, Master Thesis, Department of Industrial Engineering, The University of Iowa, Iowa City, IA.

Monk, R. (1972). Vibration Measurement Gives Early Warning of Mechanical Faults, *Process Engineering*, November, pp. 135-137.

Wheeler, P. G. (1968). Bearing Analysis Equipment Keeps Downtime Down, *Plant Engineering*, Vol. 25, pp. 87-89.

Gersch, W. & Liu, T. S. (1976). Time Series Methods for the Synthesis of Random Vibration Systems, *ASME Journal of Applied Mechanical,* Vol. 43, No. 1, pp. 159-165.

Akaike, H. (1969). Power Spectrum Estimation through Autoregression Model Fitting, *Ann. Inst. Stat. Math.*, Vol. 21, pp. 407-419.

Akaike, H. (1974). A New Look at the Statistical Model Identification, *IEEE Transactions on Automation Control*, Vol. AC-19, December, pp. 716-723.

Gersch, W.; Brotherton, T. & Braun, S. (1983). Nearest Neighbor-Time Series Analysis Classification of Faults in Rotating Machinery, *Transactions of the ASME*, Vol. 105, April, pp. 178-184.

Swingler, D. N. (1980). Frequency Errors in MEM processing, *IEEE Transactions on Acoustic, Speech, Signal Processing*, Vol. ASSP-28, April, pp. 257-259.

Kay, S. M. (1988). Modern Spectral Estimation: Theory and Application, Prentice Hall Inc., NJ.

Marple, S. L. Jr. (1987). *Digital Spectral Analysis with Applications*, Prentice Hall Inc., NJ.

Mathew, J. & Alfredson, R. J. (1984). The Condition Monitoring of Rolling Element Bearings Using Vibration Analysis," Journal of Vibration, Acoustics, *Stress, and Reliability in Design,* Vol. 106, July, pp. 447-453.

Mathew, J. (1989). Monitoring the Vibrations of Rotating Machine Elements--An Overview, *ASME*, DE Vol. 18-5, pp. 15-22.

Dyer, D. & Stewart, R. M. (1978). Detection of Rolling Element Bearing Damage by Statistical Vibration Analysis, *Transactions of the ASME Journal of Mechanical Design*, Vol. 100, April, pp. 229-235.

Grossberg, S. (1976a). Adaptive Pattern Classification and Universal I: Parallel Development and Coding of Neural Feature Detectors, *Biological Cybernetics,* Vol. 23, pp. 121-134.

Grossberg, S. (1976b). Adaptive Pattern Classification and Universal II: Feedback Expectation Olfaction and Illusions, *Biological Cybernetics,* Vol. 23, pp. 187-202.

Grossberg, S. (1987a). Competitive Learning: from Interactive Activation to Adaptive Resonance, *Cognitive Science*, Vol. 11, pp. 23-63.

Banquet, J. P. & Grossberg, S. (1987). Probing Cognitive Processes Through the Structure of Event-related Potentials during Learning: an Experimental and Theoretical Analysis, *Applied Optics*, Vol. 26, No. 23, December, pp. 4931-4944.

21

Conditioning Monitoring and Fault Diagnosis for a Servo-Pneumatic System with Artificial Neural Network Algorithms

Mustafa Demetgul[1], Sezai Taskin[2] and Ibrahim Nur Tansel[3]
[1]*Marmara University, Technical Education Faculty, Mechanical Education Department, Goztepe, Istanbul,*
[2]*Celal Bayar University, Department of Electronics and Automation, Turgutlu, Manisa,*
[3]*Florida International University, Department of Mechanical and Materials Engineering, 10555 W. Flagler St. EC 3400, Miami, FL,*
[1,2]*Turkey*
[3]*USA*

1. Introduction

On-line monitoring of manufacturing process is extremely important in modern manufacturing for plant safety, maximization of the production and consistency of the product quality (Song et al., 2003). The development of diagnostic systems for the industrial applications has started in early 1970s. The recent developments in the microelectronics have increased their intelligence and let them found many industrial applications in last two decades (Mendonca et al., 2009; Shi & Sepehri, 2004). The intelligent data analysis techniques are one of the most important components of the fault diagnosis methods (Uppal et al, 2002; Uppal & Patton, 2002). In this study, the faults of a pneumatic system will be monitored by using the artificial neural networks (ANN).

When the speed control and magnitude of the applied force is not critical, pneumatic systems are the first choice. They are cheap, easy to maintain, safe, clean, and components are commercially available. They have even been used for precise control of industrial systems (Nazir & Shaoping, 2009; Ning & Bone, 2005). Unfortunately, their nonlinear properties and some limitations at their damping, stiffness and bandwidth characteristics avoid their widespread applications (Belforte et al., 2004; Tsai & Huang, 2008, Bone & Ning, 2007; Taghizadeh et al., 2009; Takosoglu et al., 2009).

The interest for the development of diagnostic methods for pneumatic and hydraulic systems has increased in the last decade (Nakutis & Kaškonas, 2008). Researchers concentrated on the detection of the faults of the components. The condition of the pneumatic and hydraulic cylinders (Wang et al., 2004), and digitally controlled valves (Karpenko et al., 2003) were the main focus of the studies. Some of the other considered faults were leakage of the seals (Nakutis & Kaškonas, 2005, 2007; Yang, 2006; Sepasi & Sassani, 2010), friction increase (Wang et al., 2004; Nogami et al., 1995) and other

malfunctions (Bouamama et al., 2005). The monitored signals can be divided into two groups according to their frequencies. Acoustic emission is an excellent example of high frequency monitoring signal (Yang, 2006; Chena et al., 2007). The frequency of the pressure, flow, and timing signals are low (Sepasi & Sassani, 2010; Nogami et al., 1995; Bouamama et al., 2005; Nakutis & Kaškonas, 2005, 2008; Wang et al., 2004; Karpenko et al., 2003; Li & Kao, 2005; McGhee et al., 1997). The gathered signals are encoded to obtain their most descriptive features. The encoded signals were classified by using various classification techniques such as ANNs (Karpenko et al., 2003; Nakutis & Kaškonas, 2003; Sepasi & Sassani, 2010; Nogami et al., 1995; McGhee et al., 1997), fuzzy method (Mendonca et al., 2009; Uppal & Patton, 2002), neuro fuzzy method (Shi & Sepehri, 2004; Uppal & Patton, 2002), statistical technique (Song et al., 2003), bond graphs (Bouamama et al., 2005), genetic programming (Wang et al., 2004; Yang, 2006), and expert/intelligent systems (Chen & Mo, 2004).
It is not difficult to develop programs for classification of the sensory signals of pneumatic systems. However, these programs should be carefully modified when the characteristics of the signals change. Many researchers have worked on the development of ANNs. Generally, most of the ANNs are ready to take the advantage of future parallel hardware. By considering these facts ANNs will be used for the classification in this study.
Mainly, there are two types of ANNs: supervised and unsupervised. The supervised ANNs require an initial training. Unsupervised ones may start to monitor the signals without any training. Among the supervised ANNs, the feed-forward ANNs (FFNN) have been widely used. The Back-propagation (BP) algorithm is the most popular one for estimation of the weights and were used in many applications (Bryson & Ho, 1969, Rumelhart et al., 1976, Huang et al., 2007; Lu et al., 2000; Tansel et al., 2009; Aykut et al., 2010; Tansel & Demetgul & Sierakowski, 2009; Demetgul et al., 2009). Quasi-Newton approaches such as Levenberg-Marquardt was developed to increase the speed of the estimation and is available in the MATLAB ANN Toolbox (Beale et al., 2010). Fuzzy ARTMAP method (Carpenter et al., 1991, Carpenter et al., 1992) allowed the use of the Adaptive Resonance Theory (ART) for the supervised learning (Grossberg, S., 1987). Among the unsupervised ANNs Adaptive Resonance Theory 2 (ART2) (Grossberg, S., 1987, Carpenter& Grossberg, 1987, Rajakarunakaran et al., 2008, Lee et al., 2003, Belforte et al., 2004) has been successfully used for classification in many applications. This approach was improved further by the development of fuzzy ART (Carpenter et al., 1991a, Carpenter et al., 1991b). In this study the data was classified by using the BP, fuzzy ARTMAP, ART2 and fuzzy ART.
In the following section the theoretical background of the ANNs will be presented very briefly. The experimental setup, results and the conclusion will follow it.

2. Theoretical background of the tested ANNs

In this section the ANNs will be very briefly reviewed since detailed information is available at the listed references.

2.1 Supervised ANN

In this study, two supervised ANNs were used. FFNN became popular with the widespread use of the BP (Bryson & Ho, 1969, Rumelhart et al., 1976) algorithm. The FFNN have multiple layers. Generally, single hidden layer is used. The user determines the number of the hidden neurons of this layer by trial and error. The number of the neurons of the input

and output layers depends on the application. The BP estimates the weights of the neurons by updating them after the forward and backward propagation of error. The learning rate and the momentum are two important parameters of the BP for training the network successfully (Chen & Mo, 2004; McGhee et al., 1997). Levenberg-Marquardt algorithm (Beale et al., 2010) generally estimates the parameters of the FFNNs. It finds the best weights by minimizing the function. It works effectively for many applications. Levenberg-Marquardt algorithm available at the MATLAB toolbox was used in this study (Beale et al., 2010).
Fuzzy ARTMAP (Carpenter et al., 1991, Carpenter et al., 1992) use the fuzzy logic and ART ANNs. It evaluates the similarity by considering the fuzzy subsethood and ART category choice. The vigilance is used to determine the size of the "category boxes" or sensitivity of the ANN. One of the very important advantages of the ARTMAP with or without the fuzzy component over the FFNNs is the use of the vigilance based on our experience. Aaron Garrett's (Garrett, 2003) code was used for the training and testing of the fuzzy ARTMAP method.

2.2 Unsupervised ANN

ART2 type ANN evaluates the characteristics of the inputs and assign them a category (Carpenter & Grossberg, 1987; Lee et al., 2003, Yang et al., 2004, Na et al., 2008). If the signal looks like one of the previously presented signals, it will be classified in the same category. On the other hand, if the signal is different than the previously presented ones a new category is assigned for it. The sensitivity of ART2 depends to the vigilance. At the low vigilances, it has higher tolerance. When the vigilance approaches to one it will be more selective.
Fuzzy ART use fuzzy set theory in the ART1 type ANN structure. With the help of the MIN operator of the fuzzy set theory the classification of the binary and analog input patterns is possible. The vigilance parameter adjust the selectivity of the ANN. In this study Aaron Garrett (Garrett, 2003) implementation of the fuzzy ART was used.

3. Experimental setup and performed experiments:

The diagram of the experimental setup is presented in Fig.1. The pneumatic system created motions along the X and Y axes. The operation of the system was managed by an SPC 200 two axis servopneumatic controller. Each axes could be operated in the coordinated or autonomous mode. Controller was also responsible from the digital I/O including the communications with the other devices. A pressure transducer was used to measure the supply pressure of the system. 5/3-way proportional valves controlled the flow of the pressurized air into the proper chambers of the cylinders.
A proportional valve (Festo MPYE-5 1/8 LF-010B) controlled the displacement of pneumatic cylinder in the x direction. The valve was connected to the both chambers of the pneumatic rodless cylinder (Festo DGPL-25-450-PPV-A-B-KF-GK-SV). The stroke length and the diameter of the cylinder were 450 mm and 25 mm respectively. A linear potentiometer (Festo MLO-POT-450-TLF) was attached to the side on the actuator to measure the piston position. The valve had the neutral spool position under 5 V control voltage.
Another pneumatic rodless cylinder (Festo DGPIL-25-225-PPV-B-KF-AIF-GK-SV-AV) created the motion in the Y direction. The stroke length and the diameter of the cylinder were 225 mm and 25 mm respectively. A contactless absolute magnetostrictive linear displacement sensor was used to measure the strokes of the piston. A gripper was attached to the cylinder.

1. Air compressor
2. Service Unit
3. SPC 200 Controller
4. Analog Pressure Transducers
5. Gripper
6. Y axis
7. Linear Potentiometer for x axis
8. X axis
9. NI Compact FieldPoint System
10. Power Supply

Fig. 1. Servo-pneumatic positioning system of the Festo Didactic. The components of the system are the following:

Experimental data was collected by using the National Instrument (NI) compact FieldPoint measurement system with control modules. The LabVIEW program environment controlled the measurement system. The values of four analog parameters were monitored. Three of these parameters were the pressure readings of the cylinders creating the motion in the x and y directions and the overall system. The Fourth analog input was the readings from the linear potentiometer. The gripper action was monitored from the digital signals coming from data acquisition card. The diagram of the components of the servo-pneumatic system is shown in Figure 2.

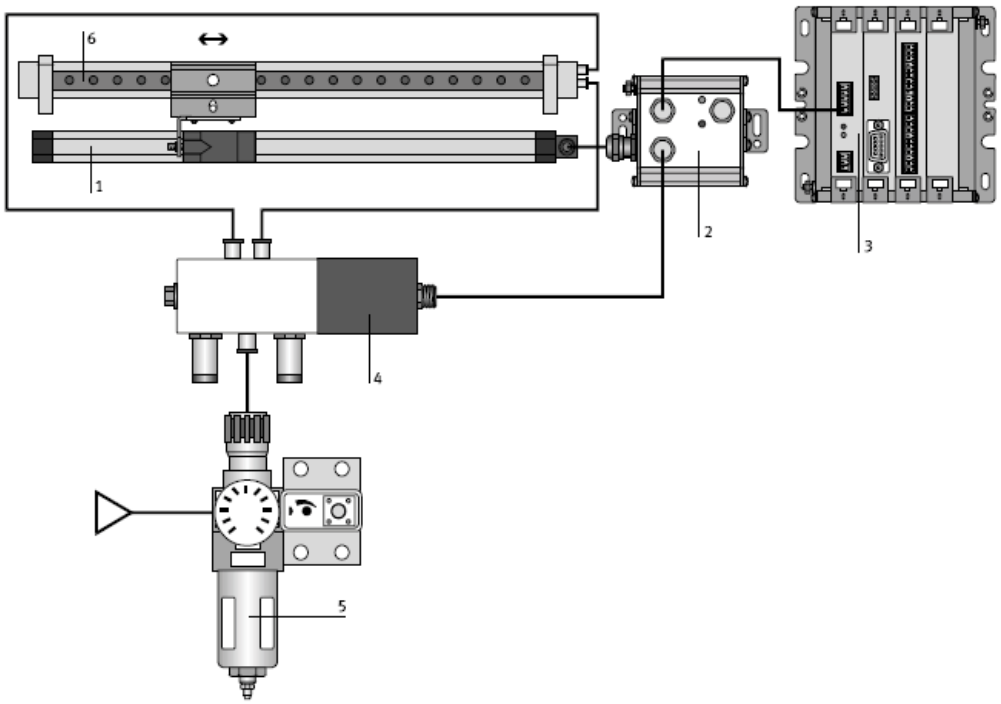


Fig. 2. The servo-pneumatic system components for X axis. (1. Measuring system, 2. Axis interface, 3. Smart positioning controller SPC 200, 4. Proportional directional control valve, 5. Service unit, 6. Rodless cylinder) ((festodidactic.com, 2010)

The servo-pneumatic system simulated the operation of food preparation. Jars were put individually on a conveyor belt by the packaging system. A handling device with servo-pneumatic NC axis transferred these jars to a pallet. The precise motion of the NC axis is essential for completion of the task (Festo Didactic, 2010).

The user interface of the LabVIEW program is presented in Fig. 3. The display shows the pressures of the overall system and two cylinders creating the motions along the X and Y axes. Also the displacement of one of the cylinder and gripper action (pick and place) is demonstrated.

In this study, the pneumatic system was operated at the normal and 4 different faulty conditions. The experimental cases are listed in Table 1. There were 15 experimental cases. The data was collected at the same condition 3 times when the system was operated in the normal and 4 faulty modes.

Operational condition	Experiment #	Recalled as
Normal operation of the Servo Pneumatic System	1	Normal
x axis error positioning	2	Fault 1
y axis error positioning	3	Fault 2
Pick faults for gripper	4	Fault 3
Place faults for gripper	5	Fault 4

Table 1. Operating conditions

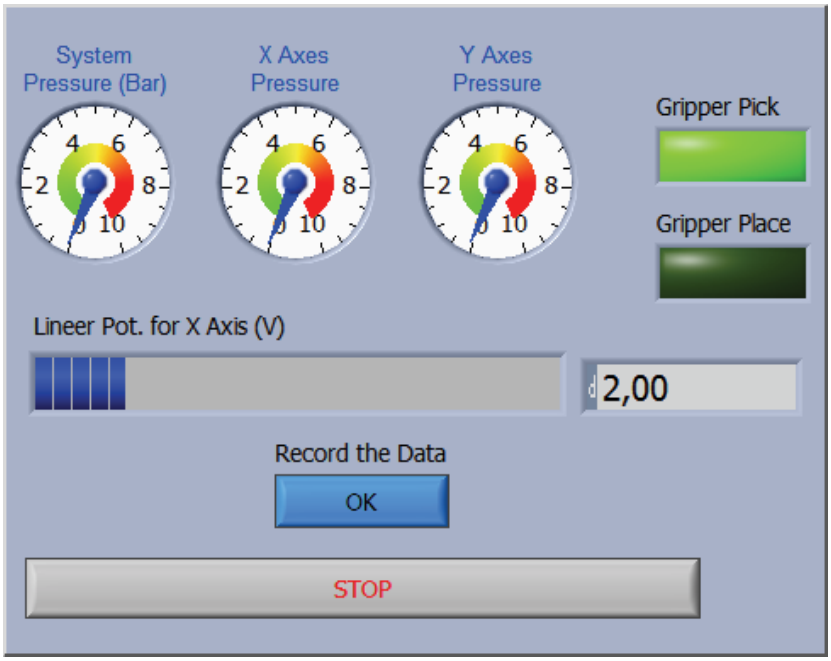


Fig. 3. Data collection visual front panel of LabVIEW

The signals of the gripper pick (Fig.4) and place (Fig.5) sensors, the pressure sensors of the cylinders in the x (Fig.6) and y (Fig.7) directions, the voltage output of the linear potentiometer of the x axis (Fig.8) are presented in the corresponding figures.

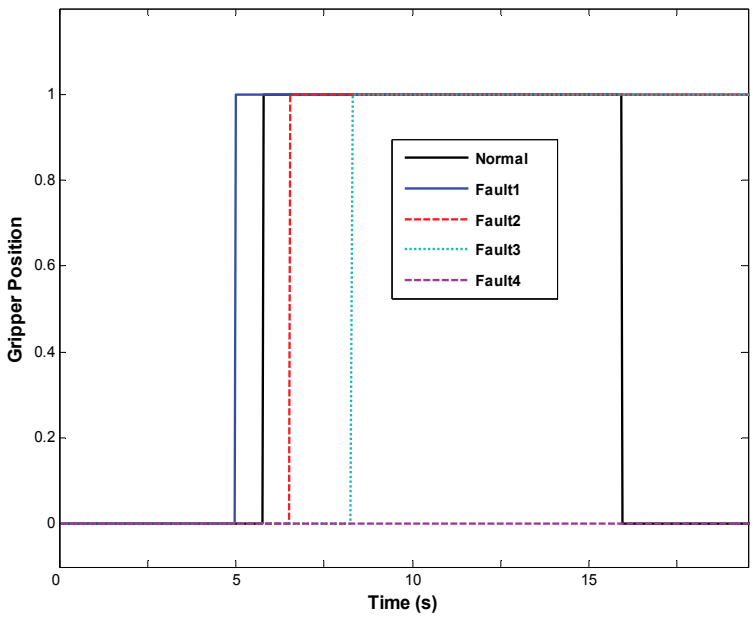
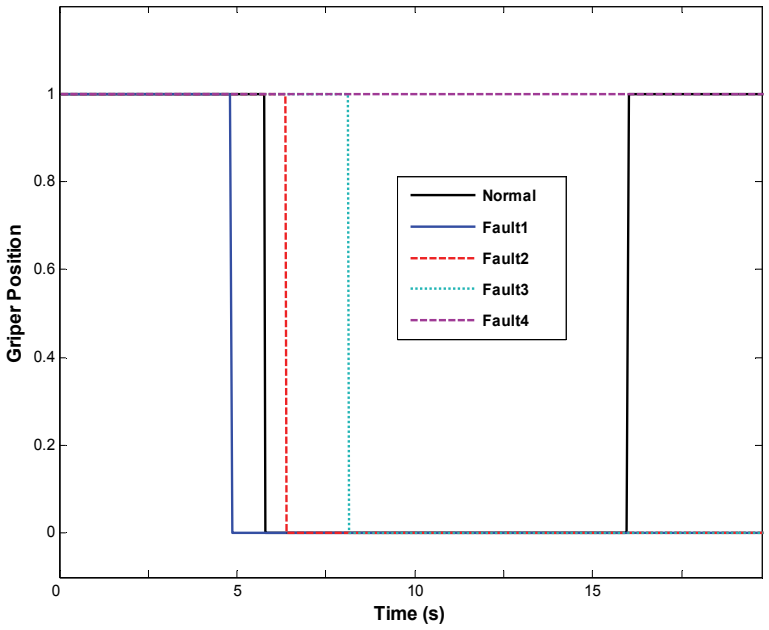


Fig. 4. Gripper Pick



Sensor

Fig. 5. Gripper Place Sensor

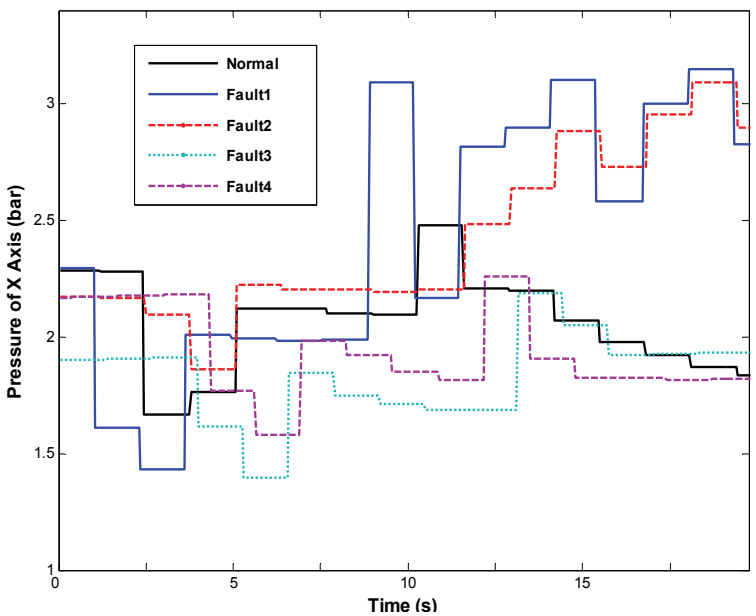
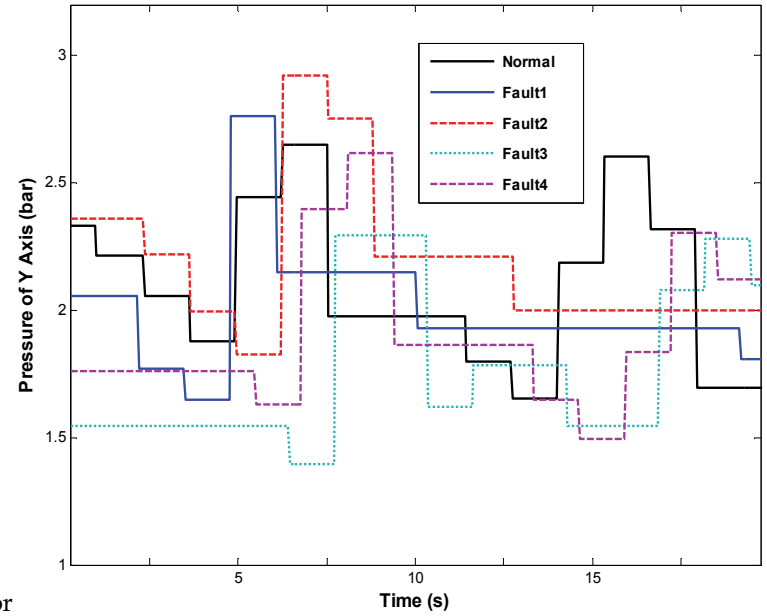


Fig. 6. X Axis Pressure



Sensor

Fig. 7. Y Axis Pressure Sensor

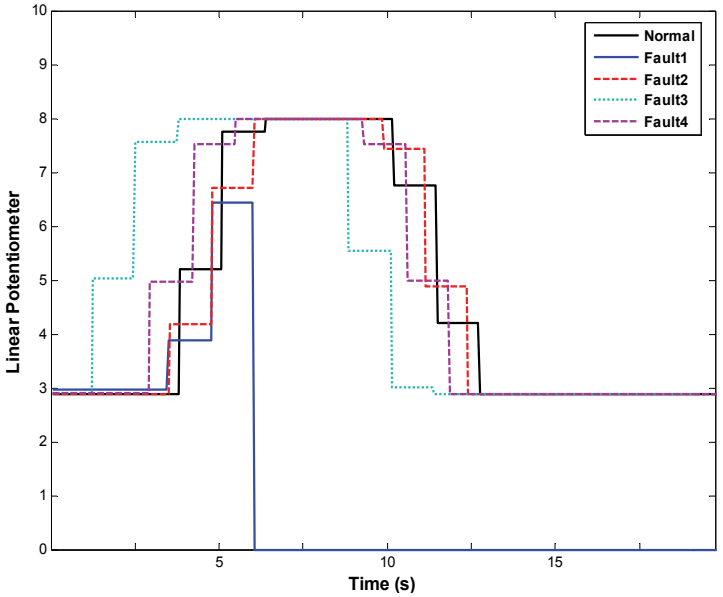


Fig. 8. X Axis linear potentiometer signal

4. Proposed encoding method

The sensors provided long data segments during the operation of the system. To represent the characteristics of the system the sensory signals were encoded by selecting their most descriptive futures and presented to the ANNs.

Two gripper sensor signals were monitored one for pick (Fig.4) and one for place (Fig.5). Their outputs were either 0 V or 1V. The gripper pick and place signals were encoded by identifying the time when the value raised to 1V and when it fell down to 0V. The signals of the pressure of x axis (Fig.6), pressure of y axis (Fig.7) and main pressure were encoded by calculating their averages. For the linear potentiometer (Fig.8) the times when the signal fell below 7V and when it went over 7V were identified and used during the classification.

5. Results

The expected results from the ANN classification are presented in Fig.9. Ideally, once the ANN experiences the normal and each faulty mode, someone may expect it to identify each one of them accurately. In our case this means, an unsupervised ANN create maximum 5 categories and assign each one of them to the normal and 4 fault modes. Similarly, the output of the supervised ANNs are supposed to be an integer value between 1 and 5 depending on the case. It is very difficult to classify the experimental data in 5 different categories unless the encoded cases have very different characteristics, repeatability is very high and noise is very low. In the worst case, we expect the ANN to assign at least two categories and locate the normal operation and faulty ones in separate categories. The output of the supervised ANN could be 0 and 1 in such cases. The ANN estimates in the ideal and accdptialbe worst case scenario are demonstrated in Fig.9. In the following sections, the performance of the supervised and the unsupervised ANNs are outlined.

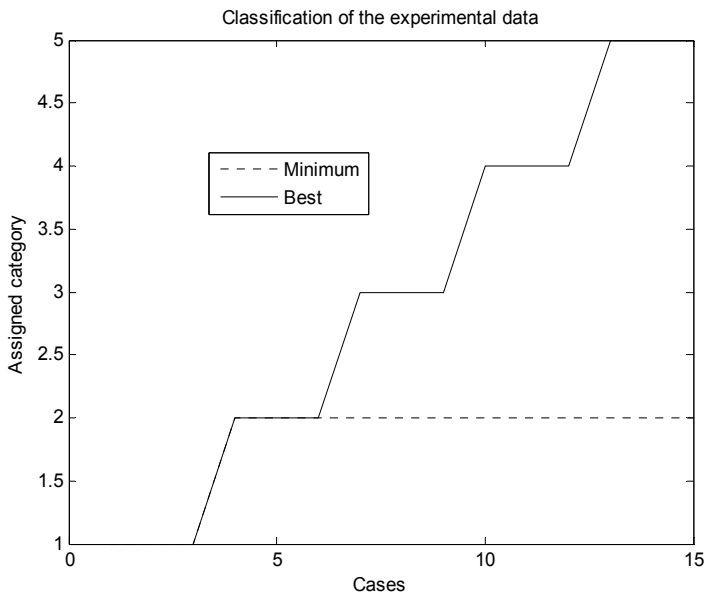


Fig. 9. The output of the ANNs for classification of normal and 4 faulty modes.

5.2 Performance of the supervised ANNs:

Performance of the feed-forward-network (FFN) was evaluated by using the Levenberg Marquardt algorithm. The FFN had 9 inputs and 1 output. The outputs of the cases were 1, 2, 3, 4, 5 for Normal, Fault1, Fault2, Fault3 and Fault4 respectively. For training only one sample of the normal and 4 faulty cases were used. Since the FNN type ANNs do not have any parameters to adjust their sensitivity they have to be trained with very large number of cases which will teach the network expected response for each possible situation. Since, one sample for each one of the normal and 4 faulty cases was too few for effective training, we generated semi experimental cases. The semi-experimental cases were generated from these samples by changing the each input with ±1% steps up to ±10%. We generated 100 semi-experimental cases in addition to the original 5 cases with this approach. The FNN had 8 neurons at the hidden layer. The FFN was trained with 105 cases.

The FFN type ANN was trained by using the Levenberg-Marquardt algorithm of the Neural Network Toolbox of the MATLAB. The training was repeated several times. The same semi-experimental data generation procedure was used to generate 200 additional test cases from the 10 experimental cases which had 2 tests at each condition (1 normal and 4 faulty ones). The average estimation errors were 5.55e-15% for the training and 8.66% for the test cases. The actual and estimated values for the training and test cases are presented in Fig.10 and Fig.11 respectively. The ANN always estimated the training cases with better than 0.01% accuracy. The accuracy of the estimations of the test cases was different at each trail. These results indicated that, without studying the characteristics of the sensory signals very

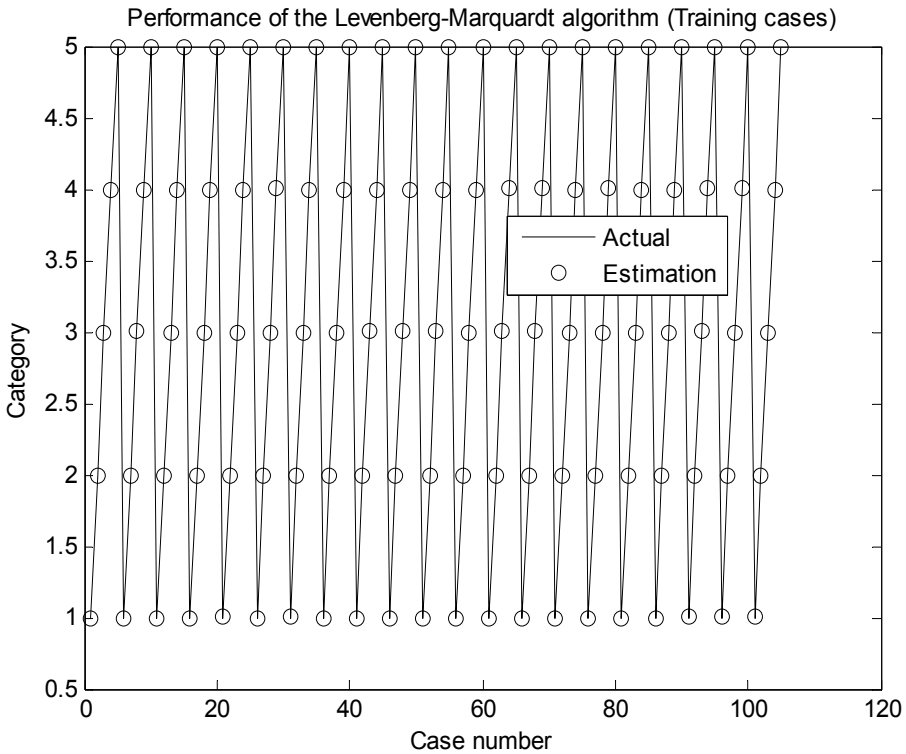


Fig. 10. The FFN type ANN estimations for the training cases.

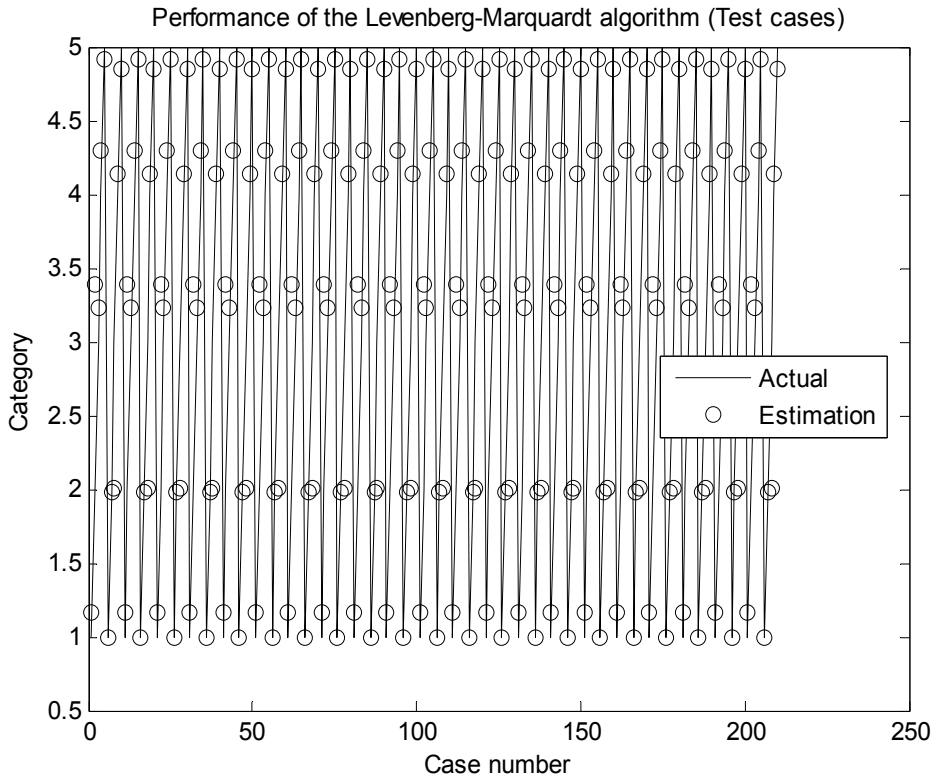


Fig. 11. The FFN type ANN estimations for the test cases.

carefully, the ANN may estimate the normal and faulty cases; however, for industrial applications the characteristics of the data may change in much larger range than ours and working with much larger experimental samples are advised.

The same analysis was repeated by using the fuzzy ARTMAP. The fuzzy ARTMAP adjusts the size of the "category boxes" according to the selected vigilance value. The ANN estimates the category of the given case as -1 if the fuzzy ARTMAP do not have proper training. So, we did not need to use the semi-experimental data. The fuzzy ARTMAP was trained by using 5 cases (normal and 4 fault modes). It was tested by using the 10 cases (2 normal and 8 faulty cases (2 samples at each fault modes)). The vigilance was changed from 0.52 to 1 with the steps of 0.02. The identical performance was observed for the training and test cases when the vigilance was selected between 0.52 and 0.83 (Fig.12). All the training cases were identified perfectly. The normal and all the faulty ones were distinguished accurately. The fuzzy ARTMAP only confused two test cases belong to Fault 2 and 3. The performance of the fuzzy ARTMAP started to deteriorate at the higher vigilances since the "category boxes" were too small and the ANN could not classify some of the test cases. The number of the unclassified cases increased with the increasing vigilance.

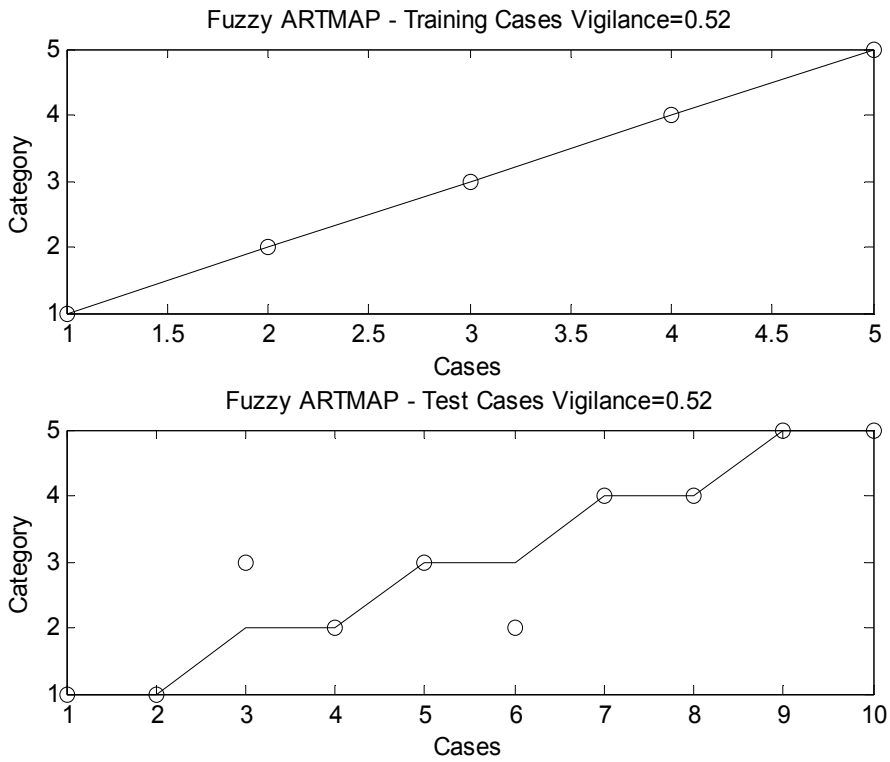


Fig. 12. The performance of the fuzzy ARTMAP type ANN.

5.1 Performance of the unsupervised ANNs:

Performance of the ART2 is shown in Table 2. It distinguished the normal and faulty cases. Among the faults, the Fault 3 was identified all the time by assigning a new category and always estimating it accurately. The ART2 could not distinguish Fault 1, 2 and 4 from each other. The best vigilance values were in the range of 0.9 and 0.9975. When these vigilances were used ART2 distinguished the normal operation, faulty cases and Fault 3. The same results are also presented with a 3D graph in Fig.13.

The results of the Fuzzy ART program are presented in Fig.14. The number of assigned categories varied between 2 and 15 for the vigilance values of 0.5 and 1. When the vigilance was 0.5, the Fuzzy ART distinguished the normal and faulty operation but could not classify the faults. Fuzzy ART started to distinguish Fault 4 when the vigilance was 0.65. It started to distinguish Fault 3 and 4 for the vigilance value of 0.77. When the vigilance reached to 0.96 it could distinguished 10 categories and classified all the cases accurately. Multiple categories were assigned to the normal and some of the faulty operation modes.

Condition of the system	Experiment	Vigilance values				
		0.9 - 0.9975	0.998	0.9985	0.999	0.9995
Normal	Test 1	1	1	1	1	1
	Test 2	1	2	2	2	2
	Test 3	1	2	2	2	2
Fault 1	Test 1	2	3	3	3	3
	Test 2	2	3	3	3	4
	Test 3	2	3	3	3	4
Fault 2	Test 1	2	3	3	3	4
	Test 2	2	3	3	3	4
	Test 3	2	3	3	3	5
Fault 3	Test 1	3	4	4	4	6
	Test 2	3	4	4	4	6
	Test 3	3	4	4	4	6
Fault 4	Test 1	2	3	3	3	5
	Test 2	2	3	3	3	5
	Test 3	2	3	3	3	7

Table 2. The estimated categories with the ART 2 algorithm

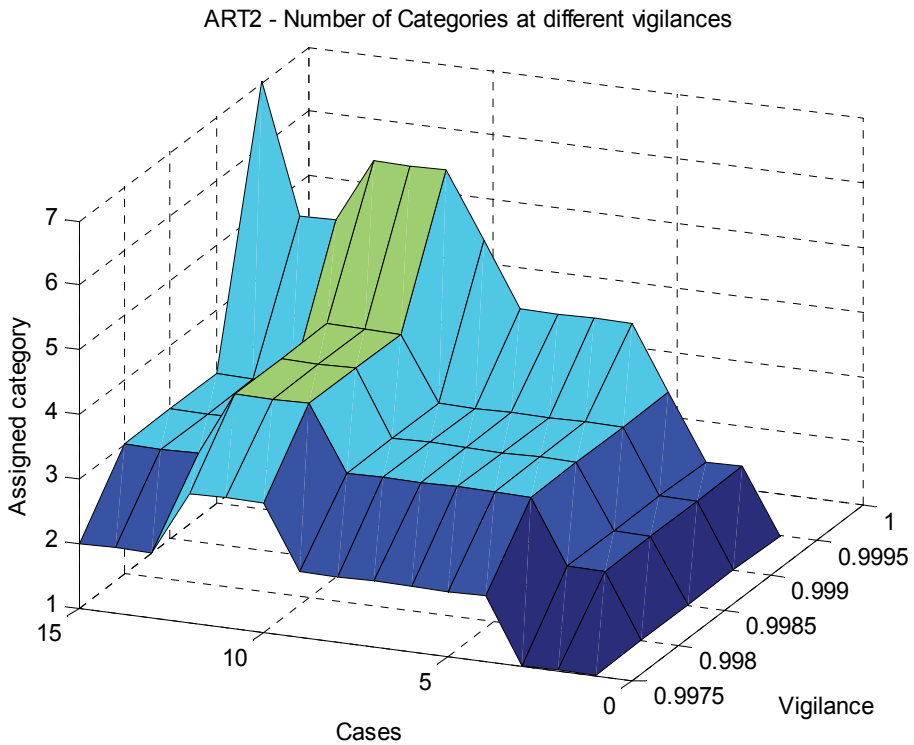


Fig. 13. The graphical presentation of the ART2 results in the Table 1.

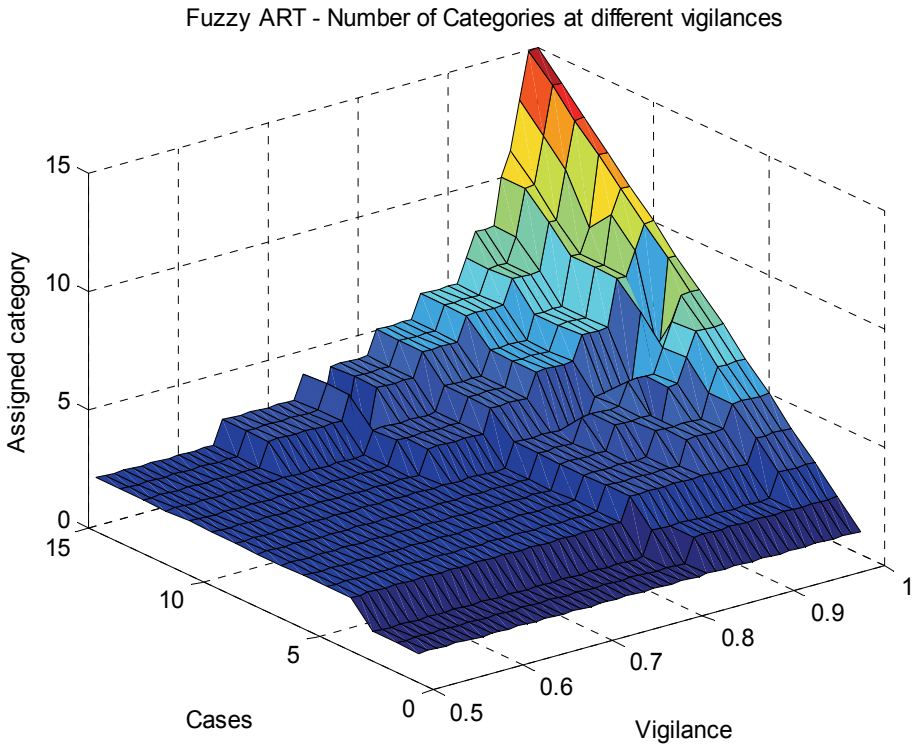


Fig. 13. The estimations of the Fuzzy ART

6. Conclusion

A two axis servopneumatic system was prepared to duplicate their typical operation at the food industry. The system was operated at the normal and 4 faulty modes. The characteristics of the signals were reasonably repetitive in each case. Three pressure, one linear displacement and two digital signals from the gripper were monitored in the time domain. The signals were encoded to obtain their most descriptive futures. There were 15 experimental cases. The data was collected at the same condition 3 times when the system was operated in the normal and 4 faulty modes. The encoded data had 9 parameters. The performances of two supervised and two unsupervised neural networks were studied.

The 5 experimental cases were increased to 105 by generating semi experimental data. The parameters of the FFN was calculated by using the Levenberg-Marquardt algorithm. The average estimation errors were 5.55e-15% for the training and 8.66% for the test cases. The fuzzy ARTMAP was trained with 5 cases including one normal and 4 faulty modes. It estimated the 8 of the 10 test cases it never saw before perfectly. It confused the two faulty cases among each other.

The ART2 and fuzzy ART were used to evaluate the performance of these unsupervised ANNs on our data. Both of them distinguished the normal and faulty cases by assigning different categories for them. They had hard time to distinguish the faulty modes from each

other. Since they did not need training, they are very convenient for industrial applications. However, it is unrealistic to expect them to assign different categories for the normal operation and each fault modes, and classify all the incoming cases accurately.

7. References

Aykut, S., Demetgul, M. & Tansel, I. N. (2010). Selection of Optimum Cutting Condition of Cobalt Based Alloy with GONN. *The International Journal of Advanced Manufacturing Technology,* Vol. 46, No. 9-12, pp. 957-967.

Beale, M. H., Hagan, M. T. & Demuth, H. B. (2010). Neural Network Toolbox 7 User Guide, Mathworks.
http://www.mathworks.com/help/pdf_doc/nnet/nnet.pdf.

Belforte, G., Mauro, S. & Mattiazzo, G. (2004). A method for increasing the dynamic performance of pneumatic servo systems with digital valves, *Mechatronics,* Vol.14, pp. 1105–1120.

Bryson, A. E., Ho, Y. C. (1975). Applied optimal control: optimization, estimation, and control. *Taylor & Francis Publishing*, pp. 481.

Bouamama, B.O. et al. (2005). Fault detection and isolation of smart actuators using bond graphs and external models, *Control Engineering Practice,* Vol. 13, pp.159-175.

Carpenter, G. A & Grossberg, S. (1987). ART-2: Self-Organization of Stable Category Recognition Codes for Analog Input Pattern, *Applied Optics*, Vol. 26, pp. 4919-4930.

Carpenter, G. A., Grossberg S., & Reynolds J. H. (1991). ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network, *Neural Networks* , Vol. 4, pp.565-588.

Carpenter, G. A., Grossberg, S. & Rosen, D. B. (1991a). ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition, *Neural Networks* , Vol. 4, pp. 493-504.

Carpenter, G. A., Grossberg, S. & Rosen, D. B. (1991b). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system, *Neural Networks* , Vol. 4, pp. 759-771.

Carpenter, G. A. et al. (1992). Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps, *IEEE Transactions on Neural Networks*, Vol. 3, pp. 698-713.

Chen, C. & Mo, C. (2004). A method for intelligent fault diagnosis of rotating machinery, *Digital Signal Processing,* Vol. 14, pp. 203–217.

Chena, P. et al. (2007). A study of hydraulic seal integrity. *Mechanical Systems and Signal Processing*, Vol. 21, pp.1115-1126.

Demetgul M.; Tansel IN. & Taskin S. (2009). Fault Diagnosis of Pneumatic Systems with Artificial Neural Network Algorithms. *Expert Systems with Applications*, Vol. 36, No. 7, pp. 10512-10519.

Festo Didactic GmbH & Co. PneuPos Manuel, 2010.
http://www.festodidactic.com/ov3/media/customers/1100/092276500112124319 4.pdf.

Garrett, A. (2003). Fuzzy ART and Fuzzy ARTMAP Neural Networks,

http://www.mathworks.com/matlabcentral/fileexchange/4306.

Gary M. & Ning, S., (2007). Experimental Comparison of Position Tracking Control Algorithms for Pneumatic Cylinder Actuators, *IEEE/ASME Transactions on Mechatronics*, October 2007.

Grossberg S. (1987). Competitive learning: From interactive activation to adaptive resonance, *Cognitive Science*, Vol. 11, pp. 23-63.

Huang, R.; Xi, L.; Li, X.; Liu, C.R.; Qiu, H. & Lee, J. (2007). Residual life predictions for ball bearings based on self-organizing map and back propagation Neural network methods, *Mechanical Systems and Signal Processing,* Vol. 21, pp.193–207.

Karpenko, M.; Sepehri, N. & Scuse, D. (2003). Diagnosis of process valve actuator faults using a multilayer Neural network. *Control Engineering Practice*, Vol. 11, pp.1289-1299.

Lee, I. S.; Kim, J. T.; Lee, J. W.; Lee, D. Y. & Kim, K. Y. (2003). Model-Based Fault Detection and Isolation Method Using ART2 Neural Network, *International Journal of Intelligent Systems*, Vol.18, pp. 1087–1100.

Li, X. & Kao, I. (2005). Analytical fault detection and diagnosis (FDD) for pneumatic systems in robotics and manufacturing automation. Intelligent Robots and Systems, IEEE/RSJ International Conference, pp.2517-2522.

Lu, P. J.; Hsu, T. C.; Zhang, M. C. & Zhang, J. (2000), An Evaluation of Engine Fault Diagnostics Using Artificial Neural Networks, *ASME J. Eng. Gas Turbines Power*, Vol. 123, pp. 240–246.

McGhee, J.; Henderson, I. A. & Baird, A. (1997). Neural networks applied for the identification and fault diagnosis of process valves and actuators, *Measurement*, Vol. 20, No. 4, pp. 267-275.

Mendonça, L. F.; Sousa, J. M. C. & Costa, J. M. G. (2009). An architecture for fault detection and isolation based on fuzzy methods, *Expert Systems with Applications.* Vol. 36 pp. 1092–1104.

Na, C. & Lizhuang, M. (2008). Pattern Recognition Based on Weighted and Supervised ART2. *International Conference on Intelligent System and Knowledge Engineering*, Xiamen, China, Nov. 17-18, 2008,Vol. 2, pp. 98-102.

Nakutis, Ž., Kaškonas, P. (2005). Application of ANN for pneumatic cylinder leakage diagnostics. *Matavimai*, Vol. 36, No.4, pp.16-21.

Nakutis, Ž. & Kaškonas, P. (2007. Pneumatic cylinder diagnostics using classification methods. Instrumentation and Measurement Technology Conference Proceedings, Warsaw, 1-3 May 2007, pp.1-4.

Nakutis, Ž. & Kaškonas, P. (2008). An approach to pneumatic cylinder on-line conditions monitoring. *Mechanika. Kaunas: Technologija*, Vol. 4, No. 72, pp.41-47.

Nazir, M. B. & Shaoping, W. (2009). Optimization Based on Convergence Velocity and Reliability for Hydraulic Servo System, *Chinese Journal of Aeronautics,* Vol. 22, pp. 407-412.

Ning, S. & Bone, G. M. (2005). Development of a nonlinear dynamic model for a servo pneumatic positioning system, *Proc. IEEE Int. Conf. Mechatronics Autom.*, pp. 43–48.

Nogami, T. et al. (1995). Failure diagnosis system on pneumatic control valves by neural networks. *IEEE International Conference on Neural Networks*, Vol. 2, pp. 724-729.

Rajakarunakaran, S.; Venkumar, P.; Devaraj, D. & Rao, K.S.P. (2008). Artificial neural network approach for fault detection in rotary system, Applied Soft Computing, Vol. 8, No. 1, pp.740–748.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1976). Learning internal representations by error propagation. In Rumelhart, D. E. and McClelland, J. L., editors, Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations, MIT Press, Cambridge, MA. pp 318-362.

Sepasi, M. & Sassani, F. (2010). On-line Fault Diagnosis of Hydraulic Systems Using Unscented Kalman Fitler, *International Journal of Control and Automation Systems,* Vol. 8, No. 1, pp. 149-156.

Shi, L. & Sepehri, N. (2004). Adaptive Fuzzy-Neural-Based Multiple Models for Fault Diagnosis of a Pneumatic Actuator, *Proceeding of the 2004 American Control Conference*, Boston. Massachusetts, June 30 -July 2, 2004.

Song, S. O.; Lee, G. & Yoon, E.N. (2004). Process Monitoring of an Electro-Pneumatic Valve Actuator Using Kernel Principal Component Analysis, *Proceedings of the International Symposium on Advanced Control of Chemical Processes*. Hong Kong, China, 11-14 January 2004.

Taghizadeh, M.; Ghaffari, A. & Najafi, F. (2009). Improving dynamic performances of PWM-driven servo-pneumatic systems via a novel pneumatic circuit, *ISA Transactions,* Vol.48, pp. 512-518.

Takosoglu, J. E.; Dindorf, R. F. & Laski, P. A. (2009). Rapid prototyping of fuzzy controller pneumatic servo-system. International Journal of Advanced Manufacturing Technology, Vol. 40, No.3/4, pp. 349-361.

Tansel, I. N. ; Demetgul, M. ; Leon, R. A. ; Yenilmez, A. & Yapici A. (2009). Design of Energy Scavengers of Structural Health Monitoring Systems Using Genetically Optimized Neural Network Systems, Sensors and Materials, Vol. 21, No. 3, pp. 141-153.

Tansel I. N.; Demetgul M. & Sierakowski R. L. (2009). *Detemining Initial Design Parameters By Using Genetically Optimized Neural Network Systems. Sapuan SM, Selangor S, Mujtaba IM(ed) Composite Materials Technology*: Neural Network Applications. Taylor & Francis(CRC Press).

Tsai Y. C. & Huang, A. (2008). Multiple-surface sliding controller design for pneumatic servo systems, *Mechatronics,* Vol. 18, pp. 506–512.

Uppal, F.J.; Patton, R.J. & Palade, V. (2002). Neuro-fuzzy based fault diagnosis applied to an electro-pneumatic valve, *Proceedings of the 15th IFAC World Congress*, Barcelona, Spain, 2002, pp. 2483-2488.

Uppal, F. J.; & Patton, R. J. (2002). Fault Diagnosis of an Electro-pneumatic Valve Actuator Using Neural Networks With Fuzzy Capabilities, *European Symposium on Artificial Neural Networks*, Bruges, Belgium, 24-26 April 2002, pp. 501-506.

Wang, J. et al. (2004). Identification of pneumatic cylinder friction parameters using genetic algorithms. *IEEE/ASME Transactions on Mechatronics*, Vol. 9, No.1, pp. 100-104.

Yang, B. S.; Han, T. & An J. L. (2004). ART–KOHONEN neural network for fault diagnosis of rotating machinery, Mechanical Systems and Signal Processing, Vol. 18, pp. 645–657.

Yang, W.X. (2006). Establishment of the mathematical model for diagnosing the engine valve faults by genetic programming. *Journal of Sound and Vibration,* Vol. 293, pp.213- 226.

22

Neural Networks' Based Inverse Kinematics Solution for Serial Robot Manipulators Passing Through Singularities

Ali T. Hasan[1], Hayder M.A.A. Al-Assadi[2] and Ahmad Azlan Mat Isa[2]
[1]*Department of Mechanical and Manufacturing Engineering*
University Putra Malaysia, 43400UPM,Serdang, Selangor,
[2]*Faculty of Mechanical Engineering, University Technology MARA (UiTM)*
40450 Shah Alam,
Malaysia

1. Introduction

Before moving a robot arm, it is of considerable interest to know whether there are any obstacles present in its path. Computer-based robots are usually served in the joint space, whereas objects to be manipulated are usually expressed in the Cartesian space because it is easier to visualize the correct end-effector position in Cartesian coordinates than in joint coordinates. In order to control the position of the end-effector of the robot, an *inverse kinematics (IK)* solution routine should be called upon to make the necessary conversion (Fu et al., 1987).

Solving the inverse kinematics problem for serial robot manipulators is a difficult task; the complexity in the solution arises from the nonlinear equations (trigonometric equations) occurring during transformation between joint and Cartesian spaces. The common approach for solving the IK problem is to obtain an analytical close-form solution to the inverse transformation, unfortunately, close-form solution can only be found for robots of simple kinematics structure. For robots whose kinematics structures are not solvable in close-form; several numerical techniques have been proposed; however, there still remains several problems in those techniques such as incorrect initial estimation, convergence to the correct solution can not be guarantied, multiple solutions may exist and no solution could be found if the Jacobian matrix was in singular configuration (Kuroe et al., 1994; Bingual et al., 2005). A velocity singular configuration is a configuration in which a robot manipulator has lost at least one motion degree of freedom DOF. In such configuration, the inverse Jacobian will not exist, and the joint velocities of the manipulator will become unacceptably large that often exceed the physical limits of the joint actuators. Therefore, to analyze the singular conditions of a manipulator and develop effective algorithm to resolve the inverse kinematics problem in the singular configurations is of great importance (Hu et al., 2002).

Many research efforts have been devoted towards solving this problem, one of the first algorithms employed was the Resolved Motion Rate-Control method (Whitney, 1969), which uses the pseudoinverse of the Jacobian matrix to obtain the joint velocities corresponding to a given end-effector velocity, an important drawback of this method was

the singularity problem. To overcome the problem of kinematics singularities, the use of a damped least squares inverse of the Jacobian matrix has been later proposed in lieu of the pseudoinverse (Nakamura & Hanafusa, 1986; Wampler, 1986).
Since in the above algorithmic methods the joint angles are obtained by numerical integration of the joint velocities, these and other related techniques suffer from errors due to both long-term numerical integration drift and incorrect initial joint angles. To alleviate the difficulty, algorithms based on the feedback error correction are introduced (Wampler & Leifer, 1988). However, it is assumed that the exact model of manipulator Jacobian matrix of the mapping from joint coordinate to Cartesian coordinate is exactly known. It is also not sure to what extent the uncertainty could be allowed. Therefore, most research on robot control has assumed that the exact kinematics and Jacobian matrix of the manipulator from joint space to Cartesian space are known. This assumption leads to several open problems in the development of robot control laws (Antonelli et al., 2003).
Intelligent control has been introduced as a new direction making control systems able to attribute more intelligence and high degree of autonomy. Artificial Neural Networks (ANN) have been widely used for their extreme flexibility due to the learning ability and the capability of non linear function approximation, a number of realistic control approaches have been proposed and justified for applications to robotic systems (D'Souza et al.,2001; Ogawa et al., 2005; Köker, 2005; Hasan et al., 2007; Al-Assadi et al., 2007), this fact leads to expect ANN to be an excellent tool for solving the IK problem for serial manipulators overcoming the problems arising.
Studying the IK of a serial manipulator by using ANNs has two problems, one of these is the selection of the appropriate type of network and the other is the generating of suitable training data set (Funahashi, 1998;Hasan et al., 2007).
Different methods for gathering training data have been used by many researchers, while some of them have used the kinematics equations (Karilk & Aydin, 2000; Bingual et al., 2005), others have used the network inversion method (Kuroe et al., 1994; Köker, 2005), another have used the cubic trajectory planning (Köker et al., 2004) and others have used a simulation program for this purpose (Driscoll, 2000). However, there are always kinematics uncertainties presence in the real world such as ill-defined linkage parameters, links flexibility and backlashes in gear train.
A learning method of a neural network has been proposed by (Kuroe et al., 1994), such that the network represents the relations of both the positions and velocities from the Cartesian coordinate to the joint space coordinate. They've driven a learning algorithm for arbitrary connected recurrent networks by introducing adjoin neural networks for the original neural networks (Network inversion method). On-line training has been performed for a 2 DOF robot.
(Graca and Gu, 1993) have developed a Fuzzy Learning Control algorithm. Based on the robotic differential motion procedure, the Jacobian inverse has treated as a fuzzy matrix and has learned through the fuzzy regression process. It was significant that the fuzzy learning control algorithm neither requires an exact kinematics model of a robotic manipulator, nor a fuzzy inference engine as is typically done in conventional fuzzy control. Despite the fact that unlike most learning control algorithms, multiple trials are not necessary for the robot to "learn" the desired trajectory. A major drawback was that it only remembers the most recent data points introduced, the researchers have recommended neural networks so that it would remember the trajectories as it traversed them.

The solution of the Inverse kinematics, which is mainly solved in this chapter, involves the development of two network's configurations to examine the effect of the Jacobian Matrix in the solution.
Although this is very difficult in practice (Hornik, 1991), training data were recorded experimentally from sensors fixed on each joint (as was recommended by (Karilk and Aydin, 2000). Finally the obtained results were verified experimentally.

2. Kinematics of serial robots

For serial robot manipulators, the vector of Cartesian space coordinates x is related to the joint coordinates q by:

$$x = f(q) \tag{1}$$

Where $f(\cdot)$ is a non-linear differential function.
If the Cartesian coordinates x were given, joint coordinates q can be obtained as:

$$q = f^{-1}(x) \tag{2}$$

Using ANN to solve relation (2), for getting joint position q, mapping from the joint space to the Cartesian space is uniquely decided when the end effector's position is calculated using direct kinematics (Köker et al., 2004; Ogawa et al., 2005; Hasan et al., 2006), as shown in Figure 1(a). However, the transformation from the Cartesian to the joint space is not uniquely decided in the inverse kinematics as shown in Figure 1(b).

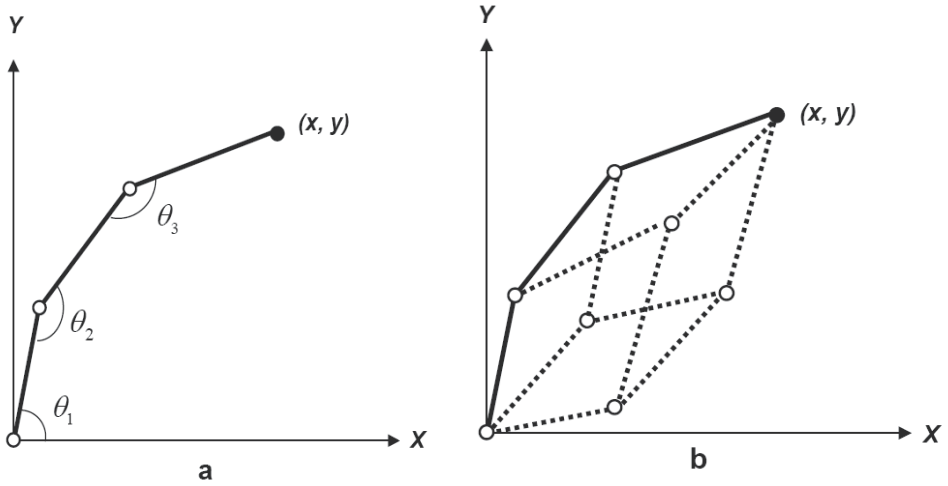


Fig. 1. Three DOF robot arm.
a) Joint angles and end-effector's coordinates (forward kinematics).
b) Combination of all possible joint angles (Inverse Kinematics).

Model-based methods for solving the IK problem are inadequate if the structure of the robot is complex, therefore; techniques mainly based on inversion of the mapping established between the joint space and the task space of the manipulator's Jacobian matrix have been proposed for those structures that cannot be solved in closed form.

If a Cartesian linear velocity is denoted by V, the joint velocity vector $\dot{q}$ has the following relation:

$$V = J\dot{q} \tag{3}$$

Where J is the Jacobian matrix.
If V, is a desired Cartesian velocity vector which represents the linear velocity of the desired trajectory to be followed. Then, the joint velocity vector $\dot{q}$ can be resolved by:

$$\dot{q} = J^{-1}V \tag{4}$$

At certain manipulator configurations, the Jacobian matrix may lose its full rank. Hence as the manipulator approaches these configurations (singular configurations), the Jacobian matrix becomes ill conditioned and may not be invertible. Under such a condition, numerical solution for equation (4) results in infinite joint rates.
In differential motion control, the desired trajectory is subdivided into sampling points separated by a time interval Δt between two terminal points of the path. Assuming that at time t_i the joint positions take on the value $q(t_i)$, the required q at time $(t_i + \Delta t)$ is conventionally updated by using:

$$q(t_i + \Delta t) = q(t_i) + \dot{q}\Delta t \tag{5}$$

Substituting Eqns. (2) and (4) into (5) yields:

$$q(t_i + \Delta t) = f^{-1}(x)(t_i) + J^{-1}V\Delta t \tag{6}$$

Equation (6) is a kinematics control law used to update the joint position q and is evaluated on each sampling interval. The resulting $q(t_i + \Delta t)$ is then sent to the individual joint motor servo-controllers, each of which will independently drive the motor so that the manipulator can be maneuvered to follow the desired trajectory (Graca & Gu, 1993).

3. Data collection procedure

Trajectory planning was performed to generate the angular position and velocity for each joint, and then these generated data were fed to the robot's controller to generate the corresponding Cartesian position and linear velocity of the end-effector, which were recorded experimentally from sensors fixed on the robot joints.
In details, trajectory planning was performed using cubic trajectory planning method .In trajectory planning of a manipulator, it is interested in getting the robot from an initial position to a target position with free of obstacles path. Cubic trajectory planning method has been used in order to find a function for each joint between the initial position, θ_0, and final position, θ_f of each joint.
It is necessary to have at least four-limit value on the $\theta(t)$ function that belongs to each joint, where $\theta(t)$ denotes the angular position at time t.
Two limit values of the function are the initial and final position of the joint, where:

$$\theta(0) = \theta_0 \tag{7}$$

$$\theta(t_f) = \theta_f \tag{8}$$

Additional two limit values, the angular velocity will be zero at the beginning and the target position of the joint, where:

$$\dot{\theta}(0) = 0 \tag{9}$$

$$\dot{\theta}(t_f) = 0 \tag{10}$$

Based on the constrains of typical joint trajectory listed above, a third order polynomial function can be used to satisfy these four conditions; since a cubic polynomial has four coefficients.
These conditions can determine the cubic path, where a cubic trajectory equation can be written as:

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{11}$$

The angular velocity and acceleration can be found by differentiation, as follows:

$$\dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2 \tag{12}$$

$$\ddot{\theta}(t) = 2a_2 + 6a_3 t \tag{13}$$

Substituting the constrains conditions in the above equations results in four equations with four unknowns:

$$\begin{aligned} \theta_0 &= a_0, \\ \theta_f &= a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3, \\ 0 &= a_0, \\ 0 &= a_1 + 2a_2 t_f + 3a_3 t_f^2 \end{aligned} \tag{14}$$

The coefficients are found by solving the above equations.

$$\begin{aligned} a_0 &= \theta_0, \\ a_1 &= 0, \\ a_2 &= \frac{3}{t_f^2}(\theta_f - \theta_0), \\ a_3 &= \frac{-2}{t_f^3}(\theta_f - \theta_0) \end{aligned} \tag{15}$$

Angular position and velocity can be calculated by substituting the coefficients driven in Eqn. (15) into the cubic trajectory Eqns. (11) and (12) respectively (Köker et al., 2004), which yield:

$$\theta_i(t) = \theta_{i0} + \frac{3}{t_f^2}(\theta_{if} - \theta_{i0})t^2 - \frac{2}{t_f^3}(\theta_{if} - \theta_{i0})t^3, \tag{16}$$

$$\dot{\theta}_i(t) = \frac{6}{t_f^2}(\theta_{if} - \theta_{i0})t - \frac{6}{t_f^3}(\theta_{if} - \theta_{i0})t^2 \tag{17}$$

$i = 1, 2, ..., n$ Where n is the joint number.

Joint angles generated ranged from amongst all the possible joint angles that do not exceed the physical limits of each joint. Trajectory used for the training process has meant to be random trajectory rather than a common trajectory performed by the robot in order to cover as most space as possible of the robot's working cell.
The interval of 1 second was used between a trajectory segment and another where the final position for one segment is going to be the initial position for the next segment and so on for every joint of the six joints of the robot.
After generating the joint angles and their corresponding angular velocities, these data are fed to the robot controller, which is provided with a sensor system that can detect the angular position and velocity on one hand and the Cartesian position and the linear velocity of the end-effector on the other hand; which are recorded to be used for the networks' training. As these joints are moving simultaneously with each other to complete the trajectory together.

4. Artificial Neural Networks

Artificial neural networks (ANNs) are collections of small individual interconnected processing units. Information is passed between these units along interconnections. An incoming connection has two values associated with it, an input value and a weight. The output of the unit is a function of the summed value. ANNs while implemented on computers are not programmed to perform specific tasks. Instead, they are trained with respect to data sets until they learn the patterns presented to them. Once they are trained, new patterns may be presented to them for prediction or classification (Kalogirou, 2001).
The elementary nerve cell called a neuron, which is the fundamental building block of the biological neural network. Its schematic diagram is shown in Figure 2.
A typical cell has three major regions: the cell body, which is also called the soma, the axon, and the dendrites. Dendrites form a dendritic tree, which is a very fine bush of thin fibbers around the neuron's body. Dendrites receive information from neurons through axons-Long fibbers that serve as transmission lines. An axon is a long cylindrical connection that carries impulses from the neuron. The end part of an axon splits into a fine arborization. Each branch of it terminates in a small end bulb almost touching the dendrites of neighbouring neurons. The axon-dendrite contact organ is called a *synapse.* The synapse is where the neuron introduces its signal to the neighbouring neuron (Zurada, 1992; Hasan et al., 2006), to stimulate some important aspects of the real biological neuron. An ANN is a group of interconnected artificial neurons usually referred to as "node" interacting with one another in a concerted manner; Figure 3 illustrates how information is processed through a single

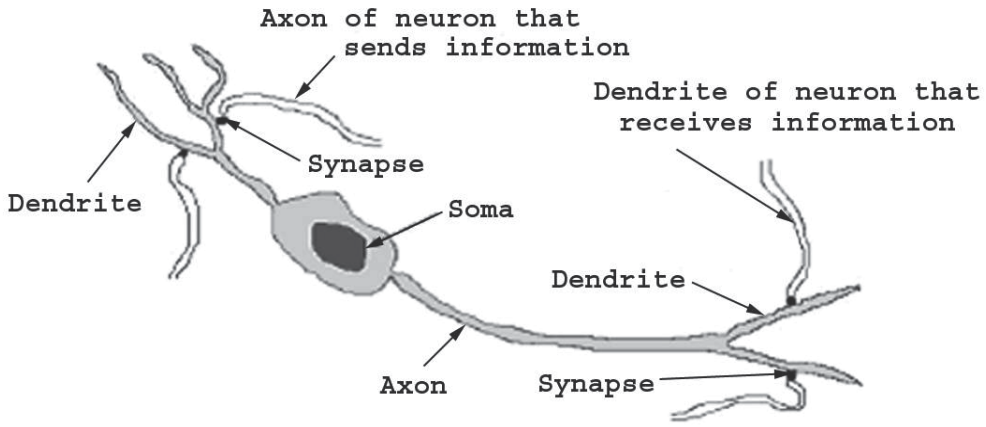


Fig. 2. Schematic diagram for the biological neuron

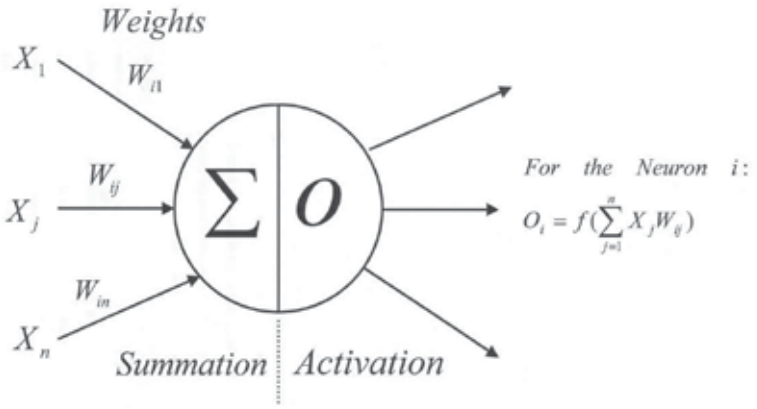


Fig. 3. Information processing in the neural unit

node. The node receives weighted activation of other nodes through its incoming connections. First, these are added up (summation). The result is then passed through an activation function and the outcome is the activation of the node. The activation function can be a threshold function that passes information only if the combined activity level reaches a certain value, or it could be a continues function of the combined input, the most common to use is a sigmoid function for this purpose. For each of the outgoing connections, this activation value is multiplied by the specific weight and transferred to the next node (Kalogirou, 2001; Hasan et al., 2006).

An artificial neural network consists of many nods joined together usually organized in groups called 'layers', a typical network consists of a sequence of layers with full or random connections between successive layers as Figure 4 shows. There are typically two layers with connection to the outside world; an input buffer where data is presented to the network, and an output buffer which holds the response of the network to a given input pattern, layers distinct from the input and output buffers called 'hidden layer', in principle there could be more than one hidden layer, In such a system, excitation is applied to the input layer of the network.

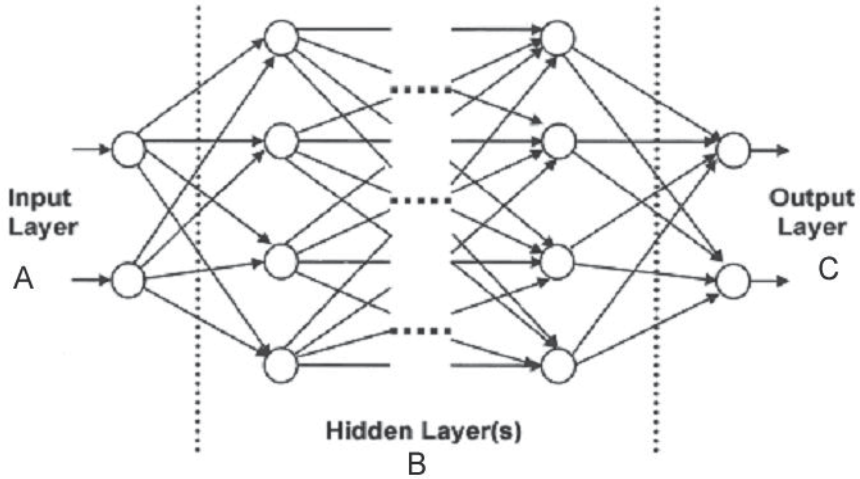


Fig. 4. Schematic diagram of a multilayer feedforward neural network

Following some suitable operation, it results in a desired output. Knowledge is usually stored as a set of connecting weights (presumably corresponding to synapse efficiency in biological neural system) (Santosh et al., 1993). A neural network is a massively parallel-distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the human brain in two respects; the knowledge is acquired by the network through a learning process, and interneuron connection strengths known as synaptic weights are used to store the knowledge (Haykin, 1994).

Training is the process of modifying the connection weights in some orderly fashion using a suitable learning method. The network uses a learning mode, in which an input is presented to the network along with the desired output and the weights are adjusted so that the network attempts to produce the desired output. Weights after training contain meaningful information whereas before training they are random and have no meaning (Kalogirou, 2001).

Two different types of learning can be distinguished: supervised and unsupervised learning, in supervised learning it is assumed that at each instant of time when the input is applied, the desired response d of the system is provided by the teacher. This is illustrated in Figure 5-a. The distance ρ [d,o] between the actual and the desired response serves as an error measure and is used to correct network parameters externally. Since adjustable weights are assumed, the teacher may implement a reward-and-punishment scheme to adopt the network's weight. For instance, in learning classifications of input patterns or situations with known responses, the error can be used to modify weights so that the error decreases. This mode of learning is very pervasive.

Also, it is used in many situations of learning. A set of input and output patterns called a training set is required for this learning mode. Figure 5-b shows the block diagram of unsupervised learning. In unsupervised learning, the desired response is not known; thus, explicit error information cannot be used to improve network's behaviour. Since no information is available as to correctness or incorrectness of responses, learning must somehow be accomplished based on observations of responses to inputs that we have marginal or no knowledge about (Zurada, 1992).

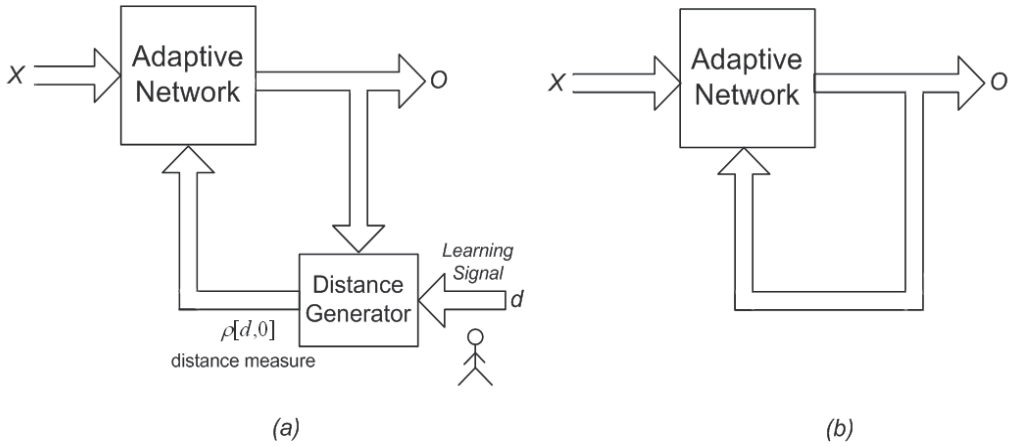


Fig. 5. Basic learning modes

The fundamental idea underlying the design of a network is that the information entering the input layer is mapped as an internal representation in the units of the hidden layer(s) and the outputs are generated by this internal representation rather than by the input vector. Given that there are enough hidden neurons, input vectors can always be encoded in a form so that the appropriate output vector can be generated from any input vector (Santosh et al., 1993).

As it can be seen in figure 4, the output of the units in layer ***A*** (Input Layer) are multiplied by appropriate weights W_{ij} and these are fed as inputs to the hidden layer. Hence if O_i are the output of units in layer ***A,*** then the total input to the hidden layer, i.e., layer ***B*** is:

$$Sum_B = \sum_i O_i W_{ij} \tag{18}$$

And the output O_j of a unit in layer ***B*** is:

$$O_j = f(sum_B) \tag{19}$$

Where *f* is the non-linear activation function, it is a common practice to choose the sigmoid function given by:

$$f(O_j) = \frac{1}{1+e^{-O_j}} \tag{20}$$

As the nonlinear activation function.

However, any input-output function that possesses a bounded derivative can be used in place of the sigmoid function. If there is a fixed, finite set of input-output pairs, the total error in the performance of the network with a particular set of weights can be computed by comparing the actual and the desired output vectors for each presentation of an input vector. The error at any output unit e_K in the layer **C** can be calculated by: -

$$e_K = d_K - O_K \tag{21}$$

Where d_K is the desired output for that unit in layer **C** and O_K is the actual output produced by the network .the total error ***E*** at the output can be calculated by:

$$E = \frac{1}{2}\sum_{K}(d_K - O_K)^2 \tag{22}$$

Learning comprises changing weights so as to minimize the error function and to minimize ***E*** by the gradient descent method. It is necessary to compute the partial derivative of ***E*** with respect to each weight in the network. Equations (19) and (19) describe the forward pass through the network where units in each layer have there states determined by the inputs they received from units of lower layer. The backward pass through the network that involves "back propagation " of weight error derivatives from the output layer back to the input layer is more complicated. For the sigmoid activation function given in Equation (20), the so-called delta-rule for iterative convergence towards a solution maybe stated in general as:

$$\Delta W_{JK} = \eta \delta_K O_J \tag{23}$$

Where η is the learning rate parameter, and the error δ_K at an output layer unit K is given by:

$$\delta_K = O_K(1 - O_K)(d_K - O_K) \tag{24}$$

And the error δ_J at a hidden layer unit is given by:

$$\delta_J = O_J(1 - O_J)\sum_{K}\delta_K W_{JK} \tag{25}$$

Using the generalize delta rule to adjust weights leading to the hidden units is back propagating the error-adjustment, which allows for adjustment of weights leading to the hidden layer neurons in addition to the usual adjustments to the weights leading to the output layer neurons. A back propagation network trains with two step procedures as it is

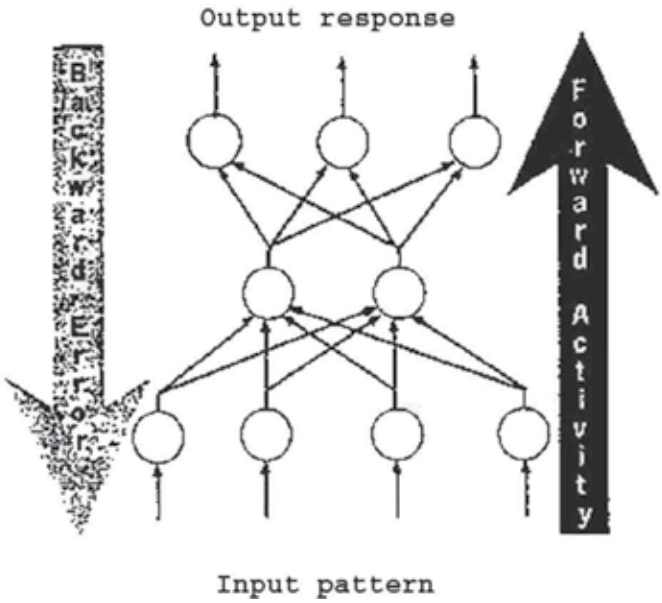


Fig. 6. Information flow through a backpropagation network

shown in figure 6, the activity from the input pattern flows forward through the network and the error signal flows backwards to adjust the weights using the following equations:

$$W_{IJ} = W_{IJ} + \eta \delta_J O_I \tag{26}$$

$$W_{JK} = W_{JK} + \eta \delta_K O_J \tag{27}$$

Until for each input vector the output vector produced by the network is the same as (or sufficiently close to) the desired output vector (Santosh et al., 1993).
ANNs while implemented on computers are not programmed to perform specific tasks. Instead, they are trained with respect to data sets until they learn the patterns presented to them. Once they are trained, new patterns may be presented to them for prediction or classification (Kalogirou, 2001).

5. ANN implementation

Two supervised feedforward ANN have been designed using C programming language to overcome the singularities and uncertainties in the arm configurations. Both networks consist of input, output and one hidden layer, every neuron in each of the networks is fully connected with each other, sigmoid transfer function was chosen to be the activation function, generalized backpropagation delta learning rule (GDR) algorithm was used in the training process.
Off-line training was implemented; Trajectory planning was performed for 600 data set for every 1-second interval from amongst all the possible joint angles in the robot's workspace, then data sets were recorded experimentally from sensors fixed on the robot joints as was recommended by (Karilk and Aydin, 2000), 400 set were used for the training while the other 200 sets were used for the testing the network.
All input and output values are usually scaled individually such that overall variance in data set is maximized, this is necessary as it leads to faster learning, all the vectors were scaled to reflect continuous values ranges from –1 to 1.
FANUC M-710i robot was used in this study, which is a serial robot manipulator consisting of axes and arms driven by servomotors. The place at which arm is connected is a joint, or an axis. This type of robot has three main axes; the basic configuration of the robot depends on whether each main axis functions as a linear axis or rotation axis. The wrist axes are used to move an end effecter (tool) mounted on the wrist flange. The wrist itself can be wagged about one wrist axis and the end effecter rotated about the other wrist axis, this highly non-linear structure makes this robot very useful in typical industrial applications such as the material handling, assembly of parts and painting.
The networks' implementation carried out on two phases, the first was the training phase where the performance of the two networks were compared, and then the network that has shown better response has been chosen to apply the testing data during the testing phase, which has been implemented through two stages. The first stage was the simulation then the results were verified experimentally.

5.1 Training phase

To examine the effect of considering the Jacobian Matrix for the Inverse Kinematics solution two networks have been designed and compared. ANN technique has been utilized where

learning is only based on observation of the input output relationship unlike other schemes that require an explicit system model.

5.1.1 The first configuration (3 - 6 configuration)

As in our previous research (Hasan et al., 2006; Hasan et al., 2007), the input vector for the network consists of the position of the end effector of the robot along the X, Y and Z coordinates of the global coordinate system, while the output vector was the angular position of each of the 6 joints as can be seen in Figure 7.

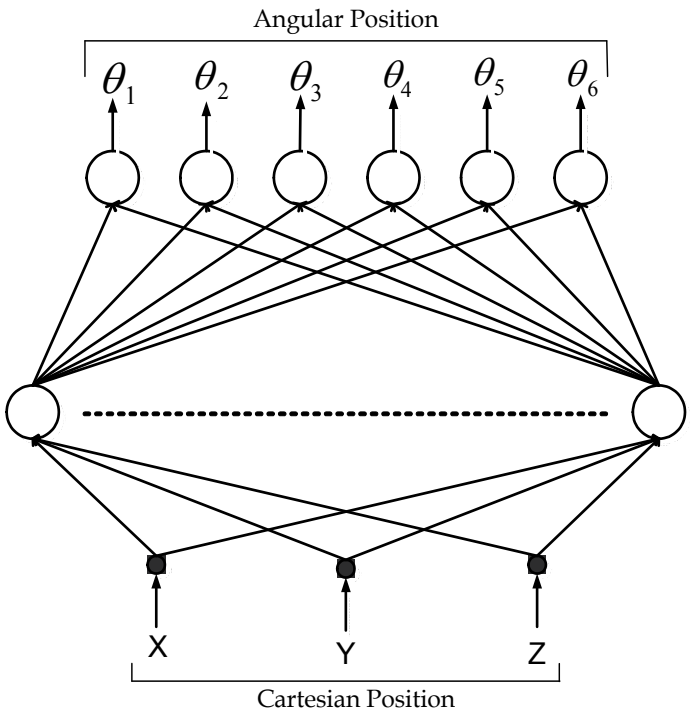


Fig. 7. The topology of the first configuration network

Although the number of training patterns was doubled, number of neurons in the hidden layer still the same as previous which is 43 only a little difference in the learning factor was experienced which was 0.5 in this case by trial and error.
Figure 8 shows the building knowledge curve for this configuration while Table 1 shows the percentage of error of each of the 6 joints after the training was finished after 1.5 million iterations.

Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
13.193%	10.855%	3.478%	14.748%	12.243%	8.348%

Table 1. Error percentages obtained after training (first configuration)

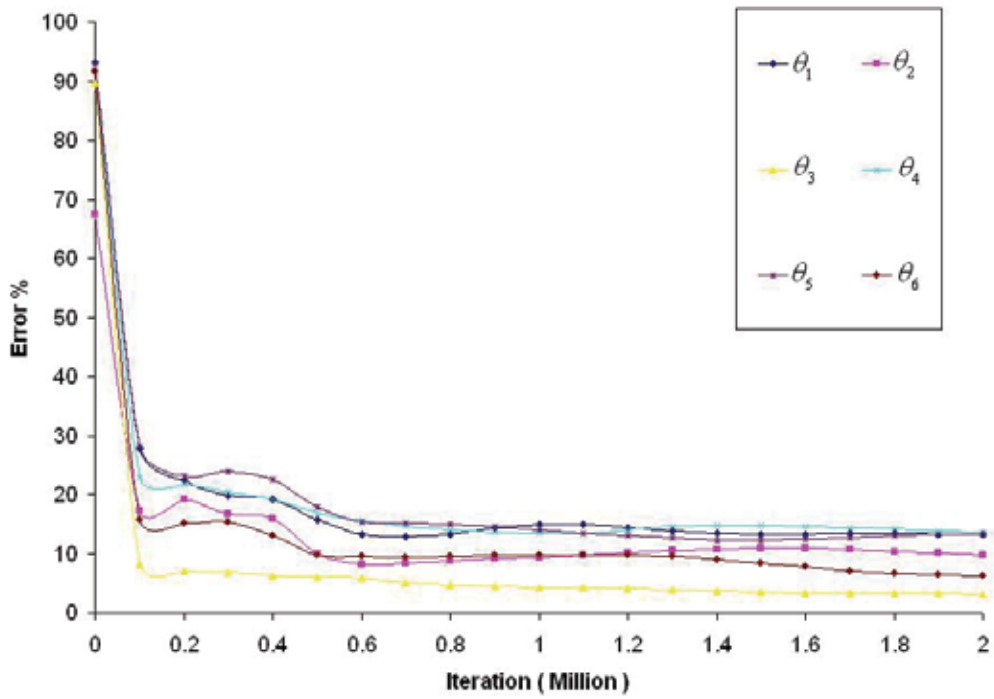


Fig. 8. The learning curve for the first configuration

5.1.2 The second configuration (4 – 12 configuration)

To examine the effect of considering the Jacobian matrix to the IK solution, another network has been designed, as in Figure 9, the new network consists of the Cartesian Velocity added to the input buffer and the angular velocity of each of the 6 joints added to the output buffer of the previous network.

Number of the neurons in the hidden layer was set to be 77 with constant learning factor of 0.9 by trial and error.

Figure 10 shows the building knowledge curve while table 2 shows the percentage of error of each of the 6 joints after the training was finished after 1.5 million iterations.

	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
Angular Position	2.817 %	1.645%	0.88%	3.163%	3.125%	2.09%
Angular Velocity	2.65%	3.018%	1.683%	3.208%	2.525%	1.6%

Table 2. Error percentages obtained after training (second configuration)

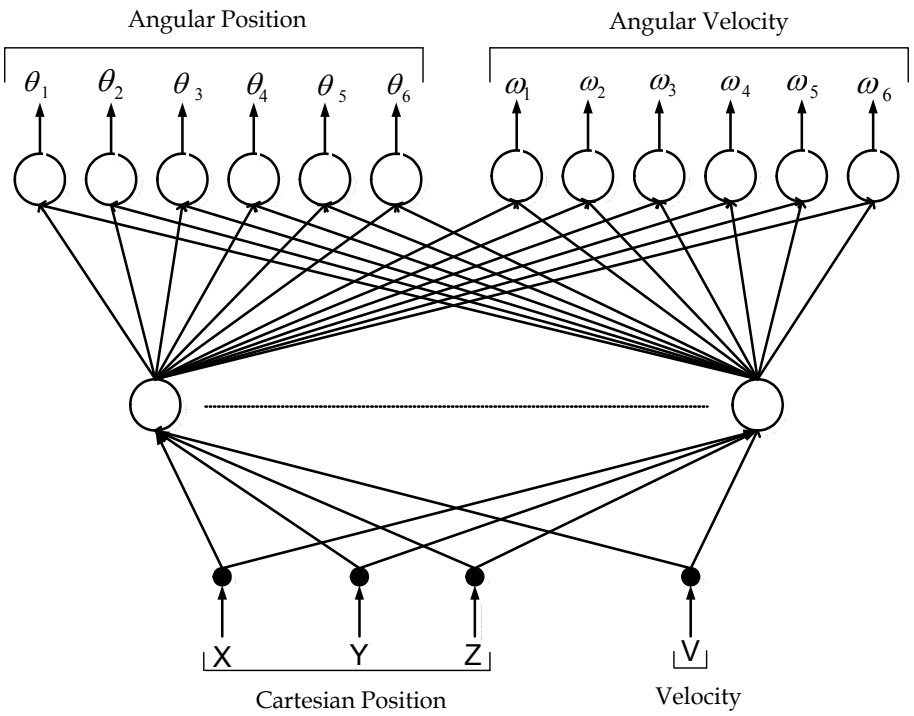


Fig. 9. The topology of the second configuration network

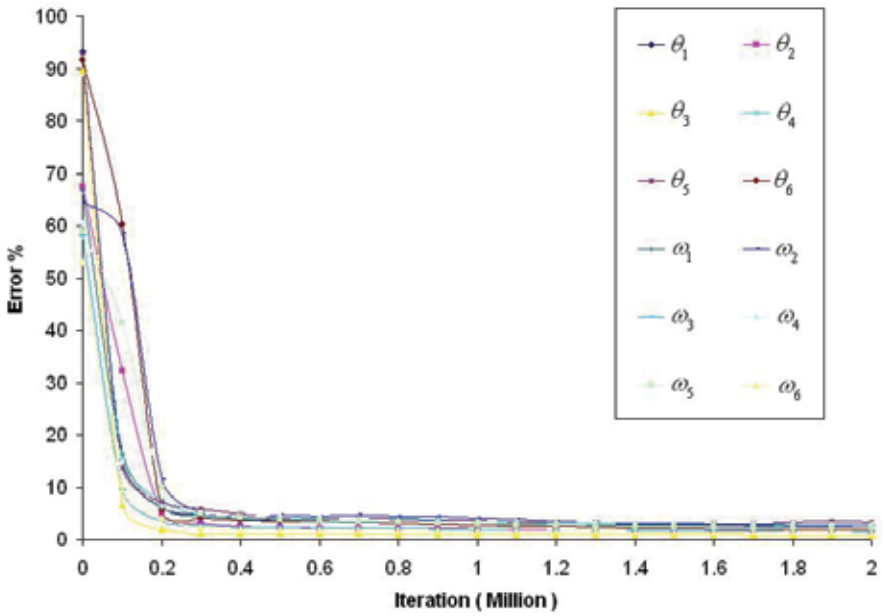


Fig. 10. The learning curve for the second configuration

5.1.3 Networks' performance

The performance of the two networks was measured as the difference between desired and actual system output.

To drive the robot to follow a desired trajectory, it will be necessary to divide the path into small portions, and to move the robot through all intermediate points. To accomplish this task, at each intermediate location, the robot's IK equations are solved, a set of joint variables is calculated, and the controller is directed to drive the robot to the next segment, when all segments are completed, the robot would be at the end point as desired.

Figures 11 to 13 show the experimental trajectory tracking for the robot over the X, Y and Z Coordinates of the global coordinates system for both of the networks compared to each other verses the desired trajectory.

As can be seen through these Figures, the performance of the first network has improved when considering the Jacobian Matrix in the second network, in terms of precision and iteration

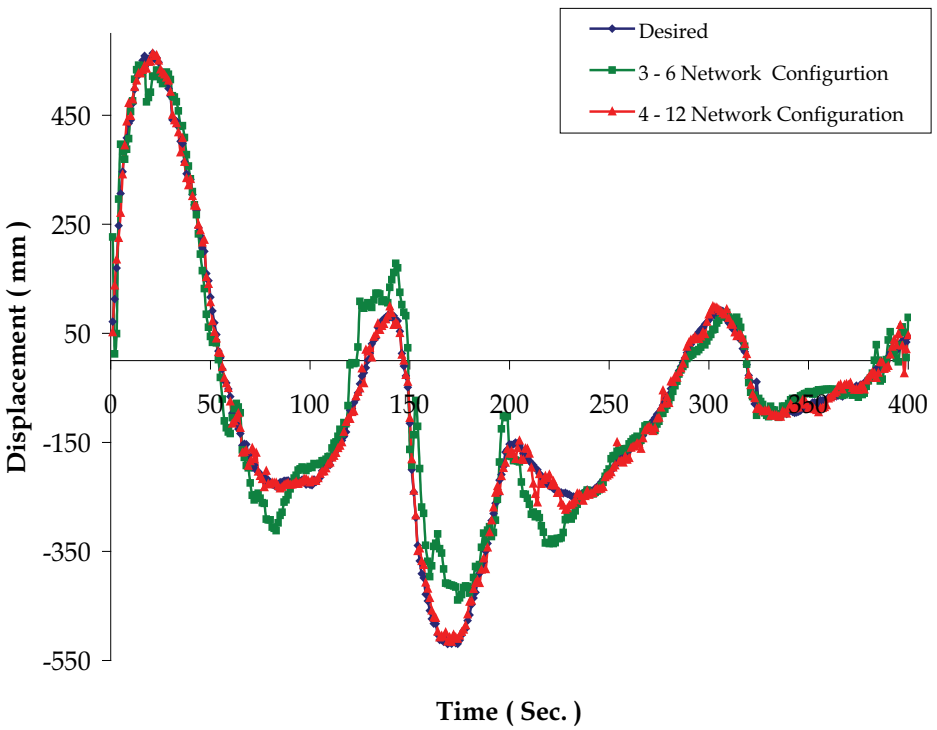


Fig. 11. Trajectory tracking for both configurations compared to each other after the training was finished for the X coordinate

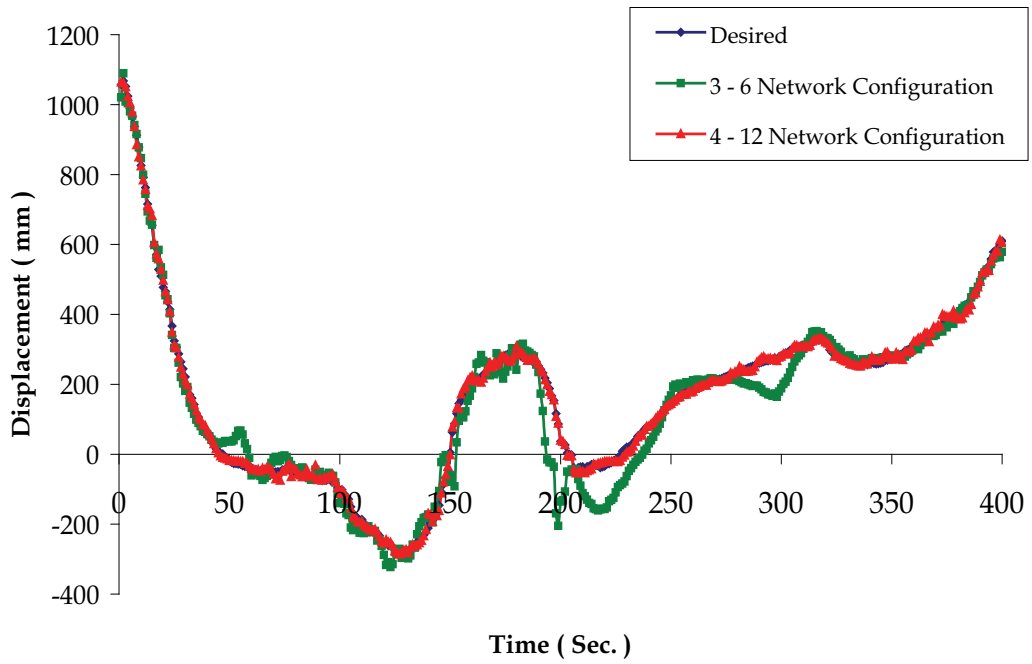


Fig. 12. Trajectory tracking for both configurations compared to each other after the training was finished for the Y coordinate

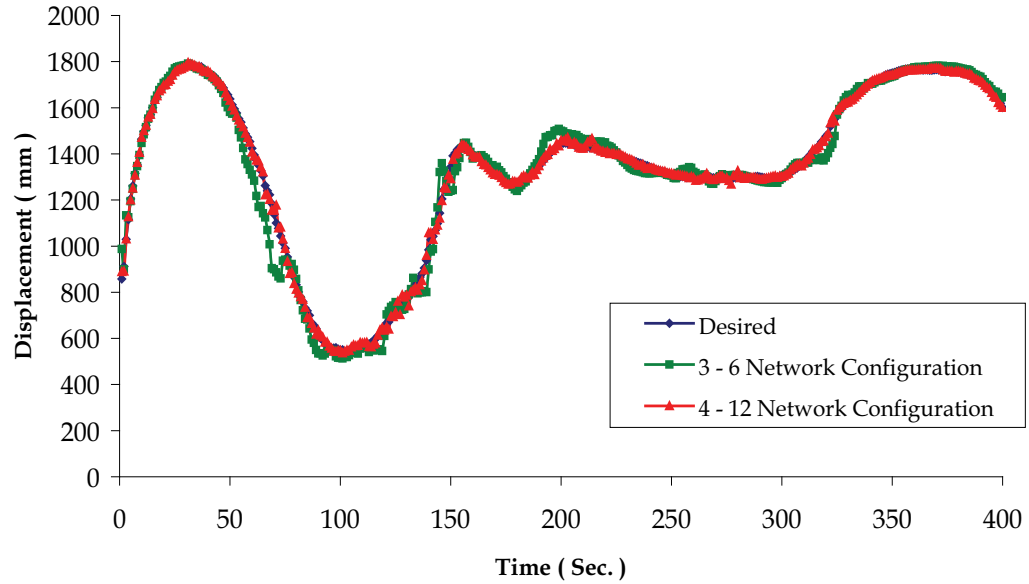


Fig. 12. Trajectory tracking for both configurations compared to each other after the training was finished for the Z coordinate

5.2 Testing phase

New data that has never been introduced to the network before have been fed to the second configuration network to test its ability to make prediction and generalization to any set of data later (as it has shown better response than the first configuration network).

Testing data were meant to pass through singular configurations (fourth and fifth joints); these configurations have been determined by setting the determinant of the Jacobian matrix to zero.

Table 3 shows the percentages of error for the testing data set for each joint during simulation stage.

	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
Angular Position	1.8%	0.245%	0.2%	5.32%	9.885%	0.08%
Angular Velocity	3.82%	2.875%	1.47%	2.64%	3.28%	1.41%

Table 3. Error percentages obtained for testing data through simulation stage

In order to verify the testing results during simulation stage, experiment has been performed to make sure that the output is the same or sufficiently close to the desired trajectory, and to show the combined effect of error, Figures 13, 14 and 15 show the predicted trajectory tracking of the X, Y, and Z coordinates respectively. Locus of which robot is passing through singular configurations are also shown.

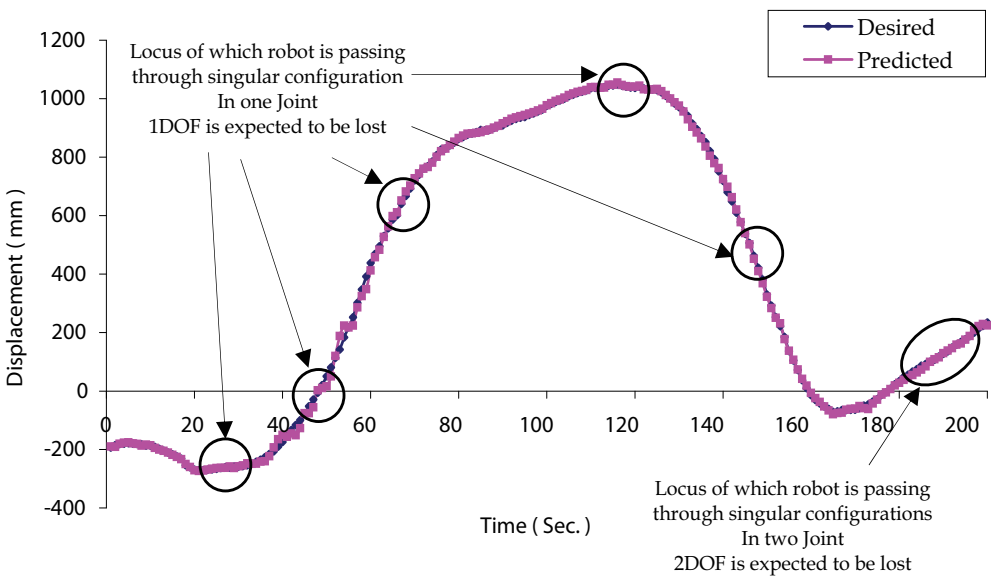


Fig. 13. Predicted trajectory for the X coordinate

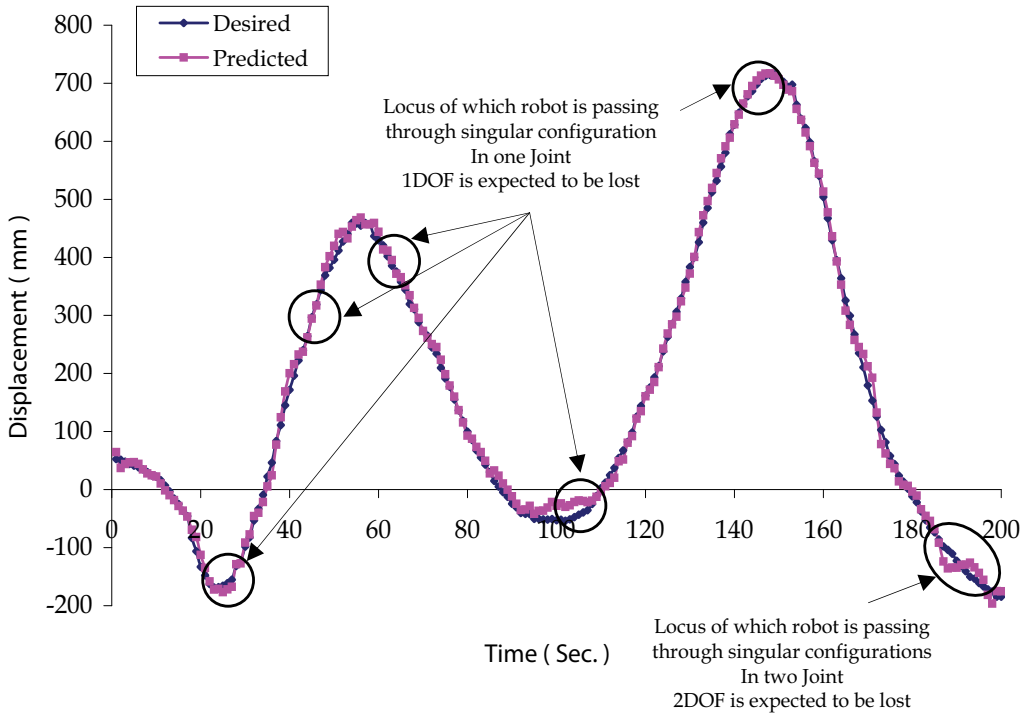


Fig. 14. Predicted trajectory for the Y coordinate

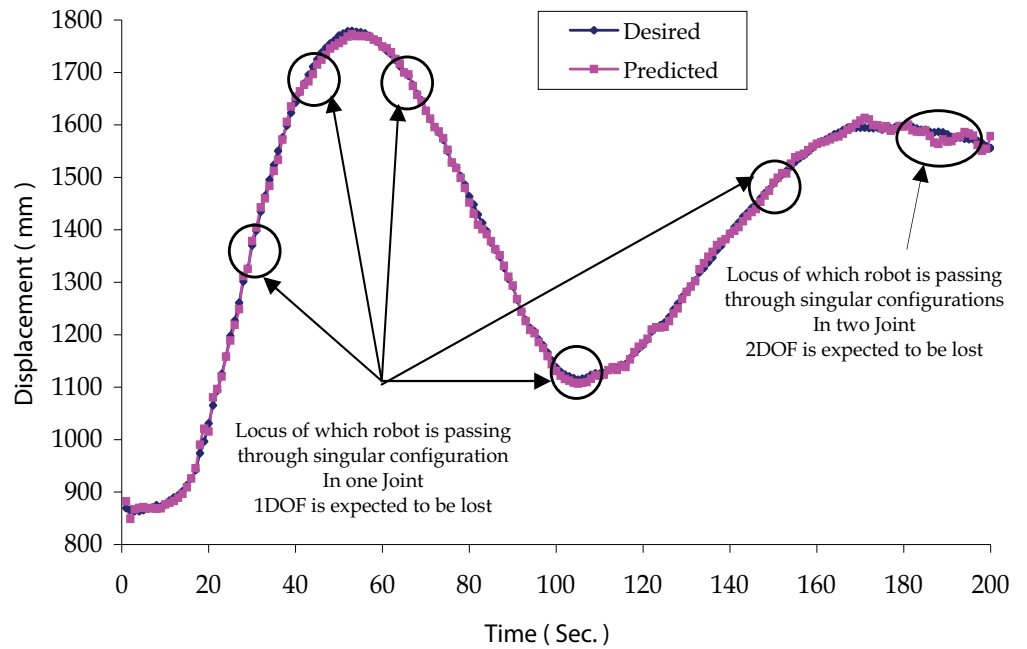


Fig. 15. Predicted trajectory for the Z coordinate

The error percentages in the experimental data are shown in table 4.

X	Y	Z
6.444%	16.355%	0.462%

Table 4. Error percentages obtained for testing data through experimental stage

6. Conclusions and recommendations

In order to overcome the drawbacks of some control schemes which depends on modeling the system being controlled, ANN technique has been utilized where learning is done iteratively based only on observation of input-output relationship unlike most other control schemes, which is a significant advantage of using ANN technology.

In the first network, although the number of hidden neurons was the same with the previous research despite the fact that the number of training patterns was doubled, an important remark is that the error percentage was higher than it was in the previous research which leads to a conclusion that this network configuration does not have the ability to learn huge number of patterns and its use will be limited to small number of data patterns.

As this research has shown, the consideration of the Jacobian Matrix in the solution of the Inverse Kinematics problem using neural networks gives a better response. As compared to the Fuzzy Learning Control algorithm results, the trained network was able to remember not only the training data but also was able to predict unknown trajectories as well as can be seen through the testing phase which is a significant advantage of using this approach.

Backpropagation algorithm has been used as a learning algorithm with sigmoid transfer function as an activation function in all neurons, we would like to recommend that a different learning algorithm, different activation function and/or different number of hidden layers to be used in order to achieve, if possible, a better response in terms of precision and iteration.

7. References

Al-Assadi, H.M.A.A.; Hamouda, A.M.S.; Ismail, N. & Aris, I. (2007). An adaptive learning algorithm for controlling a two-degree-of-freedom serial ball-and-socket actuator. *Proceedings of the IMechE Part I Journal of Systems & Control Engineering*, Vol.221, No. 7,pp.1001-1006.

Antonelli, G.; Chiaverini, S. & Fusco, G. (2003). A new on-line algorithm for inverse kinematics of robot manipulators ensuring path-tracking capability under joint limits. *IEEE Transaction on Robotics and Automation*, Vol.19, No.1, pp. 162-167.

Bingual, Z.; Ertunc, H.M. & Oysu, C. (2005). Comparison of Inverse Kinematics Solutions Using Neural Network for 6R Robot Manipulator with Offset. *ICSC congress on Computational Intelligence*.

Driscoll, J.A. (2000). Comparison of neural network architectures for the modeling of robot inverse kinematics. *Proceedings of the IEEE*, south astcon, pp. 44-51.

D'Souza, A.; Vijayakumar, S. & Schaal, S. (2001). Learning Inverse Kinematics. *Proceedings of the 2001 IEEE/ RSJ International Conference on Intelligent Robots and Systems*, pp.298-303, Maui, Haw- USA.

Fu, K.S.; Gonzalez, R.C. & Lee, C.S.G. (1987). *Robotics control, Sensing, Vision and intelligence*, McGraw-Hill Book Co, New York.

Funahashi, K.I., 1998. On the approximate realization of continuous mapping by neural networks. *Journal of Neural Networks*, Vol.2, No.3, pp.183-192.

Graca, R.A. & Gu, Y. (1993). A Fuzzy Learning Algorithm for Kinematic Control of a Robotic System. *Proceeding of the 32nd Conference on Decision and Control*, pp.1274-1279, San Antonio, Texas, USA.

Hasan, A.T.; Hamouda, A.M.S.; Ismail, N. & Al-Assadi, H.M.A.A. (2007). A new adaptive learning algorithm for robot manipulator control. *Proceeding of the IMechE, Part I: Journal of System and Control Engineering*, Vol.221, No.4, pp. 663-672.

Hasan, A.T.; Hamouda, A.M.S.; Ismail, N. & Al-Assadi, H.M.A.A.(2006). An adaptive-learning algorithm to solve the inverse kinematics problem of a 6 D.O.F serial robot manipulator. *Journal of Advances in Engineering Software*, Vol.37, pp. 432-438.

Haykin S. (1994). *Neural Networks*. A Comprehensive Foundation. New York: Macmillan.

Hornik, K. (1991). Approximation capabilities of multi-layer feed forward networks. *IEEE Trans. Neural Networks*, Vol.4, No.2, pp. 251-257.

Hu, Z.; FU, Z. & Fang, H. (2002). Study of singularity robust inverse of Jacobian matrix for manipulator, *Proceedings of the First International Conference on Machine Learning and Cybernetics*, pp. 406-410,China, Beijing.

Kalogirou, S.A. (2001) Artificial Neural Networks In Renewable Energy Systems Applications: a review. *Renewable and Sustainable Energy Reviews*. Vol. 5,pp.373-401.

Karilk, B. & Aydin, S. (2000). An improved approach to the solution of inverse kinematics problems for robot manipulators. *Journal of Engineering applications of artificial intelligence*, Vol.13, pp.159-164.

Köker, R. (2005). Reliability-based approach to the inverse kinematics solution of robots using Elman's networks. *Engineering Applications of Artificial Intelligence*, Vol.18, pp. 685-693.

Köker, R.; Öz, C.; Çakar.T. & Ekiz, H. (2004). A study of neural network based inverse kinematics solution for a three-joint robot. *Journal of Robotics and Autonomous Systems*, Vol.49, pp. 227–234.

Kuroe, Y.; Nakai, Y. & Mori, T. (1994). A new Neural Network Learning on Inverse Kinematics of Robot Manipulators, *International Conference on Neural Networks, IEEE world congress on computational Intelligence*. Vol.5, pp. 2819-2824.

Nakamura, Y. & Hanafusa, H. (1986). Inverse kinematic solutions with singularity robustness for robot manipulator control, *Journal of Dynamic Systems Measurements Control*, Vol. 108,pp. 163–171.

Ogawa, T.; Matsuura, H. & Kanada, H. (2005). A Solution of Inverse Kinematics of Robot Arm Using Network Inversion. *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation*.

Santosh, A. ;Devendra P. Garg. (1993). Training back propagation and CMAC neural networks for control of a SCARA robot. *Journal of Engineering Applications of Artificial Intelligence*. Vol.6.No.2. pp.105-115.

Wampler, C. W. & Leifer, L. J. (1988). Applications of damped least-squares methods to resolved-rate and resolved-acceleration control of manipulators. *Journal of Dynamic Systems Measurements Control*, Vol. 110,pp. 31–38.

Wampler, C. W. (1986). Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods, *IEEE Transaction Syst., Man, Cybernetics*. Vol. 16,pp. 93–101.

Whitney. E. (1969). Resolved motion rate control of manipulators and human prostheses. *IEEE Transaction Man–Mach. Systems*, Vol. MMS–10,pp.47–53.

Zurda, M. J. (1992). *Introduction to Artificial Neural System Network*. West Publishing Companies, ISBN 0-314-93397-3, St. Paul, MN, USA.

Edited by Kenji Suzuki

Artificial neural networks may probably be the single most successful technology in the last two decades which has been widely used in a large variety of applications. The purpose of this book is to provide recent advances of artificial neural networks in industrial and control engineering applications. The book begins with a review of applications of artificial neural networks in textile industries. Particular applications in textile industries follow. Parts continue with applications in materials science and industry such as material identification, and estimation of material property and state, food industry such as meat, electric and power industry such as batteries and power systems, mechanical engineering such as engines and machines, and control and robotic engineering such as system control and identification, fault diagnosis systems, and robot manipulation. Thus, this book will be a fundamental source of recent advances and applications of artificial neural networks in industrial and control engineering areas. The target audience includes professors and students in engineering schools, and researchers and engineers in industries.

ISBN 978-953-51-4505-9

Photo by Shutterstock

IntechOpen