



IntechOpen

New Applications of Artificial Intelligence

*Edited by Pedro Ponce,
Arturo Molina Gutierrez and Jaime Rodriguez*



NEW APPLICATIONS OF ARTIFICIAL INTELLIGENCE

Edited by **Pedro Ponce, Arturo Molina
Gutiérrez** and **Jaime Rodríguez**

New Applications of Artificial Intelligence

<http://dx.doi.org/10.5772/61364>

Edited by Pedro Ponce, Arturo Molina Gutierrez and Jaime Rodriguez

Contributors

Hong Qi, Feng Liu, Ping Jiang, Yiliao Song, Mohammad Divandari, Hiram Ponce, Ernesto Moya-Albor, Jorge Brieva, Luis Ibarra, Colin Webb, Gelu-Ovidiu Tirian, Raluca Rob, Caius Panoiu

© The Editor(s) and the Author(s) 2016

The moral rights of the and the author(s) have been asserted.

All rights to the book as a whole are reserved by INTECH. The book as a whole (compilation) cannot be reproduced, distributed or used for commercial or non-commercial purposes without INTECH's written permission.

Enquiries concerning the use of the book should be directed to INTECH rights and permissions department (permissions@intechopen.com).

Violations are liable to prosecution under the governing Copyright Law.



Individual chapters of this publication are distributed under the terms of the Creative Commons Attribution 3.0 Unported License which permits commercial use, distribution and reproduction of the individual chapters, provided the original author(s) and source publication are appropriately acknowledged. If so indicated, certain images may not be included under the Creative Commons license. In such cases users will need to obtain permission from the license holder to reproduce the material. More details and guidelines concerning content reuse and adaptation can be found at <http://www.intechopen.com/copyright-policy.html>.

Notice

Statements and opinions expressed in the chapters are these of the individual contributors and not necessarily those of the editors or publisher. No responsibility is accepted for the accuracy of information contained in the published chapters. The publisher assumes no responsibility for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained in the book.

First published in Croatia, 2016 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

New Applications of Artificial Intelligence

Edited by Pedro Ponce, Arturo Molina Gutierrez and Jaime Rodriguez

p. cm.

Print ISBN 978-953-51-2534-1

Online ISBN 978-953-51-2535-8

eBook (PDF) ISBN 978-953-51-5776-2

We are IntechOpen, the first native scientific publisher of Open Access books

3,450+

Open access books available

110,000+

International authors and editors

115M+

Downloads

151

Countries delivered to

Our authors are among the
Top 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Meet the editors



Pedro Ponce studied engineering in automation and control and graduated in 1995. Subsequently, he completed his graduate studies, obtaining the degree of Master of Science in 1998 and Doctor of Science in 2002. He worked as a field and designer engineer in several industries. He specialized in the areas of industrial automation systems, electrical machines, electric drives, power electronics, conventional and digital control, expert systems, neural networks fuzzy logic, biological artificial systems, and evolutionary systems. He is a professor and a researcher at Tecnológico de Monterrey campus Ciudad de Mexico.



Arturo Molina Gutiérrez is a professor and researcher, as well as Vice President of Research, Postgraduate Studies and Continuing Education, at the Tecnológico de Monterrey. He has a bachelor's degree in computational systems and a master's degree in computational sciences (1990) from the Tec de Monterrey, Campus Monterrey (1986), a doctorate degree in mechanics from the Technical University of Budapest (1992), and another in manufacturing systems from the Loughborough University of Technology in England (1995).



Jaime Rodriguez is a professor and researcher at Instituto Politecnico Nacional, Mexico City. He served as coordinator of postgraduate programs in electrical engineering for 3 years at ESIME-SEPI-IPN. He has received a bachelor's degree in electrical engineering in Cuba and a master's degree in electric drives as well as a PhD degree in control of induction motors and power electronics; he received both degrees in Moscow, Russia. His research areas are focused on electric drives, power electronics, electric cars, and alternative energy. He has conducted research projects for solving industrial applications for the private and government sectors.

Contents

Preface XI

Section 1 Power Electronics and Artificial Intelligence 1

Chapter 1 Intelligent System for Controlling a Three-Phase Active Filter 3

Raluca Rob, Gelu Ovidiu Tirian and Caius Panoiu

Chapter 2 Comparison Study of AI-based Methods in Wind Energy 27

Ping Jiang, Feng Liu and Yiliao Song

Chapter 3 Fuzzy Logic Control of Switched Reluctance Motor Drives 59

M. Divandari and B. Rezaie

Section 2 Novel Applications by Artificial Intelligence 91

Chapter 4 Advantages of Fuzzy Control While Dealing with Complex/Unknown Model Dynamics: A Quadcopter Example 93

Luis Ibarra and Colin Webb

Chapter 5 Retrieval of Optical Constant and Particle Size Distribution of Particulate Media Using the PSO-Based Neural Network Algorithm 123

Hong Qi, Ya-Tao Ren, Jun-You Zhang, Li-Ming Ruan and He-Ping Tan

Chapter 6 A Novel Artificial Organic Controller with Hermite Optical Flow Feedback for Mobile Robot Navigation 145

Hiram Ponce, Ernesto Moya-Albor and Jorge Brieva

Preface

New Applications of Artificial Intelligence is a mandatory book for undergraduate and graduate students, engineers, and researchers because it covers the most relevant applications in several areas based on Artificial Intelligence (AI), so the reader gains knowledge about the new methodologies and process for implementing AI methods. There are a lot of books that address AI applications. However, this book shows the newest applications reached according with the technological changes that are presented nowadays. Those changes drastically appear in digital systems or other parallel areas which allow to improve the performance of AI algorithms. Hence, sometimes, the AI algorithms have to be redesigned in order to run in microcontrollers or FPGAs. The topics covered generate a structured book, so it could be used as a textbook, but it is designed to be accessible to a wide audience interested in AI.

On the other hand, the book has two main sections; one of them is about power electronics, in which there are several new applications of AI, such as three-phase active filter controlled by a LabVIEW based on intelligent systems for power quality. The second section is about novel applications of AI methods in several areas; this topic shows a wide selection of applications and new methods of AI to present a complete view of AI. For instance, one of the chapters shows a novel nature-inspired and intelligent control system for mobile robot navigation using a fuzzy-molecular inference (FMI) system as the control strategy and a single vision-based sensor device, i.e., image acquisition system, as feedback, and they improve the performance of mobile robot navigation.

We edited this book because we were excited about the newest applications of AI as an integrated topic, which consists of several areas. Thus, an application of AI is a set of technological tools that create new possibilities for solving engineering problems, and this book presents a complete set of clear examples that motivate the readers to create their own applications based on the book examples.

Pedro Ponce

Tecnologico de Monterrey
Escuela de Posgrado Ingenieria, Mexico City, Mexico

Arturo Molina Gutiérrez

Tecnologico de Monterrey
Escuela de Posgrado Ingenieria, Mexico City, Mexico

Jaime Rodríguez

Instituto Politecnico Nacional
ESIME, Mexico City, Mexico

Power Electronics and Artificial Intelligence

Intelligent System for Controlling a Three-Phase Active Filter

Raluca Rob, Gelu Ovidiu Tirian and Caius Panoiu

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/63640>

Abstract

This chapter describes the base algorithm and the implementation of a three-phase active filter controlled by a LabVIEW software application. This intelligent system was designed for power quality improvement in functioning of an electrothermal installation with electromagnetic induction. The active filtering device is composed by pulse width modulation (PWM) converter, which processes the power and the controller that is responsible for signal processing. The PWM converter accomplishes the current compensation from the power distribution. The filter controller processes the signal for a real-time determination of the references for instantaneous compensating current. These currents are continuously supplying the PWM converter. The LabVIEW application is also responsible for the electric parameter's computing. The application uses a data acquisition board connected to a computer and an active filter realized with insulated gate bipolar transistors (IGBTs). This work also contains real measurement sets and parameter variation obtained by using the presented soft-controlled device.

Keywords: active filtering, current harmonic compensation, frequency modulation, electrothermal installation, electrical parameters, Labview, data acquisition

1. Introduction

The nonlinear elements contained by electrothermal installation lead to three main negative effects, which are reflected in the power quality: circulation of reactive power, appearance of harmonic currents and unbalance of the power supply system.

The problems that appear in power quality are due to the functioning of static electronic converters, power electronic devices, arc furnaces or fluorescent lamps. The frequency

converters, which come in the electric scheme of the electrothermal installation, develop unfavourable effects into the power system: distortion regime generated by the voltage or current waveform, additional heating developed by the high value of the effective currents, or cable insulation damage. These problems can lead to malfunctioning of other electric devices, which are supplied from the same power distribution and also a quality diminution in electric power delivering [1].

2. Description of the intelligent system for controlling a three-phase active filter

2.1. Electrothermal installation based on electromagnetic induction studied as a generator of distorting regime into the power distribution

2.1.1. The electric scheme of the electrothermal installation

The electric scheme of the electrothermal device has a diode bridge rectifier and an inverter source. Therefore, it represents a current harmonic source. The current waveforms are deeply distorted from the sinusoidal shape, the harmonic current amplitudes are more influenced by the impedance of the power system, and the rectified voltage is less dependent on it.

2.2. Technical characteristics of the electrothermal installation

The hardening of electrothermal installation presented in **Figure 1** is composed of a converter CTC100K15 and two inductors for hardening the materials. CTC100K15 has the following electric characteristics: supplying voltage 3×400 V, 50 Hz, rated current 27 A, control voltage 24 Vdc, consumed power at high frequency 15 kW, voltage at medium frequency 500 Vac.

-Power supply system. The electrothermal installation is supplied from the three-phase power distribution on 0.4 kV, through a general distribution board.

-Solid-state relay. The on/off switching of the installation is made with three static contactors WG480-D50Z (Solid-state relay-SSR).

-Diode bridge rectifier. In the electric scheme, two diode bridge rectifiers KBPC 3508 are introduced with the following electric characteristics: maximum recurrent peak reverse voltage of 800 V, maximum RMS bridge input voltage of 560 V, maximum forward voltage drop per element of 1.2 V and maximum reverse current at rated DC blocking voltage per element of 1 μ A.

-Single-phase inverter. In order to control the frequency in the hardening process, an inverter with four IGBT transistors is used: FF150R12KS4 with the following characteristics: collector-emitter breakdown voltage of 1200 V, collector-emitter saturation voltage of 3.2 V and continuous collector current $I_{Cmax} = 225$ A.

-High-frequency transformers. For supplying the hardening inductors, the electrical scheme contains two high-frequency toroidal transformers—T1: 660/500 V, 40 kVA, 70/100 kHz; T2: 500/500 V, 15 kVA, 70/120 kHz.

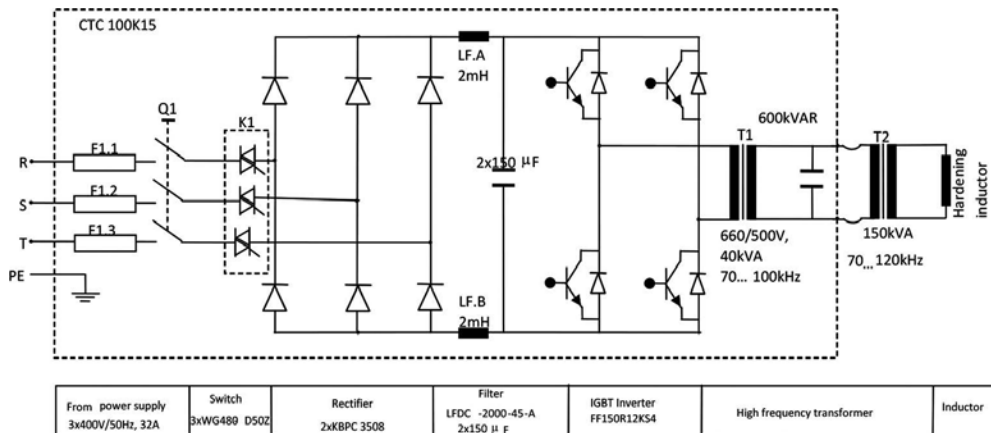


Figure 1. The electric scheme of the electrothermal installation.

2.2.1. Measurements of the electrical parameters using a data acquisition system

Data acquisition system (**Figure 2**) also follows the acquisition of samples from phase voltages and phase currents at the point of common coupling. In order to accomplish the system for measuring and computing the electrical parameters acquisition, an adapting block and a data acquisition board connected to a computer were used. The connection of data acquisition system to PCC [2] is presented in **Figure 2**. The phase voltages and phase currents were acquired using the adapting block, which was designed in order to accomplish the galvanic isolation between the electrothermal installation and the data acquisition system and also to realize the compatibility of the voltage levels. The data acquisition board has the following main characteristics: 16 analogic inputs with 16-bit resolution and 250 kS/s, two analogic outputs with 16-bit resolution and 740 kS/s, and 24 TTL digital in/out and one digital trigger.

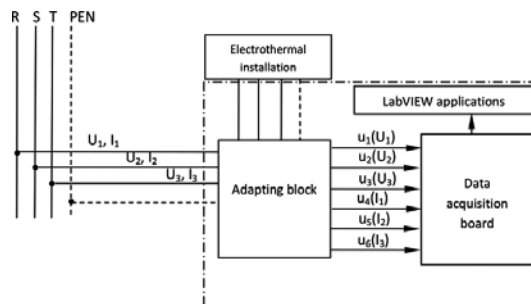


Figure 2. Connection to the power distribution of the data acquisition system.

In order to process the acquired data, two soft applications designed in LabVIEW 2011 were used: an acquisition application and a computing application. Using the acquisition application, the voltage and current samples are stored into a file. This file is loaded by the computing application, which is able to compute the most important electrical parameters that characterize the functioning of the electrothermal installation.

The algorithm configures 10 analogical input channels. The maximum sampling rate provided by the acquisition device is 250 kS/s, so, the maximum sampling rate for one channel is 25 kS/s. Equation (1) defines the number of samples on each period:

$$N = f_s \cdot T = 25 \text{ kS/s} \cdot 0.02 \text{ s} = 500 \text{ samples} \quad (1)$$

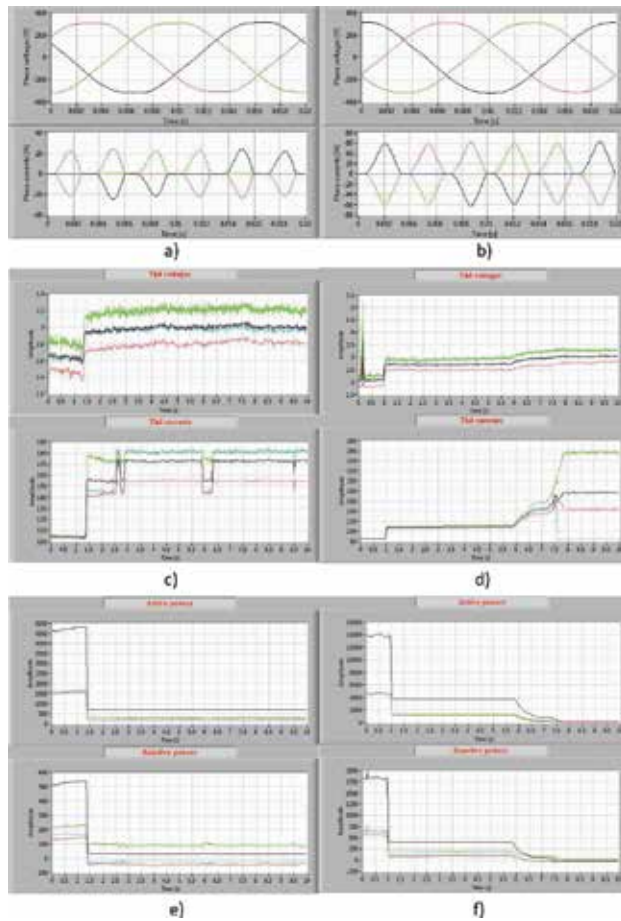


Figure 3. The variation of the electrical parameters: (a, b) Phase voltages and currents on 20 ms for 4.5 and 15 kW; (c, d) THD of voltages and currents on 10 s for 4.5 and 15 kW; (e, f) active and reactive powers on 10 s for 4.5 and 15 kW.

The acquisition is developed on a 10-s interval. The number of the acquired samples on each channel is 250,000. Using presented applications, 10 measurement sets were accomplished by increasing with 10% the power of electrothermal installation. In **Figure 3**, the variation on 0.02 and 10 s for phase voltages, phase currents, and also the variation on 10 s of total harmonic distortion (THD) for phase voltages, phase currents and the active and reactive powers are presented when the absorbed power of the power installation is 4.5 and 15 kW. Some observations must be made. In figures that represent the variation form of the currents and voltages, on period of 0.02 s, phase 1 is red, phase 2 is green and phase 3 is blue. For the variations on 10 s, the colour signification is black for all three phases, red for phase 1, green for phase 2 and blue for phase 3.

In **Table 1**, the electrical parameters obtained by accomplishing 10 measurement sets using data acquisition system are synthesized. In **Figure 4**, the variation of the current and voltage harmonic spectrum is presented. **Table 2** presents the harmonic current values.

Absorbed power (kW)	Phase voltages (V)	Phase currents (A)	THD voltages (%)	THD currents (%)
1.5	226.5	3.5	2.95	140
3	227.0	6.0	2.80	120
4.5	226.5	10.0	2.65	105
6	226.6	10.5	2.70	105
7.5	226.5	11.0	2.65	100
9	226.5	13.0	2.50	98
10.5	226.3	15.5	2.40	95
12	227.0	19.0	2.35	92
13.5	227.0	24.0	2.10	90
15	226.5	28.0	2.00	85

Table 1. Electrical parameters measured with data acquisition system.

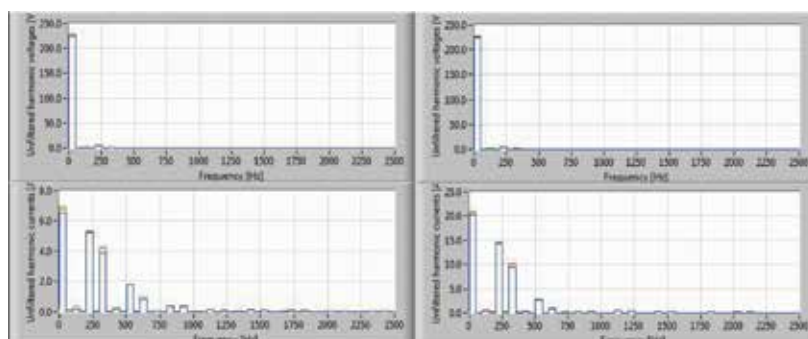


Figure 4. The variation of the current and voltage harmonic spectrum: (a) for 4.5 kW; (b) for 15 kW.

Installation power (kW)	Harmonic currents (A)									
Rank →	1	3	5	7	9	11	13	15	17	19
3.0	4.0	–	3.0	2.7	–	1.5	1.0	0.3	0.1	–
4.5	6.7	–	5.0	4.0	–	2.8	1.0	0.1	–	–
6.0	6.0	–	4.3	3.8	–	1.8	1.0	–	–	–
7.5	8.0	–	6.0	4.8	–	2.0	1.5	0.2	–	–
9.0	9.2	–	7.0	5.3	–	2.0	1.5	0.1	–	–
10.5	11.5	–	8.0	6.0	–	2.0	1.5	0.1	–	–
12.0	14.0	–	10.0	7.5	–	2.4	1.8	–	–	–
13.5	17.0	–	12.5	9.5	–	2.4	1.9	–	–	–
15.0	20.0	–	15.0	10.0	–	2.5	2.0	–	–	–

Table 2. Current harmonic spectrum.

The data acquisition system presents the advantage of permanent improvement concerning the soft applications that compute the electrical parameters in real time. The application displays the variation on 0.02 and 10 s of the phase voltages, phase currents, THD for phase voltages and phase currents and also the variation of active and reactive powers.

By studying **Figures 3** and **4**, we can conclude that the voltages are balanced and symmetrical and THD for phase voltages are approximately constant and can be neglected. By studying **Figure 3**, we can observe that the variation on 10 s presents three stages. Studying **Table 1**, the values of the phase currents are increasing (3.5/28 A) and the THD for phase currents are decreasing (140/85 A) with the increasing of the absorbed power. Even in this situation, the values of THD for phase currents fall below standard limits. The most important conclusion is that the harmonic currents must be compensated using appropriate devices. **Table 3** synthesizes the values for the electrical parameters.

Electrical parameters	Measurements with data acquisition system	
	4.5 kW (30% P _n)	15 kW (100% P _n)
Phase voltage (V)	226.5	226.5
Phase current (A)	10	27
Active power (W)	4700	14,000
Reactive power (VAr)	500	2000
THD voltages (%)	2.65	2
THD currents (%)	105	85

Table 3. Measured and computed values for the electrical parameters.

2.3. Designing of the harmonic compensation equipment

In order to compensate the harmonic distortion, the electrothermal installation must be equipped with harmonic compensation devices for bringing the parameters into standard limits. Therefore, a power-quality improvement system was designed. For decreasing the total harmonic distortion of the phase currents, the system contains a three-phase passive filtering system accorded on 5th, 7th, 11th and 13th harmonic currents and a shunt active filter, which controls an IGBT module using the software application. The function of monitoring the electrical parameters and controlling the hardware devices is accomplished by a LabVIEW application. As well, this application is able to generate pulses to command the active filter and also to receive the compensating currents [3] by data acquisition board. In **Figure 5**, the block scheme with the connection of the harmonic compensation devices is presented.

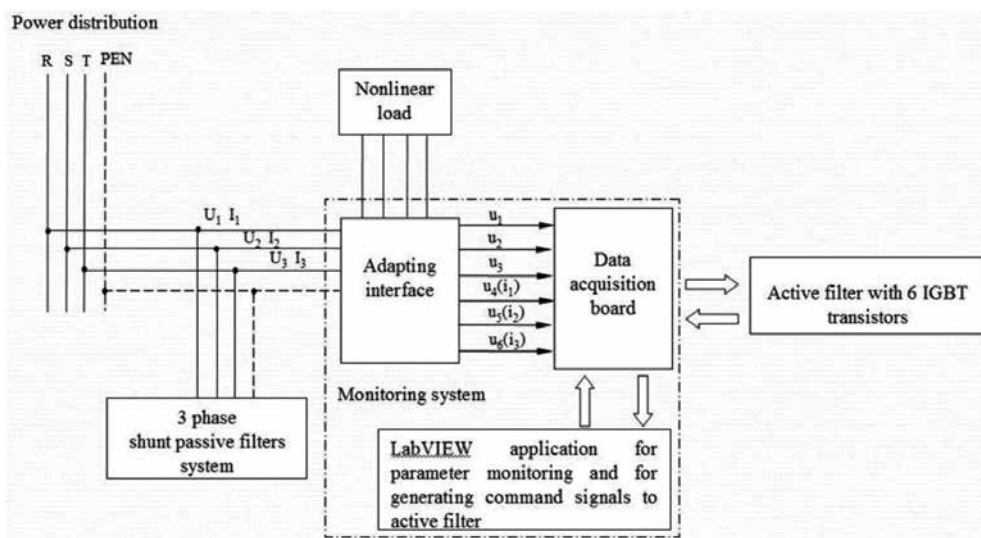


Figure 5. The connection of the harmonic compensation equipment.

2.3.1. Harmonic compensation using active filters

Shunt active filter determines in real time the compensating current and commands a power converter to synthesize it accurately.

Shunt active filter generally consists of two main blocks: pulse width modulation (PWM) converter for power processing and the active filter controller for signal processing. The PWM converter is responsible for power processing in synthesizing the compensating current that should be drawn from the power system.

The shunt active filter controller works in a closed-loop manner, continuously measuring the distorted current and computing the instantaneous values of the compensating current for the PWM converters [4].

The PWM converter should have high switching frequency in order to accurately reproduce the compensating currents [5].

The present paragraph details the algorithm for designing a three-phase active filter accomplished with IGBT transistors.

The distorted signals of currents and voltages are assumed by algorithm which uses the Fast Fourier Transformer for extracting the phase and the amplitude of the fundamental component.

The signals, which represent the fundamental components for the current and voltages, are calculated in complex form as Eqs. (2) and (3):

$$\underline{I}_{11} = I_{11} \cdot e^{j\theta_{11}}, \underline{I}_{21} = I_{21} \cdot e^{j\theta_{21}}, \underline{I}_{31} = I_{31} \cdot e^{j\theta_{31}} \quad (2)$$

$$\underline{U}_{11} = U_{11} \cdot e^{j\theta_{11}}, \underline{U}_{21} = U_{21} \cdot e^{j\theta_{21}}, \underline{U}_{31} = U_{31} \cdot e^{j\theta_{31}} \quad (3)$$

where I_{ik} represents the effective value of I_{ik} , θ is the phase difference, i is the phase and k is the current harmonic order.

The notations for the voltages are the same.

The positive sequence components of the voltages and currents are calculated by the algorithm. For generating the variation form of the positive sequence components, the application follows sinusoidal signal generators. The amplitudes of the generated signals are equal to the modules of the current fundamental component.

The residual currents result by subtracting the positive sequence of the fundamental components from distorted currents.

Using the symmetrical component method, the positive and negative components of the fundamental components of the current and voltage can be determined as in the relations:

$$\underline{I}_{11}^+ = \frac{1}{3}(\underline{I}_{11} + a \cdot \underline{I}_{21} + a^2 \cdot \underline{I}_{31}), \underline{U}_{11}^+ = \frac{1}{3}(\underline{U}_{11} + a \cdot \underline{U}_{21} + a^2 \cdot \underline{U}_{31}) \quad (4)$$

$$\underline{I}_{21}^+ = \frac{1}{3}(\underline{I}_{21} + a \cdot \underline{I}_{31} + a^2 \cdot \underline{I}_{11}), \underline{U}_{21}^+ = \frac{1}{3}(\underline{U}_{21} + a \cdot \underline{U}_{31} + a^2 \cdot \underline{U}_{11}) \quad (5)$$

$$\underline{I}_{31}^+ = \frac{1}{3}(\underline{I}_{31} + a \cdot \underline{I}_{11} + a^2 \cdot \underline{I}_{21}), \underline{U}_{31}^+ = \frac{1}{3}(\underline{U}_{31} + a \cdot \underline{U}_{11} + a^2 \cdot \underline{U}_{21}) \quad (6)$$

$$\underline{I}_{11}^{-} = \frac{1}{3}(\underline{I}_{11} + a \cdot \underline{I}_{21} + a^2 \cdot \underline{I}_{31}), \underline{U}_{11}^{-} = \frac{1}{3}(\underline{U}_{11} + a \cdot \underline{U}_{21} + a^2 \cdot \underline{U}_{31}) \quad (7)$$

$$\underline{I}_{21}^{-} = \frac{1}{3}(\underline{I}_{21} + a \cdot \underline{I}_{31} + a^2 \cdot \underline{I}_{11}), \underline{U}_{21}^{-} = \frac{1}{3}(\underline{U}_{21} + a \cdot \underline{U}_{31} + a^2 \cdot \underline{U}_{11}) \quad (8)$$

$$\underline{I}_{31}^{-} = \frac{1}{3}(\underline{I}_{31} + a \cdot \underline{I}_{11} + a^2 \cdot \underline{I}_{21}), \underline{U}_{31}^{-} = \frac{1}{3}(\underline{U}_{31} + a \cdot \underline{U}_{11} + a^2 \cdot \underline{U}_{21}) \quad (9)$$

where a is the shift operator:

$$a = e^{j \cdot 2\pi/3} \quad (10)$$

The residual currents are obtained by subtracting the positive components of the currents from the distorted signals (input signals)

$$i_{1r}(t) = i_1(t) - i_{11}^{+}(t), i_{2r}(t) = i_2(t) - i_{21}^{+}(t), i_{3r}(t) = i_3(t) - i_{31}^{+}(t) \quad (11)$$

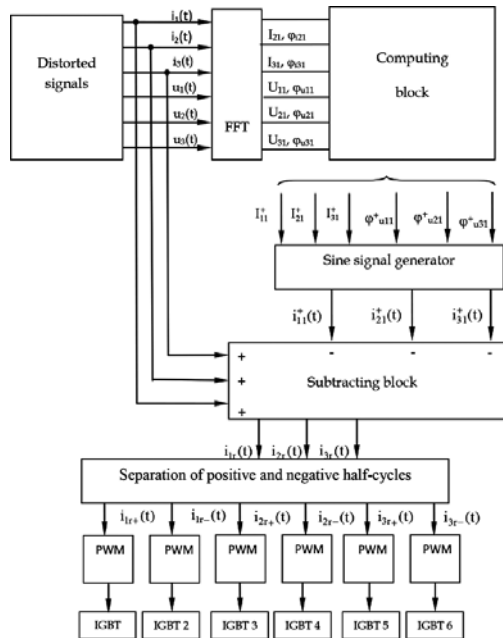


Figure 6. Generation of residual currents.

In order to generate PWM pulses proportional to the amplitude of the input signal samples, the algorithm contains six PWM modulators: three for positive and three for negative cycles of the residual currents. The six residual currents, $i_{1r+}(t)$, $i_{1r-}(t)$, $i_{2r+}(t)$, $i_{2r-}(t)$, $i_{3r+}(t)$, $i_{3r-}(t)$, are assumed by PWM modulators that generate pulses with the width proportional to the amplitude of the residual signals.

The modulation frequencies f_m are the following: 2500, 4000, 5000, 10,000 and 12,500 Hz. The block scheme for generating the residual currents is presented in **Figure 6**.

2.3.2. Description of the computing application for electrical parameters. Implementation of the active filter algorithm in LabVIEW environment

The LabVIEW computing application has to monitor the electrical parameters and also to control the active filter. This application also receives the compensating currents from the active filter using the data acquisition board [6]. In the following, the computing application is described.

Using the data acquisition board, the computing application can control the active filter system in order to generate the compensation currents.

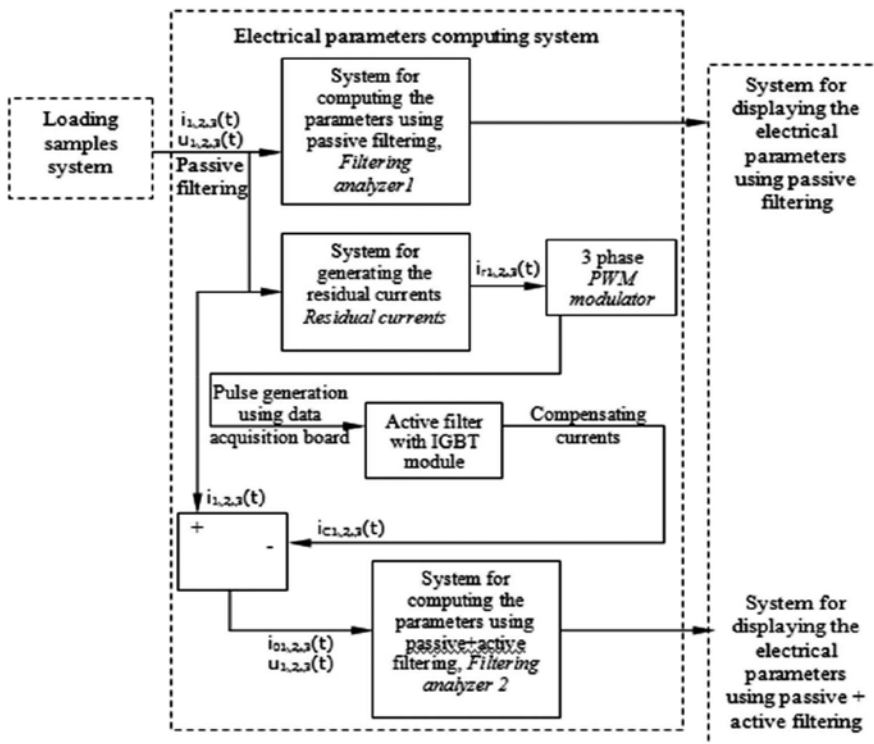


Figure 7. Block scheme of the active filtering system.

The control application is accomplished by subroutines which are functioning from the main. The block scheme of the active filtering algorithm is presented in **Figure 7**. The LabVIEW source code is presented in **Figure 8**.

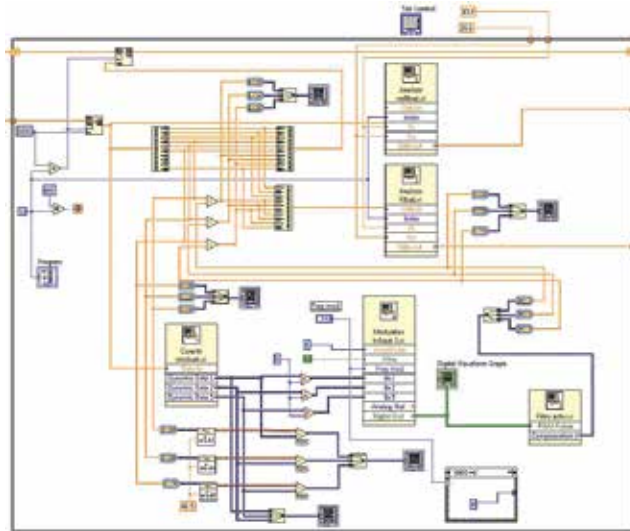


Figure 8. LabVIEW source code for computing application.

In the first sequence of the sequential structure which is named *System for loading samples*, the application allows the sample files to be loaded.

These files contain samples from currents and voltages that they have been assumed by the proposed system during the functioning of the electrothermal device when the passive filtering system was connected at the point of common coupling.

2.3.2.1. *System for loading samples*

The algorithm configures 10 analogical input channels.

The maximum sampling rate provided by the acquisition device is 250 kS/s, so, the maximum sampling rate for one channel is 25 kS/s.

The LabVIEW application is structured in such a way that the measuring set can be selected from the front panel. The algorithm generates a vector and its elements are the amplitudes of the assumed samples.

2.3.2.2. *System for electrical parameters computing*

The second sequence of the application contains the computing system, which calculates the electrical parameters using the samples assumed during the functioning of the electrothermal installation equipped with the passive filtering system. The samples of voltage and current

signals are taken from the loading matrix presented in the first sequence. With appropriate functions from LabVIEW library, the algorithm divides the currents from the voltage samples, calculating the effective values of the phase voltages and currents on each period of the power system.

The computing application is structured in subroutines, which will be detailed in the following section.

2.3.2.2.1. Parameter analyser subroutine

The present subroutine computes the electrical parameters that characterize the functioning of the electrothermal installation which are as follows: the effective values for the phase voltage and phase currents, the total values of the voltages and currents, apparent, active, reactive and distorted powers and also the total harmonic distortion of voltages and currents.

In **Figure 9**, the LabVIEW code for the parameter analyser is presented.

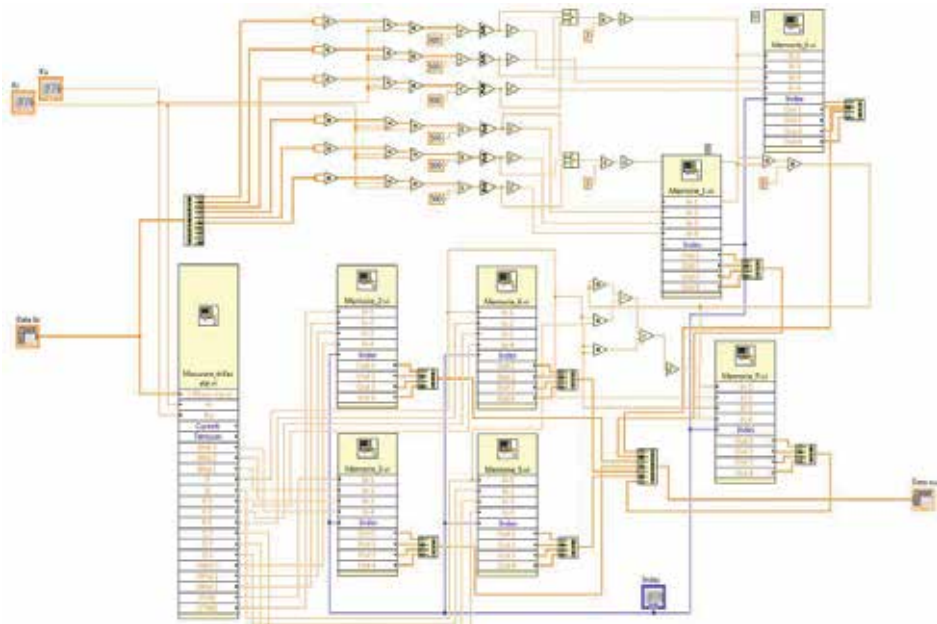


Figure 9. LabVIEW source code for parameter analyser subroutine.

2.3.2.2.2. Residual current subroutine

The presented subroutine is based on the algorithm described here and uses another subroutines, *Current phasors* and *Voltage phasors* for computing the positive sequence components of the current and voltage fundamental components.

2.3.2.2.3. PWM modulator

The algorithm must generate PWM pulses, which must be proportional with the input signals' sample amplitude; therefore, it includes six PWM modulators: three for negative and three for positive cycles of the residual currents. Mathematical functions from LabVIEW library must separate the cycles. The modulation frequencies for PWM modulators can be selected from the first panel of the application. This property offers a high accuracy level to the output signals.

The PWM modulators are subroutines, which run under the main application. The six residual currents, $i_{1r+}(t)$, $i_{1r-}(t)$, $i_{2r+}(t)$, $i_{2r-}(t)$, $i_{3r+}(t)$, $i_{3r-}(t)$, are acquired by PWM modulators, which accomplishes pulses that have proportional width with the amplitude of the residual currents. PWM modulators are presented in **Figure 10**.

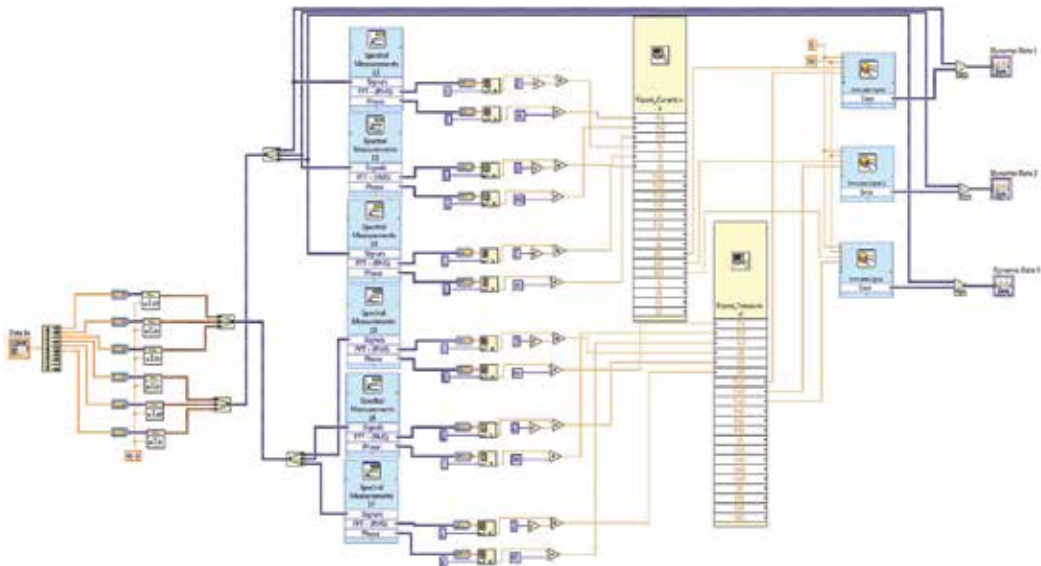


Figure 10. LabVIEW source code for residual current subroutine.

The modulation frequencies f_m are the following: 2500, 4000, 5000, 10,000 and 12,500 Hz and can be chosen from the front panel. Each subroutine generates an array containing a pulse. The sample number for each pulse depends on the maximum internal clock frequency of the acquisition board and also the PWM controller modulation frequencies f_m , as in Eq. (12):

$$N_{\text{samples}} = f_{\text{clock}} / f_m \quad (12)$$

These number of samples (40, 25, 20, 10 and 8), correlated with frequencies of 2500, 4000, 5000, 10,000 and 12,500 Hz, are used to select the index of the sample, which gives the width of the PWM controller pulse, as in **Figure 11**.

The generated pulses width depends on the residual signal amplitude. In order to determine the generated pulses width, a base of selection of the sample was created, using the residual signal samples. The base of selection represents one one-dimensional (1D) array generated in the following manner: the 500 samples/period of the residual signal are interleaved using the LabVIEW function *Interleave 1D Array*. The algorithm obtains a 1D array of 2000 samples that implies a single residual signal period, sampled four times slower. This array is repeated five times in order to get a new 1D array with 10,000 elements, which was previously mentioned, being the base of selection of the sample.

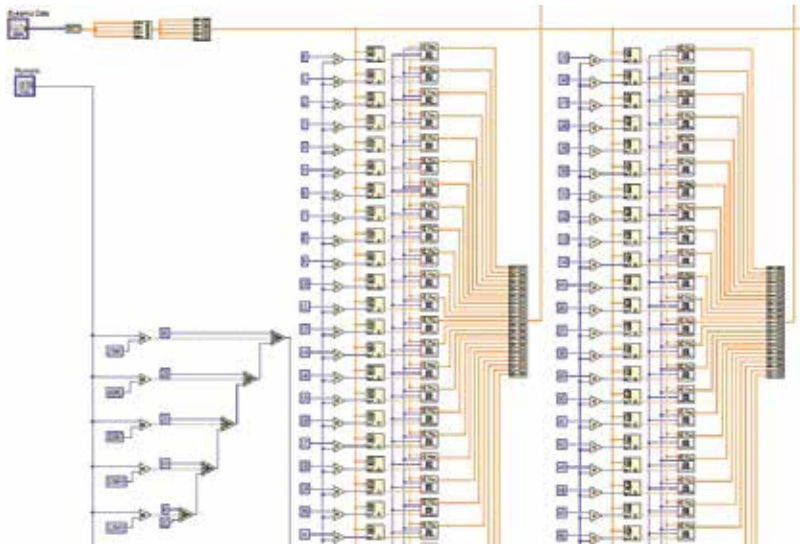


Figure 11. PWM modulator for one half-cycle.

The sample selection that determines the pulse width is realized using the function *Index Array* that returns a base of selection element. In the following, the situations that appear by varying the modulation frequency f_m are related.

If $f_m = 2500$ Hz, the sample width is taken from index $n_{2500} = 40 \cdot i$

If $f_m = 4000$ Hz, the sample width is taken from index $n_{4000} = 25 \cdot i$

If $f_m = 5000$ Hz, the sample width is taken from index $n_{5000} = 20 \cdot i$

If $f_m = 10,000$ Hz, the sample width is taken from index $n_{10,000} = 10 \cdot i$

If $f_m = 12,500$ Hz, the sample width is taken from index $n_{12,500} = 8 \cdot i$, where $i = 0 \dots 249$ represents the order number for the generated pulses.

The number of pulses generated by PWM is 50, 80, 100, 200 and 250 and is correlated with frequencies of 2500, 4000, 5000, 10,000 and 12,500 Hz.

It can be observed that the duration of generated pulses has the lowest value when the modulation frequency has the highest value, but the product of them is a constant equal to 2000.

2.3.3. Measurements accomplished during the functioning of the electrothermal installation equipped with harmonic compensation devices

2.3.3.1. Variation of the electrical parameters using passive filtering system

The computing application uses the information from txt files generated when the passive filtering system is connected. In **Figures 12** and **13**, the variation of the electrical parameters on 10 s is presented: phase voltages, phase currents, THD for voltages and currents, and active and reactive powers. The absorbed power of the electrothermal installation is 4.5 and 15 kW and only the passive filtering system is connected. The colour signification for the parameter variation is the following: black for all three phases, red for phase 1, green for phase 2 and blue

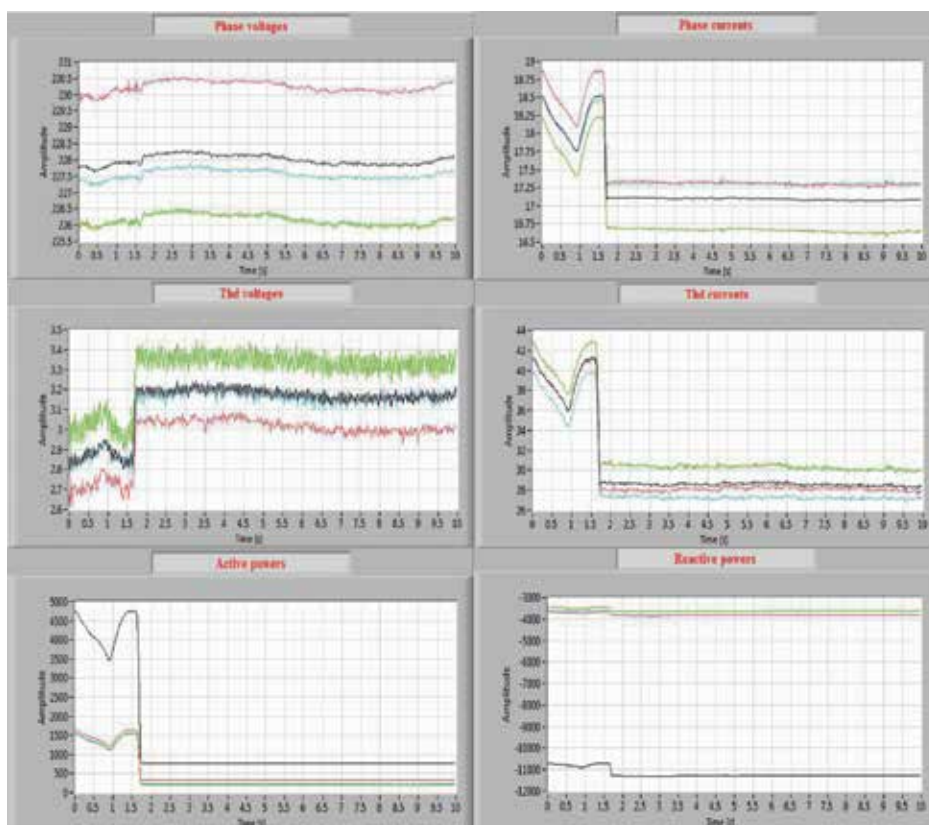


Figure 12. Variation of the electrical parameters, passive filtering, $P = 4.5$ kW.

for phase 3. In **Figure 14**, the variation of the current harmonic spectrum with passive filters is presented.

Table 4 synthesizes the average values generated by the variation from **Figures 12 and 13**, and **Table 5** synthesizes the values from **Figure 14**.

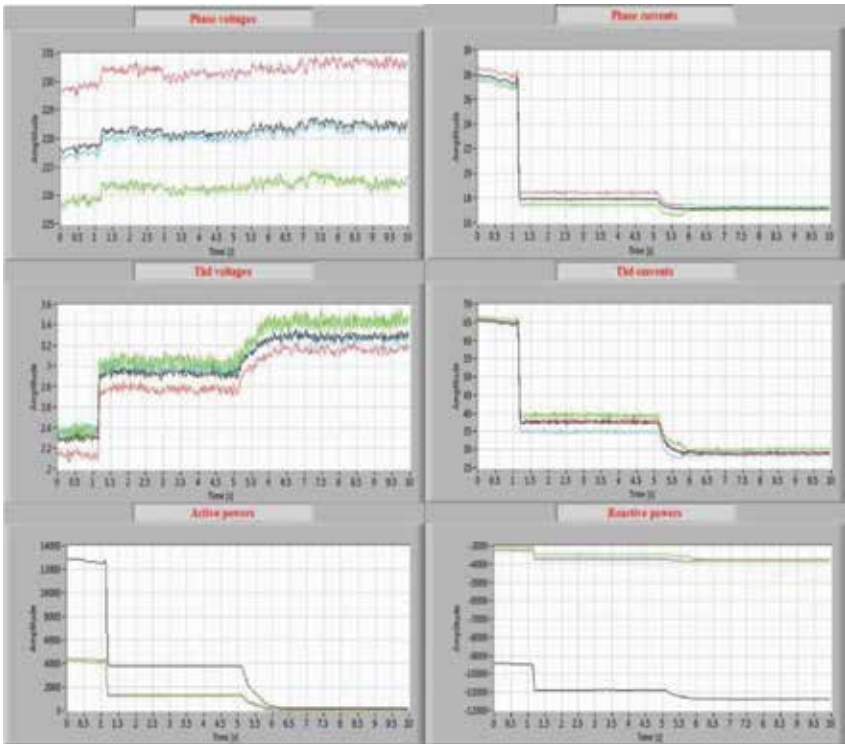


Figure 13. Variation of the electrical parameters, passive filtering, $P = 15 \text{ kW}$.

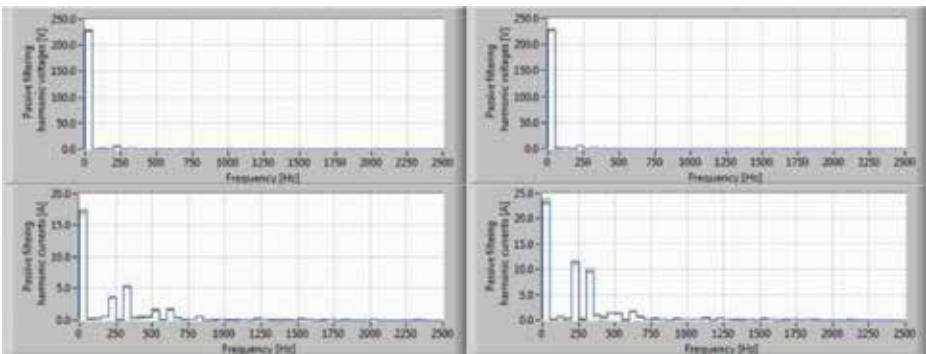


Figure 14. Variation of the harmonic current spectrum with passive filtering system (4.5 and 15kW).

Phase voltage (V)	Phase current (A)	Active power (W)	Reactive power (VAr)	Distorted power (VAd)	Apparent power (VA)	THD for voltages (%)	THD for currents (%)
P = 1.5 kW (10% P_n)							
227.6	17.0	1300	-11,000	3000	11,476	3.15	28.6
P = 3 kW (20% P_n)							
228.0	17.4	2700	-11,000	3000	11,717	3.00	33.0
P = 4.5 kW (30% P_n)							
228.0	18.0	4000	-10,800	4800	12,477	2.90	39.0
P = 6 kW (40% P_n)							
228.0	18.1	4000	-10,800	4800	12,477	2.90	37.0
P = 7.5 kW (50% P_n)							
228.0	18.7	5000	-10,800	5000	12,909	2.85	42.5
P = 9 kW (60% P_n)							
228.4	19.3	5800	-10,700	5000	13,158	2.85	45.0
P = 10.5 kW (70% P_n)							
228.8	19.5	6000	-10,700	6000	13,656	2.80	45.0
P = 12 kW (80% P_n)							
228.2	22.2	8800	-10,000	7000	15,048	2.65	55.0
P = 13.5 kW (90% P_n)							
228.0	25.0	11,000	-10,000	8000	16,882	2.50	60.0
P = 15 kW (100% P_n)							
227.5	28.0	13,000	-9500	10,000	18,954	2.30	65.0

Table 4. Electrical parameters obtained with passive filtering system for specified active powers (percent of nominal power).

Active power (Kw) Rank →	Harmonic currents (A)											
	1	3	5	7	9	11	13	15	17	19	21	
1.5	16	–	2	4	–	3	2	–	–	–	–	
3	16	–	3	4	–	2	2	–	–	–	–	
4.5	17	–	4	5	–	2	2	–	–	–	–	
6	17	–	3	5	–	2	2	–	–	–	–	
7.5	17.5	–	4	6	–	2	2	–	–	–	–	
9	18	–	5	6	–	2	3	–	–	–	–	
10.5	18	–	5	6	–	2	3	–	–	–	–	
12	20	–	7	7.5	–	2	3	–	–	–	–	
13.5	21	–	9	8	–	2	2	–	–	–	–	
15	23	–	11	10	–	2	3	–	–	–	–	

Table 5. Average values for the harmonic current spectrum with passive filtering system.

2.3.3.2. Variation of the electrical parameters using active filtering system

Table 6 synthesizes the average values generated by the variation from **Figures 15** and **17**, and **Table 7** works with **Figure 16**.

Phase voltage (V)	Phase current (A)	Active power (W)	Reactive power (VAr)	Distorted power (VAd)	Apparent power (VA)	THD for voltages (%)	THD for currents (%)
P = 1.5 kW (10% P_n)							
228	14.7	9500	-1250	3000	10,041	3.15	16.0
P = 3kW (20% P_n)							
228	15.5	10,500	-800	2000	10,719	3.00	17.5
P = 4.5 kW (30% P_n)							
228	15.8	10,600	-600	2100	10,823	2.90	17.5
P = 6 kW (40% P_n)							
228	15.0	9800	-2000	3000	10,442	2.90	20.0
P = 7.5 kW (50% P_n)							
228	16.0	10,500	-1250	3000	10,991	2.85	20.0
P = 9 kW (60% P_n)							
228.3	16.5	10,800	-1200	2200	11,087	2.85	19.0
P = 10.5 kW (70% P_n)							
228.7	16.5	11,000	-250	2500	11,283	2.80	18.1
P = 12 kW (80% P_n)							
228.3	19.0	12,500	-1200	3000	12,911	2.61	21.0
P = 13.5 kW (90% P_n)							
228	21.5	14,000	-80	2300	14,188	2.50	17.0
P = 15 kW (100% P_n)							
227.8	23.0	15,000	-100	2500	15,207	2.30	17.0

Table 6. Electrical parameters obtained with passive and active filtering system for specified active powers (percent of nominal power).

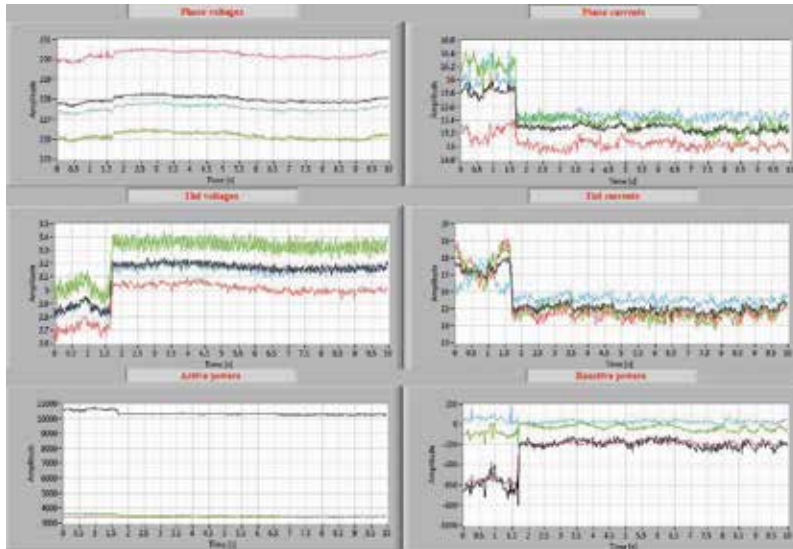


Figure 15. Variation of the electrical parameters, passive and active filtering, $P = 4.5$ kW.



Figure 16. Variation of the harmonic current spectrum with passive and active filtering system.

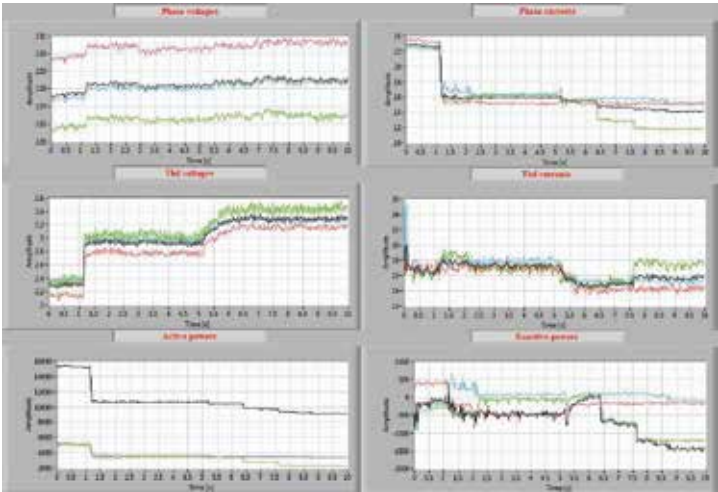


Figure 17. Variation of the electrical parameters, passive and active filtering, $P = 15$ kW.

Active power (kW)	Harmonic currents (A)											
Rank →	1	3	5	7	9	11	13	15	17	19	21	
1.5	14	–	2	4	–	3	2	–	–	–	–	
3	15	–	2	4	–	2	2	–	–	–	–	
4.5	15	–	2	1	–	2	1	–	–	–	–	
6	15	–	2	3	–	1	1	–	–	–	–	
7.5	15	–	3	2	–	1	1	–	–	–	–	
9	16	–	2	1	–	1	1	–	–	–	–	
10.5	16	–	2	2	–	1	1	–	–	–	–	
12	18	–	3	3	–	1	1	–	–	–	–	
13.5	20	–	2	3	–	1	1	–	–	–	–	
15	21	–	3	3	–	1	1	–	–	–	–	

Table 7. Harmonic currents with passive and active filtering system.

In **Figures 15** and **17**, the variation of the electrical parameters is presented when the passive and active filtering system is connected and **Figure 16**, variation of the current harmonic spectrum, is presented.

The variation on 0.02 s of the active filter signals is presented in **Figures 18–22**, during the functioning of the electrothermal installation at nominal power of 15 kW. The active filtering system generates an input signals soft compensation; therefore, the phase currents get more sinusoidal. This achievement is followed by the total harmonic distortion of the phase currents that is decreased. These parameter variations are obtained by choosing the modulation frequency f_m of the active filter at the following values: 12,500, 10,000, 5000, 4000 and 2500 Hz.

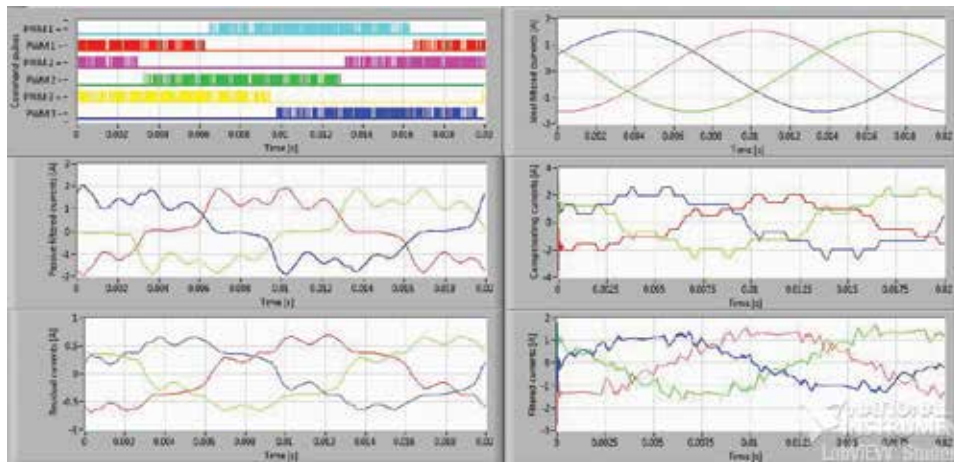


Figure 18. Variation of the active filter signals, $P = 15$ kW, $f_m = 12,500$ Hz.

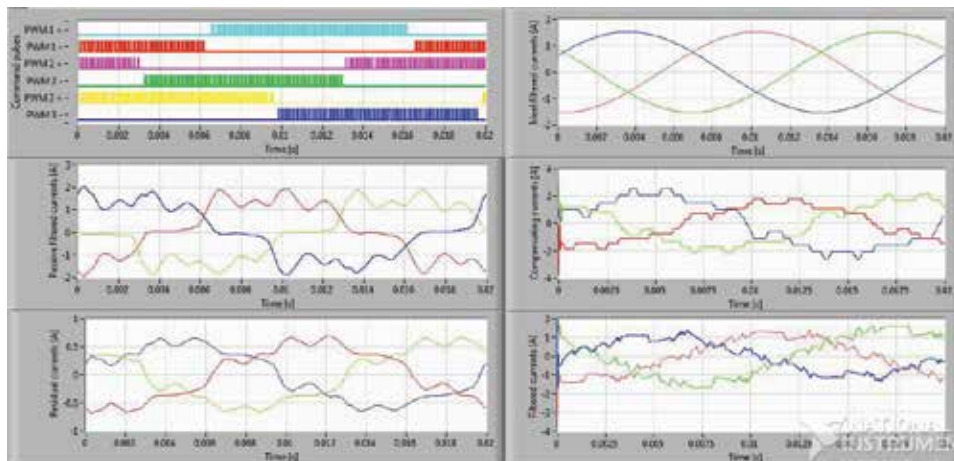


Figure 19. Variation of the active filter signals, $P = 15$ kW, $f_m = 10,000$ Hz.

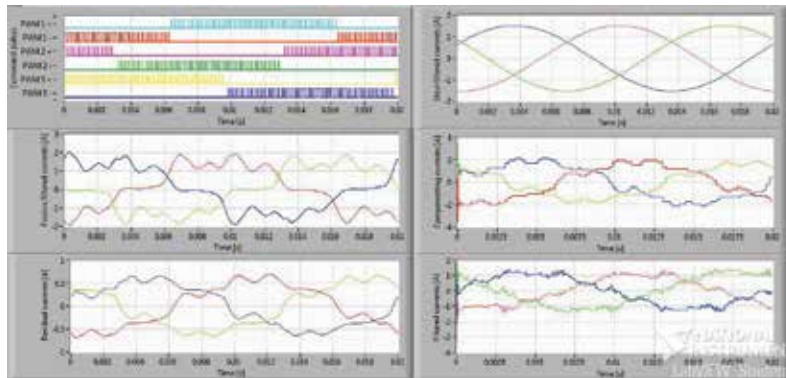


Figure 20. Variation of the active filter signals, $P = 15 \text{ kW}$, $f_m = 5000 \text{ Hz}$.

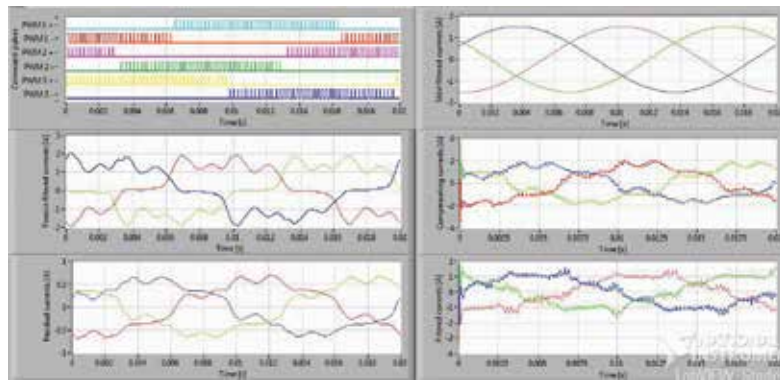


Figure 21. Variation of the active filter signals, $P = 15 \text{ kW}$, $f_m = 4000 \text{ Hz}$.

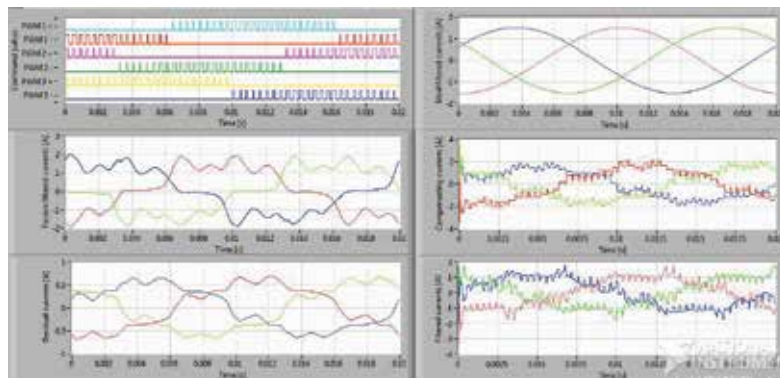


Figure 22. Variation of the active filter signals, $P = 15 \text{ kW}$, $f_m = 25,000 \text{ Hz}$.

By choosing the active filter modulation frequency, the following signals are depicted:

- *Commanding pulses* represent the signals that are generated by PWM modulator subroutine. The commanding signals represent pulses with variable duration and constant amplitude, they being proportional with the distorted signal sample amplitudes, which are assumed into the active filter application. PWM1+ signal represents the PWM modulation of the positive cycles of the phase 1, and PWM1- signal represents the PWM modulation of the negative cycles of the phase 1 and so on.

- *Passive filtered current signals* represent the phase currents, which are assumed using this system when the passive filter system is connected.

- *Residual current signals* represent the current signals that must be compensated.

Ideal filtered current signals are ideal-simulated filtered currents.

- *Compensating current signals* are the signals generated by the active filtering system.

- *Filtered current signals* represent the phase currents generated by using the passive and active filtering system.

The colour signification is presented in the following: cyan for PWM1+, red for PWM 1-, magenta for PWM 2+, green for PWM2-, yellow for PWM 3+ and blue for PWM 3-. In the other figures, phase 1 is red, phase 2 is green and phase 3 is blue. The phase currents are represented in red for phase 1, green for phase 2 and blue for phase 3.

In **Table 8**, the variation of THD of the phase currents is synthesized, when the electrothermal installation is equipped with passive filters and also with the entire filtering system, that is, passive and active filtering system.

Active power (kW)	THD (%) Passive filtering	THD (%) Passive + active filtering for $f_m =$				
		12,500 Hz	10,000 Hz	5000 Hz	4000 Hz	2500 Hz
1.5	28	16	1.5	28	16	1.5
3	33	17	3	33	17	3
4.5	39	17	4.5	39	17	4.5
6	37	20	6	37	20	6
7.5	42	20	7.5	42	20	7.5
9	45	19	9	45	19	9
10.5	45	18	10.5	45	18	10.5
12	55	21	12	55	21	12
13.5	60	17	13.5	60	17	13.5

Table 8. Variation of THD of the phase currents.

A very important conclusion regarding the electrical parameter variation is the voltage balancing [7]. The phase voltages are constant [8] during the 10% increase of the consumed power of the installation and they are situated between 227.5 and 228.8 V in **Table 4** and also in **Table 6**.

3. Conclusions

This paper describes an intelligent system, which is able to command and control an active filtering system using a software application and a data acquisition board.

The electrothermal installation, which is equipped only with passive harmonic compensation devices, generates distorted regime into the power system.

When the active filtering system is connected, the effective values of the phase currents increase from 23 to 28 A. Another important conclusion is the voltage balancing.

The electrothermal installations with electromagnetic induction are current harmonic sources. Therefore, the low values of the total harmonic distortion of the phase voltages and the very high values of the total harmonic distortion of the phase currents are obtained. As it can be seen in **Table 4** and also in **Table 6**, THD for phase voltages lays in standard limits. The most important improvement that is reached in the present study consists in decreasing the phase currents' THD when the active filtering system is connected, from 60 to 17%. This is due to decreasing of the distorted power from 10,000 to 2500 VAd when the active filtering system is connected.

Concerning the variation of the active filter signals, it can be concluded that the number of pulses for each cycle is decreased with the decreasing of the modulation frequency. The duration of the pulses becomes proportionally higher because the pulse-frequency generation by the LabVIEW appropriate functions decreases. The active filter presents the highest accuracy when the modulation frequency is chosen on a maximum level (12,500 Hz), because a less number of samples from the input signals are processed. Therefore, the amplitude of the unwished oscillations, which overlaps the filtered signals, can be neglected. The level of these oscillations is decreasing with the increasing of the modulation frequency.

The monitoring and filtering system presented in this chapter represents an alternative method to the classical harmonic compensation systems, which determines high costs [9]. The software and devices described here represent laboratory equipment, which has great advantage for permanent improvement.

Acknowledgements

This work was supported by a grant from the Romanian National Authority for Scientific Research and Innovation, CNCS – UEFISCDI, project number PN-II-RU-TE-2014-4-1788.

Author details

Raluca Rob*, Gelu Ovidiu Tirian and Caius Panoiu

*Address all correspondence to: raluca.rob@fih.upt.ro

Department of Electrical Engineering and Industrial Informatics, Faculty of Engineering
Hunedoara, Politehnica University of Timisoara, Romania

References

- [1] F. Hideaki, A. Hirofumi. The unified power quality conditioner: the integration of series and shunt-active filters. *IEEE Transactions in Power Electronics*. 1998;13(2):315–322.
- [2] K. Wada, T. Shimizu. Mitigation method of 3rd-harmonic voltage for a three-phase four-wire distribution system based on a series active filter for the neutral conductor. *IEEE IAS Annual Meeting*. 2002;1:64–69.
- [3] F. Z. Peng, H. Akagi, and A. Nabae. A new approach to harmonic compensation in power systems – a combined system of shunt passive and series active filters. *IEEE Transactions on Industry Applications*. 1990;26(6):983–990.
- [4] P. W. Hammond. A new approach to enhance power quality for medium voltage ac drives. *IEEE Transactions on Industry Applications*. 1997;33(1):202–208.
- [5] W. Tangtheerajaronwong, T. Hatada, K. Wada, and H. Akagi. Design and performance of a transformerless shunt hybrid filter integrated into a three-phase diode rectifier. *IEEE Transactions on Power Electronics*. 2007;22(5):1882–1889.
- [6] Z. Pan, F. Z. Peng. A sinusoidal PWM method with voltage balancing capability for diode-clamped five-level converters. *IEEE Transactions on Industry Applications*. 2009;45(3):1028–1034.
- [7] A. Iagar, G. N. Popa, and C. M. Dinis. The influence of home nonlinear electric equipment operating modes on power quality. *WSEAS Transactions on Systems*. 2014;13:357–367.
- [8] M. Panoiu, F. Neri. Open research issues on modelling, simulation and optimization in electrical systems. *WSEAS Transaction on Systems*. 2014;13:332–334.
- [9] G. N. Popa, S. Deaconu, C. M. Diniş, and A. Iagăr. A case study of an analogical distance relay for the 110kV electric power lines. *WSEAS Transactions on Systems*. 2010;9(10): 1063–1072.

Comparison Study of AI-based Methods in Wind Energy

Ping Jiang, Feng Liu and Yiliao Song

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/63716>

Abstract

Wind energy forecasting is particularly important for wind farms due to cost-related issues, dispatch planning, and energy market operations. Thus, improving forecasting accuracy becomes an urgent task for researchers in the field of wind energy. However, there is limited research to discuss an overall comparison among various forecasting types, which is a foundation for future works with respect to wind energy prediction because this comparison may reveal whether there is a best model for a specific forecasting type or specific data in this field. For the purpose of laying a strong foundation for wind energy research, this chapter introduces five basic forecasting models, which are Autoregressive Moving Average Model (ARMA), Back-Propagation Neuron Network (BPNN), Support Vector Regression (SVR), Extreme Learning Machine (ELM), and Adaptive Network-Based Fuzzy Inference System (ANFIS) with implement codes before comparing the forecasting effectiveness of five different models in three wind farms based on five forecasting types. Comparison results indicate that each model has great divergent forecasting results in different wind farms and every forecasting type has its own “best model.”

Keywords: wind power, AI-based models, forecasting, comparison, Matlab codes

1. Introduction

The important environmental advantages of renewable energy sources have been significantly noticed, which results in most industrialized countries committed to developing the installation of wind power plants [1]. The share of total installed capacity of wind power in China has reached approximately 27% in the global capacity, which is 96.37 GW, due to the historically high installation of new wind power capacity.¹ Moreover, a renewable-energy-

¹ N.E. Administration, http://www.nea.gov.cn/2015-02/12/c_133989991.htm (2015).

oriented power system was proposed as the fundamental aim of China's energy transformation during the Asia-Pacific Economic Cooperation (APEC) Conferences.² Advanced prediction techniques are urgently needed to integrate wind energy into the electrical power grid in a manner that benefits both Transmission System Operators (TSOs) and Independent Power Producers (IPPs) [2,3].

Since wind energy has the inherently intermittent nature and stochastic nonstationarity, it brings significant levels of uncertainty to system operators [4]. Thus, accurate wind forecasts are of primary importance to solve operational, planning, and economic problems in the growing wind power scenario [4,5]. Current wind power forecasting research has been divided into point forecasts (also called deterministic predictions) [6–8] and uncertainty forecasts [9,10]. Deterministic forecasts deliver specific amounts of wind power and focus on reducing the forecasting error [11]. By contrast, it is essential to decision-making processes and electricity market trading strategies [12] that uncertainty forecasts provide uncertainty information for system operators to manage the wind power generation of wind farms [11,13].

The existing approaches published in the literature with respect to wind energy prediction can be divided into three categories—artificial intelligent model, physical model, and statistical model—and sometimes a hybrid one, which integrates advantages of different categories, is involved. Researchers often utilize them to forecast various types of wind speed and wind power by various types including the multi-step forecasting, long-term wind speed (power) forecasting, and so on. However, few literature exist to discuss an overall comparison among various forecasting types, which is a foundation for future works of wind energy prediction because this comparison may reveal whether there is a best model for a specific forecasting type or specific data in the field of wind energy.

The following parts in this paper are demonstrated by four sections: methodology briefly introduces forecasting approaches that this paper adopts; data collection specifically illustrates three types of wind data; simulation and result displays and evaluates the final results of forecasting effectiveness of each model; and conclusion draws the main results that this paper investigates.

2. Methodology

In this section, we will introduce five basic models including Autoregressive Moving Average Model (ARMA), Back-Propagation Neuron Network (BPNN), Support Vector Regression (SVR), Extreme Learning Machine (ELM), and Adaptive Network-Based Fuzzy Inference System (ANFIS) as forecasting methods in this paper to predict wind speed. For the purpose of clearly demonstrating procedures of forecasting wind speed using these methods, this chapter will attach the Matlab code after introduction of each model and the wind speed series is defined as follows:

² N.E. Administration, http://www.nea.gov.cn/2015-06/08/c_134305870.htm (2015).

$$WS = (ws_1, ws_2, \dots, ws_N) \quad (1)$$

where N is a positive integer and $ws_n \in (0, +\infty) \subset \mathbb{R}$ represents the wind speed of time n .

2.1. Autoregressive moving average model

2.1.1. Brief introduction of ARMA

This section displays ARMA model and the ARMA(p, q) model can be expressed as follows [14]:

$$y_t = \delta + \sum_{i=1}^p \phi_i y_{t-i} + \sum_{j=1}^q \phi_j e_{t-j} + e_t \quad (2)$$

Where δ is the constant term of the ARMA model, ϕ_i is the i th autoregressive coefficient, ϕ_j is the j th moving average coefficient, e_t is the error term at time period t , and represents the value of wind speed observed or forecasted at time period t . Thus, the first step in applying ARMA model to forecast wind speed is to identify p and q , which is related to the stationarity of the time series, which means that the stationarity assumption of observed series should be checked first. For this purpose, inspection of the run plots and Auto Correlation Function (ACF) plots can be used for deciding on the order of differencing.

2.1.2. Implement for wind speed forecasting

Since the Matlab has the ARMA package, we will implement this model to forecast wind speed using five steps as follows:

1. Identify the domain of p and q . In this chapter, we set $p \in \{1, 2, 3, 4, 5\}$ and $q \in \{1, 2, 3, 4, 5\}$.
2. For each pair (p, q) , calculate corresponding model's AIC value.
3. Select the pair (p, q) , which makes corresponding model's AIC value to be minimum one among all of pairs of (p, q) as the best (p, q) and denote this pair as (p_test, q_test) .
4. Apply (p_test, q_test) to establish ARMA model.
5. Forecast m steps using the model established in step (4).

The detailed Matlab codes are listed in Code 1.

Code 1. Matlab Codes of ARMA.

```

function [y_output,p_test,q_test] = arma_dynmc(Ws,m)

z = Ws; step = m;

test = [];

for p = 1:5

    for q = 1:5

        m = armax(z,[p,q]);

        AIC = aic(m);

        test = [test;p q AIC];

    end

end

[min_aic,min_i] = min(test(:,3));

p_test = test(min_i,1);q_test = test(min_i,2);

m = armax(z,[p_test,q_test]);

P = predict(m,z,step);

y_output = P(end-step+1:end,1)';

```

2.2. Back-propagation neuron network

2.2.1. Brief introduction of BPNN

Mccelland and Rumelhart developed the BP neural network model in 1985. There are three layers in a particular network: the input, hidden, and output layers. Each layer has designed nodes, whose functions aim to calculate the inner product of the input vector and weight vector by the transfer function [15]. The process of BPNN is demonstrated by the following steps:

1. Initialize. Assume that the input layer has n nodes, and hidden layer has l nodes, and output layer has m nodes. Let the network distribute values randomly for each threshold value θ_j, γ_i and the connection weight $w_{ij}, v_{jt}, i = 1, 2, \dots, n, j = 1, 2, \dots, l$, and $t = 1, 2, \dots, m$.
2. Calculate the output of hidden layer. Assume $\mathbf{X} \in [-1, 1]^n$ is the input space and $\mathbf{Y} \in [-1, 1]^m$ is the output space, which means \mathbf{X} is an n -dimensional space and \mathbf{Y}

is an m -dimensional space. We denote each element in \mathbf{X} as x_i and each element in \mathbf{Y} is y_t . Thus, the output of hidden layer can be calculated using the following function:

$$H_j = f\left(\sum_{i=1}^n w_{ij}x_i - \theta_j\right), j = 1, 2, \dots, l \quad (3)$$

$f(\bullet)$ is the transfer function and is expressed as follows:

$$f(z) = \frac{1}{1 + e^{-z}}$$

3. Calculate the output of the network. Using H_j , v_{jk} , and γ_t , the output of the network can be calculated as follows:

$$O_t = \sum_{j=1}^l v_{jk} H_j - \gamma_t, t = 1, 2, \dots, m \quad (4)$$

4. Error calculation. The error between y_t and O_t can be expressed as $e_t = O_t - y_t$.
5. Weights updating. According to back-propagation algorithm, the weights w_{ij} , v_{jt} can be updated by using the following functions:

$$w_{ij} = w_{ij} + \eta H_j (1 - H_j) x_i \sum_{t=1}^m w_{jt} e_t \quad (5)$$

$$v_{jt} = v_{jt} + \eta H_j e_t \quad (6)$$

where η is the learning rate and $i = 1, 2, \dots, n$, $j = 1, 2, \dots, l$, and $t = 1, 2, \dots, m$.

6. Biases updating. Similarly, the biases θ_j , γ_t can be updated by using the following functions:

$$\theta_j = \theta_j + \eta H_j (1 - H_j) \sum_{t=1}^m w_{jt} e_t \quad (7)$$

$$\gamma_t = \gamma_t + e_t \quad (8)$$

7. Iteration. If the error is more than the expected value, then execute steps (2)–(6). Otherwise, denote θ_j , γ_t , w_{ij} and v_{jt} as the optimized parameter of this network with respect to \mathbf{X} and \mathbf{Y} .

2.2.2. Implement for wind speed forecasting

In this part, how to apply BPNN to forecast wind speed will be introduced in detail and the MATLAB code for predicting wind speed will be attached. There are six steps to forecast wind speed applying BPNN as follows:

1. Design forecasting mode. In this chapter, AI-based model will apply the following mode to forecast wind speed of different wind databases

$$(ws_{t-1}, ws_{t-2}) \xrightarrow{\text{forecast}} ws_t \quad (9)$$

Eq. (8) indicates that the input space \mathbf{X} for BPNN model is a two-dimensional space and the output space \mathbf{Y} is a one-dimensional space, meaning that $n = 2$ and $m = 1$ as described in Section 2.2.1.

2. Assign training samples. Based on step (1), the input and output of training samples are expressed as follows:

$$Train_input = \begin{pmatrix} ws_1 & ws_2 & \cdots & ws_{N-2} \\ ws_2 & ws_3 & \cdots & ws_{N-1} \end{pmatrix} \quad (10)$$

$$Train_output = (ws_3 \quad ws_4 \quad \cdots \quad ws_N) \quad (11)$$

The Matlab codes of this step are listed in Code 2.

Code 2. Matlab codes to construct training samples.

```

n = length(Ws);

for i = 1:2

    Train_input(i,:) = Ws(i:n+i-3);

End

Train_output = Ws(3:n)

```

3. Normalize *Train_input* and *Train_output*. Since $\mathbf{X} \in [-1, 1]^n$ and $\mathbf{Y} \in [-1, 1]^m$, *Train_input* and *Train_output* need to be adjusted to satisfy this condition. First, the maximum and minimum values of each row in *Train_input* need to be calculated using the following functions:

$$Min_{input}^{(i)} = \min\{i^{th} \text{ row in } Train_input\} \quad (12)$$

$$Max_{input}^{(i)} = \max\{i^{th} \text{ row in } Train_input\} \quad (13)$$

Let $D_{input}^{(i)} = Max_{input}^{(i)} - Min_{input}^{(i)}$ and normalized $Train_input$ can be obtained by the following equation ($i = 1, 2$):

$$Train_inputm = \begin{pmatrix} \frac{ws_1 - Min_{input}^{(1)}}{D_{input}^{(1)}} & \frac{ws_2 - Min_{input}^{(1)}}{D_{input}^{(1)}} & \dots & \frac{ws_{N-2} - Min_{input}^{(1)}}{D_{input}^{(1)}} \\ \frac{ws_2 - Min_{input}^{(2)}}{D_{input}^{(2)}} & \frac{ws_3 - Min_{input}^{(2)}}{D_{input}^{(2)}} & \dots & \frac{ws_{N-1} - Min_{input}^{(2)}}{D_{input}^{(2)}} \end{pmatrix} \quad (14)$$

Similarly, normalized $Train_output$ can be obtained by the following equations:

$$Min_{output}^{(i)} = \min\{i^{th} \text{ row in } Train_output\}$$

$$Max_{output}^{(i)} = \max\{i^{th} \text{ row in } Train_output\}$$

$$D_{output}^{(i)} = Max_{output}^{(i)} - Min_{output}^{(i)}$$

$$Train_outputm = \begin{pmatrix} \frac{ws_3 - Min_{output}^{(1)}}{D_{output}^{(1)}} & \frac{ws_4 - Min_{output}^{(1)}}{D_{output}^{(1)}} & \dots & \frac{ws_N - Min_{output}^{(1)}}{D_{output}^{(1)}} \end{pmatrix} \quad (15)$$

The Matlab codes of this step are listed in Code 3.

Code 3. Matlab codes to normalize training samples.

```
[Train_inputm, inputs] = mapminmax(Train_input);
```

```
[Train_outputm, outputs] = mapminmax(Train_output);
```

4. Assign parameters and train the network. We assign that the network has three layers and the input layer has two nodes, the hidden layer has two nodes and the output layer has one node. Thus, according to Section 2.2.1, the optimized weights and biases of this

network can be calculated and denoted by $\theta_{ij}, \gamma_{ij}, w_{ij}, v_{ij}, i = 1, 2, j = 1, 2$, and $t = 1$. The Matlab codes of this step are listed in Code 4.

Code 4. Matlab codes to build and train BPNN.

```
Net = newff(Train_input, Train_output, 2);

Net = train(Net, Train_input, Train_output);
```

Net is the trained network and can be applied to forecast wind speed in the following steps:

5. Forecast ws_{N+1} . Based on the forecasting mode in step (1), we need to apply ws_{N-1} and ws_N to forecast ws_{N+1} , which means that the *Test_input* is $(ws_{N-1}, ws_N)^T$. Then, $D_{input}^{(i)}$, $Max_{input}^{(i)}$ and $Min_{input}^{(i)}$ obtained in step (3) will be applied to normalize *Test_input* and the normalized *Test_input* can be expressed using *Test_inputm* as follows:

$$Test_inputm = \begin{pmatrix} \frac{ws_{N-1} - Min_{input}^{(1)}}{D_{input}^{(1)}} \\ \frac{ws_N - Min_{input}^{(2)}}{D_{input}^{(2)}} \end{pmatrix} \quad (16)$$

Applying Eqs. (3) and (4) in Section 2.2.1, we can obtain the output of the network and denote it by $O \in \mathbb{R}$. Thus, the forecasting value of ws_{N+1} is $Ws_forecast = O \times D_{output}^{(1)} + Min_{output}^{(1)}$.

The Matlab codes of this step are listed in Code 5.

Code 5. Matlab codes to forecast using established BPNN (single step ahead)

```
Test_inputm = mapminmax('apply', Test_input, inputs);

O=sim(Net, Test_inputm);

Ws_forecast= mapminmax('reverse', O, outputs);
```

6. Forecast ws_{N+2} . Based on the forecasting mode in step (1), we need to apply ws_N and forecasting value of ws_{N+1} , $Ws_forecast$, to forecast ws_{N+2} , which means that the *Test_input2* is $(ws_N, Ws_forecast)^T$. Then, $D_{input}^{(i)}$, $Max_{input}^{(i)}$ and $Min_{input}^{(i)}$ obtained in step (3) will be

applied to normalize $Test_input2$ and the normalized $Test_input2$ can be expressed using $Test_inputm2$ as follows:

$$Test_inputm2 = \begin{pmatrix} \frac{ws_N - Min_{input}^{(1)}}{D_{input}^{(1)}} \\ \frac{Ws_forecast - Min_{input}^{(2)}}{D_{input}^{(2)}} \end{pmatrix} \quad (17)$$

Applying Eqs. (3) and (4) in Section 2.2.1, we can obtain the output of the network and denote it by $O_2 \in \mathbb{R}$. Therefore, the forecasting value of ws_{N+2} is $Ws_forecast2 = O_2 \times D_{output}^{(1)} + Min_{output}^{(1)}$.

The Matlab codes of this step are listed in Code 6.

Code 6. Matlab codes to forecast using established BPNN (two-step ahead).

```
Test_inputm2 = mapminmax('apply', Test_input2, inputs);

O2=sim(Net, Test_inputm2);

Ws_forecast2= mapminmax('reverse', O, outputs);
```

2.3. Support vector regression

2.3.1. Brief introduction of SVR

SVR is the most common application of support vector machines (SVMs) and constructs a hyperplane that separates examples with maximum margin, to categorize or forecast series [16]. Given the training data $\{(x_1, y_1), \dots, (x_l, y_l)\} \subset \mathbf{X} \times \mathbb{R}$, where $\mathbf{X} = \mathbb{R}^d$ denotes the space of input patterns, then the goal of SVR is to find a function $f(x)$ that is as flat as possible with at most ε deviation from the actually obtained targets y_i for all the training data. The simplest, linear formula for the output of a linear SVR is defined as [17]

$$f(x) = \langle w, x \rangle + b, \quad w \in X, b \in \mathbb{R} \quad (18)$$

where $\langle w, x \rangle$ denotes the dot product in X . One way to get the largest flatness is to minimize the w in Eq. (18) and this problem can be written as Eq. (19) by introducing slack variables ξ_i , ξ_i^* into a convex optimization problem function:

$$\begin{aligned}
& \text{minimize } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \\
& \text{subject to } \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon \\ \xi_i, \xi_i^* \geq 0 \end{cases}
\end{aligned} \tag{19}$$

The constant $C > 0$ determines the trade-off between the flatness of f and the amount up to which deviations larger than ε are tolerated.

To figure out the support vector regression function, the dual problem of Eq. (19) is defined in Eq. (20) by constructing a Lagrange function from the objective function:

$$\begin{aligned}
& \text{maximize } \begin{cases} \frac{1}{2} \sum_{i,j=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle x_i, x_j \rangle \\ -\varepsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) + \sum_{i=1}^l y_i (\alpha_i - \alpha_i^*) \end{cases} \\
& \text{subject to } \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \text{ and } \alpha_i, \alpha_i^* \in [0, C]
\end{aligned} \tag{20}$$

With the solution of (α, α^*) from Eq. (20), the support vector regression function can be written as follows:

$$f(x) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \langle x_i, x \rangle + b \tag{21}$$

When it comes to the nonlinear regression problem, the training patterns x_i can be mapped into a high-dimensional space, where the nonlinear regression problem is transformed into a linear one. The expansion of Eq. (21) is defined as Eq. (22):

$$f(x) = \sum_{i=1}^N (\alpha_i^* - \alpha_i) k(x_i, x; g) + b \tag{22}$$

where α_i^* and α_i are Lagrange multipliers and $k(x_i, x; g)$ is the kernel function, in which g is a parameter and generally set as $1/d$.

2.3.2. Implement for wind speed forecasting

We use the libsvm package (Version 3.17) to implement forecasting task of wind speed using SVR. There are six steps and steps (1)–(3) are same as in steps (1)–(3) in Section 2.2.2. Thus, we start to introduce from step (4).

(4) Assign parameters and train the SVR model. We assign that C is 4 and g is 0.5. Thus, according to Section 2.3.1, the optimized solution (α, α^*) can be calculated. The Matlab codes of this step are listed in Code 7.

Code 7. Matlab codes to establish and train SVR.

```
Model = svmtrain(Train_output', Train_input', '-C 4 -g 0.5');
```

Model is the trained SVR model and can be applied to forecast wind speed by the following steps:

(5) Forecast ws_{N+1} . Based on the forecasting mode in step (1), we need to apply ws_{N-1} and ws_N to forecast ws_{N+1} , which means that the $Test_input$ is $(ws_{N-1}, ws_N)^T$. Then, $D_{input}^{(i)}$, $Max_{input}^{(i)}$ and $Min_{input}^{(i)}$ obtained in step (3) will be applied to normalize $Test_input$ and the normalized $Test_input$ can be expressed using $Test_inputm$ as follows:

$$Test_inputm = \begin{pmatrix} \frac{ws_{N-1} - Min_{input}^{(1)}}{D_{input}^{(1)}} \\ \frac{ws_N - Min_{input}^{(2)}}{D_{input}^{(2)}} \end{pmatrix} \quad (23)$$

Applying Eqs. (3) and (4) in Section 2.2.1, we can obtain the output of the network and denote it by $O \in \mathbb{R}$. Thus, the forecasting value of ws_{N+1} is $Ws_forecast = O \times D_{output}^{(1)} + Min_{output}^{(1)}$. The Matlab codes of this step are listed in Code 8.

Code 8. Matlab codes to forecast using established SVR (single step ahead).

```
Test_inputm = mapminmax('apply', Test_input, inputs);
```

```
O=svmpredict(0, Test_inputm',Model);
```

```
Ws_forecast= mapminmax('reverse', O, outputs);
```

(6) Forecast ws_{N+2} . Based on the forecasting mode in step (1), we need to apply ws_N and the forecasting value of ws_{N+1} , $Ws_forecast$, to forecast ws_{N+2} , which means that the $Test_input2$ is $(ws_N, Ws_forecast)^T$. Then, $D_{input}^{(i)}$, $Max_{input}^{(i)}$ and $Min_{input}^{(i)}$ obtained in step (3) will be applied to normalize $Test_input2$ and the normalized $Test_input2$ can be expressed using $Test_inputm2$ as follows:

$$Test_inputm2 = \begin{pmatrix} \frac{ws_N - Min_{input}^{(1)}}{D_{input}^{(1)}} \\ \frac{Ws_forecast - Min_{input}^{(2)}}{D_{input}^{(2)}} \end{pmatrix} \quad (24)$$

Applying Eqs. (3) and (4) in Section 2.2.1, we can obtain the output of the network and denote it by $O_2 \in \mathbb{R}$. Therefore, the forecasting value of ws_{N+2} is $Ws_forecast2 = O_2 \times D_{output}^{(1)} + Min_{output}^{(1)}$. The Matlab codes of this step are listed in Code 9.

Code 9. Matlab codes to forecast using established SVR (two-step ahead).

```
Test_inputm2 = mapminmax('apply', Test_input2, inputs);

O2=svmpredict(0,Test_inputm2',Model);

Ws_forecast2= mapminmax('reverse', O, outputs);
```

2.4. Extreme learning machine

2.4.1. Brief introduction of ELM

Huang et al. [18] investigated the ELM, which is an effective and efficient learning algorithm. The ELM aims to randomly initialize the weights and biases of SLFN and then to explicitly calculate the hidden layer output matrix and hence the output weights. Because of the nature of non-adjusted weights and bias, the network can be established using a very low computational cost [19]. Then, the ELM with l hidden neurons and transfer function $\varphi(\cdot)$ can approximate the N samples with zero error as

$$\sum_{j=1}^l \beta_j \varphi(\mathbf{w}_j \mathbf{x}_k + b_j) = y_k, k = 1, 2, \dots, N \quad (25)$$

which can be written as $HB=Y$, with

$$H(\mathbf{w}_1, \dots, \mathbf{w}_l, b_1, \dots, b_l, \mathbf{x}_1, \dots, \mathbf{x}_N) = \begin{pmatrix} \varphi(\mathbf{w}_1 \mathbf{x}_1 + b_1) & \cdots & \varphi(\mathbf{w}_l \mathbf{x}_1 + b_l) \\ \vdots & \ddots & \vdots \\ \varphi(\mathbf{w}_1 \mathbf{x}_N + b_1) & \cdots & \varphi(\mathbf{w}_l \mathbf{x}_N + b_l) \end{pmatrix} \quad (26)$$

where w_i is the weight vector connecting the i th hidden neuron and the input nodes, β_i is the weight vector connecting the i th hidden neuron and the output neurons, and b_i is the threshold of the i th hidden neuron, and $B = (\beta_1^T \dots \beta_l^T)^T$, $Y = (y_1^T \dots y_N^T)^T$

Then, the output weights B can be calculated from the hidden layer output matrix H and the target values Y as $B^{\wedge} = H^{\dagger}Y$, where H^{\dagger} is the Moore-Penrose generalized inverse of the matrix H .

2.4.2. Implement for wind speed forecasting

We use Matlab to compile ELM model and provide two *.m files, which are Elmtrain.m and Elmpredict.m, in Appendix. Elmtrain function is similar to train function for BPNN and svmtrain function for SVR and Elmpredict function is similar to sim function for BONN and svmpredict function for SVR. In detail, there are six steps for forecasting wind speed using ELM model and steps (1)–(3) are same as in steps (1)(3) in Section 2.2.2. Thus, we start to introduce them from step (4).

(4) Assign parameters and train the ELM model. We design that the input layer of ELM model has two nodes, and the hidden layer of ELM has two nodes, and the output layer of ELM model has one node. Thus, according to Section 2.4.1, the optimized B can be calculated. The Matlab codes of this step are listed in Code 10.

Code 10. Matlab codes to establish and train ELM.

```
[W,b,B,TF,TYPE] = elmtrain(Train_input, Train_output, 2);
```

W is the weight matrix of the input layer and hidden layer, and b is the bias vector of input layer and hidden layer, and B is B^{\wedge} in Section 2.4.1, and TF is the transfer function $\varphi(\cdot)$, and $TYPE$ is model's functions (classification or regression). We select sigmoid function as the transfer function $\varphi(\cdot)$ of ELM, which is same as that of BPNN.

(5) Forecast ws_{N+1} . Based on the forecasting mode in step (1), we need to apply ws_{N-1} and ws_N to forecast ws_{N+1} , which means that the $Test_input$ is $(ws_{N-1}, ws_N)^T$. Then, $D_{input}^{(i)}$, $Max_{input}^{(i)}$ and $Min_{input}^{(i)}$ obtained in step (3) will be applied to normalize $Test_input$ and the normalized $Test_input$ can be expressed using $Test_inputm$ as follows:

$$Test_inputm = \begin{pmatrix} \frac{ws_{N-1} - Min_{input}^{(1)}}{D_{input}^{(1)}} \\ \frac{ws_N - Min_{input}^{(2)}}{D_{input}^{(2)}} \end{pmatrix} \quad (27)$$

Applying Eqs. (3) and (4) in Section 2.2.1, we can obtain the output of the network and denote it by $O \in \mathbb{R}$. Thus, the forecasting value of ws_{N+1} is $Ws_forecast = O \times D_{output}^{(1)} + Min_{output}^{(1)}$. The Matlab codes of this step are listed in Code 11.

Code 11. Matlab codes to forecast using established ELM (single step ahead).

```
Test_inputm = mapminmax('apply', Test_input, inputs);
```

```
O=elmpredict(Test_inputm, W,b,B,TF,TYPE);
```

```
Ws_forecast= mapminmax('reverse', O, outputs);
```

(6) Forecast ws_{N+2} . Based on the forecasting mode in step (1), we need to apply ws_N and forecasting value of ws_{N+1} , $Ws_forecast$, to forecast ws_{N+2} , which means that the $Test_input2$ is $(ws_N, Ws_forecast)^T$. Then, $D_{input}^{(i)}$, $Max_{input}^{(i)}$ and $Min_{input}^{(i)}$ obtained in step (3) will be applied to normalize $Test_input2$ and the normalized $Test_input2$ can be expressed using $Test_inputm2$ as follows:

$$Test_inputm2 = \begin{pmatrix} \frac{ws_N - Min_{input}^{(1)}}{D_{input}^{(1)}} \\ \frac{Ws_forecast - Min_{input}^{(2)}}{D_{input}^{(2)}} \end{pmatrix} \quad (28)$$

Applying Eqs. (3) and (4) in Section 2.2.1, we can obtain the output of the network and denote it by $O_2 \in \mathbb{R}$. Therefore, the forecasting value of ws_{N+2} is $Ws_forecast2 = O_2 \times D_{output}^{(1)} + Min_{output}^{(1)}$. The Matlab codes of this step are listed in Code 12.

Code 12. Matlab codes to forecast using established ELM (two-step ahead).

```
Test_inputm2 = mapminmax('apply', Test_input2, inputs);
```

```
O2=svmpredict(Test_inputm2, W,b,B,TF,TYPE);
```

```
Ws_forecast2= mapminmax('reverse', O, outputs);
```

2.5. Adaptive network-based fuzzy inference system

2.5.1. Brief introduction of ANFIS

ANFIS is introduced to compensate for the disability of conventional mathematical tools to address uncertain systems, such as human knowledge and reasoning processes. There are two contributions of FIFs restructured: proposing a standard method for transforming ill-defined factors into identifiable rules of FIS and using an adaptive network to tune the membership functions. We assume that there are two fuzzy if-then rules contained in the system, two inputs (x and y) and one output (z), and the processes of ANFIS are described in **Figure 1** [20–21].

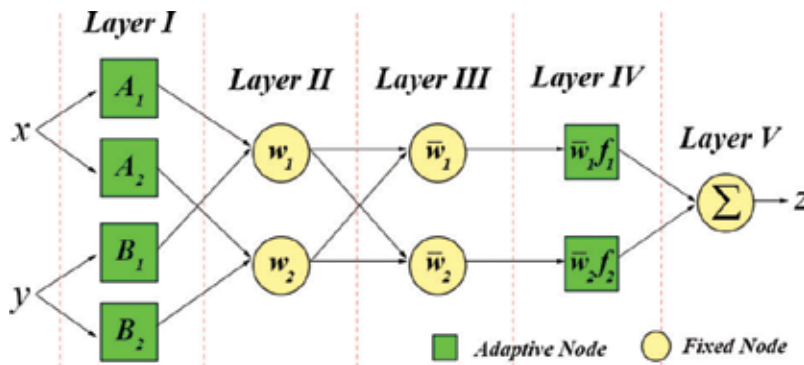


Figure 1. Five processes included in this figure. In an ANFIS architecture, circles are fixed nodes without parameters and squares represent adaptive nodes whose parameters are determined by training data and a gradient-based learning procedure.

Layer I: Mapping a certain input x to a fuzzy set $O_i^{(1)}$ for every node i by the member functions μ_{A_i} , which is usually bell-shaped with a parameter set $\{a_i, b_i, c_i\}$, as is y

$$O_i^{(1)} = \mu_{A_i}(x), \text{ where } \mu_{A_i}(x) = \frac{1}{1 + \left[\left(\frac{x - c_i}{a_i} \right)^{b_i} \right]} \text{ or } \mu_{A_i}(x) = e^{-\left(\frac{x - c_i}{a_i} \right)^2} \quad (29)$$

Layer II: In this layer, each circle node performs the connection “AND” and multiplies inputs, as well as sends the product out:

$$O_i^{(2)} = \omega_i = \mu_{A_i}(x) \times \mu_{B_i}(y) \quad (30)$$

Layer III: Every circle node in this layer calculates a normalized firing strength, namely a ratio of the rule’s firing strength to the sum of all rules’ firing strengths:

$$O_i^{(3)} = \bar{\omega}_i = \frac{\omega_i}{\sum \omega_i} \quad (31)$$

Layer IV: Assume the rules of this system are as follows:

Rule 1: If x is A_1 and y is B_1 , then $f_1 = p_1x + q_1y + r_1$

Rule 2: If x is A_2 and y is B_2 , then $f_2 = p_2x + q_2y + r_2$

Then, the outputs of the adaptive nodes in this layer are computed by

$$O_i^{(4)} = \bar{\omega}_i f_i = \bar{\omega}_i (p_1x + q_1y + r_1) \quad (32)$$

Layer V: The overall output is the weighted average of all incoming signals:

$$O_i^{(5)} = \sum_i \bar{\omega}_i f_i = \frac{\sum_i \omega_i f_i}{\sum_i \omega_i} \quad (33)$$

Particularly, in this case,

$$O_i^{(5)} = \sum_{i=1}^2 \bar{\omega}_i f_i = \sum_{i=1}^2 (\bar{\omega}_i x) p_i + (\bar{\omega}_i y) q_i + \bar{\omega}_i r_i \quad (34)$$

2.5.2. Implement for wind speed forecasting

We use the `genfis3` function and `ANFIS` function in Matlab to implement forecasting task of wind speed using ANFIS. There are six steps and steps (1)–(3) are same as in steps (1)–(3) in Section 2.2.2. Thus, we start to introduce from step (4).

(4) Assign parameters and train the ANFIS model. We set the number of iteration of ANFIS as 100. Thus, the optimized parameters can be calculated. The Matlab codes of this step are listed in Code 13.

Code 13. Matlab codes to establish and train ANFIS.

```
fismat = genfis3(Train_input', Train_output');
out_fis1 = anfis([Train_input' Train_output'], fismat, 100);
```

The `out_fis1` is the trained ANFIS and can be applied to forecast wind speed.

(5) Forecast ws_{N+1} . Based on the forecasting mode in step (1), we need to apply ws_{N-1} and ws_N to forecast ws_{N+1} , which means that the $Test_input$ is $(ws_{N-1}, ws_N)^T$. Then, $D_{input}^{(i)}$, $Max_{input}^{(i)}$ and $Min_{input}^{(i)}$ obtained in step (3) will be applied to normalize $Test_input$ and the normalized $Test_input$ can be expressed using $Test_inputm$ as follows:

$$Test_inputm = \begin{pmatrix} \frac{ws_{N-1} - Min_{input}^{(1)}}{D_{input}^{(1)}} \\ \frac{ws_N - Min_{input}^{(2)}}{D_{input}^{(2)}} \end{pmatrix} \quad (35)$$

Applying Eqs. (3) and (4) in Section 2.2.1, we can obtain the output of the network and denote it by $O \in \mathbb{R}$. Thus, the forecasting value of ws_{N+1} is $Ws_forecast = O \times D_{output}^{(1)} + Min_{output}^{(1)}$. The Matlab codes of this step are listed in Code 14.

Code 14. Matlab codes to forecast using established ANFIS (single step ahead).

```
Test_inputm = mapminmax('apply', Test_input, inputs);
```

```
O=evalfis(Test_inputm', out_fis1);
```

```
Ws_forecast= mapminmax('reverse', O, outputs);
```

(6) Forecast ws_{N+2} . Based on the forecasting mode in step (1), we need to apply ws_N and forecasting value of ws_{N+1} $Ws_forecast$, to forecast ws_{N+2} , which means that the $Test_input2$ is $(ws_N, Ws_forecast)^T$. Then, $D_{input}^{(i)}$, $Max_{input}^{(i)}$ and $Min_{input}^{(i)}$ obtained in step (3) will be applied to normalize $Test_input2$ and the normalized $Test_input2$ can be expressed using $Test_inputm2$ as follows:

$$Test_inputm2 = \begin{pmatrix} \frac{ws_N - Min_{input}^{(1)}}{D_{input}^{(1)}} \\ \frac{Ws_forecast - Min_{input}^{(2)}}{D_{input}^{(2)}} \end{pmatrix} \quad (36)$$

Applying Eqs. (3) and (4) in Section 2.2.1, we can obtain the output of the network and denote it by $O_2 \in \mathbb{R}$. Therefore, the forecasting value of ws_{N+2} is $Ws_forecast2 = O_2 \times D_{output}^{(1)} + Min_{output}^{(1)}$. The Matlab codes of this step are listed in Code 15.

Code 15. Matlab codes to forecast using established ANFIS (two-step ahead).

```
Test_inputm2 = mapminmax('apply', Test_input2, inputs);
```

```
O2=evalfis(Test_inputm2', out_fis1);
```

```
Ws_forecast2= mapminmax('reverse', O, outputs);
```

3. Data collection

We select three types of wind data to train and test forecasting models, which are 10-minute wind speed data in wind farm A (this database is denoted by WFD1), 15-minute wind speed data in wind farm B (this database is denoted by WFD2), and 15-minute wind speed data in wind farm C (this database is denoted by WFD3), and wind farms B and C are in the same city in China.

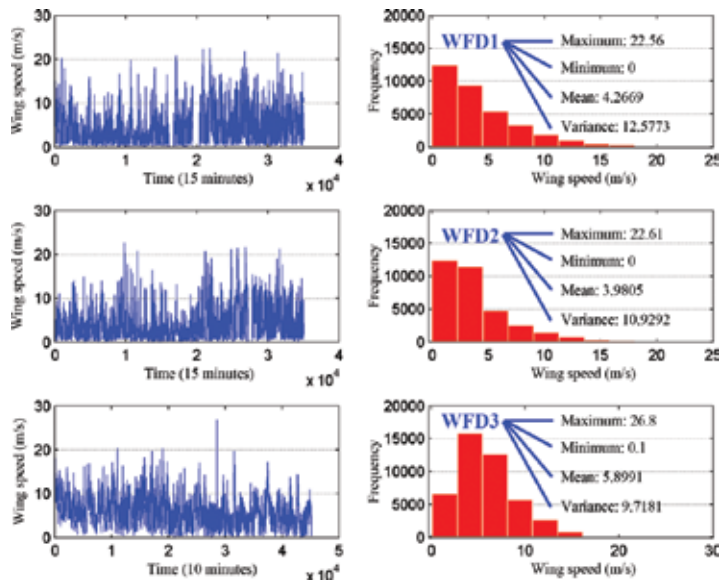


Figure 2. Descriptive information and histograms of three wind farm databases.

In this section, the descriptive information of each database will be provided and we will test the similarity between WFD2 and WFD3 using Friedman test. In the beginning, **Figure 2** shows the three databases and their histograms.

From **Figure 2**, this is apparent that WFD1 and WFD2 have similar maximum and minimum values because they are in the same city. However, WFD1 and WFD2 have different mean and

variance values. Thus, we use Friedman test to figure out whether both of two time series are significantly different, which is shown in **Table 1**. In **Table 1**, the p -value of Friedman test is very small ($2.46\text{e-}11$), which indicates that WFD1 and WFD2 have a significant difference.

Source	SS	df	MS	Chi-squared	Prob>Chi-squared
Columns	73.3	1	73.2825	44.57	2.46e-11
Interaction	61568.7	17,085	3.6037		
Error	22,649	34,172	0.6628		
Total	84,291	68,343			

Table 1. The result of Friedman test between WFD1 and WFD2.

For WFD3, its mean value is significantly different from others and the histogram of WFD3 is also different from those of other databases. From the description of these three databases, it can be seen that they are very different, which lays a strong foundation for the comparison study.

4. Simulation and results

For these different databases, we will test 1-h-ahead, 4-h-ahead, 1-h average, 4-h average, and 1-day average wind speed forecasting effectiveness by using five methods separately, which is to say that we conduct five experiments of different time scales or time horizons in this comparison study. To test the overall forecasting effectiveness of each model, the testing period is 30 weeks in 2014 and we apply three criteria to evaluate forecasting error, which are the mean absolute percent error (MAPE), root-mean-squared error (RMSE), and mean absolute error (MAE). The formulas of three criteria are expressed as follows:

$$e_t = \hat{y}_t - y_t \quad (37)$$

$$MAPE = \frac{1}{N} \sum_{t=1}^N \frac{|e_t|}{y_t} \times 100\% \quad (38)$$

$$RMSE = \frac{1}{N} \sqrt{\sum_{t=1}^N e_t^2} \quad (39)$$

$$MAE = \frac{1}{N} \sum_{t=1}^N |e_t| \quad (40)$$

where t represents the time t , y_t is the actual value in time t , \hat{y}_t is the forecasting value in time t , and N is the length of testing period.

4.1. Experiment I: 1-h-ahead wind speed forecasting

In this experiment, we will test the 1-h-ahead forecasting effectiveness of five models. It should be pointed that 1-h-ahead prediction means a four-step forecast for WFD1 and WFD2 and a means six-step forecast for WFD3 because WFD3 is constructed by 10-min wind speeds. The main results are shown in **Table 2**, which contains MAPE, RMSE, and MAE of each model when forecasting 1-h-ahead wind speed.

Wind farms	Criteria	ANFIS	BPNN	ELM	SVM	ARMA
WFD1	MAE	1.4193	0.8066	0.8335	0.9471	1.4293
	MAPE	43.67%	19.25%	19.51%	30.74%	30.96%
	RMSE	1.5092	0.9083	0.9375	1.0521	1.5574
WFD2	MAE	1.8110	0.7651	0.7649	0.8590	2.9846
	MAPE	49.82%	20.21%	19.76%	28.11%	83.01%
	RMSE	1.8837	0.8611	0.8607	0.9550	3.0937
WFD3	MAE	1.2618	0.9781	0.8571	0.9055	3.0764
	MAPE	25.22%	20.26%	16.29%	19.05%	64.92%
	RMSE	1.3735	1.1019	0.9797	1.0290	3.1724

Table 2. The forecasting results of Experiment I.

From **Table 2**, it is apparent that BPNN has the best performance among these five models in WFD1 and ELM outperforms others in WFD2 and WFD3 by three criteria. By contrast, the forecasting accuracy of ANFSI and ARMA is poor and even cannot be accepted in some wind farms, such as the performance of ARMA in WFD1 and WFD2 and the performance of ANFIS in WFD1 and WFD2. These results show that each wind farm has the corresponding best forecasting model and there is no single model that can perform best in every wind farm by three criteria.

4.2. Experiment II: 4-h-ahead wind speed forecasting

Experiment II shows the 4-h-ahead forecasting effectiveness of wind speed, and the 4-h-ahead forecasting is equal to a 16-step forecast for WFD1 and WFD2 and is a 24-step forecast for WFD3. **Table 3** descripts forecasting results of five models in three wind farms.

In **Table 3**, BPNN is the best model in WFD1 and WFD2 by three criteria and outperforms other models in three databases by MAE and RMSE. ANFIS has the lowest MAPE in WFD3 among these models but has high MAPE, MAE, and RMSE in WFD1 and WFD2. It is similar to Experiment I that ARMA does not have a decent forecasting results in three wind farms.

These results indicate that model's forecasting effectiveness varies a lot when the wind farm changes (such as the effectiveness of ANFIS).

Wind farms	Criteria	ANFIS	BPNN	ELM	SVM	ARMA
WFD1	MAE	1.9441	1.4385	1.4725	1.6147	2.4799
	MAPE	54.41%	35.36%	36.67%	52.28%	55.59%
	RMSE	2.1531	1.6664	1.7062	1.8376	2.7478
WFD2	MAE	1.9937	1.3251	1.3492	1.4246	2.8887
	MAPE	54.96%	34.71%	36.36%	45.85%	78.64%
	RMSE	2.1770	1.5440	1.5647	1.6301	3.1138
WFD3	MAE	1.3250	1.3176	1.3575	1.3951	2.6248
	MAPE	26.04%	26.80%	27.20%	30.12%	53.30%
	RMSE	1.5494	1.5347	1.5924	1.6122	2.8495

Table 3. The forecasting results of Experiment II.

4.3. Experiment III: 1-h average wind speed forecasting

In Experiment III, we will forecast 1-h average wind speed in three wind farms using ANFIS, BPNN, ELM, SVM, and ARMA, and 1-h average wind speed forecasting is a one-step forecast for three wind farms.³ **Table 4** shows the forecasting effectiveness of each model in this experiment.

Wind farms	Criteria	ANFIS	BPNN	ELM	SVM	ARMA
WFD1	MAE	1.5664	0.8586	0.8315	0.9069	1.2883
	MAPE	44.82%	20.15%	19.33%	24.92%	27.58%
	RMSE	2.0981	1.2376	1.1882	1.2805	1.8759
WFD2	MAE	1.7780	0.7924	0.7607	0.7919	2.5731
	MAPE	47.44%	20.30%	19.35%	21.92%	80.67%
	RMSE	2.3788	1.1838	1.1298	1.1712	3.3596
WFD3	MAE	0.8873	0.7964	0.7943	0.7914	1.1274
	MAPE	16.90%	15.07%	14.95%	14.91%	20.75%
	RMSE	1.1568	1.0568	1.0548	1.0539	1.4833

Table 4. The forecasting results of Experiment III.

³ It should be pointed that every day has 24 1-h average wind speed.

From **Table 4**, ELM and SVM have decent forecasting accuracy in WFD3, but ELM is better than SVM and other models in WFD1 and WFD2. ANFIS has a good performance in WFD3 and cannot forecast 1-h average wind speed with a reasonable accuracy. Similarly for ARMA, it works well in WFD1 and WFD3 but has a high MAPE in WFD2.

4.4. Experiment IV: 4-h average wind speed forecasting

It is similar to Experiment III, but this experiment shows the one-step-ahead wind speed forecasting effectiveness of each model. **Table 5** demonstrates the results of five models obtained in three wind farms.

Wind farms	Criteria	ANFIS	BPNN	ELM	SVM	ARMA
WFD1	MAE	2.0516	1.6435	1.6209	1.5747	2.1725
	MAPE	55.75%	41.64%	38.76%	36.76%	48.81%
	RMSE	2.6411	2.1411	2.1478	2.1241	2.8977
WFD2	MAE	2.1538	1.4282	1.3489	1.3422	2.4222
	MAPE	55.53%	36.02%	34.27%	30.86%	74.71%
	RMSE	2.8318	1.9498	1.8597	1.8933	3.1035
WFD3	MAE	1.2564	1.2015	1.1483	1.1689	1.6348
	MAPE	23.37%	22.33%	21.13%	21.19%	29.41%
	RMSE	1.5951	1.5249	1.4856	1.5074	2.0953

Table 5. The forecasting results of Experiment IV.

The analyzing results in **Table 5** are quite different from that in **Table 4**. Specifically, SVM model has better performance in WFD1 and WFD2 than other models, and ELM is the best forecasting model in WFD3. For each model, wind speed in WFD3 can be forecasted more accurately than that in WFD1 and WFD2, and AI-based models (ANFIS, BPNN, ELM, and SVM) outperform ARMA in WFD3.

4.5. Experiment V: 1-day average wind speed forecasting

In this experiment, we will test the one-step-ahead forecasting effectiveness based on 1-day average wind speed database. Similar to other experiments, **Table 6** demonstrates each model's forecasting effectiveness.

Table 6 shows that SVM has the best forecasting performance on three criteria in three wind farms. By contrast, ARMA and ANFIS have the higher MAPE, MAE, and RMSE than other models.

Wind farms	Criteria	ANFIS	BPNN	ELM	SVM	ARMA
WFD1	MAE	2.1001	1.9069	1.8377	1.7675	2.5753
	MAPE	52.78%	42.99%	45.70%	36.81%	60.59%
	RMSE	2.5729	2.3784	2.3053	2.2692	3.1655
WFD2	MAE	1.9384	1.9710	1.9038	1.7653	1.9271
	MAPE	51.81%	47.82%	48.19%	38.04%	52.61%
	RMSE	2.4685	2.4758	2.3848	2.3573	2.4679
WFD3	MAE	1.9288	1.8385	1.5886	1.5487	1.8763
	MAPE	32.30%	30.40%	26.39%	24.79%	30.47%
	RMSE	2.4229	2.3061	1.9152	1.9159	2.2568

Table 6. The forecasting results of Experiment V.

5. Conclusion

The inherently intermittent nature and stochastic nonstationarity of wind sources bring great levels of uncertainty to system operators and are an urgent problem in the wind-forecasting field. This chapter introduces some basic forecasting approaches and corresponding procedures to forecast wind speed (detailed codes are attached). After simulating these methods (the testing period is 30 weeks) in MATLAB based on three databases, conclusions can be drawn as follows:

1. None of ANFIS, BPNN, ELM, SVM, and ARMA has the best forecasting performance in all experiments.
2. Based on the experimental results of different forecasting types (1-h-ahead forecasting, 4-h-ahead forecasting, 1-h-average-ahead forecasting, 4-h-average-ahead forecasting, and 1-day-average-ahead forecasting), the best model varies in time scales and time horizons.
3. The forecasting effectiveness differs a lot from database to database (in Experiment II, ANFIS is the best model of which MAPE is 26.04% in WFD3 but has a high MAPE, 54.41%, in WFD1).
4. AI-based models are more suitable for wind speed forecast than ARMA.

According to these conclusions, it is obvious that wind speed forecasting models or systems need to be built up under specific conditions in wind farms and model selection in wind speed forecasting plays a significant role to improve the accuracy.

Acknowledgements

The work was supported by the National Natural Science Foundation of China (grant no. 71573034).

Appendices

In the appendix, we will provide two functions which are coded by Matlab to implement training process (elmtrain.m) and predicting process (elpredict.m) of ELM. The elmtrain.m is expressed as follows:

Code A1. Matlab function file to establish and train ELM.

```
function [IW,B,LW,TF,TYPE] = elmtrain(P,T,N,TF,TYPE)
```

```
% ELMTRAIN Create and Train a Extreme Learning Machine
```

```
% Syntax
```

```
% [IW,B,LW,TF,TYPE] = elmtrain(P,T,N,TF,TYPE)
```

```
% Description
```

```
% Input
```

```
% P—Input Matrix of Training Set ( $R \times Q$ )
```

```
% T—Output Matrix of Training Set ( $S \times Q$ )
```

```
% N—Number of Hidden Neurons (default = Q)
```

```
% TF—Transfer Function:
```

```
% 'sig' for Sigmoidal function (default)
```

```
% 'sin' for Sine function
```

```
% 'hardlim' for Hardlim function
```

```
% TYPE—Regression (0,default) or Classification (1)
```

```
% Output
```

```
% IW—Input Weight Matrix ( $N \times R$ )
```

```
% B—Bias Matrix ( $N \times 1$ )
```

```
% LW—Layer Weight Matrix ( $N \times S$ )
```

```
% Example
```

```
% Regression:
```

```
% [IW,B,LW,TF,TYPE] = elmtrain(P,T,20,'sig',0)

% Y = elmtrain(P,IW,B,LW,TF,TYPE)

% Classification

% [IW,B,LW,TF,TYPE] = elmtrain(P,T,20,'sig',1)

% Y = elmtrain(P,IW,B,LW,TF,TYPE)

% See also ELMPREDICT

% Yu Lei,11-7-2010

% Copyright www.matlabsky.com

% $Revision:1.0 $

if nargin < 2

    error('ELM:Arguments','Not enough input arguments.');
```

```

[R,Q] = size(P);

if TYPE == 1

    T = ind2vec(T);

end

[S,Q] = size(T);

% Randomly Generate the Input Weight Matrix

IW = rand(N,R) * 2-1;

% Randomly Generate the Bias Matrix

B = rand(N,1);

BiasMatrix = repmat(B,1,Q);

% Calculate the Layer Output Matrix H

tempH = IW * P + BiasMatrix;

switch TF

    case 'sig'

        H = 1 ./ (1 + exp(-tempH));

    case 'sin'

        H = sin(tempH);

    case 'hardlim'

        H = hardlim(tempH);

end

% Calculate the Output Weight Matrix

% find(isnan(T) == 1)

LW = pinv(H) * T';

```

The `elmpredict.m` is expressed as follows:

Code A2. Matlab function file to forecast using establish ELM.

```
function Y = elmpredict(P,IW,B,LW,TF,TYPE)

% ELM PREDICT Simulate an Extreme Learning Machine

% Syntax

% Y = elmtrain(P,IW,B,LW,TF,TYPE)

% Description

% Input

% P—Input Matrix of Training Set (R*Q)

% IW—Input Weight Matrix (N*R)

% B—Bias Matrix (N*1)

% LW—Layer Weight Matrix (N*S)

% TF—Transfer Function:

% 'sig' for Sigmoidal function (default)

% 'sin' for Sine function

% 'hardlim' for Hardlim function

% TYPE—Regression (0,default) or Classification (1)

% Output

% Y—Simulate Output Matrix (S*Q)

% Example

% Regression:

% [IW,B,LW,TF,TYPE] = elmtrain(P,T,20,'sig',0)

% Y = elmtrain(P,IW,B,LW,TF,TYPE)
```

```

% Classification

% [IW,B,LW,TF,TYPE] = elmtrain(P,T,20,®sig',1)

% Y = elmtrain(P,IW,B,LW,TF,TYPE)

% See also ELMTRAIN

% Yu Lei,11-7-2010

% Copyright www.matlabsky.com

% $Revision:1.0 $

if nargin < 6

    error('ELM:Arguments','Not enough input arguments.');
```

end

```

% Calculate the Layer Output Matrix H

Q = size(P,2);

BiasMatrix = repmat(B,1,Q);

tempH = IW * P + BiasMatrix;

switch TF

    case 'sig'

        H = 1 ./ (1 + exp(-tempH));

    case 'sin'

        H = sin(tempH);

    case 'hardlim'

        H = hardlim(tempH);

end

% Calculate the Simulate Output

Y = (H' * LW)';
```

```
if TYPE == 1

    temp_Y = zeros(size(Y));

    for i = 1:size(Y,2)

        [max_Y,index] = max(Y(:,i));

        temp_Y(index,i) = 1;

    end

    Y = vec2ind(temp_Y);

end
```

Author details

Ping Jiang, Feng Liu* and Yiliao Song

*Address all correspondence to: liuf13@lzu.edu.cn or feng.liu.1990@ieee.org

School of Statistics, Dongbei University of Finance and Economics, Dalian, Liaoning, China.

References

- [1] I.J. Ramirez-Rosado, L.A. Fernandez-Jimenez, C. Monteiro, J. Sousa, and R. Bessa. Comparison of two new short-term wind-power forecasting systems. *Renewable Energy*. 2009;34(7):1848–1854. DOI: 10.1016/j.renene.2008.11.014
- [2] G. Sideratos, N.D. Hatziargyriou. Wind power forecasting focused on extreme power system events. *IEEE Transactions on Sustainable Energy*. 2012;3(3):445–454. DOI: 10.1109/TSTE.2012.2189442
- [3] P. Zhao, J. Wang, J. Xia, Y. Dai, Y. Sheng, J. Yue. Performance evaluation and accuracy enhancement of a day ahead wind power forecasting system in China. *Renewable Energy*. 2012;43(1):234–241. DOI: 10.1016/j.renene.2011.11.051
- [4] K. Bhaskar, S.N. Singh. A WNN-assisted wind power forecasting using feed-forward neural network. *IEEE Transactions on Sustainable Energy*. 2012;3(2):306–315. DOI: 10.1109/TSTE.2011.2182215

- [5] L. Han, C.E. Romero, Z. Yao. Wind power forecasting based on principle component phase space reconstruction. *Renewable Energy*. 2015;81(1):737–744. DOI: 10.1016/j.renene.2015.03.037
- [6] L. Silva. A feature engineering approach to wind power forecasting: GEFCom 2012. *International Journal of Forecasting*. 2014;30(2):395–401. DOI: 10.1016/j.ijforecast.2013.07.007
- [7] W.P. Mahoney, et al. A wind power forecasting system to optimize grid integration. *IEEE Transactions on Sustainable Energy*. 2012;3(4):670–682. DOI: 10.1109/TSTE.2012.2201758
- [8] C. Croonenbroeck, C.M. Dahl. Accurate medium-term wind power forecasting in a censored classification framework. *Energy*. 2014;73(1):221–232. DOI: 10.1016/j.energy.2014.06.013
- [9] N.B. Karayiannis, M.M. Randolph-Gips. On the construction and training of reformulated radial basis function neural networks. *IEEE Transactions on Neural Networks*. 2003;14(4):835–846. DOI: 10.1109/TNN.2003.813841
- [10] P. Kou, F. Gao, X. Guan. Sparse online warped Gaussian process for wind power probabilistic forecasting. *Applied Energy*. 2013;108(1):410–428. DOI: 10.1016/j.apenergy.2013.03.038
- [11] R.J. Bessa, V. Miranda, A. Botterud, J. Wang, E.M. Constantinescu. Time adaptive conditional kernel density estimation for wind power forecasting. *IEEE Transactions on Sustainable Energy*. 2012;3(4):660–669. DOI: 10.1109/TSTE.2012.2200302
- [12] A. Carpinone, M. Giorgio, R. Langella, A. Testa. Markov chain modeling for very short term wind power forecasting. *Electric Power Systems Research*. 2015;122(1):152–158. DOI: 10.1016/j.epsr.2014.12.025
- [13] P. Pinson. Very short term probabilistic forecasting of wind power with generalized logit-normal distributions. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*. 2012;61(4):555–576. DOI: 10.1111/j.1467-9876.2011.01026.x
- [14] E. Erdem, J. Shi. ARMA based approaches for forecasting the tuple of wind speed and direction. *Applied Energy*. 2011;88:1045–1044. DOI: 10.1016/j.apenergy.2010.10.031
- [15] L. Xiao, J. Wang, R. Hou, J. Wu. A combined model based on data pre-analysis and weight coefficients optimization for electrical load forecasting. *Energy*. 2015;82:524–549. DOI: 10.1016/j.energy.2015.01.063
- [16] Z. Guo, J. Zhao, W. Zhang, J. Wang. A corrected hybrid approach for wind speed prediction in Hexi Corridor of China. *Energy*. 2011;36(3):1668–1679. DOI: 10.1016/j.energy.2010.12.063
- [17] J. Wang, J. Hu. A robust combination approach for short-term wind speed forecasting and analysis – combination of the ARMA (Autoregressive Integrated Moving Average), ELM (Extreme Learning Machine), SVM (Support Vector Machine) and LSSVM (Least

Square SVM) forecasts using a GPR (Gaussian Process Regression) model. *Energy*. 2015;93(1):41–56. DOI: 10.1016/j.energy.2015.08.045

- [18] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew. Extreme learning machine: theory and applications. *Neurocomputing*. 2006;70:489–501. DOI: 10.1016/j.neucom.2005.12.126
- [19] J. Zhao, J. Wang, F. Liu. Multistep forecasting for short-term wind speed using an optimized extreme learning machine network with decomposition-based signal filtering. *Journal of Energy Engineering*. Article ID: 04015036, 2015. DOI: 10.1061/(ASCE)EY.1943-7897.0000291.
- [20] Z. Zhang, Y. Song, F. Liu, J. Liu. Daily average wind power interval forecasts based on an optimal adaptive-network-based fuzzy inference system and singular spectrum analysis. *Sustainability*. 2016;8(2): 125. DOI: 10.3390/su8020125.
- [21] S.J.R. Jiang. ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man and Cybernetics*. 1993;23(3):665–685. DOI: 10.1109/21.256541

Fuzzy Logic Control of Switched Reluctance Motor Drives

M. Divandari and B. Rezaie

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/63642>

Abstract

In this chapter, the electromechanical behavior of switched reluctance motor (SRM) is first modeled by analyzing the related nonlinear differential equations. In the model, the estimation of rotor speed is also considered. After modeling, the effects of torque ripple, radial force, and acoustic noise are investigated. As we know, torque ripple and acoustic noise are two of the main disadvantages of a switched reluctance motor. Thus, a fuzzy logic current compensator is proposed both for reducing the peak of radial force and for decreasing acoustic noise effects. In the parts that torque reduces, the fuzzy logic current compensator injects additional current for each phase current to overcome the torque ripple. Also, the fuzzy logic current compensator reduces speed estimation error. The speed estimation is carried out using a hybrid sliding mode observer which estimates the rotor position and speed for a wide speed range. These new approaches have been simulated using MATLAB/SIMULINK for a nonlinear model of switched reluctance motor. The simulation results indicate that proposed methods decrease the maximum radial force and the torque ripple while the maximum torque is preserved. Also, these results show that proposed methods will estimate the rotor position and speed with high precision for all speeds from near zero speeds up to rated speed. These procedures have the advantages of simple implementation on the every switched reluctance motor drive without extra hardware, low cost, high reliability, low vibration, and excellent performance at long term.

Keywords: SRM, torque ripple, acoustic noise, radial force, fuzzy logic

1. Introduction

The switched reluctance machines could be claimed to have a pivotal role in the changes in electrical machines technology since the mid-1960s. Switched reluctance motor (SRM) torque

is generated by rotor moving where the inductance of each phase is maximized. The simple structure is the main attractive feature as there is no winding or permanent magnet. Therefore, the manufacturing cost is lower than other electrical machines. SRMs have a wide range of applications in industry, such as hybrid vehicles, wheelchairs, aircrafts, starter/generator systems, washing machines, and other industry and home applications [1, 2].

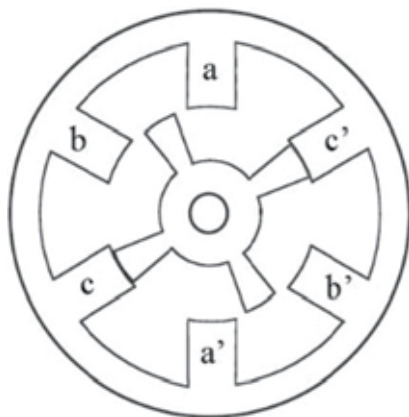


Figure 1. 6/4-pole SRM.

The SRM windings are excited by an external DC supply and current is injected only into one phase. A simple SRM with three stator phases is shown in **Figure 1**. It is clear that the stator has six poles and the rotor possesses four poles. In the SRM, DC circuit supplies are simpler in construction of power switches than those of other motor drives. Therefore, many manufacturers have very high intends for applying SRMs in home electrical motors. Also, high torque at low speeds can make SRM suitable for direct use without gearbox. However, in comparison with high performance motors, SRM suffers from a high-level acoustical noise and mechanical vibration. When one phase is excited, the rotor rotates until two of its poles are aligned with two stator poles of the excited phase and torque is generated by increasing the inductance while reluctance of the magnetic circuit decreases. The stator poles and salient rotor provide an un-uniform air gap, which leads to nonlinear behavior of electromagnetic equations in SRM [3]. As we know, vehicles need initial acceleration and gradability with minimum power rating. Rahman et al. have investigated the capabilities of the SRM for the electric vehicle (EV) and hybrid EV applications. These investigations have been carried out in two stages. First stage involves the machine design and the finite element analysis of static characteristic and in second stage, the finite element field solution have been used in the development of a nonlinear model. Experimental results have demonstrated the improvement of PF at a high-speed operation [4]. A large number of methods have been also introduced to accomplish position sensor-less operation of the SRM during the last 15 years. Mese and Torrey have presented a new approach to the sensor-less control of the SRM. In this method, an artificial neural network (ANN) acts on the measurement of the phase flux linkages and phase currents. The investigations conducted throughout this research have shown that a properly trained

and utilized ANN is capable of estimating the rotor position of SRM within acceptable accuracy limits. The method deserves consideration as a candidate for integration into practical SRM drive systems [5]. Most modern high-performance AC drives utilize synchronous frame current regulators, which see a pure DC current command in steady state. Schulz et al. have introduced a designed methodology for digital proportional–integral current regulators that may be used for the highly nonlinear SRM control. In this procedure, the main goal was to focus on the development of a fully digital implementation of the inner current control loop for high-performance SRM drives operating with high saturation. Simulation and experimental results have demonstrated the excellent performance over the entire operating regime [6]. Rotor position sensors, such as shaft encoders and Hall effect sensors, have been applied in SRM drives to determine the rotor position. These sensors increase drive cost and decrease reliability of the SRM drive in industrial applications. Gao et al. proposed a sensor-less control scheme for SRM drive at low speed. In this study, incremental inductance of each active phase is estimated by using the terminal measurement of this phase. Simulation and experimental results have shown that errors between estimated position and actual position (measurement position) are less than one mechanical degree [7]. Husain and Hossain have presented the modeling, simulation and control aspects of four-quadrant SRM drives. A complex model has been described for the physical motor simulation to incorporate the important dynamics of the SRM. The results obtained from the final simulation model are extremely close to the experiments. Such a model could be reliably used for performance evaluations and future developments [8]. A sensor-less drive that decreases cost and increases reliability, which extracts rotor position information indirectly from electric signals of motor terminals, is highly desirable. A sliding mode observer, with its advantages of inherent robustness parameters of uncertainty, computational simplicity, and high stability, provides a powerful approach to implement sensor-less schemes. Divandari et al. have presented the estimated the rotor position and speed of SRM drive based on hybrid observer (HO). The HO has provided online estimates of the rotor position and speed for a wide speed range using the current sliding mode observer (CSMO) and flux linkage sliding mode observer (FSMO). In addition, a fuzzy logic current compensator (FLCC) for reducing torque ripple is presented. Simulation results have shown that proposed SRM drive decreases estimation errors and torque ripple, effectively. The method has good performance in a wide speed range [9, 10]. One of the main problems in sensor-less SRM drive is position sensing at startup. Yan-Tai et al. [11] proposed a method for position sensing at startup using quadratic polynomial regression. In this method, without using magnetic specifications, position estimation at startup can be done by a microcontroller. The magnetic flux in the SRM passes across the air gap in an approximate radial direction generating radial forces on the stator and rotor, resulting in magnetic noise and vibrations. Unbalanced radial forces acting on a rotor shaft are undesirable as they cause motor vibrations. Lin and Yang proposed a new excitation approach for the torque and radial force control of a 12/8 poles SRM. In this method, the SRM drive can generate simultaneously the modified shaft radial force and rated torque. The experimental results verified that the SRM produced smoother torque ripples when the proposed commutation scheme was used [12]. Harmonics magnitude of the radial force has basically role in acoustic noise generation. Takiguchi et al. [13] proposed a new method for elimination of second and third harmonics. They compared

experiment results with finite-element method. Also, for acoustic noise reduction, Hyong-Yeol et al. [14] introduced an applicable approach by new rotor and stator design. In this approach, laminations of rotor and stator have similar skew angle and radial force can be decreased significantly by changing air gap uniform. As we know, because of SRM dynamical nonlinearities, its control is complex. Hajatipour and Farrokhi have introduced an adaptive intelligent control based on the Lyapunov stability theory to control the speed of SR motors with good accuracies and performances. The proposed controller is composed of a speed controller and a torque controller. Moreover, the torque ripple reduction was achieved by employing a neural network for torque estimation. The simulation results showed a good performance of the proposed controller in speed controlling and torque ripple reduction [15]. Most observers are static and once their gains have been determined. As we know, most observers are based model and uncertain in parameters will increase estimation errors. Therefore, online gain regulation can improve deficiency of based model observers. A new generation of observers named dynamic observers (DO) described by Divandari et al. [16]. Dynamic observers can be used in tracking and drive systems. Also, main problem of SRM is high torque ripple, and this problem will be solved by optimization of motor design and improvement of drive performance. We suppose a SRM is already built; therefore, torque ripple can be minimized by the optimization of the current waveform in the SRM drive. Divandari et al. have proposed a new simple procedure for minimizing the torque ripple via fuzzy logic control of the SRM. Simulation results have demonstrated that proposed drive estimates rotor speed with high accuracy for a wide speed range. Also, the fuzzy logic current compensator optimizes the torque ripple and decreases estimation error. In this chapter, to define a SRM drive, simulation, and investigations of the SRM behavior such as torque ripple, rotor position estimation, and acoustic noise, we present a nonlinear model of the SRM by using experimentally measured characteristics. After modeling, with current profile optimization via fuzzy logic control (FLC), the torque ripple is minimized. Acoustic noise sources in the SRM depend on torque ripple and radial force. Therefore, we discuss about radial force and important parameters in radial force and torque ripple, together. Calculations of radial force with two methods are expressed as well. By using FLC, the torque ripple and radial force are optimized. Also, in this chapter, the study of a low-cost, sensor-less-based, and speed-controlled SRM drive system is considered. A hybrid algorithm (HA) with CSMO and flux sliding mode observer (FSMO) is developed to estimate rotor position and speed of the SRM. The observer gains will be updated separately online at a wide speed range with band of estimated errors. Also, by means of indirect speed estimation, we propose torque compensation by current profile.

In the following paragraphs, this chapter presents an intelligent solution for SRM drives.

2. Non-linear model of SRM

To study the characteristics of SRM drive system such as current profile, torque ripple, rotor estimation, radial force, and acoustic noise, we should present a nonlinear model of SRM. Also, for SRM differential equations, it is required to study dynamic characteristics consisting of flux

linkage and torque. The measured flux linkage data and torque data applied in this chapter for each phase of SRM are shown in **Figures 2** and **3** [16].

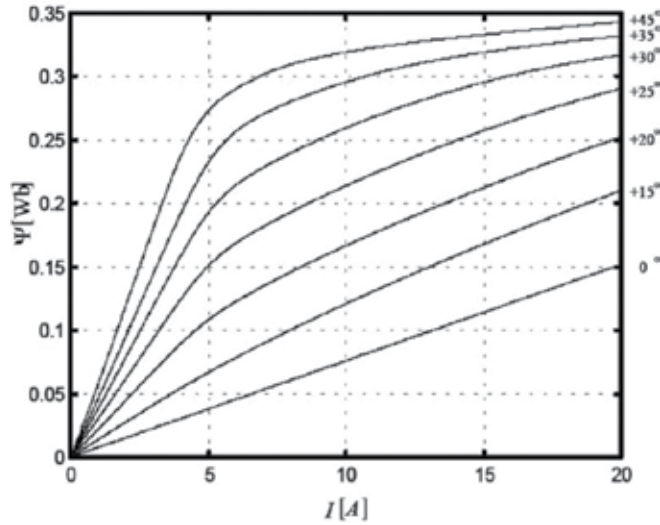


Figure 2. The profile of measured SRM flux linkage versus current, versus position [2].

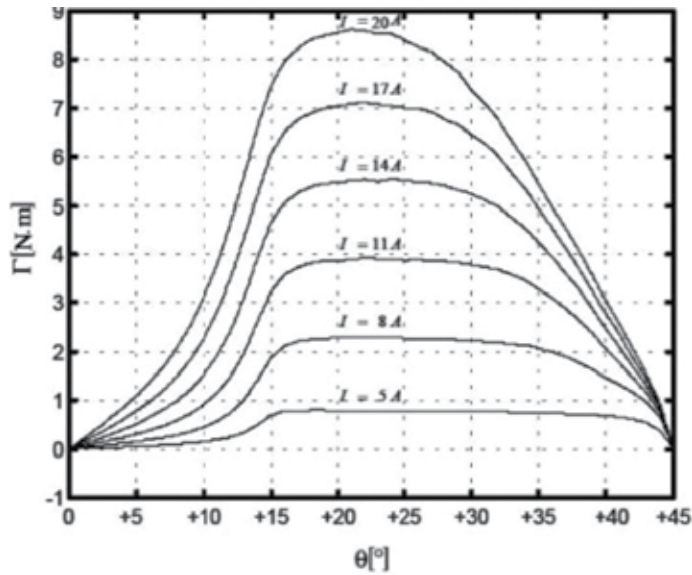


Figure 3. The profile of measured SRM torque versus current, versus position [2].

The torque and flux depend on position and current. The flux and voltage for each phase of SRM can be expressed as:

$$\phi_j = L_j(i_j, \theta) i_j \quad (1)$$

$$\begin{aligned} V_j &= R_j i_j + \frac{d\phi_j(i_j, \theta)}{dt} = R_j i_j + \frac{\partial \phi_j}{\partial i_j} \frac{di_j}{dt} + \frac{\partial \phi_j}{\partial \theta} \frac{d\theta}{dt} \\ &= R_j i_j + L_j \frac{di_j}{dt} + \frac{\partial \phi_j}{\partial \theta} \omega \end{aligned} \quad (2)$$

Finally, the torque of SRM is

$$W_c(\theta, i_j, l_g, L_r) = \int_0^{i_j} \phi(\theta, i_j, l_g, L_r) di \quad (3)$$

$$T_j = \left[\frac{\partial W_c(\theta, i_j, l_g, L_r)}{\partial \theta} \right]_{i_j = \text{const}} \quad (4)$$

$$T_e = \sum_{j=1}^n T_j \quad (5)$$

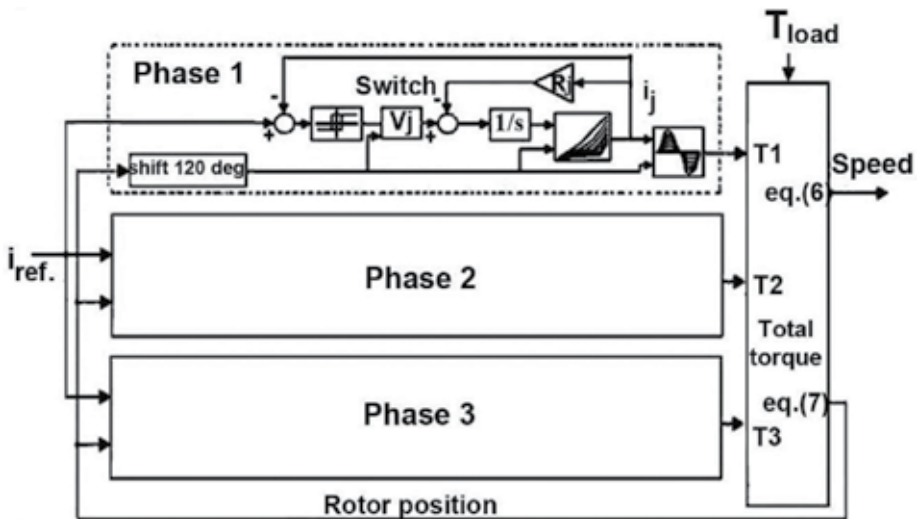


Figure 4. Block diagram of nonlinear model of three-phase SRM [2].

The mathematical motion of the motor by the action of electromagnetic torque and load torque is

$$T_e - T_l = J \frac{d\omega}{dt} + B\omega \quad (6)$$

The equation of rotation is

$$\frac{d\theta}{dt} = \omega \quad (7)$$

According to differential equations and measured data, block diagram of nonlinear model of three-phase SRM is shown in **Figure 4**.

3. Torque Ripple minimization with FLCC

As we know, the main disadvantage of SRM is a high level of torque ripple. One of the methods to decrease the torque ripples is the deformation of the current profile [17]. This chapter presents a new simple technique for minimizing the torque ripple via FLC. This technique is based on the transfusion of additional current in each phase by using FLCC [10].

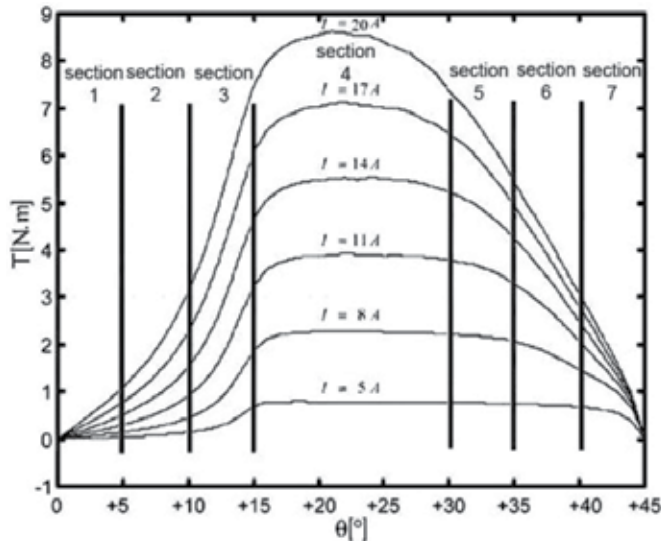


Figure 5. Nonlinear SRM torque curves (A–G sections) [2].

Furthermore, torque ripple is one of the main acoustic noise sources in SRM drive. Therefore, torque ripple minimization techniques reduce mechanical vibrations on the bearing and increase estimation errors. In this method, nonlinear torque curves are divided into seven sections (1–7). These sections are shown in **Figure 5**. Each section is 5° and in each 5° , the nonlinear torque behavior can be considered linear. SRM torque in section 5 is nearly flat, but in other sections, negative torque slop will be compensated by using FLCC. The 1–7 sections of torque curves have been listed in **Table 1**. Also, the one-phase block diagram of fuzzy logic torque ripple minimization is illustrated in **Figure 6**. Fuzzy rules are defined for a wide speed range (near zero speeds up to rated speed). FLCC acts on the reference current when torque curve is 1–3 and 5–7 sections. According to current compensation, new reference current is

$$I_{ref.new} = I_{ref.} + I_{Comp.} \quad (8)$$

Position ($^\circ$)	Sections
$0 \leq \theta \leq 5$	1
$5 \leq \theta \leq 10$	2
$10 \leq \theta \leq 15$	3
$15 \leq \theta \leq 30$	4
$30 \leq \theta \leq 35$	5
$35 \leq \theta \leq 40$	6
$40 \leq \theta \leq 45$	7

Table 1. Sections 1–7 of nonlinear torque.

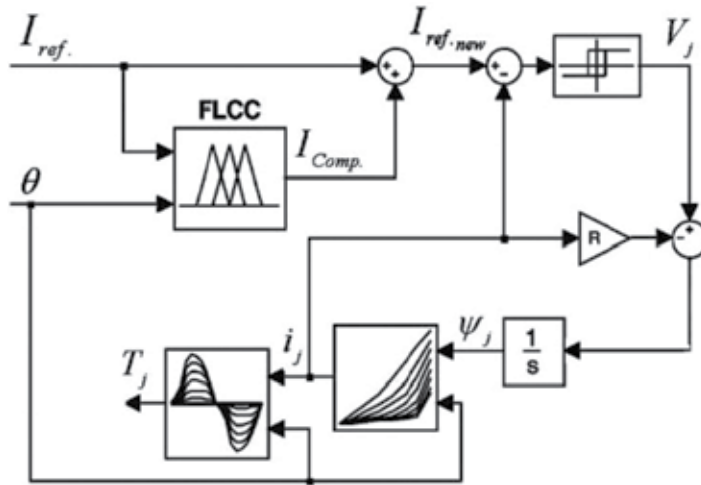


Figure 6. One phase block diagram of fuzzy logic torque ripple minimization [2].

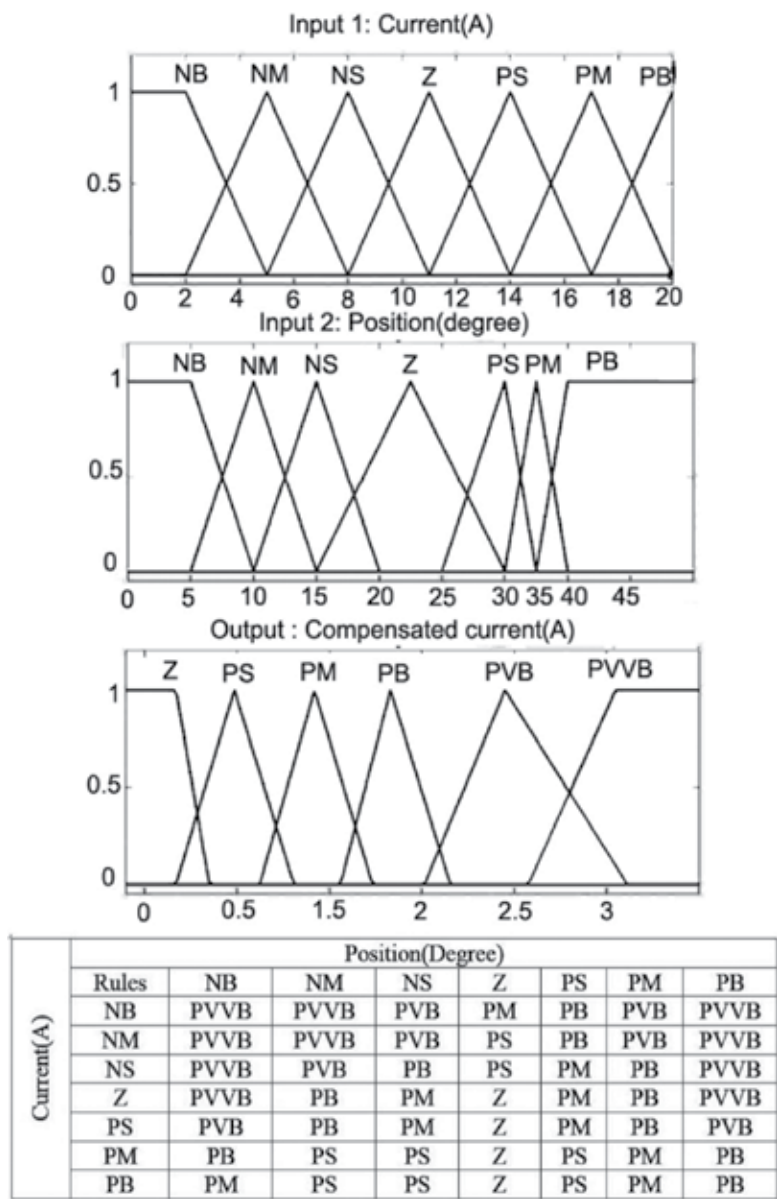


Figure 7. Fuzzy logic current compensator.

As we know, fuzzy membership function arrangements and fuzzy rules determination have a main role in designing FLCC for torque ripple minimization. In this technique, the current and rotor position with seven membership functions as inputs and the compensated current with six membership functions as output are determined. In the sections of A, B and F, G which torque decrease with big negative slope, fuzzy rules set up on PVB or PVVB but in the C and

E sections which torque reduces with slow negative slope, fuzzy rules set up on PB and PVB. Also, at high speeds, fuzzy rules are moderated to PB because high current peaks in motor current will cause damage to power switches and the SRM. **Figure 7** shows the input and output membership functions as well as the fuzzy rules. Input 1 (current) is defined 0–20 A, input 2 (position) 0–45 degree, and output (compensated current) 0–3 A [index 1].

4. Acoustic noise reduction

Mechanical vibration and the acoustic noise in SRMs cause that they could not commercially yet competitive with other electrical machine drives. Noise sources recognition and elimination or reduction of noise in electrical machine drives can be dated since the 1940s when new materials were presented to improve electrical machines designs [17]. As we know, SRM radial force and torque ripple are the important sources of acoustic noise generation in SRM drives. When stator winding of each phase is energized by DC external supply, a magnetic flux will cross from air gap and an approximate radial force excites diverse mode shapes. Moreover, energizing stator winding in the SRM with a DC voltage generates lateral force, tangential force, and torque. The current in the stator windings could generate a magnetic flux in stator winding that would emit acoustic noise [2].

Radial force is relevant to many parameters such as air gap length, position, rotor length, and rotor radius. Also, radial force depends on current square and turns number of stator windings, and sound power is relevant to radial force square. In the last decade, researches illustrate that most of the studies focus on the torque ripple minimization while stator and rotor shape design and radial magnetic force have principle rule at acoustic noise of the SRM. In this procedure, torque ripple will be minimized and the maximum radial force decreases while we try to maintain maximum torque by the current profile deformation via FLC. This technique is based on transfusion of additional current by using FLC.

4.1. Calculation of SRM radial force

As it was mentioned in previous sections, the radial force and acoustic noise are relevant to many SRM parameters. Magnetic specifications, machine design, and material type are effective in the radial force calculation. For more study, we should first analyze the equations of radial, tangential, and lateral forces. **Figure 8** shows radial, tangential, and lateral forces in the SRM. The flux density in the SRM air gap between stator and rotor is given as

$$B_g(\theta, i_j, l_g, L_r) = \frac{\phi}{L_{r,i} \cdot \theta} = \mu_0 \cdot H_g = \mu_0 \cdot \frac{N_{ph,i}}{l_g} \quad (9)$$

$$N_{ph} = \frac{l_g}{\mu_0 \cdot L_{r,i} \cdot j} \cdot \frac{\phi}{\theta}$$

where N_{ph} is the turns number for each phase, r is the outer rotor radius, H_g is the magnetic field strength, and μ_0 is the air permeability.

The electrical energy (dW_e) is expressed by

$$dW_e = i_j \cdot d\lambda = i_j \cdot d(N_{ph} \cdot d\phi) = N_{ph} \cdot i_j \cdot d\phi = \frac{l_g}{\mu_0 \cdot L_r \cdot r} \cdot \frac{\phi}{\theta} \cdot d\phi \quad (10)$$

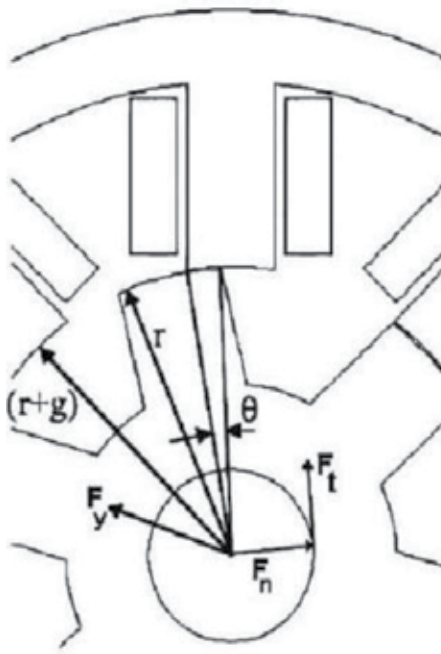


Figure 8. Radial, tangential, and lateral forces in SRM [2].

The stored energy in the magnetic field (W_s) is

$$W_s = \frac{l_g \cdot L_r \cdot r \cdot \theta}{2\mu_0} \cdot B_g^2(\theta, i_j, l_g, L_r) = \frac{l_g}{2\mu_0 \cdot L_r \cdot r} \cdot \frac{\phi^2}{\theta} \quad (11)$$

where $l_g \cdot L_r$ is the air gap specifications and constant value. The energy equation is

$$dW_e = dW_s + dW_m \quad (12)$$

where dW_m is the mechanical energy and dW_s is the field energy. In this chapter, we propose two procedures for tangential, radial, and lateral force calculations. In the first procedure, of the tangential force calculations obtained from Eq. (12) as

$$dW_s = -\frac{l_g}{2\mu_0 L_r r} \cdot \frac{\varphi^2}{\theta^2} d\theta + \frac{l_g}{\mu_0 L_r r} \cdot \frac{\varphi}{\theta} d\varphi \quad (13)$$

By substituting Eqs. (10) and (13) in Eq. (12), the mechanical energy is calculated as

$$dW_m = \frac{l_g}{2\mu_0 L_r r} \cdot \frac{\varphi^2}{\theta^2} d\theta \quad (14)$$

Moreover, electromagnetic torque is

$$T_e = \frac{dW_m}{d\theta} = \frac{l_g}{2\mu_0 L_r r} \cdot \frac{\varphi^2}{\theta^2} = \frac{l_g r L_r}{2\mu_0} B_g^2(\theta, i_j, l_g, L_r) \quad (15)$$

Also, the tangential force is obtained by dividing tangential torque by the radius of the rotor pole, yielding

$$F_t = \frac{T_e}{r} = \frac{l_g L_r}{2\mu_0} B_g^2(\theta, i_j, l_g, L_r) \quad (16)$$

and the radial force is given by

$$F_n = \frac{dW_m}{dl_g} = \frac{1}{2\mu_0 r L_r} \cdot \frac{\varphi^2}{\theta} = \frac{r L_r \theta}{2\mu_0} B_g^2(\theta, i_j, l_g, L_r) \quad (17)$$

Similarity, the lateral force can be derived as

$$F_y = \frac{l_g r \theta}{2\mu_0} B_g^2(\theta, i_j, l_g, L_r) \quad (18)$$

The ratio of the radial force to the tangential force from Eqs. (17) and (18) becomes

$$\frac{F_n}{F_t} = \frac{r \cdot \theta}{l_g} \quad (19)$$

and rotor angle is given as

$$\theta(\text{rad}) = \frac{4 \cdot \pi}{P_s \cdot P_r} \quad (20)$$

where P_s is number of stator poles and P_r is the number of rotor poles. In the next procedure, we should apply Eq. (3) and flux equation. Flux equation is given as

$$\begin{aligned} \varphi &= B_g(\theta, i_j, l_g, L_r) \cdot A = \frac{\mu_0 \cdot L_r \cdot r \cdot \theta \cdot N_{ph}^2}{2l_g} \cdot i_j \\ &= \frac{\mu_0 \cdot L_r \cdot r \cdot \theta \cdot N_{ph}^2}{l_g} \cdot i_j \end{aligned} \quad (21)$$

Substituting Eq. (21) in Eq. (3), co-energy is obtained as

$$W_c(\theta, i_j, l_g, L_r) = \frac{\mu_0 \cdot L_r \cdot r \cdot \theta \cdot N_{ph}^2}{2l_g} \cdot i_j^2 \quad (22)$$

Finally, torque and tangential, radial and lateral forces are obtained as

$$T_e = \frac{\partial W_c(\theta, i_j, l_g, L_r)}{\partial \theta} = \frac{l_g}{2\mu_0 \cdot L_r \cdot r} \cdot \frac{\varphi^2}{\theta^2} \quad (23)$$

$$F_t = \frac{T_e}{r} = K \cdot \left(\frac{l_g \cdot L_r}{2\mu_0} \right) \quad (24)$$

$$F_n = \frac{\partial W_c(\theta, i_j, l_g, L_r)}{\partial l_g} = K \cdot \left(\frac{-r \cdot L_r \cdot \theta}{2\mu_0} \right) \quad (25)$$

$$F_y = \frac{\partial W_c(\theta, i_j, l_g, L_r)}{\partial L_r} = K \cdot \left(\frac{l_g \cdot r \cdot \theta}{2\mu_0} \right) \quad (26)$$

respectively, where

$$K = \frac{\mu_0^2 \cdot N_{ph}^2 \cdot i_j^2}{l_g^2} \quad (27)$$

Eqs. (23), (25), and (27) illustrate the radial force and torque. It is clear that radial force and torque are directly relevant to current square.

4.2. Calculation of SRM acoustic noise

The acoustic noise value is relevant to circumferential perversion due to the radial force wave. The analytical presentation for the circumferential perversion can be defined as

$$D_{\text{circum}(f_{\text{exc}})} = \frac{\frac{12 \cdot F_n(f_{\text{exc}}) \cdot R_m}{m^4 \cdot E} \cdot \left(\frac{R_m}{h_s} \right)^3}{\sqrt{\left\{ 1 - \left(\frac{f_{\text{exc}}}{f_m} \right)^2 \right\}^2 + \left(\frac{\delta}{\pi} \cdot \frac{f_{\text{exc}}}{f_m} \right)^2}} \quad (28)$$

$$f_{\text{exc}}(n) = n \cdot f_p = \frac{n \cdot \omega \cdot N_{rp}}{60} \quad (29)$$

where

$D_{\text{circum}(f_{\text{exc}})}$	amplitude of dynamic deflection (m)
F_n	amplitude of radial force wave (N/m ²)
δ	logarithmic decrement = $2 \cdot \pi$ damping factor
N_{rp}	number of rotor poles
f_p	fundamental frequency of phase current (Hz)
$F_{\text{exc}(n)}$	excitation frequency (Hz) with $n = 1, 2, 3, \dots$ harmonic numbers
R_m	mean radius of stator yoke
h_s	stator pole height
$m \cdot f_m$	circumferential mode number and mode frequency
E	module of stator material elasticity

Sound power radiated by an electric machine can be expressed as

$$P = 4 \sigma_{\text{rel}} \rho c \pi^3 f_{\text{exc}}^2 D_{\text{circum}} R_{\text{out}} l_{\text{stk}} \quad (30)$$

where

P	radiated sound power (W)
σ_{rel}	relative sound intensity $\sigma_{\text{rel}} = k_2/(1+k_2)$
k	wave number $k = (2 \cdot \pi \cdot R_{\text{out}} \cdot f_{\text{exc}})/c$
C	sound speed (m/s)
Q.c	415 N s/m ³ at 20°C (ρ is air density)

$R_{\text{out}}, l_{\text{stk}}$ outer radius and stack length of the stator (m) [2]. Depending on the threshold of human ear sensation, the reference of sound power level (P_{ref}) is 10–12 W. Consequently, the acoustic noise power level in decibels becomes

$$L_{\omega} = 10 \log\left(\frac{2 \cdot P}{P_{\text{ref}}}\right) \quad (31)$$

5. Acoustic Noise Rreduction (ANR) with FLC

In the SRM drive, radial force is dependent on

$$L_{\omega} \propto P \propto D_{\text{circum}} \propto F_n \propto K \propto i_1^2 \quad (32)$$

Also, SRM torque is function of

$$T_e \propto K \propto i_1^2 \quad (33)$$

In this procedure, we suppose that SRM has been already built. Our aim is to optimize radial force and torque ripple in the SRM drive so far as keep maximum torque using phase current waveform. **Figure 9** illustrates the characteristics of torque and radial force of SRM at rated current. Two sections (1 and 2) can be seen in **Figure 9**. In the A area not only torque decreases but also radial force increases and even will be maximized, but in the B area, torque increases and will be maximized. Also, radial force will not be maximized.

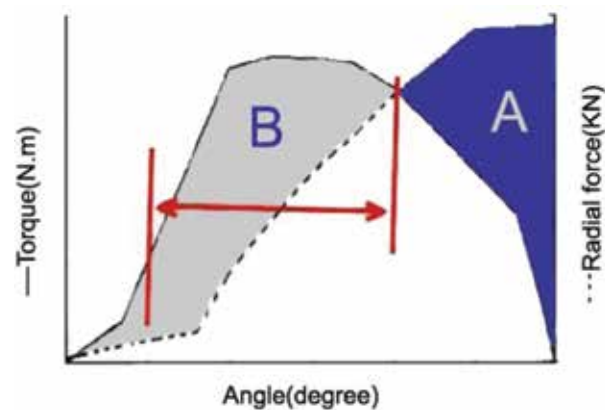


Figure 9. Characteristic of torque and radial force of SRM at rated current [2].

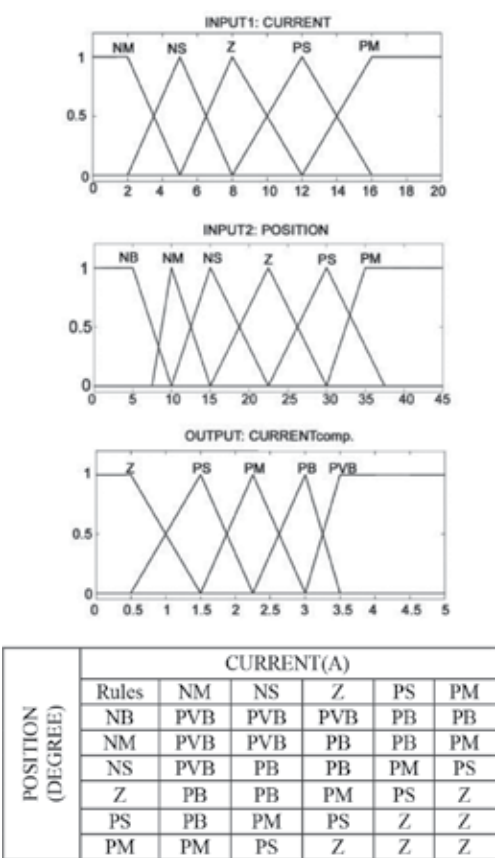


Figure 10. Fuzzy logic member functions for current waveform.

Therefore, with discussions of torque ripple minimization and fuzzy logic rules, we can keep the maximum torque and reduce radial force, and consequently, acoustic noise will be reduced.

Finally, **Figure 10** shows the fuzzy logic member functions for current waveform and fuzzy logic rules for acoustic noise reduction (ANR) and torque ripple minimization, respectively. Input 1 (current) is defined 0–20 A, input 2 (position) 0–45°, and output (compensated current) 0–5 A [index 1].

6. Estimate of position and speed of SRM

Applying a shaft encoder in electrical machine drive decreases the reliability and increases the cost of drive. Therefore, usually researchers measure electrical signals consisting of voltages and currents for position estimation. The recent well-known methods for position estimation in the SRM drives have been focused on the three methods:

1. Model-based observer.
2. Inductance-based measurement applying current fall time or rise time.
3. Inductance-based estimation applying the two separated methods: (a) demodulation and constant current and (b) constant flux used to sensor signals [17].

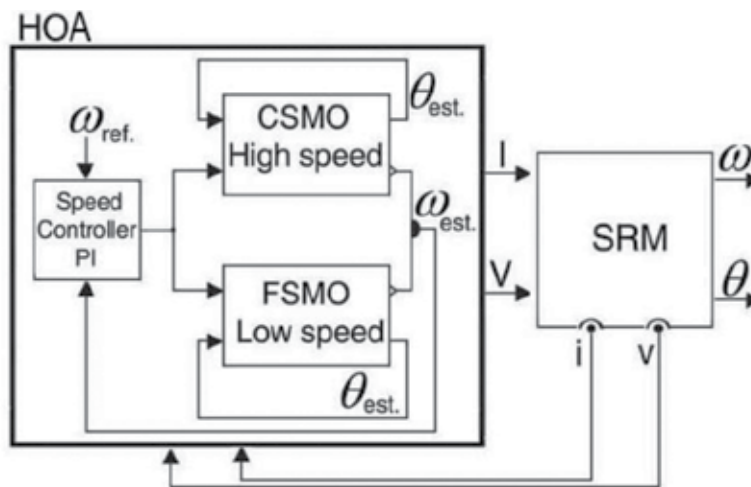


Figure 11. Block diagram of HOA [9].

Researchers have proposed many methods of sensor-less SRM drives in the last decade. In these methods, estimation techniques have been applied for either rated speed or low speeds. One of these methods is sliding mode observer (SMO). SMOs have many advantages such as inherent robustness in parameters uncertainty, fast computational, but they have some disadvantages such as sensitive to model changes and selecting of SMO gains.

In this method, a hybrid observer algorithm (HOA) is proposed to estimate position in the SRM drive for a wide speed range. Proposed HOA consists of two SMO: CSMO and FSMO. The CSMO and FSMO gains will be regulated in high and low speed with value of estimation error, respectively [16].

The block diagram HOA is illustrated in **Figure 11**. HOA consists of CSMO, FSMO, and a proportional integral controller for speed control.

6.1. Current sliding mode observer

Proposed CSMO for sensor-less SRM drive has two strategies:

- Measuring of electrical signals (voltages and currents).
- Voltages as inputs to nonlinear model in which currents will be estimated.

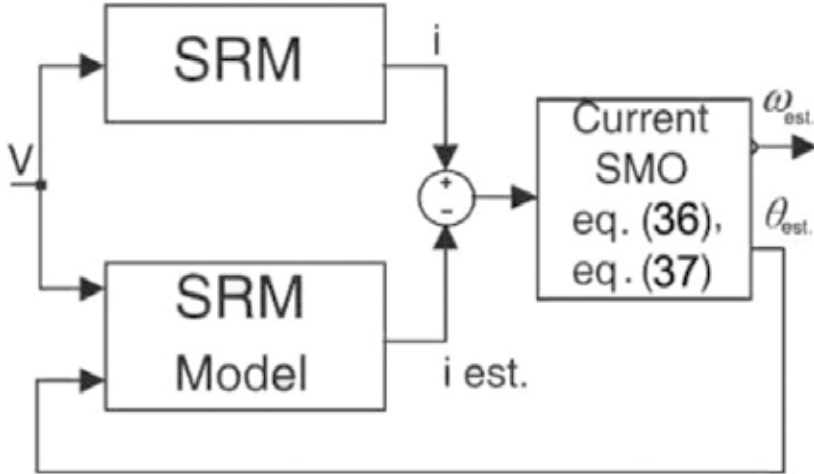


Figure 12. Block diagram of CSMO [9].

Principle of CSMO operating is constructed on the current error for high speeds. Block diagram of CSMO is shown in **Figure 12**.

According to the system differential equations derived from the nonlinear model of SRM, a CSMO for rotor position and speed can be defined as follows:

$$S(t) = i_j - i_{jest}. \quad (34)$$

$$S_{cont.} = \text{sgn.} \sum_{j=1}^n S(t) \quad (35)$$

Differential equations of CSMO are

$$\dot{\theta}_{est.} = \omega_{est.} + \alpha_{\theta C} S_{cont.} \quad (36)$$

$$\dot{\omega}_{est.} = T_{est.} + \alpha_{\omega C} S_{cont.} \quad (37)$$

The estimation error is defined as follows:

$$e_{\theta} = \theta_{est.} - \theta, e_{\omega} = \omega_{est.} - \omega$$

Consequently, by subtracting Eq. (36) from (7) and Eq. (37) from (6), we get the error dynamics:

$$\dot{e}_{\theta} = e_{\omega} - \alpha_{\theta C} S_{cont.} \quad (38)$$

$$\dot{e}_{\omega} = \left[\frac{T_e - T_l}{J} \right] - \left[\frac{T_e - T_l}{J} \right]_{est.} - \alpha_{\omega C} S_{cont.} \quad (39)$$

By appropriately choosing the two CSMO gains $\alpha_{\theta C}$, $\alpha_{\omega C}$ can make:

$$e_{\theta} \dot{e}_{\theta} < 0 \Rightarrow (e_{\theta} \Rightarrow 0) \Rightarrow (\theta_{est.} \Rightarrow \theta) \quad (40)$$

$$e_{\omega} \dot{e}_{\omega} < 0 \Rightarrow (e_{\omega} \Rightarrow 0) \Rightarrow (\omega_{est.} \Rightarrow \omega) \quad (41)$$

6.2. Flux sliding mode observer

The FSMO acts on the flux linkage error for low speeds. In this SMO, voltages and currents of each phase will be measured. **Figure 13** illustrates the FSMO block diagram. The flux linkage of j th phase is estimated by using the voltage and current as follows:

$$\lambda_{j_{est.}}(t) = \int_0^t (V_j(\tau) - i_j(\tau) \cdot r_j) d\tau \quad (42)$$

where v_j , i_j , and r_j are the voltage, current, and resistance of j th phase. An accurate but simplified flux linkage model is used to obtain the phase flux linkage and is given as follows:

$$Z(i_j, \theta) = i_j \cdot W_j(\theta)$$

$$W_j(\theta) = \cos(N_r \cdot \theta_{est.} - \frac{(n-1)2\pi}{N_{ph}})$$

$$\lambda_j = \lambda_s \cdot Z(i_j, \theta) \cdot (1 + \frac{Z(i_j, \theta)^2}{2}) \quad (43)$$

where N_r is the number of rotor poles, and N_{ph} is the number of phases, $\theta_{est.}$ is the estimated position, and λ_s is the saturated flux linkage consequently, flux linkage error is

$$e_\lambda = \sum_{j=1}^{N_{ph}} \frac{dW_j(\theta)}{d\theta} \bigg|_{\theta=\theta_{est.}} \cdot (\lambda_j - \lambda_{j_{est.}}) \quad (44)$$

and differential equations of FSMO are

$$\dot{\theta}_{est.} = \omega_{est.} + \alpha_{\theta F} \text{sgn}(e_\lambda) \quad (45)$$

$$\dot{\omega}_{est.} = \alpha_{\omega F} \text{sgn}(e_\lambda) \quad (46)$$

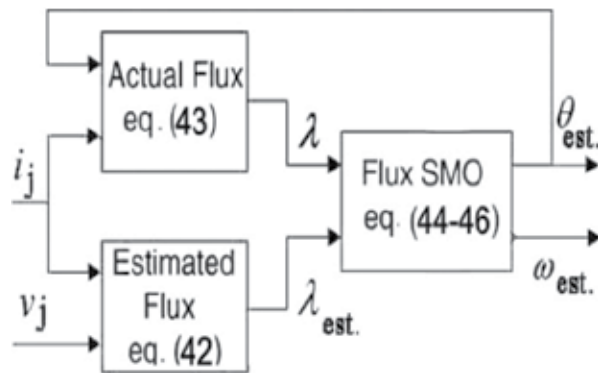


Figure 13. Block diagram of FSMO [9].

By appropriately choosing the two FSMO gains $\alpha_{\theta F}$, $\alpha_{\omega F}$ can make:

$$e_{\lambda} \Rightarrow 0 \Rightarrow \theta_{est.} \Rightarrow \theta \text{ \& } \omega_{est.} \Rightarrow \omega \quad (47)$$

6.3. Indirect speed estimation by torque estimation

Illustrated in **Figure 14** sensor-less SRM drive scheme is proposed. In this proposed method, sensor-less SRM drive is model based and torque estimator just acts based on the measured phase voltages and measured currents. Finally, in this approach, a PI controller is applied for SRM speed control [18].

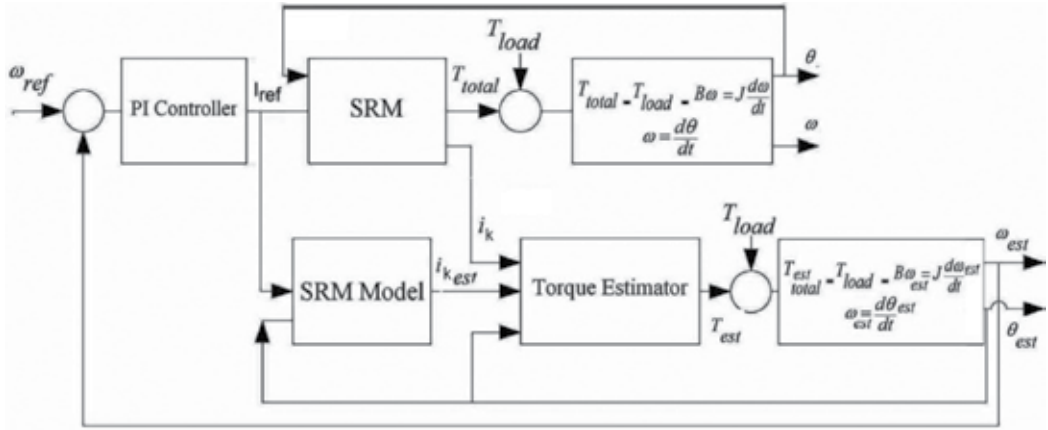


Figure 14. The block diagram of sensor-less SRM drive [18].

The sensor-less SRM drive is designed to estimate indirect speed and rotor position. The SRM torque is functional of phase current and rotor position. Thus, error of phase current and estimated current of the SRM plays basic role in torque estimation.

Current error for each phase e_k can be defined as follows:

$$e_k = i_k - i_{k_{est}}. \quad (48)$$

For the purpose of current compensation, current error should be bounded. Error bound can be described as

$$\beta < e_k < -\beta \quad (49)$$

$$\gamma < \dot{e}_k < -\gamma \quad (50)$$

Torque estimation algorithm is obtain as

$$\text{if } :e_k < -\beta \Rightarrow i_{kcomp.} = i_k + \alpha.f(e_k, e_k) \quad (51)$$

$$\text{elseif } :e_k > \beta \Rightarrow i_{kcomp.} = i_k - \alpha.f(e_k, e_k) \quad (52)$$

$$\text{elseif } :i_{kcomp.} = i_k \quad (53)$$

where α, β, γ are determined using fuzzy logic.

Figure 15 illustrates the adaptive fuzzy logic of torque estimator of the k th phase. The torque of each phase using lockup table and compensated current can be calculated using Eqs. (51)–(53).

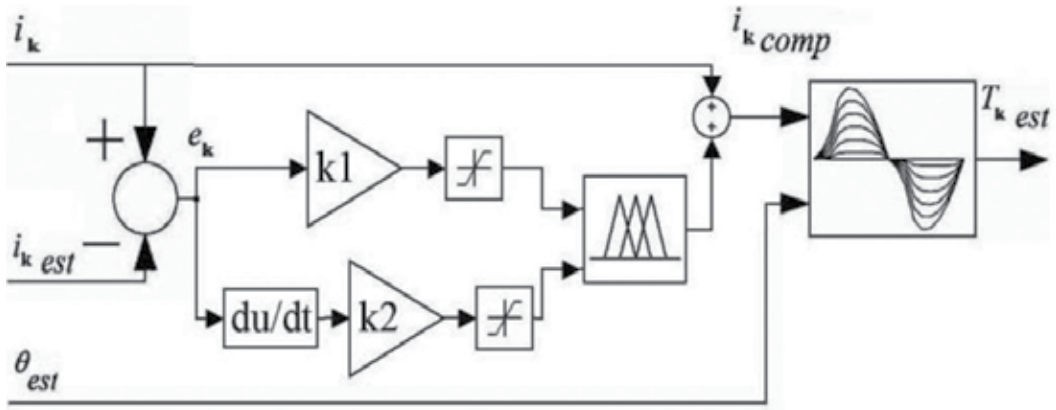


Figure 15. Adaptive fuzzy logic of torque estimator [18].

In fuzzy logic compensator, fuzzy rules act on current error (with five membership function: NB, NS, Z, PS, PB) and current error derivative (with five membership functions: NB, NS, Z, PS, PB) as inputs and compensated current (with five membership functions: NB, NS, Z, PS, PB) as output. **Figure 16** shows fuzzy logic membership functions. In this section, current error -0.4 to 0.4 as β (input 1), current error derivative -0.3 to 0.3 as γ (input 2), and $\alpha.f(e_k, e_k)$ -0.3 to 0.3 as output are evaluated. Also, inputs and output ranges are normalized between -1 and 1 . **Table 2** shows fuzzy logic rules. It is clear that arrangements of rules are symmetric and have an opposite behavior of current error and the evaluated error derivative [index 1].

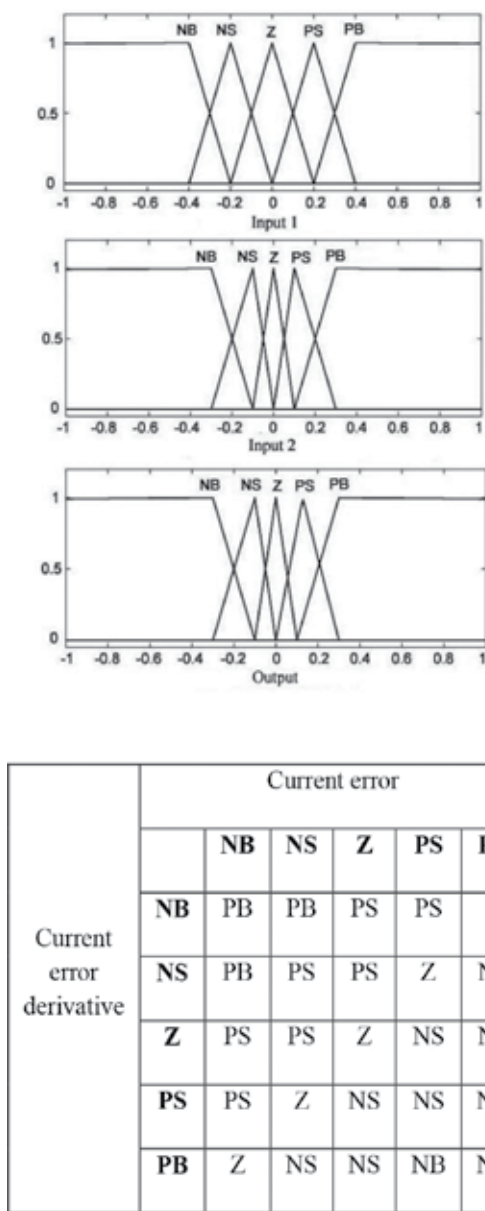


Figure 16. Fuzzy logic member functions: input 1, current error; input 2, current error derivative; output, compensated current.

$$\text{input 1 : } \mu_{i1}(x) = \{0,1\} \tag{54}$$

$$\text{input 2 : } \mu_{i2}(x) = \{0,1\} \tag{55}$$

$$\text{output: } \mu_o(x) = \{0, 1\} \quad (56)$$

where $\mu_{i1}(x)$, $\mu_{i2}(x)$, $\mu_o(x)$ are membership functions.

7. Simulation results

7.1. Nonlinear model simulation of SRM

The proposed methods are simulated by MATLAB/ SIMULINK, where the parameters of SRM are listed in **Table 2**. In these simulations, turn-on/turn-off angles have been selected for minimizing the torque ripple and optimizing the speed estimation.

$L_{\min} = 8 \text{ mH}$	Minimum inductance each phase
$L_{\max} = 60 \text{ mH}$	Maximum inductance each phase
$\beta_r = \beta_s = 30^\circ$	Rotor/stator pole angle
$\Delta I = 0.2 \text{ A}$	Hysteresis current band
$\Omega = 1500 \text{ rpm}$	Speed of motor
$V = 150 \text{ V}$	Motor voltage
$I = 10 \text{ A}$	Motor current
$N_s = 6$	Number of stator poles
$N_r = 4$	Number of rotor poles
$N_{\text{ph}} = 3$	Phases number
$R = 1.3 \Omega$	Resistance each phase

Table 2. SRM parameters.

Figure 17 illustrates the simulation results without FLCC for $\omega = 1500 \text{ rpm}$ with PI speed controller. These results show the speed waveform, one phase current and total torque.

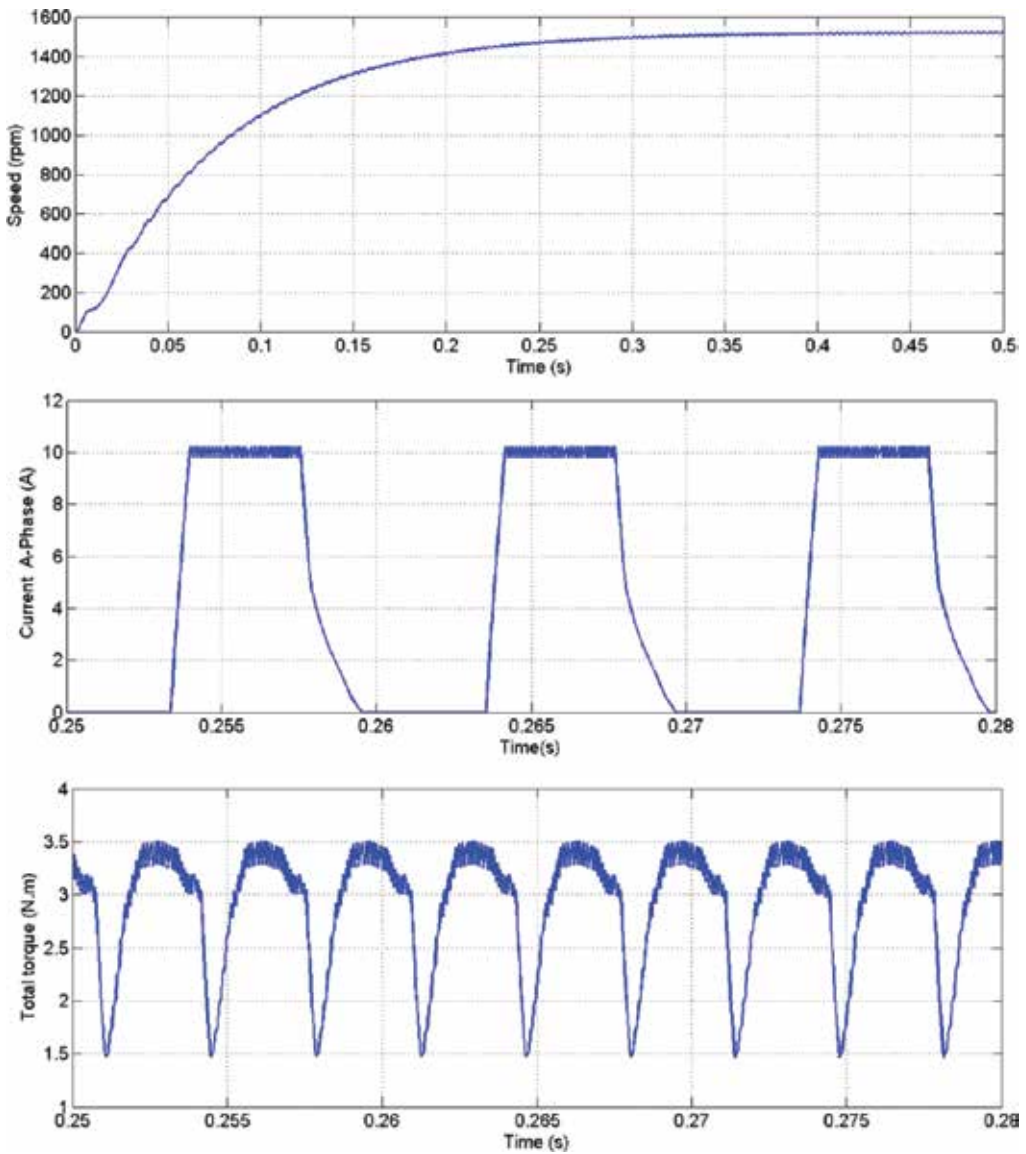


Figure 17. Simulation results of SRM at 1500 rpm without FLCC.

7.2. Torque ripple minimization

Figure 18 shows the simulation results with FLCC for $\omega = 1500$ rpm. These results show the one phase current wave form and total torque. In Figure 18, minimum torque and maximum torque are 2.8 and 3.2 N m ($\Delta T = T_{\max} - T_{\min} = 0.4$ N m), respectively. As it can be seen, phase current waveform is varied for torque ripple reduction.

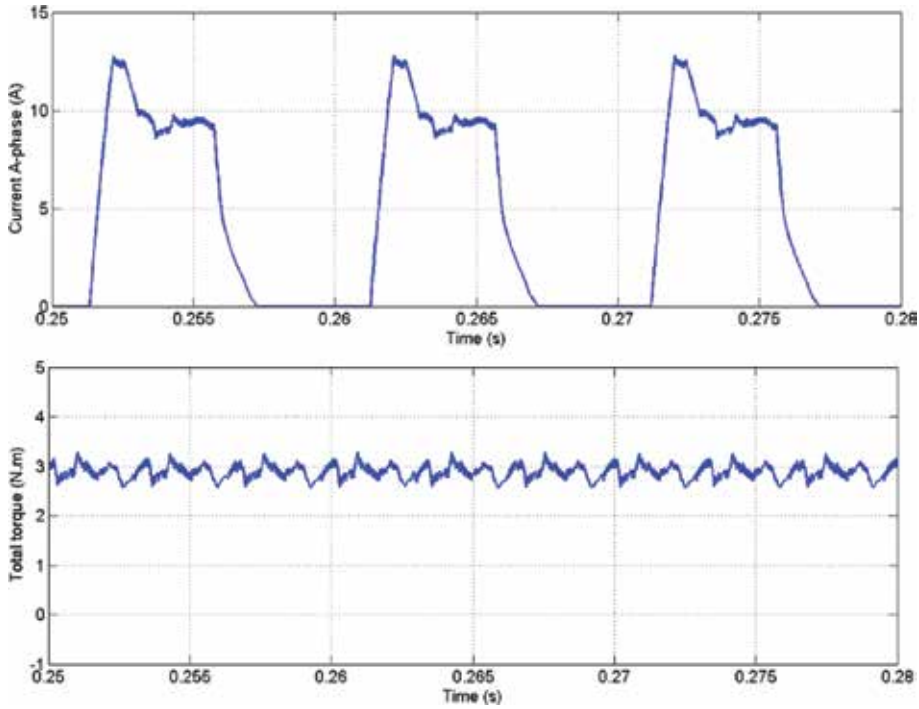


Figure 18. Simulation results of TRM at 1500 rpm with FLCC.

7.3. Acoustic noise reduction

Figure 19 illustrates the simulation results ANR with FLC for $\omega = 1500$ rpm. These results show the one phase current wave form, one phase radial force, and total torque, respectively. In **Figure 19**, FLC optimizes reference current by means of reduction of maximum radial force and keeping the maximum torque and torque ripple minimization. It can be observed that the maximum radial force, the maximum torque, and the torque ripple are 2×10^6 N, 3.5 N m, and ($\Delta T = T_{\max} - T_{\min} = 0.7$ N m), respectively.

7.4. Estimation of rotor position and speed with HO

Finally, **Figure 20** illustrates the simulation results of estimation of rotor position and speed at 50, 100, 500, 1000, 1500 rpm with HO. In **Figure 20**, speed and estimated speed, position and estimated position, and speed estimation error waveform for a wide speed range are shown.

7.5. Speed estimation by torque estimation

Figure 21 illustrates the simulation results for 1500 RMP. In **Figure 21**, estimated speed and actual speed, estimated torque and actual torque, and compensated current are exhibited.

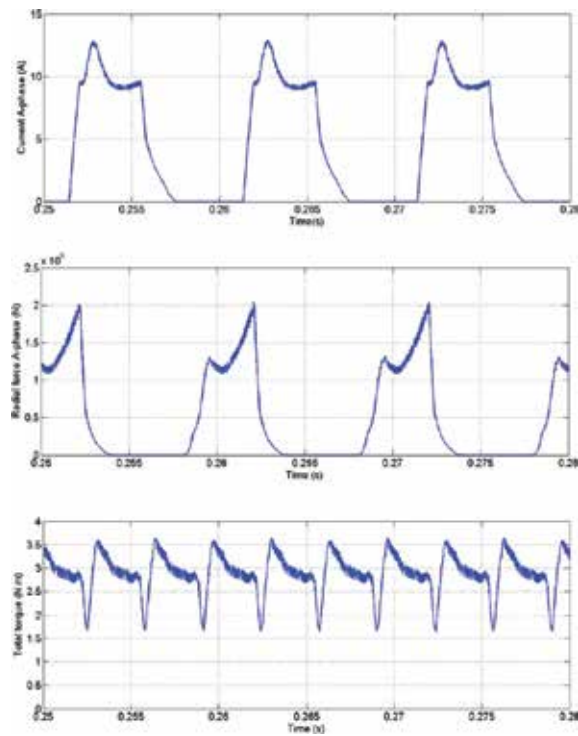


Figure 19. Simulation results of ANR at 1500 rpm with FLC.

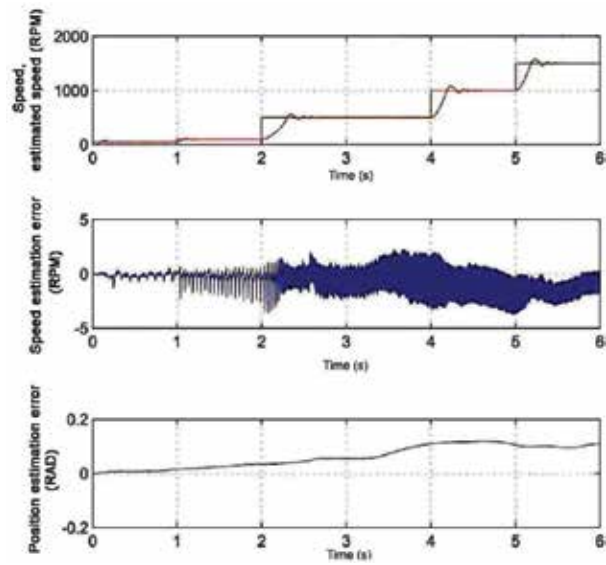


Figure 20. Simulation results of estimation of rotor position and speed at 50, 100, 500, 1000, and 1500 rpm with HO.

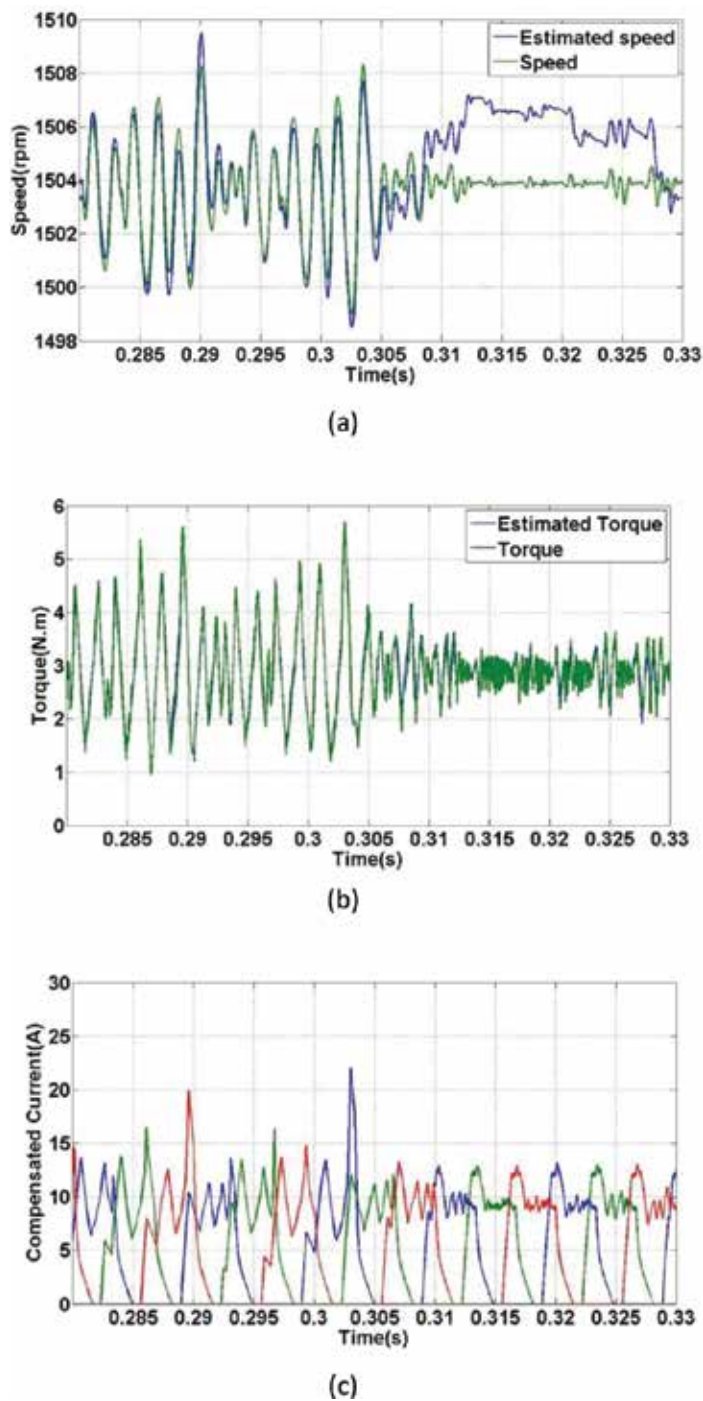


Figure 21. Simulation results of sensor-less SRM drive at 1500 rpm. (a) Estimated speed and actual speed. (b) Estimated torque and actual torque. (c) Compensated current.

8. Conclusion

This study has introduced a new phase current waveform by the 6/4 pole SRM which analyzed and simulated by means of torque ripple and acoustic noise reduction in the SRM drives. The control scheme is performed on the current profile while torque is lower than rated torque and then radial force is maximized. When both torque ripple and radial force are maximized, highest acoustic noise level can be obtained. Nonlinear relations between torque ripple and radial force on the current show that main factor in the torque ripple minimization and acoustic noise reduction is current profile. In this chapter, torque ripple and radial force have been optimized by deformation of the phase current profile by using FLC. Also, a hybrid observer scheme is described. It uses only phase currents and voltages that can be easily measured by motor terminals to estimate rotor position, speed, without extra hardware, which makes it economized. Also, online updates of gains and automatic observer selection are very effective on the speed estimation errors and position estimation errors at steady state. Moreover, current profile improvement for torque modification is used in indirect speed estimation. Simulation results have demonstrated that the motor vibrations could be reduced by the proposed method, and current waveform optimization in the SRM drive effectively decreases acoustic noise magnitude. It can be obviously seen that the method has good performance in wide range. In addition, this method effectively decreases estimation errors and torque ripple in wide speed range.

Nomenclature

ϕ_j	flux linkage of the j th phase (Wb)
L_j	inductance of the j th phase (mH)
V_j	voltage of the j th phase (V)
i_j	current of the j th phase (A)
R_j	resistance of the j th phase (Ω)
t	time (s)
θ	rotor position (rad/s)
ω	speed (rpm)
$(\partial\phi_j/\partial\theta).\omega$	backward electromotive force (EMF) (V)
$(\partial\phi_j/\partial i_j).\omega$	incremental inductance of the j th phase (mH)
W_c	co-energy (J)
T_j	torque of j th phase (N m)
T_l	load torque (N m)
T_e	electromagnetic torque (N m)

N	phases number
J	moment of inertia (kg.m ²)
l_g	air gap length (m)
L_r	rotor length (m)

Index 1

Fuzzy style: Mamdani (fuzzy logic toolbox)

And method: choose min, prod, or custom, for a custom operation.

Or method: choose max, probor (probabilistic or), or custom, for a custom operation.

Implication: choose min, prod, or custom, for a custom operation.

Aggregation: choose max, sum, probor, or custom, for a custom operation.

Defuzzification: for Mamdani-style inference, choose centroid, bisector, mom (middle of maximum), som (smallest of maximum), lom (largest of maximum), or custom, for a custom operation.

Author details

M. Divandari* and B. Rezaie

*Address all correspondence to: mohammad.divandari@gmail.com

Faculty of Electrical and Computer Engineering, Babol Noshirvani University of Technology, Babol, Iran

References

- [1] T. J. E. Miller. Electronic Control of switched Reluctance Machines. 2001, Reed Educational and Professional Publishing Ltd. Oxford, London.
- [2] M. Divandari and A. Dadpour. Radial Force and Torque Ripple Optimization for Acoustic Noise Reduction of SRM Drives via Fuzzy logic Control. 9th IEEE International Conference on Industrial Applications, Sau Paulo, Brazil, November 8–10, 2010.
- [3] N. Radimov, N. Ben-Hail, and R. Rabinovici. Simple Model of Switched-Reluctance Machine Based Only on Aligned and Unaligned Position Data. IEEE Transaction on Magnetic. 2004; 40(3) 1562–1572.

- [4] K. M. Rahman, B. Fahimi, G. Suresh, A. V. Rajarathnam, and M. Ehsani. Advantages of Switched Reluctance Motor Applications to EV and HEV: Design and Control Issues. *IEEE Transaction on Industry Applications*. 2000; 36(1) 111–121.
- [5] E. Mese and D. A. Torrey. An Approach for Sensor-less Position Estimation for Switched Reluctance Motors Using Artificial Neural Networks. *IEEE Transaction on Power Electronics*. 2002; 17(1) 66–75.
- [6] S. E. Schulz and K. M. Rahman. High-Performance Digital PI Current Regulator for EV Switched Reluctance Motor Drives. *IEEE Transaction on Industry Applications*. 2003; 39(4) 1118–1126.
- [7] H. Gao, F. R. Salmasi, and M. Ehsani. Inductance Model-Based Sensor-less Control of the Switched Reluctance Motor Drive at Low Speed. *IEEE Transaction on Power Electronics*. 2004; 19(6) 1568–1573.
- [8] I. Husain and S. A. Hossain. Modeling, Simulation and Control of Switched Reluctance Motor Drives. *IEEE Transaction on Industrial Electronics*. 2005; 52(6) 1625–11634.
- [9] M. Divandari, A. Koochaki, M. Jazaeri, and H. Rastegar. A Novel Sensor-less SRM Drive via Hybrid Observer of Current Sliding Mode and Flux linkage. *IEEE/IEMDC International Conference Electric Machines & Drives*, Antalya, Turkey, May 2007.
- [10] M. Divandari, A. Koochaki, A. Maghsoodloo, H. Rastegar, and J. Noparast. High Performance SRM Drive with Hybrid Observer and Fuzzy Logic Torque Ripple Minimization. *IEEE/ ISIE International Symposium on Industrial Electronics*, Vigo, Spain, June 2007.
- [11] Yan-Tai Chang and Ka Wai Eric Cheng. Sensor-less position estimation of switched reluctance motor at startup using quadratic polynomial regression. *IEEE Transaction on Electric Power Applications*. 2013; 7(7) 618–626.
- [12] F.-C. Lin and S.-M. Yang. Instantaneous Shaft Radial Force Control with Sinusoidal Excitations for Switched Reluctance Motors. *IEEE Transaction on Energy Conversion*. 2007; 22(3), 629–636.
- [13] M. Takiguchi, H. Sugimoto, N. Kurihara, and A. Chiba. Acoustic Noise and Vibration Reduction of SRM by Elimination of Third Harmonic Component in Sum of Radial Forces. *IEEE Transaction on Energy Conversion*. 2015; 30(3) 883–891.
- [14] Hyong-Yeol Yang, Young-Cheol Lim, and Hyun-Chul Kim. Acoustic Noise/Vibration Reduction of a Single-Phase SRM Using Skewed Stator and Rotor. *IEEE Transaction on Industrial Electronics*. 2013; 60(10) 4292–4300.
- [15] M. Hajatipour and M. Farrokhi. Adaptive Intelligent Speed Control of Switched Reluctance Motors with Torque Ripple Reduction. *Elsevier Journal of Energy Conversion and Management*. 2008; 49(2) 1028–1038.

- [16] M. Divandari, R. Barzamini, A. Dadour, and M. Jazaeri. A Novel Dynamic Observer and Torque Ripple minimization via fuzzy logic for SRM Drives. IEEE/ISIE International Conference on Industrial Electronic, Seoul, South Korea, July 2009.
- [17] R. Krishnan. Switched Reluctance Motor Drives. Modeling, Simulation, Analysis, Design, and Applications, 2001, CRC Press LLC. Florida, USA.
- [18] M. Divandari, B. Rezaie, B. Askari-Ziarati. Estimation of Sensor-less SRM Drive Using Adaptive-Fuzzy Logic Control. IEEE/ElConRusNW Conference, Saint Peterzburg, Russia, 2 Feb 2016.

Novel Applications by Artificial Intelligence

Advantages of Fuzzy Control While Dealing with Complex/Unknown Model Dynamics: A Quadcopter Example

Luis Ibarra and Colin Webb

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/62530>

Abstract

Commonly, complex and uncertain plants cannot be faced through well-known linear approaches. Most of the time, complex controllers are needed to attain expected stability and robustness; however, they usually lack a simple design methodology and their actual implementation is difficult (if not impossible). Fuzzy logic control is an intelligent technique which, on its basis, allows the translation from logic statements to a nonlinear mapping. Although it has been proven to effectively deal with complex plants, many recent studies have moved away from the basic premise of linguistic interpretability. In this work, a simple fuzzy controller is designed in a clear way, privileging design easiness and logical consistency of linguistic operators. It is simulated together to a nonlinear model of a quadcopter with added actuators variability, so the robust operation of the controller is also proven. Uneven gain, bandwidth, and time-delay variations are applied among quadcopter's motors, so the simulations results enclose those characteristics which could be found in reality. As those variations can be related to actuators' performance, an analysis can be driven in terms of the features which are not commonly included in mathematical models like power electronics drives or electric machinery. These considerations may shorten the gap between simulation and actual implementation of the fuzzy controller. Briefly, this chapter presents a simple fuzzy controller which deals with a quadcopter plant as a first approach to intelligent control.

Keywords: Fuzzy control, intelligent control, quadcopter, robust control, simulated uncertainty, artificial intelligence, non-parametric variability

1. Introduction

Power electronics systems are commonly comprised of nonlinear time-varying components which impose an overall complex dynamic behavior. As these systems are commonly integrated together as actuators with electric machinery or other transducers, the resulting plants are hard to analyze, model, and control. As linear models and linear controllers have been thoroughly studied, linear approximations to such systems are desired. However, such simplifications omit some important features of the physical media and lead to higher controller burdens or impossible realization [1]. Consequently, power electronic systems in plants are dealt with using three main methods: (1) facing their complexity by using difficult approaches, (2) simplifying their features and constraining controller design, or (3) finding nonparametric models which approximately capture the system behavior.

As control specifications cannot be followed by traditional controllers, artificial intelligence (AI) can be incorporated to the control strategy to achieve higher goals [2]. AI is a set of computational techniques which try to imitate human brain processing or practices found in nature to solve a given problem. For control systems, these methodologies can provide acceptable, or even improved, performance based on any amount of previous knowledge [3]. The increased usage of AI and its increased acceptance are mostly due to some important characteristics. Firstly, they can deal with uncertainty [4] as they can be adapted to a system's variations, sensor noise, or be defined in an uncertain manner [1]. Secondly, they are versatile, so mixed methods involving two or more AI techniques and traditional methods are readily found in literature as pointed in [2, 5, 6]. Finally, they are general, so their application to different case studies is possible and facilitates their continuous improvement. They have been proven to work properly as a direct substitution for linear controllers (as pointed in [7] on fuzzy systems) and as effective alternatives for controlling nonlinear, and uncertain plants [8].

Fuzzy systems (FS) compose one major part of the intelligent control spectrum. They are based on fuzzy logic first proposed by Zadeh [9], which expands set theory in such a way that an element can belong to a set with some membership value, instead of the traditional binary belongingness of "true" or "false". Although fuzzy control will be later discussed in detail, it is worth mentioning that it provides a way to deal with uncertainty through linguistic "elastic" categories and logic inferential rules. Narrowing the distance from human logic to numerical implementation has permitted control systems to be defined in terms of linguistic control rules from experts [6]. This is possible as fuzzy logic control (FLC) specifically converts linguistic rules into a nonlinear mapping [10], allowing dynamic modeling and controller description to be done as simple linguistic statements. The previously presented point of view related to designer experience represents only the most basic approach to FLC; however, it stands as an effective control technique which requires little knowledge about the plant [10, 11].

There are more specific reasons why FLC can be preferred over or together with traditional techniques. Linder and Shafai [12] specify that a quantitative controller's refinement tuning cannot be properly performed, as there is no guidance on the meaning of each parameter. They specify that the controller's refinement tuning can be properly executed when the controller

is built over linguistic rules that preserve their actual meaning. They also say that achieving a quantitative control of nonlinear systems is difficult and not quite realizable. This last statement is confirmed by Precup and Preitl [7] who use a model-based FLC constrained by H-norms. FLC can be implemented despite the complexity as commented by Khan et al. [13] who presented a fuzzy controller tuned through genetic algorithms: “...the unnecessary mathematical rigorosity, preciseness and accuracy involved with the design of the controllers have been a major drawback”. They show that if FLC cannot cope with specific desired controller characteristics, there are intelligent ways in which the controller can be tuned, avoiding experts’ possible limitations and preserving interpretability. There is no intention from the authors of this chapter to discredit advanced traditional control techniques; however, there are many works which point to real benefits of using intelligent control.

FLC has suffered of criticism for lacking a systematic design approach and a stability analysis method; in addition, the high cost of setting up a FLC as a replacement of an industrial standard PID has discouraged its usage [10]. The PID controller has been used not only as a standard controller but also as a reference for comparison when it comes to evaluating a controller’s performance. Most control designers are familiar with PID, which is taken as an easy alternative and a starting point when facing a given plant. The familiarity to PID controller, its known performance, and its relative implementation feasibility have obscured some of the clear benefits of FLC. Nevertheless, Nithya et al. [14] shows that FLC exhibits an improved performance over PID controllers when dealing with first-order plus dead time systems; Gupta et al. [11] remark that nonlinear characteristics make PID unsuitable while intelligent control can cope with control specifications; and Linder and Shafai [12] and Kalavathi and Reddy [6] say that FLC greatly outperforms PI/PID controllers. In some cases, FLC is becoming an immediate choice when dealing with nonlinear, time-delayed, and high-order systems [15]; parameter variance has also been properly tackled by FLC [7]. Altogether, it is accepted as a better alternative than its PID counterpart [4]. As said by Precup and Preitl about fuzzy control: “The use of linguistic rules and approximate reasoning is a simple and easy way to solve concrete control problems. Sophisticated models and control design tools are not necessary. It is sometimes the only way to begin to approach the control of a complex, uncertain, or ill-defined plant” [7].

Some efforts have also been made to expand FLC application by finding similarities to traditional controllers, providing model-based approaches, integrating with other traditional/intelligent techniques, and incorporating features like adaptability to FLCs. Some examples of the aforementioned directions can be found in works like [2] where the equivalence of a FLC with a PID controller together with a relay is explained. Self-regulated fuzzy controllers which incorporate adaptability are also found in the literature as in [4], and robustness enhancement has been achieved by merging fuzzy logic with sliding-mode control as presented in [16] and by Feng [10] who presents a survey on model-based FLC. However, FLC is known as a robust alternative on its own merits [5]. Measuring noise can be incorporated directly if FLC is designed based on type-2 fuzzy logic [1], and simple applications are possible in industrial standard controllers as reported in [4] where an OMRON PLC is used.

Fuzzy logic, as a fundamental part of intelligent control, can face complex systems with relative ease, still providing robustness and logical interpretability. Quadcopters are a great example

of such systems which integrate uncertainty and parameter variability, electric machinery, and power electronics; due to their nature, they seem as a good test-bench to evaluate FLC performance. Quadcopters have gained some field in the research area as they are particularly well suited for unmanned tasks such as search and rescue, surveillance, remote inspection, and as military units [17]. These tasks involve parameters which are not easily obtained or estimated, and can result in uncertain modeling ([17, 18]) which must be alleviated by dependable controllers. Altogether, quadcopters represent a challenge for control designers nowadays [16].

Most of the time, mathematical models cannot fully represent the physical reality of a given plant, and in the case of quadcopters, the inner processing, power electronics, and electric machinery are often assumed not to impose restrictions to system performance. Actually, the quadcopter's motor information is normally unknown, so approximation techniques are needed as in [17]; their response to input desired values is normally assumed to be immediate, repeatable, and reliable even when complete studies about BLDC motors have shown the opposite as in [11]. All aforementioned subsystems are complex and would need a deep mathematical analysis to provide valid simulated results when integrated. Achteik et al. [18] propose a non-intelligent controller and specifically say that quadcopter modeling must avoid simplifications like linearization and that algorithms must be exact as the update rates of processing units are low. Li and Song [5] also identify the overall quality of the model as problematic for control design. Actual implementation also requires the measured data to be "clean," and sometimes redundancy is recommended as noise is unavoidable [19]. It is quite obvious that no control technique will tackle all of these problems at once due to increased complexity; however, model simplification together with uncertainty considerations could lead to a valid solution focusing on robustness against model uncertainties and sensors noise.

In this spirit, many works have simulated, tested, and compared traditional simple, complex, and intelligent techniques over quadcopter devices and models. A short survey about non-intelligent control techniques applied to quadcopters was presented by Li and Song [5] specifically covering the PID, LQR, H, sliding-mode, feedback linearization, adaptive control, and backstepping techniques, presenting their individual benefits and drawbacks. An adaptable controller based on a real-time quadcopter model was tested by Achteik et al. [18] showing great performance even with complete parametric ignorance; Mirzaei et al. [16] present a fuzzy indirect adaptive sliding-mode controller which provide robust performance toward noise and plant uncertainties.

It is not the main aim of this chapter to present a novel control technique or to solve all aforementioned issues regarding quadcopter control, but to test the effectiveness of simple FLC to face this complex plant, focusing on the effects of assuming quadcopter's subsystems to be non-ideal. As most traditional simple controllers require assumptions about the specific quadcopter dynamics, or limitations on the controllers to provide a mostly hovering behavior, the approach of this work is to show how linguistic interpretability can cope with less conservative control specifications while ensuring a certain amount of inherent robustness. For this purpose, a fuzzy controller will be designed and simulated together with an ideal quadcopter model so that a robustness comparison can be made later, when the processing

unit, power electronics, and electric machinery are assumed to incur a time delay and parametric variance. This analysis provides some insight to the “natural” benefits of intelligent control and its great potential to confront complex scenarios (besides its criticized heuristic approach), while presenting a simple and effective way to deal with quadcopter dynamics with some commentary facing actual implementation of the controller.

Nevertheless, fuzzy control has also been used from more formal points of view which consider model-dependency, self-tuning, stability analysis, hybrid methodologies, etc. Chen et al. [20] present some major ways to incorporate FLC on robust control study area; Li and Ruan [4] show a very simple way to provide adaptability to a FLC, while Li et al. [2] provide adaptability to a PID controller through fuzzy logic and Chen et al. [21] to a smith predictor. Complex systems can also be modeled through Takagi-Sugeno (TS) fuzzy models [10] and some plants have been controlled by TS controllers and TS observers as shown in [3]. TS fuzzy systems are also behind gain scheduling topologies like those presented in [7] and [22]. Other intelligent techniques can be incorporated to enhance the design process, as in [13] where genetic algorithms are used to tune the FLC to guarantee optimal performance. Gupta et al. [11] show how FLC has been merged with artificial neural networks and traditional PID controllers to control BLDC motors. Lastly, Nithya et al. [14] show tuning schemes, auto-tuning, and model-based tuning FLC for first-order plus dead time models (FOPDT) in their cited literature; they propose a simple Mamdani FLC to deal with these kinds of systems.

We will use a Mamdani FLC which is opportunely simplistic in nature. It shows how it is adequate to take one of the simplest fuzzy topologies to deal with a complex (and current) plant, and it shows how the control design methodologies can come from basic logic premises. The simple FLC has been improved in many ways, so the perspective shown in this work should be taken as a starting point and as a sufficient statement to favor the usage of intelligent controllers alone or together with other techniques.

2. Fuzzy logic control

Fuzzy logic first intent as expressed by Zadeh [9] was to solve a membership problem. Including elements into a labeled set can be easy and straightforward as long as those elements can be clearly identified as members of the set. However, there will be some elements whose membership is ambiguous or unclear. The real issue here is that a label assigned to a set includes a meaning, subjected to human interpretation. Height measurements will not be easily classified in labeled sets as “Tall”, “Medium”, and “Short” because there is not a clear boundary between them, and their meaning is also subject to the experience and context of who interprets it. Consequently, crisp elements such as height measurements can belong to more than one set. Their membership should not be binary (true/false), but fuzzy $[0, 1]$, representing how much an element belongs to each set attached to a particular linguistic meaning. The membership value of an element can be assigned depending on any kind of considerations regarding its relation to the set’s label; a common way to manage it is assigning a membership function $\mu(x) \in [0, 1]; x \in X$ where X represents the universe of discourse of the

measured variable. A different interpretation involves the set boundaries which, when fuzzy, can be told to be “elastic” [23]. It is similar to a blurry zone where included elements can still explain the set’s meaning to some extent.

The usual language used to describe fuzzy sets has led to many misconceptions which have obscured its acceptance and usage. As said by Zadeh [24] “...fuzzy logic is not fuzzy. Fuzzy logic is precise. Basically, fuzzy logic is a precise logic of imprecision.” In this way, applying fuzzy logic to a control problem permits the designer to make precise decisions about imprecise data. As variables are not forcefully fitted into crisp sets, the consequences of wrong classification are minimized, so it alleviates problems due to noise, disturbance, and overall uncertainty. As fuzzy logic appeals to human reasoning, a mathematical approach will difficultly find its way to describe it. This makes FLC to seem heuristic and restrictive. As this problem has been mitigated due to the research efforts made during the past 30 years, fuzzy logic acceptance on various computer and control science areas keeps growing and expanding the potential of current proposed solutions.

A standard representation of fuzzy sets is done through membership functions mapped on a n -dimensional space, where $(n-1)$ dimensions are devoted to elements’ measured features and one extra dimension to depict their membership toward a given set. For elements described by only one feature, a 2-D mapping as shown in **Figure 1** can be used. A fuzzy set is told to be standard whenever it is described by a normalized (with values ranging $[0, 1]$), convex membership function. Most commonly used shapes for membership functions are piecewise linear (triangular, trapezoidal) and continuous (sigmoid, polynomial, and Gaussian). There are some works devoted to test and compare fuzzy systems built with different types of membership functions which try to elucidate the benefits and drawbacks of each one. Actually, there is not an “ultimate function” to describe memberships. Fuzzy systems are commonly very problem-dependent, and the main objective of placing a given problem in a fuzzy manner (at least for a non-model-based perspective) is to benefit from its uncertainty treatment and its linguistic inference relations. Both of the uncertainty treatment and linguistic inference relations are independent of the shape used.

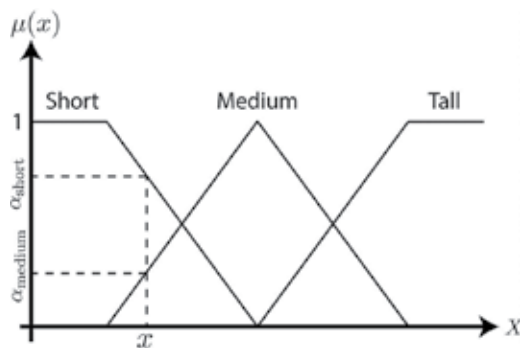


Figure 1. Standard representation of fuzzy sets.

A fuzzy inference system (the foundation of FLC) describes a relation between causes and consequences which follow from the inferential relation of (n -dimensional) input and output concepts. Each concept can be represented by a fuzzy set over a measurable feature in such a way that each input element is related to an output one, so a mapping is derived from linguistic rules. The linguistic rules have the following general form of an implication: *if x_1 is A_1^1 , x_2 is A_2^2 , ... AND x_n is A_n^n , then C_p^1 , C_p^2 , ... AND C_r^m are true.* The preceding statement considers n input elements x classified over their own universe of discourse, each separated by any number of labeled sets A . On the other hand, there are m outputs, each containing any number of consequences of the statement. As a practical example, a statement about quadcopter control can be as follows: If "altitude" is "normal" and "z-direction" is "falling", then "motor1 speed" is set to "increase", "motor2 speed" is set to "increase", "motor3 speed" is set to "increase", and "motor4 speed" is set to "increase". It is noteworthy that any amount of inputs and outputs can be incorporated to a fuzzy inference system, and depending on the desired precision, fuzzy sets can be added or removed from a specific universe of discourse, e.g., adding "rapidly falling" and "slowly falling" to "z-direction", or "increase a lot" to "motor speed". A figure showing the general scheme of fuzzy inference is shown in **Figure 2**.

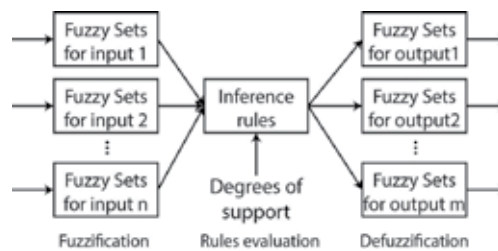


Figure 2. Standard Mamdani inference system.

A fuzzy inference system should never be seen as a set of binary conditionals. Many rules can activate at once (since set boundaries are not crisp, but fuzzy) with an individual rule-strength dependent on the membership values of each element to the labeled sets which compose the rule. A more adequate conceptual approach to fuzzy inference systems is understanding them as a nonlinear overlapping (blending, merging, weighing) of binary conditionals which result in a combined consequence. FLCs are fuzzy inference systems which commonly take the close-loop error as their input and provide a control signal as their output. The first approach of fuzzy logic to a control problem was developed by Mamdani [25] who controlled a steam engine by building the fuzzy inference machine, considering the output consequences to also be fuzzy sets. If consequences are also fuzzy sets, the general statement presented above implies that the control signal, found over a finite universe of discourse, will be also separated by fuzzy sets with meaningful labels. A FLC using this particular perspective about the controller outputs is normally referred as Mamdani-type fuzzy controller. Even if the actual numerical solution for a Mamdani FLC has not been given yet, there is one feature of this type

of controllers which has no evident solution: the controller output should not be a fuzzy set, so how can a crisp value be obtained from a fuzzy inference process?

Before answering the previous question, there is one different approach to look at which needs no fuzzy-crisp output conversion. When consequences are not fuzzy sets, but actual mathematical functions, the FLC is called to be of Takagi-Sugeno-type (TS), referring to the work of Takagi and Sugeno [26] who proposed this approach. Whenever the consequences are defined in this way, the whole FLC will work as a weighted nonlinear blending of those functions. The functions can grow in complexity from a simple constant to a dynamic model, so a TS-FLC is more general than a Mamdani one. It also expands fuzzy systems' potential as they could now perform gain-scheduling controllers or build fuzzy models for dynamic nonlinear systems, for example. Although a TS-FLC seems to absolutely outperform a Mamdani controller, it lacks of full linguistic interpretability. In consequence, the fine tuning of such a controller cannot be performed in terms of logical assumptions, and its construction would limit the fuzzy logic potential (in terms of artificial reasoning).

2.1. FLC algorithm

As noted in the preceding description, most of the inference process can be assumed transparently. However, conversion between crisp and fuzzy variables has yet to be discussed. There are various valid ways to perform such conversion; still, the standard approach will be presented in a more formal manner. Any Mamdani controller will require three steps in the inference process: fuzzification, rules evaluation, and defuzzification as shown in **Figure 2**.

The fuzzification process implies the conversion of crisp input values to fuzzy ones. Fuzzy values are obtained after evaluating the membership of a crisp value to all of the sets pertaining to its universe of discourse. All resulting values are important as they determine how strongly each rule should be executed. If more than one input is considered, there will be AND or OR operators along the inference rules' antecedents, so the evaluated membership of each input must be taken into account properly. Formally, for a universe of discourse where n fuzzy sets have been defined, the membership of an input value x to the sets can be written as $\mu_{A_i}(x) \in [0, 1]; i = 1, 2, \dots, n$. Commonly, μ_{A_i} can be used as a shortcut to express the membership value of a given input to the set A_i .

The rules evaluation is the core of inferential reasoning and it needs the values obtained in the previous step to correctly weight all of the rules in order to provide a logical response. Whenever a rule considers only one input, its triggering strength will be μ_{A_i} without any doubt.

However, for rules with multiple inputs, there must exist some way to evaluate an AND operator for fuzzy values that are non-binary (OR operator can also be used, although it is not discussed in the present work). The different ways of determining the overall strengths of causes of inference rules are called "implication methods," and the work of Zhu et al. [27] can be reviewed for more details.

There are two main ways of evaluating the AND operator in fuzzy rules' antecedents. The first approach implies calculating the minimum of all the membership functions involved in the

rule causes so $\mu_{A_i} \text{ AND } \mu_{B_j} = \min(\mu_{A_i}, \mu_{B_j})$. The second approach uses the product of all rule causes as $\mu_{A_i} \text{ AND } \mu_{B_j} = \mu_{A_i} \mu_{B_j}$. Both approaches are conceptually similar and their usage is mostly designer-dependent as there are no drastic differences between them. It is worthwhile to mention that the purpose of a binary AND operator is to provide a true output only when all its inputs are true. As inputs' validity is not absolute, but graded, both aforementioned methods provide a similar behavior. If only two inputs are being evaluated, the result of the AND operator will be totally true if and only if both inputs are totally true; whenever any of the two inputs is totally false, the AND result will be totally false. For all of the cases in-between, i.e., (0, 1), the resulting validity will be decimated depending on how false the inputs are.

The cogency of the rule's consequences will be as much as the result of the operated inputs, which determine the overall rule strength. So, each rule will output a certain set of weights related to a set of consequences. It is easy to see from this point that different rules can output different weights for a single consequence, so a way to solve this issue is needed. The fundamental principle behind consequences aggregation is that it must satisfy all the rules which required it as an implication output. A simple approach would lead to take the maximum strength among the rules as it will surpass all needed values. From a different perspective, the aggregation can be seen as the sum of all related rules, limited to always be lesser or equal to one.

Lastly, the rules can be also weighted in terms of their importance to the problem. This is an additional weighting value called degree of support and is defined in a closed interval [0, 1]. Each rule can then be heuristically highlighted or obscured as desired. Actually, the tuning process can be based on the selection of proper degrees of support. Each consequence (fuzzy sets in this case) should be then lowered to meet the obtained overall weight. This process can be commonly driven in two different ways. The minimum method trims the membership function of the set by applying $\mu_{C_i} := \min(\mu_{C_i}(x), r_j) \forall x \in X$, where C_i is the consequent fuzzy set and r_j is the resulting overall strength, also called alpha-cut. The product method reduces the magnitude of the set by means of simple arithmetic, so $\mu_{C_i}(x) := r_j \mu_{C_i}$. Finally, the rule evaluation process can be divided in to three substeps: antecedents evaluation, which determines the rule strength due to inputs; rule implication, which logically links the antecedents to the consequences and adds the degree of support; consequences aggregation, which finally sets the overall rule strength and lowers the output sets. For the purposes pursued in terms of fuzzy implication, the resulting weighted consequences' union is the fuzzy output itself: A complex fuzzy set which represents the combination of all rules subjected to all inputs called the aggregated output fuzzy set. However, a controller cannot output a fuzzy set as a valid control signal, so an additional step is needed.

The conversion from a fuzzy set to a crisp value is commonly known as defuzzification. The most popular method used is the Centroid Calculation Method, which obtains a geometrical centre of the aggregated output fuzzy set with respect to the output universe of discourse. It is easy to find a conceptual support for this method as the whole output set represents all the values that are a valid output of the inference system, together with a measure of how possible it is they are the right output (their membership degree). The centroid method will find the

weighted average of all of outputs' possible values in terms of their resulting membership. Consequently, it will find the set's prototype value (a single value which effectively represents the whole set). The computation of this method is not different from a centre of mass calculation as shown in (1).

$$f(y) = \frac{\sum_{i=1}^n \mu_C(y_i) y_i}{\sum_{i=1}^n \mu_C(y_i)} \quad (1)$$

As can be noted above, the whole fuzzy inference system can effectively output a crisp value based on logic statements. The membership functions assigned to each set should effectively represent its meaning to guarantee that the process is consistent with reality or expected performance. Membership functions can be built in many different ways based on linguistic approaches, heuristic considerations, statistical consistency of data, or intelligently with the aid of an additional technique. Altogether, the adjustment of the design parameters presented above leads to a common tuning concept for fuzzy controllers. The number of sets to describe each universe of discourse, their membership functions, the rules assignments together with the methods chosen to evaluate the antecedents and consequences, the degree of support of each rule, and the defuzzification phase are variable parameters which lead to different performances (Fateh [8] shows the predominance of membership function shapes in tuning variations). Adjusting all these parameters in an analytical way or by an automatic process is very difficult, which somehow reduces FLCs acceptance, and their comparison and analysis potentials. Nonetheless, while a human logic-based approach to fuzzy tuning may be heuristic, it would effectively collect the desired features and specifications in a much easier way if compared with traditional approaches. *"Fuzzy control's main intent is to implement symbolic, linguistic control laws based on qualitative models of the plant, and control behaviours"* [12].

It is worthwhile to mention that a TS-FLC algorithm would follow almost the same steps, but will avoid the defuzzification process as its outputs are fully determined as functions. As functions are easier to manipulate through traditional mathematical well-known techniques, TS-FLCs are predominant in today's research trends and many hybrid techniques have been proposed to design, analyze, model, and tune TS-FLC ([4, 7, 6, 10]). On the other hand, improvements to the presented Mamdani's algorithm have been also proposed. For example, Li and Ruan [4] propose a way for the controller to find the inference rules automatically and Kahn et al. [13] fully tune a Mamdani controller by using genetic algorithms.

3. Quadcopter model

Quadcopters have built up much interest in recent years due to their physical characteristics and what their features allow them to do. As stated by Achtelik et al. [18], they can accomplish a range of tasks, including search and rescue, industrial inspection and observation, navigation and sensor data collection, and applications in the field of education and research. Additionally, they have virtually no legal requirements, and require minimal time and effort to begin

a test or flight. We chose to analyze quadcopters because of the range of applications they contain, and because they allow us to show how a complicated multivariable nonlinear system (the quadcopter model) can be controlled by simple fuzzy logic to produce a valid output.

There are many physical variables involved in quadcopters' flight that make them mathematically complex. They are classified as unmanned aerial vehicles (UAV) and as vertical take-off and landing (VTOL) devices. Their movement depends only on four vertical throttle forces whose individual variations and overall momentum conservation can determine the six degrees of freedom (DOFs) which rule any quadcopter [11]. The control design for this device is complicated, as it is for most of flying robots. A quadcopter is an under-actuated system subjected to big dynamic changes related to little parametric variations.

The quadcopter is controlled by changing the rotation speed of the propellers, where two of the propellers are spinning clockwise (CW) and two counterclockwise (CCW). The angular momentum is balanced in all axes if all of the propellers are spinning at the same speed, due to the CW and CCW spinning propellers' counteraction. To rotate about the vertical axis (yaw), for example, the CW spinning propellers are decelerated and the CCW spinning propellers are accelerated. Keeping the overall thrust constant, the quadcopter will turn right as a result of changing angular momentum. The cross (pitch) axis and the front (roll) axis are not affected because the CW and CCW spinning propellers are mounted on opposite sides. To rotate about the roll axis to the left side, for example, the right-side motors are sped up and the left-side motors are slowed down. Yaw is unaffected because of the balanced angular momentum about that axis. Accelerating or decelerating all motors equally results in ascending and descending, respectively, without affecting the pitch, roll, or yaw [18].

Because the quadcopter is complex and it is an under-actuated non-holonomic aircraft with fewer degrees of freedom than control variables, it is essential to simplify and extract the essential factors to create a kinematic model. Moreover, controlling this vehicle requires complex control algorithms, since the six degrees of freedom must be controlled only by four parallel force inputs [17].

The quadcopter model has many limitations that demand more precise control methods. While the model may roughly simulate the motion of the quadcopter, it is under-actuated, nonlinear, strongly coupled, and statically unstable. The model leaves air resistance, body gravity, driving force of propellers, and gyro force up to estimations by constants. This creates uncertainty in the model, which increases the difficulty in designing an accurate controller. Some components even lag or have delays while in use [5]. Additionally, quadcopters are typically propelled by brushless DC motors together with power electronics drivers, which are complicated to model. As a result, the model fails to account for the motor constants correctly [17]. These factors cause problems in modeling, as brushless DC motors, for example, are multivariable and nonlinear systems, where speed and position sensing of traditional machines with Permanent Magnet Brushless DC motors require a high degree of accuracy [11].

In a traditional model, all motor parameters would need to be detailed, making the model infeasible to replicate. In order to tackle power electronics and electric machinery uncertain and unknown behavior, a robust technique is needed. Given the limitations, an approximated

non-parametric model which accounts for time-delays, bandwidth variations, and gain uncertainties was used as an alternative together with a quadcopter traditional model. In addition, we use fuzzy logic with heuristic input-output relations to handle these vague and complex situations.

3.1. Quadcopter model

The specific kinematic model equations we will use in conjunction with fuzzy algorithms are from (2) to (7). A reference figure for equations consistency is shown in **Figure 3**.

$$m \ddot{x} = -k \sum_{i=1}^4 \omega_i^2 \sin \theta \cos \phi - B_x \dot{x} |\sin \theta \cos \phi| \quad (2)$$

$$m \ddot{y} = -k \sum_{i=1}^4 \omega_i^2 \sin \phi \cos \theta - B_y \dot{y} |\sin \phi \cos \theta| \quad (3)$$

$$m \ddot{z} = k \sum_{i=1}^4 \omega_i^2 \cos \theta \cos \phi - mg - B_z \dot{z} |\cos \theta \cos \phi| \quad (4)$$

$$I_L \ddot{\theta} = -B_\theta \dot{\theta} + kl (\omega_1^2 - \omega_3^2 - \omega_2^2 + \omega_4^2) \cos \pi / 4 \quad (5)$$

$$I_L \ddot{\phi} = -B_\phi \dot{\phi} + kl (\omega_1^2 - \omega_3^2 + \omega_2^2 - \omega_4^2) \cos \pi / 4 \quad (6)$$

$$I_Z \ddot{\psi} = -B_\psi \dot{\psi} + b (-\omega_1^2 - \omega_3^2 + \omega_2^2 + \omega_4^2) + I_M (-\dot{\omega}_1 - \dot{\omega}_3 + \dot{\omega}_2 + \dot{\omega}_4) \quad (7)$$

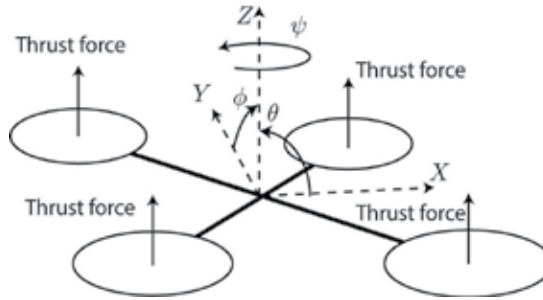


Figure 3. Quadcopter with its own reference frame.

where m is the total mass of the system, l represents the length of the arms, and each propeller speed is represented as ω_i and k is the lift constant. I_L stands for the moment of inertia of X and Y axes while I_z is the moment of inertia with respect to the Z axis. The values B_i represent the air-drag i direction, b is the torque drag, and g stands for gravity. Please notice that all equations are given in terms of the quadcopter's reference frame so a tilt on the θ direction will directly modify the X axis. The model has not been referenced to ground as additional complexity would be added to the model without really aggregating dynamical reality. As the model could be later referenced to earth by simple trigonometric operations, this step is omitted in this work.

4. Testing variability

The presented quadcopter model represents a perfect model of how it would respond if the motor constants, air drag, and other variables were estimated exactly. However, a perfect model does not provide a valid representation of the motion of a real nonlinear system. As noted in the aforementioned text, we deal with lots of uncertainty in which the expected model deviates from the actual behavior of the system. Regarding the quadcopter system, many components exist in lag or delays while in actual use, which contributes to the deviation from the model [5]. Aspects like motor torque and rotational speeds and thrust of the motors vary to some degree and are averaged using test bench data, despite being unequal for each motor [18]. Other methods have also been implemented to counter or reduce uncertainty in systems.

One plain way of battling uncertainty is through proportional-integral-derivative controllers (PID). They are commonly used and useful in control theory and technology because they perform well for a wide class of processes. They give robust performance for specific operating conditions, and are easy to implement with analog or digital hardware, not to mention they are very familiar to engineers. However, plants that have time delays cannot be controlled effectively by a simple PID controller. Fuzzy logic controllers outperform PID controllers because they have no integral accumulator to increase the error of the naturally delayed system [14].

The nonlinearities of a quadcopter, and even of a quadcopter model, are hard to control for a PID as it is a linear controller in nature. Any linear control will be only capable of providing linear control commands to nonlinear error variations, so its reach cannot be extended to the whole operating space of a given plant. Consequently, using a linear controller over a quadcopter will restrict the system to maintaining a smooth hovering undisturbed flight. An additional drawback is that different linear controllers will be needed, as a single tuning will not suffice parametric inequalities among similar components of the quadcopter. This kind of consideration also increases the gap between simulation and actual implementation.

In order to face the difficulties exposed so far, the derived quadcopter model has not ignored relevant physical details apart from environmental wind speed and possible static parametric variations. Such uncertainty has a bigger impact if present at the propellers because controller's outputs are commonly given as thrust commands, assuming infinite bandwidth and no delay.

As explained earlier, the on-board sensors and processing unit, the power electronics driver, and the electric machinery composing each propeller should not be simulated. Incorporating a precise simulation of such elements would immensely increase model's complexity, resulting on an infeasible testbed. Consequently, a simplified model of the propeller is needed which still incorporates time delay, limited bandwidth, and non-ideal gain. In order to account for this deviation and represent uncertainties, we propose using a First-Order Plus Dead Time (FOPDT) model to make the quadcopter simulation more realistic and non-ideal. In addition, a statistical variation of FOPDT parameters can be simulated so the robustness of the FLC is tested.

Similar approaches have been used among literature to simplify model processing while still keeping the desired characteristics. Precup and Preitl [7] represent uncertainties by using a simplified plant model where nonlinear servo systems were used in conjunction with a simplified linearized model of a controlled plant. The controller even includes an inserted variable to notify changes from the nominal expected behavior. Rather than a formal dynamic approach to the system, they use a simplified model similar to FOPDT, as we do instead of simulating power electronics formally. In testing the system, we use a nonlinear model, the opposite of Preitl and Precup [28], whose sensitivity analysis is done over linearized models of the controller. Our tests are not restricted by the simplifications or assumptions they used in their linearized model.

As the introduced variations will result on different operating conditions, the FLC can be measured to be robust to some extent, e.g., a percentage variation of FOPDT parameters. The testing conditions are to be kept the same for all variations so its sensitivity to those changes can be evaluated. Altogether, a simple FLC can be proven to face a complex nonlinear system over a range of operating conditions for a tracking problem. It is clear that the usage of the FLC over the plant will describe an extended operating zone with tolerable performance. If the real plant is supposed to be within that zone, then the controller will be more likely to require minimum modifications for the implementation phase.

4.1. Robustness and boundedness

In design of linear controllers, robustness is achieved when the compensator is designed for a consistent qualitative plant model that encompasses all specified plant perturbations or variations. It is important to recognize the trade-off between maintaining stability robustness and maintaining an increased tracking performance [12]. Among literature, it is possible to confirm that a robust control design will need a previous analysis of plant variations. Variations are commonly bounded within some margins which represent the worst case situations the plant can exhibit. An effective robust controller can manage the whole area described by these variations.

There are ways in which this problem has become more manageable. In [20], a fuzzy observer-based linear controller provides rough tuning to approximate the nonlinear control system, while an optimal H attenuation provides precise tuning to make the robustness optimal. Here, both advanced tracking performance and improved robustness are sought. The paper also illustrates the use of bounding matrices to fit a nonlinear system inside certain bounds through

consideration of an approximate error. They also employ a bounded bandwidth to deal with robust performance control. To create the fuzzy controller, an optimization procedure uses the interior point method to minimize the quadratic boundary while testing stability, so that both controller is optimal and the boundary is minimal. The design of this controller allows the fuzzy algorithm to develop a robustness and function with a reduced effect of external disturbance [20].

In another example, robustness in the case of [4] is assumed to be better in FLC because the controller will adapt itself under varying conditions. However, here no expected performance is guaranteed. In [2], robustness is measured in terms of probability over an inference cell, based on the magnitude of the change of the inference output and the variation of each membership variation, taking the derivative of the inference output with respect to the corresponding alpha cut. This is integrated over the whole inference cell to finally evaluate the robustness [2].

Because control error increases with more conservative bounds, solving the robustness issue implies bounding model parameters. Bounding model parameters means assigning an expected range to the performance of the parameter. Not just any bounds will serve; it is important to find meaningful bounds based on knowledge of the uncertainties and the modelled system. For the quadcopter application, the parameters should be based on simulations or experimental results [18].

Physical constraints can impose boundaries to the controller parameter, so the controller output should not surpass some allowed characteristic of the actuator. In [8], an output scaling factor for motor voltage control does not exceed the maximum voltage allowed for that motor. Ferrero et al. [1] conclude that even measurement uncertainties should be bounded. In [3], they assume their system parameters to be uncertain but bounded to 19% around the nominal value. In [7], the variations are arbitrarily limited.

For a system that is not totally known, a set of systems can enclose it. Preitl and Precup [28] derive sensitivity models related to the step function of the control system for sensitivity analysis. With the plants described in [3], the system is proposed as $A + \Delta$ where Δ is the expected variation of the matrix used. The graphs from [7] demonstrate limits for different design parameters. Additionally, in [22], boundaries are also established so that the model lies within an expected zone. The controller is also designed considering bounded conditions.

The main idea is that a controller which faces a system set will control anything in between. In [12], stability is tested for parametric bounded variations. In [7], two controllers are tuned: (1) a nominal PI controller with respect to the set-point, and (2) a PI controller with respect to the disturbance inputs of the control system. The goal in this research was to blend the controllers to permit a bump-free transfer from one PI controller to the next, and good behavior in the operating zones covered by such blending. This guarantees the achievement of desired control system performance indices, overshoot, settling time, and robust stability [7].

5. Simulations and results

The chapter intentions will be expanded for sake of clarity. There are no models which thoroughly represent a given plant. Such incompleteness can cause many problems which take the implementation far from the expected simulated results. However, going into deep detail about subsystems comprising a plant would be inefficient (or even impossible) due to increased complexity. To tackle this problem, such systems can be approximated by equations which exhibit a similar behavior. As a single equation would hardly fit the system's response, those relegated characteristics can be assumed to be contained somewhere between marginal cases of the simplified equation. In consequence, not a model, but a set of models is proposed to enclose the system reality. A controller can be robustly tuned to face the whole set of simplified systems providing at least two benefits: (1) The real plant can be controlled as long as its features are actually bounded by the set of models, and (2) the marginal control specifications are known from worst-case simulations, so the real model can be controlled inside those borders.

Intelligent controllers, specifically FLC, seem to be a good option when dealing with complex plants and uncertain conditions. Besides FLC being formally analyzed and having model-based solutions that can be easily found in literature, its more remarkable feature is the "translation" of linguistic fuzzy rules and measurements to a nonlinear mapping. A fuzzy controller can be tuned by means of practical observation or experience, almost disregarding the plant's complexity. In this spirit, a complex nonlinear plant was modelled and further varied to show the advantages of a simple heuristically tuned FLC in terms of stability and robustness.

A quadcopter is a UAV which incorporates sensors, a processing unit, power electronics drivers, and electric machinery. With its own complexity as a flying under-actuated device, its effective control is still a challenge nowadays. In this work, the quadcopter is modelled without the use of linearization processes, and a FLC is heuristically tuned to control it under ideal model conditions. In order to account for the variability present in the control process, a set of FOPDT approximation models are used to curtail actuator performance. As the quadcopter model mainly depends on actuators' speed, many model's parameters are being considered uncertain by varying the FOPDT equation. The FLC original tuning (without variations) is preserved and its performance is analyzed toward percentage variations of motors gains, bandwidth, and discrete time delays. Altogether, a good performance of the FLC would move the process forward to a successful implementation process. Additionally, a simple approach to intelligent control design could be proven to be sufficient for such a varying complex plant.

5.1. Methodology

The whole testbed was programmed in LabVIEW 2015 platform. The quadcopter model was solved through a custom Runge-Kutta of 4th order so that its response can be evaluated step-wise. This numerical method was chosen as further implementation on a restricted-size FPGA

is desired. The FLC was designed through the Fuzzy System Designer tool included as a part of the Control and Simulation toolkit for LabVIEW. The FOPDT model was discretized and solved as a recurrent equation dependent on the sampling time which, for all tests, was adjusted to 10 ms.

The ideal gains and bandwidths are set to 1 and 50 rad/s. They will be varied by a uniform random number generator limited to predefined relative limits of [0%, 2%, 5%, 10%, 15%] and [0%, 10%, 20%], respectively. The time delays will be managed as integer variations of sensor and actuator lags. For every simulated test, a time window of 20 s will be used and the squared error of all states will be measured. Each test will be repeated 200 times to allow the random generator to describe the variation uniformly and a final average will be computed per state.

A medium-sized quadcopter of 2 kg is considered. Its parameters are shown in **Table 1** below:

Constant	Abbreviation	Value	Units
Gravity	g	9.81	m/s^2
Arm length	l	0.225	m
X/Y moment of inertia	I_L	4.856E-3	$kg\ m^2$
Z Moment of inertia	I_Z	8.801E-3	$kg\ m^2$
Lift constant	k	2.1E-4	$kg\ m/rev^2$
Torque drag	b	1.14E-7	$kg\ m^2/rev^2$
X drag	B_x	.25	kg/s
Y drag	B_y	.25	kg/s
Z drag	B_z	.25	kg/s
θ drag	B_θ	.01	$kg\ m^2/s$
ϕ drag	B_ϕ	.01	$kg\ m^2/s$
ψ drag	B_ψ	5E-3	$kg\ m^2/s$
Rotors' moment of inertia	I_M	3.357E-5	$kg\ m^2$

Table 1. Considered quadcopter parameters.

For control tuning purposes, quadcopter actuators are assumed to be of infinite bandwidth, equal gain (among all 4 motors), and to exhibit no delay. In **Figure 4**, four main blocks are shown: (1) fuzzy MIMO controller, (2) a converter from tilt to actual motor power, (3) the quadcopter model, and (4) the bounding conditions block.

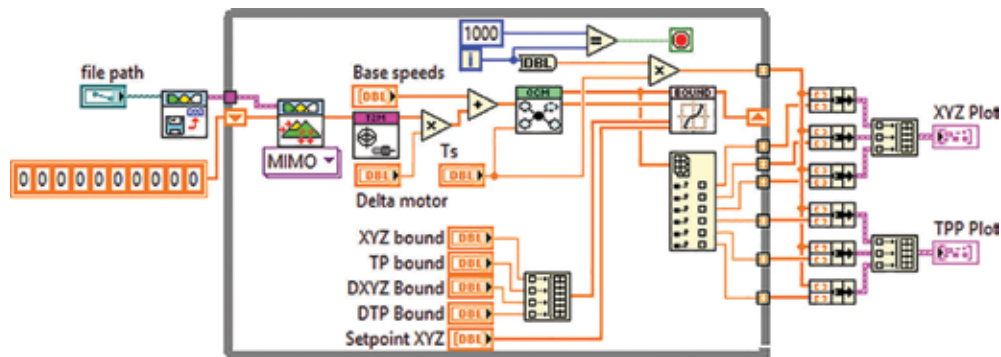
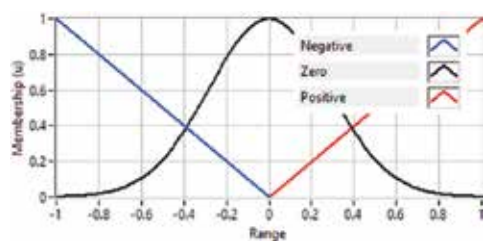


Figure 4. Testbed without FOPDT.

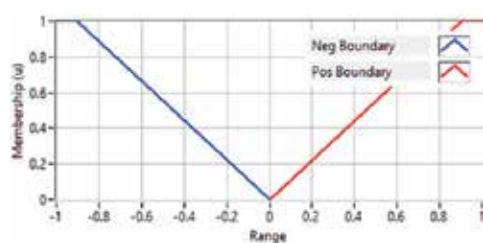
The FLC data are retrieved from a separate file and the control actions are executed in a black-box fashion. In search of a simple and logical controller, its output is designed in terms of forward tilt, right tilt, and overall thrust instead of individual motor power. It is clearer for a designer to identify whether the quadcopter needs to tilt rather than if an individual motor would provide the desired result; in addition, this simplifies the controller rules, as will be explained later. As the quadcopter still needs the individual motor power information, the second block converts the titling control signal into four separated increment or decrement actions. The number of fuzzy sets per input or output has been also arbitrarily determined to be of a maximum of three. The main interest here is showing a simple controller which is easy to understand despite possible better alternatives.

For improved versatility, all controller's inputs and outputs are normalized. This means that all considered universes of discourse have been limited to $[-1, 1]$. Pre- and post-processing scaling steps are needed to fit the actual needed intervals, and that is the main reason why the bounding block has been included. Having all the intervals normalized allows the designer to require a little effort in defining the membership functions as the scaling conditions will tune their reach. It is important that additional to scaling, a coerce condition is included so all the parameters can be represented within their universe of discourse. This implies that all the possible values are bounded (pre-known) by the designer.

In this case, the position control is attained by using one single FLC which requires the following inputs: X, Y, and Z error signals; X, Y, and Z speed signals; θ and ϕ angles; and $\dot{\theta}$ and $\dot{\phi}$ angular velocity signals. These inputs can recreate a basic proportional-derivative FLC; however, instead of taking the error's derivative, the velocity measurements are used so the rules are clearer. All the inputs are presented in **Table 2**.



Input membership functions for X, Y, and Z errors. The three of them have the exact same description.

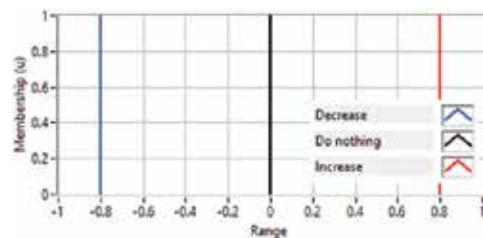


Input membership functions for:

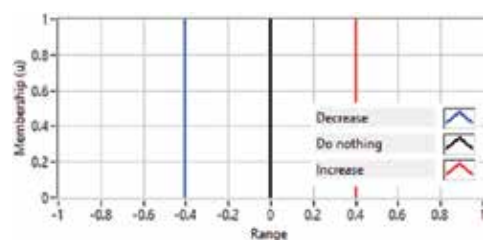
- X, Y, and Z velocities
- θ and ϕ angles
- $\dot{\theta}$ and $\dot{\phi}$ angular velocities

They are meant to limit the speed of the quadcopter, so they are named as “boundaries”. The seven of them have the exact same description.

Table 2. Input membership functions.



Output membership functions for “forward tilt” and “right tilt”. The two of them have the exact same description.



Output membership functions for “all propellers”.

Table 3. Output membership functions.

The outputs of the controller were also furnished to be similar among them and are shown in **Table 3**. The tilt information is divided into two as there are two angles of interest with respect to the quadcopter. An additional output named “all propellers” permits the system to perform homogeneous changes in the Z axis. All outputs’ membership functions are defined using singletons; they are functions which only take the value of 1 at a particular point. As defuzzification is made based on weighing the influences of these functions, the resulting system is equivalent to a TS-FLC with constant outputs. Thus, each control signal will be a smooth transition between those constants.

As the motor inputs for the model are given on an interval $[0, 155]$, the outputs of the FLC were multiplied by 10 and summed to a base speed. In this way, the fuzzy controller is providing correction variations to a predefined thrust which is close to the equilibrium force of the quadcopter. This computation permits the testbed to start from a condition where the vehicle is assumed to be already on flight.

Despite of the number of inputs, only 15 rules are needed to set the necessary logic to control the quadcopter. Nine of them are of a compensating nature and 6 are restrictive:

- Compensating nature:

- IF 'Zerr' IS 'Negative' THEN 'All propellers' IS 'Decrease'
- IF 'Zerr' IS 'Zero' THEN 'All propellers' IS 'Do nothing'
- IF 'Zerr' IS 'Positive' THEN 'All propellers' IS 'Increase'
- IF 'Xerr' IS 'Negative' THEN 'Tilt Right' IS 'Decrease'
- IF 'Xerr' IS 'Zero' THEN 'Tilt Right' IS 'Do nothing'
- IF 'Xerr' IS 'Positive' THEN 'Tilt Right' IS 'Increase'
- IF 'Yerr' IS 'Negative' THEN 'Tilt Forward' IS 'Decrease'
- IF 'Yerr' IS 'Zero' THEN 'Tilt Forward' IS 'Do nothing'
- IF 'Yerr' IS 'Positive' THEN 'Tilt Forward' IS 'Increase'

- Restrictive nature:

- IF 'DZ' IS 'Neg Boudary' THEN 'All propellers' IS 'Increase'
- IF 'DZ' IS 'Pos Boundary' THEN 'All propellers' IS 'Decrease'
- IF 'DX' IS 'Neg Boundary' OR 'Theta' IS 'Pos Boundary' OR 'DTheta' IS 'Pos Boundary' THEN 'Tilt Right' IS 'Increase'
- IF 'DX' IS 'Pos Boundary' OR 'Theta' IS 'Neg Boundary' OR 'DTheta' IS 'Neg Boundary' THEN 'Tilt Right' IS 'Decrease'
- IF 'DY' IS 'Neg Boundary' OR 'Phi' IS 'Pos boundary' OR 'DPhi' IS 'Pos Boundary' THEN 'Tilt Forward' IS 'Increase'

- IF 'DY' IS 'Pos Boundary' OR 'Phi' IS 'Neg Boundary' OR 'DPhi' IS 'Neg Boundary' THEN 'Tilt Forward' IS 'Decrease'

The FLC tuning is done through a trial-and-error process over the scaling factors of all inputs and outputs. The system is then configured as shown in **Table 4**.

X, Y, and Z errors	0.5
θ and ϕ boundaries	0.08
X, Y, and Z speed boundaries	0.48
θ and ϕ angular velocity boundaries	0.3
All outputs	10

Table 4. Tuning parameters for the FLC.

The resulting controller performance is shown in **Figure 5** for set-point values of [0.4, 0.3, 0.2] for [X, Y, Z], respectively. The tests regarding FOPDT variability are done for a common set-point of 0 (for which the ideal model exhibits no error operation). All of the tests evaluate how well the system can hold the zero reference, so the computed error accounts for the variations allowed by the controller. The modified testbed is shown in **Figure 6** and the error results in **Tables 5 to 8**, which include test conditions as follows:

- Bandwidth set to 50 rad/s, no sensor extra delay, no actuator delay
- Bandwidth set to 50 rad/s, one extra sensor delay, no actuator delay
- Bandwidth set to 50 rad/s, no sensor extra delay, one actuator delay
- Bandwidth set to 100 rad/s, no sensor extra delay, two actuator delays

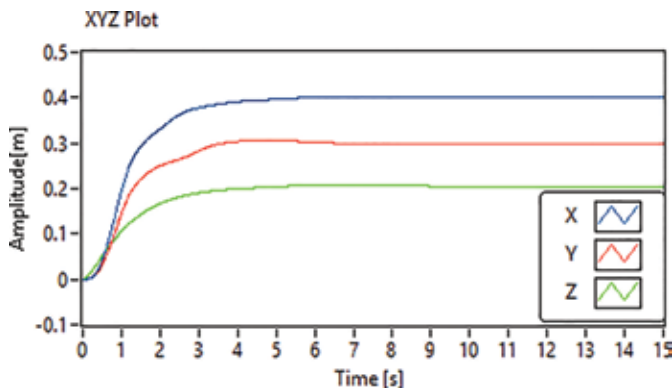


Figure 5. Test for tuned controller.

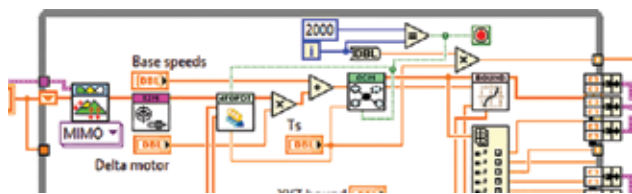


Figure 6. Testbed including FOPDT.

Delay	0 delay														
BW 50	0%					10%					20%				
Gain 1	X	Y	Z	T	P	X	Y	Z	T	P	X	Y	Z	T	P
0%	3.54	6.12	5.79	2.85	4.76	3.56	6.74	5.79	2.88	4.89	3.58	8.65	5.79	2.90	5.14
	E-06	E-08	E-02	E-07	E-08	E-06	E-08	E-02	E-07	E-08	E-06	E-08	E-02	E-07	E-08
2%	9.58	7.61	5.79	1.59	1.19	7.83	7.36	5.80	1.37	1.11	9.00	8.05	5.79	1.58	1.27
	E-05	E-05	E-02	E-06	E-06	E-05	E-05	E-02	E-06	E-06	E-05	E-05	E-02	E-06	E-06
5%	5.01	4.45	5.81	7.52	6.72	4.93	4.87	5.80	7.55	6.90	5.32	5.02	5.85	8.18	7.64
	E-04	E-04	E-02	E-06	E-06	E-04	E-04	E-02	E-06	E-06	E-04	E-04	E-02	E-06	E-06
10%	1.99	1.54	5.84	2.78	2.34	2.00	2.00	5.81	2.90	2.91	1.98	1.89	5.80	2.95	2.80
	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05
15%	4.12	4.39	5.88	6.35	6.43	4.77	4.67	5.81	7.05	6.97	4.77	5.14	5.83	7.60	7.63
	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05

Table 5. Reference results for 0 delay conditions.

Delay	1 extra sensor delay														
BW 50	0%					10%					20%				
Gain 1	X	Y	Z	T	P	X	Y	Z	T	P	X	Y	Z	T	P
0%	7.27	8.19	5.80	3.84	1.05	7.39	8.88	5.80	3.91	1.09	7.67	1.07	5.80	4.07	1.17
	E-06	E-08	E-02	E-07	E-07	E-06	E-08	E-02	E-07	E-07	E-06	E-07	E-02	E-07	E-07
2%	9.55	9.36	5.81	1.83	1.52	8.99	8.66	5.81	1.78	1.41	9.12	7.83	5.81	1.77	1.35
	E-05	E-05	E-02	E-06	E-06	E-05	E-05	E-02	E-06	E-06	E-05	E-05	E-02	E-06	E-06
5%	5.85	5.46	5.79	9.48	8.75	5.86	5.35	5.82	9.44	8.72	5.47	5.36	5.83	9.10	8.60
	E-04	E-04	E-02	E-06	E-06	E-04	E-04	E-02	E-06	E-06	E-04	E-04	E-02	E-06	E-06
10%	2.35	1.98	5.88	3.53	3.12	2.50	2.32	5.85	3.94	3.69	2.32	2.24	5.86	3.53	3.51
	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05
15%	4.13	4.62	5.86	6.46	6.95	5.25	4.49	5.88	7.75	7.21	4.71	4.32	5.85	7.22	7.06
	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05

Table 6. Results considering 1 extra sensor time delay.

Delay	1 actuator delay—20s														
BW 50	0%					10%					20%				
Gain 1	X	Y	Z	T	P	X	Y	Z	T	P	X	Y	Z	T	P
0%	7.37	7.64	5.80	3.41	9.14	7.35	8.37	5.80	3.44	9.57	7.14	1.02	5.80	3.45	9.83
	E-06	E-08	E-02	E-07	E-08	E-06	E-08	E-02	E-07	E-08	E-06	E-07	E-02	E-07	E-08
2%	9.41	9.18	5.81	1.75	1.51	9.51	9.00	5.80	1.85	1.50	1.10	9.04	5.81	2.08	1.63
	E-05	E-05	E-02	E-06	E-06	E-05	E-05	E-02	E-06	E-06	E-04	E-05	E-02	E-06	E-06
5%	5.72	5.01	5.82	8.71	7.92	6.04	5.12	5.81	9.29	8.17	5.80	5.82	5.80	9.36	9.53
	E-04	E-04	E-02	E-06	E-06	E-04	E-04	E-02	E-06	E-06	E-04	E-04	E-02	E-06	E-06
10%	2.27	2.28	5.81	3.50	3.58	2.23	2.00	5.85	3.41	3.14	2.49	2.15	5.82	3.86	3.51
	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05
15%	4.95	4.99	5.88	7.54	7.64	4.56	4.65	5.82	7.21	7.06	4.95	5.18	5.87	7.95	8.14
	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05

Table 7. Results considering 1 actuator time delay.

Delay	2 actuator delay—20s														
BW 100	0%					10%					20%				
Gain 1	X	Y	Z	T	P	X	Y	Z	T	P	X	Y	Z	T	P
0%	1.92	1.21	6.44	9.92	8.97	1.87	1.05	6.52	9.94	9.04	2.05	1.27	6.79	1.06	9.63
	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-01	E-02
2%	2.04	1.22	6.46	9.89	8.95	2.02	1.18	6.49	1.01	9.11	2.04	1.37	6.81	1.04	9.46
	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-01	E-02	E-02	E-02	E-02	E-01	E-02
5%	2.47	1.79	6.49	9.81	8.88	2.41	1.98	6.57	1.02	9.21	2.67	2.29	6.89	1.05	9.43
	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-01	E-02	E-02	E-02	E-02	E-01	E-02
10%	4.12	3.33	6.54	9.62	8.69	3.95	3.50	6.67	9.77	8.81	3.95	3.80	6.85	1.05	9.47
	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-01	E-02
15%	5.52	4.71	6.72	9.36	8.39	5.35	4.85	6.78	9.64	8.55	5.08	4.87	7.04	9.73	8.82
	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02

Table 8. Results considering 100 rad/s bandwidth and 2 actuator time delays.

6. Analysis and conclusions

There are two things to comment about the controller before continuing with data analysis. It does not include an accumulator, so steady-state errors are expected. In addition, the scaling of X, Y, and Z errors could be constrained into a lower margin to force the controller to react toward smaller inputs. However, errors were assumed to be bounded on 0.05 m, so lowering this value would saturate the controller whenever higher set-points are requested. The quadcopter model is started with a base motor power of 76.5, equivalent to the 49% of the total

thrust. This is enough to make the quadcopter to rise before the controller action compensates for the error. For ideal conditions and without the FOPDT model, the system response presents a steady-state error of about 5 mm as shown in **Figure 7**. As motors are assumed to be identical, no variations appear on X and Y axes.

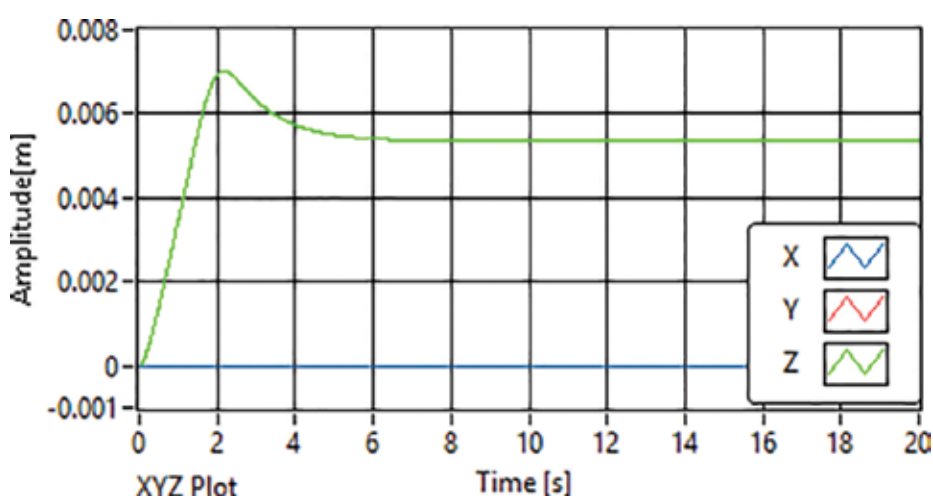


Figure 7. Reference test and steady state error of Z axis.

After performing all the tests with incorporated variability, the controller preserved stability in all cases, except for a 2 delay on the actuator side for which no data were collected. This particular case will be discussed afterwards. In order to ease analysis, four more tables (from **Tables 9 to 12**) are shown below. Each has the following rows: “Gain diff” measures the absolute difference in error calculation due to gain variations; “Col Avg” computes the average error value of a given state; “Delta State” compares each state with respect to the previous lower bandwidth test; and “Delta Ref” compares the “Col Avg” row with those of the test performed with no delays.

Delay	0 delay														
BW 50	0%					10%					20%				
Gain 1	X	Y	Z	T	P	X	Y	Z	T	P	X	Y	Z	T	P
Gain Diff	4.12	4.39	8.10	6.32	6.43	4.77	4.67	1.73	7.02	6.96	4.77	5.14	3.16	7.57	7.62
	E-03	E-03	E-04	E-05	E-05	E-03	E-03	E-04	E-05	E-05	E-03	E-03	E-04	E-05	E-05
Col Avg	1.34	1.29	5.82	2.01	1.91	1.47	1.45	5.80	2.18	2.14	1.48	1.52	5.81	2.31	2.26
	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05
Delta State						8.70%	10.77%	-0.31%	7.44%	10.42%	0.37%	4.96%	0.13%	5.88%	5.64%
Delta Ref	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Table 9. Analysis for reference conditions.

Delay	1 extra sensor delay														
BW 50	0%					10%					20%				
Gain 1	X	Y	Z	T	P	X	Y	Z	T	P	X	Y	Z	T	P
Gain Diff	4.13	4.62	5.50	6.42	6.94	5.24	4.49	7.35	7.71	7.19	4.71	4.32	4.72	7.18	7.05
	E-03	E-03	E-04	E-05	E-05	E-03	E-03	E-04	E-05	E-05	E-03	E-03	E-04	E-05	E-05
Col Avg	1.43	1.45	5.83	2.23	2.22	1.69	1.49	5.83	2.57	2.38	1.54	1.44	5.83	2.38	2.32
	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05
Delta State						14.9	2.54%	0.06%	13.1	6.81%	-9.80%	-3.50%	-0.01%	-8.22%	-2.99%
						2%			9%						
Delta Ref	6.86	12.	0.09%	10.	16.	14.	2.67%	0.45%	18.	11.	4.04%	-5.	0.31%	2.80%	2.28%
	%	14%		87%	13%	67%			21%	63%		72%			

Table 10. Analysis for first test.

Delay	1 actuator delay – 20s														
BW 50	0%					10%					20%				
Gain 1	X	Y	Z	T	P	X	Y	Z	T	P	X	Y	Z	T	P
Gain Diff	4.94	4.99	7.67	7.51	7.63	4.55	4.65	2.19	7.18	7.06	4.94	5.18	7.28	7.92	8.13
	E-03	E-03	E-04	E-05	E-05	E-03	E-03	E-04	E-05	E-05	E-03	E-03	E-04	E-05	E-05
Col Avg	1.58	1.57	5.82	2.43	2.44	1.50	1.45	5.82	2.35	2.24	1.63	1.60	5.82	2.60	2.56
	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05	E-03	E-03	E-02	E-05	E-05
Delta State						-5.31	-8.54	-0.13	-3.03	-8.93	7.97	9.37	0.11	9.36	12.
						%	%	%	%	%	%	%	%	%	51%
Delta Ref	17.	21.	0.01	20.	27.	1.95%	0.22%	0.19%	8.21%	4.65%	10.	5.10%	0.17%	12.	12.
	59%	92%	%	45%	24%						36%			36%	87%

Table 11. Analysis for second test.

Delay	2 actuator delay - 20s														
BW 100	0%					10%					20%				
Gain 1	X	Y	Z	T	P	X	Y	Z	T	P	X	Y	Z	T	P
Gain Diff	3.60	3.50	2.73	-5.59	-5.73	3.48	3.80	2.66	-3.07	-4.88	3.02	3.60	2.48	-9.09	-8.11
	E-02	E-02	E-03	E-03	E-03	E-02	E-02	E-03	E-03	E-03	E-02	E-02	E-03	E-03	E-03
Col Avg	3.21	2.45	6.53	9.72	8.78	3.12	2.51	6.61	9.93	8.95	3.16	2.72	6.88	1.04	9.36
	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-02	E-01	E-02
Delta State						-3.02%	2.42%	1.18%	2.10%	1.91%	1.26%	7.63%	3.94%	4.13%	4.43%
Delta	229	179	12.13%	482	458	202	163	13.82%	456	418	204	168	18.33%	447	413
Ref	3.99%	9.51%		505	437	1.63%	6.89%		177	668	0.72%	7.06%		866	368
				.67%	.12%				.34%	.30%				.68%	.83%

Table 12. Analysis for last test.

It can be seen that the uneven gain variation is responsible for the highest increase of error in all tests. On the other hand, bandwidth modifications do not generate an evident trend as error is sometimes even lower than previous ones. It is clear from this point that bandwidth unevenness is not quite relevant from a statistical point of view; however, angle variation shows an increasing error most of the time. This is due to a constant need to change the propellers' speed because command signals are not fully applied simultaneously. For strongly varying angles, an overall vibration effect will be visible on the quadcopter.

It is worth to mention that the actuator delay will affect the controller performance more strongly than a sensor delay. Those actuator time-related issues may exist due to power electronic drives and integrated close-loops. They can also appear if the motors' information is not collected properly due to sensing or measuring problems on the inner loop and filtering delays. An additional feature observed throughout the simulations is that Z remains almost unaltered, mostly due to its greater independence to little angle variations.

As can be noted, an additional test was performed considering the motors bandwidth to be of 100 rad/s. Stability was not preserved for 2 actuator time delays and 50 rad/s bandwidth, so it was doubled to investigate the effect of improved actuator characteristics. A slow motor performance causes the controller signals to wait some iterations before their effect is fully applied. If the motors vary in bandwidth, some control signals will be delivered on time and some will not, increasing the error seen by the controller. This effect would lead to wrong compensation signals which, if delayed, will hardly keep a stable plant. Remember that the FLC tuning was not modified, so its success will depend on the quadcopter's overall bandwidth.

It is also noteworthy that even if control signals are executed fast enough to keep stability, chattering may appear due to late compensation. This effect (show in **Figure 8**) can be incorrectly identified as noise, which can damage control performance.

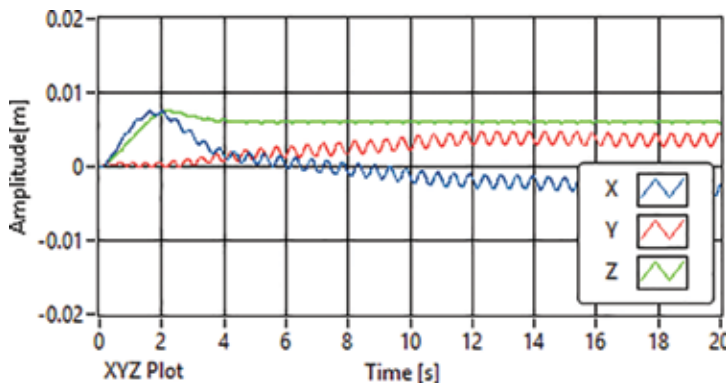


Figure 8. Chattering effect due to time delay of actuators.

A simple FLC has been proven to control a quadcopter nonlinear model subjected to variability (FOPDT) on its actuators. It is a very simple controller which seeks clarity and simplicity and

that still exhibits robust performance when tested under model variations. The controller seems unimpaired by uneven gains and bandwidth variations among quadcopter motors, as well as single time delays present before or after controller block execution. It is possible to say that unevenness on motors' bandwidth does not have a drastic effect. However, an overall low bandwidth will greatly restrict the system's stability.

The controller allowed a maximum 15 mm absolute error among tests and kept the simulated quadcopter stable for motor-unbalanced variations up to 15% in gain and 20% in bandwidth. The controller is capable of dealing with time delay too. However, the actuators' response speeds seem to be a major restriction of this. This analysis permits the FLC system presented to be considered as a simple option for actual implementation. Its tuning and further improvement through more dependable techniques can lead to a realizable controller which observes better specifications.

Author details

Luis Ibarra^{1*} and Colin Webb²

*Address all correspondence to: ibarra.luis@itesm.mx

1 Tecnológico de Monterrey, Mexico City, Mexico

2 Massachusetts Institute of Technology, Cambridge, USA

References

- [1] A. Ferrero, A. Federici, S. Salicone. Instrumental uncertainty and model uncertainty unified in a modified fuzzy inference system. *IEEE Transactions on Instrumentation and Measurement*. 2010;59(5):1149-1157. DOI: 10.1109/TIM.2010.2044257
- [2] H-X. Li, L. Zhang, K-Y. Cai, G. Chen. An improved robust fuzzy-PID controller with optimal fuzzy reasoning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*. 2005;35(6):1283-1294. DOI: 10.1109/TSMCB.2005.851538
- [3] J. Zhang, M. Fei. Analysis and design of robust fuzzy controllers and robust fuzzy observers of nonlinear systems. In: *Sixth World Congress on Intelligent Control and Automation*; 2006; Dalian, China. IEEE; 2006. pp. 3767-3771. DOI: 10.1109/WCICA.2006.1713075
- [4] X.Z. Li, D. Ruan. Comparative study of fuzzy control, PID control, and advanced fuzzy control for simulating a nuclear reactor operation. *International Journal of General Systems*. 2000;29(2):263-279. DOI: 10.1080/03081070008960933

- [5] Y. Li, S. Song. A survey of control algorithms for quadrotor unmanned helicopter. In: 2012 IEEE Fifth International Conference on Advanced Computational Intelligence; 2012; Nanjing, China. IEEE; 2012. pp. 365-369. DOI: 10.1109/ICACI.2012.6463187
- [6] M.S. Kalavathi, C.S.R. Reddy. Performance evaluation of classical and fuzzy logic control techniques for brushless DC motor drive. In: 2012 IEEE International Power Modulator and High Voltage Conference; 2012; San Diego, USA. IEEE; 2012. pp. 488-491. DOI: 10.1109/IPMHVC.2012.6518787
- [7] R.-E. Precup, S. Preitl. PI-Fuzzy controllers for integral plants to ensure robust stability. *Information Sciences*. 2007;177(20):4410-4429. DOI: 10.1016/j.ins.2007.05.005
- [8] M.M. Fateh. Robust fuzzy control of electrical manipulators. *Journal of Intelligent & Robotic Systems*. 2010;69(3):415-434. DOI: 10.1007/s10846-010-9430-y
- [9] L.A. Zadeh. Fuzzy sets. *Information and Control*. 1965;8(3):338-353. DOI: 10.1016/S0019-9958(65)90241-X
- [10] G. Feng. A survey on analysis and design of model-based fuzzy control systems. *IEEE Transactions on Fuzzy Systems*. 2006;14(5):676-697. DOI: 10.1109/TFUZZ.2006.883415
- [11] R.A. Gupta, R. Kumar, A.K. Bansal. Artificial intelligence applications in permanent magnet brushless DC motor drives. *Artificial Intelligence Review*. 2010;33(3):175-186. DOI: 10.1007/s10462-009-9152-3
- [12] S.P. Linder, B. Shafai. Behavior-based robust fuzzy control. In: *Intelligent Control (ISIC)*, 1998. Held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Intelligent Systems and Semiotics (ISAS); 1998; Gaithersburg, USA. IEEE; 1998. pp. 233-238. DOI: 10.1109/ISIC.1998.713666
- [13] S. Kahn, S.F. Abdulazeez, L.W. Adetunji, A.H.M.Z. Alam, M.J.E. Salami, S.A. Hameed, A.H. Abdalla, M.R. Islam. Design and implementation of an optimal fuzzy logic controller using genetic algorithm. *Journal of Computer Science*. 2008;4(10):799-806.
- [14] S. Nithya, A.S. Gour, N. Sivakumaran, N. Anantharaman. Measurement and control of process using soft computing. *Instrumentation Science and Technology*. 2008;36(2): 194-208. DOI: 10.1080/10739140701850944
- [15] R.-E. Precup, R.-C. David, E.M. Petriu, M.-B. Rădac, S. Preitl, J. Fodor. Evolutionary optimization-based tuning of low-cost fuzzy controllers for servo systems. *Knowledge-Based Systems*. 2013;38(Advances in Fuzzy Knowledge Systems: Theory and Application):74-84. DOI: 10.1016/j.knosys.2011.07.006
- [16] Mojtaba Mizaei, Mohammad Eghtesad, Mohammad Alishahi. A new robust fuzzy method for unmanned flying vehicle control. *Journal of Central South University*. 2015;22(6):2166-2182. DOI: 10.1007/s11771-015-2741-1
- [17] J.A. Kunzumann, C.A. Berry. Investigation in the control of a four-rotor aerial robot. In: B. Morrison, M.H. Hamza, editors. *Proceedings of the 2nd IASTED International*

- Conference on Robotics, Robo 2011; 2011; Pittsburgh, USA. ACTA Press; 2011. pp. 181-188. DOI: 10.2316/P.2011.752-055
- [18] M. Achtelik, T. Bierling, J. Wang, L. Höcht, F. Holzapfel. Adaptive control of a quadcopter in the presence of large/complete parameter uncertainties. In: AIAA Infotech at Aerospace Conference and Exhibit 2011; 2011; St. Louis, USA. ARC; 2011. pp. AIAA 2011-1485.
- [19] S. Gupte, P.I.T. Mohandas, J.M. Conrad. A survey of quadrotor unmanned aerial vehicles. In: 2012 Proceedings of IEEE Southeastcon; 2012; Orlando, USA. IEEE; 2012. pp. 1-6. DOI: 10.1109/SECon.2012.6196930
- [20] B.-S. Chen, C.-S. Tseng, H.-J. Uang. Robustness design of nonlinear dynamic systems via fuzzy linear control. IEEE Transactions on Fuzzy Systems. 1999;7(5):571-585. DOI: 10.1109/91.797980
- [21] H. Chen, Z. Zouaoui, Z. Chen. Application of fuzzy logic to reduce modelling errors in PIDSP for FOPDT process control. In: 19th Mediterranean Conference on Control Automation (MED); 2011; Corfu, Grece. IEEE; 2011. pp. 1478-1483. DOI: 10.1109/MED.2011.5983205
- [22] D. Senthilkumar, C. Mahanta. Identification of uncertain nonlinear systems for robust fuzzy control. ISA Transactions. 2010;49(1):27-38. DOI: 10.1016/j.isatra.2009.07.005
- [23] L.A. Zadeh. Fuzzy logic—a personal perspective. Fuzzy Sets and Systems. 2015;281(Celebrating the 50th Anniversary of Fuzzy Sets):4-20. DOI: 10.1016/j.fss.2015.05.009
- [24] L.A. Zadeh. A summary and update of "fuzzy logic". In: Proceedings of the 2010 IEEE International Conference on Granular Computing; 2010; San Jose, USA. IEEE; 2010. pp. 42-44. DOI: 10.1109/GrC.2010.144
- [25] E.H. Mamdani. Application of fuzzy algorithms for control of simple dynamic plant. Proceedings of the Institution of Electrical Engineers. 1974;121(12):1585-1588. DOI: 10.1049/piee.1974.0328
- [26] T. Takagi, M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man and Cybernetics. 1985;15(1):116-132. DOI: 10.1109/TSMC.1985.6313399
- [27] X. Zhu, Z. Huang, S. Yang, G. Shen. Fuzzy implication methods in fuzzy logic. In: Fourth International Conference on Fuzzy Systems and Knowledge Discovery; 2007; Haikou, China. IEEE; 2007. pp. 154-158. DOI: 10.1109/FSKD.2007.327
- [28] S. Preitl, R.-E Precup. Sensitivity study of a class of fuzzy control systems. Periodica Polytechnica, Electrical Engineering. 2006;50(3-4):255-268.

Retrieval of Optical Constant and Particle Size Distribution of Particulate Media Using the PSO-Based Neural Network Algorithm

Hong Qi, Ya-Tao Ren, Jun-You Zhang,
Li-Ming Ruan and He-Ping Tan

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/62446>

Abstract

An improved neural network algorithm was proposed and applied to the inverse radiative problems. A multi-strategy particle swarm optimization was applied to improve the performance of the back propagation multi-layer feed-forward neural network algorithm. Three commonly used particle size distribution (PSD) functions in a one-dimensional particle system were retrieved using the proposed algorithm. In addition, the optical constant was also estimated, and the measurement errors were considered. Results show that the proposed algorithm can be applied to the retrieval of PSDs and optical constant even with measurement errors. Finally, the proposed algorithm was applied to the simultaneous estimation of the PSDs and optical constant using the multi-wavelength and multi-thickness method.

Keywords: PSO-based neural network algorithm, optical constant, particle size distribution, particulate media, inverse radiation problem

1. Introduction

Participating medium is widely used in the petroleum chemical industry, bio-medicine, infrared detection, remote sensing, and many other fields [1, 2]. The particle system is one kind of participation media, such as liquids or gases, with small particles in microns or even nanometer level. Researches on the microphysical properties of particulate media, such as the optical properties and particle size distributions (PSDs), are very important for many engineering

applications, such as power plant, coal-fired furnaces, combustion chambers, or other utilization systems, which will contribute to improve the combustion efficiency [3–5] and environment control [6, 7]. Especially, thermal radiation is the dominant mode of heat transfer in power plant [8], and the optical properties and PSD of fly ash particles are of significant importance in radiative heat transfer analysis [9]. Knowledge of the optical constants of the particulate fly ash is necessary for monitoring of temperature and condition, and prediction of heat transfer through fly ash formed in the pulverized fuel-fired furnace. However, determination of optical constants requires a priori information on the particle number density and size distribution. Based on this fact, uncertainty in the PSD is the principal source of error in determining single particle optical properties. In many cases of practical interest, however, these quantities are not known. In spite of the widely use of high temperature fly ash particles, their refractive indices, particularly in the infrared range, are known insufficiently. This is especially true of the optical properties of infrared particles being considered in the high-efficiency combustors. Accurate values of the optical constants of particles, which are significant functions of wavelength, are of particular interest in radiative transfer analysis in power plant because of their significant influence on heat transfer in such applications. Light scattering analyses of the structure and refractive indices of fly ash particles have been proposed as a means of studying the interactions between fly ash particles, flue gases, and the surrounding atmosphere [10]. Moreover, fly ash is a very promising combustion by-product of the coal-fired power station. For example, it can be added into the cement pastes to improve their fluidity where the PSD is an important control parameter [11]. Generally, optical properties and PSD of participating particulate medium are of great importance in power plant systems and have evoked the wide interest of many researchers.

The optical constant of participating medium particle system cannot be directly measured by experiments [12]. Its direct calculation through the measurable physical parameters is also difficult. So its value is usually obtained by inversion algorithm combining with experimental measurements. Optical constant contains real part n and imaginary part k . These two parts, respectively, represent refraction behavior and absorption behavior of participating media. The optical constants play an indispensable role in determining the radiation properties of a medium. Accurate knowledge of the refractive index over a wide range of wavelength is indispensable for many engineering applications. For instance, combustion diagnostic techniques employing laser light scattering require accurate refractive indices in the visible wavelengths. Various environmental, atmospheric, astrophysical, and military applications rely on data for such particles in both the visible and the infrared wavelength. However, even for the same kind of participating particle system, its optical constants may also be variable because of some factors, such as temperature range, geometry size, and wavelength range. Because of its importance for figuring out the radiative transfer mechanism in the particle system, it has attracted significant attention of scholars all over the world [13–19]. In fact, the accurate determination of the refractive index of dispersed media should be considered today as an unsolved problem, open to research.

To date, the commonly used particle size measurement technique can be divided by its principle into the following categories: microscope method, sieving method, sedimentation,

and ultrasonic method. For the practical problem of interest, the sieving method is rarely used recently because it has low precision and it is extremely easy to damage the samples in the measurement process [20]. With the development of the microscope, the accuracy of the microscope method improves. However, it is very time consuming. Ultrasonic method is more suitable for higher particle concentration due to its better ability of transmission in the participating medium compared to light. However, for some unstable particles systems, the ultrasonic method is not applicable. Therefore, spectral extinction method is more widely applied because of its wide measurement range, relatively simple measuring principle and device [21, 22].

Literature survey reveals that there is no reported work about estimating of optical constants and PSDs using artificial neural network (ANN). An important advantage of this method in comparison with other inverse heat transfer modeling approaches, such as the Gauss–Newton method [23], conjugate gradient method [24], genetic algorithm [25], and ant colony optimization [26, 27], to name a few, is that detailed knowledge of the geometrical and thermal properties of the system (such as wall conductivity and emissivity) is not necessary [28, 29]. In many cases, the measurement of such physical properties is extremely difficult or even impossible. Moreover, to use all conventional inverse methods, the direct problem must be solved firstly. This may lead to significant computing errors and excessively time-consuming calculations. ANN solutions are based on experimental data rather than mathematical governing equations [30]. The direct model needs not be solved repeatedly, which will save considerable computation time. Furthermore, genetic algorithm and other evolutionary algorithms require large populations and involve long computing time for convergence, which make them being inappropriate for many inverse heat transfer problems. In ANN, the only iterative part is in training, which is containing simple mathematical relations. Therefore, the proposed method is much faster than conventional iterative methods.

In this chapter, a particle swarm optimization (PSO) based back propagation (BP) neural network algorithm was developed to estimate the PSD and optical constant of a particle system. As one kind of intelligence optimization algorithms, the BP multi-layer feed-forward ANN algorithm was proposed by D.E. Rumelhart firstly [31]. In complicated systems with several effective input parameters, ANN can be used to predict output data. The BP neural network algorithm is a stochastic heuristic intelligent optimization technique, which was investigated thoroughly [32, 33]. However, there remain some unsolved problems. For example, as the initial weight coefficients are generated randomly, it may lead to the instability of the algorithm. Furthermore, the algorithm may fall into local optimum. Therefore, in the present work, to improve the performance of the BP neural network algorithm, the PSO algorithm was applied.

The remainder of this chapter is organized as follows. First, the principle of the inverse method was introduced briefly. Afterwards, the direct model was presented. Subsequently, the PSDs and optical constant were retrieved separately using the PSO-based BP neural network algorithm. Finally, a technique was proposed to obtain the PSDs and optical constant simultaneously.

2. Inverse method

2.1. BP neural network

An ANN model includes five parts, i.e. input values (x_1, x_2, \dots, x_n) , thresholds of network, weight $(W[w_{1,1}, w_{1,2}, \dots, w_{1,N}])$, transfer function, and output values. The size of weight stands for the importance of input data for forward transmission by indicating the connection degree between the input information and related neurons. The basic unit of neural network is the neuron. In each neuron, the sum of input values is weighted and added with a parameter called bias, and the sum (see Equation 1) is passed through a function, which is called transfer function or activation function. The transfer function calculates the output of a neuron from its input.

$$net_j = \sum_{i=1}^n w_{ij} x_i(t) \quad (1)$$

Some ANN models include several layers. Each layer includes several neurons and performs a simple process on data. The actions of neurons do not directly depend on the summation but a certain threshold. Only when the summation net_j is bigger than the threshold, neurons can be activated by activation function. Otherwise, the neuron will not send signals.

The working process of the BP neural network includes training and prediction. Training is to prepare the network to work for certain system, which is of highly importance. In the training process, the weights are adjusted to obtain better performance of the network. Prediction is to solve the inverse problem by using the prepared network [34]. The detailed principle of the BP neural network is described in Ref. [32], which will not be repeated here.

2.2. The PSO algorithm

The PSO, which has rooted in artificial life and social psychology as well as engineering and computer science, is a random searching technique inspired by bird foraging. It differs from evolutionary computation methods in that the population members, called particles, are flown through the problem hyperspace. The PSO finds the optimum value more quickly than traditional evolutionary algorithms due to the fact that PSO uses a combination of local and global searches with the sharing evolutionary information among the individual particles. Many modifications have been made to improve convergence speed of the basic PSO algorithm and to increase the diversity of particles [35]. The velocity and position of each particle in the basic PSO can be obtained as follows [36]:

$$\mathbf{V}_i(t+1) = w\mathbf{V}_i(t) + c_1 \cdot r_1 \cdot [\mathbf{P}_i(t) - \mathbf{X}_i(t)] + c_2 \cdot r_2 \cdot [\mathbf{P}_g(t) - \mathbf{X}_i(t)] \quad (2)$$

$$\mathbf{X}_i(t+1) = \mathbf{X}_i(t) + \mathbf{V}_i(t+1) \quad (3)$$

where c_1 and c_2 represent two positive constants called acceleration coefficients; r_1 and r_2 are uniformly distributed random numbers in the interval $[0, 1]$. w is the inertia weight factor, which is used to control the impact of the previous velocities on the current velocity. In the present work, w is defined as $w = 0.9 - 0.4 \times t/t_{\max}$ where t and t_{\max} represent the current and maximum generation number, respectively. $\mathbf{X}_i(t) = [X_{i1}(t), X_{i2}(t), \dots, X_{in}(t)]^T$ denotes the present location of particle i , which represents a potential solution. $\mathbf{V}_i(t) = [V_{i1}(t), V_{i2}(t), \dots, V_{in}(t)]^T$ denotes the present velocity of particle i , which is based on its own and its neighbors' flying experience. $\mathbf{P}_i(t) = [P_{i1}(t), P_{i2}(t), \dots, P_{in}(t)]^T$ is the 'local best' in the t th generation of the swarm. The global best position of the swarm is expressed as \mathbf{P}_g .

In the present work, an improved multi-strategy PSO (MSPSO) was applied to BP neural network training [37]. Three improvements were made based on the standard PSO. First, the opposition learning [38] strategy is introduced into every iteration of the PSO algorithm. The idea of opposition learning is that when a new individual particle $\mathbf{X} = (X_1, X_2, \dots, X_n)$ is generated during the evolution process, its opposite position $\bar{\mathbf{X}} = (\bar{X}_1, \bar{X}_2, \dots, \bar{X}_n)$ should be considered to test if it is better than \mathbf{X} . If x_i is in the interval of $[a_i, b_i]$, \bar{X}_i can be calculated by $\bar{X}_i = a_i + b_i - X_i$. Second, the acceleration coefficient c_2 is set to vary with iterations instead of being constant. Generally speaking, c_2 represents social ability of the swarm. In the later stage of the iteration, the society guide ability gradually strengthens, which makes the diversity of the swarm decline. It should properly weaken in the later stage of the algorithm. Therefore, c_2 can be set as follows [37]:

$$c_2 = c_{2\max} - \frac{t}{t_{\max}}(c_{2\max} - c_{2\min}) \quad (4)$$

where $c_{2\max}$ and $c_{2\min}$ are the maximum and minimum of c_2 , which are set as 2.05 and 0.4, respectively, in the present work [37].

Finally, it is known that the standard PSO may be trapped into local optimum as the decreasing of the population diversity. Therefore, a multi-start strategy is introduced to the MSPSO, which means that, when the optimal solution is not updated for S_{\max} iterations, it will force the particle swarm to initialize, but the previous optimal solution is recorded, where S_{\max} is the maximum number of stagnation. Note that during the particle initialization, it still reserves the individual best position and global best position. The detailed model flow chart of MSPSO is shown in **Figure 1**.

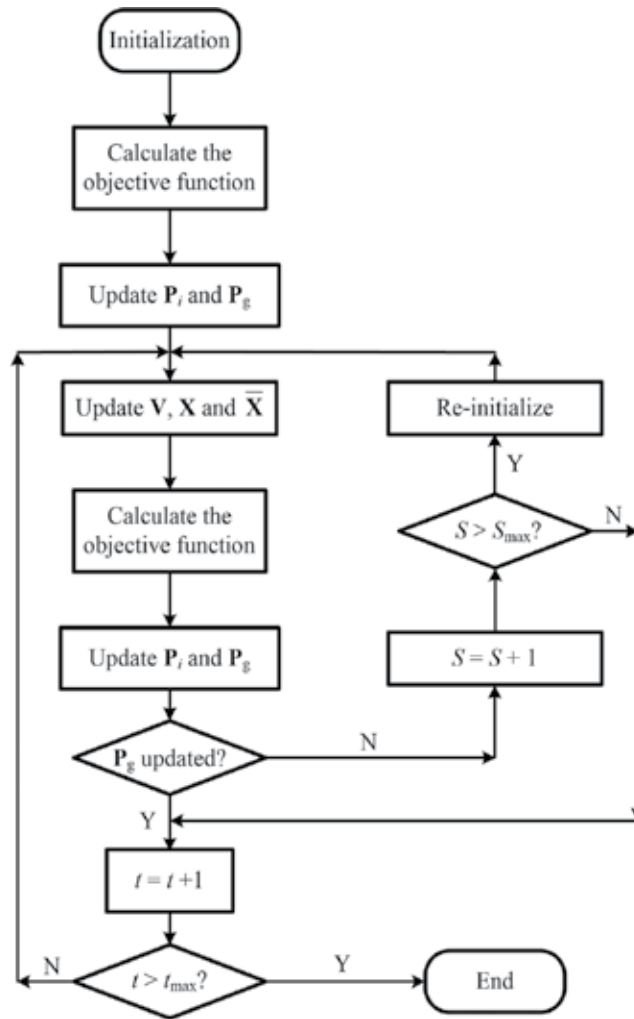


Figure 1. Flowchart MSPSO algorithm.

2.3. MSPSO-BP algorithm

In the present work, the MSPSO is applied to optimize the threshold and weight of the BP neural network. The training process is required before prediction for BP neural network, which is introduced in Section 2.1. Before the optimization of BP neural network, the topological structure of the network should be decided. For a three-layered BP neural network, i.e. to choose the node numbers in input layer, hidden layer, and output layer. The structure of input layer and output layer is mainly determined by the number of input and output data. The node number in hidden layer is of significant importance, which highly affects the training speed and training accuracy. Too many nodes may lead to slow convergence rate and over learning. However, if the node number is not large enough, the network may have small

training precision. The values of the node number can be determined using many different ways [39]. After the node number of the hidden layer is determined, the next step is to decide the weight and threshold values, which are optimized by the MSPSO in the present work. Each particle stands for a possible solution of the optimal weight and threshold.

3. Direct model

A one-dimensional (1-D) absorbing, scattering, and non-emitting particle system was under consideration in the present work (see **Figure 2**). The left side of the system was exposed to continuous wave laser beam of different wavelengths. For the short interaction time and low laser intensity, the media were assumed to be cold without considering the thermal effect caused by the interaction of the laser and media. The radiative transfer equation in a 1-D particle system can be expressed as follows [40]:

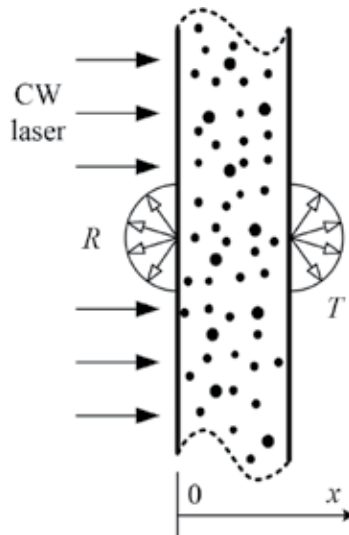


Figure 2. The schematic of a 1D slab absorbing, scattering, and non-emitting particle system exposed to collimated continuous wave laser.

$$\frac{\partial I(x, \theta)}{\partial x} = -(\kappa_\lambda + \sigma_{s\lambda})I(x, \theta) + \frac{\partial_{s\lambda}}{2} \int_0^\pi I(x, \theta') \Phi_\lambda(\theta', \theta) \sin \theta' d\theta' \quad (5)$$

where I is the intensity in direction θ at location x . The absorption and scattering coefficients are denoted by κ_λ and $\sigma_{s\lambda}$, respectively. The subscript λ stands for the wavelength of the incident laser. The scattering phase function $\Phi_\lambda(\theta', \theta)$ represents the probability that radiation, which propagates from the incoming direction θ' , will be scattered into the direction θ .

According to Mie theory, when the sphere particle is irradiated by a laser beam, the extinction, scattering, and absorption efficiency can be calculated by the following equation [41]:

$$Q_{\text{ext}} = \frac{2}{\chi^2} \text{Re} \left[\sum_{n=1}^{\infty} (2n+1)(a_n + b_n) \right] \quad (6)$$

$$Q_{\text{sca}} = \frac{2}{\chi^2} \sum_{n=1}^{\infty} (2n+1) (|a_n|^2 + |b_n|^2) \quad (7)$$

$$Q_{\text{abs}} = Q_{\text{ext}} - Q_{\text{sca}} \quad (8)$$

where the extinction, scattering, and absorption efficiency are denoted by Q_{ext} , Q_{sca} , and Q_{abs} respectively. Re is the real part of complex number, χ represents the scale parameter that can be expressed as $\pi D/\lambda$ for spherical particles. D represents the diameter of particles. λ stands for the wavelength of incident laser. Mie scattering coefficients are denoted by a_n and b_n , which can be written as follows [41]:

$$a_n = \frac{\psi'_n(m\chi)\psi_n(\chi) - m\psi_n(m\chi)\psi'_n(\chi)}{\psi'_n(m\chi)\xi_n(\chi) - m\psi_n(m\chi)\xi'_n(\chi)} \quad (9)$$

$$b_n = \frac{m\psi'_n(m\chi)\psi_n(\chi) - \psi_n(m\chi)\psi'_n(\chi)}{m\psi'_n(m\chi)\xi_n(\chi) - \psi_n(m\chi)\xi'_n(\chi)} \quad (10)$$

where $m (= n + ik)$ represents the optical constant of particle system. $\xi_n(\chi) = \psi_n + i\chi_n$, $\psi_n(\chi)$, and $\chi_n(\chi)$ are Ricatti-Bessel functions. The details of Mie theory are well demonstrated in Ref. [41].

The absorption coefficient κ_λ and scattering coefficient $\sigma_{s\lambda}$ of particle system can be calculated by the following equations:

$$\kappa_\lambda = \int_0^\infty C_{\text{abs},\lambda}(D) N(D) dD \quad (11)$$

$$\sigma_{s\lambda} = \int_0^\infty C_{\text{sca},\lambda}(D) N(D) dD \quad (12)$$

where $C_{\text{abs}, \lambda}$ and $C_{\text{sca}, \lambda}$ stand for absorption and scattering cross section, respectively. For sphere particles, absorption and scattering cross section can be calculated by using Mie theory [41]; $N(D)$ represents the number density of particles with diameter D .

It is known that the PSDs can be described by certain distribution function, among which the commonly used parameter distribution function is as follows [42]:

$$f_{\text{R-R}}(D) = \frac{\sigma}{D} \times \left(\frac{D}{\bar{D}}\right)^{\sigma-1} \times \exp\left[-\left(\frac{D}{\bar{D}}\right)^{\sigma}\right] \quad (13)$$

$$f_{\text{S-N}}(D) = \frac{1}{\sqrt{2\pi}\sigma} \times \exp\left[-\frac{(D-\bar{D})^2}{2\sigma^2}\right] \quad (14)$$

$$f_{\text{L-N}}(D) = \frac{1}{\sqrt{2\pi}D\ln\sigma} \times \exp\left[-\frac{(\ln D - \ln \bar{D})^2}{2(\ln \sigma)^2}\right] \quad (15)$$

where \bar{D} represents the characteristic diameter parameter, and σ is the dispersion ratio. The volume frequency distribution is denoted by $f(D)$.

4. Results and discussion

To demonstrate the validity of the proposed MSPSO-BP algorithm in the inverse radiative analysis, several different test cases are considered in this section. Firstly, the PSDs are retrieved using multi-wavelength method. Secondly, the optical constant is obtained for L-N distribution. The influence of measurement errors is considered. Finally, the PSDs and optical constant are retrieved simultaneously using a secondary optimization method. Each case was implemented using FORTRAN code, and the developed program was executed on an Intel Core i7 PC.

4.1. The retrieval of PSDs

Multi-wavelength method [43] has been applied to obtain the PSDs accurately and effectively. Therefore, to test the influence of wavelength number on the retrieval accuracy and efficiency, different amounts of wavelengths were used in the present work, which, to be specific, are one, two, and three wavelengths. For the single wavelength, the wavelength was set as 0.55 μm . For the two wavelengths, 0.55 and 0.63 μm were applied. And 0.55, 0.633, 0.649 μm were used for the three-wavelength method. The optical constants (n, k) corresponding to different wavelengths were set as (1.381, 0.00426), (1.377, 0.00162), and (1.376, 0.00504). The retrieval of

the PSDs was solved by minimizing the objective function, which was defined as the sum of the square residuals between the predicted and the expected reflectance and transmittance. Therefore, the objective function can be expressed as follows:

$$F_{\text{obj}} = \sum_{i=1}^n \left[\left(\frac{R_{\text{pre}}(i) - R_{\text{exp}}(i)}{R_{\text{exp}}(i)} \right)^2 + \left(\frac{T_{\text{pre}}(i) - T_{\text{exp}}(i)}{T_{\text{exp}}(i)} \right)^2 \right] \quad (16)$$

The number density of particle system was set to a fixed value $N_0 = 5.0 \text{ cm}^{-3}$. The range of parameters in monomodal distributions are set as $\bar{D} \in (0.1, 1.1)$ and $\sigma \in (1.4, 3.4)$. To obtain adequate training samples for training network, 1000 sets of data were generated randomly within the initial range. Nine hundred sets were randomly selected as training samples, and others were chosen as prediction samples to test the performance of trained network. Afterwards, the well-trained and well-tested network will be used to retrieve the PSD of the particle system. The control parameters of MSPSO-BP are shown in **Table 1**. Also, the relative errors were applied to demonstrate accuracy retrieval results, which can be expressed as follows:

$$\varepsilon = \left| \frac{\text{PRE} - \text{EXP}}{\text{EXP}} \right| \times 100\% \quad (17)$$

where PRE and EXP stand for the predicted and expected values of the results, respectively.

Maximum iteration number	200
Momentum factor	$L_r = 0.1$
Weight	$w_{\text{max}} = 0.9; w_{\text{min}} = 0.5$
Learning factors	$c_1 = 2.05; c_{2\text{max}} = 2.05; c_{2\text{min}} = 0.4;$ $c_2 = c_{2\text{max}} - \frac{t}{t_{\text{max}}} \times (c_{2\text{max}} - c_{2\text{min}})$
Maximum stagnation time	$S_{\text{max}} = 20$

Table 1. Parameter setting of MSPSO-BP.

In the present work, three commonly used PSDs were applied, i.e. R-R, S-N, and L-N (see Equations 13–15). The training results, which are the mean relative errors of all the prediction samples, for different PSDs and different number of incident wavelengths are shown in **Table 2**.

It can be seen that MSPSO-BP has excellent prediction performance for the three kinds of PSD functions. Besides, with the number of wavelengths increasing, the error between predicted values and expected values of prediction samples gradually decreases, which means that the multiple-wavelength method can improve the prediction accuracy effectively. Also, it is

noticeable that the calculation time is also increasing with the number of wavelengths used, which is caused by the increasing calculation times of the direct problem.

PSDs	Wavelength number	$\varepsilon_\sigma(\%)$	$\varepsilon_{\bar{D}}(\%)$
R-R	1	0.036	0.066
	2	0.009	0.021
	3	0.004	0.007
S-N	1	0.059	0.076
	2	0.016	0.021
	3	0.001	0.004
L-N	1	0.061	0.095
	2	0.012	0.034
	3	0.002	0.005

Table 2. The training results of different PSDs.

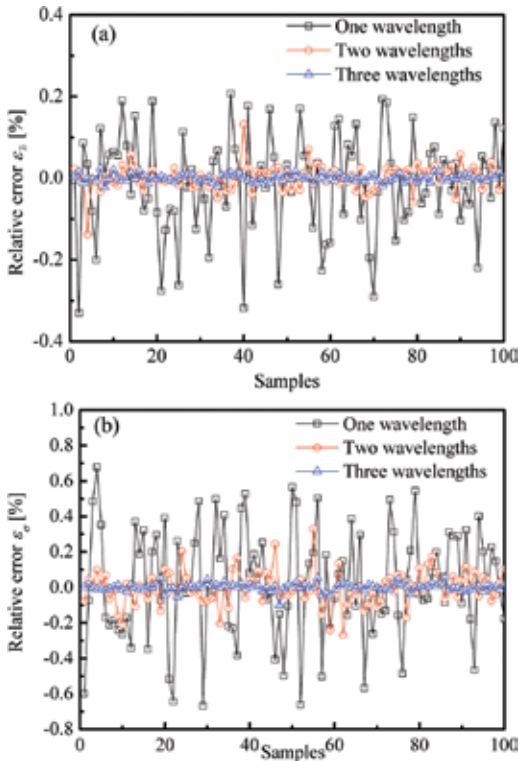


Figure 3. The prediction results of the L-N distribution (a) σ and (b) \bar{D} .

After the test of the network is finished, the network can be used to retrieve the PSD of the particle system. The results of the 100 sets of test data are illustrated in **Figure 3**. It demonstrates clearly that increasing the incident wavelength will improve the accuracy the prediction results. To show the performance of the MSPSO-BP algorithm, a case was presented to retrieve the PSD parameters of the L–N distribution. The true value of PSD parameters is set as $\sigma = 2.0$ and $\bar{D}=0.6$. Each case was run for 10 times. The inversion results are shown in **Table 3**. It is evident that the retrieval results are accurate when using two- and three-wavelength methods. However, the results of single wavelength are not acceptable. This can be contributed to the insufficient information supplied by the single wavelength method. Therefore, it is suggested that more than one incident wavelength should be used in the retrieval of PSDs.

Wavelength number		\bar{D}	σ	Time (s)
1	Estimated value	0.552	2.230	872
	Relative error (%)	8.00	11.5	
2	Estimated value	0.607	1.957	1129
	Relative error (%)	1.17	2.15	
3	Estimated value	0.602	1.996	1398
	Relative error (%)	0.33	0.20	

Table 3. Retrieval results of L–N distribution.

The number of training sample is a very important parameter for the BP neural network. It should be dealt with very carefully. Therefore, different numbers of training samples are applied to the inversion of PSDs using three-wavelength method. The numbers of training samples are set as 600, 700, 800, and 900, respectively. The retrieval results are listed in **Table 4**.

Training sample number		\bar{D}	σ	Time (s)
600	Estimated value	0.585	2.093	1019
	Relative error (%)	2.50	4.65	
700	Estimated value	0.611	2.052	1127
	Relative error (%)	1.83	2.60	
800	Estimated value	0.595	2.013	1286
	Relative error (%)	0.83	0.65	
900	Estimated value	0.602	1.996	1398
	Relative error (%)	0.33	0.20	

Table 4. The influence of training samples.

It can be seen that by increasing the amount of the training samples, more accurate retrieval results can be obtained, which can be seen more clearly in **Figure 4**. Network training time is obviously related to the number of training data. Training the network with more data tends to spend more time. Therefore, to save computation time, less training samples should be applied. However, it will reduce the accuracy of the retrieval results at the same time (see **Figure 4**). Therefore, when using the neural network-based algorithm for an inverse problem, the number of the training samples should be decided very carefully.

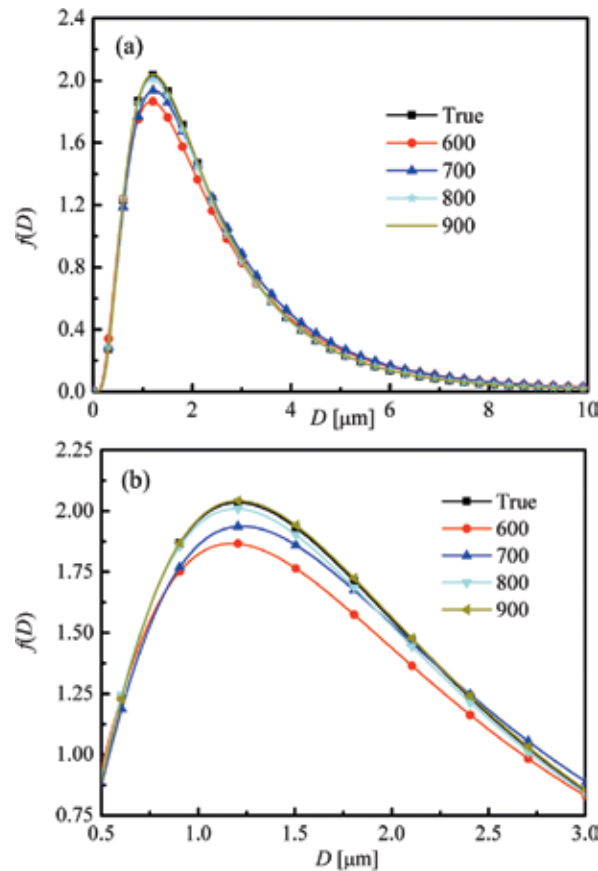


Figure 4. The influence of training samples.

4.2. The estimation of optical constant

In this section, MSPSO-BP algorithm is applied to retrieve the optical constant, i.e. n and k . It is known that the optical constant is different for different wavelengths. Multi-wavelength method is no longer effective in this situation as more wavelengths mean more unknown optical constants. Therefore, other methods should be applied to obtain more information about the particle system. In the present work, different thicknesses of particle system were

applied. To test the effectiveness of the multi-thickness method, the optical constant of a 1-D sample with thickness of 0.1, 0.2, and 0.3 m was retrieved.

First, the MSPSO-BP should be trained. The interval of the optical constant was set as $n \in (1.3, 1.6)$ and $k \in (0.001, 0.01)$. The PSD was set as L-N distribution and $\bar{D}=1.0$ and $\sigma = 2.0$. The incident wavelength was set as $0.55 \mu\text{m}$, which means $n = 1.381$ and $k = 0.00426$. Other parameters were set the same as last section. One thousand sets of n and k were generated within the interval randomly, 900 of which were used as training, and others were used to test performance of the trained network. The test results are shown in **Figure 5**.

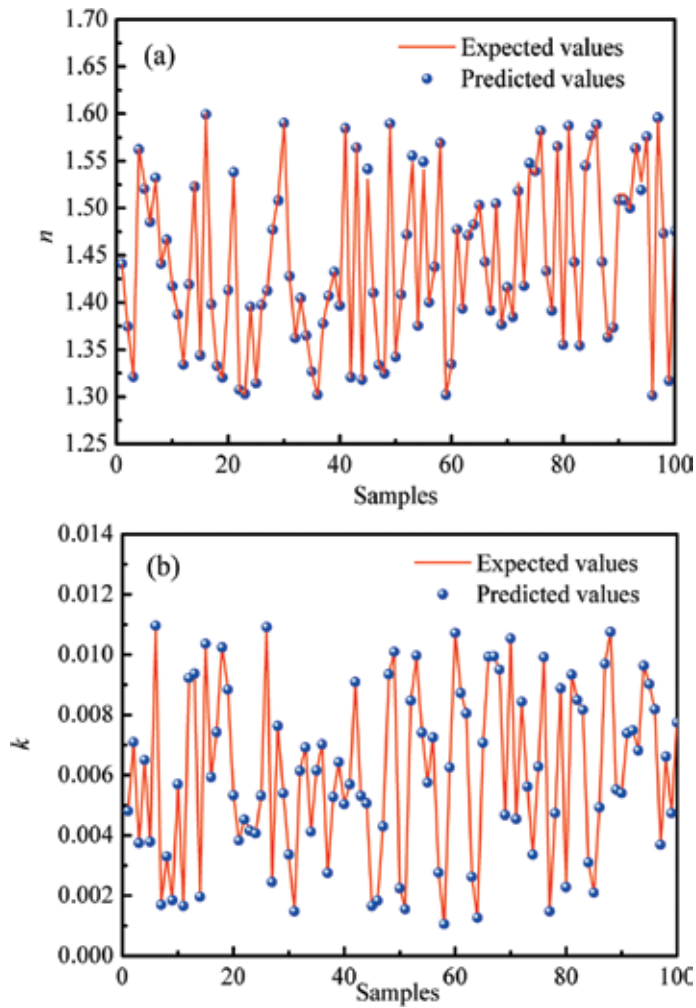


Figure 5. The comparison of the expected and predicted results of (a) n and (b) k of the 100 test samples.

It is evident that the predicted values and expected values of n and k are in a good agreement, which indicates that the trained network has an excellent ability for retrieving optical constant.

Therefore, the trained network will be used to conduct the inversion of optical constant. The optical constant that needs to be estimated was set as $(n, k) = (1.4, 0.006)$. Considering the fact that the inevitable measurement errors will occur in practical measurement, random errors were added to the original reflectance and transmittance. The results are shown in **Table 5**. It can be seen that even when the measurement noise is 5%, the retrieval results of the optical constants are still acceptable.

Measurement error		n	k
0%	Retrieval result	1.4002	6.0037×10^{-3}
	Relative error (%)	0.014	0.062
1%	Retrieval result	1.3995	5.9750×10^{-3}
	Relative error (%)	0.036	0.417
3%	Retrieval result	1.4012	5.9650×10^{-3}
	Relative error (%)	0.086	0.583
5%	Retrieval result	1.4038	5.9440×10^{-3}
	Relative error (%)	0.271	0.933

Table 5. The retrieval results of optical constant with and without measurement errors.

4.3. The retrieval of optical constant and PSDs

In this section, a secondary optimization method [44] was applied to obtain the optical constant and PSDs simultaneously by combining the multi-wavelength and multi-thickness methods. The diffuse transmittance, diffuse reflectance, and collimated transmittance were applied. The details of this method are described in Ref. [44] and will not be repeated here.

The wavelengths of two incident beams were set as 0.55 and 0.65 μm . And its corresponding optical constants were $1.432-0.00753i$ and $1.426-0.00794i$, respectively. Two samples, which were basically the same except for their thicknesses, are used. The original PSDs that need to be retrieved are shown in **Table 6**. Particle number density was set as 5.0 cm^{-3} . The searching range of the optical constant and PSDs was set as $n \in (1.3, 1.6)$ and $k \in (0.001, 0.01)$, $\bar{D} \in (0.1, 1.1)$, and $\sigma \in [1.4, 3.4]$. Ten thousand sets of optical constant and PSDs were generated randomly within the parameter range. After network was trained, it was applied to obtain the optical constant and PSDs. Each case was run for 10 times. The retrieval results are shown in **Table 7**.

PSDs	R-R	S-N	L-N
\bar{D}, σ	0.8, 2.8	0.7, 2.5	0.6, 2.0

Table 6. The PSDs parameters.

PSDs	n, k	$\epsilon_{\text{rel}} (\%)$	\bar{D}, σ	$\epsilon_{\text{rel}} (\%)$	Time (s)
R-R	1.451, 0.00765	1.33, 1.59	0.543, 2.435	9.50, 2.60	1258
	1.443, 0.00806	1.19, 1.51			
S-N	1.4199, 0.00732	0.84, 2.78	0.842, 2.466	5.25, 11.93	1468
	1.4137, 0.00774	0.85, 2.52			
L-N	1.4181, 0.0072	0.97, 4.28	0.709, 2.436	18.25, 21.78	1742
	1.4130, 0.0075	1.33, 0.41			

Table 7. The retrieval results of optical constant and PSDs.

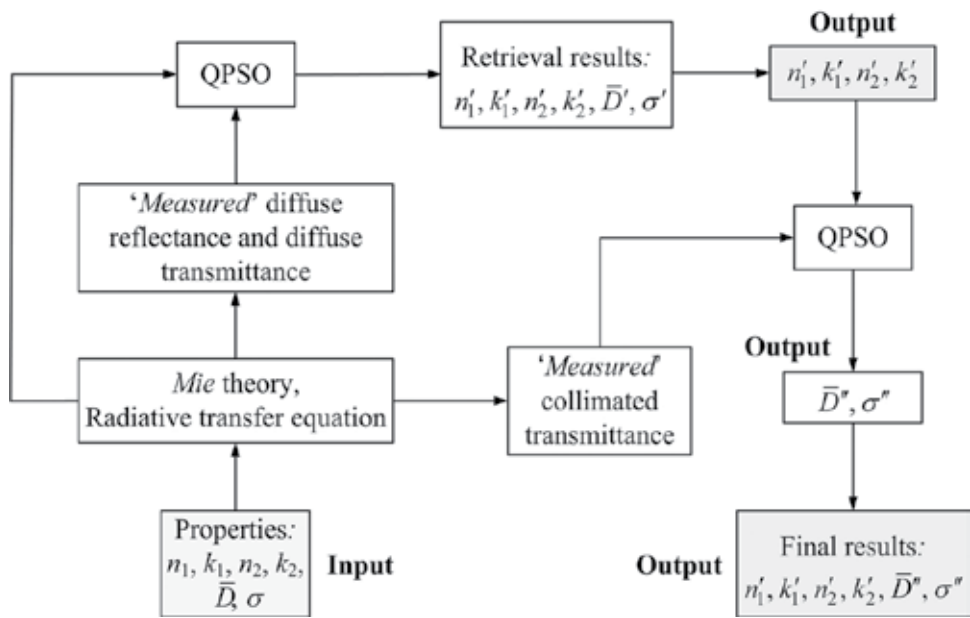


Figure 6. The flowchart for the whole optimization procedure.

It can be seen that for the same distribution, the retrieval results of optical constant are much smaller than those of the PSDs. The optical constant can be retrieved with a tolerable error. However, the average retrieved results of the PSDs, i.e. \bar{D} and σ , are not tolerable, and the maximum retrieval error can reach more than 20%. The reason the relative errors of the PSDs are so much larger than those of n and k is that the sensitivity coefficients of \bar{D} and σ are much lower than those of n and k [44]. In other words, the radiative signals are insensi-

tive to the PSDs, which mean even when the relative errors of the retrieved values of \bar{D} and σ are very large, the deviation in the calculated reflectance and transmittance signals can be ignored.

It can be concluded that the influences of the PSDs on the diffused reflectance and transmittance signals are less significant than that of the optical constant. Even though the retrieved results of the PSDs were not accurate, those of the complex refractive index were relatively satisfactory. Hence, the accuracy of the retrieved PSDs needed to be improved which meant much more information, which was sensitive to the PSDs, should be included in the inverse procedure. It was found that the collimated transmittance was more sensitive to \bar{D} and σ [44]. Therefore, the collimated transmittance signal was applied to improve the inversion accuracy of the PSDs. Based on the above analysis, a secondary optimization was performed in which \bar{D} and σ were the unknown parameters that needed to be retrieved, and the optical constant retrieved by the first optimization was applied as the original values. The inverse procedure is shown in **Figure 6**.

The results of the secondary optimization are shown in **Table 8**. It is evident that the estimated results of the secondary optimization were much more accurate than those of the first optimization, which proved that the applied inversion method is suitable for the simultaneous estimation of the complex refractive indexes and PSDs.

Function	Original values	Estimated values (\bar{D}, σ)	ε_{rel} (%)
R-R	0.8, 2.8	0.807, 2.809	0.88, 0.32
S-N	0.7, 2.5	0.711, 2.508	1.57, 0.32
L-N	0.6, 2.0	0.605, 2.0447	0.75, 1.98

Table 8. The second retrieval results of PSDs.

5. Conclusion

A multi-strategy PSO was applied to improve the performance of the BP multi-layer feed-forward ANN algorithm. The proposed MSPSO-based ANN method was introduced to solve the inverse radiation problems for the first time. Three commonly used PSD functions in a 1-D particle system were retrieved using the proposed algorithm. In addition, the optical constant was estimated, and the influence of measurement errors upon the precision of the estimated results was also investigated. Finally, the proposed algorithm was applied to the simultaneous estimation of the PSDs and optical constant using the multi-wavelength and multi-thickness method. All the retrieval results show that the proposed MSPSO-BP ANN can be applied to estimate the PSDs and optical constant in 1-D absorbing, scattering, and non-emitting system accurately even with noise data. Meanwhile, the secondary optimization can be recognized as an efficient way to obtain the PSDs and optical constant simultaneously by applying the multi-wavelength and multi-thickness method.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Nos. 51476043 and 51576053), the Major National Scientific Instruments and Equipment Development Special Foundation of China (No. 51327803), and the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (No. 51421063) and are gratefully acknowledged. A very special acknowledgement is also made to the editors and referees who make important comments to improve this chapter.

Author details

Hong Qi*, Ya-Tao Ren, Jun-You Zhang, Li-Ming Ruan and He-Ping Tan

*Address all correspondence to: qihong@hit.edu.cn

Harbin Institute of Technology, Harbin, China

References

- [1] Lu XD, Hsu PF. Reverse Monte Carlo method for transient radiative transfer in participating media. *Journal of Heat Transfer* 2004;126(4):621–627. DOI: 10.1115/1.1773587
- [2] Kumar S, Mitra K. Microscale aspects of thermal radiation transport and laser applications. *Advances in Heat Transfer* 1999;33:187–294. DOI: 10.1016/S0065-2717(08)70305-8
- [3] Hurt RH, Gibbins JR. Residual carbon from pulverized coal fired boilers: 1. Size distribution and combustion reactivity. *Fuel* 1995;74(4):471–480. DOI: 10.1016/0016-2361(95)98348-I
- [4] Mei L, Lu XF, Wang QH, Pan Z, Hong Z, Ji XY. The experimental study of fly ash recirculation combustion characteristics on a circulating fluidized bed combustor. *Fuel Processing Technology* 2014;118:192–199. DOI: 10.1016/j.fuproc.2013.09.002
- [5] Sun YP, Lou C, Zhou HC. Estimating soot volume fraction and temperature in flames using stochastic particle swarm optimization algorithm. *International Journal of Heat and Mass Transfer* 2011;54(1–3):217–224. DOI: 10.1016/j.ijheatmasstransfer.2010.09.049
- [6] Seames WS. An initial study of the fine fragmentation fly ash particle mode generated during pulverized coal combustion. *Fuel Processing Technology* 2003;81(2):109–125. DOI: 10.1016/S0378-3820(03)00006-7

- [7] Li H, Liu G, Cao Y. Content and distribution of trace elements and polycyclic aromatic hydrocarbons in fly ash from a coal-fired CHP plant. *Aerosol Air Quality Research* 2014;14:1179–1188. DOI: 10.4209/aaqr.2013.06.0216
- [8] Viskanta R, Mengüç MP. Radiation heat transfer in combustion systems. *Progress in Energy and Combustion Science* 1987;13(2):97–160. DOI: 10.1016/0360-1285(87)90008-6
- [9] Marakis JG, Papapavlou CH, Kakaras E. A parametric study of radiative heat transfer in pulverised coal furnaces. *International Journal of Heat and Mass Transfer* 2000;43(16):2961–2971. DOI: 10.1016/S0017-9310(99)00347-6
- [10] Wyatt PJ. Some chemical, physical, and optical properties of fly ash particles. *Applied Optics* 1980;19(6):975–983. DOI: 10.1364/AO.19.000975
- [11] Lee SH, Kima HJ, Sakaib E, Daimonb M. Effect of particle size distribution of fly ash–cement system on the fluidity of cement pastes. *Cement and Concrete Research* 2003;33(5):763–768. DOI: 10.1016/S0008-8846(02)01054-2
- [12] He ZZ, Qi H, Yao YC, Ruan LM. An effective inversion algorithm for retrieving bimodal aerosol particle size distribution from spectral extinction data. *Journal of Quantitative Spectroscopy and Radiative Transfer* 2014;149:117–127. DOI: 10.1016/j.jqsrt.2014.08.002
- [13] Willis C. The complex refractive index of particles in a flame. *Journal of Physics D: Applied Physics* 1970;3:1944–1956. DOI: 10.1088/0022-3727/3/12/324
- [14] Volz FE. Infrared refractive index of atmospheric aerosol substances. *Applied Optics* 1972;11(4):755–759. DOI: 10.1364/AO.11.000755
- [15] Gupta RP, Wall TF. The complex refractive index of particles. *Journal of Physics D: Applied Physics* 1981;14(6):L95–L98. DOI: 10.1088/0022-3727/14/6/003
- [16] Shu Y, Zhou XJ, Zhao YZ. A theoretical study of multi-wavelength lidar exploration of optical properties of atmospheric aerosols. *Advances in Atmospheric Sciences* 1986;1:23–38. DOI: 10.1007/BF02680043
- [17] Erlick C. Effective refractive indices of water and sulfate drops containing absorbing inclusions. *Journal of the Atmospheric Sciences* 2006;63:754–763. DOI: <http://dx.doi.org/10.1175/JAS3635.1>
- [18] Shen Y, Draine BT, Johnson ET. Modeling porous dust grains with ballistic aggregates. I. Geometry and optical properties. *The Astrophysical Journal* 2008;689:260–275. DOI: 10.1086/592765
- [19] Pecharroman C, Gaspera ED, Martucci A, Galindod RE, Mulvaney P. Determination of the optical constants of gold nanoparticles from thin-film spectra. *The Journal of Physical Chemistry* 2015;119(17):9450–9459. DOI: 10.1021/jp512611m
- [20] Li WK, Wu YX, Huang ZM, Fang R, Lv JF. Measurement results comparison between laser particle analyzer and sieving method in particle size distribution. *Powder Science and Technology* 2007;5:003.

- [21] Tang H, Lin JZ. Retrieval of spheroid particle size distribution from spectral extinction data in the independent mode using PCA approach. *Journal of Quantitative Spectroscopy and Radiative Transfer* 2013;115:78–92. DOI: 10.1016/j.jqsrt.2012.09.005
- [22] Rizzi R, Guzzi R, Legnani R. Aerosol size spectra from spectral extinction data: The use of a linear inversion method. *Applied Optics* 1982;21(9):1578–1587. DOI: 10.1364/AO.21.001578
- [23] Muhieddine M, Canot E, March R. Heat transfer modeling in saturated porous media and identification of the thermophysical properties of the soil by inverse problem. *Applied Numerical Mathematics* 2012;62(9):1026–1040. DOI: 10.1016/j.apnum.2012.02.008
- [24] Lu S, Heng Y, Mhamdi A. A robust and fast algorithm for three-dimensional transient inverse heat conduction problems. *Journal of Heat and Mass Transfer* 2012;55(25–26): 7865–7872. DOI: 10.1016/j.ijheatmasstransfer.2012.08.018
- [25] Gosselin L, Tye-Gingras M, Mathieu-Potvin F. Review of utilization of genetic algorithms in heat transfer problems. *International Journal of Heat and Mass Transfer* 2009;52(9):2169–2188. DOI: 10.1016/j.ijheatmasstransfer.2008.11.015
- [26] Stephany S, Becceneri JC, Souto RP, Velho HFD, Neto AJS. A pre-regularization scheme for the reconstruction of a spatial dependent scattering albedo using a hybrid ant colony optimization implementation. *Applied Mathematical Modelling* 2010;34(3):561–572. DOI: 10.1016/j.apm.2009.06.006
- [27] Zhang B, Qi H, Ren YT, Sun SC, Ruan LM. Inverse transient radiation analysis in one-dimensional participating slab using improved ant colony optimization algorithms. Inverse transient radiation analysis in one-dimensional participating slab using improved ant colony optimization algorithms. *Journal of Quantitative Spectroscopy and Radiative Transfer* 2014;133:351–363. DOI: 10.1016/j.jqsrt.2013.08.020
- [28] Ermis K, Erek A, Dincer I. Heat transfer analysis of phase change process in a finned-tube thermal energy storage system using artificial neural network. *International Journal of Heat and Mass Transfer* 2007;50(15):3163–3175. DOI: 10.1016/j.ijheatmasstransfer.2006.12.017
- [29] Jambunathan K, Hartle SL, Ashforth-Frost S, Fontama VN. Evaluating convective heat transfer coefficients using neural networks. *International Journal of Heat and Mass Transfer* 1996;39(11):2329–2332. DOI: 10.1016/0017-9310(95)00332-0
- [30] Gevrey M, Dimopoulos I, Lek S. Review and comparison of methods to study the contribution of variables in artificial neural network models. *Ecological Modelling* 2003;160(3):249–264. DOI: 10.1016/S0304-3800(02)00257-0
- [31] Durbin R, Rumelhart DE. Product units with trainable exponents and multi-layer networks. In: Soulié FF, Hérault J, editors. *Neurocomputing*. Berlin Heidelberg: Springer; 1990. p. 15–26. DOI: 10.1007/978-3-642-76153-9_2

- [32] Sadeghi BHM. A BP-neural network predictor model for plastic injection molding process. *Journal of Materials Processing Technology* 2000;103(3):411–416. DOI: 10.1016/S0924-0136(00)00498-2
- [33] Yu F, Xu XZ. A short-term load forecasting model of natural gas based on optimized genetic algorithm and improved BP neural network. *Applied Energy* 2014;134:102–113. DOI: 10.1016/j.apenergy.2014.07.104
- [34] Zhou JL, Duan ZC, Li Y. PSO-based neural network optimization and its utilization in a boring machine. *Journal of Materials Processing Technology* 2006;178:19–23. DOI: 10.1016/j.jmatprotec.2005.07.002
- [35] Ren YT, Qi H, Chen Q, Ruan LM. Inverse transient radiative analysis in two-dimensional turbid media by particle swarm optimizations. *Mathematical Problems in Engineering* 2015;2015:680823. DOI: 10.1155/2015/680823
- [36] Shi Y, Eberhart RC. A modified particle swarm optimizer. In: *IEEE International Conference on Evolutionary Computation*; 4–9 May 1998; Anchorage, AK. Anchorage, AK: IEEE; 1998. p. 69–73. DOI: 10.1109/ICEC.1998.699146
- [37] Wang QY, Ma HZ, Cao SR. A multi-strategy particle swarm optimization algorithm and its application on hybrid magnetic levitation. *Proceedings of the CSEE* 2014;34(30):5416–5424.
- [38] Rahnamayan S, Tizhoosh HR, Salama MMA. Opposition versus randomness in soft computing techniques. *Applied Soft Computing* 2008;8(2):906–918. DOI: 10.1016/j.asoc.2007.07.010
- [39] Huang GB, Chen L, Siew CK. Universal approximation using incremental constructive feed forward networks with random hidden nodes. *IEEE Transactions on Neural Networks* 2006;17(4):879–892. DOI: 10.1109/TNN.2006.875977
- [40] Modest MF. Backward Monte Carlo simulations in radiative heat transfer. *Journal of Heat Transfer* 2003;125(1):57–62. DOI: 10.1115/1.1518491
- [41] Bohren CF, Huffillan DR. *Absorption and Scattering of Light by Small Particles*. New York: John Wiley & Sons; 2008.
- [42] He ZZ, Qi H, Yao YC, Ruan LM. Inverse estimation of the particle size distribution using the fruit fly optimization algorithm. *Applied Thermal Engineering* 2015;88:306–314. DOI: 10.1016/j.applthermaleng.2014.08.057
- [43] He ZZ, Qi H, Wang YQ, Ruan LM. Inverse estimation of the spheroidal particle size distribution using ant colony optimization algorithms in multispectral extinction technique. *Optics Communications* 2014;328:8–22. DOI: 10.1016/j.optcom.2014.04.042
- [44] Ren YT, Qi H, Chen Q, Ruan LM, Tan HP. Simultaneous retrieval of the complex refractive index and particle size distribution. *Optics Express* 2015;23:19328–19337. DOI: 10.1364/OE.23.019328

A Novel Artificial Organic Controller with Hermite Optical Flow Feedback for Mobile Robot Navigation

Hiram Ponce, Ernesto Moya-Albor and Jorge Brieva

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/62466>

Abstract

This chapter describes a novel nature-inspired and intelligent control system for mobile robot navigation using a fuzzy-molecular inference (FMI) system as the control strategy and a single vision-based sensor device, that is, image acquisition system, as feedback. In particular, FMI system is proposed as a hybrid fuzzy inference system with an artificial hydrocarbon network structure as defuzzifier that deals with uncertainty in motion feedback, improving robot navigation in dynamic environments. Additionally, the robotics system uses processed information from an image acquisition device using a real-time Hermite optical flow approach. This organic and nature-inspired control strategy was compared with a conventional controller and validated in an educational robot platform, providing excellent results when navigating in dynamic environments with a single-constrained perception device.

Keywords: Artificial organic networks, artificial hydrocarbon networks, fuzzy-molecular inference system, robotics, optical flow

1. Introduction

Navigation of mobile robots is still a challenging problem due to the uncertain and dynamics of environments [1]. Typically, navigation consists of different tasks [1] such as: perception, exploration, mapping, localization, path planning, and path execution. In this work, navigation is limited to a strategy or methodology that guides a mobile robot to accomplish a task through an environment with obstacles—static or dynamic—without collisions, using sensors. Also, this work considers a robotic system with a unique vision-based sensor for perception.

Artificial intelligence (AI) has been widely used for navigation of mobile robots. Different AI methods have been proposed, such as: fuzzy logic [2], artificial neural networks [3], hybrid neuro-fuzzy controllers [4–6], evolutionary and meta-heuristics algorithms [7], and nature-inspired computing (e.g., particle swarm optimization and ant colony optimization) [8, 9]. Artificial intelligence has been used in different levels of cognition such as: reasoning, decision-making, and learning.

Furthermore, some applications using AI-based controllers and features based on optical flow to compute the control law are present in the literature. For instance, Williams et al. [10] propose a support vector machines (SVM)—kernel to carry out vehicle tracking using optical flow features. Meta-cognitive neuro-fuzzy inference systems (McFIS) for accurate detection of human actions from video sequences have been proposed by Subramanian et al. [11]. In fact, they employ optical flow-based features, and the functional relationship between these optical flow-based features and action classes is approximated using the McFIS classifier. Boroujeni et al. [12] propose a method for obstacle detection using optical flow and *k*-means clustering-based classifier to determine obstacles' positions. Narayanan et al. [13] aim to develop a real-time system capable of rapidly detecting and tracking vehicles in a video stream using a monocular vision system. The detection is based on the Viola–Jones classifier and optical flow features [13]. In addition, Crnko et al. [14] propose a biomimetic measurement of optical flow and centroid for visual-servo control of hover flight.

In that way, this work proposes a novel nature-inspired and intelligent controller for navigation of mobile robots constrained to a single vision-based sensor, using a recent hybrid fuzzy inference system and artificial hydrocarbon networks (AHN) approach so-called fuzzy-molecular inference (FMI) system. This fuzzy-molecular inference system inherits characteristics from fuzzy inference systems (FIS) such as storage of knowledge from experts, dealing with uncertainty, and linguistic interpretation [15]; but also from artificial organic networks (AON), a learning method inspired on chemical organic compounds that allow packaging information in molecular units, organizing of information, and dealing with aggressive uncertainty and noise [16]. In advance, fuzzy-molecular inference systems can deal with uncertainty and under aggressive conditions of noise, with relative easiness of implementation [15]. Examples of FMI-based controllers can be found in literature [15, 17]. In addition, this proposed controller uses a real-time Hermite optical flow (RT-HOF) approach for detecting direction of moving objects in the visual range of the mobile robot. In fact, RT-HOF is a method that determines the displacement of each pixel at some time in an image sequence representing the motion of objects in the scene or the relative motion of the sensor, with the usage of the Hermite transform (HT) [18] as a biological image model in real-time. This RT-HOF approach has been proved to be easy to implement and faster than conventional optical flow methods [19], giving advantages to be used in online navigation.

The rest of the chapter is organized as follows: Section 2 briefly describes the fundamentals of the proposal: artificial hydrocarbon networks algorithm as part of artificial organic networks method, the fuzzy-molecular inference system, and the real-time Hermite optical flow method. Section 3 formalizes the proposed nature-inspired and intelligent control for mobile robots and its design. Later on, Section 4 describes the implementation of the proposed artificial

organic controller in a mobile robot system. Section 5 presents experimental results of the proposed controller, and Section 6 concludes the work.

2. Nature-inspired and intelligent computing

This section describes the fundamentals of the proposed controller for robot navigation. First, artificial hydrocarbon networks learning algorithm is introduced. Then, the fuzzy-molecular inference system is described, and lastly, the real-time Hermite optical flow approach is presented.

2.1. Artificial hydrocarbon networks

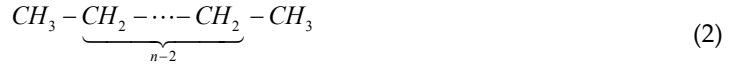
Artificial hydrocarbon networks (AHN) is a nature-inspired learning algorithm based on the artificial organic networks framework. In fact, artificial organic networks (AON) technique is a meta-heuristic method and framework inspired on chemical organic structures and the energy involved in the interaction among them. Inspiration of AON method considers packaging information in independent structures named molecules which they can relate among them using chemical energy-based heuristics [16]. Therefore, AON method provides modularity, inheritance, organization, and structural stability in the output response [16].

Thus, artificial hydrocarbon network method is a supervised learning technique that inherits from AON framework, restricted in the shape and type of molecules occupied [16]. From the latter, AHN is inspired on chemical hydrocarbon compounds that constraint the method only using two atomic units, that is, hydrogen and carbon atoms, that can be related together with at most one and four degrees of freedom, respectively, as described below.

The basic unit with information of AHN is the so-called *CH*-molecule structure. It is made of two or more atoms related among them in order to define a functional behavior φ centred in the carbon atom and parameterized with the other hydrogen atoms linked to it, as shown in Eq. (1), where σ is a real value called the carbon value, $H_i \in \mathbb{C}$ is the i th hydrogen atom linked to the carbon atom, $k \leq 4$ represents the number of hydrogen atoms in the molecule, and x is the input signal representing a stimulus of the overall molecule.

$$\varphi(x) = \sigma \prod_{i=1}^k (x - H_i) \quad (1)$$

Then, these molecules can form other types of structures so-called compounds, which include nonlinear relationships among all molecules [16]. In this work, forming compounds are restricted to be linear and saturated compounds like Eq. (2), where CH_k represents a molecule with $k < 4$ hydrogen atoms associated to it, n is the number of *CH*-molecules in the compound, and $A - B$ represents a simple bond between two molecules A and B . At the end, this linear and saturated compound is made of two CH_3 molecules and $(n - 2)$ CH_2 molecules [20].



In terms of the behavior of the compound, it is expressed as a nonlinear function ψ that depends on the molecular behaviors and the type of bond between molecules. In this work, ψ is defined as a piecewise function of nCH -molecules like Eq. (3) [17], where L_i represents the i th bound at which a molecule can act over the input space, and φ_i is the i th molecular behavior in the forming compound. To this end, if the input domain is in the interval $x \in [x_{min}, x_{max}]$, then $L_0 = x_{min}$ and $L_n = x_{max}$.

$$\psi(x) = \begin{cases} \varphi_1(x) & L_0 \leq x < L_1 \\ \dots & \dots \\ \varphi_n(x) & L_{n-1} \leq x \leq L_n \end{cases} \quad (3)$$

Lastly, an artificial hydrocarbon network is a mixture AHN of a set of compounds interacting among them in definite ratios named stoichiometric coefficients α_m as shown in Eq. (4), where c is the number of compounds in the mixture, and ψ_m is the m th behavior of compound in the mixture [16]. In this work, the artificial hydrocarbon network structure is restricted to be one compound such that $AHN(x) = \psi_1(x)$.

$$AHN(x) = \sum_{m=1}^c \alpha_m \psi_m(x) \quad (4)$$

- 1: Set the maximum number of compounds c in the mixture AHN .
- 2: Set the maximum number of molecules n per compound.
- 3: Set the current number of compounds $c' = 1$.
- 4: While c' is less or equal to c and c' is not optimum:
 - 5: Initialize a compound with current number of molecules $n' = 2$.
 - 6: While n' is less or equal to n and n' is not optimum:
 - 7: Build the structure of compound (select the shape of molecules).
 - 8: Compute hydrogen and carbon values, as well as bounds.
 - 9: If n' is not optimum, then increase n' in a unit.
 - 10: End-while
 - 11: If c' is not optimum, then increase c' in a unit.
- 12: End-while
- 13: Compute the stoichiometric coefficients for all compounds.
- 14: Return AHN as a mixture of compounds.

Figure 1. AHN Algorithm to Create an Artificial Hydrocarbon Network.

Chemical heuristic rules for searching and building an optimum artificial hydrocarbon network with specific carbon and hydrogen values are computed using the so-called AHN-algorithm depicted in **Figure 1** (for a detailed description, see for example [21]). Finally, examples of applications using this technique can be found in literature [20, 22–24].

2.2. Fuzzy-molecular inference system

Fuzzy-molecular inference (FMI) systems are ensembles of artificial hydrocarbon networks and fuzzy inference systems [15] that consist of three steps: fuzzification, fuzzy inference engine, and molecular defuzzification. The overall process maps crisp inputs to a fuzzy space, and resulting values are returned to as crisp outputs using a molecular approach based on artificial hydrocarbon networks. In addition, a knowledge base stores information about a specific problem [15].

The basis fuzzy-molecular rule is shown in Eq. (5), where x_i is a linguistic variable of the antecedents in a set of fuzzy partitions, F_i , y_f is the variable of the consequent in the f th fuzzy-molecular rule, M_j is the j -molecular unit of a linear and saturated hydrocarbon compound stimulated with the resulting fuzzy implication value of the T -norm Δ .

$$R_f : \text{if } \Delta(x_1 \in F_1, \dots, x_i \in F_i, \dots, x_k \in F_k), \text{ then } y_f \in M_j \quad (5)$$

First at the fuzzification step, the fuzzy-molecular inference system receives an input signal x also known as the linguistic variable, and it is mapped to a fuzzy value in the range of $[0,1]$. Then, let F be a fuzzy set and $\mu_F(x)$ the membership function of F , $\forall x \in X$ where X is the input domain space. Moreover, $\mu_F(x) \in [0, 1]$ is a real value representing the degree of belonging to x into the fuzzy set F . Additionally, this linguistic variable can be partitioned into k different fuzzy sets F_i . Then, its membership value can be computed over the set of membership functions $\mu_{F_i}(x)$ for all $i = 1, \dots, k$. To this end, the shape of the membership functions depends on the purpose of the problem domain [15].

The second step considers the fuzzy inference engine as a mechanism that computes the consequent value y_f of a set of fuzzy rules that accept membership values of linguistic variables, as already shown in Eq. (5). If assuming that $\mu_\Delta(x_1, \dots, x_k)$ is the result of any T -norm function, then Eq. (5) can be rewritten as Eq. (6), using μ_Δ as the min function [15].

$$R_f : \text{if } \Delta(x_1 \in F_1, \dots, x_i \in F_i, \dots, x_k \in F_k), \text{ then} \\ y_f = \varphi_j \left(\min \{ \mu_{F_1}(x_1), \dots, \mu_{F_k}(x_k) \} \right) \quad (6)$$

Lastly, the molecular defuzzification step computes the crisp output value y as Eq. (7) using all fuzzy rules of the form as Eq. (6), where μ_{Δ_f} is the f th fuzzy evaluation of the antecedents, and y_f is the f th consequent value for all fuzzy rules [15].

$$y = \frac{\sum \mu_{\Delta f}(x_1, \dots, x_k) \cdot y_f}{\sum \mu_{\Delta f}(x_1, \dots, x_k)} \quad (7)$$

The proper knowledge of the specific problem domain can be enclosed into a knowledge base, summarizing all fuzzy rules of Eq. (6), assigning any *CH*-molecule of the compound to each antecedent [15]. To this end, examples of applications using the FMI system can be found in literature [15–17, 22, 25].

2.3. Real-time Hermite optical flow

Optical flow (OF) is defined as a two dimensional distribution of “apparent velocities” of the objects in the scene that in most cases are associated with intensity variations of the objects into an image sequence [26]. It is represented by a vector field that represents the displacement of pixels. For example, **Figure 2** shows the vector field obtained from frames 43 and 44 of the camera motion sequence [27].



Figure 2. Optical Flow Representation in a Sequence of Images [19]. (a) Frame 43 of the *Camera Motion Sequence*. (b) Frame 44 of the *Camera Motion Sequence*. (c) Optical Flow Result.

Classical differential optical flow methods assume that the intensity value of the objects remain constant in two consecutive time instants of an image sequence $L(X, t)$. This assumption was proposed by Horn and Schunck in 1981 [28] and is known as the *constant intensity constraint* like Eq. (8) where $X = (x, y, 1)^T$ is the pixel location, $W = (u, v, 1)^T$ represents the horizontal and vertical pixel displacements, respectively, between two images at time t and $(t + 1)$.

$$L(X + W, t + 1) - L(X, t) = 0 \quad (8)$$

If small displacements are assumed, a Taylor expansion is considered; and the *optical flow constraint* Eq. (9) is obtained where $\nabla_3 L = (L_x, L_y, L_t)^T$ are the derivatives of the intensity image $L(X)$ in x, y directions and time.

$$W^T (\nabla_3 L) = 0 \quad (9)$$

If only the constant intensity constraint is used, it is not possible to determine the two components of displacement u and v ; then, an ill-posed problem is obtained known as the aperture problem where only the normal component of the motion can be obtained [28]. Therefore, it needs additional constraints to fully calculate the optical flow. Recent differential optical flow methods have proposed additional constraints to overcome the aperture problem and to improve accuracy of the displacement obtained, that is, add local image constraints.

Horn and Schunck [28] proposed the *smoothness constraint* which assumes that the flow is smooth, that is, pixels within a neighborhood have similar magnitude and orientation. Thereby, the optical flow estimation method of Horn can be found minimizing, by an iterative approach, the energy functional of Eq. (10), where $\nabla_3 L = (L_x, L_y, L_t)^T$, Ω is the image domain, and α is a smoothness parameter. In most cases, the constant intensity constraint assumption of Eq. (8) is not fulfilled. Therefore, to overcome this problem, an additional term independent to intensity change is required. In [29, 30], a *constant gradient constraint* is proposed as shown in Eq. (11). A way to represent the most important visual characteristics is using bio-inspired models as the Hermite transform [31].

$$\int_{\Omega} \left(W^T (\nabla_3 L \nabla_3 L^T) W + \alpha |\nabla W|^2 \right) dX = 0 \quad (10)$$

$$\nabla L(X + W) - \nabla L(X) = 0 \quad (11)$$

The Hermite transform is an image model based in a polynomial decomposition from a perceptual standpoint. This image model emulates the local processing and the response of receptive fields, for example, the Gaussian derivative model of the human vision system [31, 32]. In fact, it is computed by convolution of the image $L(x, y)$ with the analysis filters $D_{m,n-m}$ like in Eq. (12), where $L_{m,n-m}(x, y)$ are the cartesian Hermite coefficients, m and $(n - m)$ are the analysis order in the orthogonal directions x and y , and (x_0, y_0) represents the position of the sampling lattice S .

$$\begin{aligned} L_{m,n-m}(x_0, y_0) = & \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} L(x, y) D_{m,n-m}(x_0 - x, y_0 - y) dx dy \\ & n = 0, 1, \dots, \infty \\ & m = 0, 1, \dots, n \end{aligned} \quad (12)$$

The filter functions $D_{m,n-m}(x, y)$ are defined by the analysis window $v^2(x, y)$ and the polynomials orthogonal to this window $G_{m,n-m}(x, y)$ as expressed in Eq. (13). At last, the separable Hermite filters $D_{m,n-m}(x, y) = D_m(x)D_{n-m}(y)$ are defined with a Gaussian $v(x, y) = \frac{1}{\sigma\sqrt{\pi}} \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right)$ with

normalization factor for a unitary energy $\text{inv}^2(x, y)$, as in Eq. (14). **Figure 3** shows an example of the Hermite filters $D_n(x)$ for $N = 5$ ($n = 0, 1, \dots, N$).

$$D_{m,n-m}(x,y) = G_{m,n-m}(-x,-y)v^2(-x,-y) \quad (13)$$

$$D_n(x) = \frac{(-1)^n}{\sqrt{2^n n!}} \frac{1}{\sigma \sqrt{\pi}} H_n\left(\frac{x}{\sigma}\right) \exp\left(-\frac{x^2}{\sigma^2}\right) \quad (14)$$

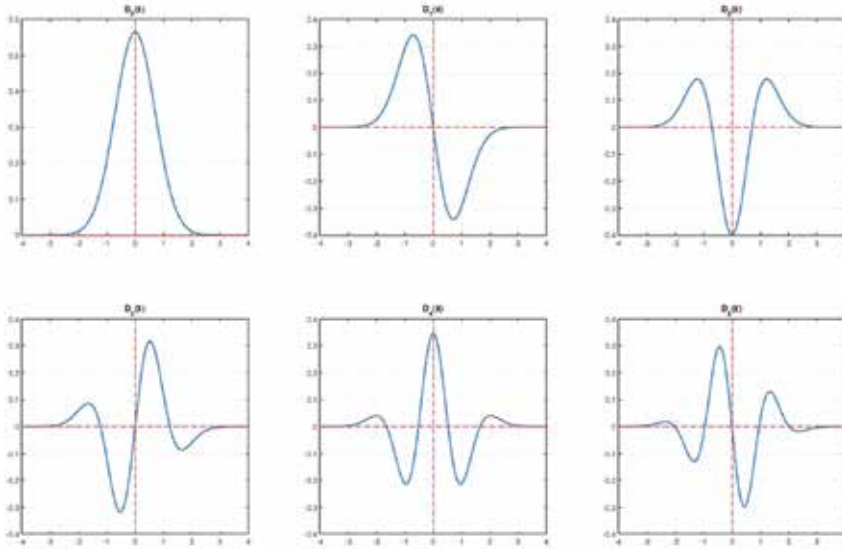


Figure 3. Hermite Filters $D_n(x)$ for $N = 5$ ($n = 0, 1, \dots, N$).

The steered Hermite transform (SHT) can be defined by rotating the cartesian Hermite coefficients $L_{m,n-m}(x, y)$ by the angular functions $g_{m,n-m}(\theta)$ such that Eq. (15) holds, where $l_{m,n-m,\theta}(x, y)$ are steered Hermite coefficients for the local angle θ , and $g_{m,n-m}(\theta)$ expresses the directional selectivity of the filter as shown in Eq. (16) [33].

$$l_{m,n-m,\theta}(x,y) = \begin{cases} \sum_{k=0}^n (L_{k,n-k}(x,y))(g_{k,n-k}(\theta)), & m = 0 \\ 0, & m > 0 \end{cases} \quad (15)$$

$$g_{m,n-m}(\theta) = \sqrt{\binom{n}{m}} (\cos^m \theta) (\sin^{n-m} \theta) \quad (16)$$

The steered Hermite coefficients $l_{m,n-m,\theta}(x, y)$ allow to adapt the analysis process to the local content of the image, for example, using directional Gaussian derivatives filters [34], and the

angle θ can be estimated by a criterion of maximum energy direction (e.g., the gradient of the image). **Figure 4** shows the steered Hermite coefficients of the *House* test image [35] for $N = 3$ ($n = 0, 1, 2, 3$).



Figure 4. An Ensemble of the Steered Hermite Coefficients of the *House* Test Image for $N = 3$ ($n = 0, 1, \dots, N$). [36].

The steered Hermite transform can be used to increase the accuracy of the optical flow by using a biological image model and by incorporating the steered Hermite coefficients as constraint independent to intensity changes. In [19], a modified version of Horn and Schunck approach was proposed. This includes an expansion of the *constant intensity constraint* and incorporates the *steered Hermite coefficient constraint* shown in Eq. (17), where γ is a weight parameter.

$$\left[L_0(X+W) - L_0(X) \right] + \gamma \left[\sum_{n=1}^N l_{n,\theta}(X+W) - \sum_{n=1}^N l_{n,\theta}(X) \right] = 0, \quad (17)$$

Applying a Taylor expansion of Eq. (17) and reducing terms using the relation $L_k = \langle L(x), H_k\left(\frac{x}{\sigma}\right) \rangle$ [18], it can be obtained the *optical flow Hermite constraint* equation as denoted in Eq. (18):

$$\left[uL_{0,1}(X) + vL_{1,0}(X) + \frac{\partial L_0(X)}{\partial t} \right] + \gamma \sum_{n=1}^N \left[ul_{n,\theta(n)+1}(X) + vl_{n,\theta(n)+1}(X) + \frac{\partial l_{n,\theta}(X)}{\partial t} \right] = 0 \quad (18)$$

Where

$$l_{n,\theta_{(m)+1}}(X) = \sum_{n=1}^N L_{(m)+1,n-m}(X) \cdot g_{m,n-m}(\theta),$$

$$l_{n,\theta_{(n)+1}}(X) = \sum_{n=1}^N L_{m,(n+1)-m}(X) \cdot g_{m,n-m}(\theta).$$

```

1: procedure REAL-TIME HERMITE OPTICAL FLOW METHOD
2:    $\alpha \leftarrow$  Smoothing parameter
3:    $\gamma \leftarrow$  Weight parameter
4:    $Num\_of\_itera \leftarrow$  Number of iterations
5:    $N \leftarrow$  Maximum Hermite polynomial degree
6:    $M \leftarrow$  Square Gaussian window size ( $M + 1$ )
7:    $img1 \leftarrow$  Imagen in time  $t$ 
8:    $img2 \leftarrow$  Imagen in time  $t + 1$ 
9:    $u \leftarrow$  Matrix of zeros of size of  $img1$ 
10:   $v \leftarrow$  Matrix of zeros of size of  $img1$ 
11:  Compute the analysis Hermite filters  $D_{m,n-m}$  from Eq. (5) and (6):
12:  for  $n = 0$  to  $N$  do
13:    for  $m = 0$  to  $n$  do
14:       $L_{m,n-m} = img1 \otimes D_{m,n-m}$   $\triangleright$  Cartesian Hermite coefficients  $img1$ 
15:       $Lw_{m,n-m} = img2 \otimes D_{m,n-m}$   $\triangleright$  Cartesian Hermite coefficients  $img2$ 
16:       $\theta = \arctan \frac{L_{0,1}}{L_{1,0}}$   $\triangleright$  Phase of the gradient of Hermite coefficients  $img1$ 
17:       $\theta w = \arctan \frac{Lw_{0,1}}{Lw_{1,0}}$   $\triangleright$  Phase of the gradient of Hermite coefficients  $img2$ 
18:       $g_{m,n-m}(\theta) = \sqrt{\binom{n}{m}} \cos^m \theta \cdot \sin^{n-m} \theta$   $\triangleright$  Angle functions  $img1$ 
19:       $gw_{m,n-m}(\theta w) = \sqrt{\binom{n}{m}} \cos^m \theta w \cdot \sin^{n-m} \theta w$   $\triangleright$  Angle functions  $img2$ 
20:       $l_{n,\theta} = \sum \sum_{n=1}^N \left( L_{m,n-m} \right) \left( g_{m,n-m}(\theta) \right)$   $\triangleright$  ST coefficients  $img1$ 
21:       $l_{n,\theta_{(m)+1}}(X) = \sum_{n=1}^N L_{(m)+1,n-m}(X) \cdot g_{m,n-m}(\theta)$ 
22:       $l_{n,\theta_{(n)+1}}(X) = \sum_{n=1}^N L_{m,(n+1)-m}(X) \cdot g_{m,n-m}(\theta)$ 
23:       $lw_{n,\theta} = \sum \sum_{n=1}^N \left( Lw_{m,n-m} \right) \left( gw_{m,n-m}(\theta) \right)$   $\triangleright$  ST coefficients  $img2$ 
24:    end for
25:  end for
26:   $L_{0t} = Lw_{0,0} - L_{0,0}$   $\triangleright$  Temporal derivatives
27:   $l_{n,\theta_t} = \sum_{n=1}^N lw_{n,\theta w} - \sum_{n=1}^N l_{n,\theta}$ 
28:  for  $i = 1$  to  $Num\_of\_itera$  do
29:     $\bar{u} = u \otimes \begin{bmatrix} \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \\ \frac{1}{6} & 0 & \frac{1}{6} \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \end{bmatrix}$   $\triangleright$  Compute local averages of the flow
30:     $\bar{v} = v \otimes \begin{bmatrix} \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \\ \frac{1}{6} & 0 & \frac{1}{6} \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \end{bmatrix}$ 
31:    Update  $u$  and  $v$  using the Eq. 27 and 28
32:  end for
33:  return  $u, v$ 
34: end procedure

```

Figure 5. Pseudocode for the Real-Time Hermite Optical Flow [19].

Using variational calculus, the corresponding Euler–Lagrange equations computed are like in Eqs. (19) and (20) where $L_{0t}(X) = \frac{\partial L_0(X)}{\partial t}$, $l_{n,\theta_t}(X) = \frac{\partial l_{n,\theta}(X)}{\partial t}$, and $\nabla^2 u$ represent the Laplacian of u . To this end, **Figure 5** shows the pseudocode for the real-time Hermite optical flow (RT-HOF) iterative solution by Gauss–Seidel.

$$\left[L_{0,1}^2(X) + \gamma L_{0,1}(X) \sum_{n=1}^N \left(l_{n,\theta_{(n)+1}}(X) \right) \right] u + \left[L_{0,1}(X) L_{1,0}(X) + \gamma L_{0,1}(X) \sum_{n=1}^N \left(l_{n,\theta_{(m)+1}}(X) \right) \right] v =$$

$$\alpha^2 \nabla^2 u - \left[L_{0r}(X) + \gamma \sum_{n=1}^N \left(l_{n,\theta_t}(X) \right) \right] \quad (19)$$

$$\left[L_{0,1}(X) L_{1,0}(X) + \gamma L_{1,0}(X) \sum_{n=1}^N \left(l_{n,\theta_{(n)+1}}(X) \right) \right] u + \left[L_{1,0}^2(X) + \gamma L_{1,0}(X) \sum_{n=1}^N \left(l_{n,\theta_{(m)+1}}(X) \right) \right] v =$$

$$\alpha^2 \nabla^2 v - \left[L_{1,0}(X) L_{0r}(X) + \gamma \sum_{n=1}^N \left(l_{n,\theta_t}(X) \right) \right] \quad (20)$$

3. Nature-inspired and intelligent controller for robot navigation

As stated above, this work proposes a nature-inspired and intelligent controller for navigation of mobile robots constrained to a single vision-based sensor, like **Figure 6**. In fact, this artificial organic controller consists of a fuzzy-molecular inference system as the control strategy (i.e., using a hybrid fuzzy inference system with artificial hydrocarbon networks), and a motion feedback from a single vision-based sensor that processes image sequences using the real-time Hermite optical flow approach. Following, main components of the artificial organic controller are described.

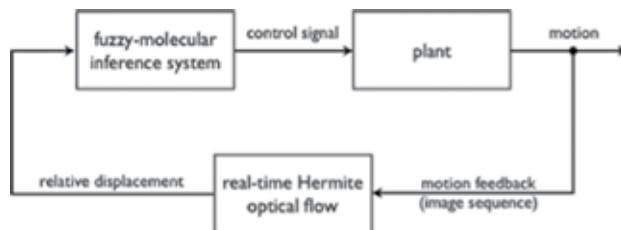


Figure 6. Block Diagram of the Proposed Artificial Organic Control.

3.1. Control law based on a Fuzzy-molecular inference system

Consider a mobile robot with a single vision-based sensor, for example, a video camera, which it can move around a flat environment. If the environment consists of static and dynamic obstacles, the aim of the motion controller is to avoid obstacles as well as to move the robot in a proper way.

In that sense, the robot perceives the environment through the camera, obtaining relative displacements of the objects in front of the robot (see Section 3.2) in two axes: horizontal (\hat{u}) and vertical (\hat{v}). Then, these relative displacements are used as inputs to the robot controller, as shown in **Figure 6**. In particular, the control law is proposed to be implemented using the fuzzy-molecular inference system as follows:

3.1.1. Fuzzification step

The design of this step considers selecting the set of inputs to the controller. As described above, two inputs are used: relative horizontal displacement \hat{u} and relative vertical displacement \hat{v} . Fuzzification also requires partitioning the domain space into k fuzzy sets $\{F_1, \dots, F_k\}$ with their corresponding membership functions $\{\mu_{F_1}(\hat{u}), \dots, \mu_{F_k}(\hat{u})\}$ for input \hat{u} and $\{\mu_{F_1}(\hat{v}), \dots, \mu_{F_k}(\hat{v})\}$ for input \hat{v} . For example, fuzzy partitions for \hat{u} can be: “negative,” “zero,” and “positive”. Notice that the number of fuzzy sets k might be different for each input. It is remarkable to say that the membership functions should be selected accordingly.

3.1.2. Fuzzy inference engine step

This step requires the definition of fuzzy rules with respect to the goal of the controller. Notice that the consequent values y_f in Eq. (6) are computed using the molecular behavior φ_j of the molecule M_j assigned to that specific rule R_j . Thus, a set of *CH*-molecules has to be defined (see below).

3.1.3. Molecular defuzzification step

The last step of the proposed controller considers the computation of the control signal y that has to be selected in order to activate the associated actuator in the robot, for example, the direct current motor coupled to wheels. For instance, if the mobile robot has a differential mechanism with two wheels, then two DC motors are coupled, for example, left motor (y_L) and right motor (y_R). Thus, defuzzification step uses Eq. (7) to compute y_L and y_R , and a set of *CH*-molecules defined for each output signal (y_L and y_R).

As restricted in previous section, the set of *CH*-molecules are arranged in a linear and saturated hydrocarbon compound like Eq. (2). Thus, it is required to set the number of molecules n in the compound, the set of hydrogen atoms H_i for each molecule, the carbon value σ for each molecule, and the set of bounds L_i at which the molecules act. Refer to Eqs. (1) to (4) if needed. As an example, **Figure 7** shows a typical representation of the artificial hydrocarbon compound [15, 16] so-called structural formula, where H values represent the hydrogen values, exponent values at C 's represent carbon values, and L values are the bounds of molecules.

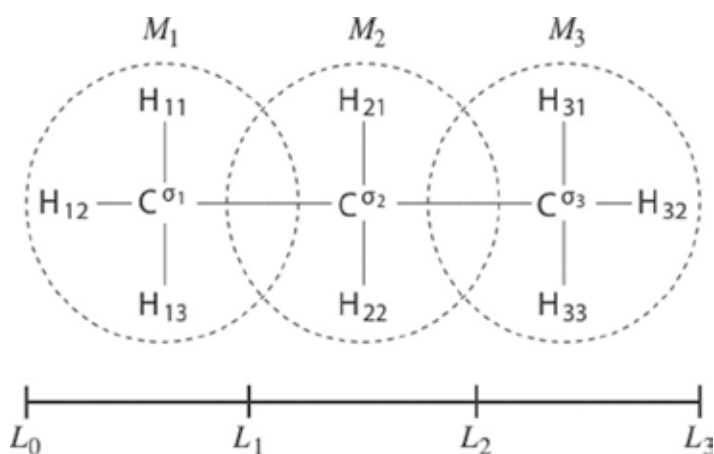


Figure 7. Example of the Structural Formula Representation of an Artificial Hydrocarbon Compound.

3.2. Motion feedback using the RT-HOF approach

A vision-based sensor is proposed to use in mobile robots in order to retrieve motion feedback. In particular, a video camera located in front of the robot can be used. For instance, two adjacent images are obtained. Then, the real-time Hermite optical flow algorithm is used to compute relative displacements of objects between the two images. This procedure outputs a map of displacements. Since the relative displacements in the map are two dimensional vectors, decomposition in horizontal (u) and vertical (v) axes is computed for each displacement vector. To this end, the mean value per axis of all decompositions is calculated, obtaining the relative mean horizontal displacement and the relative mean vertical displacement values, \hat{u} and \hat{v} , respectively.

If one object at the time is in front of the video camera, then \hat{u} represents the mean displacement of the object relative to the horizontal perspective of the image, that is, the displacement of the object from left-to-right or right-to-left in front of the robot, and \hat{v} represents the mean displacement of the object relative to the vertical perspective of the image, that is, the displacement of the object from near-to-far or far-to-near the robot. Thus, both \hat{u} and \hat{v} values can be employed as motion feedback of the robot and inputs of the proposed controller.

4. Implementation on navigation mobile robots

This section introduces the case study employed for implementing and validating the proposed artificial organic controller based on the FMI system and the RT-HOF approach for motion feedback. First, a description of the robotics system is presented. Then, the design of the particular artificial organic controller is shown. Lastly, it describes the process of calibration of the motion feedback system.

4.1. Description of the robotics system

The robotics system used in this case study is based on a previous work reported in [19]. The mobile robot was implemented with the LEGO Mindstorms EV3 as both the mechanical and the electrical platform because it is easy and practical to use. The robot was built as tank configuration using two direct current (DC) motors and two rubber bands as actuators, and the robot was planned to be in a differential steering configuration. Also, a webcam was mounted on the robot to be used as the single vision-based sensor. For experimental purposes, the intelligent brick of the LEGO Mindstorms EV3 set was used as the interface between the mobile robot and a computer with MATLAB. The latter was employed to compute the control of the whole system. Both the webcam and the intelligent brick communicated with the computer via USB. **Figure 8** shows a diagram of the robotic system [19]. The technical specifications of the robotic system are summarized in **Table 1**.

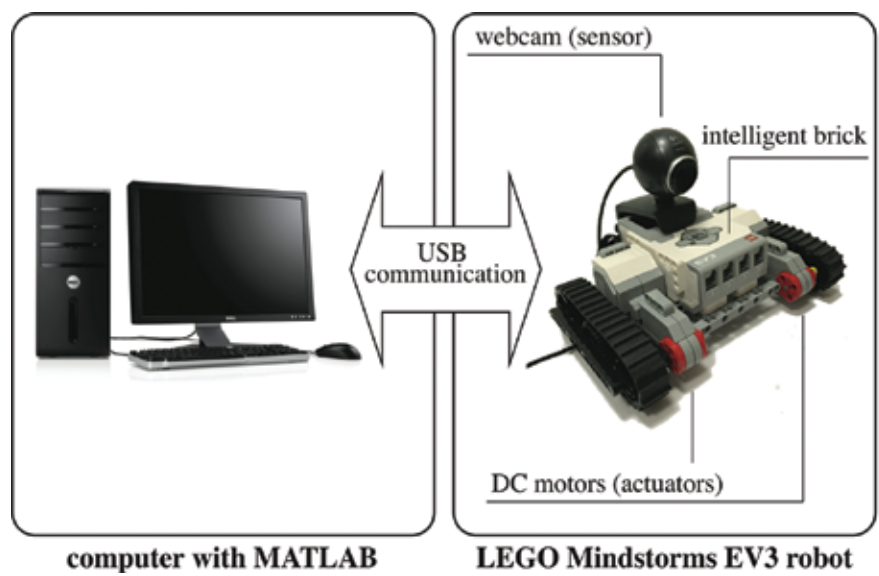


Figure 8. Schematic of the Robotic System Used in the Case Study Based on [19].

Component	Description
Webcam sensor	1.3 M pixels, 1280 × 1024 max resolution
DC motor actuators	9V-input, 0.55A-nominal, 2.03 W mechanical power, 4.95 W electrical power, 160 rpm max nominal
USB communication	9600 baud rate
Windows-based workstation	Intel Xeon Six-Core Processor 2.6 GHz, 32 GB RAM

Table 1. Technical Specifications of the Robotic System [19]

4.2. Design of the artificial organic controller

The fuzzy-molecular inference system-based controller was developed using the methodology described in Section 3. Two inputs were selected as follows: relative mean horizontal displacement \hat{u} and relative mean vertical displacement \hat{v} . Also, the input domain space was partitioned in three fuzzy sets for each input like: “negative” (N), “zero” (Z), and “positive” (P). The proposed membership functions for both inputs \hat{u} and \hat{v} are depicted in **Figures 9** and **10**, respectively.

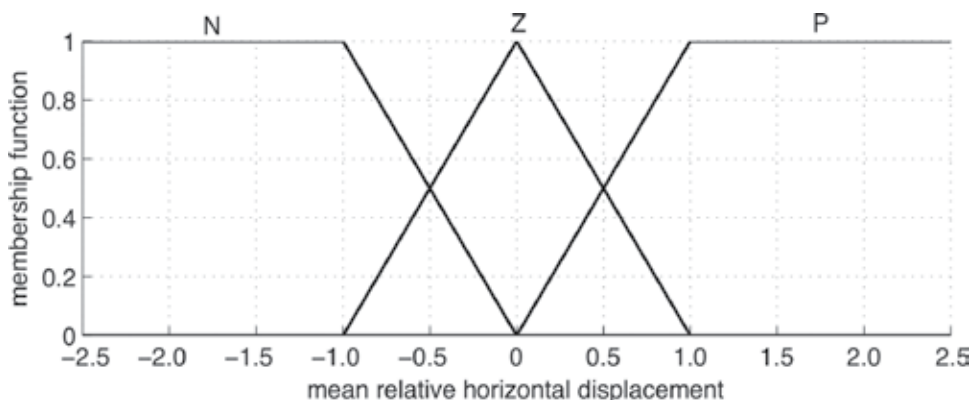


Figure 9. Input Membership Functions of u .

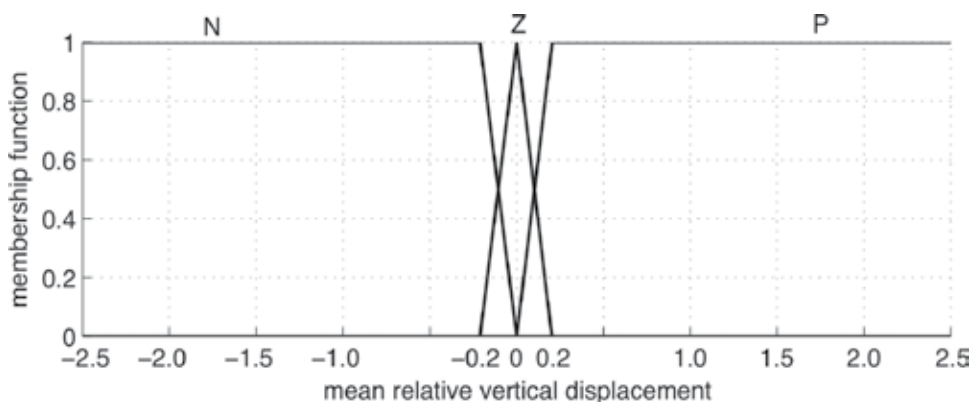


Figure 10. Input Membership Functions of v .

In addition, **Table 2** shows the set of fuzzy rules to this artificial organic controller, and **Figure 11** shows the artificial hydrocarbon compound developed for this case study. Notice that there are three CH-molecules representing “move-backward” (M_B), “stop-motion” (M_S), and “move-forward” (M_F), and two identical compounds are used, one for each wheel in the robot.

\hat{u}	\hat{v}	y_L	y_R
N	N	M_F	M_S
N	Z	M_F	M_F
N	P	M_S	M_F
Z	N	M_F	M_S
Z	Z	M_F	M_F
Z	P	M_S	M_F
P	N	M_F	M_B
P	Z	M_F	M_F
P	P	M_B	M_F

Table 2. Set of Fuzzy Rules Used in the Artificial Organic Controller

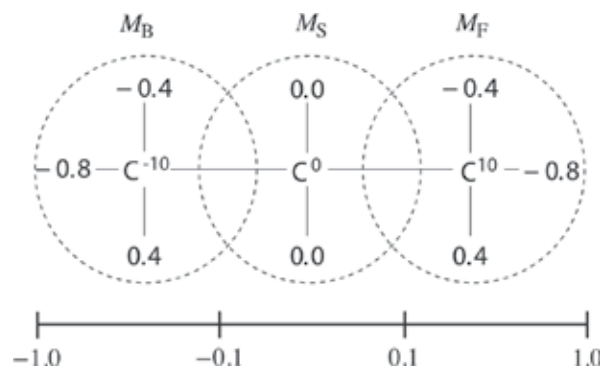


Figure 11. Structural Formula of the Proposed Artificial Hydrocarbon Compound.

4.3. Calibration of the motion feedback

Previous work [19] proved that the real-time Hermite optical flow algorithm (**Figure 5**) can be used in soft and hard real-time frameworks, arising the opportunity to employ this technique in mobile robots.

In fact, the motion feedback system implemented in this work acquires images at 100ms of frame rate. Then, a pair of two adjacent images are analysed using the real-time Hermite optical flow algorithm of **Figure 5**. A map of the displacements is then obtained showing where objects in images are moving. In particular, an automated segmentation is done in order to get only the displacements that reach an empirical threshold, assuming that dynamic obstacles perform displacements greater than this threshold. For this case study, a threshold value of $T = 1.0$ was selected, using trial-and-error. Additionally, this case study supposes that one obstacle at most is in front of the robot and restricts the visual region of the robot at the centre of the frames

(i.e., frames are divided in three vertical regions, selecting the central one). Later on, the automated segmentation finds the dynamic obstacle and computes its local displacements. To this end, the robotics system computes the mean values \hat{u} and \hat{v} , as described in Section 3.2.

5. Experimental results

In order to validate the performance of the proposed controller, three tests were conducted about the actions taken from the mobile robot when dealing with dynamic obstacles. In addition, a comparison between the proposed controller and a rule-based controller previously developed in [19] was run. In the following section, these experimental results are summarized.

5.1. Tests on avoiding dynamic obstacles

This experiment consisted on steering the robot when dynamic objects are across the vision range of it. In particular, the robot avoids the obstacle in the opposite direction of its movement. In that sense, the proposed artificial organic controller is able to compute the relative displacement of the obstacle with respect to the robot, and then, the nature-inspired control strategy could calculate the power supply offering to the two motors, independently.

In advance, three tests were conducted. The first test considered the dynamic object (i.e., another robot) to move from left-to-right perpendicular to the direction of the robot, expecting that the robot steers to the left. The second test consisted in the same dynamic object, but moving from right-to-left perpendicular to the direction of the robot, expecting that the latter moves to the right. Finally, the third test considered a dynamic obstacle moving in a perpendicular direction with respect to the direction of the robot. **Tables 3 to 5** summarize the mean relative horizontal \hat{u} and vertical \hat{v} displacements computed with the RT-HOF algorithm, the output power supplies for both motors, left y_L and right y_R , in the robot, and the interpretation of the steering direction of the robot. In addition, **Figures 12 to 14** show the trajectories obtained from the artificial organic controller in the robot during the three tests.

\hat{u}	\hat{v}	y_L	y_R	direction
0.00	0.00	15.1	15.1	straight
1.73	0.88	-15.1	15.1	left (fast)
1.22	2.09	-15.1	15.1	left (fast)
1.01	-0.03	0.0	8.7	left (slow)
0.00	0.00	15.1	15.1	straight

Table 3. (Left-to-Right Test) Inputs and Outputs Values of the Artificial Organic Controller

\hat{u}	\hat{v}	y_L	y_R	direction
-2.34	-0.64	15.1	0.0	right (slow)
-0.86	-0.53	8.0	-0.2	right (slow)
-0.26	-1.18	4.2	4.5	straight
0.11	0.61	9.5	9.2	straight
-1.01	-0.36	15.1	0.0	right (slow)
-0.56	-0.26	1.3	0.2	stop

Table 4. (Right-to-Left Test) Inputs and Outputs Values of the Artificial Organic Controller

\hat{u}	\hat{v}	y_L	y_R	direction
-1.39	0.28	15.1	-15.1	Right (fast)
-0.24	0.23	4.7	5.2	Straight
0.28	0.16	2.8	2.3	Straight
-0.48	0.22	1.2	0.3	Stop
-1.29	0.90	15.1	-15.1	Right (fast)

Table 5. (Diagonal Test) Inputs and Outputs Values of the Artificial Organic Controller

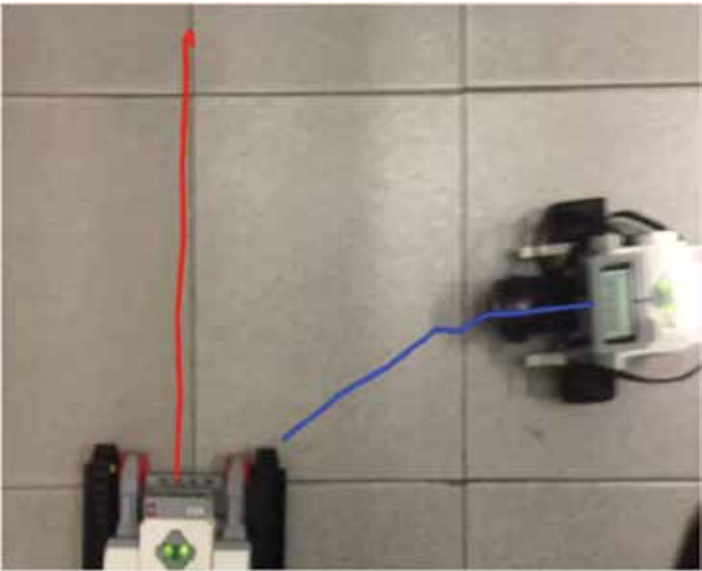


Figure 12. Test 1 (Left-to-Right): Trajectory Response of the Artificial Organic Controller.

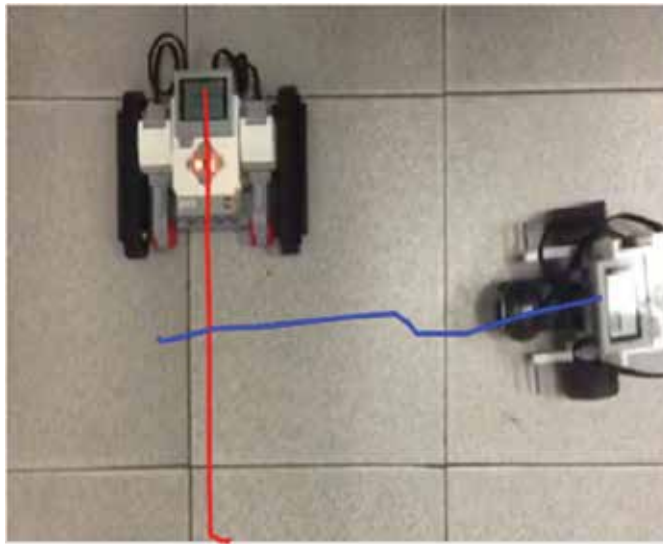


Figure 13. Test 2 (Right-to-Left): Trajectory Response Of The Artificial Organic Controller.

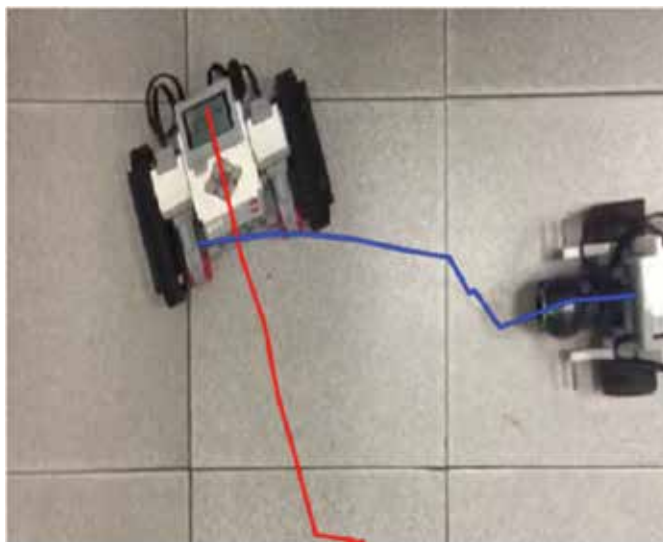


Figure 14. Test 3 (Diagonal): Trajectory Response of the Artificial Organic Controller.

Observing at **Figures 12** and **13**, the correction of the direction is done fast in the robot while looking the dynamic obstacle in front. Reactiveness in the third test (**Figure 14**) is quite lower than in the first two tests. Currently, the robot can avoid obstacles in all situations. As an example, **Figure 15** shows the visual range of the robot and the relative displacements obtained from the RT-HOF algorithm.

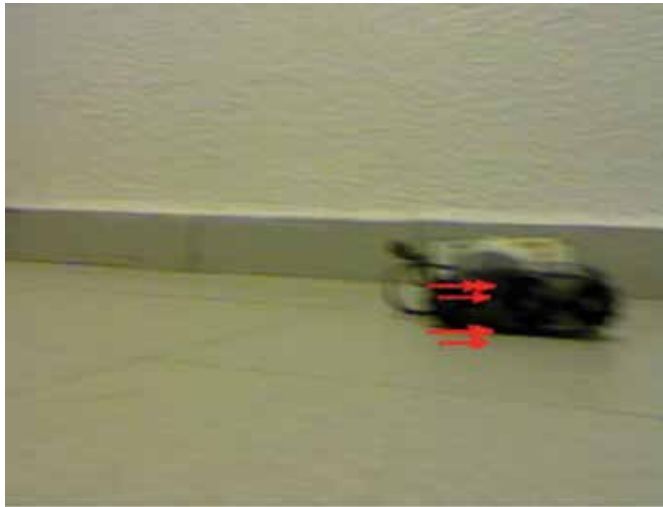


Figure 15. Example of the Visual Range of the Robot and the Relative Displacements Obtained from the RT-HOT Algorithm.

5.2. Comparison of mobile robot navigation controllers

An inspection comparison was also done in order to highlight advantages and limitations on the performance of the proposed artificial organic controller. In fact, the controller in the previous work [19] reported a rule-based controller. **Figures 16 to 18** show the response of the avoidance obstacles in the same configuration of tests: Dynamic object moves from left-to-right, from right-to-left and in diagonal, respectively.

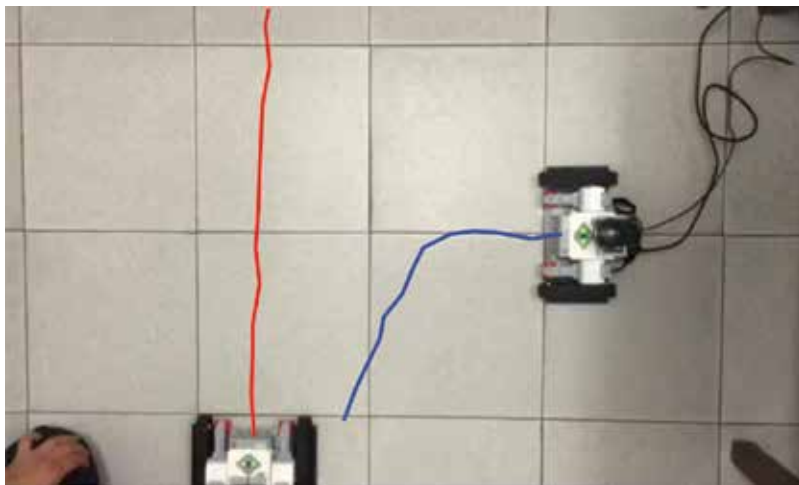


Figure 16. Test 1 (Left-to-Right): Trajectory Response of the Rule-Based Controller [19].

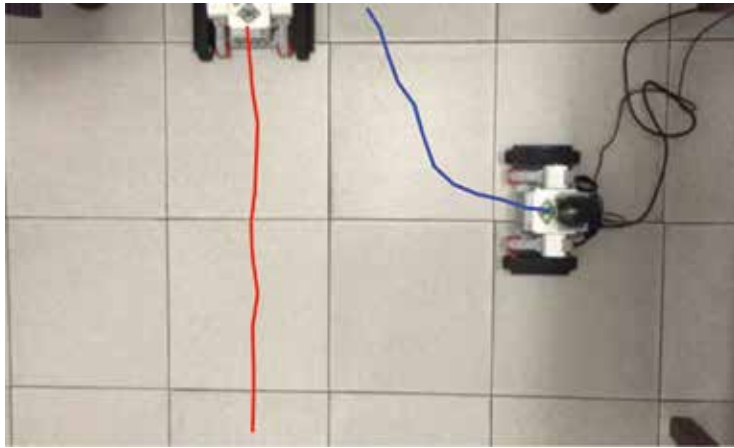


Figure 17. Test 2 (Right-to-Left): Trajectory Response of the Rule-Based Controller [19].

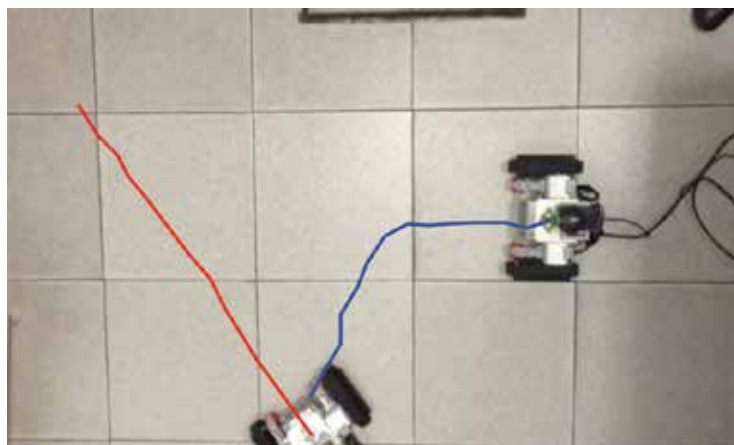


Figure 18. Test 3 (Diagonal): Trajectory Response Of The Rule-Based Controller [19].

First, the new artificial organic controller has better performance in terms of time reaction. Using the rule-based controller, the robot reacts in large time with respect to the distance of the dynamic object. It also considers a slow action that can be seen in the curvature of the depicted trajectory, and the robot does not return to the main direction (once the direction is corrected, the robot does not go back to the previous direction). The latter might be a disadvantage in the overall time of navigation, and the orientation of the robot could be lost.

On the other hand using the proposed artificial organic controller, the robot reacts in short time with respect to the distance of the dynamic object. In addition, once the steering is performed, the power supply to the wheels is recovered to the initial power. Moreover, the action is done fast (in contrast with the other controller) and the direction of the robot comes

back close to the initial orientation. As an advantage, this controller might be used to surround dynamic obstacles that can minimize the overall time of the performance.

It is remarkable to say that the rule-based controller is a reactive approach in contrast to the deliberative approach performed with the artificial organic controller. From the perspective of agents systems, deliberative approaches can deal with uncertainty, imprecise, and noise in a suitable way [37]. In addition, the artificial organic controller can be more scalable, reliable, and easy to understand. In advance, using the hybrid fuzzy-molecular inference system gives an easy way to tune and interpret the performance of the controller, in comparison with the rule-based controller. Lastly, learnability is allowed in the proposed controller.

6. Conclusions

This chapter presents a novel artificial organic controller for mobile robot navigation, based on a hybrid fuzzy-molecular inference system as the nature-inspired control strategy and a single vision-based motion feedback using the real-time Hermite optical flow approach.

In fact, this artificial organic controller proposed the fuzzy-molecular inference system because it inherits characteristics from fuzzy inference systems like storage of knowledge from experts, dealing with uncertainty, and linguistic interpretation; but also from artificial hydrocarbon networks method that allows packaging information in molecular units, organizing of information, and dealing with aggressive uncertainty and noise. In addition, this proposed controller uses a real-time Hermite optical flow approach.

Experimental results showed that the proposed artificial organic controller can avoid dynamic obstacles. In addition, the comparison with a rule-based controller highlights that the proposed controller is easy to implement and to understand. Also, it is scalable and reliable in terms of uncertainty, imprecise, and noise.

For future work, this research is looking for implementing a learning mechanism in the artificial organic controller in order to automate the adaptation to the environment, and membership functions could be tuned automatically. To this end, complex environments should be tested to improve the real-time Hermite optical flow algorithm.

Author details

Hiram Ponce*, Ernesto Moya-Albor and Jorge Brieva

*Address all correspondence to: hponce@up.edu.mx

Vision, Machines & Signals Group, Faculty of Engineering, Universidad Panamericana, Mexico City, Mexico

References

- [1] Barrera, A., editor. *Mobile Robots Navigation*. ISBN: 978-953-307-076-6, InTech; 2010. pp. 680. Available from: <http://www.intechopen.com/books/mobile-robots-navigation>
- [2] Llorca, D., Milanes, V., Alonso, P., Gavilan, M., Daza, I., Perez, J., Sotelo, M., *Autonomous Pedestrian Collision Avoidance Using a Fuzzy Steering Controller*. IEEE Transactions on Intelligent Transportation Systems. 2011; 12 (2):390–401.
- [3] Tsankova, D., *Neural Networks Based Navigation and Control of a Mobile Robot in a Partially Known Environment*. In: Barrera, A., editor. *Mobile Robots Navigation*. ISBN: 978-953-307-076-6, InTech; 2010. p. 197–222. Available from: <http://www.intechopen.com/books/mobile-robots-navigation/neural-networks-based-navigation-and-control-of-a-mobile-robot-in-a-partially-known-environment>
- [4] Kayacan, E., Ramon, H., Kaynak, O., Saeys, W., *Towards Agrobots: Trajectory Control of an Autonomous Tractor Using Type-2 Fuzzy Logic Controllers*. IEEE/ASME Transactions on Mechatronics. 2014; 20 (1):287–298.
- [5] Ponce, P., Molina, A., Mendoza, R., Ruiz, M., Monnrad, D., Fernandez, L., *Intelligent Wheelchair and Virtual Training by LabVIEW*. In: Sidorov, G., Hernandez, A., Reyes, C., editors. *Advances in Artificial Intelligence*. Switzerland: Springer; 2010. p. 422–435. doi:10.1007/978-3-642-16761-4_37
- [6] Ponce, P., Molina, A., Mendoza, R. *Wheelchair and Virtual Environment Trainer by Intelligent Control*. In: Iqbal, S., Boumella, N., Figueroa, J., editors. *Fuzzy Controllers – Recent Advances in Theory and Applications*. InTech; 2012.
- [7] Juang, C., Lai, M., Zeng, W., *Evolutionary Fuzzy Control and Navigation for Two Wheeled Robots Cooperatively Carrying an Object in Unknown Environments*. IEEE Transactions on Cybernetics. 2014; 45 (9):1731–1743.
- [8] Juang, C., Chang, Y., *Evolutionary-Group-Based Particle-Swarm-Optimizer Fuzzy Controller with Application to Mobile-Robot Navigation in Unknown Environments*. IEEE Transactions on Fuzzy Systems. 2011; 19 (2):379–392.
- [9] Ponce, H., editor. *Nature-Inspired Computing for Control Systems*. 1st edn. Switzerland: Springer; 2016. pp. 289. doi:10.1007/978-3-319-26230-7
- [10] Williams, O., Blake, A., Cipolla, R., *A Sparse Probabilistic Learning Algorithm for Real-Time Tracking*. In: 9th IEEE International Conference on Computer Vision; 2003.
- [11] Subramanian, K., Suresh, S., *Human Action Recognition Using Meta-Cognitive Neuro-Fuzzy Inference System*. In: 2012 International Joint Conference on Neural Networks; 2012. p. 1–8.

- [12] Boroujeni, N., Etemad, S., Whitehead, A., Fast Obstacle Detection Using Targeted Optical Flow. In: 19th IEEE International Conference on Image Processing; 2012. p. 65–68.
- [13] Narayanan, V., Crane, C., Active Relearning for Robust On-Road Vehicle Detection and Tracking Control. In: 13th International Conference on Automation and Systems; 2013. p. 124–129.
- [14] Crnko, P., Capson, D., Biomimetic Measurement of Optical Flow and Centroid for Visual-Servo Control of Hover Flight. In: 2012 IEEE International Instrumentation and Measurement Technology Conference; 2012. p. 348–353.
- [15] Ponce, H., Ponce, P., Molina, A., Artificial Hydrocarbon Networks Fuzzy Inference System. *Mathematical Problems in Engineering*. 2013; 2013 (2013):531031. doi: 10.1155/2013/531031
- [16] Ponce, H., Ponce, P., Molina, A., Artificial Organic Networks: Artificial Intelligence Based on Carbon Networks. Switzerland: Springer; 2014. pp. 228.
- [17] Molina, A., Ponce, H., Tello, G., Ramirez, M., Artificial Hydrocarbon Networks Fuzzy Inference Systems for CNC Machines Position Controller. *The International Journal of Advanced Manufacturing Technology*. 2014; 72 (9):1465–1479.
- [18] Moya-Albor, E., Escalante-Ramirez, B., Vallejo, E., Optical Flow Estimation in Cardiac CT Images Using the Steered Hermite Transform. *Signal Processing*. 2013; 28 (3):267–291.
- [19] Moya-Albor, E., Brieva, J., Ponce, H., Mobile Robot With Movement Detection Controlled by a Real-Time Optical Flow Hermite Transform. In: Ponce, H., editor. *Nature-Inspired Computing for Control Systems*. Switzerland: Springer; 2016. p. 231–263.
- [20] Ponce, H., Ponce, P., Molina, A., Adaptive Noise Filtering Based on Artificial Hydrocarbon Networks: An Application to Audio Signals. *Expert Systems with Applications*. 2013; 41 (14):6512–6523.
- [21] Ponce, H., Ponce, P., Molina, A., The Development of an Artificial Organic Networks Toolkit for LabVIEW. *The Journal of Computational Chemistry*. 2015; 36 (7):478–492.
- [22] Ponce, H., Ponce, P., Molina, A., A Novel Robust Liquid Level Controller for Coupled-Tanks Systems Using Artificial Hydrocarbon Networks. *Expert Systems with Applications*. 2015; 42 (22):8858–8867.
- [23] Ponce, H., Miralles-Pechuan, L., Martinez-Villaseñor, M., Artificial Hydrocarbon Networks for Online Sales Prediction. In: Pichardo, O., Herrera, O., Arroyo, G., editors. *Advances in Artificial Intelligence and Its Applications*; Springer; 2015. p. 498–508. doi: 10.1007/978-3-319-27101-9_38
- [24] Ponce, H., Martinez-Villaseñor, M., Miralles-Pechuan, L., Comparative Analysis of Artificial Hydrocarbon Networks and Data-Driven Approaches for Human Activity Recognition. In: Garcia-Chamizo, J., Fortino, G., Ochoa, S., editors. *Ubiquitous Com-*

- pating and Ambient Intelligence: Sensing, Processing, and Using Environmental Information; Springer; 2015. p. 150–161. doi:10.1007/978-3-319-26401-1_15
- [25] Ponce, H., Ibarra, L., Ponce, P., Molina, A., A Novel Artificial Hydrocarbon Networks Based Space Vector Pulse Width Modulation Controller for Induction Motors. *American Journal of Applied Sciences*. 2014; 11 (5):789–810. doi:10.3844/ajassp.2014.789.810
 - [26] Gibson, J., The perception of the visual world. *The American Journal of Psychology*. 1951; 64 (1):622–625.
 - [27] Liu, C., Freeman, W., Adelson, E., Weiss, Y., Human-Assisted Motion Annotation. In: *IEEE Conference on Computer Vision and Pattern Recognition*; IEEE; 2008. p. 1–8.
 - [28] Horn, B., Schunck, B., Determining Optical Flow. *Artificial Intelligence*. 1981; 17 (1): 185–203.
 - [29] Nagel, H., Constraints for the Estimation of Displacement Vector Fields from Image Sequences. In: *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*; 1983. p. 945–951.
 - [30] Nagel, H., Enkelmann, W., An Investigation of Smoothness Constraints for the Estimation of Displacement Vector Fields from Image Sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1986; 8 (1):565–593.
 - [31] Martens, J., The Hermite Transform: Theory. *IEEE Transactions on Acoustics, Speech and Signal Processing*. 1990; 38 (1):1595–1606.
 - [32] Young, R., The Gaussian Derivative Theory of Spatial Vision: Analysis of Cortical Cell Receptive Field Line-Weighting Profiles. Detroit, Mich, USA: General Motors Research Laboratories; 1985.
 - [33] Silvan-Cardenas, J., Escalante-Ramirez, B., The Multiscale Hermite Transform for Local Orientation Analysis. *IEEE Transactions on Image Processing*. 2006; 15 (1):1236–1253.
 - [34] Van Dijk, A., Martens, J., Image Representation and Compression With Steered Hermite Transforms. *Signal Processing*. 1997; 56 (1):1–16.
 - [35] Levkin, H.G., Image Processing/Videocodex/Programming [Internet]. 2004. Available from: <http://www.hlevkin-com/TestImages/additional.htm>. Accessed: 20 Jan 2013.
 - [36] Moya-Albor, E., Optical Flow Estimation Using the Hermite Transform [thesis]. Mexico City, Mexico: Universidad Nacional Autonoma de Mexico; 2013.
 - [37] Weiss, G., Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. London, UK: MIT Press; 2000. pp. 648.



*Edited by Pedro Ponce, Arturo
Molina Gutierrez and Jaime Rodriguez*

This book has a complete set of applications of artificial neural networks that allow the reader to gain experience about the new systems for implementing and developing artificial intelligence (AI) methods, which can run in several digital systems. On the other hand, the book shows the newest algorithms of artificial intelligence that provide a wide spectrum of research areas in which AI can be deployed.

There are a lot of books that address AI applications. However, this book shows the newest applications reached according with the technological changes that are presented nowadays. Those changes drastically appear in digital systems or other parallel areas that allow to improve the performance of AI algorithms. Hence, sometimes, the AI algorithms have to be redesigned in order to run in microcontrollers or FPGAs. The topics covered generate a structured book, so it could be used as a textbook, but it is designed to be accessible to a wide audience interested in AI.

Photo by vchal / iStock

IntechOpen

