IntechOpen

Design, Control and Applications of Mechatronic Systems in Engineering

Edited by Sahin Yildirim

DESIGN, CONTROL AND APPLICATIONS OF MECHATRONIC SYSTEMS IN ENGINEERING

Edited by **Şahin Yildirim**

Design, Control and Applications of Mechatronic Systems in Engineering
http://dx.doi.org/10.5772/65144
Edited by Sahin Yildirim

Contributors

Mustafa Engin, Ren-Jung Chang, Y. H. Lai, Marian Gaiceanu, Luis Ismael Minchala-Avila, Darwin Astudillo, Kenneth Palacio, Andres Vazquez, Ümit Önen, Fatih Botsali, Yusuf Şahin, Mete Kalyoncu, Mustafa Tınkır, Dilşad Engin, Jesús García-Guzmán, Farah Villa-López, Jorge Alberto Vélez-Enríquez, Alfredo Ramírez-Ramírez, Luz Aurora García-Mathey

Notice

First published in Croatia, 2017 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

Design, Control and Applications of Mechatronic Systems in Engineering
Edited by Sahin Yildirim

p. cm.

Print ISBN 978-953-51-3125-0

Online ISBN 978-953-51-3126-7

eBook (PDF) ISBN 978-953-51-4853-1

Meet the editor

Dr. Yildirim has received his MSc degree in Mechanical Engineering in 1990 from Erciyes University, Turkey. He has received his PhD degree in 1998 from Systems Engineering Department, Cardiff University, UK. He has established Mechatronics Engineering Department in 2005, Kayseri, Turkey. He is head of the Mechatronics Engineering Department and head of Foreign Relations Office of Erciyes University. He is lecturing the control theory, robotics, principles of mechatronic systems, and neural network applications in engineering at Erciyes University, Turkey. He has supervised some national and international projects such as mobile nurse robot design and control, active vehicle system design and control, and two-legged walking robot control. He has reviewed more than 200 papers. He is also author of more than 150 journal articles, conference papers, and book chapters. He is currently supervising more than 30 PhD and MSc students and editing 4 journals. His research area consists of neural network, mechatronic systems, control theory and applications, mobile and industrial robots, and vehicle dynamics and control.

Contents

Preface

Mechatronics combines all fields of mechanical and electrical engineering. This book deals with designed, produced, and controlled mechatronic systems in engineering and real-time applications. Designs to improve mechatronic system's performance by the use of advanced technology instruments in their construction inevitably introduce flexibility into the structure of the system.

The approach described in Chapter 1 is to develop an embedded controller design concept for mechatronic sun tracker with the highest energy production efficiency. On this developed real-time control, algorithm uses hybrid control loop by combining the energy-efficient open loop with the high tracking accuracy closed loop. The algorithm focused not only on improving the energy production efficiency but also on decreasing the power consumption during tracking. Designed dual axis sun tracker and fixed PV system were tested during the year. The system performance parameters for both systems were measured and stored with data logger. Collected data were examined and compared. On a clear-sky day, performance data proved that the sun tracker produced 40.7% more energy than the fixed one. The efficiency of the sun tracker decreases to 32.1% on partly cloudy day.

Nowadays, laboratory works have been increased for improving real-time practical experience. This kind of application is outlined in Chapter 2. This chapter focuses on designs in the didactical manner and allows students and researchers to implement autonomously the mechatronic projects based on the DC drive. Starting from the nameplate data of the DC motor, the provided MATLAB software has the ability to calculate the DC motor parameters and the parameters of the current and speed controllers, and includes the numerical implementation of the DC motor-cascaded control and energy balance, and of the load torque estimator. The price of the proposed DC drive is down due to the fact that the load torque is very expensive and it is replaced by the load torque estimator. By using the Z transform, the procedure of numerical control is implemented. The implementation of the digital filters and controllers is also provided. Numerical simulations confirm the accuracy of the proposed DC drive discretization method. Moreover, the real-time control based on Arduino platform has been performed. In order to increase the DC drive performances, the online estimator of the DC motor parameters is provided, and the feed-forward control voltage component is added.

There are some laboratory instruments and practical application mechatronic systems for education training and teaching. In order to increase real-time application background for students, web-based laboratory system is described in Chapter 3. This chapter focuses on the implementation of web-based laboratories that allow the remote operation of experiments used as training exercises in undergraduate engineering courses. The remote laboratories were developed using LabVIEW® software, and they enable remote control and

monitoring of laboratory equipment, allowing engineering students to perform experiments in real time, at their own pace, from anywhere and whenever it is suitable for them. Besides the experimental training that the web-based laboratories provide to students, the system is also a powerful teaching tool since real-time demonstrations of the experiments can be performed, and they also can be simultaneously monitored by a group of students. This approach is highly beneficial for engineering schools in developing countries, as resources can be shared through the Internet. A description of the system and three proposed experiments is presented, together with the experimental results.

Micromechatronic System's (MMS) design and control have been increased to overcome some problems in engineering applications. Chapter 4 proposes a micromechatronic gripper system. The proposed system has been designed, fabricated, and tested. In this investigation, by following realization axioms, the micromechatronic gripper system including polyurethane (PU) gripper mechanism and shape memory alloy (SMA) actuator was also designed and developed. The micromechatronic gripper system was realized with a cross-sectional area of ($\pi/4$) ×5002μm2 for clean room operation. A synergetic operation of SMA actuator for driving micro-gripper mechanism was investigated in visual-based control. By incorporating with inverse Preisach compensator, an explicit self-tuning controller through Ziegler-Nichols criterion was selected for controlling the self-biased SMA actuator. The application of the gripper system for gripping and transporting a glass particle of 30μm was tested.

Sensors and digital signal conditioning are the main components of mechatronic systems. Mechatronic system's performance and accuracy can be improved with these components. This improvement is investigated in Chapter 5. This chapter presents a measurement system in mechatronics consisting of temperature sensors and signal conditioning circuits, providing detailed information on design process of an embedded measurement and linearization system. This system uses a 32-bit microcontroller for thermocouple (T/C) cold junction compensation, amplification of low output voltage, then conversion to digital, and linearization of the type K thermocouple's output by software to output a desired signal. Piecewise, look-up table and spline (polynomial degree>1) methods are used in linearization software, and the implemented embedded system for linearization of a type K T/C is presented as an example study. The obtained results are compared to give an insight to the researchers who work on measurement systems in mechatronics.

Due to advanced technology, some scientific real-time design and control applications on lower limb exoskeleton for disabled people who have walking difficulties have been investigated. In Chapter 6, an approach of a lower limb exoskeleton mechatronic design is presented. However, the design aims to be used as a walking support device focused on patients who suffer of partial lower body paralysis due to spine injuries or a stroke. First, the mechanical design is presented, and their results are validated through dynamical simulations performed in Autodesk Inventor and MATLAB. Second, a communication network design is proposed in order to establish a secure and fast data link between sensors, actuators, and microprocessors. Finally, patient-exoskeleton system interaction is presented and detailed. Movement generation is performed by means of digital signal processing techniques applied to electromyography (EMG) and electrocardiography (EEG) signals. Such interaction system design is tested and evaluated in MATLAB, for which results are presented and explained. A proposal of real-time supervisory control is also presented as part of the integration of every component of the exoskeleton.

Advanced controller is very important to lower limb exoskeleton for stability, supporting walking conditions in regular prescribed trajectories of legs. Chapter 7 presents the results of research work on design, actuator selection, and motion control of a lower extremity exoskeleton developed to provide legged mobility to spinal cord–injured (SCI) individuals. The exoskeleton has two degrees of freedom per leg. Hip and knee joints are actuated in the sagittal plane by using DC servomotors. Additional effort supplied by the user's arms through crutches is defined as user support rate (USR). Experimentally determined USR values are considered in actuator torque computations for achieving a realistic actuator selection. A custom-embedded system is used in the control of the exoskeleton. Reference joint trajectories are determined by using clinical gait analysis (CGA). Three-loop cascade controllers with current, velocity, and position feedback are designed for controlling the joint motions of the exoskeleton. A nonlinear ARX model is used in the determination of controller parameters. Overall performance and assistive effect of WSE-2 are experimentally investigated by conducting tests with a paraplegic patient with T10 complete injury.

Prof. Dr. Şahin Yildirim
Erciyes University,
Kayseri, Turkey

Chapter 1

Embedded Controller Design for Mechatronics System

Mustafa Engin

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/67987

Abstract

Many mechatronic systems have challenging control difficulties due to the high nonlinear structures and time-varying dynamic behaviors. In addition to these, there are external disturbances, which cannot be predicted and change according to uncontrolled working environments. Conventional controllers are insufficient to solve the aforementioned problems and to compensate the environmental disturbances. Therefore, adaptive controllers have been proposed as a solution to these inefficiencies of conventional controllers. Adaptive control is applied for solving the control problem of the mechatronic sun tracker that ensures the movement of the mechanism to harvest maximum energy coming from solar to the PV module surface during the sunshine duration. For this type of control problems, conventional controllers are very limited and they have a lot of deficiencies. The adaptive mechanism governed by an adaptation law is the heart of any adaptive controller. We establish the adaptation law for the plant control system using Lyapunov stability theory. This adaptation law is precise for a generic second-order system; hence, it is obviously applicable for adaptive control of other second-order systems in different realms, such as industrial production system, military, and robotics.

Keywords: embedded system, microcontroller, interfacing, adaptive control, sun tracker

1. Introduction

Time-varying dynamic behaviors of many mechatronics system and the existence of unpredictable external disturbances also present challenging control problems. If conventional controllers such as propotional integral derivative (PID), controllers are used, they must be re-tuned for different operating conditions. This task is costly, time-consuming, and even unfeasible. Unlike conventional controllers, adaptive control systems use special adaptation mechanisms that can self-tune their control parameters to recompense unforeseen deviations in the controller

circuit, sensors, and their signal processing sections or in the system workplace environment. Therefore, adaptive control strategy has been progressively applied for solving difficult control problems. Adaptive control system design is much complicated than conventional controller design due to the complexity of adaptive structure and the issue of unknown or time-varying parameters.

In this chapter, we have shown the design-verification flow with various phases: mathematical model; process calculation and modeling; controller design in both continuous-time and discrete spaces; controller simulation; and in-hardware implementation and testing. The mathematical model, which is given in terms of a set of mathematical equations of energy and material balances, provides insightful understanding of the dynamic behaviors of the plant. The process calculation is necessary to obtain steady-state data for control system design in later phases. The reference model is constructed with a mathematical approach and proven to be stable and to ensure the sun tracker's steady-state properties. The discrete-time version must be accomplished prior to implementation of the adaptive design on digital computers. We successfully design and implement the embedded adaptive controller using ARM Cortex-M core processor. We apply the state-of-the-art testing technique in which we took performance data over a data logger. During the system test process, the embedded model reference adaptive controller effectively controlled the attitude of sun tracker and completely fulfilled all controller objectives.

2. Principles of adaptive control

The interest in designing autonomous working machine for such mechatronic systems motivated the development of adaptive control theory, and first steps were taken to automatically adjust the controller parameters to changing mechanism dynamics during the working. There were some stability problems, which arose with the usage of the developed adaptive control theory. Several researchers studied instability of and modifications in the adaptation mechanism of the adaptive control [1]. After these studies, several improvements were added to the theory, and efforts to develop the adaptive control theory have continued to the present day.

As shown in **Figure 1**, an adaptive control system has two loops. The first one is feedback loop same as all traditional closed-loop control systems. Second one is parameter adjustment loop, which is inherently slow feedback loop. Adaptation scheme can be adjusted to controller parameters in two different methods. In the indirect method, at any adjustment step, new controller parameters are obtained by solving the design problem such as self-tuning regulators. Direct adaptive controller technique means that the controller parameters are directly updated each time step by means of an adjustment rule without the characteristics of the process, and its disturbances being determined first. In the literature, it is seen that three different adaptive control schemes are preferred in the control of mechatronic systems. First one is gain-scheduling adaptive control. It needs additional measurement about environment and plant operation parameters. Second one is self-tuning adaptive control. It observes the plant output and adjusts controller gain to maintain the desired output level. Third one is model reference adaptive control. It uses a model that is approximately similar with plant. Every step compares the reference model output and plant output and then adjusts controller gain to get the desired plant output level.

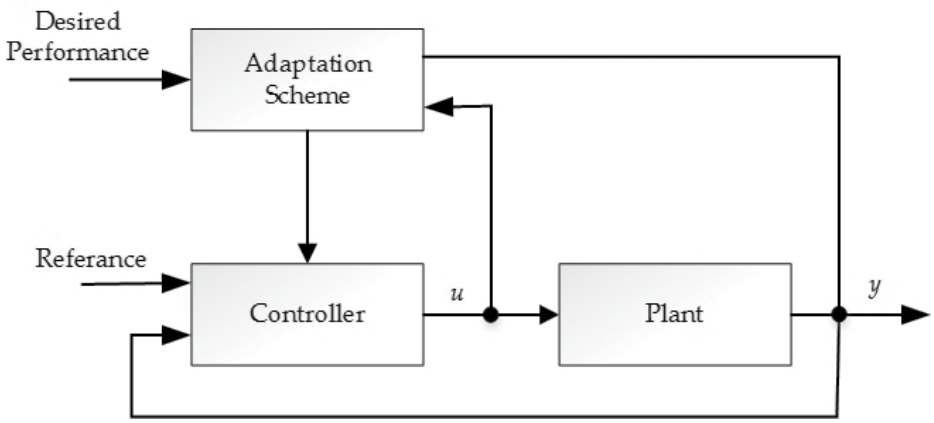

Figure 1. Block diagram of general adaptive controller.

2.1. Gain-scheduling adaptive control scheme

The gain-scheduling approach comes out of the simple idea of changing the gains of a controller according to variations in the operation. This method is very strong in cases where the parameters of the plant change with respect to a subsidiary variable that is available for measurement. Gain-scheduling control in fact is a mapping operation from plant parameters to controller parameters; hence, it can be implemented as a function or a lookup table especially in digital systems. As shown in **Figure 2**, gain-scheduling scheme needs auxiliary measurements to adjust the controller gain.

Despite the many successful industrial applications, traditional gain-scheduling control was considered as an applied approach before 1990 and its theoretical properties were not sufficiently studied [2]. Traditional gain-scheduling linearizes the plant model on predefined operating point and a local controller is set up around this point. Giving probable disturbance variation to the local controller as input and measuring plant output parameter for each input,

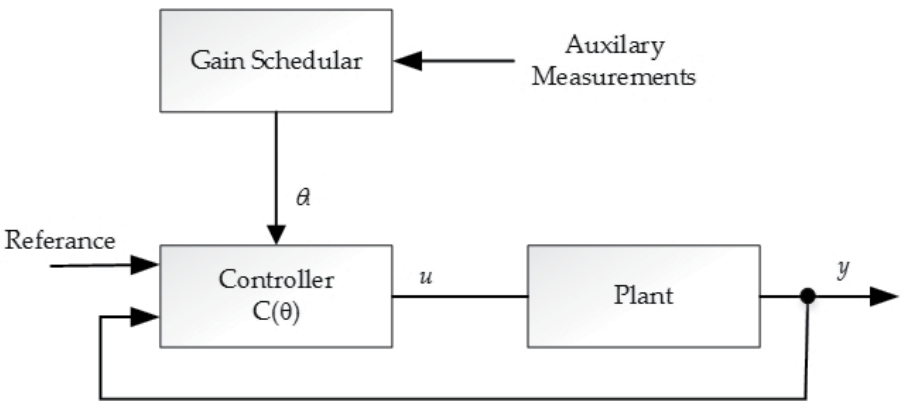

Figure 2. Adaptive controller with gain-scheduling scheme.

the measured output parameter is averaged to define plant output, also controller gain. Controller output unification is such a technique, for example used in Ref. [3], where different controllers are applied at the same time, then outputs are interpolated to determine controller output. Concerning closed-loop behavior, this attitude brings on fine outcomes; nonetheless, the closed-loop stability is only guaranteed for selected fixed operating points. The biggest advantage of traditional gain-scheduling scheme is that gain-scheduling controller always exists when stability conditions are covered.

Modern gain-scheduling control techniques that guarantee global stability were developed in the 1990s [4, 5]. These techniques need online measurements for the varying parameters and also their derivatives. Although a large amount of research has recently been carried out on synthesis and analysis, the existing techniques can still be very conservative for practical applications [6–8]. Before elaborating on the modern gain-scheduling control techniques found in the literature, first some techniques are presented to identify linear parameter varying (LPV) models, since modern gain-scheduling control techniques synthesize controllers directly from LPV of the system. LPV modeling techniques are still under development and a practical and easy algorithm to make an LPV model from experimental system measurements is still missing [9]. In some applications, LPV models are obtained by transforming the underlying physical laws to a parameter-dependent state-space form [10]. This approach remains restricted to academic examples, since it is too time-consuming to reveal the underlying physical laws for industrially relevant large-scale systems. Furthermore, it is difficult to decide which physical effects are applicable and must be involved in the model and which effects can be ignored. Therefore, experimental system identification techniques are more appropriate for industrial applications.

The main problems with gain scheduling are that stability and performance are assured only at the local stationary operating point, but the global stability of the parameter-dependent system is not assured for fast deviations of the fluctuating parameter. Therefore, the applicability of the technique relies on a significant number of simulations and real-time experiments [11]. Even so, it is still one of the powerful adaptive control approaches that is widely used in industrial applications.

2.2. Self-tuning adaptive control scheme

As shown in **Figure 3**, self-tuning adaptive controller includes an estimator section that uses plant output and input parameters as input data to estimate the controller gain tuning.

In the self-tuning adaptive control scheme, an analytical relation between the plant parameters and controller gains is evaluated. During the application, the varying system parameters are identified using a parameter identification technique. These identified system parameters are used to calculate adaptive controller gains through the analytical relation evaluated. Several identification methods such as least squares, maximum likelihood and extended Kalman filtering can be used for parameter identification [12, 13]. Furthermore, various analytical procedures, which use system parameters for control design, can be used for self-tuning control, for example, gain-phase margin design and linear quadratic regulator. This approach was originally proposed in Ref. [14] and explained clearly in Ref. [15]. Various applications of the self-tuning control on spacecrafts were demonstrated in Refs. [16, 17].

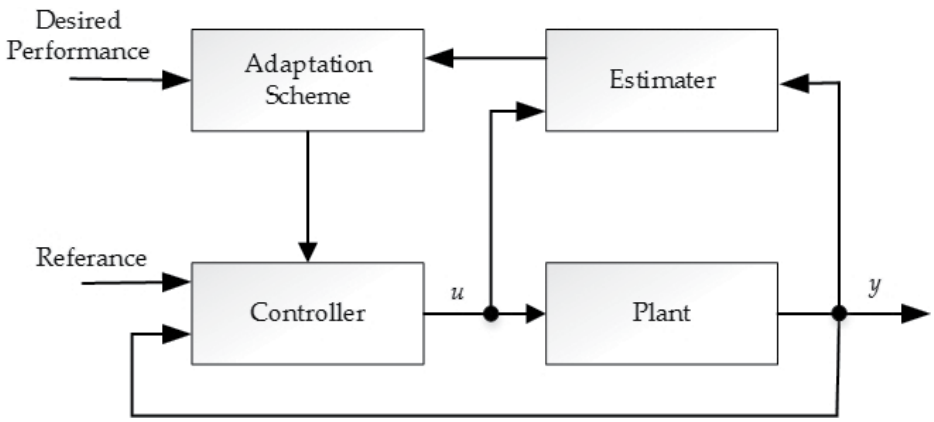

Figure 3. Self-tuning adaptive control system.

Paradoxically, the rather poor performance of the control function leads to unusually striking variations in process variables during control operation. While self-tuning technology has not yet comply perfected, its benefits may be worth the risk. A self-adjusting controller can monitor and adjust the changing gains and time constants of the process to match the adjustment. Even a stationary process may show variable accuracy and time constants if they are not sufficiently linear.

2.3. Model reference adaptive control scheme

MRAC is especially effective when the objective is to track the output with a command signal or trajectory [18, 19]. As shown in **Figure 4**, MRAC scheme uses a second plant model that behaves like an ideal plant to adjust the controller gain. This control scheme can also be used

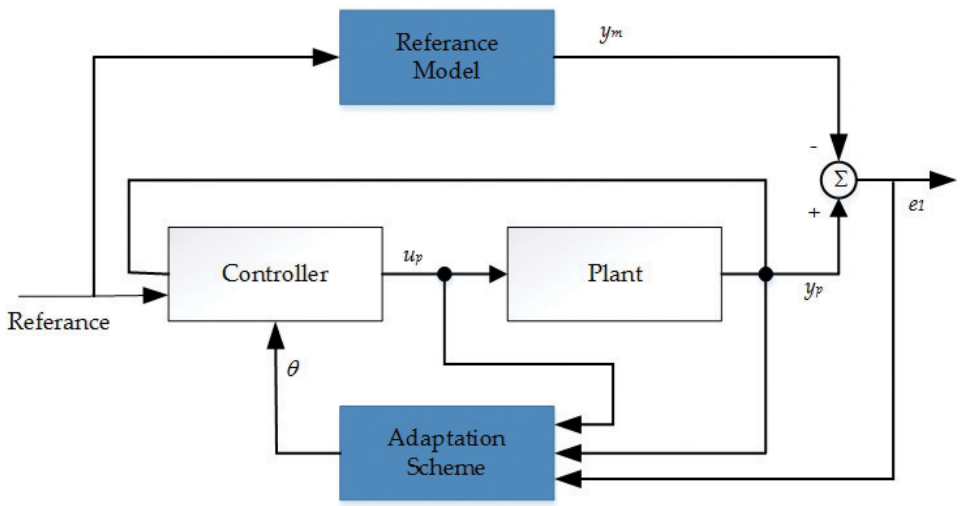

Figure 4. Block diagram of the MRAC.

for controlling unknown LTI plants, both stable and unstable. Two algorithm variations are common in implementing the MRAC scheme. First is the direct model MRAC in which the adaptation is applied using the implicit development of the plant model structure. In the second variation, the model of the plant is developed explicitly. To date the theory behind this technique, and its formal stability proofs have been so matured that there are several important designs using MRAC in the industry with special processors or microcontrollers for their implementation. This technique does not use online identification of the process and the disturbances; the controller also does not exploit the a priori knowledge of a parameter-dependent model, but simply controls the deviations on the plant and output of the reference model. Since the controller is only adjusted when considerable errors occur, this approach is more suited for processes with very slow parameter variations.

Depending on the application, different online parameter estimators can be preferred; this leads to further classification of MRAC. In many mechatronic systems for the defined operating point, system dynamics can be mathematically modeled representing the MRAC general system model.

Objective of the MRAC is to adjust the plant output y_p to the desired level. To describe the plant input u_p, first calculate the reference model output y_m for particular reference input signal $r(t)$. Assuming the system is single input single output (SISO) LTI, its transfer function, $G_p(s)$, is described by Eq. (1).

$$G_p(s) = \frac{y_p}{u_p} = k_p \frac{Z_p(s)}{R_p(s)} \tag{1}$$

k_p is the high frequency gain of plant known sign, $R_p(s)$ and $Z_p(s)$ are polynomials and their degrees are known, and they are n, and m respectively. After polynomial degrees are defined, system relative degree becomes, $n^* = n - m$. Relative degree defines system solution method. For $n^* = 1, 2$ proposed preceding solution methods can be followed for a mechatronic system, which has the same relative degree. But extending the same solution to systems that have $n^* \geq 3$ is still a major obstacle for ongoing researches. According to model reference control scheme, reference model is built similar to plant characteristics. Thus, the reference model transfer function, $W_m(s)$, can be described as follows [1].

$$W_m(s) = \frac{y_m}{r} = k_m \frac{Z_m(s)}{R_m(s)} \tag{2}$$

$Z_m(s)$ and $R_m(s)$ are polynomials same as the plant polynomials, and k_m is high frequency gain of the reference model. Combining control law and known plant parameters, we should choose an adaptive law that allows real-time estimates for the controller parameters. Take in consideration this idea of the control law:

$$u_p = \Theta_1^{*T} \frac{\alpha(s)}{\bigwedge(s)} u_p + \Theta_2^{*T} \frac{\alpha(s)}{\bigwedge(s)} y_p + \Theta_3^* y_p + \Theta_4^* r \tag{3}$$

$\alpha(s) = [s^{n-2}, \ldots\ldots, s, 1]^T$ if $n \geq 2$ otherwise $\alpha(s) = 0$, Θ_1^*, $\Theta_2^* \in \mathcal{R}^{n-1}$, Θ_3^*, $\Theta_4^* \in \mathcal{R}^1$ are constants and they will be defined during system modeling process. $\bigwedge(s)$ is a random Hurwitz polynomial with the degree $n-1$ and $\bigwedge(s) = \bigwedge_0(s)\, Z_m(s)$.

Referring selected control law (3), general plant model can be written in the form

$$y_p = k_p \frac{s + b_0}{s^2 + a_1 s + a_0} u_p \tag{4}$$

Notice that the transfer function (4) has the same form as the plant model. As noted before, reference relative degree should be chosen the same as that of the plant.

$$y_m = W_m(s) r = k_m \frac{s + b_{m0}}{s^2 + a_{m1} s + a_{m0}} r \tag{5}$$

Control law can be rewritten according to the newly defined plant model and reference model.

$$u_p = \Theta_1^* \frac{1}{(s+\lambda)} u_p + \Theta_2^* \frac{1}{(s+\lambda)} y_p + \Theta_3^* y_p + \Theta_4^* r \tag{6}$$

$\alpha(s) = 1$ and $\bigwedge(s) = (s + \lambda)$. The control law (Eq. (7)) can be rewritten in the form

$$u_p = \frac{\Theta_2^* + \Theta_3^*(s+\lambda)}{(s+\lambda) - \Theta_1^*} y_p + \frac{\Theta_4^*(s+\lambda)}{(s+\lambda) - \Theta_1^*} r \tag{7}$$

The closed-loop transfer function is reformulated using new form of the control law (7) and plant model (4), then:

$$\frac{y_p}{r} = G_c(s) = \frac{k_p(s+b_0)\Theta_4^*(s+\lambda)}{(s^2 + a_1 s + a_0)\Big((s+\lambda) - \Theta_1^*\Big) - k_p(s+b_0)(\Theta_2^* + \Theta_3^*(s+\lambda))} \tag{8}$$

By matching $G_c(s) = W_m(s)$ and defining $\theta_4^* = \frac{k_m}{k_p}$ and $\lambda = b_{m0}$ we have

$$(s^2 + a_1 s + a_0)\Big((s+\lambda) - \Theta_1^*\Big) - k_p(s+b_0)\Big(\Theta_2^* + \Theta_3^*(s+\lambda)\Big) = (s+b_0)(s^2 + a_{m1} s + a_{m0}) \cdots \tag{9}$$

By taking the coefficients of s on both sides of Eq. (9), it can be expressed as the matrix algebraic form:

$$\begin{bmatrix} -1 & 0 & -k_p \\ -a_1 & -k_p & -(k_p b_{m0} + k_p b_0) \\ -a_0 & -k_p b_0 & -k_p b_0 b_{m0} \end{bmatrix} \begin{bmatrix} \theta_1^* \\ \theta_2^* \\ \theta_3^* \end{bmatrix} = \begin{bmatrix} a_{m1} + b_0 - b_{m0} - a_1 \\ a_{m0} + b_0 a_{m1} + a_1 b_{m0} - a_0 \\ b_0 a_{m0} + a_0 b_{m0} \end{bmatrix} \tag{10}$$

If we know plant parameters that is in the control law (6), and with parameters $\Theta_4^* = \frac{k_m}{k_p}$ then we can solve the model reference problem using Eq. (10).

In this chapter, Lyapunov's stability theory will be used to calculate adjusting parameters in the MRAC system. First, a differential equation which contains adjusting parameters for error, $e = y_p - y_m$ is derived. Then, Lyapunov function and adaptation mechanism are set according to error.

For a given continuous-time mechatronic system, plant and reference model steady-state space form can be written as follows:

$$\dot{x} = Ax + Bu,\ \dot{x}_m = A_m x_m\ +\ B_m u_c \tag{11}$$

$\boldsymbol{x}$ is state vector of plant and A, and B are constant matrices with suitable dimensions with the system transfer function. Similarly, x_m state vector of plant and A_m and B_m are constant matrices. The control law can be represented in state space form:

$$u(t) =\ Mu_c(t) - Lx(t),\ e(t) =\ x(t) - x_m(t) \tag{12}$$

For given control law, time derivative of error function can be written as

$$\frac{de}{dt} =\ Ax\ +\ Bu_p - A_m x_m -\ B_m u_c \tag{13}$$

$$=\ A_m e\ +\ (A - A_m - BL)x\ +\ (BM - B_m)u_c \tag{14}$$

$$=\ A_m e\ +\ \Psi(x, uc)\cdot(\theta - \theta^0) \tag{15}$$

We can choose following equation as a candidate Lyapunov function.

$$V = \frac{1}{2}\left[e^T Pe + \frac{1}{\gamma}(\theta - \theta^0)^T(\theta - \theta^0)\right] \tag{16}$$

Time derivative of Lyapunov function

$$\dot{V} = \frac{1}{2}e^T[PA_m + A_m^T P]e + (\theta - \theta^0)^T \Psi^T Pe + \frac{1}{\gamma}(\theta - \theta^0)^T \dot{\theta} \tag{17}$$

We can solve Lyapunov function for $P = P^T > 0$

$$PA_m + A_m^T P = -Q,\ Q > 0 \tag{18}$$

And the updated law can be stated as follows.

$$\dot{\theta} = -\gamma \Psi^T Pe = -\gamma \Psi^T(x,\ u_c)\cdot P\cdot(x - x_m) \tag{19}$$

Then, the updated law stated time derivative of Lyapunov function can be rearranged:

$$\dot{V} = -\frac{1}{2}e^T(t)Qe(t),\ e(t) \to \infty \tag{20}$$

3. Case study: design embedded controller for mechatronic sun tracker

Because the gain parameters of the PID controller, which deviate depend on environmental disturbance and working conditions, cannot be tuned during operation, adaptive control methods were offered to eliminate this problem. Photovoltaic (PV) panel only converts solar radiation to electric current when the sun's rays fall perpendicular to the panel surface. Naturally, it is possible only solar noon every day. By adjusting the PV panel to the sun starting from sunrise to sunset, the PV panel output performance may rise up to 46% when compared with fixed slope PV panel [20]. Sun rays' incidence angle changes largely diurnally and slightly seasonally. Sun tracker controller includes two main sections; the first section starts with defining the sun's trajectory. It needs location, time, and date data. Second section is adaptive position control system. Serial tracker architectures consisting of revolute and prismatic joints ensure the rotational movement with two degrees-of-freedom. The tracker is base-fixed on the ground, and an apparatus joins the base to the support of the panels with three legs. This two-part apparatus is designed to revolve the panel-supporting mechanism around a vertical axis. The second part adjusts the slope of the PV panel by prismatic junction. Prismatic junction is controlled by DC motor-driven linear actuators. It is highly precise by design, especially when compared to pneumatic and hydraulic actuators. Screw-based mechanical linear actuators allow to advance or retreat the motive rod by extremely small increments, which is required for the exact positioning of the sun tracker. Electric linear actuator consumes extremely low electric energy available in 12 V DC that can be supplied by the solar panel supported by a battery. Linear actuators can be unusually small, especially when considering the range of motion that is required for moving the sun tracker.

3.1. PV panel attitude model

The robotic manipulator orientation can be defined to different frames, which are reference and sun, and three angles, which are roll, pitch, and yaw. These angles are usually denoted by Ψ, θ, and φ. Sun tracker has two degrees-of-freedom, and to find the sun it has to make two successive rotations. These two rotations can be expressed as compound matrices. It is necessary to get the matrix multiplication of the two preview rotations to obtain the rotation matrix from reference to the sun. Details of the kinematic analysis of sun tracker can be seen in Ref. [21].

The sun tracking algorithm calculates the elevation, α and azimuth, Ψ, angles using latitude, longitude, and date and time data. Then, it uses these parameters to define the slope of the PV panel. α is the solar elevation angle that shows the orientation of the system in the vertical plane and can be defined using Eq. (21):

$$\alpha = \operatorname{asin}(\sin\delta\sin\phi + \cos\delta\cos\phi\cos\omega) \tag{21}$$

where ϕ is the latitude, ω is the hour angle, and δ is the solar declination. It can be defined using Cooper's equation [22].

$$\delta = 23.45\sin\left(\frac{360}{365}(284 + N)\right) \tag{22}$$

N is Julian day and azimuth angle, Ψ, is calculated by using Eq. (23).

$$\Psi = \operatorname{asin}\left(\frac{\cos\delta \sin\omega}{\cos\alpha}\right) \tag{23}$$

3.2. Mathematical model of tracker

As shown in **Figure 5**, the main components of a sun tracker for controller design are two DC motors, sensors to define input reference signal, and position sensors. Mathematical model of the sun tracker mainly deals with actuators for both axes. Both axes have the same type of DC motor except the gear reducer. The major differential equations are given in Eqs. (24) and (25). The Laplace transform of DC motor model equations is given in Eqs. (27)–(30) and the block diagram of this model is shown in **Figure 6**.

$$\frac{di}{dt} = \frac{1}{R}\left(Ri + V_a - K_e\frac{d\theta}{dt}\right) \tag{24}$$

$$\frac{d^2\theta}{d^2t} = \frac{1}{J}\left(K_t i - b\frac{d\theta}{dt}\right) \tag{25}$$

$$v_b(s) = K_r w(s) \tag{26}$$

$$I(s) = \frac{V_a + K_t w(s)}{R + Ls} \tag{27}$$

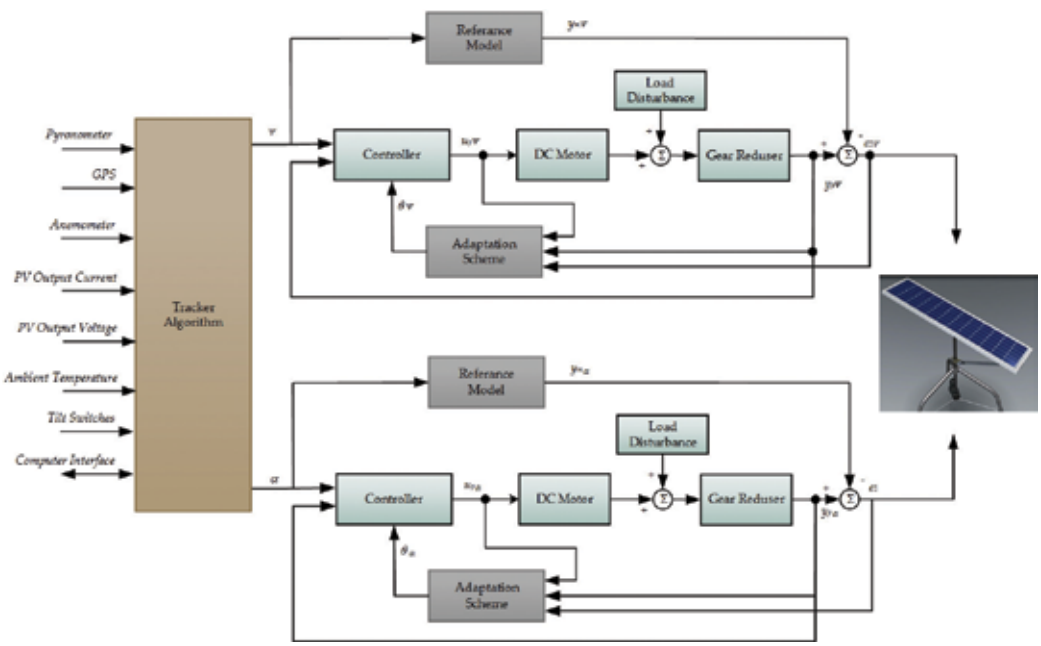

Figure 5. Components of the sun tracker.

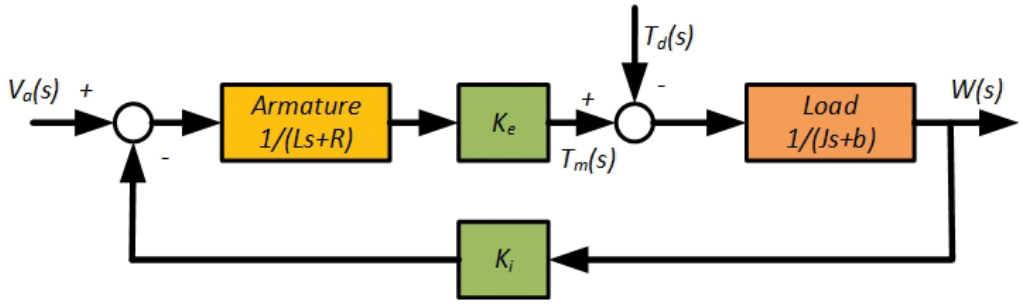

Figure 6. DC motor Simulink model.

$$T_l(s) = Js^2\theta(s) + bs\theta(s) \tag{28}$$

$$T_l(s) = T_m(s) - T_d(s) \tag{29}$$

where I is armature current, V_a is armature voltage, R is the armature resistance, L is armature inductance, V_b is back-emf, K_e is motor coefficient, K_t is back-emf coefficient, J is rotor inertia, b is viscous damping ratio, T_m is motor torque, T_l is load torque, and T_d is the disturbance torque. According to **Figure 7**, the transfer function of DC motor can be represented in Eq. (30).

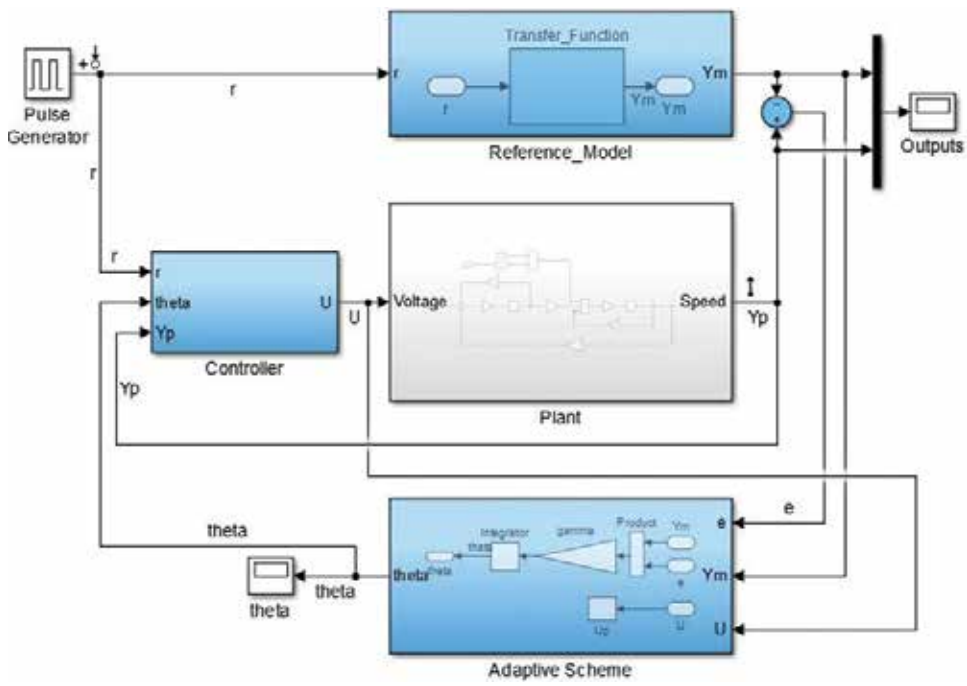

Figure 7. Simulink simulation model of MRAC.

$$\frac{w(s)}{V_a(s)} = \frac{K_e}{(R + Ls)(Js + bs) + K_e K_t} \tag{30}$$

The sun tracker Simulink can be built using DC motor mathematical model, Eqs. (24) and (30), and adaptive model reference controller scheme, Eqs. (6), (9), (24), and (30). As shown in **Figure 8**, Simulink model consists of three main parts: reference model, plant, and adaptive controller. Plant disturbance, T_d, can be changed manually and plant output y_p adapts this variation quickly. Simulation results are shown in **Figure 8** for motor speed and motor voltage.

The hardware design assembles the embedded microcontroller to the solar rotation mechanism consisting of two rotary DC motors driven by PMDC drives, linear actuators controlled by these DC motors, a pyranometer for solar radiation measurements, an anemometer for obtaining wind speed data, a global positioning system (GPS), tilt switches and potentiometers for position sensing. The GPS data in sentences that contain a string of characters is transferred to the microcontroller continuously through serial RS-232 port. The content of these sentences contains the longitude, latitude, altitude, date, and time for the specific location. As a result, the GPS clock signal is used to update the microcontroller's internal time periodically, and thus effects of the long-term errors are eliminated. As part of the effort to improve solar tracker reliability and performance, a pyranometer has been added to the solar tracker. This pyranometer, which is placed on the tracker, allows the data acquisition system to measure precisely the irradiance faced by the PV modules. Besides, monitoring developed algorithm uses pyranometer during optimization process to predict tilt and azimuth angle paths. In solar tracker systems, azimuth angle is usually measured with a potentiometer.

The solar tracker' position is bounded by mechanical limit switches to ensure robust operation. A micro-roller switch mounted on the base of the solar tracker prevents multiple revolution windup of the azimuth tracking stage. Two more limit switches are used for the zenith

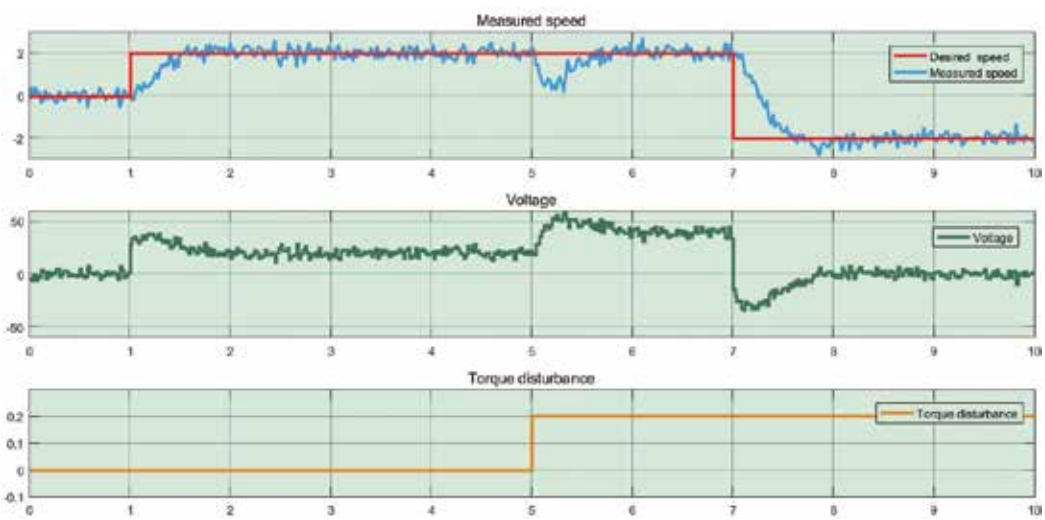

Figure 8. Simulink simulation results for a given input reference signal.

tracking stage to prevent over-travel damage to the linear actuator mechanism. Initial reset balancing is provided by four tilt switches (east, west, south and north). An anemometer measures the local wind speed to protect tracker components from over-wind speed. For more data on hardware design, see Ref. [23].

3.3. Software design

The developed sun tracking algorithm precisely determines sun angles and times for sunrise, solar noon, and sunset all year long. The flowchart of the algorithm is given in **Figure 9**. This

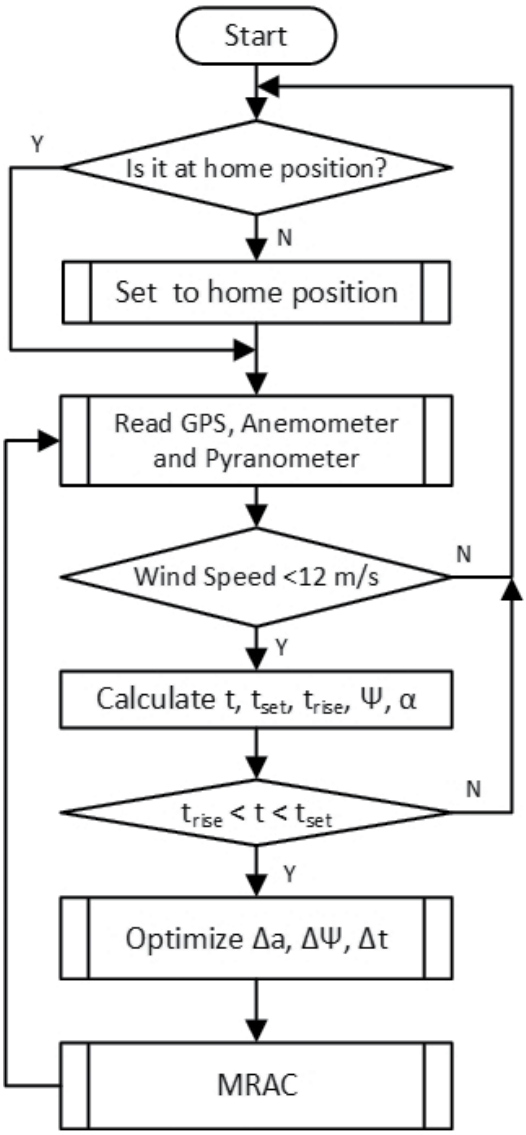

Figure 9. Adaptive controller sun tracker algorithm.

algorithm software calculates the sun angles just by using the date, time and precise longitude, latitude, and elevation of the location through a GPS system.

The microcontroller first sets the tracker to home position and then fetches GPS data to calculate the sunrise and sunset times. For more information, see Ref. [21]. The present solar time is compared with the calculated sunrise and sunset times to make a decision whether the sun tracking procedure will be performed. If the current solar time is between sunrise and sunset times, the microcontroller reads wind speed value from the anemometer to define whether the sun tracker can move safely. At night-time, it waits for the next interval. The sun tracker works in tracking mode if the wind speed is under 12 m/s, else it stands in table position to protect the tracking system. During the tracking phase, the microcontroller reads the pyranometer value to check if there is enough solar radiation to generate power. Otherwise, the sun tracker stays at home position until solar radiation rises at least to its lower limit. When the solar radiation reaches the desired value, the algorithm calls optimization subroutines that calculate the optimum $\Delta\alpha$ and $\Delta\Psi$ angles. Finally, the microcontroller calls the model reference adaptive control subroutine.

MRAC subroutine reads the azimuth, $\Delta\Psi$, and tilt, $\Delta\alpha$, angles, and converts them to a PWM signal to drive the tilt and azimuth motors. The calculated angles α(t) and (t) are then subtracted from the former position values. In accordance with the angle difference and sign, the microcontroller sends PWM and direction signals to the motor drives. Azimuth and tilt motors move the solar panel to a new position to precisely track the sun's trajectory. The two motors make different movements to approach the desired position of the solar panel by comparing the measured angles with the calculated ones. When the sun sets, the system goes back to home position, ready for the next sunrise.

4. Results and discussions

Embedded controller design is inherently a multidisciplinary issue. It involves dynamic systems analysis, modeling, estimation, simulation, control, signal processing, microcontroller architecture programming, power electronics, and motor drives. Embedded adaptive controller design began with the creation of the mathematical model of the sun tracker. The algorithm of the model reference adaptive controller was created in the next stage. In the last stage the program written in the light of this information was embedded in the microcontroller. Embedded controller for sun tracker implied by using Arm cortex M4 core microcontroller TM4C129 produced by TI [24]. Embedded code for adaptive algorithm was produced using Simulink model. The other part of embedded code was written in embedded C using Keil IDE [25], then the two codes were combined. After compiled debug and code optimization was made, it was embedded to the microcontroller.

The designed sun tracker was tested first in laboratory, then it was tested outdoor. System parameters, such as PV panel output current, voltage, solar irradiance, and temperature, were measured and logged during May 12 to November 23, 2016. Analysis of these data revealed that the sun tracker generated 39.2% more energy than the fixed slope system. When compared, the

generated energy efficiency is slightly less than [20, 21, 23]. The total generated energy is high than [23] also consumed energy by DC motors higher. The PID controller, which was designed in Ref. [23], was optimized according to minimum energy consumption. It follows the sun's trajectory when it quarantines generated energy at least bigger than the consumed energy. As a result, this criterion increases the yield of generated energy when sun tracking accuracy is lower. Inversely, designed model reference adaptive controller tracks the sun very precisely; this increases both the energy generated and consumed. Meanwhile, MRAC continues tracking the sun in changing environmental conditions such as higher wind speed and higher ambient temperature. It inherently adapts to changing conditions easily, but PID controller usually loses track of the sun under same conditions. Sample day performances of tracker and fixed ones are given in **Figure 10** for sunny and cloudy days. For ideal tracker, the power curve should be like rectangular, but early morning and close to sunset, the sun's trajectory changes rapidly, so the consumed energy increases. The same phenomenon takes place on partly cloudy days; adaptive controller tracks quickly changing sun condition and both consumed and generated energies increase.

The proposed framework for dual-axis sun tracker could be realized in hardware easily and follow-up the hardware the control algorithm can be designed. It will be quite an efficient and cost-effective solution and in the long term it will provide an efficient solution for the users that need high energy from limited area with accurate alignment with sun, such as space stations, sea vehicles, and automatically aligned large telescopes.

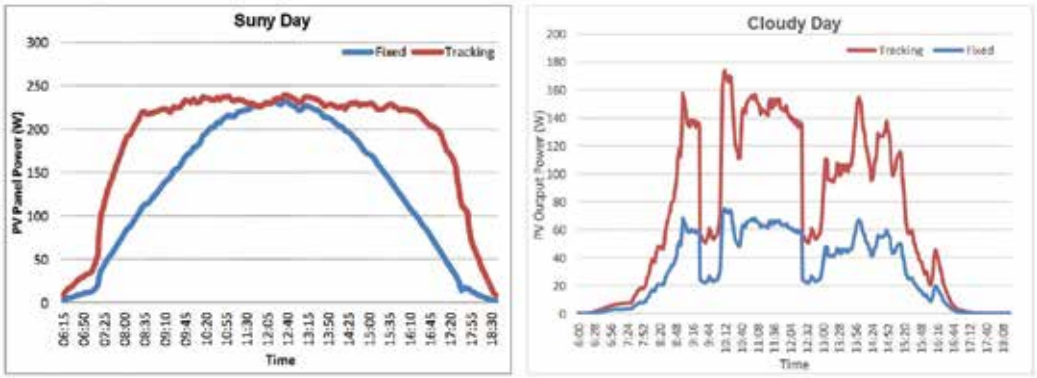

Figure 10. Outdoor test results of sun tracker for sunny and cloudy day.

5. Conclusion

Designed dual-axis sun trackers with embedded adaptive controller and fixed PV system were tested for 6 months. The system performance parameters for both systems were measured and stored using a data logger. The collected data is examined and compared. On a clear sky day, performance data proved that the embedded adaptive controller sun tracker produced 39.2% more energy than the fixed one. The efficiency of the sun tracker decreases to 34% on partly

cloudy days. Furthermore, embedded adaptive controller is compared with embedded PID controller in terms of energy production, power consumption, reaction time, and code size.

Author details

Mustafa Engin

Address all correspondence to: mustafa.engin8@gmail.com

Electronic Technology Department, Ege Higher Vocational School, Ege University, Bornova, İzmir, Turkey

References

[1] M. Stefanovic, M.G. Safonov. Safe adaptive control: data-driven stability analysis and robust synthesis (vol. 405). Springer; 2011.

[2] J.S. Shamma, M. Athans. Analysis of gain scheduled control for nonlinear plants. IEEE Transactions on Automatic Control, vol. 35, pp. 898–907, 1990.

[3] N. Aouf, D.G. Bates, I. Postlethwaite, B. Boulet. Scheduling schemes for an integrated flight and propulsion control system. Control Engineering Practice, vol. 10, pp. 685–696, 2002.

[4] P. Apkarian, P. Gahinet, G. Becker. Self-scheduled H8 control of linear parameter-varying systems: a design example. Automatica, vol. 31, pp. 1251–1261, 1995.

[5] G. Becker, A. Packard. Robust performance of linear parametrically varying systems using parametrically-dependent linear feedback. Systems & Control Letters, vol. 23, pp. 205–215, 1994.

[6] M. Pakmehr, N. Fitzgerald, E.M. Feron, J.S. Shamma, A. Behbahani. Gain scheduling control of gas turbine engines: stability by computing a single quadratic Lyapunov function. In ASME Turbo Expo 2013: Turbine Technical Conference and Exposition, San Antonio, Texas, USA, 2013, p. V004T06A027.

[7] R. Márquez, T.M. Guerra, M. Bernal, A. Kruszewski. Asymptotically necessary and sufficient conditions for Takagi–Sugeno models using generalized non-quadratic parameter-dependent controller design. Fuzzy Sets and Systems, vol. 306, pp. 48–62, 2017.

[8] T. Bakka, H.-R. Karimi, S. Christiansen. Linear parameter-varying modelling and control of an offshore wind turbine with constrained information. IET Control Theory & Applications, vol. 8, pp. 22–29, 2014.

[9] C.S. Liu, Q. Ye, S.J. Zhang. Adaptive control for a class of uncertain linear parameter-varying flight aircraft systems. Int. J. Adapt. Control Signal Process. 2017, vol. 31, pp. 210–222.

[10] L. Fernandes, F. Moraes Filho, J. Paulo, J. Oliveira. Gain scheduling adaptive control applied to a particular mixer-settler equipment. Control Engineering Practice, vol. 21, pp. 1121–1127, 2013.

[11] C. Hoffmann, H. Werner. A survey of linear parameter-varying control applications validated by experiments or high-fidelity simulations. IEEE Transactions on Control Systems Technology, vol. 23, pp. 416–433, 2015.

[12] J.L. Crassidis, J.L. Junkins. Optimal estimation of dynamic systems. CRC Press, Boca Raton, FL, USA, 2011.

[13] N. Hovakimyan, C. Cao. L1 adaptive control theory: guaranteed robustness with fast adaptation. Philadelphia, PA: Society for Industrial and Applied Mathematics; 2010.

[14] K.S. Narendra, A.M. Annaswamy. Stable adaptive systems. Courier Corporation; Mineola, New York, USA, 2012.

[15] K.J. Åström, B. Wittenmark. On self tuning regulators. Automatica, vol. 9, pp. 185–199, 1973.

[16] R. Chiang, W. Breckenridge, E. Wong. Self-tuning thruster control for Cassini spacecraft. In Guidance, Navigation, and Control Conference, San Diego, CA, USA, 1996, p. 3822.

[17] L. Shieh, J. Sunkel, Z. Yuan, X. Zhao. Self-tuning control of attitude and momentum management for the space station. Journal of Guidance, Control, and Dynamics, vol. 15, pp. 17–27, 1992.

[18] B. Shackcloth, R. Butchart. Synthesis of model reference adaptive systems by Liapunov's second method. In Theory of self-adaptive control systems. Springer, New York, USA, 1966, pp. 145–152.

[19] A.J. Calise, M. Sharma, J.E. Corban. Adaptive autopilot design for guided munitions. Journal of Guidance, Control, and Dynamics, vol. 23, pp. 837–843, 2000.

[20] M. Engin, D. Engin. Optimization controller for mechatronic sun tracking system to improve performance. Advances in Mechanical Engineering, vol. 5, p. 146352, 2013.

[21] M. Engin, D. Engin. Optimization mechatronic sun tracking system controller's for improving performance. In IEEE International Conference on Mechatronics and Automation (ICMA), Takamatsu, Kagawa, Japan, 2013, pp. 1108–1112.

[22] T. Shivalingaswamy, B. Kagali. Determination of the declination of the sun on a given day. European Journal of Physics Education, vol. 3, 2012, pp. 17–22.

[23] M. Engin, D. Engin. Design of real time embedded PID controller for sun tracking robot manipulator. In IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), 2014, pp. 670–675.

[24] TI. (2015, January 12). Tiva™ C Series LaunchPad Evaluation Kit. Retrieved from http://www.ti.com/tool/ek-tm4c123gxl, 22 November 2016.

[25] KEIL. (2014, March 25). Embedded development tools (UVision 5.01 ed.).

Chapter 2

Embedded Control of the DC Drive System for Education

Marian Gaiceanu

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/67461

Abstract

Taking into account the spread use of the DC motor in robotics, the main objective of the chapter is to promote the knowledge regarding the development of the DC drive system, in didactical manner. Basically, for one motor control the cascaded loop DC drive system is used. Numerical implementation of the DC drive system based on Z-transform discrete method is provided. The electrical drive supposes both control types: the torque and speed. The feedback signals of the control loops are provided by using incremental encoder and of an original load torque estimator. Based on the DC motor dynamics, the appropriate parameters of the speed and torque controllers are provided automatically. On line parameters estimation of the DC motor can assure the performances of the DC drive controllers. The symmetrical and modulus optimum criteria are used. In order to prove the feasibility of the proposed solution, the entire Matlab script of the cascade control of the DC drive is provided. The strategy of the cascaded loop control at constant flux is divided into dynamic regime and into steady state regime. The chapter strategy follows two directions: the first, knowledge developments, the second, the applicative research through the real time implementation.

Keywords: Arduino, Matlab®, Simulink, H-bridge, DC motor, speed control, torque control, load torque estimator, discrete control system synthesis, Z-transform

1. Introduction

The chapter is addressed to researchers, students, and the users of industrial power converters and electric drives or mechatronics. The chapter contains the mandatory theoretical aspects and methods in order to develop successful DC drive system, being the most used in mechatronics applications. Upon implementing the proposed chapter applications, the readers will have the capability to develop own electric drive system, to bring the idea in practice very fast, and for better understanding of using the drive systems being open to add own contributions in

the mechatronics field. The chapter contains the basic concepts and techniques to design, simulate, and implement an electric drive system through a well-structured technical guide. By completing this chapter, the researchers will be familiarized with applying the concepts and techniques to manage adequately an electric drive system, widely found in industry and in mechatronics projects. The chapter strategy follows two directions: first, the theoretical approach (a practical discretization method is provided), and the second, the applicative research (numerical simulations and real-time implementations are provided). The chapter contains a high level of the cross-disciplinarily: electric machines, power electronics converters, Matlab/Simulink, real-time implementation, electrotechnics, measurements, and advanced control techniques.

In Section 2, an overview of the adjustable DC drives is given with the purpose to point out the development trend in the technology. In the next section, the dynamical equations of the DC drive system are provided. Section 4 contains the design of the rotor current and speed controllers. The load torque estimator is described in Section 5. Section 6 comprises the energetically model of the DC drive. Based on the on-line parameters identification procedure [1] the energy balance of the DC drive is performed and the appropriate energetic model is deducted in the Section 6 [2]. The optimal control based on linear quadratic controller is designed in the Section 7. The numerical approach by using Z-transform is provided in Section 8, and the simulation results are provided in Section 9. The real-time implementation of the speed control is included in Section 10. The conclusions of the chapter are placed in Section 11. The entire Matlab script is provided.

2. Overview of the DC drives

The DC machine can be controlled by using power converters. At low speed, the DC drives are very often used. There are two mainly control techniques: armature voltage control (constant torque region) and field current control (constant power region). The armature voltage control is obtained by maintaining constant field current and the field current control by maintaining the constant armature voltage [3].

3. The mathematical model of the DC drive system

Under certain hypotheses, the mathematical model of the DC machine could be obtained in state space form.

Supposing a constant value of the DC motor parameters, the DC machine has the compensated winding, the magnetic flux maintained at the constant value, the state space mathematical model of the DC machine becomes as follows [2]:

$$\begin{bmatrix} \frac{\mathrm{d}n(t)}{\mathrm{d}t} \\ \frac{\mathrm{d}i_A(t)}{\mathrm{d}t} \end{bmatrix} = \begin{bmatrix} -\frac{F_v}{J} & \frac{C_m}{cJ} \\ -\frac{C_e}{T_A R_A} & -\frac{1}{T_A} \end{bmatrix} \cdot \begin{bmatrix} n(t) \\ i_A(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{T_A R_A} \end{bmatrix} u(t) + \begin{bmatrix} -\frac{1}{cJ} \\ 0 \end{bmatrix} T_L(t) \tag{1}$$

in which J is the reduced inertia moment to the electric motor shaft, T_A, the armature time constant, R_A, the armature resistance, and the specific constants

$$c_t = \frac{\pi}{30}, \qquad C_m = k\phi, \qquad C_e = c_t C_m, \tag{2}$$

in which k is the DC motor constant.

The state vector of the DC machine contains two states, $\mathbf{x}(t) = \begin{bmatrix} n(t) \\ i_A(t) \end{bmatrix}$, the control vector is only the rotor voltage $\mathbf{u}(t) = \mathbf{u}_A(t)$, the perturbation vector is the load torque $\mathbf{w}(t) = T_L(t)$, and the output of the DC motor is the speed $\mathbf{y}(t) = n(t)$. Under these circumstances, the mathematical model of the DC motor becomes

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A} \cdot \mathbf{x}(t) + \mathbf{B} \cdot \mathbf{u}(t) + \mathbf{G} \cdot \mathbf{w}(t) \\ \mathbf{y}(t) &= \mathbf{C} \cdot \mathbf{x}(t), \end{aligned} . \tag{3}$$

Taking into consideration the above mentioned hypothesis, the mathematical model of the DC motor Eq. (1) is linear and invariant.

In order to obtain variable speed, there are two topologies of the power converters for DC motor applications: ac-dc and dc-dc. The mathematical model of the power converter, under certain conditions (continuous conduction, neglecting the voltage drop in conduction mode), represents an ideal power amplifier k_D [2].

4. The cascaded loop control of the DC drive system

The DC drive contains two control loops: the inner—the armature current loop and the outer—the speed loop, **Figure 1** [2]. Therefore, the control is in cascaded manner. The current feedback is delivered through an adequate current sensor and the speed feedback is obtained based on numerical derivative of the encoder signals. The desired speed is set as reference signal from a potentiometer. Due to the higher power necessity, the dc-dc four quadrants power converter is added. Therefore, by means of the DC voltage control of the DC machine, the torque and speed are adequately adjusted according to the references. The parameters of the current regulator are obtained by using the *modulus criterion*, and the parameters of the speed controller are delivered by using the *symmetrical optimum criterion* (*Kessler* version).

By taken into account the 10 (V) voltage unified system, the transfer function of the proportional-integral (PI) current controller is expressed as

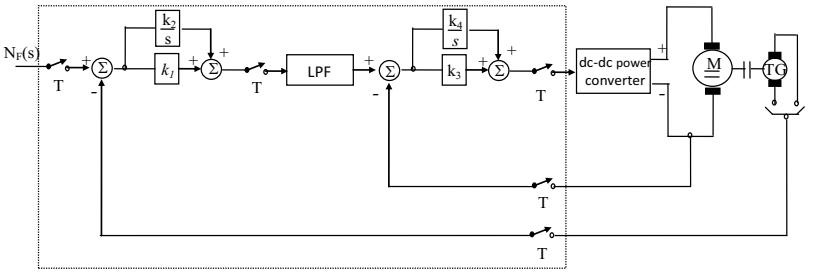

Figure 1. Block diagram of the digital DC speed cascaded control.

$$H_{RI}(s) = k_3 + \frac{k_4}{s} \tag{4}$$

with the following parameters [2]:

$$k_3 = \frac{T_A R^A{}_A}{2T_{\Sigma I} k_D k_I},\ k_4 = \frac{R_A}{2T_{\Sigma I} k_D k_I} \tag{5}$$

where:

- $k_D = \frac{U_N}{10}$, transfer function of the dc-dc power converter;
- $k_I = \frac{10}{I_{A\max}} \left[\frac{V}{A}\right]$, attenuation factor of the current transducer;
- $T_{\Sigma I} = T_I$ the sum of the parasitic time constant of the current loop is determined by the time constant of the current transducer.

The transfer function of the PI speed controller has the form:

$$H_{RT}(s) = k_1 + \frac{k_2}{s} \tag{6}$$

the parameters of the speed controller [2] being deducted from symmetrical optimum criterion:

$$k_1 = \frac{k_I c J}{2T_{\Sigma N} C_m k_T} \tag{7}$$

$$k_2 = \frac{k_I c J}{8T_{\Sigma N}^2 C_m k_T} \tag{8}$$

in which:

- $k_T = \frac{10}{n_N} \left[\frac{V}{rpm}\right]$, attenuation factor of the speed transducer;
- $T_{\Sigma N} = 2 \cdot T_{\Sigma I} + T_N$, the sum of the parasitic time constants of the speed loop, T_N being the time constant of the speed transducer.

5. Load torque estimator

In **Figure 2**, the operational model of the load torque estimator [4, 5] is shown. The calculus of the load torque estimator parameters is based on the imposed performances: desired pulsation ω_0^*, and the desired overshoot ξ^*, respectively:

$$\tau = \frac{2\xi^*}{\omega_0^*} \tag{9}$$

$$c = 2\xi^*\omega_0^*. \tag{10}$$

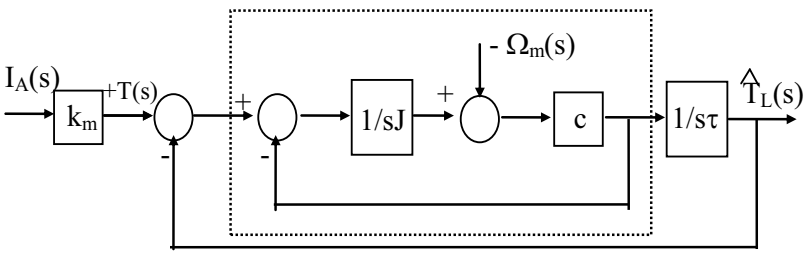

Figure 2. The second-order load torque estimator [2].

6. The energetically model of the DC drive

Based on the mathematical model of the DC machine Eq. (1), the energetic model has been deducted (**Figure 3**) [2], in which T_o is the mechanical friction torque, T_{Fv}—the viscous force equivalent torque, T_f—the Foucault currents losses equivalent torque, and T_H—the hysteresis losses equivalent torque.

By using the electrical and mechanical equations of the DC motor at the constant flux, the energetic components of the DC drive are obtained:

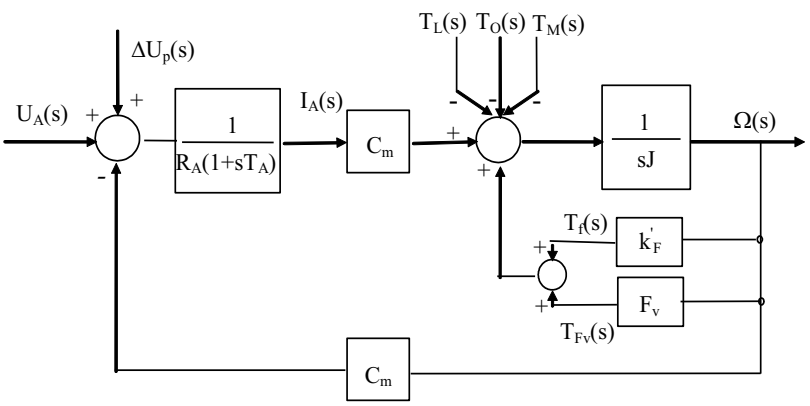

Figure 3. The energetic model of the DC machine.

$$u_A(t) = R_A \cdot i_A(t) + L_A \cdot \frac{\mathrm{d}i_A(t)}{\mathrm{d}t} + e(t) + \Delta u_p(t) \tag{11}$$

$$T_e(t) = T_L(t) + F_v \cdot \Omega(t) + J \cdot \frac{\mathrm{d}\Omega(t)}{\mathrm{d}t}. \tag{12}$$

By multiplying Eq. (11) with $i_A(t)$ term, and Eq. (12) with $\Omega(t)$, the power balance of the DC drive can be obtained. In this way, the specific power components are obtained:

- The electrical power

$$P_1(t) = u_A(t) \cdot i_A(t) \tag{13}$$

- Different expressions of the electromagnetic power

$$P(t) = T_e(t) \cdot \Omega(t) = C_m i_A(t)\Omega(t) = e(t) i_A(t) \tag{14}$$

- The rotor copper power losses

$$P_{cu} = R_A \cdot i_A^2(t) \tag{15}$$

- The accumulated power in the rotor inductance

$$P_L = L_A i_A(t) \frac{\mathrm{d}i_A(t)}{\mathrm{d}t} \tag{16}$$

- The brush power losses

$$P_p(t) = \Delta u_p(t) \cdot i_A(t) \tag{17}$$

- The mechanical power

$$P_u(t) = T_L(t) \cdot \Omega(t) \tag{18}$$

- The accumulated power in the rotational masses

$$P_j = J\Omega(t) \frac{\mathrm{d}\Omega(t)}{\mathrm{d}t} \tag{19}$$

- The viscous power losses

$$P_f(t) = T_f \cdot \Omega(t) \tag{20}$$

The magnetization frequency of the rotor ferromagnetic core is

$$f = \frac{p\Omega(t)}{2\pi}. \tag{21}$$

At the constant flux, the total core losses are

$$p_{\text{Fe}} = k_H \cdot f + k_F \cdot f^2 \tag{22}$$

or by

$$p_{Fe}(t) = k_H^{'}\Omega(t) + k_F^{'}\Omega^2(t). \tag{23}$$

Taking into consideration the deducted magnetic core losses Eq. (23), the corresponding equivalent torque is as follows

$$T_{Fe}(t) = \frac{P_{Fe}}{\Omega(t)} = k_H' + k_F' \cdot \Omega(t) \tag{24}$$

or denoted by:

$$T_{Fe}(t) = T_H(t) + T_F(t). \tag{25}$$

The mechanical losses are constant. Therefore, the corresponding torque is as follows:

$$T_0(t) = \frac{P_0}{\Omega(t)}. \tag{26}$$

By applying the Laplace transform to Eqs. (11) and (12), the electromechanical model of the DC motor at constant flux is obtained:

$$U_A(s) - E(s) - \Delta U_p(s) = R_A I_A(s) + sL_A I_A(s) \tag{27}$$

$$T(s) = T_L(s) + T_0(s) + T_H(s) + T_F(s) + T_f + sJ\Omega(s) \tag{28}$$

Based on Eqs. (27) and (28), the energetic operational model of the DC motor is obtained as in **Figure 3** [2].

6.1. Estimation of the DC drive losses components

By using zero-order hold, the numerical energies are calculated as a sum of the local areas: $\sum_{i=0}^{N} V_i T$ (N the maximum sampling interval with constant sampling time T).

The losses in the rotor windings at the k-th sampling time are as follows:

$$\Delta P_{\text{Cu}}(kT) = R_A \cdot I_A^2(kT), \tag{29}$$

and the total rotor copper energy over the control interval can be deducted as

$$\Delta W_{\mathrm{Cu}} = \sum_{i=1}^{N} R_A \cdot I_A^2(iT) \cdot T. \tag{30}$$

In the same way, the following energies are calculated:

- The input energy of the DC motor

$$W_{P_1} = \sum_{i=1}^{N} U(iT) \cdot I_A(iT) \cdot T \tag{31}$$

- The accumulated energy in the rotor inductance

$$\Delta W_L = \sum_{i=1}^{N} L_A \cdot I_A(iT) \cdot \left[I_A(iT) - I_A\Big((i-1)T\Big) \right] \tag{32}$$

- The accumulated energy in the rotational mass

$$\Delta W_J = \sum_{i=1}^{N} J \cdot \Omega(iT) \cdot \left[\Omega(iT) - \Omega\Big((i-1)T\Big) \right] \tag{33}$$

The brush, the mechanical, and the core losses have been ignored.

By knowing the above mentioned energies, the output energy could be calculated:

$$W_{\mathrm{out}} = W_{P_1} - \Delta W_{\mathrm{Cu}} - \Delta W_L - \Delta W_J \tag{34}$$

By using the same methodology, the losses of the power converter and the transformer are obtained.

7. Linear quadratic control

The DC machine operates at maximum efficiency at rated operating point, therefore, in steady state. During transient regime, the copper losses component is the higher loss in the DC machine. In order to reduce the power losses in DC drive, the optimal control block diagram is proposed. The proposed solution combines the advantages of the conventional cascaded control (PI) and optimal control [6, 7]. The designed software switch goes in position 1 when the speed error is zero, otherwise goes in position 2 (**Figure 4**).

7.1. Problem formulation

The problem statement of the optimal control supposes a design of the performance index (functional cost). By minimizing the functional cost, the optimal control solution is obtained without any constraints. The optimal control limits are controlled by introducing the weighting matrices [6, 7]. The objectives of the optimal control include the objectives of the conventional drive system (regulation, stability assurance, and robustness to perturbations) and the energy reduction.

http://dx.doi.org/10.5772/67461

The gate drive control is denoted by GDC, the voltage control by u_c (**Figure 4**), the load torque estimator by TL estimator block, the tacho generator by TG, the imposed speed reference and the imposed firing angles. The switch is toggled between position 1 (for conventional control) and fixed position 2 (for optimal control). The conventional control is activated during the steady-state regimes, and the optimal control is activated during transients (starting, stopping, and reversing regimes).

In order to control the DC motor in an optimal manner (**Figure 5**), the adopting functional cost is with infinite time horizon:

$$J = \frac{1}{2}\int_0^\infty [\mathbf{x}^\mathbf{T}(\mathbf{t})\mathbf{Q}\mathbf{x}(\mathbf{t}) + \mathbf{u}^\mathbf{T}(\mathbf{t})\mathbf{R}\mathbf{u}(\mathbf{t})]dt. \tag{35}$$

By minimizing the adopted performance function, the optimal control is delivered:

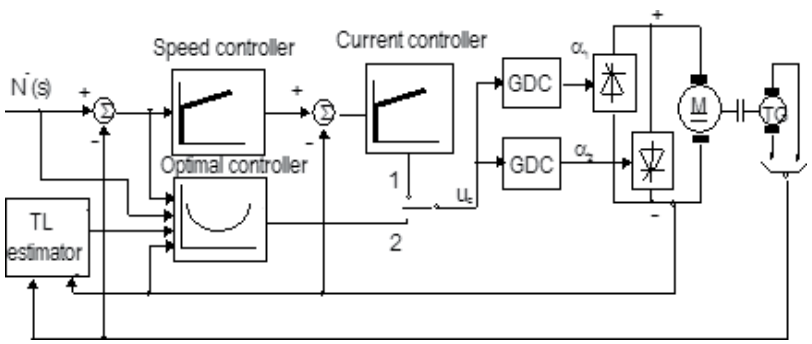

Figure 4. Optimal control system of the DC motor at constant flux.

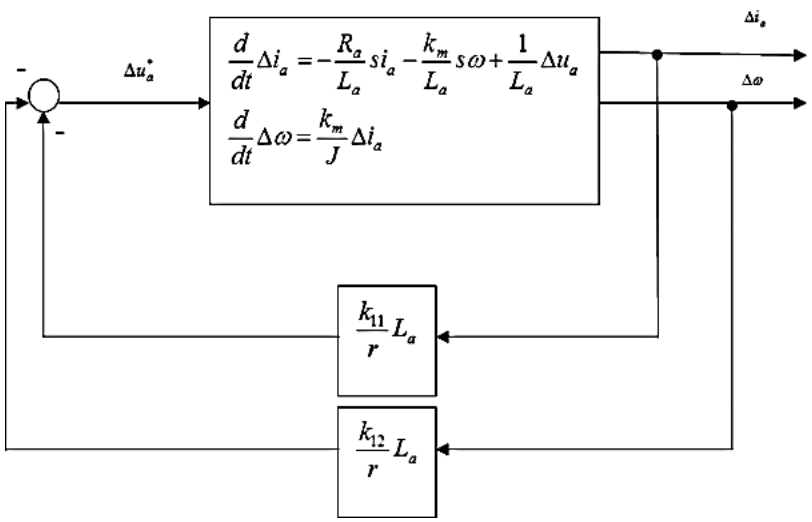

Figure 5. The optimal DC drive system.

$$u^*(t) = -\mathbf{R}^{-1}\mathbf{B}^{\mathbf{T}}\mathbf{K}\mathbf{x}(\mathbf{t}), \tag{36}$$

in which the symmetrical gain matrix **K**:

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} > 0, \tag{37}$$

is the solution of the algebraic Riccati equation (ARE):

$$\mathbf{KA} + \mathbf{A}^{\mathbf{T}}\mathbf{K} - \mathbf{KBR}^{-1}\mathbf{B}^{\mathbf{T}}\mathbf{K} + \mathbf{Q} = \mathbf{0}. \tag{38}$$

8. Numerical implementation of the DC drive system

The continuous signals are converted into digital format by using sampling and hold stages. The reversibility takes place through the digital-to-analog converters (DACs), that is, the digital signals are converted into continuous ones.

The properties of the zero-order hold are to maintain the captured signal at the constant value over the entire sampling period. Therefore, the obtained transfer function is

$$H_{zoh}(s) = \frac{1 - e^{-sT}}{s}. \tag{39}$$

8.1. Mathematical model of the digital current control loop

In **Figure 6**, the block diagram of the numerical model of the current loop is shown [2].

The switch symbol from **Figure 6** represents the sampling and hold process, T being the sampling time.

The transfer between the "s" plane and "Z" one is made by using the Z-transform. By using the transfer function of the zero-order hold, the discrete transfer function could be obtained according to

$$H_{\mathrm{FI}}(z^{-1}) = Z[H_{\mathrm{EOZ}}(s) \cdot H_{\mathrm{FI}}(s)] \tag{40}$$

or in the other form

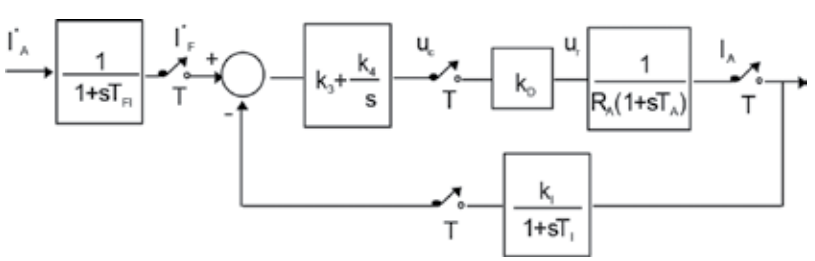

Figure 6. The armature current control loop.

$$H_{\mathrm{FI}}(z^{-1}) = Z\left[\frac{1-e^{-sT}}{s}\cdot\frac{1}{1+sT\mathrm{FI}}\right]. \tag{41}$$

Taking into account that $z = \mathrm{e}^{sT}$, the discrete transfer function Eq. (41) becomes

$$H_{FI}(z^{-1}) = \frac{1-z^{-1}}{T_{FI}}\cdot Z\left[\frac{1}{s\left(s+\frac{1}{T_{FI}}\right)}\right]. \tag{42}$$

Taking into account the residuum theorem of the simple poles, the Z-transform can be calculated as

$$H_{FI}(z^{-1}) = \frac{1-z^{-1}}{T_{FI}}\cdot\left[\frac{1}{\frac{1}{T_{FI}}}\cdot\frac{1}{1-z^{-1}\cdot\mathrm{e}^{T\cdot 0}}+\frac{1}{-\frac{1}{T_{FI}}}\cdot\frac{1}{1-z^{-1}\cdot\mathrm{e}^{-\frac{T}{T_{FI}}}}\right]. \tag{43}$$

By the other side:

$$H_{\mathrm{FI}}(z^{-1}) = \frac{I^*_{\mathrm{FI}}(z^{-1})}{I^*_A(z^{-1})}. \tag{44}$$

By taking into consideration Eqs. (43) and (44), the following recurrent relation is obtained

$$I^*_{\mathrm{FI}}(k) = I^*_{\mathrm{FI}}(k-1)\cdot e^{-\frac{T}{T_I}} + I^*_A(k-1)\cdot\left(1-e^{-\frac{T}{T_I}}\right). \tag{45}$$

By starting from the current transducer transfer function

$$\frac{I_R(s)}{I_A(s)} = \frac{k_I}{1+sT_I} \tag{46}$$

a similar recurrent equation for the current feedback is obtained:

$$I_R(k) = I_R(k-1)\cdot\mathrm{e}^{-\frac{T}{T_I}} + I_A(k-1)\cdot\left(1-\mathrm{e}^{-\frac{T}{T_I}}\right)\cdot k_I.^I \tag{47}$$

The discrete implementation of the current controller

$$H_{\mathrm{RI}}(s) = k_2 + \frac{k_4}{s}, \tag{48}$$

is made by using in the similar manner as for the current filter and using the linearity propriety of the Z-transform

$$H_{\mathrm{RI}}(z^{-1}) = k_3(1-z^{-1})Z\left[\frac{1}{s}\right] + k_4(1-z^{-1})Z\left[\frac{1}{s^2}\right]. \tag{49}$$

On the other hand

$$H_{RI}(z^{-1}) = \frac{U_C(z^{-1})}{I^*_{FI}(z^{-1}) - I_R(z^{-1})}. \tag{50}$$

Taking into account the Z-transform table, and using Eqs. (49) and (50), the digital implementation of the current regulator is

$$U_c(k) = U_c(k-1) + k_3 \cdot [I_{FI}^*(k) - I_R(k)] + (k_4 T - k_3)[I_{FI}^*(k-1) - I_R(k-1)]. \tag{51}$$

Taking into consideration the dc-dc power converter ideal amplifier model, the rotor voltage is obtained:

$$u_A(k) = k_D u_c(k). \tag{52}$$

By taken into consideration one of the following equations, the output signal of the rotor current loop is obtained:

- By neglecting the electromotive voltage

$$I_A(k) = I_A(k-1) \cdot e^{-\frac{T}{T_A}} + u_A(k-1) \cdot \left(1 - e^{-\frac{T}{T_A}}\right) \cdot \frac{1}{R_A} \tag{53}$$

- By considering the electromotive voltage

$$I_A(k) = I_A(k-1) \cdot \mathrm{e}^{-\frac{T}{T_A}} + [u_A(k-1) - C_e N(k-1)] \cdot \left(1 - \mathrm{e}^{-\frac{T}{T_A}}\right) \cdot \frac{1}{R_A}. \tag{54}$$

8.2. Mathematical model of the digital speed control loop

In **Figure 7**, the block diagram of the digital speed control loop is shown, in which the equivalent closed loop transfer function of the armature current control is replaced by current loop block.

Taking into consideration the block diagram shown in **Figure 7**, in the similar manner, the digital implementation of the speed loop is obtained.

Therefore, the imposed filtered speed is

$$N_{FT}^*(k) = N_{FT}^*(k-1) \cdot e^{-\frac{T}{T_{FT}}} + N^*(k-1) \cdot \left(1 - e^{-\frac{T}{T_{FT}}}\right). \tag{55}$$

The digital equation of the recurrent feedback speed is

$$N_R(k) = N_R(k-1) \cdot e^{-\frac{T}{T_T}} + N^*(k-1) \cdot \left(1 - e^{-\frac{T}{T_T}}\right) k_T. \tag{56}$$

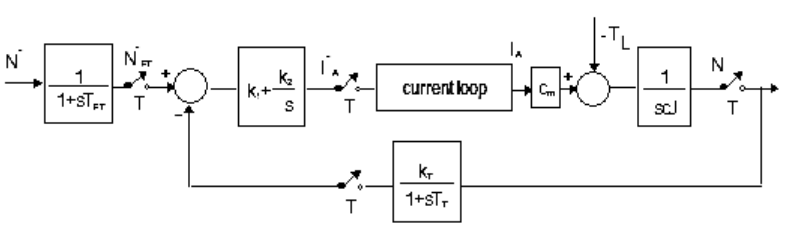

Figure 7. The speed control loop.

The speed controller for numerical implementation is as:

$$I_A^*(k) = I_A^*(k-1) + k_1 \cdot [N_{FT}^*(k) - N_R(k)] + (k_2 T - k_1)[N_{FT}^*(k-1) - N_R(k-1)], \tag{57}$$

with the obtained numerical signal of the speed feedback

$$N(k) = N(k-1) + \frac{T}{cJ} \cdot [C_m I_A(k-1) - T_L(k-1)]. \tag{58}$$

By organizing the numerical equations in the specified order Eqs. (55)–(57), (45), (47), (51)–(53), (58), the digital representation of the DC drive system is obtained.

9. Numerical simulation results

Taking into consideration the following DC motor of 12 W, 12 V/1 A, 90 r.p.m. at rated load torque of 1.37 Nm, the cascaded control loops are implemented in Matlab by using the deducted numerical equations. The following numerical results have been obtained.

In **Figure 8** (left side), the dynamic mechanical characteristic for a starting under different load torque variation (**Figure 8**, right side) is shown. At the same time, in **Figure 9**, the implemented load torque estimator result is presented.

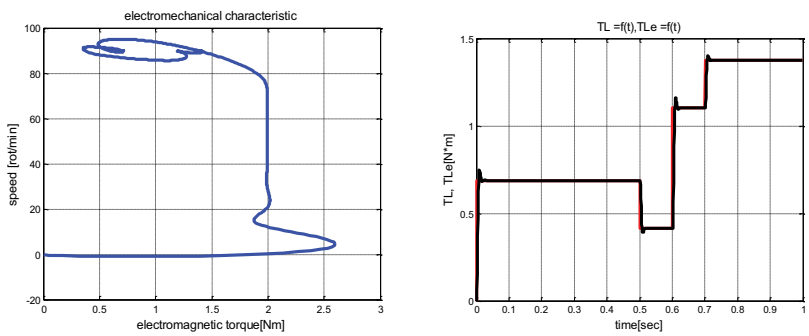

Figure 8. Mechanical characteristic of the DC motor (left side). Real (step signal) and the estimated load torque (delayed signal).

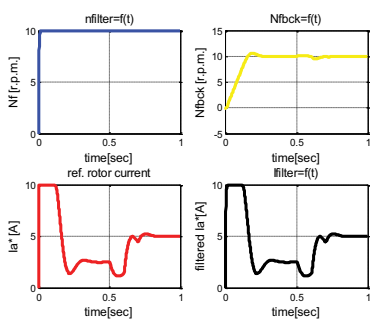

Figure 9. The control signals of the DC drive.

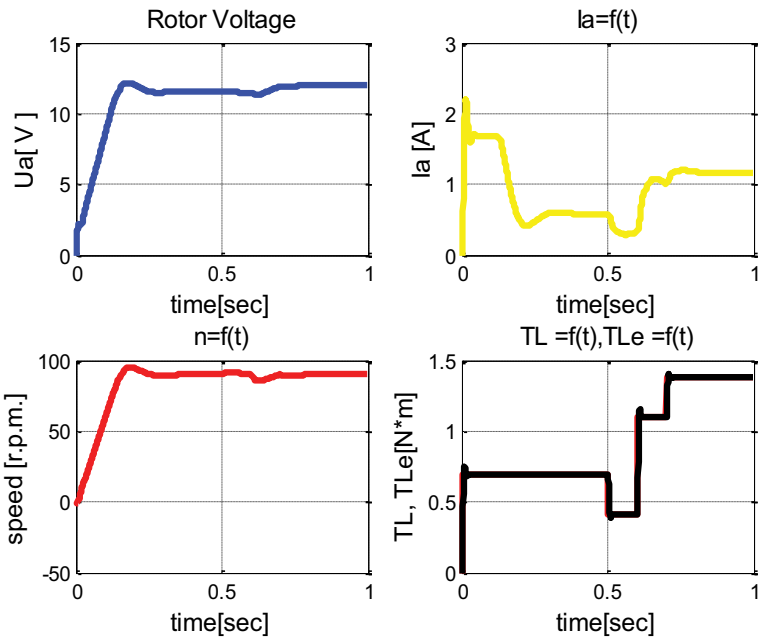

Figure 10. The power signals of the DC drive.

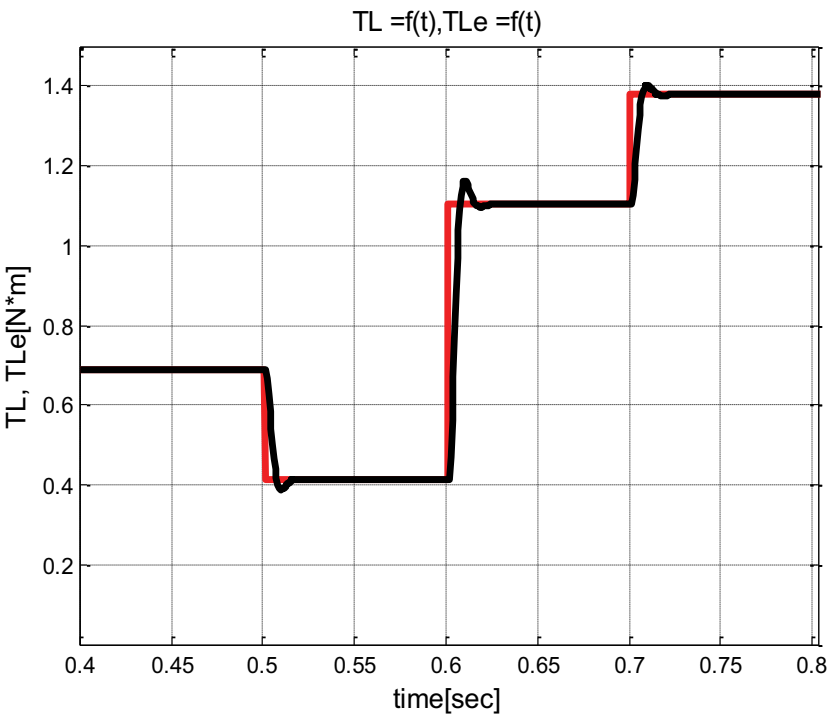

Figure 11. The performances of the second-order load torque estimator.

http://dx.doi.org/10.5772/67461

Based on the above mentioned digital implementation and taking into consideration that the DC drive is formed by two parts, control and power, the following simulation results are obtained. Therefore, on the control side, there are only analog voltages ([0–10] V unified system) (**Figure 9**). On the power side, the real signals are shown (**Figure 10**). In **Figure 9**, the following DC dive signals are shown: the filtered speed reference (*nfilter*), the feedback speed (*Nfbck*), the output of the speed controller (i.e., the reference of the rotor current, *ref. rotor current*), and the filtered rotor current reference (*Ifilter*).

On power side, the real values of the DC drive signals are presented: the rotor voltage (*Ua*), the rotor current (*Ia*), the controlled feedback speed as the output signal of the DC drive (*n*), and the load torque signals (the real, *TL* and the estimated, *TLe*). The more accurate estimator performances are presented on the [0.4–0.8]s time interval, in **Figure 11**.

10. Real-time speed control

Based on the Arduino platform, the conventional control of the DC drive system is implemented in real time (**Figure 12**). It consists of the cascaded control, on-line estimation of the DC motor parameters (**Figure 13**) based on the recursive least squares method [1], and real-time estimation of the load torque (**Figure 14**) [2]. The speed feedback is assured by using an incremental encoder [8]. In order to improve the speed response of the DC drive system, a feedforward additional voltage component has been introduced, as in **Figure 15** [9]. In **Figures 16–18**, the build s-function and the associated Simulink blocks for speed and position calculus from the encoder signals are shown.

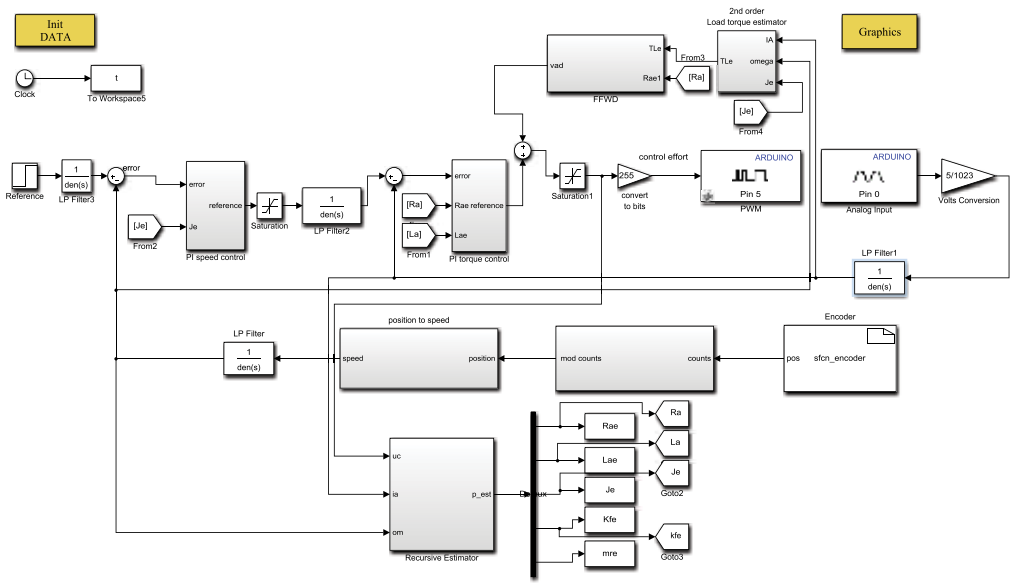

Figure 12. The embedded DC drive cascaded control.

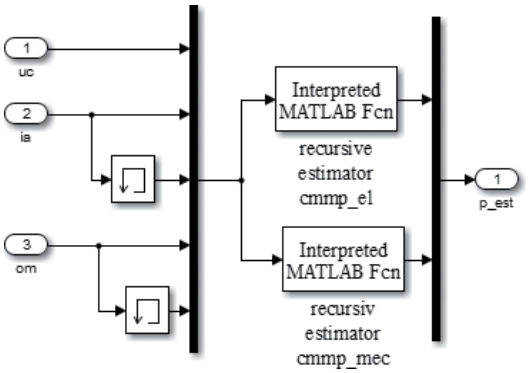

Figure 13. The on-line estimator of the DC motor.

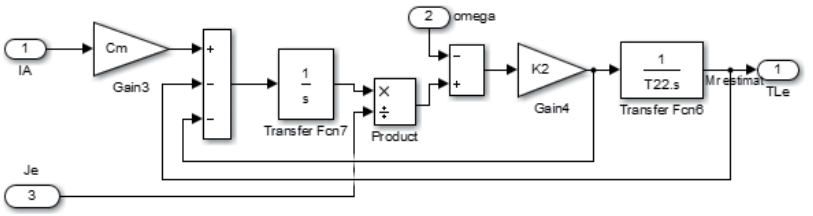

Figure 14. The second-order load torque estimator.

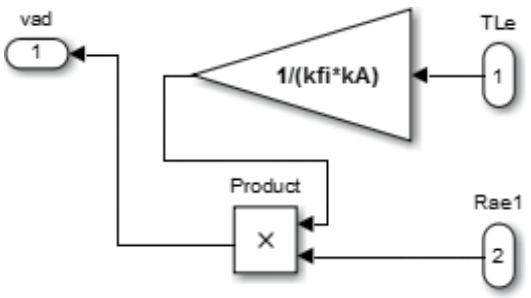

Figure 15. The feedforward control voltage component.

The appropriate Matlab functions [1] from **Figure 13** are delivered as follows:

```
function tetael=cmmp_el(in)
global tetael_an Pel_an lambda_el Kfi T
y=in(1)-Kfi*in(4);
fi=[ in(2);(in(2)-in(3))/T];
L=(Pel_an*fi)/(lambda_el+fi'*Pel_an*fi);
```

http://dx.doi.org/10.5772/67461

```
tetael=tetael_an+L*(y-fi'*tetael_an);
Pel=(Pel_an-(Pel_an*fi*fi'*Pel_an)/(lambda_el+fi'*Pel_an*fi))/
lambda_el;
tetael_an=tetael;
Pel_an=Pel;

function tetamec=cmmp_mec(in)
global tetamec_an Pmec_an lambda_mec alpham denumm Pm Kfi T
y=Kfi*in(2);
fi=[ (in(4)-in(5))/T;1;in(4)];
denumm=(lambda_mec+fi'*Pmec_an*fi);
L=(Pmec_an*fi)/denumm;
alpham=(y-fi'*tetamec_an);
tetamec=tetamec_an+L*alpham;
Pm=L*fi'*Pmec_an;
Pmec=(Pmec_an-Pm)/lambda_mec;
tetamec_an=tetamec;
Pmec_an=Pmec;
```

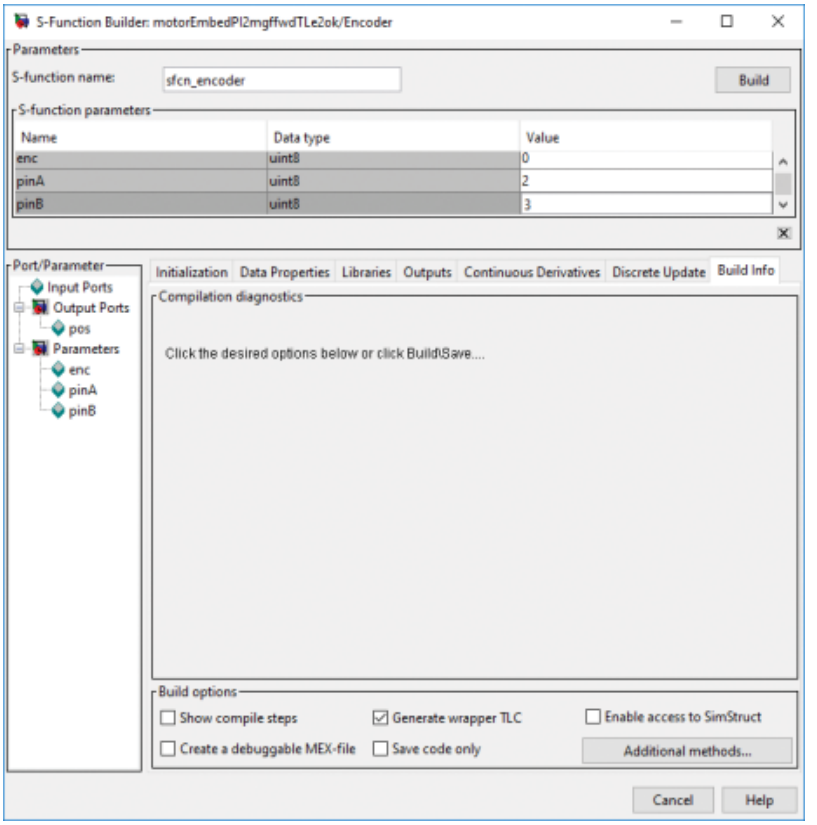

Figure 16. The s-function block of the encoder [8].

By applying the numerical derivative of the position adapted to gear ratio the real speed value is obtained (**Figure 19**).

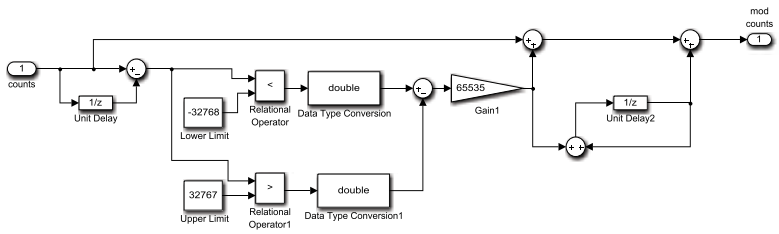

Figure 17. The calculus of the DC rotor position [8].

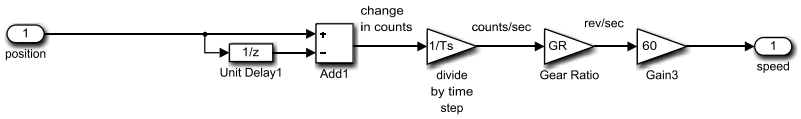

Figure 18. The numerical derivation of the position [8].

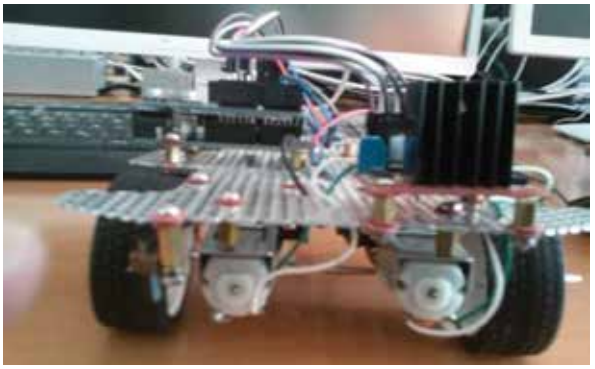

Figure 19. The DC drive platform.

11. Conclusions

This chapter is designed in the didactical manner that allows students and researchers to implement autonomously the mechatronic projects based on the DC drive. Starting from the nameplate data of the DC motor, the provided Matlab software has the ability to calculate the DC motor parameters, the parameters of the current and speed controllers, and includes the numerical implementation of the DC motor cascaded control, energy balance and of the load torque estimator. The price of the proposed DC drive is down due to the fact that the load torque is very expensive and it is replaced by the load torque estimator. By using the Z-transform, the procedure of numerical control is implemented. The implementation of the digital filters and controllers are also provided. Numerical simulations confirm the accuracy of the

proposed DC drive discretization method. Moreover, the real-time control based to Arduino platform has been performed. In order to increase the DC drive performances, the on-line estimator of the DC motor parameters is provided, and the feedforward control voltage component has been added.

Acknowledgements

This work was supported by a grant of the Romanian National Authority for Scientific Research, CNDI–UEFISCDI, project number PN-II-PT-PCCA-2011-3.2-1680.

Appendix

```
% CONVENTIONAL AUTOMATION OF THE DIRECT CURRENT MOTOR
clear;clf;
%                DC DRIVE SYSTEM
% Nameplate Data
%    -Rated Power
      Pn=12        %W
%    -Rated Voltage
      Un=12          %V
%    -Rated Speed
      nn=90          %rot/min
%    -moment of inertia
      Jt=0.02        %kgm^2
      J=Jt;
%    -Rated efficiency
      etan=0.86 %
% Main data calculus
%    -Electrical Power(input)
      P1=Pn/etan     %W
%    -Rated current
      Ian=P1/Un    %A
%    -DC motor losses
      deltap=P1-Pn    %W
%      -Rated winding losses
        pcun=deltap/2    %W
%    -Rotor resistance
      Ra=pcun/(Ian^2)    %ohmi
%    -Electromagnetic Power
      P=Pn+deltap/2    %W
%    -Electromagnetic torque
      Mn=30*P/pi/nn    %Nm
%    -Mechanical constant
```

```
     kfi=Mn/Ian
     Cm=kfi
%    -Electrical constant
     kefi=pi*kfi/30
     Ce=kefi
%    -Electromotive voltage
     E=Ce*nn    %V
%    -Rotor resistance value check Ra
     Ra=(Un-E)/Ian    %ohmi
%    -volant moment
     g=9.81;    %gravit. acc.
     GD2=4*g*Jt    %m^2
%    -Maximum torque
     Mmax=2*Mn    %Nm
%    -Maximum rotor current
     Iamax=Mmax/Cm    %A
%    -Electrical constant (from identification procedure)Ta
     Ta=7/(10^3)    %sec
%    -Rotor inductance
     La=Ta*Ra    %H
%    -Load torque
     Mr=1*Mn;    %Nm
     k=pi/30;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%              Current loop- tuning of the controller parameters
%       (modulus criterion - KESSLER variant)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%current transducer
%%%%%%%%%%%%%%%%%%%%
%    -for DC motor
     ka=1/Ra    %(1/ohmi)
%    -attenuation factor of the current transducer
     ki=10/Iamax    %(V/A)
%    -Current transducer time constant
     Ti=0.003    %sec
%%%%%%%%%%%%%%%%%%%%%%
%DC-DC Power Converter
%%%%%%%%%%%%%%%%%%%%%%
%    -gain factor
     kd=Un/10
%%%%%%%%%%%%%%%%%%%%%%%
% PI Current Controller
%%%%%%%%%%%%%%%%%%%%%%%
%    -sum of the parasitic constant of the current loop
```

```
     Tsi=Ti    %sec
%   -Proportional coefficient
     tau1=Ta    %sec
%   -gain factor
     k1=1/(2*Tsi*kd*ka*ki)
%   -Time constant of the current filter
     Tf1=Ti    %sec
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%              Speed loop- tuning of the controller parameters
%       (symmetrical criterion - KESSLER variant)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%speed transducer
%%%%%%%%%%%%%%%%%%%%
%   -attenuation factor of the speed transducer
     kt=10/nn    %(V/r.p.m.)
%   -Time constant
     Tt=Ti     %sec
%   -gain factor of the mechanical inertia(1/s*Jt)
     kj=375/GD2;
%   PARAMETERS OF THE PI SPEED CONTROLLER
%   -sum of the parasitic constant of the speed loop
     Tst=2*Tsi+Tt     %sec
%   -Proportional coefficient
     tau2=4*Tst     %sec
%   -gain factor
     k2=ki/(kj*kt*Cm*8*(Tst^2))
%   -Speed time constant
     Tf2=Tt
%              NUMERICAL SIMULATION OF THE DC DRIVE SYSTEM
%   - sampling time
     T=Ta/10;    %sec
%   -initial conditions
%   a)current loop
     Ir=0;   %A    %output signal of the current transducer
     If=0;   %A    %output signal of the current filter Ii
     U=0;    %V    % DC-DC voltage
     Ia=0;   %A    % rotor current
%   b)speed loop
     Nf=0;   %r.p.m.    %output signal of the speed filter Ni
     Nr=0;   %r.p.m.    %output signal of the speed transducer
     Ni=10;  % V    reference speed
     N=0;    %r.p.m.    real speed
     Ii=0;   %A    reference current -initial value
     t=0;    %sec    % initial time
```

```
     tf=1;   %sec    %final time
     i=1;    %counter
%%%%%%%%%%%%%%%%%%%%%%%%%%
%power balance estimation
%%%%%%%%%%%%%%%%%%%%%%%%%%
   swcu=0;      %estimation of the copper losses
   swl=0;      %estimation of the accum. energy in the rotor inductance
   swj=0;      %estimation of the accum. energy in thea motion masses
   wpr=0;      %estimation of the input energy
   wp=0;
%%%%%%%%%%%%%%%%%%%%%%%
%Load torque estimator
%%%%%%%%%%%%%%%%%%%%%%%
T0=0.0024    %2*T;
zita=0.707;
Mre=0;    vt2=0;    vt4=0;
taue=2*zita*T0;
k2e=2*zita/kj/T0;
while (t<=tf),
   vtnf(i)=Nf;    %speed vector
   vtnr(i)=Nr;      vtii(i)=Ii;
   vtn(i)=N;        vtir(i)=Ir;
   vtif(i)=If;      vtu(i)=U;
   vtia(i)=Ia;      mre(i)=Mre;
   % perturbation load torque wp
   mre(i)=Mre;
   mp(i)=wp;
   epsilon(i)=wp^2-Mre^2;
   %output signal of the speed filter at the current step(k)
   Nfu=exp(-T/Tt)*Nf+(1-exp(-T/Tt))*Ni;
   % speed feedback at the current step(k)
   Nru=exp(-T/Tt)*Nr+kt*(1-exp(-T/Tt))*N;
   %output signal of the speed controller=reference of the rotor current(k)
   Iiu=Ii+k2*tau2*(Nfu-Nru)+k2*(T-tau2)*(Nf-Nr);
   % allowable limits of the rotor current
   if Iiu>10,
        Iiu=10;
   elseif Iiu<-10,
        Iiu=-10;
      else
        Iiu=Iiu;
   end
   % filtered reference of the rotor current,at the current step(k)
   Ifu=exp(-T/Ti)*If+(1-exp(-T/Ti))*Ii;
```

http://dx.doi.org/10.5772/67461

```
%feedback rotor current (k)
Iru=exp(-T/Ti)*Ir+ki*(1-exp(-T/Ti))*Ia;
% output of the current controller= rotor voltage (k)
Uu=U+k1*kd*tau1*(Ifu-Iru)+k1*kd*(T-tau1)*(If-Ir);
%load torque estimator
vt2u=vt2+T*kj*(Cm*Ia-Mre-vt4);
Mreu=Mre+T*k2e*(vt2-N)/taue;
%rotor current(k)
Iau=exp(-T/Ta)*Ia+ka*(1-exp(-T/Ta))*(U-Ce*N);
 wl(i)=La*Iau*(Iau-Ia); %acum. en. in the L.rotor inductance.at the
 current sampling
 swl=swl+wl(i);      % acum.en. in the L.rotor through the starting
 period
%rotor speed(k)
Nu=N+kj*T*(Cm*Ia-wp);
vt4u=(vt2u-Nu)*k2e;
%load torque variation
if t<=0.5,
  wp=0.5*Mr;
elseif (t<=0.6) & (t>0.5),
      wp=0.3*Mr;
  elseif (t<=0.7) & (t>0.6),
      wp=0.8*Mr;
    elseif(t<=0.8) & (t>0.7),
           wp=1*Mr; %alfa*t;
elseif(t<=0.9) & (t>0.8),
  wp=1*Mr;
else
  wp=1*Mr;
end
% acum. en.in the rotational mass
wj(i)=(k^2)*Jt*Nu*(Nu-N);
swj=swj+wj(i);
%energy estimation
pcu(i)=Ra*(vtia(i)^2);     % copper power losses
swcut(i)=pcu(i)*T;     % energy losses at each sampling time
swcu=swcu+swcut(i);     % rotor energy losses
pr(i)=vtu(i)*vtia(i);     % the input electrical power
wpr=wpr+pr(i)*T;     % the input energy
Nf=Nfu;      Nr=Nru;      Ii=Iiu;
N=Nu;
Ir=Iru;      %Ir(k-1)
If=Ifu;      %If(k-1)
U=Uu;      %U(k-1)
```

```
    Ia=Iau;      %Ia(k-1)
    Mre=Mreu;      vt2=vt2u;      vt4=vt4u;
    t=t+T;      % simulation time increment
    i=i+1;      % counter increment
  end
  wu=wpr-swcu-swl-swj % DC motor output poweren.
  eta=wu/wpr; %the efficiency of the DC drive
figure(1)
t=0:T:tf;

  plot(vtia*Cm,vtn,'b','LineWidth',3)
  title('electromechanical characteristic','FontSize',12)
  xlabel('electromagnetic torque[ Nm] ','FontSize',12)
  ylabel('speed[ rot/min] ','FontSize',12)
  grid

  fprintf('vizualizati caracteristica mecanica')
 figure(2)
  %t=0:T:tf-T;
  t=0:T:tf;
  subplot(2,2,1);
  %plot(t,vtnf(1:length(t)),'c','LineWidth',3)
  plot(t,vtnf,'b','LineWidth',3)
  title('nfilter=f(t)','FontSize',12)
  xlabel('time[ sec] ','FontSize',12)
  ylabel('Nf[ r.p.m.] ','FontSize',12)
  grid

  subplot(2,2,2);
  plot(t,vtnr,'y','LineWidth',3)
  title('Nfbck=f(t)','FontSize',12)
  xlabel('time[ sec] ','FontSize',12)
  ylabel('Nfbck[ r.p.m.] ','FontSize',12)
  grid

  subplot(2,2,3);
  plot(t,vtii,'r','LineWidth',3)
  title('ref. rotor current','FontSize',12)
  xlabel('time[ sec] ','FontSize',12)
  ylabel('Ia*[ A] ','FontSize',12)
  grid

  subplot(2,2,4);
```

http://dx.doi.org/10.5772/67461

```
  plot(t,vtif,'k','LineWidth',3)
  title('Ifilter=f(t)','FontSize',12)
  xlabel('time[ sec] ','FontSize',12)
  ylabel('filtered Ia*[ A] ','FontSize',12)
  grid

figure(3)
  subplot(2,2,1);
  plot(t,vtu,'b','LineWidth',3)
  title('Rotor Voltage','FontSize',12)
  xlabel('time[ sec] ','FontSize',12)
  ylabel('Ua[ V] ','FontSize',12)
  grid

  subplot(2,2,2);
  plot(t,vtia,'y','LineWidth',3)
  title('Ia=f(t)','FontSize',12)
  xlabel('time[ sec] ','FontSize',12)
  ylabel('Ia[ A] ','FontSize',12)
  grid

  subplot(2,2,3);
  plot(t,vtn,'r','LineWidth',3)
  title('n=f(t)','FontSize',12)
  xlabel('time[ sec] ','FontSize',12)
  ylabel('speed[ r.p.m.] ','FontSize',12)
  grid

  subplot(2,2,4);
  plot(t,mp,'r',t,mre,'k','LineWidth',3)
  title('TL =f(t),TLe =f(t)','FontSize',12)
  xlabel('time[ sec] ','FontSize',12)
  ylabel('TL, TLe[ N*m] ','FontSize',12)
  grid
  fprintf('vizualizati fig.2')
figure(4)
  t=0:T:tf;
  %subplot(1,2,1);
  plot(t,mp,'r',t,mre,'k','LineWidth',3)
  title('TL =f(t),TLe =f(t)','FontSize',12)
  xlabel('time[ sec] ','FontSize',12)
  ylabel('TL, TLe[ N*m] ','FontSize',12)
  grid
```

```
  fprintf('vizualizati fig.3')
figure(5)
  t=0:T:34*T;
  i=1:35;
  subplot(1,2,1);
  plot(t,mp(i),'r','LineWidth',3)
  title('Mr =f(t)','FontSize',12)
  xlabel('time[ sec] ','FontSize',12)
  ylabel('Mr[ N*m] ','FontSize',12)
  grid

  subplot(1,2,2);
  plot(t,mre(i),'g','LineWidth',3)
  title('Mr^ =f(t)','FontSize',12)
  xlabel('time[ sec] ','FontSize',12)
  ylabel('Mre[ N*m] ','FontSize',12)
  grid
  fprintf('vizualizati fig.4')
  %energy components
  swcu %copper losses
  wpr %input energy
  swl %accum. Energy in rotor inductance Lrotor
  swj %accum. Energy in rotational mass
  eta %efficiency during starting
  omn=1/T0
  tr=-log(0.05*sqrt(1-zita^2))/zita/omn
```

Author details

Marian Gaiceanu

Address all correspondence to: marian.gaiceanu@ieee.org

Integrated Energy Conversion Systems and Advanced Control of Complex Processes Research Center, Dunarea de Jos University of Galati, Romania

References

[1] Gaiceanu M, Solea R, Codres B, Eni C. On-line Identification of the DC Motor Parameters by using Least Mean Square Recursive Method. Resita: Analele Universitatii "Eftimie Murgu". 2014. 1453–7397. **2**(ANUL XXI):ISSN.

[2] Rosu E, Bivol I, Nichita C, Gaiceanu M. Energetical Optimization of the Electromechanical Conversion Systems (Optimizarea energetica a sistemelor de conversie electromecanice). Bucharest: Technical House. 1999. 210 p. ISBN 973-31-1329-8.

[3] Puangdownreonga D, Nawikavatan A, Thammarat C. Optimal design of I-PD controller for DC motor speed control system by Cuckoo Search. Procedia Computer. 2016;**86**:1877–0509/83–86. DOI: 10.1016/j.procs.2016.05.021.

[4] Grignion D, Chen X, Tan Y, Qian H, Kar N. Robust load disturbance torque estimation for a DC motor drive system. IFAC Proceedings Volumes. 2012;**45**(13):460–466. DOI: 10.3182/20120620-3-DK-2025.00139.

[5] Babaei M, Asgharei R, Ahmarinejad A. Electromagnetic torque and speed estimators for permanent magnet synchronous motor drive systems. Energy Procedia. 2016;**100**:291–296. DOI: 10.1016/j.egypro.2016.10.180.

[6] Thanga Raj C, Srivastava SP, Agarwal P. Differential evolution based optimal control of induction motor serving to textile industry. IAENG International Journal of Computer Science. 2008;**35**(2):IJCS_35_2_03.

[7] Gaiceanu M. Linear quadratic regulator for the three-phase grid connected power converter. In: 2nd International Conference on Micro Nano Devices, Structure and Computing Systems; January 23–24; Shenzhen, China. 2013.

[8] Carnegie Mellon University. Control Tutorials for MATLAB and Simulink [Internet]. 2006. Available from: http://ctms.engin.umich.edu [Accessed: 2016].

[9] Springer International Publishing AG, Switzerland

Chapter 3

Remote Laboratories for Teaching and Training in Engineering

Jesús García-Guzmán, Farah H. Villa-López,
Jorge A. Vélez-Enríquez, Luz A. García-Mathey and
Alfredo Ramírez-Ramírez

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/67459

Abstract

Typical mechatronic systems are a combination of advanced technologies involving several disciplines. This multidisciplinary approach to the development of industrial applications provides great opportunities for the implementation of e-learning environments and collaborative schemes. Engineering education, in particular, benefits from many of these advances, among which, virtual instrumentation is a useful tool for the development of virtual environments, e-learning spaces and, particularly, remote laboratories. This chapter describes the implementation of web-based laboratories that allow the remote operation of experiments used as training exercises in undergraduate engineering courses. The remote laboratories were developed using LabVIEW® software, and they enable remote control and monitoring of laboratory equipment, allowing engineering students to perform experiments in real time, at their own pace, from anywhere, and whenever is suitable for them. Besides the experimental training that the web-based laboratories provide to students, the system is also a powerful teaching tool since real-time demonstrations of the experiments can be performed, and they also can be simultaneously monitored by a group of students. This approach is highly beneficial for engineering schools in developing countries, as resources can be shared through the Internet. A description of the system and three proposed experiments is presented, together with the experimental results.

Keywords: remote laboratory, e-learning environment, mechatronic system, virtual instrument, web-based system

1. Introduction

In engineering education, laboratories represent an important academic resource as they provide practical training in addition to the fundamental theories taught in lectures. At laboratories, engineering students have the opportunity to learn how to properly use the equipment and tools with which they will be interacting later in a professional environment, and therefore, they gain practical experience and familiarize with that equipment. For this reason, engineering schools seek to provide properly equipped laboratories. However, the maintenance of the equipment and the acquisition of new machinery imply a large investment that only a limited number of public universities can afford.

The limited budgets and the large amount of students at engineering schools make insufficient the available resources at the laboratories and consequently, it is very difficult for universities to provide quality experimental training to all engineering students. Taking advantage of new communication technologies such as the Internet and computing tools such as virtual instrumentation, the available resources can be shared, developing and implementing collaborative schemes and e-learning environments that allow the access to practical training to a larger amount of students, regardless their location.

Several collaborative schemes, aimed at engineering education, have been developed in recent years for the remote execution of experiments in distance laboratories and on different engineering fields. For example, the remote laboratory system described in Ref. [1], where experiments conducted at control engineering laboratories can be remotely operated through the Internet. On the same trend, a more recent work [2] reports on the remote control of a nonlinear system as a tool for teaching several engineering subjects. Likewise, experiments with analog electronic circuits have been studied [3] with the aid of remote laboratories based on an existent and previously validated platform. This system aims the use of nonproprietary solutions in order to promote sharing among institutions; it also performs remote measurements over real instruments in an effort to make it as reliable as a hands-on laboratory.

Remote laboratories for experimental training on mechatronics have also been implemented, as the one reported in Ref. [4], where experiments using a two-degree of freedom robot and a servomotor can be carried out remotely. Other works have also reported the development of remote experiments implemented as web-based systems in other engineering fields such as fluid mechanics [5] and electrical engineering [6].

In general, remote laboratories deal with manipulation of real equipment and experiments, and although they make use of tools such as virtual instruments (VI) and e-learning environments, they should not be confused with either simulations or virtual laboratories. There are systems, such as the one reported in Ref. [7], using different approaches, more on the style of virtual laboratories, that involve the use of expensive equipment and topologies for the emulation of sophisticated laboratories, which are rarely available in public institutions.

Several computing tools have been employed to develop remote laboratories for engineering education. As an example, Matlab and Easy Java Simulations were used for the implementation of control system experiments in the work reported in Ref. [8]. However, since

early works on distance laboratories [9, 10], LabVIEW® virtual instrumentation has been the primary used tool, and this software has become a powerful resource for the development of e-learning environments, particularly at the development stage.

Although much work has been done for several years in the field of remote laboratories, there are still many challenges before they can be considered a quotidian resource for teaching and training in engineering. Some of these challenges and proposed solutions have already been considered [11, 12], but there is still much work left to do.

The remote mechatronic systems presented in this chapter follow the outcomes of a previous work [13] and aim to provide practical training to a larger amount of students in a public university with engineering schools in five campuses, looking to overcome current limitations such as the lack of economic resources, the consequent insufficiency of laboratory equipment, and the limited flexibility of laboratory scheduling. Three experiments for training on different engineering fields have been implemented in a web-based learning environment so that they can be remotely operated over the Internet. The web-based system was developed using LabVIEW® virtual instrumentation, and its configuration allows multiple students to access the experiments simultaneously, making the proposed system suitable for teaching and practical training.

2. Description of the web-based system

Traditionally, students who are physically at the laboratories are able to carry out experiments by making all the required connections of the laboratory equipment, and then they manually operate tools and machinery in order to execute experimental exercises. Frequently, they also measure and calculate some parameters so as to better understand the theoretical knowledge. In contrast, remote laboratories are implemented as web-based system, replacing manual operation of the experiments and enabling remote access by means of software, such as LabVIEW® virtual instrumentation, and additional hardware.

A particular web-based system has been developed for the implementation of several experiments in three different remote laboratories. The general characteristics of the system, namely overall structure, virtual instrumentation, and data acquisition (DAQ) system, are described in this section.

2.1. Overall structure

The overall outline of the developed system is shown in **Figure 1**. The required laboratory equipment to perform the experiment is already connected at the laboratory and thanks to the use of virtual instrumentation and additional hardware the experiment can be controlled through a computer interface and over the Internet.

A laptop computer equipped with LabVIEW® software is located at the actual laboratory where the experiment is performed. This is the *server computer* where the virtual instruments are developed for the control of the experiment execute. As stated before, additional

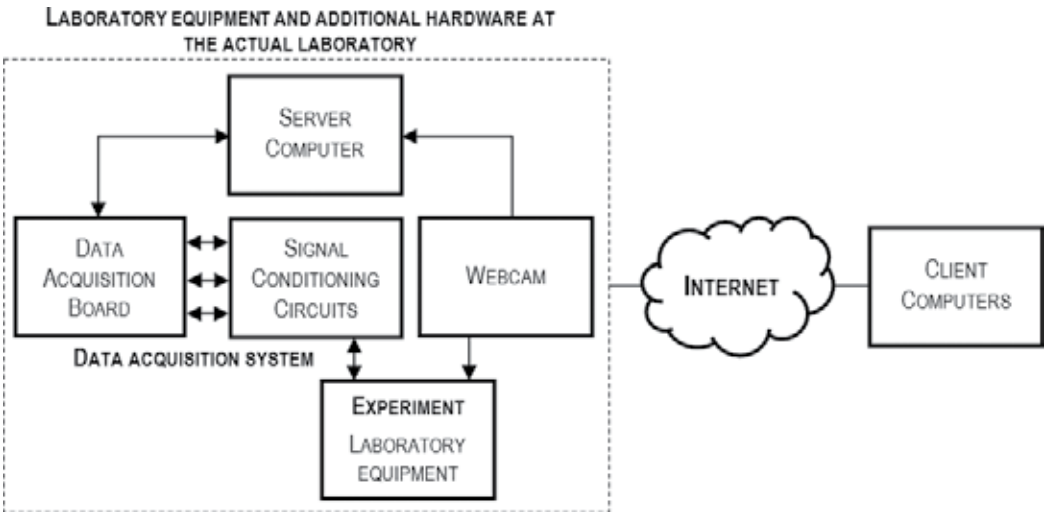

Figure 1. Overall structure of the developed web-based system.

hardware was employed for the implementation of the system, namely a *data acquisition* (DAQ) *board* and a *webcam*, both connected to USB ports of the server computer, as well as *signal conditioning circuits* for the conversion of software instructions into the required power actions on the laboratory equipment. The webcam allows video communication with the laboratory, displaying images through the computer interface in order to monitor the experiment. Using the LabVIEW® web server, the experiment is available on the Internet, and it can be accessed from anywhere providing that the user has a computer (*client computer*) with an Internet connection.

2.2. Virtual instrumentation

In order to accomplish every required task of the experiment through the computer interface, LabVIEW® software was employed to develop a set of virtual instruments. This software uses a graphical programming language in which each node is a virtual instrument, or VI, and the lines or wires connecting the nodes determine the flow of data. A virtual instrument consists of a *front panel* and a *block diagram*. The front panel represents the virtual interface for the user, and it accommodates *indicators*, or output nodes displaying data, and *controls*, or input nodes, in which data can be read from. The block diagram contains the source programming code, linking nodes with wires according to the desired flow of data. Programming structures are also represented graphically in the form of blocks with iterations or cases. The software is able to read external data in many ways, particularly through data acquisition boards and, hence, providing an excellent platform for the monitoring of real-time signals, a feature that makes LabVIEW® a very useful tool for engineering applications.

Three different main virtual instruments were developed, each of them for the implementation of one of the proposed experiments. In general, the front panels of these virtual instruments show graphical indicators, allow the control of certain parameters, and display images

http://dx.doi.org/10.5772/67459

in real time of the actual laboratory equipment controlled. Through these VI, each experiment can be controlled from the laptop computer, allowing the same practical procedures that are conventionally made by hand, e.g., pushing start/stop buttons, measuring voltages, noticing alert signals, and registering output data. Additionally, virtual instrumentation allows operations that are not easy to perform by hand, such as obtaining graphical information in real time, registering data directly into datasheet files, or recording videos of the actual experiments. It is also possible to develop tutorials using virtual instrumentation, in order to train the users in the procedures of the experiments, before they can work with the real equipment.

2.3. Data acquisition system

The data acquisition system comprising the DAQ board and the signal conditioning circuits is necessary to acquire signals sent by the laboratory equipment to the computer and to drive the actual control of the equipment after interpreting software instructions.

In order to carry out the data acquisition process, a 16-bit National Instruments multifunction DAQ board was employed for the implementation of each of the experiments, and owing to the fact that this board does not support the electric current values sent by the laboratory equipment nor it sources the required current for driving the control of the experiments, different signal conditioning circuits connected to the inputs and outputs of the DAQ board were used for each of the experiments, according to the needs. In general, these circuits were used to amplify the electrical signals sent by the digital outputs of the DAQ board and to protect this board from high voltages when receiving the electrical signals sent by the laboratory equipment. The DAQ Assistant Express VI was used in LabVIEW® software to configure the digital and/or analog ports of the DAQ board, as it was required by each experiment. Further explanation of required circuitry and DAQ board set-up is given below.

3. Remote access operation

Programming virtual instruments only to control the experiments, using a computer near to the equipment in the laboratory, may have some advantages, as the ones mentioned in Section 2.2. Nevertheless, the relevant feature of the developed systems reported in this work is their ability to be accessed from anywhere through the Internet.

The LabVIEW® web server was employed in order to remotely access, over the Internet, the developed virtual environments. Using this server, the front panel of the virtual instrument can be accessed in real time and simultaneously from different locations. However, even though multiple students can access the front panel at the same time, only one of them can control the front panel and perform the experiment at a time. This is one of the most important features of the system since this allows teachers or instructors to carry out demonstrations of the experiments in real time, while the students can monitor the experiment from their own computer. This characteristic makes the system suitable as a teaching aid, in addition to its training purposes.

How remote engineering laboratories have become a powerful tool for teaching and learning has been noticed since reviewing the perception of students about the remote operation of experiments reported in Ref. [14], and it has also been shown in Ref. [15] that remote laboratories do provide educational benefits, when comparing the remote approach with the traditional hands-on laboratories.

The procedure for the configuration of the Web server and the creation of a Web site where the virtual instruments with the laboratory experiments can be accessed remotely is explained in the following paragraphs.

3.1. Configuration of the LabVIEW® Web server and creation of the HTML file

Before students can access the front panel and remotely control the experiment, a proper configuration of the LabVIEW® server is required. In this configuration, the Web server needs to be enabled; then, it is necessary to determine which HTTP port will be used to access the virtual instrument and who will have access to it. Once this configuration has been done, the next step is the publication of the virtual instrument in a web page by creating a HTML file. For this aim, the web publishing tool, available within LabVIEW® software, was used. Using this tool, the URL of the web page is provided with the IP address of the server computer. This URL is the one used to access the front panel from a web browser.

For the developed system, the port HTTP 80 was used since the server computer runs Windows operating system. The browser access was configured in order to enable access to the front panel to all users, i.e., from any IP address, allowing them to view and control the experiments. An important part of this configuration is the control time limit; with this option in can be specified how long a student can remotely control the front panel until that control is granted to another student who is requesting or has previously booked the control of the virtual instrument.

For the creation of the HTML file, the virtual instrument to be published was selected and the option *Embedded* was chosen. Here, the option *Enable IMAQ Support* must be selected as this allows the acquisition of images required by the virtual instruments. Besides, the option *Request control when connection is established* was also enabled so that when a student access the front panel through a web browser, the control of the virtual instrument will be automatically requested. Then, the name of the HTML file is given, and it is saved. Finally, the URL is provided with the name of the file and the IP address of the computer.

3.2. Access to the front panel

As stated before, several students can view remotely the front panel of the virtual instrument, but only one of them at a time can have the control of the experiment. LabVIEW® software does not need to be installed at the client computer in order to have access to the front panel through a web browser. Instead, the applications LabVIEW Run-Time Engine and Vision Run-Time Engine, from the same version of the LabVIEW® software running the virtual instruments, are necessary.

When a student accesses the front panel via the Internet, the control of the panel is requested. However, if another student is controlling the experiment at that time, the server will queue the request and the control of the experiment will be granted to the next student on the queue list, only when the student currently having the control decides to release that control or when the time limit established is reached. To release the control, students must right-click anywhere on the front panel and select the option *Release Control of VI*. If a student who already released the control wants to have it again, the control must be requested by right-clicking and selecting *Request Control of VI*. Only the user at the server computer, who could be the teacher or instructor, is able to regain the control of the front panel at any time and he or she can also check the queue list of users.

4. Remote experiments

Three different experiments were implemented in the web-based system developed. The experiments proposed are the control of an electropneumatic system, the control of AC motors, and the manipulation of residential electrical circuits. The first one is performed at the industrial automation laboratory of the main campus, whereas the other two are conducted at the electric machines laboratory. A laptop computer, a DAQ board, and a webcam were used for each of the experiments. The laboratory equipment and additional hardware required for each experiment are permanently installed at the laboratories of the main campus, but they can be accessed by students from campuses in other cities, sharing the available resources.

4.1. Control of an electropneumatic system

The first of the experiments operated from the web-based environment is the control of an electropneumatic system. The experiment represents the automatic operation of an industrial process for stamping of parts, and it is aimed at students taking an undergraduate course on industrial automation.

Pneumatic equipment is employed to perform this experiment, and for the control of the process, an electropneumatic circuit was designed and connected using double-acting cylinders, electrically actuated valves, and proximity sensors. The motion sequence followed by the cylinders executes the stamping process. Proximity sensors are used to detect the position (retracted of extended) of the cylinders, and valves provide compressed air to the cylinders according to the programming code.

The front panel of the virtual instrument developed for the programmed control of this experiment is shown in **Figure 2**, as seen from the web browser. This virtual interface consists of different buttons that allow students to control several parameters such as the number of parts to be stamped, the drying time for each part, and the activation of the air supply. Other buttons are used to start or stop the stamping process. Indicative LEDs are also placed in this panel to show which proximity sensors are activated, and graphical indicators show an animation of the stamping process. Real-time views of the actual laboratory equipment controlled are displayed at the top right corner.

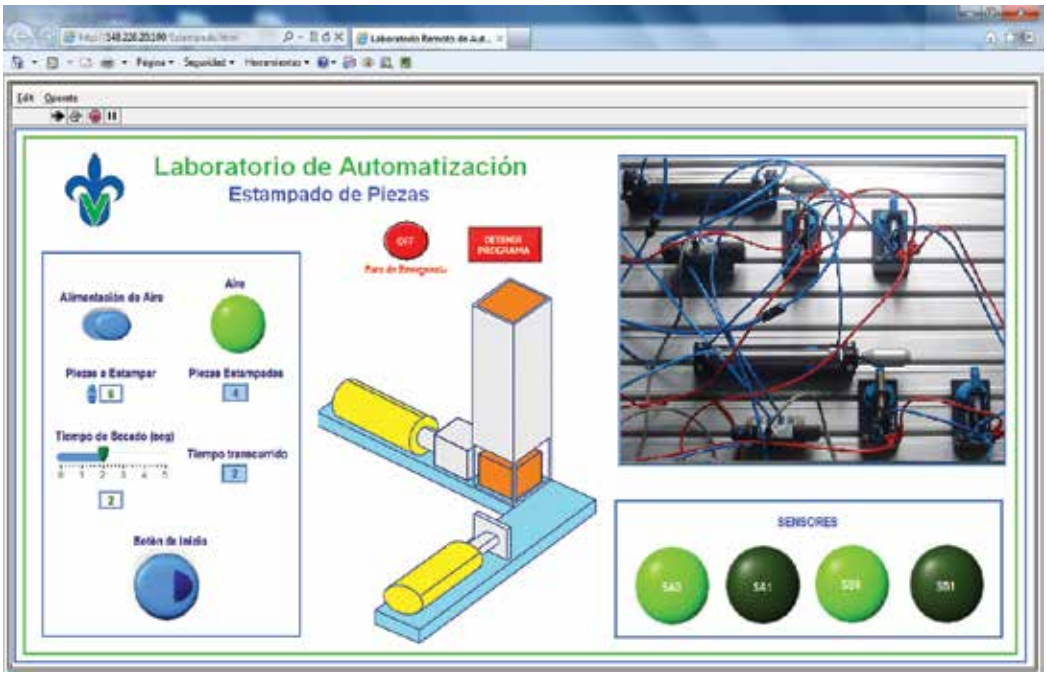

Figure 2. Front panel of the main virtual instrument for the control of the electropneumatic system [16].

The block diagram, shown in **Figure 3**, contains the programming code, in the LabVIEW® graphical programming language, called "G," of the main virtual instrument. It consists of graphical icons that are wired to determine the flow of data, and for this application, it is composed of global variables and three subroutines (known as subVIs) whose functions are the control of the system, the image acquisition, and the virtual connections of the system. Global variables pass data among the subVIs and the main virtual instrument when they are running at the same time.

For this experiment, only digital input and output ports of the DAQ board were used. The digital inputs acquire signals sent by the proximity sensors in the electropneumatic system, whereas the digital output ports generate the signals sent to the electrically actuated valves. Communication of input and output signals, however, is not direct because the DAQ board handles only low-voltage digital signals, while the components in the electropneumatic system work with higher voltage levels. For that reason, signal conditioning circuits, mounted on a breadboard, were placed between the control section and the laboratory equipment in order to amplify the electrical signals sent by the digital outputs of the DAQ board and then drive the current needed by the valves to be activated. Likewise, another circuit was used to protect the equipment when acquiring the current values from the sensors. Further details of the remote operation of this experiment are described in Ref. [16].

4.2. Control of AC motors

The second experiment implemented in a web-based laboratory is the control of two AC motors. In this experiment, two three-phase squirrel-cage induction motors can be controlled

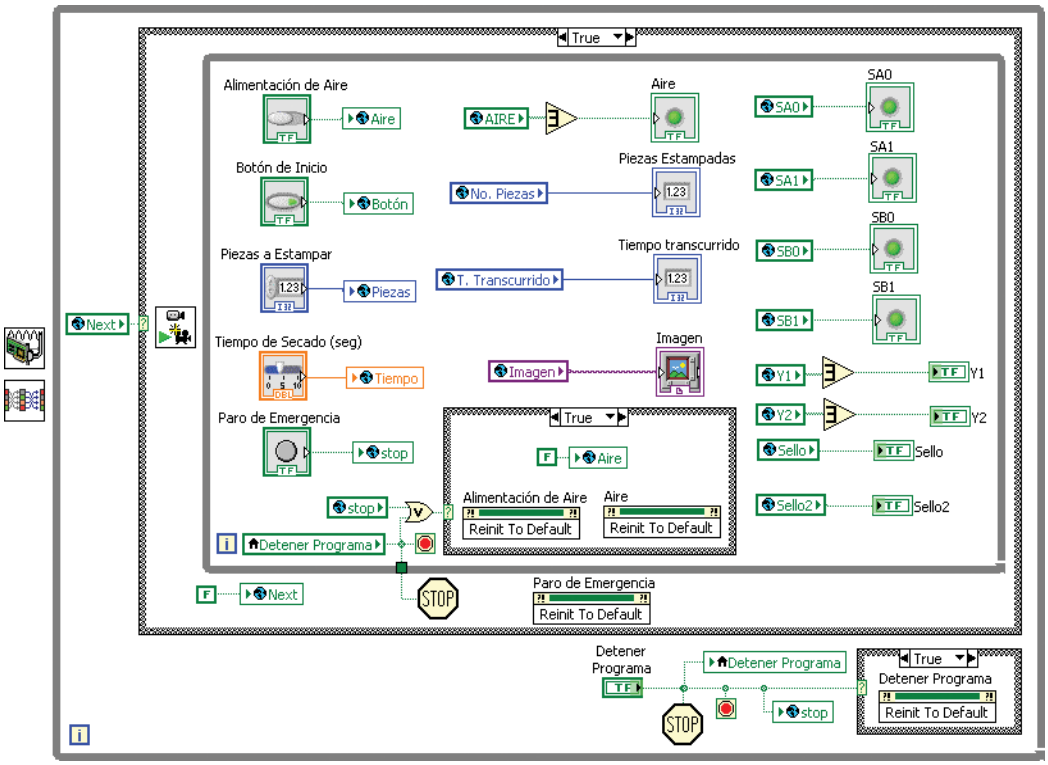

Figure 3. Block diagram of the main virtual instrument for the control of the electropneumatic system [16].

in three different ways, i.e., manually (using a virtual button), automatically, and sequentially. Students attending an undergraduate course on AC motors are the more likely to perform this experiment.

Figure 4 shows the front panel of the virtual instrument developed in LabVIEW® for the control of this experiment. Two graphical displays for voltage and current waveforms and several numeric indicators are placed in the front panel. These indicators provide useful information for students who traditionally need to measure and/or calculate parameters such as the voltage, current and real, apparent and reactive power for each phase of the tested motors. They can also visualize waveforms and phase sequence without connecting additional equipment such as oscilloscopes. For the starting of the induction motors, different buttons were placed in the pages of a tab control, according to the type of control that the students wish to perform. A camera shows the operation of the controlled motors through a real-time video image. There is also a stop button available, and all the actual equipment can be monitored in real time through the virtual interface.

Three different signal conditioning circuits were required for this experiment: one for the conditioning of the current signals acquired, a second one for the conditioning of the voltage signals acquired, and a third one for the amplification of the signals sent in order to start and stop the motors. In the case of the voltage and current signals, these must be reduced to the proper

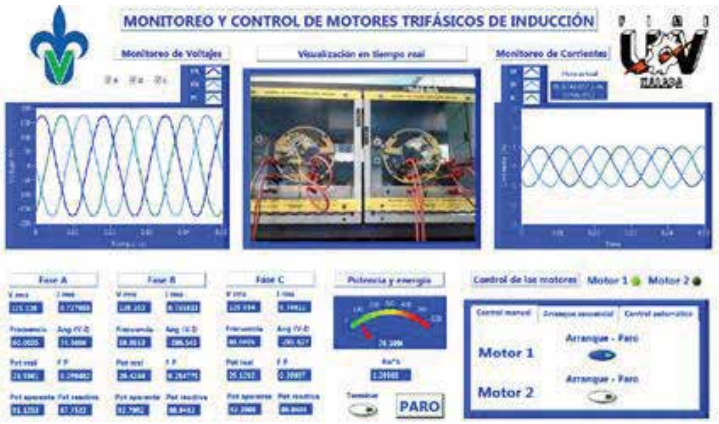

Figure 4. Front panel of the main virtual instrument developed for the control of two AC induction motors [17].

values supported by the DAQ board. The electrical signals required to start/stop the motors were sent by two different digital ports of the DAQ board, while six analog ports of the board were employed for the acquisition of the current and voltage signals. Detailed description of the implementation of this experiment can be found in the work reported in Ref. [17].

4.3. Manipulation of residential electrical circuits

The third experiment is directed to students of an electrical engineering program for training on residential electrical wiring and circuits. This experiment represents the electrical circuits installed around a house to deliver electricity for the lights and other appliances, and it allows turning on/off and monitoring the electrical circuits.

The proposed electrical wiring for a home consists of two separate circuits, both of them connected to the mains. The laboratory equipment employed for this experiment, namely cables, fluorescent and incandescent lamps, switches and other loads such as a domestic and an industrial ventilator were connected to a board already available for training on wiring at the laboratory.

The front panel of the developed virtual instrument for the control of this experiment is shown in **Figure 5**. This interface contains the controls and indicators used for turning on/off the circuits and for monitoring the electrical parameters. The manual (using a virtual button) or automatic control of the circuits can be chosen. If the automatic form is selected, time and date for turning on and off the circuits must be specified. Voltage and current values of each circuit are displayed graphically and numerically. The values of active power and reactive power, together with the power factor, are also shown, and LEDs are used to indicate whether the circuits are turned on or not. Last but not least, images of the real equipment are displayed in real time through this virtual interface.

The graphical programming code for this virtual instrument consists of five while loop structures. These structures execute the following functions: two of them perform the turning on/off of the

Figure 5. Front panel of the virtual instrument for the control and monitoring of residential electrical circuits [18].

circuits, a third one is for the image acquisition, another one for the acquisition of the electrical signals, namely current and voltage of the circuits, and the last one is for changing the background color of the front panel.

Four analog input ports of the NI-USB 6211 DAQ board were used to acquire voltage and current signals of the two circuits, while only two digital output ports of the same DAQ board were employed to generate the signals sent in order to turn on (or off) those circuits. As in the previous ones, signal conditioning circuits were also necessary for the implementation of this experiment. Further description of the experiment and the virtual environment is described in Ref. [18].

5. Experimental results

For the implementation of the proposed experiments in the web-based system developed, the equipment and components required for each experiment were connected at the corresponding laboratory. For each experiment, a NI-USB 6211 multifunctional DAQ board and a webcam were connected to a laptop computer equipped with LabVIEW® software and placed at the laboratories, as well as the required signal conditioning circuits. **Figure 6** shows the whole experimental setup in the laboratory for the remote control of AC motors. **Figure 7** shows the components, equipment, laptop computer, and power interfaces for the remote experiment on residential electrical circuits. Finally, the experimental setup for the practical exercises in the electropneumatic laboratory is shown in **Figure 8**.

Figure 6. Experimental setup for the control of AC motors [17].

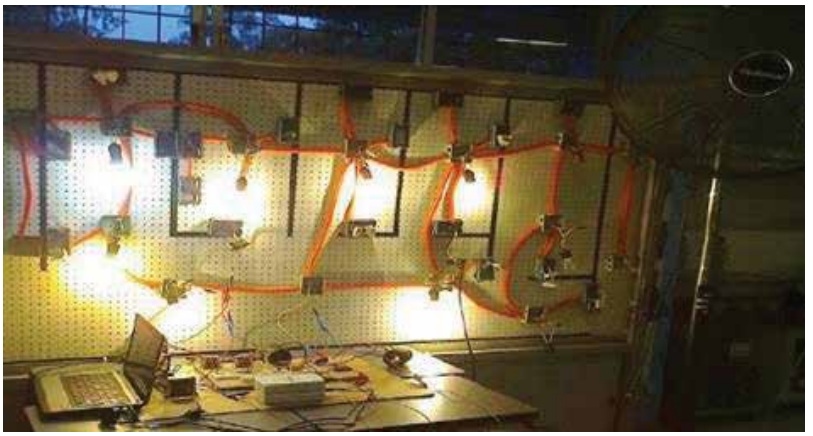

Figure 7. Experimental setup for the manipulation of residential electrical circuits [18].

Perhaps, the most important part of the project was the evaluation of these mechatronic systems when they were tested by engineering students. However, a complete evaluation of remote laboratories should cover many different aspects, from engineering and technical issues to the educational and pedagogical concerns. To this aim, standards have been defined and some studies have been reported, such as the one in Ref. [19], which have tried to cover at least part of the quality properties of remote laboratories for teaching and training in engineering.

http://dx.doi.org/10.5772/67459

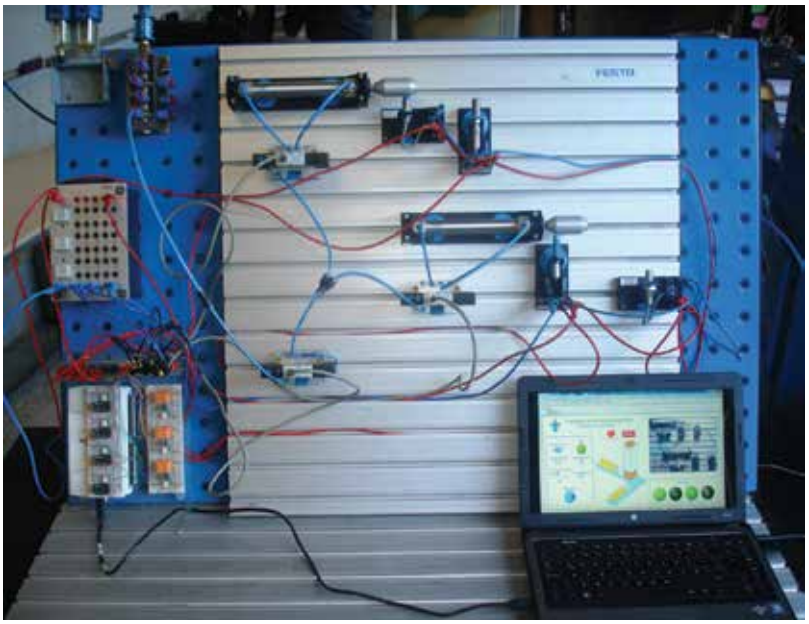

Figure 8. Experimental setup for the control of an electropneumatic system [16].

In the case of the remote laboratories described in this chapter, the virtual instruments developed for the implementation of the experiments in the web-based system were evaluated locally, at a first stage, and they worked as expected performing every required task. After this validation stage of the virtual environments, the experiments were performed over the Internet by a sample of students of the main campus in order to evaluate the remote operation of the systems. This evaluation stage was conducted in two different ways: first, by one student at a time and then simultaneously from different sites.

Students expressed their experiences when using this remote approach for experimental training and positive feedback was obtained, showing how the remote operation of the proposed experiments is a useful learning tool not only for practical experimentation but also for teaching in the engineering fields. The students were particularly satisfied with the virtual interfaces since, as they suggested, they are easy to use and they also liked the simultaneous operation of the system as a teaching aid because, as they expressed, it is possible to remotely follow the experiments while the professor performs the demonstrations.

6. Conclusion and further work

A web-based system for the remote operation of laboratory experiments has been presented. Three mechatronics experiments for practical training on different engineering fields were implemented in the web-based system developed, and they can be remotely performed from anywhere and at any time. The experiments are as follows: the control of two three-phase

squirrel-cage induction motors, the manipulation of residential electrical circuits, and the control of an electropneumatic system. For the implementation of these remote laboratories, a virtual environment for each of the experiments was developed using LabVIEW® virtual instrumentation, and the required laboratory equipment for the execution of the experiments was connected at the laboratories, together with additional hardware for the interfacing between the virtual instruments in the computers and the components and equipment in the laboratories. The experiments can be monitored in real time over the Internet and through the computer interface.

Remote laboratories are highly beneficial for engineering education. Making the experiments available over the web, students have access to practical training without time or location restrictions. Remote operation of the laboratories makes sharing the available resources within different campuses of one university possible, and it also allows the establishment of collaborative schemes between institutions. In particular, engineering schools in developing countries find this approach highly advantageous because it permits the use of local resources, which are often very limited, making them accessible through the Internet to a much higher number of students located in other campuses. Furthermore, under the establishment of proper arrangements, engineering students in developing countries can obtain access to laboratories in first world institutions, improving so the global levels of education, in accordance with the Sustainable Development Goals of the United Nations [20]. In any case, remote sharing of engineering laboratories for training and teaching among different institutions will impact by increasing the practical activities of their students, and therefore improving their learning goals, even if there are many differences between systems and protocols in universities and restrictions from one country to another, as remarked in Ref. [21].

Another important strength of the system lies in its characteristic of being accessible by several students at the same time, allowing it to be used as a teaching aid for the demonstration in real time of the experiments. In this way, the knowledge and experience of an instructor can benefit not only the local students as it occurs in conventional laboratory schemes.

A management system for the registration and authentication of students to remotely access the experiments has been previously proposed [22], and it is expected to be implemented soon with each of the remote laboratories presented. This web access system will allow students to register and schedule the usage of the experiments in order to perform the experiments by their own.

As commented before, the evaluation of the web-based laboratories extends to several different areas, and it is still in progress. At a first stage, the system was evaluated by a sample of students of the main campus in order to obtain preliminary results, but the remote laboratories are expected to be implemented as complementary exercises for different undergraduate courses offered by the schools of engineering in different campuses. At the end of each term, a poll is conducted in order to obtain feedback to determine educational achievements and usefulness of the system so as to improve its performance.

Further work is focused on the implementation in the web-based system developed of new mechatronic experiments for engineering education aimed at different fields such as fluid

mechanics, electronics, and civil engineering, improving the proposed system by adding new characteristics such as audio communication and other features currently typical of the Internet of things.

Author details

Jesús García-Guzmán[1]*, Farah H. Villa-López[2], Jorge A. Vélez-Enríquez[1], Luz A. García-Mathey[1] and Alfredo Ramírez-Ramírez[1]

*Address all correspondence to: jesusgarcia@ieee.org

1 Universidad Veracruzana, Xalapa, México

2 University of Warwick, Coventry, United Kingdom

References

[1] Barrios A, Panche S, Duque M, Grisales VH, Prieto F, Villa JL, et al. A multi-user remote academic laboratory system. Computers & Education 2013;**62**:111–122. DOI: 10.1016/j.compedu.2012.10.011

[2] Cardoso A, Sousa V, Gil P. Demonstration of a remote control laboratory to support teaching in control engineering subjects. IFAC-Papers OnLine 2016;**49**(6):226–229. DOI: 10.1016/j.ifacol.2016.07.181

[3] Hernandez-Jayo U, Garcia-Zubia J. Remote measurement and instrumentation laboratory for training in real analog electronic experiments. Measurement 2016;**82**:123–134. DOI: 10.1016/j.measurement.2015.12.017

[4] Rojko A, Hercog D, Jerzernik K. E-training in mechatronics using innovative remote laboratory. Mathematics and Computers in Simulation 2011;**82**(3):508–516. DOI: 10.1016/j.matcom.2010.10.017

[5] Sumei D, Zhiying S, Ruiqing J. Web based fluid mechanics experimental system. In: Proceedings of the 2010 International Conference on Electrical and Control Engineering; 25–27 June 2010; Wuhan, China. IEEE. 2010:pp. 3134–3137. DOI: 10.1109/iCECE.2010.765

[6] Irmak E, Bayindir R, Colak I, Soysal M. A remote laboratory experiment for 4-quadrant control of a DC motor. Computer Applications in Engineering Education 2010;**19**(4): 747–758. DOI: 10.1002/cae.20361

[7] Carpeño A, Contreras D, López S, Ruiz M, Sanz D, de Arcas G, et al. 3D virtual world remote laboratory to assist in designing advanced user defined DAQ systems based on FlexRIO and EPICS. Fusion Engineering and Design 2016;**112**:1059–1062. DOI: 10.1016/j.fusengdes.2016.02.052

[8] Villar-Zafra A, Zarza-Sanchez S, Lazaro-Villa JA, Fernandez-Canti RM. Multiplatform virtual laboratory for engineering education. In: Proceedings of the 9th International Conference on Remote Engineering and Virtual Instrumentation; 4–6 July 2012; Bilbao, Spain. IEEE. 2012:pp. 1–6. DOI: 10.1109/REV.2012.6293127

[9] Khalil A, Hasna M, Benammar M, Chaabane M, Ben Amar C. Development of a remote lab for electrical engineering program. In: Proceedings of the 3rd IEEE International Conference on E-Learning in Industrial Electronics; 3–5 November 2009; Porto, Portugal. IEEE. 2009:pp. 40–44. DOI: 10.1109/ICELIE.2009.5413214

[10] Hasim N, Basar MF, Aras MS. Design and development of a water bath control system: A virtual laboratory environment. In: Proceedings of the IEEE Student Conference on Research and Development (SCOReD); 19–20 December 2011; Cyberjaya, Malaysia. IEEE. 2011:pp. 403–408. DOI: 10.1109/SCOReD.2011.6148773

[11] Prada MA, Fuertes JJ, Serafín A, García S, Domínguez M. Challenges and solutions in remote laboratories. Application to a remote laboratory of an electro-pneumatic classification cell. Computers & Education 2015;**85**:180–190. DOI: 10.1016/j.compedu.2015.03.004

[12] Favario L, Masala E. Work-in-progress: Integrating a remote laboratory system in an online learning environment. In: 2016 IEEE Global Engineering Education Conference (EDUCON); 10–13 April 2016; Abu Dhabi, United Arab Emirates. IEEE. 2016:pp. 988–991. DOI: 10.1109/EDUCON.2016.7474672

[13] García-Guzmán J, Villa-López FH, Silva-Del-Rosario FH, Ramírez-Ramírez A, Vélez-Enríquez J, Álvarez-Sánchez EJ. Virtual environment for remote access and automation of an AC motor in a web-based laboratory. Procedia Technology 2012;**3**:224–234. DOI: 10.1016/j.protcy.2012.03.024

[14] Garcia-Zubia J, Orduna P, Angulo I, Hernandez U, Dziabenko O, Lopez-Ipina D, et al. Application and user perceptions of using the WebLab-Deusto-PLD in technical education. In: First Global Online Laboratory Consortium Remote Laboratories Workshop (GOLC); 12 October 2011; Rapid City, South Dakota, United States of America. IEEE. 2011:pp. 1–6. DOI: 10.1109/GOLC.2011.6086780

[15] Corter JE, Nickerson JV, Esche SK, Chassapis C. Remote versus hands-on labs: A comparative study. In: Proceedings of the 34th Annual Frontiers in Education; 20–23 October 2004; Savannah, Georgia, United States of America. IEEE. 2004:pp. 17–21. DOI: 10.1109/FIE.2004.1408586

[16] Villa-López FH. Control de un sistema electroneumático mediante instrumentación virtual y su implementación en un laboratorio remoto [thesis]. Xalapa, Mexico: Universidad Veracruzana 2012. Available from: http://cdigital.uv.mx/handle/123456789/32742

[17] Zamora-García DA. Monitoreo y control remoto de motores eléctricos mediante instrumentación virtual [thesis]. Xalapa, Mexico: Universidad Veracruzana 2012. Available from: http://cdigital.uv.mx/handle/123456789/32750

[18] Fernández-Orea LA. Control automático y monitorización remota de instalaciones eléctricas residenciales mediante instrumentación virtual con Labview [thesis]. Xalapa, Mexico: Universidad Veracruzana 2012. Available from: http://cdigital.uv.mx/handle/123456789/32743

[19] German-Sallo Z, Grif HS, Gligor A. Functional suitability analysis of remote laboratories in engineering education. In: 9th International Conference Interdisciplinarity in Engineering; 8–9 October 2015, Tirgu-Mures, Romania: Elsevier. 2016: pp. 1165–1175. DOI: 10.1016/j.protcy.2016.01.164

[20] United Nations. Sustainable Development Goals [Internet] 2015. Available from: https://sustainabledevelopment.un.org/sdgs [Accessed: 2016-12-28]

[21] Orduña P, Bailey PH, DeLong K, López-de-Ipiña D, García-Zubia J. Towards federated interoperable bridges for sharing educational remote laboratories. Computers in Human Behavior 2014;**30**:389–395. DOI: 10.1016/j.chb.2013.04.029

[22] Guerrero-Gutiérrez LD, Vélez-Enríquez JA, García-Guzmán J, Ramírez-Ramírez A, Lasso-Bretón JD. Consideraciones para crear un sistema de autenticación web orientado al funcionamiento de un laboratorio remoto. In: Proceedings of Universidad 2012: 8vo Congreso Internacional de Educación Superior; 13–17 February 2012; La Habana, Cuba. Ministerio de Educación Superior y las universidades de la República de Cuba. 2012.

Chapter 4

Design and Implementation of Micromechatronic Systems: SMA Drive Polymer Microgripper

Ren-Jung Chang and Yu-Heng Lai

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/67266

Abstract

A micromechatronic gripper was designed, fabricated, and tested with the proposed control system. By following realization axioms, the microgripper system including a polyurethane (PU) gripper mechanism and shape memory alloy (SMA) actuator was designed and developed. The micromechatronic gripper system was realized with cross-sectional area of $(\pi/4) \times 500^2$ μm^2 for clean room operation. A synergetic operation of SMA actuator for driving microgripper mechanism was investigated in visual-based control. By incorporating with inverse Preisach compensator, an explicit self-tuning controller through Ziegler-Nichols criterion was selected for controlling the self-biased SMA actuator. The application of the gripper system for gripping and transporting a glass particle of 30 μm was tested.

Keywords: design, control, micromechatronic systems, polymer, gripper

1. Introduction

The area of mechatronics is continually evolving, from Yasakawa's definition in the 1970s, through intelligent mechatronics and optomechatronics in the 1980s, to teleoperation and micromechatronics in the 1990s [1]. In the beginning of the 1970s, mechatronics was viewed as the combination of "mecha" from mechanism and the "tronics" from electronics [2]. Along with the evolution of mechatronics, people in mechatronic engineering used a variety of definitions [1–3]. In 1991, the International Federation for the Theory of Machines and Mechanism (IFToMM) gave the definition of mechatronics as: Mechatronics is the synergetic

combination of precision mechanical engineering, electrical control, and systems thinking in the design of products and processes. This definition had been adopted within the European Economic Community (EEC), which was later incorporated and renamed as the European Community (EC). In the perspective of industrial systems, the constituent components of mechatronic system include mechanism, actuator, sensor, and controller [2, 3]. Thus, by extending the definition of mechatronics and applying it to downscaling mechatronic systems, one may characterize a micromechatronic system as a system consisting of microscale mechanism, actuator, sensor, and controller which are designed and operated synergistically to achieve microoperation [2–4].

Miniaturization of mechatronic system has become a challenging area of technology in robotic and biomedical industries. In a specific area of micromechatronics, a microgripper system has been developed to provide a manipulation tool for the macroworld operation of microworld object. Since the first silicon microgripper was developed and proposed by Kim et al. in 1990 [5], several prototype miniature grippers have been designed and fabricated. The research trend on microgrippers has evolved toward a micromechatronic system by including force sensing and servo control [6–11]. In the industries of information, material, and biomedical engineering, microgrippers that are capable of handling small objects have many important applications [6, 7, 12–15]. In exploiting the compliant design and materials used for gripper mechanism, the researches of microelectromechanical systems (MEMS) [16] and compliant machine [17, 18] have stimulated the interests on implementing compliant microgripper systems to achieve stringent technological requirements [12–15]. Microgripper mechanism can be fabricated by utilizing MEMS technology and/or conventional precision machining technology [4]. Regarding the materials used, silicon and metal materials were employed widely for compliant micromechanism. However, due to the advance of polymer technology, the trend in using polymer to fabricate microgripper has been progressively increasing recently [11]. Actually, the realization of microgripper by employing polymer is most important in considering its practical applications. A polymer microgripper is cost-effective, reliable, easily fabricated, and mostly, fit to biomedical manipulations.

The research and development of microgripper systems has attracted numerous mechatronic engineers for over 25 years. However, in reviewing the implemented microgripper systems [13, 19], it is noted that most prototypes were developed without treating them as technological products through mechatronic approach [3], not even to mention micromechatronic method. By recognizing a microgripper system as a micromechatronic system, instead of MEMS, it will allow for its differentiation as an identifiable class of engineering products which promotes the development of micromechatronic engineering.

In this chapter, a micromechatronic gripper system with mechanism made of polymer and driven by shape memory alloy (SMA) actuator is designed, fabricated, and tested. In the first part, implementation Axioms, which are originally employed for precision design, are proposed for the development of micromechatronic systems. System design and implementation guided by realization Axioms from conceptual design, preliminary design and manufacturing consideration, as well as detail design and manufacturing procedure is described. The next part of this chapter concerns the synergetic operation of the mechatronic gripper system. The

servo control of SMA driven microgripper is described. The applications of the microgripper system are tested. Finally, the mechatronic design and implementation of the microgripper system is concluded.

2. Realization axioms for micromechatronic systems

The realization axioms, which are originally proposed by Suh in system design [20], are extended for the development of a micromechatronic system. For effective realization of micromechatronic systems, the process is guided by two realization axioms in the mappings from functional requirements (FRs) to design parameters (DPs) and then from DPs to process variables (PVs). The two realization axioms are stated as follows [21, 22]:

Axiom 1. Functional independence

An optimal design of a micromechatronic system must maintain the independence of functional requirements of the mechatronic subsystems—mechanism, sensor, actuator, and controller.

Axiom 2. Information minimization

The best design of a micromechatronic system is a design of functionally independent subsystems with the minimum information content. Here, the information content is a measure of uncertainties in physical realization of the design specifications of a system and its subsystems.

As an example of micromechatronic systems, a microgripper system is to be realized under the guidance of realization axioms. By employing the axiom of functional independence, a microgripper system can be designed and realized with the merits of the independent module design, independent functional testing, and sufficient degrees of freedom in system implementation. With the axiom of information minimization in the processes of design, assembly, and manufacturing, a microgripper system can be realized, and it is expected to satisfy the stringent requirements such as micron accuracy, clean operation, and low cost. The synergetic operation of a mechatronic microgripper system is finally achieved by implementing optimal control software.

3. Conceptual design

3.1. Design objective and constraints

In the area of micromanipulation, there are numerous functional principles which can be applied for the manipulation of microobjects [13, 14, 19, 23, 24]. In the present design, the objective is to develop a reliable mechanical microgripper employed in biomedical industries for repetitive grasping and transporting microobjects. The requirements of no lubrication and no wear are essential for clean room operations. The objects are microhard particles with diameter around 20–40 μm. The operation is to be carried out in room temperature around

25°C. Under the constraints of limited working space, the cross-sectional size of microgripper system is to be less than $(\pi/4) \times 1^2$ mm^2.

3.2. Design mappings

A conceptual design on the micromanipulation system is described through design mappings between different domains. With the established design mappings, the optimal structure of the micromanipulation system can be implemented by following the realization axioms. The relationship between the highest *FR*s and the highest level *DP*s is first established. The highest *FR*s of the manipulation system is identified as three independent functions: *FR*1 = gripping and releasing of microparticle, *FR*2 = carrying gripper and particle in microoperation, and *FR*3 = acquiring working states in micromanipulation. The corresponding *DP*s of the micromanipulation system is identified as *DP*1 = microgripper system, *DP*2 = working stages, and *DP*3 = visual system. The design mapping between the *FR*s and *DP*s in the first level can be formulated as Eq. (1):

$$\begin{bmatrix} FR1 \\ FR2 \\ FR3 \end{bmatrix} = \begin{bmatrix} a_{11} & 0 & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} DP1 \\ DP2 \\ DP3 \end{bmatrix}, \tag{1}$$

where the matrix by $[a_{ij}]$ is to be characterized in the realization of the micromanipulation system. The mapping between *FR*s and *DP*s in Eq. (1) satisfies a decoupled module design of microgripper system, working stages, and visual system. By following the axiom of functional independence, the design mapping between the *FR*s and *DP*s can be realized by utilizing mechanism, sensor, actuator, and controller. For the development of a micromanipulation system, the design procedure can be obtained from Eq. (1). The mapping between *FR*s and *DP*s in Eq. (1) can be realized by starting from a_{11}. From the mapping by a_{11}, the microgripper system will be first realized. With the microgripper system and a_{21}, the a_{22} is obtained by realizing working stages to satisfy the functional requirement. With the mapping set up by a_{11}, a_{21}, a_{22}, a_{31}, and a_{32}, the a_{33} is obtained finally by realizing visual system to satisfy the functional requirement. As a result, the design procedure is given by the sequence: $FR1 \rightarrow DP1 \rightarrow FR2 \rightarrow DP2 \rightarrow FR3 \rightarrow DP3$.

The microgripper system is the first subsystem to be realized for gripping and releasing microobject. The most essential hardware of microgripper system consists of microgripper mechanism (*DP*1.1) and microactuator (*DP*1.2). By following the Axiom 1, the microgripper mechanism and microactuator will be realized independently. Regarding the gripping and releasing function, the operation is realized by an end effector through gripper mechanism. The mapping between the execution of actuation to output motion (*FR*1.1) and the correspondent design parameters of microgripper mechanism (*DP*1.1) is further decoupled. By identifying *FR*1.1.1 = open/close gripper jaws, *FR*1.1.2 = fit microparticle, and *FR*1.1.3 = provide stable gripping and holding force as well as the corresponding *DP*1.1.1 = input to output mechanism, *DP*1.1.2 = openings and geometrical shape of gripper jaws, and *DP*1.1.3 = gripping point and contact surface, the mapping yields:

http://dx.doi.org/10.5772/67266

$$\begin{bmatrix} FR1.1.1 \\ FR1.1.2 \\ FR1.1.3 \end{bmatrix} = \begin{bmatrix} b_{11} & 0 & 0 \\ b_{21} & b_{22} & 0 \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} DP1.1.1 \\ DP1.1.2 \\ DP1.1.3 \end{bmatrix}. \tag{2}$$

The $[b_{ij}]$ in Eq. (2) is to be characterized by the realization of the conceptual design of microgripper mechanism as illustrated in **Figure 1**. Eq. (2) reveals that the achievement of stable gripping and holding object is relied on the design parameters: (1) input to output mechanism, (2) openings and geometrical shape of gripper jaws, and (3) gripping point and contact surface. In the realization of microgripper mechanism, the *DP*1.1.1 should be first implemented.

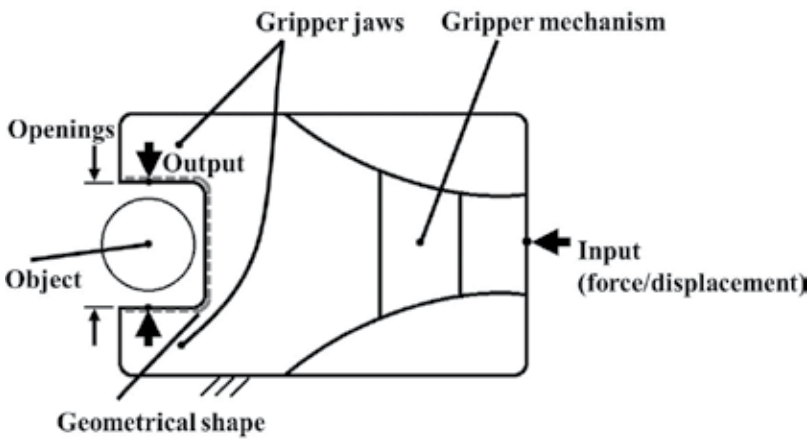

Figure 1. Conceptual design of microgripper mechanism.

From the requirement of design objective, the constraints of *DP*s in the design of microgripper system can be identified as: $DP1–C_1$ = clean operation, $DP1–C_2$ = accurate dimension, $DP1–C_3$ = micron repetitive operations, and $DP1–C_4$ = microgeometrical size. In the process domain, the corresponding *PV*s are stated as: $PV1–C_1$ = material selection, $PV1–C_2$ = fabrication method, $PV1–C_3$ = components configuration, and $PV1–C_4$ = assembly works. The functional mapping between the constraints of *DP*s and *PV*s can be formulated as Eq. (3):

$$\begin{bmatrix} DP1–C_1 \\ DP1–C_2 \\ DP1–C_3 \\ DP1–C_4 \end{bmatrix} = [c_{ij}] \begin{bmatrix} PV1–C_1 \\ PV1–C_2 \\ PV1–C_3 \\ PV1–C_4 \end{bmatrix}. \tag{3}$$

The $[c_{ij}]$ in Eq. (3) in general is not independent and the constraints are to be considered in the physical realization of microgripper mechanism and microactuator. In the process of realizing microgripper mechanism (*DP*1.1) or microactuator (*DP*1.2), the material selection, fabrication method, components configuration, and assembly works are crossly related. In the prototype development, the constraints on a microgripper mechanism are applied to its subsystems of end effector, transmission mechanism, and suspension frame. The constraints on a microactuator are

applied to its subsystems of actuating units, transmission mechanism, and structural frame. The constraints of components configuration and assembly works are applied to both microgripper mechanism and microactuator in the final assembly.

According to Eqs. (2) and (3), further detailed considerations in the conceptual design of the gripper mechanism are described. The Axiom 2 is employed for the design of a microgripper mechanism. For the minimization of information content, a mechanism is to be designed with simple configuration, symmetry, low number of links, and easy assembly. In considering the microgripper mechanism to be operated in a clean room, a one-piece compliant gripper, without assembly works, is selected and designed to provide accurate tip motion and proper gripping force. Since the microgripper mechanism needs to be reliable in micron-repetitive operations, an elastic material is selected to avoid fatigue in operations [18]. For providing a stable gripping of hard object, elastic contact gripping surfaces are used. Thus, without employing difficult surface treatment on microscale gripper surface, polymer elastomer will be considered for fabricating the gripper mechanism. For achieving the accurate and precise fabrication of microgripper mechanism, a lithography method through mask will be selected.

For accurate driving a microgripper, a microactuator which can provide micron operation with high actuating force is to be realized. At first, it is observed that SMA actuator is relatively small compared with other type of actuators under the same driving energy and output stroke [25]. If one considers the requirements to fit different applications, it is noted that the operational characteristics as well as geometrical shape and size of SMA can be conveniently designed and fabricated. A concern in fabricating SMA actuator is the constraint of geometrical size. Since a biased spring is required to overcome martensite twinning for the recovery of the initial shape in a fast two-way operation, the assembly works of SMA and biased spring make it difficult or even impossible to achieve microsize. Thus, by following the Axiom 2 and under the guidance of Eq. (3) in the realization of SMA actuator, an innovative design of SMA actuator without encountering the assembly issue is to be developed.

In order to focus on the present implementation of microgripper system, an industrial micromanipulation platform will be selected for realizing the working stages (*DP*2) and visual system (*DP*3). The selection of micromanipulation platform is guided by the Axioms 1 and 2. The visual system to be selected is to consider the specifications of constituent components: image card, image processing algorithm, microscope, illumination light, and CCD subsystem. In the selection of working stages for installing the microgripper system, one needs to consider working stroke, accuracy, speed, and degrees of freedom.

For the synergetic operation of the micromanipulation system, the control system consists of hardware and software. The electrical control hardware includes signal conversion, signal detection, impedance matching, and power amplification. The control hardware can be implemented by employing analogue and/or digital circuit elements. The controller includes cascade compensator, feed forward, and feedback control. For the operation of contact avoidance, gripping, releasing, and holding, the control signal can be implemented by employing software and running on a personal computer.

4. Preliminary design and manufacturing consideration

4.1. Gripper mechanism

Polymer elastomer is a viscoelastic material with the nonlinear and time-varying stress-strain behaviors: precondition, creep deformation, stress relaxation, and hysteresis [26]. These behaviors shown in a compliant mechanism depend on the amount of the distributed material which mainly contributes to deflected compliant motion. If the distributed material is lumped in the compliant joint, the amount of the deformed material will be the minimum and consequently, the nonlinear and time-varying effect will reduce. Thus, when the compliant-joint mechanism is operated under temperature and humidity-controlled environment, the small-deflected motion can be properly described through a linear time-invariant system which facilitates accurate motion control. As a result, a lumped-compliant-joint instead of distributed-compliant gripper mechanism is selected for achieving simplified model and accurate motion control. In designing a gripper with lumped-compliant joints, at first, one needs to select a topological linkages model for the kinematic structure of mechanism. Then the correspondent compliant structure will be transformed into a pseudo linkages model (PLM) with equivalent lumped springs [21, 26]. Finally, the compliant mechanism of the PLM is synthesized under the constraints of kinematics, material, and fabrication. In the preliminary consideration, it is expected to scale down the mechanism of a mesoscale compliant gripper which had been implemented [21]. The mesoscale compliant gripper, in which the topological structure was selected from the existing types of conventional mechanism, was a design with six links and six joints and manufactured by employing elastomer. When the gripper mechanism is scaled down to microsize; however, the stiffness of compliant joints in polymer gripper will greatly decrease. As a result, the downscaling microgripper with low input-output stiffness will cause high positioning errors in gripping operations.

For the improvement of the input-output stiffness, the modification or reconstruction of PLM is required in downscaling the mesoscale polymer gripper. It is noted that the compliant gripper is a structure with compliance lumped in its joints. The compliant joint actually will deform in several degrees of freedom when it is subjected to complex load in operations. Thus, the planner motion of compliant joint in a gripper will have both angular and linear deflections when external force and/or moment are applied. The existence of deflection other than angular deflected motion is most severe when the compliant joint is manufactured by utilizing elastomer. Considering the deflections of a right circular joint with radius r, minimum width of joint t, and thickness h as illustrated in **Figure 2**, the formulas of Paros and Weisbord [27] can be utilized for deflection analysis. For a right circular joint with $\frac{r}{t} \geq 2$ and with linear elastic material properties: Young's modulus E, Shear modulus G, and Poisson's ratio ν, the angular and shear stiffness of the joint can be expressed, respectively, as

$$K_\theta = \frac{M_z}{\Delta\theta} \approx \frac{2Ebt^{2.5}}{9\pi r^{0.5}}, \tag{4}$$

$$K_s = \frac{F_y}{\Delta s} \approx \frac{Gb}{\left(-2.57 + \pi(r/t)^{0.5}\right)}. \tag{5}$$

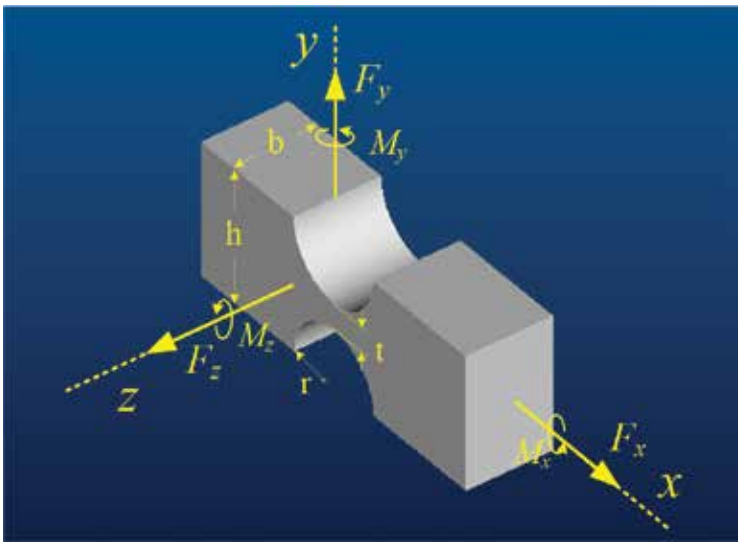

Figure 2. Geometrical dimensions and loadings of a right circular joint.

From Eqs. (4) and (5), it is observed that the relative magnitude between angular and shear deformations of a compliant joint mainly depends on the *r*/*t* ratio and material properties in design. If the geometrical size is fixed, the joint stiffness by Eqs. (4) and (5) will only depend on the material properties. Since the shear modulus of elastomer is relatively small, the deformation in the compliant joint by Eq. (5) needs to be included in a PLM model to account for the degrees of freedom in sliding. The sliding motion in the compliant joint will reduce the input-output stiffness of the downscaling mesoscale polymer gripper. In order to increase the input-output stiffness, a direct solution is to reduce the numbers of compliant joints in the gripper mechanism. By reducing the degrees of freedom of the mesoscale gripper in downscaling through eliminating two symmetric outer joints in the gripper, an equivalent PLM with one degree of freedom in kinematics is finally constructed as shown in **Figure 3**. For the compliant gripper mechanism, the PLM is identified as a mechanism with six linkages, four joints, and three sliders.

4.2. SMA actuator

An innovative SMA actuator to overcome the assembly issue is to be developed. The innovative design is to consider a self-biased SMA (SB-SMA) actuator without employing biased spring. A SB-SMA actuator can be implemented to include the effect of biased-spring in the fabrication of SMA microactuator through introducing prestrain in the SMA material. A novel SB-SMA actuator has been developed and employed for actuating microgripper in gripping and assembly applications [12]. The advantage of heating prestrained SMA wire in closing operation was investigated recently [28]. The implementation of the SB-SMA actuator will be described in the detail design and manufacturing procedure.

http://dx.doi.org/10.5772/67266

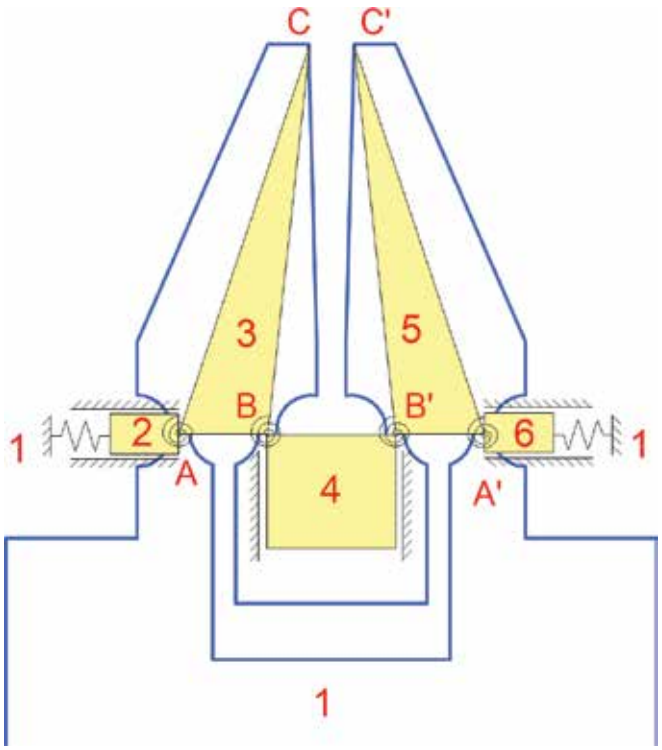

Figure 3. Preliminary design of microgripper mechanism and its PLM.

5. Detail design and manufacturing procedure

5.1. Gripper mechanism

In this phase, it is to synthesize an optimal shape and size of the compliant gripper mechanism. Under the design constraint of 1 mm width for the microgripper mechanism, an optimal PLM of gripper is obtained for the objective: amplifying the input displacement to output displacement and achieving the operation of parallel gripping.

The microgripper mechanism and its PLM are shown in **Figure 3**. In **Figure 3**, the contour line shows a geometrical shape of the gripper mechanism and its structural frame. The PLM is modeled as a six-linkage mechanism for providing one degree of freedom in input-output motion. When an actuating force F as input is applied, the actuator will drive link 4 to produce output displacement. Due to the constraints of structural frame 1, both links 3 and 5 will rotate and translate to cause the gripping operation by tips C and C'.

Regarding the PLM, the motion kinematics will be analyzed. In the following derivation, the assumption of small deformation is used. In the kinematic analysis, the two gripper arms are assumed to be driven simultaneously by a vertical sliding mechanism in gripping operation. In considering the output at gripping point E, which is closing to C, the kinematic motion of the

left gripper arm of PLM is depicted as shown in **Figure 4**. From **Figure 3** and **4**, the horizontal and vertical displacement gains can be derived as the ratio between the gripping point displacement Δx, Δy, and input displacement Δi of slider 4, respectively, as

$$G_x = \frac{\Delta x}{\Delta i} = \frac{L_2 \cos\beta}{L_1}, \tag{6}$$

and

$$G_y = \frac{\Delta y}{\Delta i} = \frac{L_2 \sin\beta}{L_1}, \tag{7}$$

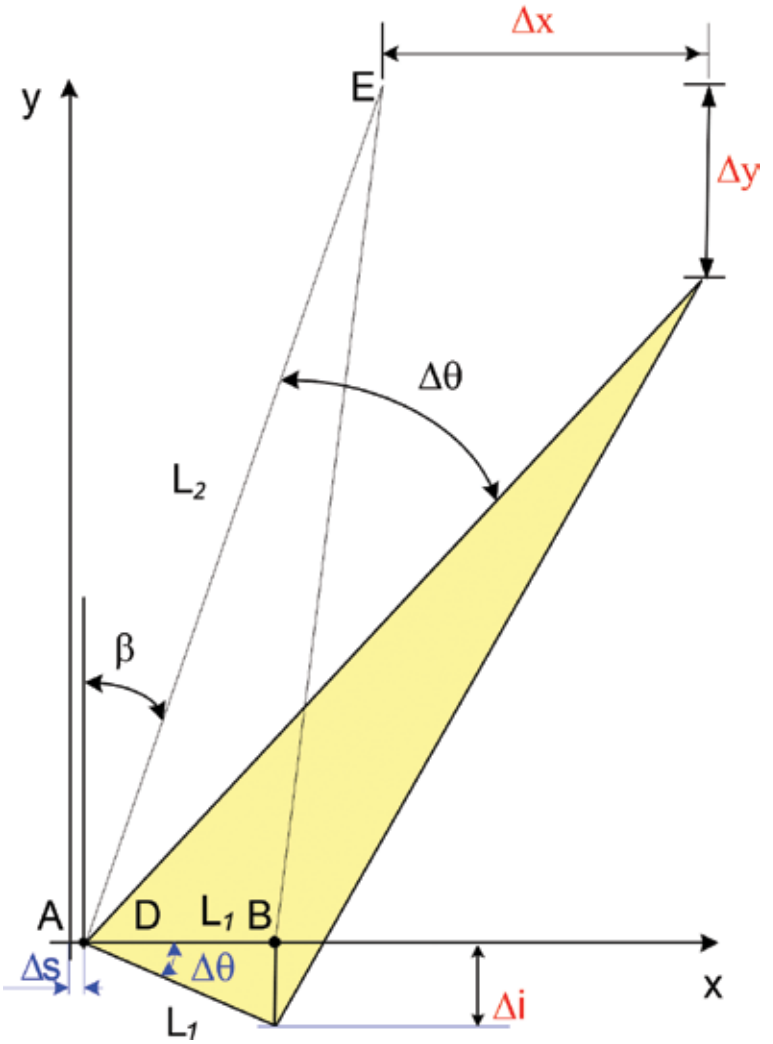

Figure 4. Kinematic motion relation of PLM in **Figure 3.**

where $L_1 = \overline{AB}$, $L_2 = \overline{AE}$, and β is the angle between a vertical line and $\overline{AE}$. For achieving parallel gripping, it is expected that $G_x \gg G_y$.The horizontal displacement of gripper tip can be expressed:

http://dx.doi.org/10.5772/67266

$$\Delta x = G_x L_1 \sin\Delta\theta. \tag{8}$$

In the synthesis of compliant gripper for accurate operation, it is noted that finite element analysis, kinematic PLM, and experimental test needs to be utilized closely together [26]. The detail geometry and size of the compliant microgripper is finally determined under the constraints of manufacturing technologies, stiffness of compliant joint, and availability of polymer material. The material of thermoplastic polyurethane (PU) film is selected from local industry. The material properties are given in **Table 1**. The r/t of the designed compliant joint is about 2.0. The size of microgripper mechanism was designed as 937 μm × 477 μm. The kinematic design gave G_x = 4.1, G_y = 1.5 by employing ANSYS analysis. The microgripper was manufactured by utilizing Excimer Laser, Exitech 2000, with mask projection through 10x size reduction by an optical lens. The final product of PU gripper mechanism was shown in **Figure 5**.

PU film	Thickness	100 μm
	Young's modulus	76.4 MPa
	Poisson's ratio	0.47
	Melting point	200°C
SMA wire	Alloy	NiTi
	Austenite starting temperature	70°C
	Diameter	38 μm
	Resistance	0.89Ω/mm

Table 1. Materials for designing gripper mechanism and actuator (@ temperature 24–26°C, humidity 40–44%).

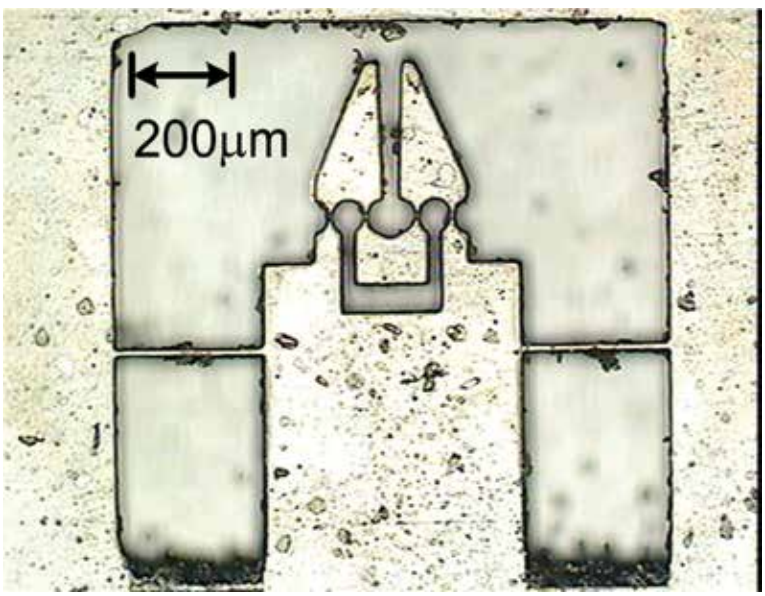

Figure 5. Final product of microgripper mechanism.

5.2. SB-SMA actuator

For manufacturing SB-SMA actuator, a roll of SMA wire was first obtained from Dynalloy, Inc. The material properties are given in **Table 1**. The SMA wire was cut to provide a 2 mm working length plus margin length on two ends for wiring. The wiring of the SMA actuator with copper wire was realized by utilizing silver glue. Two nonconductive glass plates of 1 μm thickness were used for the frame. The SMA wire was finally bent into an arc shape and glued together for introducing prestrain in the wire as shown in **Figure 6**. The heating and cooling of the SB-SMA in operation was undertaken, respectively through electrical resistance and still air. The still air was maintained at 24–26°C. In experimental test, the tip position of SB-SMA actuator was measured by utilizing a microscopic computer-vision system through correlation method [11]. The resolution of the image system was adjusted and calibrated to give 1.26 μm/pixel. A correlation method with further data processing was employed for finding the tip position of SB-SMA. For the microactuator, it provided 5 μm stroke with input 50 mA under 3 V.

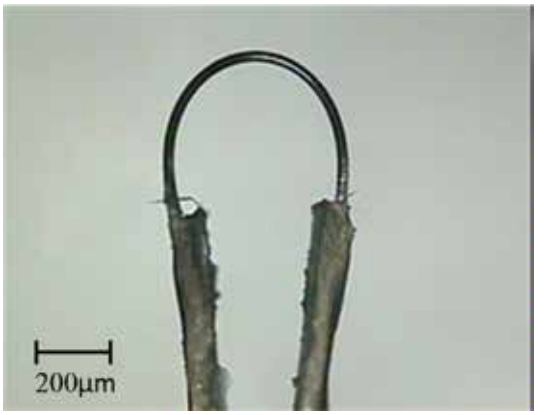

Figure 6. Final product of arc-shape SB-SMA actuator.

5.3. Assembly and test of microgripper system

The gripper system was assembled by the functional independent gripper mechanism and actuator. The assembled configuration of the planner gripper mechanism and actuator was to be perpendicular. The assembly procedure was to insert the SB-SMA actuator to the actuated point, then rotated it to the base, and finally did fine tuning and glued together [12]. For assuring a tight contact between the actuator and gripper in operation, a mechatronic approach, instead of moving and adjusting contact point [11], by applying a small biased-current to SMA in operation was preferred. After assembly, the cross-sectional area of the microgripper system was about $(\pi/4) \times 500^2$ μm^2. The testing of microgripper system was undertaken by the experimental setup with working stages and visual system as given in the installation [11]. When an electric current of 70 mA at 3 V was applied, the gripper was almost fully closed. By employing the realization Axioms, a micromechatronic gripper system was effectively implemented to satisfy design constraints at less repetitively corrective processes.

6. Servo control of SB-SMA drive microgripper

6.1. Measurement of the first-order descending curves

The Preisach model of SMA is constructed as a model consisting of many nonideal relays connected in parallel, given weights, and summed for the hysteresis behaviour. The model between input *u(t)* and output *y(t)* is expressed as [25, 29]

$$y(t) = \iint_{\alpha \geq \beta} \mu(\alpha, \beta) \gamma_{\alpha\beta}\Big(u\ (t)\Big) d\alpha d\beta, \tag{9}$$

where $\gamma_{\alpha\beta}$ are hysteresis operators for nonideal relays, α and β are the increasing and decreasing thresholds, respectively, and $\mu(\alpha, \beta)$ are Preisach functions.

The Preisach model from input current to output displacement is implemented numerically. Along with the input history *u(t)*, a sweeping region is expressed in the Preisach plane. The output change along the descending branch from α_i to β_j is defined as

$$Y(\alpha_i, \beta_j) = y_{\alpha_i} - y_{\alpha_i \beta_j}, \tag{10}$$

where y_{α_i} is the output displacement at input current value of α_i and $y_{\alpha_i \beta_j}$ is the output displacement after the current has been decreased to β_j from its maximum value of α_i. By dividing the sweeping region for calculating the output into *n(t)* trapezoids along with $Y\ (\alpha_i, \beta_j)$, the output change along the descending branch from α_i to β_j can be calculated. The output displacement by Preisach hysteresis can be calculated by utilizing the first-order descending (FOD) surface. By expressing the area of each trapezoid in Preisach plane as a difference between two triangle areas, the output displacement is derived by adding and/or subtracting all $Y\ (\alpha_i, \beta_j)$ with the corresponding triangular areas of edge (α_i, β_j). In the consideration of increasing or decreasing input *u(t)*, the output displacement is derived:

For $\dot{u}(t) \leq 0$,

$$y(t) = \sum_{k=1}^{n(t)-1} [Y(\alpha_k, \beta_{k-1}) - Y(\alpha_k, \beta_k)] + \left[Y\Big(\alpha_{n(t)}, \beta_{n(t)-1}\Big) - Y\Big(\alpha_{n(t)}, u(t)\Big)\right], \tag{11}$$

For $\dot{u}(t) > 0$,

$$y(t) = \sum_{k=1}^{n(t)-1} [Y(\alpha_k, \beta_{k-1}) - Y(\alpha_k, \beta_k)] + Y\Big(u(t), \beta_{n(t)-1}\Big). \tag{12}$$

The detail description about the experimental collection of data for FOD curves is referred to [29]. From experimental tests and with data processing, a set of FOD curves was processed, and finally depicted in **Figure 7**.

6.2. Inverse Preisach compensation

For compensating the nonlinear SB-SMA actuator accurately, the strategy is to cascade an analytical inverse Preisach model. The inverse of Preisach model, that determines the current

resulting in a desired displacement, is derived from Eqs. (11) and (12). The feed forward inverse compensator u_{Ff} for the desired output displacement $y_d(t)$ is implemented through the inverse function of Y as Y^{-1} and given by [30]:

For $\dot{u}(t) \leq 0$,

$$u_{Ff}(t) = Y_{\beta}^{-1}[\alpha_{n(t)}, \sum_{k=1}^{n(t)-1}[Y(\alpha_k, \beta_{k-1}) - Y(\alpha_k, \beta_k)] + Y(\alpha_{n(t)}, \beta_{n(t)-1}) - y_d(t), \tag{13}$$

For $\dot{u}(t) > 0$,

$$u_{Ff}(t) = Y_{\alpha}^{-1}[y_d(t) - \sum_{k=1}^{n(t)-1}[Y(\alpha_k, \beta_{k-1}) - Y(\alpha_k, \beta_k)], \beta_{n(t)-1}]. \tag{14}$$

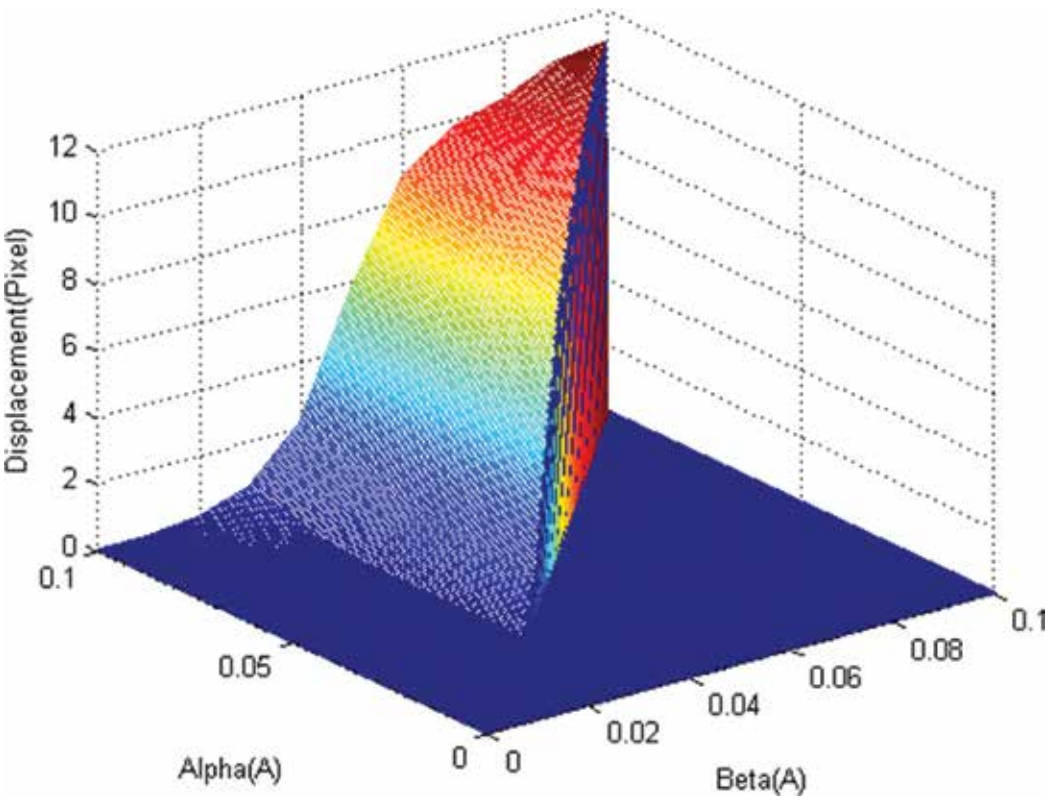

Figure 7. FOD surface of SMA.

In motion control, the SB-SMA microactuator was operated dynamically in the phase transformation during the heating and cooling processes. In experiment, the sampling frequency was 30 Hz. For investigating the dynamic behaviour of the SB-SMA actuator under analytical inverse Preisach compensation, sinusoidal input tests were undertaken. Experimental results revealed that the output displacement was highly fluctuated, especially when the SB-SMA actuator was operated to reverse its motion direction. For obtaining reliable measurement of actuator displacement by the correlation method, a low-pass filter was cascaded. Since the

bandwidth of SB-SMA actuator was less than 5 Hz, in considering the response speed of the SB-SMA actuator, a 3rd-order Butterworth filter $G_f(z)$ with 5 Hz cutoff frequency was implemented:

$$G_f(z) = \frac{0.049z^3 + 0.149z^2 + 0.149z^1 + 0.049}{z^3 - 1.162z^2 + 0.695z^1 - 0.138}. \tag{15}$$

The sinusoidal displacement response of the compensated SB-SMA under the two different input amplitudes was recorded, as shown in **Figure 8**. From the experimental results of sinusoidal input of 0.025 Hz, it was observed that the output response followed the input command after short-time transient but the error in amplitude increased as the input amplitude increased.

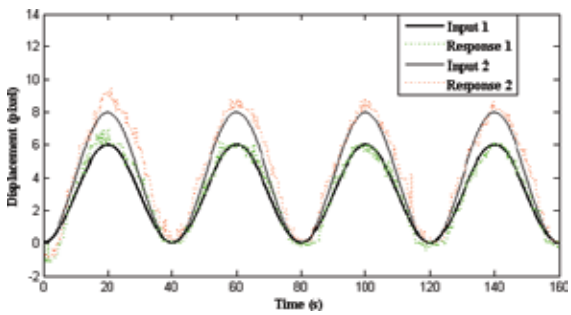

Figure 8. Sinusoidal test of SB-SMA actuator with inverse Preisach compensator.

6.3. Visual servo of microactuator

In controlling SMA actuator, the thermal and stress-dependent hysteresis behavior in phase transformation makes accurate control difficult. For the SB-SMA microactuator, the dynamic behaviour is even rather time-varying and consequently, it is difficult to achieve accurate offline compensation. A control strategy involving online identification is preferred for tracking the dynamics of SB-SMA actuator. An explicit self-tuning controller through the Ziegler-Nichols criterion [31] is selected for controlling the Preisach compensated SB-SMA actuator.

In an ideal Preisach compensated SB-SMA actuator, the dynamic model can be described as a linear time-varying system. By assuming that the SB-SMA actuator is a second-order system, the model can be described by

$$y(k) = \Theta^T(k)\Phi(k-1) + e(k). \tag{16}$$

The parameter vector and state vector, respectively in Eq. (16) is given as

$$\Phi(k-1) = [y(k-1), y(k-2), u(k-1), u(k-2)], \tag{17}$$

$$\Theta(k) = [a_1, a_2, b_1, b_2]. \tag{18}$$

By employing a recursive identification algorithm from Matlab, the parameter vector of system, Eq. (18), can be estimated to give $\hat{\Theta}(\mathrm{k}) = [\hat{a}_1, \hat{a}_2, \hat{b}_1, \hat{b}_2]$ [32]. Considering the SB-SMA

actuator is operated with proportional control gain k_p and in a unity feedback loop, the characteristic equation of the closed-loop system is obtained:

$$\Delta(z) = z^2 + bz + c = 0, \tag{19}$$

where $b = \hat{a}_1 + b_1 k_p, c = \hat{a}_2 + k_p \hat{b}_2$.

For a specific k_p, the feasible location of closed-loop poles to be located on unit circle can be classified according to the conditions: $b^2–4c > 0, b^2–4c = 0, b^2–4c < 0$. The three different cases are utilized to determine the critical gain k_{pu} and the associated critical oscillation frequency ω_k or period T_u for the feasible location of the closed-loop poles [31].

Based on the aforementioned three cases, the critical gain as well as critical oscillation period can be obtained and the control gain is derived by employing Ziegler-Nichols tuning rule. Regarding Ziegler-Nichols setting, $k_p = k_{pu}/2$ is for P control, $k_p = k_{pu}/2.2$, $k_i = 1.2k_{pu}/T_u$ is for PI control, and $k_p = 0.6k_{pu}$, $k_i = 2k_{pu}/T_u$, $k_d = k_{pu}T_u/8$ is for PID control. Thus, the PID controller with sampling time T_0 is implemented as

$$u(k) = k_p e(k) + k_i T_0 \sum_{i=1}^{k} e(i–1) + \left(\frac{k_d}{T_0}\right)[e(k)–e(k–1)]. \tag{20}$$

In practical control implementation, the control block diagram including inverse Preisach compensator, recursive least square estimator, PI tuning controller, visual position estimator, and low-pass filter for SB-SMA actuator is shown in **Figure 9**. In considering that the signal was highly fluctuated, only PI control in Eq. (20) was employed. In practice, the upper bound of k_i and k_p was limited to 2.5 to avoid excessive control input. The sampling frequency in the closed-loop system was 30 Hz. In each time step, the maximum value of the tuning gain was 0.3 to prevent the SMA vibration when system gain was changed. The low-pass filter was given by Eq. (15). The recursive least-square estimator was a forgetting factor algorithm with $\lambda = 0.98$ in Matlab function.

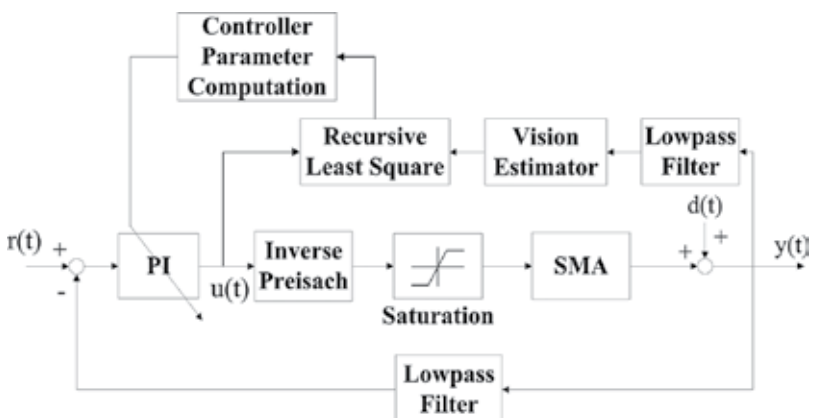

Figure 9. Control structure of self-tuning PI with inverse Preisach compensator.

http://dx.doi.org/10.5772/67266

For comparing the performance of the proposed controller, the fuzzy control schemes, which were usually utilized for system with uncertainty, with and without inverse Preisach compensator were also included. The block diagram of the fuzzy control schemes is shown in **Figure 10**. In the fuzzy control, the membership functions of error *e(t)*, error rate *de(t)*, and control input *u(t)*, were given as triangular functions. The expert knowledge by the fuzzy rules was given by the knowledge base as fuzzy sliding mode control. In fuzzy rules, the linguistic meaning of error *e* and error increment *de* are represented by utilizing symbol NB, NM, NS, ZE, PS, PM, and PB, respectively, for negative big, negative medium, negative small, zero, positive small, positive medium, and positive big. The knowledge base is shown in **Table 2**. For eliminating the steady-state error due to uncertain control bias, an integral control action with $k_i = 1.5$ was included. In the performance test, the system input was a multistep function. The experimental results of three control schemes were recorded, as shown in **Figure 11**. From the five consecutive steps, responses of **Figure 11**, the average rising time (0–100%), steady-state average absolute error, and average percentage overshoot were measured and calculated which are given in **Table 3**. According to the results listed in **Table 3**, the closed-loop performance of three different control schemes was compared. The results revealed that the inclusion of inverse Preisach compensator increased response accuracy but decreased response speed. The fuzzy control with I and without inverse Preisach gave the fastest response; however, it yielded the largest overshoot. The response speed by self-tuning PI with inverse Preisach was almost the same as that by fuzzy with I and inverse Preisach. Among the three control strategies, the self-tuning PI with inverse Preisach compensator achieved the best accuracy in both transient and stead-state responses.

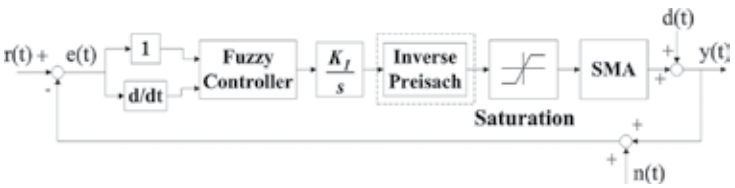

Figure 10. Structure of fuzzy control.

		Error						
		e						
Error rate		**NB**	**NM**	**NS**	**ZE**	**PS**	**PM**	**PB**
de	**PB**	ZE	PS	PM	PB	PB	PB	PB
	PM	NS	ZE	PS	PM	PB	PB	PB
	PS	NM	NS	ZE	PS	PM	PB	PB
	ZE	NB	NM	NS	ZE	PS	PM	PB
	NS	NB	NB	NM	NS	ZE	PS	PM
	NM	NB	NB	NB	NM	NS	ZE	PS
	NB	NB	NB	NB	NB	NM	NS	ZE

Table 2. Knowledge base of 49 fuzzy rules.

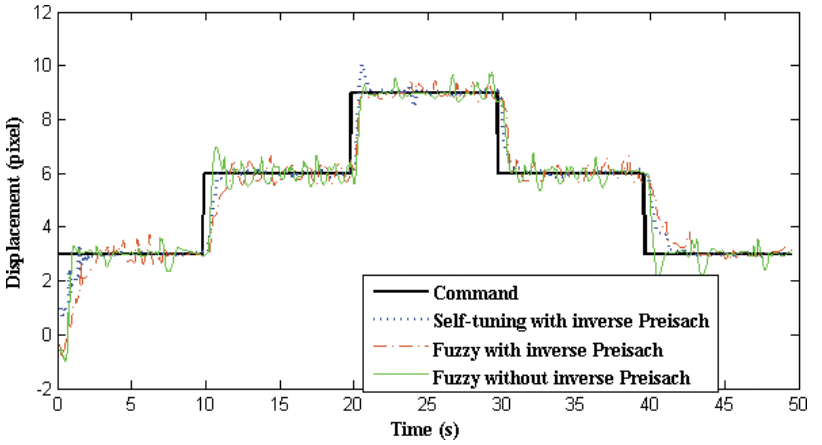

Figure 11. Multi-step input response by different control schemes.

Control method	Average rising time (s)	Steady-state average absolute error (pixel)	Average percentage overshoot (%)
Self-tuning PI with inverse Preisach	1.77	1.55	2.22
Fuzzy with I without inverse Preisach	0.64	5.72	11.26
Fuzzy with I and inverse Preisach	1.79	5.45	4.60

Table 3. System performance of three types of controller.

7. Microgripper applications

An experimental setup was installed for the testing of the gripping and transporting performance in controlling the microgripper system [11]. In the experimental tests, the measurement and testing was undertaken at room temperature 24–26°C and humidity 40–44%. The displacement measured by the image system was calibrated to give 1.26 μm/pixel. The present gripper was measured to give L_1 = 63 μm and G_x = 4.35. The G_x has an error of 5.75% in design.

In the gripping tests, a self-tuning PI with inverse Preisach control under visual servo was selected and employed for the closed-loop controller. The closed-loop block diagram is shown in **Figure 12**. In **Figure 12**, it was noted that a small backlash could be included to model the gap between the actuator driving point and the actuated point of microgripper. The backlash would be compensated by applying a small-biased current in operation.

The gripping test was undertaken by moving the microgripper to grip and transport a glass particle of 30 μm that was stick on the edge of a glass plate, as shown in **Figure 13**. After approaching the microparticle, the performance of controlling microgripper was to test its gripping operation. The input command to the gripper was to close 12 μm, i.e. 6 μm on each gripper jaw.

http://dx.doi.org/10.5772/67266

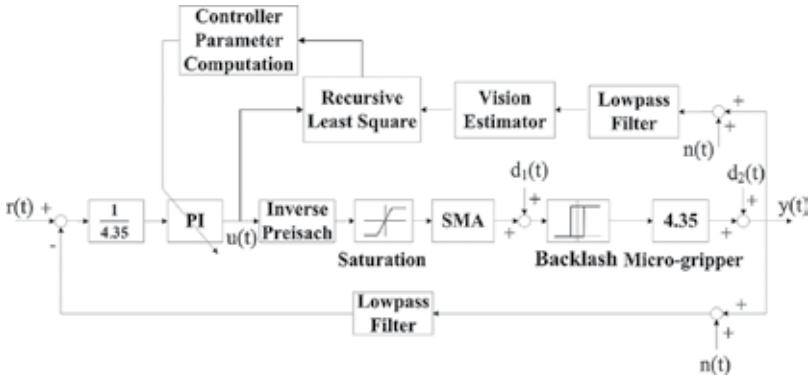

Figure 12. Block diagram of closed-loop control of SB-SMA actuated microgripper.

The displacement response of gripper jaw at one side was obtained by employing the kinematic relation of PLM together with measuring rotational angle of gripper jaw. As seen from **Figure 13(a)** and **Figure 13(b)** and employing regional scanning with edge fitting (RSEF) algorithm [7], the rotational angle $\Delta\theta$ was estimated and consequently, a one-sided horizontal displacement of gripper jaw was calculated by using Eq. (8).

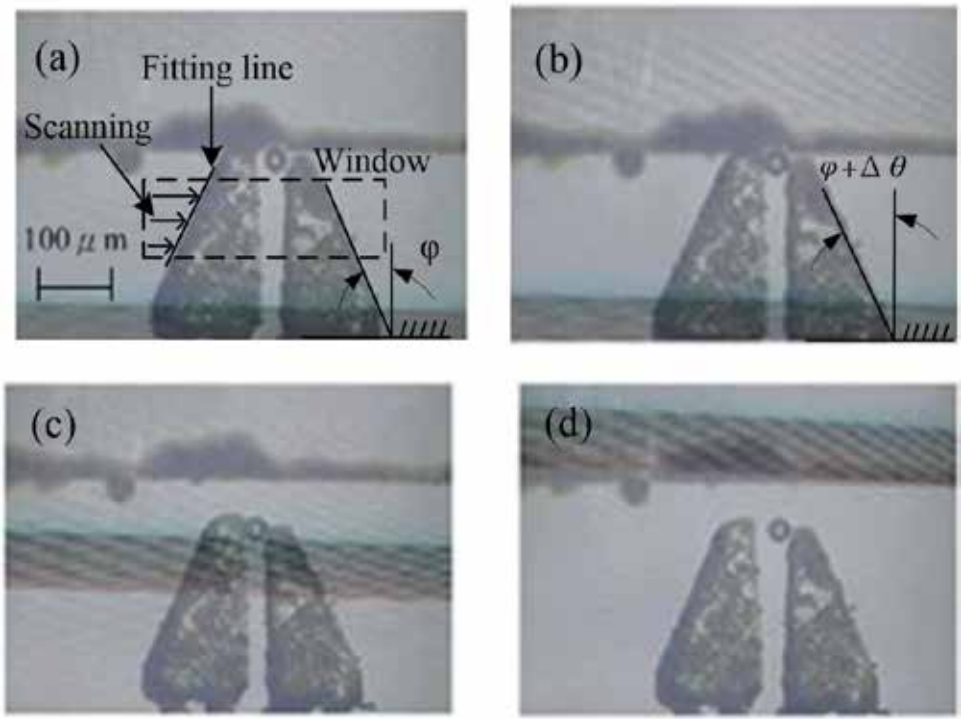

Figure 13. A microgripper is to approach (a), grasp (b), transport (c), and release (d) a 30 μm particle. The RSEF algorithm for measuring rotational angle is illustrated in (a) and (b).

For the input command of closing 12 μm, the command and gripping response of one-side jaw were recorded as shown in **Figure 14**. **Figure 14** revealed that the gripper jaw closed to almost

6 μm in 1.4 s and then decreased and oscillated until it was settled around 1 μm. In closing the gripper jaws, the first displacement peak was occurred when the gripper jaws touch and squeeze the microparticle. The contact and impact by microparticle caused highly fluctuated response in gripper jaw; however, the closed-loop controller effectively regulated the response to reach almost steady state at 4 s.

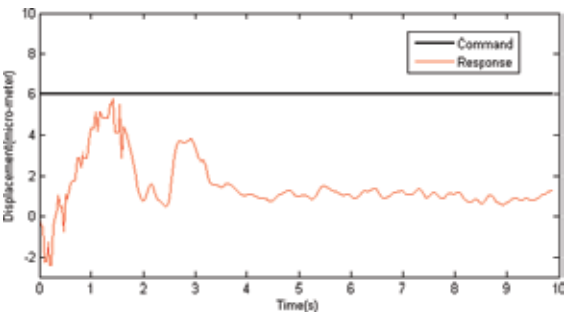

Figure 14. Response of one-side gripper jaw in gripping particle by self-tuning PI with inverse Preisach control.

The gripping operation was also utilized to estimate the particle size. The steady-state gripper closing without gripping particle was first tested and obtained. In steady state, the rotational angle was measured to give $\Delta\theta = 3.5°$ which leaded to a horizontal displacement of 16.73 μm. Consequently, without gripping particle, the horizontal displacement of gripper jaws was 33.46 μm. Considering the response in gripping particle as given by **Figure 14**, the steady-state displacement of gripper jaw was around 1 μm by one-side jaw, which was estimated as 2 μm by two-side jaws. Therefore, the size of microparticle was estimated to give 31.46 μm. The estimated error of particle size was 4.87%.

8. Conclusions

The realization axioms and design procedure are proposed for implementing micromechatronic systems. As an example of micromechatronic systems, a micromechatronic gripper was designed, fabricated, and tested with the visual-based control system. The microgripper system consisted of PU microgripper mechanism and SB-SMA actuator. By employing realization axioms, a micromechatronic gripper system was efficiently and effectively implemented to achieve high gripping performance at low cost expense. The micromechatronic gripper system was realizable with the cross-sectional area of $(\pi/4) \times 500^2$ μm^2 for clean room operation. A synergetic operation of SB-SMA actuator for driving microgripper mechanism was investigated in visual-based control. In the visual servo of SB-SMA actuator, a self-tuning PI with inverse Preisach compensator, as compared with fuzzy control schemes, gave the best accuracy in transient and steady-state responses. The micromechatronic gripper system was utilized to grasp and transport a glass particle of 30 μm. In the grasping test, the self-tuning PI with inverse Preisach compensator effectively regulated the closing response of gripper jaws in contact with glass particle. In addition, in holding the particle under visual servo, the size

of glass particle was estimated to give an error of 4.87%. The PU gripper system is cost effective to achieve high performance operation. For future sensitive biomedical field operations, the error by the prototype gripper needs to be further improved. In considering the manipulation of biological object, the gripper system needs to be tested in a liquid environment.

Acknowledgements

The authors would like to thank the Ministry of Science and Technology (MOST) for the partial support provided under contract No.105-2221-E-006-081.

Author details

Ren-Jung Chang* and Yu-Heng Lai

*Address all correspondence to: rjchang@mail.ncku.edu.tw

Department of Mechanical Engineering, National Cheng Kung University, Tainan, Taiwan, Republic of China

References

[1] Harshama F, Tomizuka M, Fukuda T. Mechatronics—what is it, why, and how? An editorial. IEEE/ASME Trans. Mechatronics. 1996;**1**(1):1–4.

[2] Kyura N, Oho H. Mechatronics—an industrial perspective. IEEE/ASME Trans. Mechatronics. 1996;**1**(1):10–15.

[3] Bishop R H, editor. The Mechatronics Handbook, 2nd ed. CRC Press. New York. 2007.

[4] Ishihara H, Arai F, Fukuda T. Micro mechatronics and micro actuators. IEEE/ASME Trans. Mechatronics. 1996;**1**(1):68–79. DOI: 10.1109/3516.491411

[5] Kim C J, Pisano A P, Muller R S, Lim M G. Design, fabrication, and testing of a polysilicon microgripper. Microstruct. Sens. Actuators. 1990;**19**:99–109.

[6] Beyeler F, Neild A, Oberti S, Bell D J, Sun Y. Monolithically fabricated microgripper with integrated force sensor for manipulating microobjects and biological cells aligned in an ultrasonic field. J. Microelectromech. Sys. 2007;**16**(1):7–15. DOI: 10.1109/JMEMS.2006.885853

[7] Chang R J, Lin C Y, Lin P S. Visual-based automation of peg-in-hole microassembly process. Trans. ASME, J. Manu. Sci. Eng. 2011;**133**(041015):1–12. DOI: 10.1115/1.4004497

[8] Lu Z, Chen P C Y, Lin W. Force sensing and control in micromanipulation. Part C: applications and reviews. IEEE Trans. Sys., Man, Cyber. 2006;**36**(6):713–724. DOI: 10.1109/TSMCC.2006.879385

[9] Kim K, Liu X, Zhang Y, Sun Y. Nanonewton force-controlled manipulation of biological cells using a monolithic MEMS microgripper with two-axis force feedback. J. Micromech. Microeng. 2008;**18:**055013.

[10] Zhang Y, Chen B, Liu X, Sun Y. Autonomous robotic pick-and-place of microobjects. IEEE Trans. Robot. 2010;**26**:200–207. DOI: 10.1109/TRO.2009.2034831

[11] Chang R J, Shiu C C, Cheng C Y. Self-biased-SMA drive PU microgripper with force sensing in visual servo. Int. J. Adv. Rob. Sys. 2013;**10**(280):1–12. DOI: 10.5772/56643

[12] Chang R J, Lin Y C, Shiu C C, Hsieh Y T. Development of SMA-actuated microgripper in micro assembly applications. The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON). 2007. DOI: 10.1109/IECON.2007.4460001

[13] Clévy C, Hubert A, Chaillet N. Micromanipulation and Micro-Assembly Systems. IEEE/RAS Int. Adv. Robot. Program. (IARP'06). 2006.

[14] Sanchez-Salmeron A J, Lopez-Tarazon R, Guzman-Diana R, Ricolfe-Viala C. Recent development in micro-handling systems for micro-manufacturing. J. Mater. Proc. Tech. 2005;**167**:499–507.

[15] Fatikow S, editor. Automated Nanohandling by Microrobots. London: Springer. 2008.

[16] Judy J W. Microelectromechanical systems (MEMS): fabrication, design and applications. Smart Mater. Struct. 2001;**10**:1115–1134.

[17] Midha A, Norton T W, Howell L L. On the nomenclature, classification, and abstractions of compliant mechanisms. Trans. ASME J. Mech. Design. 1994;**116**:270–279. DOI: 10.1115/1.2919358

[18] Howell L L. Compliant Mechanisms. New York: Wiley. 2011.

[19] Cecil, J, Bharathi Raj Kumar M B, Lu Y, Basallali V. A review of micro-devices assembly techniques and technology. Int. J. Adv. Manuf. Technol. 2015;**83**(9):1569–1581. DOI: 10.1007/s00170-015-7698-6

[20] Suh N P. The Principles of Design. New York: Oxford University Press. 1990.

[21] Chang R J, Wang H S, Wang Y L. Development of mesoscopic polymer gripper system guided by precision design axioms. Precision Eng. 2003;**7**:362–369.

[22] Chang R J, Chen C C. Using microgripper in development of automatic adhesive glue transferring and binding microassembly system. Engineering. 2010;**1.2**(1):1–11. DOI: 10.4236/eng.2010.21001

[23] Brussel H V, Peirs J, Reynaerts D, Delchambre A, Reinhart G, Roth N, Weck M, Zussman E. Assembly of microsystems. CIRP Annals. 2000;**49**:451–472. DOI: 10.1016/S0007-8506(07)63450-7

[24] Tichem M, Lang D, Karpuschewski B. A classification scheme for quantitative analysis of micro-grip principles. Assem. Autom. 2004;**24**(1):88–93.

[25] Kohl M. Shape Memory Microactuators. New York: Springer. 2004.

[26] Chang R J, Wang Y L. Integration method for input-output modeling and error analysis of four-bar polymer compliant micromachines. Trans. ASME J. Mech. Design. 1999;**121**:220–228. DOI: 10.1115/1.2829447

[27] Paros J M, Weisbord L. How to design flexure hinges. Mach. Des. 1965;**36**:151–156.

[28] Zhu P P, Cui Z W, Kesler M S, Newman J A, Manuel M V, Wright M C, Brinson L C. Characterization and modeling of three-dimensional self-healing shape memory alloy-reinforced metal-matrix composites. Mech. Mater. 2016;**103**:1–10.

[29] Gorbert R B. Control of hysteretic system with Preisach representation. Ph.D. Thesis. Canada: University of Waterloo. 1997.

[30] Mittal S, Menq C H. Hysteresis compensation in electromagnetic actuators through Preisach model inversion. IEEE/ASME Trans. Mechatronics. 2000;**5**(4):394–409. DOI: 10.1109/3516.891051

[31] Bobál V, Böhm J, Fessl J, Macháček J. Digital Self-tuning Controllers. Berlin: Springer. 2005.

[32] Åström K J, Wittenmark B. Adaptive Control, 2nd ed. New York: Addision-Wesley. 1995.

Chapter 5

Sensors and Digital Signal Conditioning in Mechatronic Systems

Dilşad Engin and Mustafa Engin

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/67986

Abstract

Essential part of a mechatronics system is the measurement system that senses the variations in the physical parameters, such as temperature, pressure, displacement, and so on, and converts it to voltage or current. The control of industrial processes and automated manufacturing systems requests accurate, moreover, linearized sensor measurements, where numerous sensors have nonlinear characteristics. In mechatronic systems, accurate measurement of the dynamic variables plays a vital role for the actuators to function properly. This chapter presents linearization methods and a measurement system in mechatronics consisting of temperature sensors and the signal-conditioning circuits, providing detailed information on design process of an embedded measurement and linearization system. This system uses a 32-bit microcontroller for thermocouple (T/C) cold junction compensation, amplification of low output voltage, then conversion to digital, and linearization of the type K thermocouple's output by software to output a desired signal. Piecewise and polynomial methods are used in linearization software, and the implemented embedded system for the linearization of a type K T/C is presented as a case study. The obtained results are compared to give an insight to the researchers who work on measurement systems in mechatronics.

Keywords: temperature sensors, sensor linearization, piecewise linearization, polynomial linearization, embedded system

1. Introduction

Mechatronic systems are composed of mechanical and electrical components, referred as "smart" systems because of the integration of sensors, actuators, and control systems. Essential part of a mechatronics system is the measurement system that senses the variations in the physical parameters, such as temperature, pressure, displacement, and so on, and converts it to an electrical quantity, viz. electric voltage or current. The measurement system consists of

mainly two parts: sensor and signal-conditioning circuit. A device that displays, records, or controls the measured variable utilizes the output of the measurement system.

In a measurement system, sensor detects the change in the physical parameters, but this measured variable is not usually in the form desired by the rest of the system. Signal-conditioning part may consist of sensor output amplification, analog-to-digital conversion (ADC), compensation, frequency-to-voltage conversion, and so on. The signal-conditioning interface is the part of the measurement system where the measured signal is converted to an analog or digital electrical signal that is required by the instrument.

The control of industrial processes and automated manufacturing systems requests accurate, moreover, linearized sensor measurements where numerous sensors have nonlinear characteristics. In mechatronic systems, accurate measurement of the dynamic variables plays a vital role for the actuators to function properly. An inaccurate measurement of the rotational position of a robot arm may result in a misplacement of an electronic part on a printed circuit board. The nonlinear voltage output of the temperature sensors embedded in the motor windings results in a lower temperature reading than the actual temperature, which may cause shutdown of the motor.

Emerging technologies in measurement systems have led to integrated signal-conditioning circuits within the sensors. Spreading the usage of microcontrollers or microprocessors in sensing technology has increasingly resulted in utilizing signal-processing functions embedded in the sensors, consequently accurate and linear signals are available at the outputs. For this reason, the appropriate selection of linearization technique is important while the processing time and memory usage of the processor must be limited for fast response and for maintaining the cost of the sensor reasonably priced. Linearization and calibration algorithm design with hardware and/or software is a broad research interest.

2. Sensors in mechatronics

The vital part of a mechatronics system is the sensing of physical parameters—either discrete or continuous. In Electrical Transducer Nomenclature and Terminology standard, a sensor is defined as "a device which provides a usable output in response to a specified measurand" [1]. Our focus will be on nonlinear sensors.

In mechatronic systems, one of the most utilized sensors is the capacitive-type sensor. Capacitance changes nonlinearly with the displacement of a diaphragm in a pressure sensor, linear or rotational motion of an object from the reference, or the displacement of vibration sensors. Linearization and calibration of this sensor has been an extensive research area [2].

The other utilized sensor is the temperature sensor, which measures the process temperature, monitor the temperature of the rotor windings, or in temperature compensation of sensors whose performance is affected by temperature changes. Temperature is the most measured and controlled dynamic variable in manufacturing and machine control. There are various types of these sensors, which use resistance change of metals (resistance temperature detector (RTD)) and

semiconductors (thermistors) and thermoelectric effect (thermocouples (T/Cs)) due to the change in temperature. Resistive temperature detectors and thermocouples are the most preferred ones due to high linearity of the RTDs and wide operating range of the T/Cs. Although highly nonlinear, thermistors are also used in temperature sensing due to their high sensitivity and low cost. Linearization of thermistors has found an extensive interest among researchers [3, 4].

Thermistor is a semiconductor-resistive temperature sensor made from metal oxides (negative temperature coefficient, NTC) and doped polycrystalline ceramic containing barium titanate and other compounds (positive temperature coefficient (PTC)). In NTC thermistors, the resistance decreases with the increase in temperature. NTC thermistors are widely used for a narrow range of −50 to 150°C.

The NTC thermistor resistance R_T is related to temperature by the empirically developed Steinhart-Hart equation:

$$\frac{1}{T} = A + B\ln R_T + C\ln(R_T)^3 \tag{1}$$

where T is in K, and A, B, and C are the thermistor coefficients given by the manufacturers, or you can calculate these values to obtain better accuracy using least-square method with the thermistor table data [3, 4].

RTDs are metal or metal alloy resistors that exhibit an increase in resistance with the increase in temperature. The operating temperature ranges from −200 to 850°C. They are characterized by a polynomial with respect to temperature as

$$R_T = R_0(1 + \alpha_1 T + \alpha_2 T^2 + \cdots + \alpha_n T^n) \tag{2}$$

where the curve-fitting coefficients are supplied for different types of RTDs.

Platinum RTDs are highly accurate (up to ±0.01% at 0°C), but they have relatively linear characteristic curve over large spans, but not exactly a straight line. The platinum RTD with a resistance value of 100 Ω at 0°C, called Pt100, has nearly linear characteristic curve as shown in **Figure 1** compared to the nonlinear 100-Ω NTC thermistor.

For platinum RTDs, according to IEC 751, the polynomial in Eq. (2) can be reduced to

$$R_T = R_0[1 + \alpha_1 T + \alpha_2 T^2 + \alpha_3(T - 100)T^3] \tag{3}$$

in which α_3 can be applied only when $T < 0$°C. The coefficients α_1, α_2, and α_3 for a standard sensor are specified in IEC751. For very high precision, researchers work on linearization methods for platinum RTDs [5, 6].

Thermocouple is a self-powered sensor where the ends of two different metals or metal alloys are weld bonded. The bead weld point is the hot junction, and the other point is the cold junction. A voltage in the order of millivolts is generated at this cold junction. This thermoelectric voltage is a function of the difference between the hot-junction and the cold-junction temperatures and also the composition of the metals as shown in **Figure 2**.

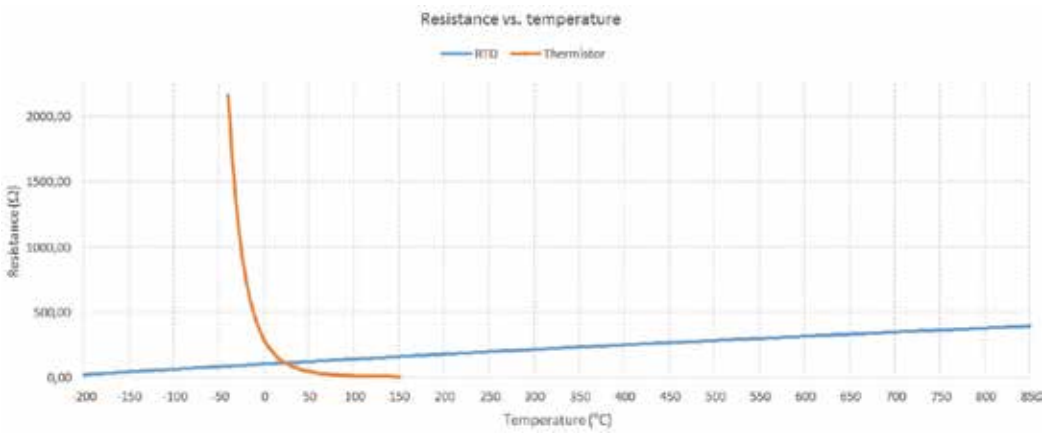

Figure 1. RTD characteristic curve, data from DIN IEC 751 temperature/resistance table compared to NTC thermistor.

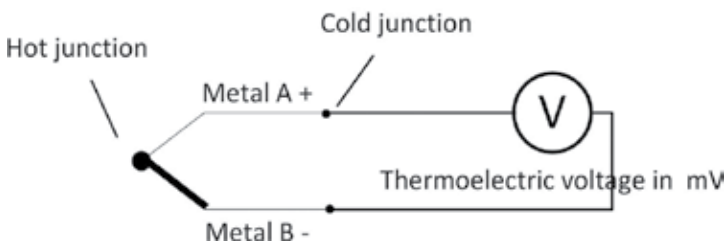

Figure 2. Thermoelectric voltage of a thermocouple.

NIST ITS-90 thermocouple database [7] presents this generated voltage for temperatures below 0°C as

$$E = \sum_{i=0}^{n} d_{i(t_{90})^i} \tag{4}$$

and for temperatures above 0°C as

$$E = \sum_{i=0}^{n} d_{i(t_{90})^i} + b_0 e^{b_1 (t_{90} - b_2)^2} \tag{5}$$

where the coefficients, b_0, b_1, b_2, and d_i, of reference equations provide the thermoelectric voltage, E, as a function of temperature, t_{90}, for the indicated temperature ranges, and $b_0 = 1.185976 \times 10^{-1}$, $b_1 = -1.183432 \times 10^{-4}$, and $b_2 = 126.9686$.

This generated voltage by T/C's cold junction can be obtained by measuring the junction temperature with a thermistor, or an IC temperature sensor fixed on an isothermal block. The measured voltage corresponds to the cold-junction temperature and it is utilized in computing

the hot-junction temperature. The voltage output of the cold-junction temperature can be fetched from the T/C reference tables according to IEC 584-1. The difference between this voltage and the measured voltage is the voltage generated by hot junction. Temperature value, which corresponds to the computed voltage, can also be fetched from the T/C temperature/mV output table. This process is the software compensation.

There are a variety of T/C types ranging from −270 to 2300°C. They are small in size, thus fast in response, and low in cost. Unfortunately, they have poor linearity and low sensitivity. The lead effect is so high that it must be compensated. Linearization, amplification of the low output, and compensation of the lead effect of temperature sensors can be carried out by analog circuit design [8, 9], by developed software embedded in a microcontroller [10, 11], or a circuit design where the sampled data stored in an SRAM is transferred to a computer for linearization process [12]. Anyway, temperature measurements with T/Cs are challenging, particularly when the temperature measurement is below 0°C.

We discuss the measurement system in mechatronics consisting of thermocouples as temperature sensors and the signal-conditioning circuits, providing detailed information on design process of an embedded measurement and linearization system. This system uses 32-bit microcontroller for T/C cold-junction compensation, analog-to-digital conversion, and T/C temperature sensor output linearization by software.

3. Methods for linearization

To overcome the poor linearity of T/Cs, linearization circuits are developed. Complicated analog electronic circuits are designed to cope with T/C's nonlinear and low output voltage problem [13, 14]. These circuits amplify the low-voltage output to a desired level and linearize the output to obtain the intended accuracy in the operating range of the system [6]. Therefore, in linearization with hardware precise assignation of the circuit elements is considerably important to achieve the essential accuracy.

Software linearization techniques are also preferred among researchers. Wei et al. first amplified the T/C output, converted it to digital by on-chip analog-to-digital converter of the microcontroller, and then linearized the output using least-squares method [14]. Sarma and Boruah amplified type K T/C output, then converted to digital by 12-bit ADC, and finally linearized the output with an eight-bit microcontroller using a piecewise polynomial of ninth degree [15]. Engin used an eight-bit microcontroller, its on-chip programmable gain amplifier and 24-bit ADC to amplify and digitize the measured type T T/C output, and built-in temperature sensor for compensation, finally linearized the output by first- and second-degree polynomials, and piecewise linear interpolation methods [11]. Some researchers utilized a T/C amplifier for amplification and cold-junction compensation and linearized the output by look-up table (LUT) embedded in the microcontroller [11, 16, 17]. Wang et al. used B-spline method for linearizing the output of nonlinear sensors [18]. The sensor linearization process comprises complex mathematical computations that an eight-bit microcontroller cannot achieve. Therefore, many researchers had rather realized calibration algorithm on a computer

through I/O interface cards. Danisman et al. initially amplified the T/C output with an instrumentation amplifier, then used an ADC for conversion to digital, and transmitted this digitized measurement to a computer where they applied artificial neural network (ANN) calibration algorithm by means of a virtual instrument [10]. Researchers who utilized low-cost microcontrollers for linearizing the sensor outputs limited the sensor's input range to a part of the full scale.

In this section, we give a brief description of the linearization methods. Mathematical models and implementation by software will be the main scope.

3.1. Least-squares regression method

A set of measured values from a sensor output needs to be fitted to a curve in order to obtain a mathematical representation of the sensor output. Linear least-squares regression is considerably the most used modeling method. This method utilizes linear algebra to determine the "best-fit" line for a data set by minimizing the sum of the squares of the vertical residuals of the data points to a modeling curve. The sum of the squares of the residuals is preferred because this warrants continuously differentiable residuals at every point contrary to the absolute error differentiation.

Our aim is to find the "best fit" particular to any finite linear combinations of the identified function instead of the best-fit line. Consequently, we can write a general equation including functions $f_1,\ldots,f_n$, and then find the coefficients $a_1,\ldots,a_n$ such that the linear combination

$$\hat{y} = a_1 f_1(x) + \ldots + a_n f_n(x) \tag{6}$$

is considered to be the best approximation to the data.

We have a set of n data points $\{(x_1, y_1),\ \ldots(x_n, y_n)\}$ with n residuals $\{(y_1 - \widehat{y_1}), \ldots(y_n - \widehat{y_n})\}$. If our model $\hat{y}$ is a good fit to the set of data points, then the mean should be small and the variance will be a measure of how the model fits the data. For this set of measured data, we define the error as the sum of the squares of the residuals as

$$E(a, b) = \sum_{i=1}^{n} (y_i - \widehat{y_i})^2 \tag{7}$$

We must determine the coefficients a and b, which minimize the error. Then, we should calculate the partial derivatives of the error such that

$$\frac{\partial E}{\partial a} = 0 \text{ and } \frac{\partial E}{\partial b} = 0 \tag{8}$$

For linear approximation to the data, we can write that our model is $\hat{y} = ax + b$, and if this is the best fit, then the residual must be $y - \hat{y} = 0$. An example of linear least-squares regression analysis for the given set of data in **Table 1** is implemented using Excel regression analysis, and the linear approximation plot is given in **Figure 3**.

http://dx.doi.org/10.5772/67986

x	0	1	2	2.5	3	3.5	4	4.75	5
y	2	1	0.5	1.125	1	1.125	0.75	0.5	0

Table 1. Example data for linear least-squares approximation.

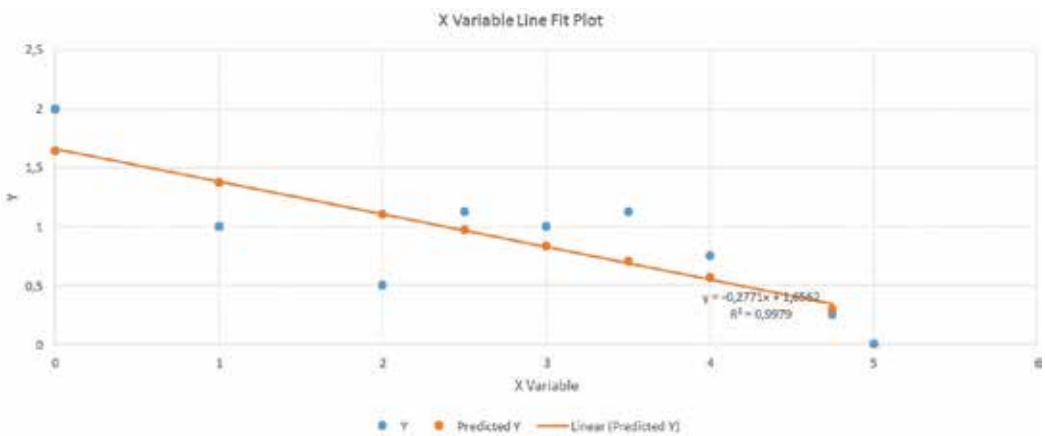

Figure 3. Linear least-squares regression plot for the given set of data.

Linear least-squares regression is the main instrument for process modeling since it is effective in finding a model that best fits, especially, a small set of data. Although there are sets of data that are better defined by nonlinear-coefficient functions, numerous practices in engineering can be described by linear models due to the fact that these processes are linear in nature or they can be approximated by a linear model within narrow ranges. In software-based sensor linearization, it provides minimum code size and consumes the lowest power.

On the other hand, for inherently nonlinear processes, it is more difficult to find a linear model to fit the set of measured data, particularly for wide range. Moreover, the computation time for linearization process run by a computer or a microcontroller will increase as the explanatory variables increase. The sensitivity to the outliers caused by improper measurements can also seriously deflect the "best-fit" line; therefore, model validation becomes critical to acquire accurate responses to the demands stimulating the construction of the model [19].

3.2. Piecewise polynomial interpolation

Curve fitting of a set of measured data from a sensor output can be accomplished by high-degree polynomial interpolants where we should abstain from equally spaced data points. This may result in a problem in producing accurate interpolant over a wide range. To overcome this difficulty is to partition the wide range into subintervals $\alpha = x_1 < x_2 < \cdots < x_n = \beta$, and then interpolate the function on each subinterval $[x_i, x_{i+1}]$ with a low-degree polynomial.

This is the piecewise polynomial interpolation idea. For linearization of the sensor outputs, we prefer linear approximations rather than polynomial fits.

Piecewise linear approximation is a technique of obtaining a function that fits a nonlinear objective function by adding binary or continuous variables and constraints to reformulate the original function [20]. The particular aim is to approximate a function of one variable in terms of sequential linear pieces. The successive data points connecting piecewise straight lines are called breakpoints.

Assume that a general nonlinear continuous function $f(x)$ of a single variable x, and x is within the interval $[\alpha, \beta]$. Call $y_i = f(x_i),\ i = 1 : n$. The breakpoints of (x) in general are $\alpha = a_1 < a_2 < \ldots < a_n = \beta$, and **Figure 4** indicates the piecewise linearization of $f(x)$ for breakpoints $[a_1, \ldots, a_9]$. In the figure, the dots denote a measured data set. Usually, we use the measured data points as the breakpoints, $[\alpha, \beta] = [x_1, x_n]$. If we have a set of n data points $\{(x_1, y_1),\ \ldots(x_n, y_n)\}$, and connect these points with straight lines, then we obtain the graph of a piecewise linear function.

The significance in reformulating a consider that we have the piecewise as a linear function is to use a distinct variable for each segment. Consider that we have the piecewise linear function $f_k(x_k)$ in **Figure 4** that has eight line segments over the range of x_k values. We now have eight new variables $[x_{k1}, \ldots, x_{k8}]$ and in general form, set

$$x_k = \sum_{i=1}^{n-1} x_{ki} \tag{9}$$

For each segment, the slope is

$$s_{ki} = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}, \tag{10}$$

and the nonlinear function can be rewritten in terms of the slopes as

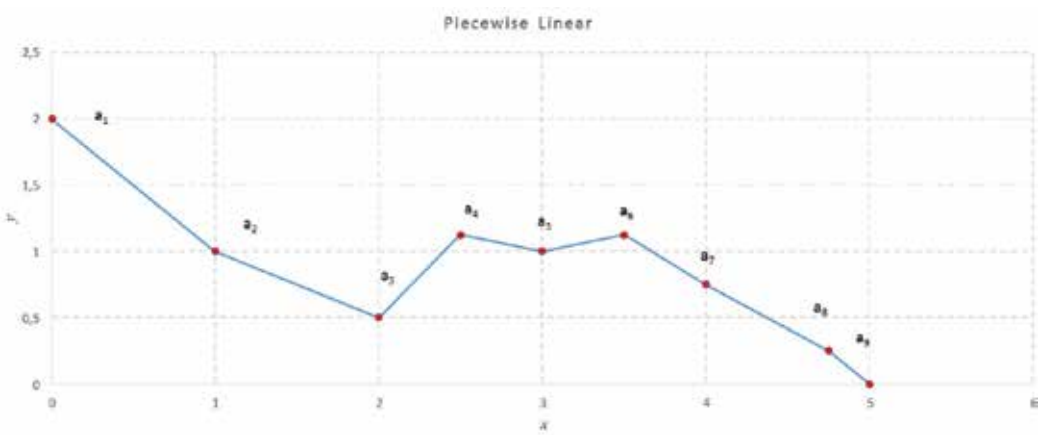

Figure 4. Piecewise linearization of the data set in **Table 1**.

$$f_k(x_k) = \sum_{i=1}^{n-1} s_{ki} x_{ki} \tag{11}$$

where $s_{k1} > s_{k2} > \cdots > s_{k(n-1)}$ condition should be satisfied for $f_k(x_k)$ to be concave in order to accurately linearize the nonlinear function $f(x)$.

Unfortunately, there are piecewise linear functions that cannot be reformulated by linear programming as depicted above. Therefore, the nonlinear function (x) can be approximated by nonlinear programming algorithms.

You may find the piecewise linearization procedure in the nonlinear programming textbooks [21, 22], where a nonlinear function $f(x)$ can be approximately linearized over the interval $[\alpha, \beta]$ as

$$L\Big(f(x)\Big) = \sum_{i=1}^{n} f(a_i) w_i, \tag{12}$$

$$x = \sum_{i=1}^{n} a_i w_i \tag{13}$$

The weights w_i are non-negative associated with the ith breakpoint as

$$\sum_{i=1}^{n} w_i = 1,\ w_i \geq 0,\ i = 1, 2, \ldots, n \tag{14}$$

This mixed integer programming ensures the validity of the approximation by the conditions in [22].

The accuracy of the linear approximation significantly relates to the number of breakpoints where a greater number of breakpoints result in more accurate approximation. Unavoidably, adding a number of breakpoints results in a considerable growth in the memory size for storing these values. When linearization of the sensor outputs by a microcontroller is accomplished, the number of breakpoints should be considered as well as the accuracy.

3.3. Piecewise linear approximation with look-up table

When polynomial, hyperbolically tangent or exponential functions are linearized, computations can be accomplished by look-up tables avoiding the Taylor-series expansion or other polynomial calculations with excessive floating point operations. The given values of the data are stored digitally in a memory. The data points in the LUT can then be approximated by piecewise linear approximation method.

Nonlinear outputs of the sensors have a variant degree of nonlinearity through the operating range. Consequently, in utilizing the linear approximation method to reach a precise accuracy, the data points in the LUT can be selected that are not equidistant. Whenever the sensor output is more "linear," the distance between the data points can be increased, thus decreasing the

memory need for LUT. In the fixed step table, the step size must be matched to the most "nonlinear" part of the function throughout the LUT.

Look-up table usage for linearization of the sensor outputs provides more accurate results particularly for small-scale-embedded systems having limited size of memory. Nevertheless, it is not recommended to make use of look-up tables for sensor outputs with high resolution, since it needs larger memory than a small-scale-embedded system has.

3.4. Cubic spline method

The linear interpolation method for sensor linearization process exhibits distortions that result in corners on the plot of the function at the breakpoints. To obtain an interpolating function that has continuity around the breakpoints, cubic splines are used. On the other hand, they are just piecewise continuous, inferring that the third derivative is not continuous. If the sensor linearization process is susceptible to the evenness of derivatives higher than the second, cubic spline method is not preferred.

Consider the general nonlinear continuous function $f(x)$ of single variable x within the interval $[\alpha, \beta]$ with breakpoints $\alpha = a_1 < a_2 < \ldots < a_n = \beta$. We have n breakpoints with $n - 1$ intervals. The cubic spline interpolation results in a piecewise continuous curve that traverses each breakpoint. For a set of n data points $\{(x_1, y_1), \ldots (x_n, y_n)\}$, and per interval, there is a unique cubic polynomial $s_i(x)$ with its particular factors as

$$s_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \text{ for } x_i \in [x_i, x_{i+1}] \tag{15}$$

The cubic polynomial requires two conditions to match the values of the data set at each end of the intervals, which results in a piecewise continuous function on $[\alpha, \beta]$,

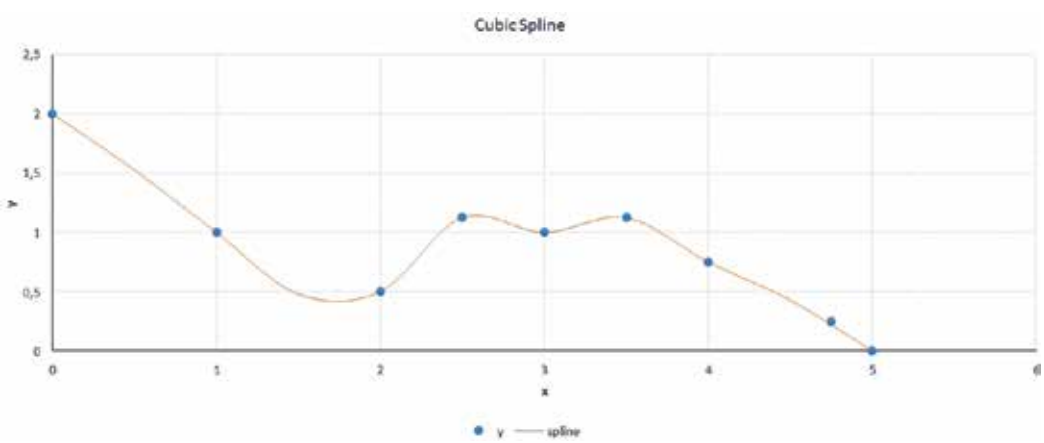

Figure 5. Cubic spline linearization of the data set in **Table 1**.

$$s_i(x_i) = y_i \text{ and } s_i(x_{+1i}) = y_{i+1} \tag{16}$$

Providing smoother interpolation, the first and the second derivatives must be continuous

$$s'_{i-1}(x_i) = s'_i(x_i),\ s''_{i-1}(x_i) = s''_i(x_i). \tag{17}$$

The remaining boundary conditions to complete the cubic spline polynomial are as follows:

1. For natural cubic spline: $s''_0(x_0) = 0,\ \ s''_{n-1}(x_n) = 0,$
2. For clamped cubic spline: $s''_0(x_0) = f'(x_0),\ \ s''_{n-1}(x_n) = f'(x_n).$

The data in **Table 1** are now linearized by cubic spline interpolation method using a function added in Excel which is developed by SRS1 Software, LLC. **Figure 5** indicates the cubic spline linearization of $f(x)$. Compared to the linear interpolation, cubic spline approximation results in a smoother curve which has continuity around the breakpoints.

4. Sensor linearization

Sensor outputs are not as linear as we expect. Consequently, linearization by hardware or software has become one of the challenging parts in measurement systems.

There are a variety of temperature sensors in the market. Yet, numerous temperature sensors have nonlinear characteristics or temperature measuring ranges are quite narrow. While utilizing these sensors, outputs are compensated, and linearized using digital circuitry and software, resulting in "smart sensors." These smart temperature sensors are thermocouples, thermistors, and resistive temperature detectors. The temperature sensors have low-voltage outputs, and nonlinear characteristic as mentioned in Section 2. Amplification and linearization of the voltage output of these temperature sensors is essential before utilizing them in industrial applications.

As the measurement range is the widest, and the linearity is poorer than most temperature sensors, thermocouples are in our scope of linearization practice. Due to their robustness to very high/very low temperatures, and oxidizing environments, they are preferred to other temperature sensors that melt with high temperature, or corrode with vapor. We present T/C compensation, and linearization with polynomial interpolation as a case study.

4.1. Theory of linearization

The development of a linearization block consists of several steps. We should first start with the nonlinear function $f(x)$, which characterizes the sensor output with respect to the input. This unknown function can be determined by a calibration process where multiple input/output (x and y) values are obtained experimentally. Using this set of data, $f(x)$ is then determined by one of the curve-fitting tools.

After obtaining the nonlinear function, $f(x)$, we can find the inverse function $f^{-1}(y)$ that is implemented in the linearization block as shown in **Figure 6**. In this part, we must make a

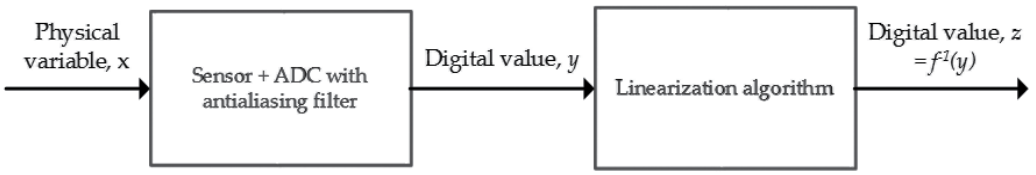

Figure 6. Sensor linearization procedure.

choice depending on the processor that we use. If our embedded measurement system is implemented on a 32-bit floating point controller, then the inverse function can be computed in real time. For 32-bit controllers, up to 13th degree polynomial interpolation can be executed with high accuracy, fast response, and minimum code size.

If the controller has an eight-bit processor with limited computing ability, then the best choice will be using a LUT stored in the memory. The function y is an n-bit integer at the output of an ADC, and it is used to fetch the inverse function $f^{-1}(y)$ in the memory. This is the procedure followed when using low-cost small-embedded systems for linearization process.

When we choose to use LUT, there will be limited source data set; therefore, an interpolation method must be used to compute the measured value. In this case, the interpolated value, $f^{-1}(y_m)$, can be calculated by using a piecewise linear interpolation method as

$$f^{-1}(y_m) \approx z_i + \frac{z_{i+1} - z_i}{y_{i+1} - y_i} \times (y_m - y_i) \tag{18}$$

where y_m is a measured variable between the points $[y_{i+1}, y_i]$ and $\frac{z_{i+1}-z_i}{y_{i+1}-y_i}$ is the slope of each segment. This procedure can be repeated for each interval between the breakpoints.

Similar procedure will be applied for cubic spline interpolation by using the equations given in Section 3.4.

Considering the required size of the memory for LUT entries and accuracy of the measurement, we know that for n-bit ADC, we need 2^n breakpoints for better accuracy. We have a contradiction whether to increase the number of breakpoints for better accuracy or to keep the number of LUT entries limited to decrease the memory size at the expense of accuracy. This decision will be given based on the circumstances. If we use a 32-bit controller with fast and high-computing ability, and large memory size, LUT size can be increased, but for an eight-bit controller it must be limited. In this case, we may limit the memory size by masking part of the bits of the digitized measurement without affecting the number of entries in the LUT. This can be considered only if the controller has a 16- or a 24-bit ADC, but limited memory size.

4.2. Thermocouple measurement setup and linearization algorithms

The most utilized thermocouples are types J, K, T, and E as given in **Table 2**. It can be interpreted from the table that the temperature ranges are the widest among other temperature sensors, but

http://dx.doi.org/10.5772/67986

Type	Composition	Temperature range
J	Iron vs Cu-Ni alloy	−210 to 1200°C
K	Ni-Cr alloy vs Ni-Al alloy	−270 to 1372°C
T	Cu vs Cu-Ni alloy	−270 to 400°C
E	Ni-Cr alloy vs Cu-Ni alloy	−270 to 1000°C

Table 2. Standard thermocouple types and their temperature ranges.

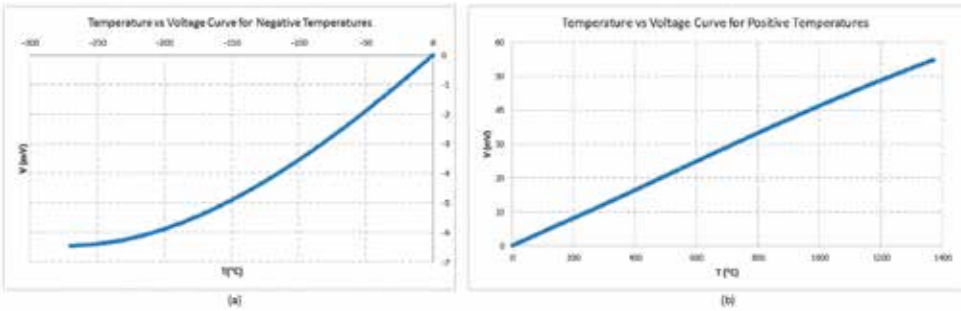

Figure 7. Characteristic curve of type K T/C (a) negative and (b) positive temperature range.

unfortunately the linearity is poor. We have selected type K T/C as its characteristic curve is considerably nonlinear, particularly in the negative temperature range as shown in **Figure 7**.

4.2.1. Hardware

The suggested temperature measuring system consists of a type K T/C, a low-voltage micropower amplifier, OPA333 with very low-offset voltage (max. 10 μV) and near-zero drift over time, 10 kΩ thermistor for cold-junction compensation of T/C, a cost-efficient 32-bit microcontroller, and a serial port driver (**Figure 8**). This system uses an analog hardware- and software-mixed linearization approach.

The low-voltage output of the type K T/C is in the range of −6.458 to 54.886 mV corresponding to −270 to 1372°C input range, so this output is amplified by an external amplifier. Then, the amplified voltage is applied to the built-in ADC input. A low-pass filter is used for noise suppression across the T/C ends.

We preferred Arduino Due based on a 32-bit ARM core microcontroller, which has 16-channel 12-bit ADCs; USB, Universal Synchronous/Asynchronous Receiver/Transmitter (USART), Serial Peripheral Interface (SPI), and I2C compatible Two-wire Interface (TWI) serial communication ports; 512 KB of flash and 100 KB of SRAM memory size are sufficient to run the linearization process by polynomial calculations, or to store the data for look-up table.

The embedded temperature measuring system is built and the amplified output of the T/C is connected to an analog channel of the microcontroller. The 12-bit built-in ADC of the controller board is used to obtain digital values corresponding to the mV output of the T/C.

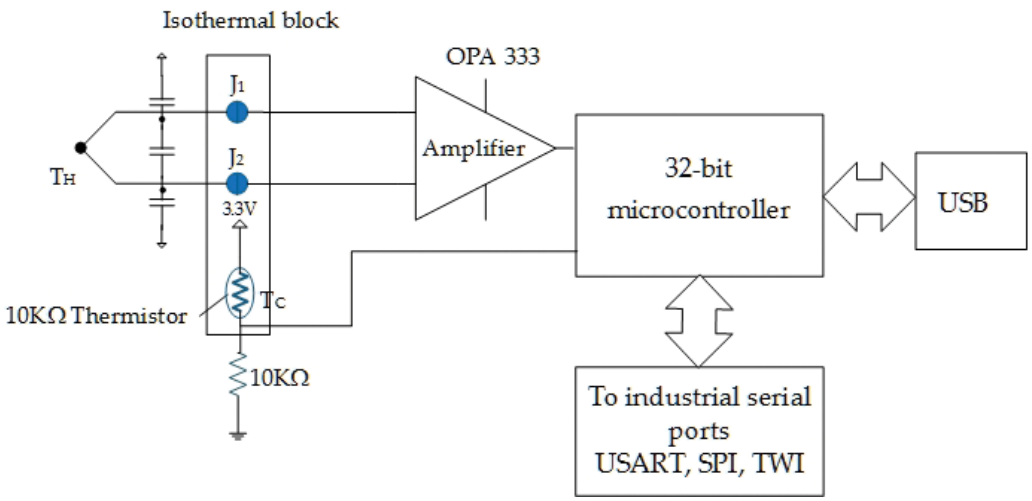

Figure 8. Temperature measuring system.

4.2.2. Software

Arduino Due board with ARM Cortex-M3 core can be programmed by its own integrated development environment (IDE) based on C/C++ programming language. The flowchart of the algorithm for the temperature measuring system is given in **Figure 9**.

The sixth degree polynomial obtained by Excel tool and the ninth degree polynomial [7] are used to reconstruct the inverse function of the T/C's characteristic function. For type K T/C, the following polynomial with the inverse coefficients given in **Table 3** is used for calculating the temperature value corresponding to the measured voltage in millivolts:

$$t_{90} = \sum_{1}^{n} d_i E^i \tag{19}$$

4.3. Linearization algorithms

The model for the type K T/C is implemented by polynomial calculations in Eqs. (4) and (5) in Simulink using NIST ITS-90 Thermocouple Database [8]. First, piecewise linear interpolation is used with LUT to linearize the temperature sensor output signal, and the interpolation values along with the measured temperature values are generated at the outputs. The model for type K T/C measuring system is shown in **Figure 10**.

The first block "Type K Thermocouple Model" calculates the polynomial model for negative and positive temperature ranges separately. The second block "ADC input and hardware model" consists of analog scaling of the measured voltage output of the TC, an anti-aliasing filter, and ADC quantizer with a sample and a hold block. The final block is for software specifications for converting ADC values to temperature where 1D LUT is used with linear and cubic spline interpolation algorithm for linearization.

http://dx.doi.org/10.5772/67986

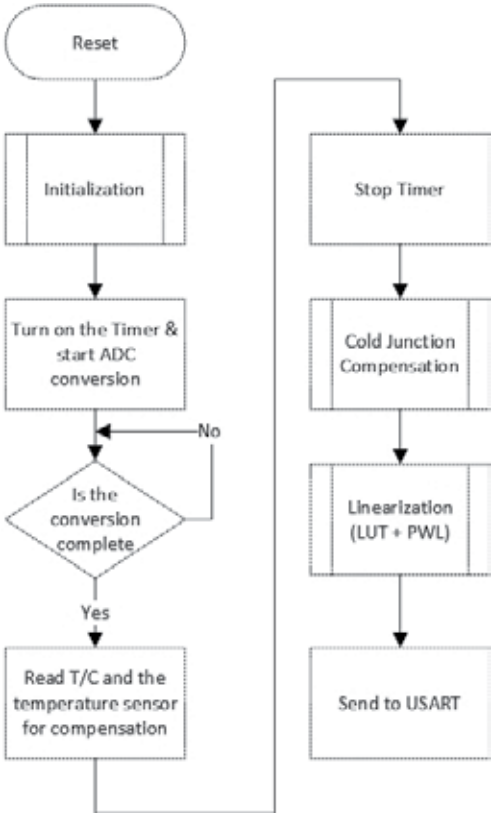

Figure 9. Flowchart of the developed software for the temperature measuring system.

Temperature range	−200 to 0°C	0 to 500°C	500 to 1372°C
Voltage range	−5.891 to 0	0 to 20.644	20.644 to 54.886
d_0	0.00E+00	0.00E+00	−1.32E+02
d_1	2.52E+01	2.51E+01	4.83E+01
d_2	−1.17E+00	7.86E−02	−1.65E+00
d_3	−1.08E+00	−2.50E−01	5.46E−02
d_4	−8.98E−01	8.32E−02	−9.65E−04
d_5	−3.73E−01	−1.23E−02	8.80E−06
d_6	−8.66E−02	9.80E−04	−3.11E−08
d_7	−1.05E−02	−4.41E−05	0.00E+00
d_8	−5.19E−04	1.06E−06	0.00E+00
d_9	0.00E+00	−1.05E−08	0.00E+00
Error range	−0.02 to 0.04	−0.05 to 0.04	−0.05 to 0.06

Table 3. Inverse coefficients for type K thermocouple.

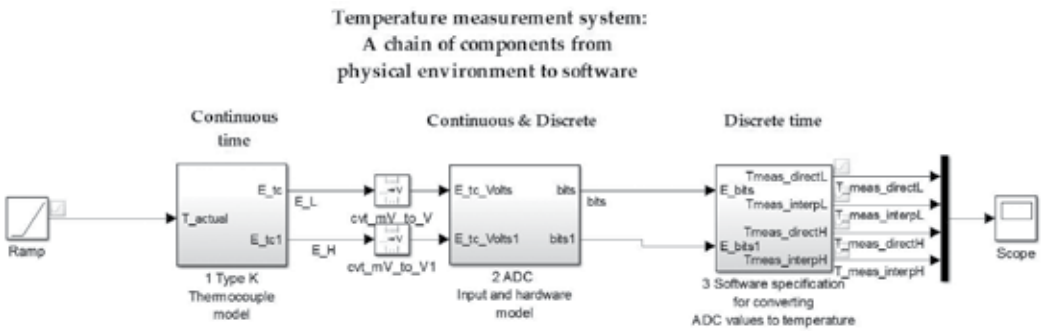

Figure 10. Model for type K T/C measuring system.

4.4. Simulation results

The linearization algorithms are simulated for the most linear and most nonlinear parts of the thermocouple characteristic curve. Least-squares, linear, quadratic, and quartic spline polynomial interpolation methods are applied for linearization over the intervals −270 to 0°C and 0–300°C, and the simulation results are given in **Figure 11**.

The Simulink model shown in **Figure 10** has given the simulation results as in **Figure 12**.

We have 2^{12} breakpoints for the 12-bit ADC conversion for the most accurate result. Consequently, in the inverse function derivation part of the simulation block, the direct and interpolated temperature values traced each other closely with an error of −9.15E−05 to 8.61E−05°C in the 0–1370°C range, and −1.51E−05 to 1.51E−05°C in the −270 to 0°C range.

4.5. Experimental results

During the test procedure, the measurements are carried out by the type K T/C, and the temperature is increased by 10°C over the interval of −270 to 1372°C. The values are saved as text file that can be evaluated in Excel. The same measurements are made by an identical T/C using the Fluke-725 process calibrator.

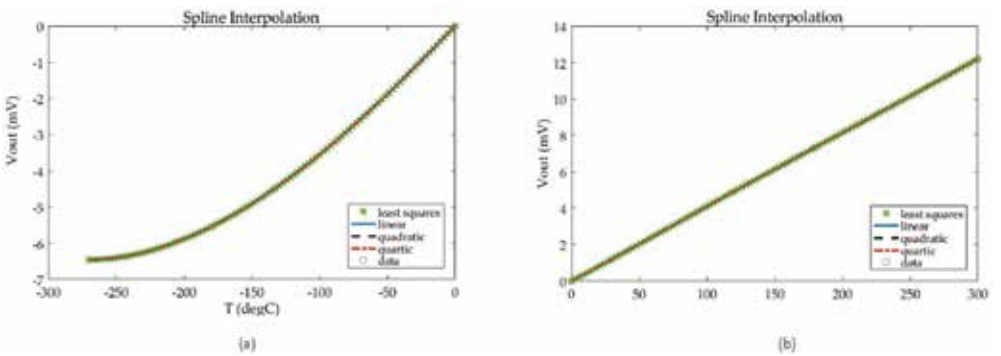

Figure 11. Spline polynomial interpolation results for (a) −270 to 0°C and (b) 0–300°C.

http://dx.doi.org/10.5772/67986

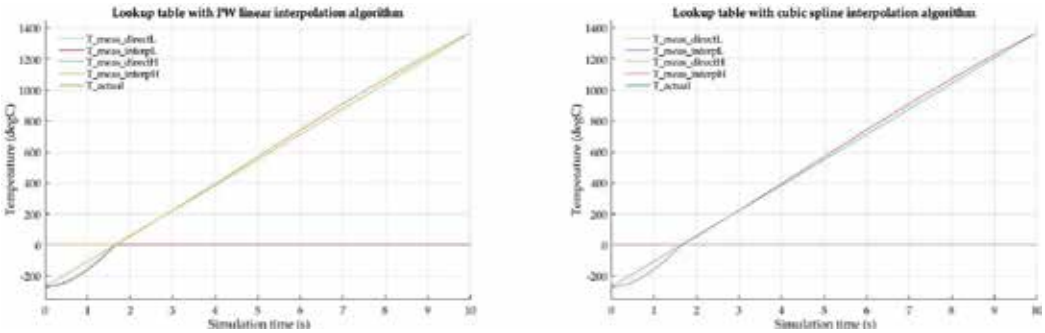

Figure 12. Linearization results for type K T/C by LUT with piecewise linear and cubic spline interpolation methods, respectively.

The obtained results are interpreted in terms of errors. The errors obtained from the piecewise linearization process with LUT for 28 breakpoints are given in **Figure 13**, and the error resulted in the positive temperature range is approximately −0.18 to 0.28°C. Compared to the linear function calculation error results declared by NIST ITS-90, −0.05 to 0.04°C for 0–500°C range, and −0.05 to 0.06°C for 500–1372°C range, the error range appears considerably higher than the "near-to-ideal" values, but piecewise linear approximation of the source data in the memory of the microcontroller optimizes the computation time.

The ninth degree polynomial inverse function is also applied to the measured T/C voltage over −200 to 1310°C range for the ranges described in **Table 3**. The measured values are incremented by about 21 μV corresponding to 0.5°C. The results in **Figure 14** revealed that for the negative portion of the temperature measurements, the sixth degree polynomial "best fits" the function, whereas the ninth degree polynomial tracks the temperature values better than the sixth degree polynomial for the positive temperature range. So, we propose a mixed linearization method for type K T/C for the negative and positive portions of the full range.

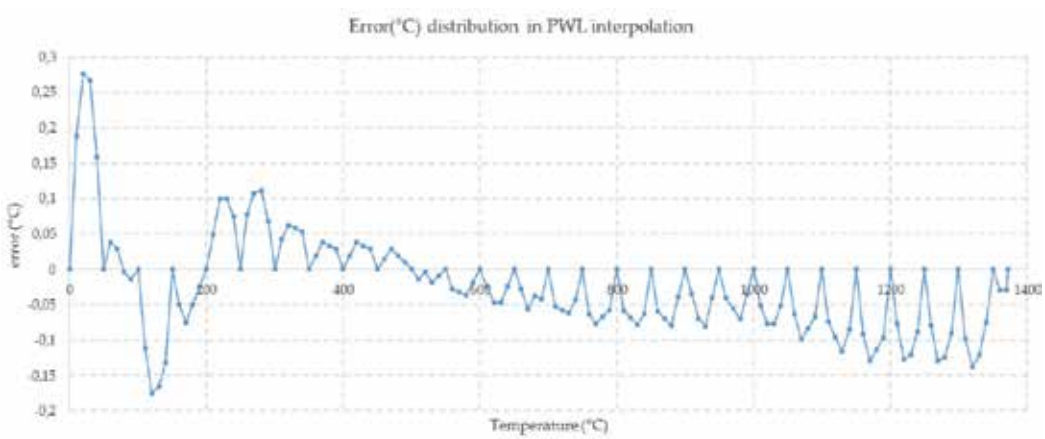

Figure 13. Error distribution for piecewise linear approximation method.

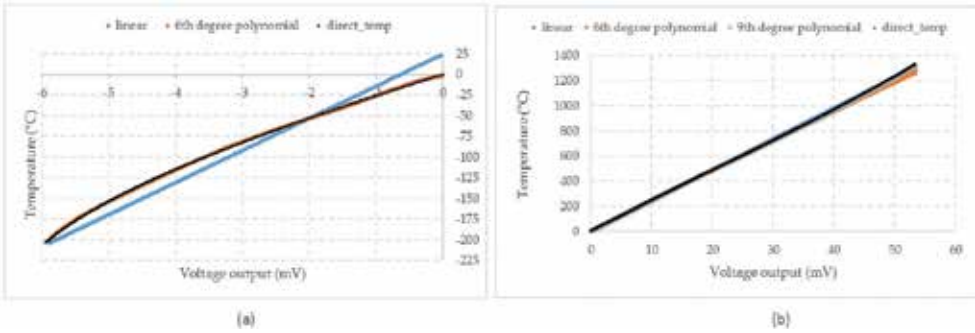

Figure 14. Linearization of T/C by polynomial for (a) negative and (b) positive temperature range.

5. Conclusion

In this chapter, we presented the measurement systems and linearization of sensors as well as the methods used in linearization. We presented the design and simulation process of a type K thermocouple as a case study. The linearization process was implemented with a 32-bit microcontroller. Type K thermocouple was connected to the Arduino Due controller board with an amplifier, and capacitors for the noise-suppression filter. Low-voltage output of the T/C, in the range of −6.458 to 54.886 mV, was amplified by an amplifier, and digitized by an internal 12-bit ADC. As the transmitters are low-power devices, a micro-power amplifier is utilized in the amplification of the output voltage of the T/C.

The cold-junction compensation of the thermocouple was realized by software using a thermistor fixed on an isothermal block. The actual type K thermocouple's voltage output in the nonlinear range was linearized by piecewise linear and polynomial interpolation methods, which were used to compute the temperature values between each breakpoint. Fewer number of breakpoints resulted in larger errors.

LUTs are used in a large range of applications including sensor linearization. The crucial points to consider are the number of LUT entries for better accuracy and the size of the LUT in design. Fewer number of LUT entries may cause serious errors in the measurement system, whereas larger number of entries may consume the memory of the controller.

In the polynomial approach for linearization, we propose 32-bit microcontrollers for better accuracy at the expense of cost compared to the small-embedded systems with limited computing ability and memory size. The calculation time for high-order polynomial equations may be too long for low-cost small-embedded systems with limited computational ability, but for implementation of the exact inverse sensor characteristic, the 32-bit controller Arduino Due is a reasonable solution.

Compared to the previous mixed signal works of the authors [11, 16], this case study offers a 32-bit ARM controller with high-computational ability and memory to run high-order polynomial interpolation method for linearization as well as higher accuracy in LUT with piecewise linearization. The ninth degree of polynomial was utilized in T/C linearization in [15], but the

range of linearization was limited to 0–200°C. We proposed a mixed polynomial linearization that best fits the negative and positive temperature ranges over the full range.

Author details

Dilşad Engin[1]* and Mustafa Engin[2]

*Address all correspondence to: dilsad.engin@ege.edu.tr

1 Ege Higher Vocational School, Control and Automation Technology Department, Ege University, Bornova, İzmir, Turkey

2 Ege Higher Vocational School, Electronic Technology Department, Ege University, Bornova, İzmir, Turkey

References

[1] Considine D. M., editor. Process Instruments and Controls Handbook. Third Edition. New York: McGraw-Hill; 1985. 1766 p. DOI: 10.1002/aic.690330127

[2] Gorbunov S. F., Tsypin B. V. Linearization of calibration characteristics of capacitance pressure sensors. Measurement Techniques. 2011;**53**(10):1113–1117. DOI: 10.1007/s11018-011-9626-y

[3] Ilić D., Butorac J., Ferković L. Temperature measurements by means of NTC resistors and a two-parameter approximation curve. Measurement. 2008;**41**(3):294–299. DOI: http://dx.doi.org/10.1016/j.measurement.2006.11.007

[4] Chen C. Evaluation of resistance–temperature calibration equations for NTC thermistors. Measurement. 2009;**42**(7):1103–1111. DOI: 10.1016/j.measurement.2009.04.004

[5] Radetić R., Pavlov-Kagadejev M., Milivojević N. The analog linearization of Pt100 working characteristic. Serbian Journal of Electrical Engineering. 2015;**12**(3):345–357. DOI: 10.2298/SJEE1503345R

[6] Zhu M. J. A thermometer based on diverse types thermocouples and resistance temperature detectors. J. Shanghai Jiaotong Univ. (Sci.). 2015;**20**(1):93–100. DOI: 10.1007/s12204-015-1594-y

[7] NIST 1999, 9.10 National Institute of Standards. NIST ITS-90 Thermocouple Database [Internet]. 12/21/1999 [Updated: 11/05/2014 15:16:48]. Available from: http://srdata.nist.gov/its90/main/ [Accessed: 2016]

[8] Aly A. A., Abo El-Lail A. S. A novel circuit for thermocouple signals linearization using AD converter. International Journal of Information Technology and Computer Science. 2013;**5**(3):56–60. DOI: 10.5815/ijitcs.2013.03.07

[9] Santhosh K. V., Roy B. K. Soft calibration technique with SVM for intelligent temperature measurement using thermocouple. In: 2014 International Conference on Advances in Electronics Computers and Communications; 10–11 Oct. 2014; Bangalore, India. IEEE; 2015. p. 1–4. DOI: 10.1109/ICAECC.2014.7002380

[10] Danisman K., Dalkiran I., Celebi F. V. Design of a high precision temperature measurement system based on artificial neural network for different thermocouple types. Measurement. 2006;**39**(8):695–700. DOI: 10.1016/j.measurement.2006.03.015

[11] Engin M. Linearization and maximizing performance of the thermocouple using embedded microcontroller. Journal of Computational Intelligence and Electronic Systems. March 2014;**3**(1):29–35. DOI: 10.1166/jcies.2014.1071

[12] Wang D., Song L., Zhang Z. Contact temperature measurement system based on tungsten-rhenium thermocouple. In: 2010 International Conference on Computer Application and System Modeling (ICCASM 2010); 22–24 Oct. 2010; Shanxi, Taiyuan. IEEE; 2010. pp. V12-660–V12-663. DOI: 10.1109/ICCASM.2010.5622449

[13] Gerstenhaber M., O'Sullivan M. Condition thermocouples before digitizing. Test & Measurement World. 2011;31:23–24.

[14] Wei G., Wang X., Sun J. Signal processing method with cold junction compensation for thermocouple. In: 2009 IEEE Instrumentation and Measurement Technology Conference; 5–7 May 2009; Singapore. IEEE; 2009. pp. 1458–1462. DOI: 10.1109/IMTC.2009.5168685

[15] Sarma U., Boruah, P. K. Design and development of a high precision thermocouple based smart industrial thermometer with on line linearisation and data logging feature. Measurement. 2010;**43**(10):1589–1594. DOI: 10.1016/j.measurement.2010.09.003

[16] Engin M., Engin D. Compensation of thermocouple nonlinearities with embedded system. In: Scientific Committee, Editorial (MECO 2016), editors. 2016 5th Mediterranean Conference on Embedded Computing (MECO); 12–16 June 2016; Bar, Montenegro. IEEE; 2016. pp. 234–237. DOI: 10.1109/MECO.2016.7525749

[17] Bengtsson L. E. Lookup table optimization for sensor linearization in small embedded systems. Journal of Sensor Technology. 2012;**2**(4):177–184. DOI: 10.4236/jst.2012.24025

[18] Wang X., Wei G., Sun J. Free knot recursive B-spline for compensation of nonlinear smart sensors. Measurement. 2011;**44**(5):888–894. DOI: 10.1016/j.measurement.2011.02.009

[19] Eds. Croarkin C. and Tobias P. /NIST/SEMATECH. e-Handbook of Statistical Methods [Internet]. 01.06.2003 [Updated: 30.10.2013]. Available from: http://www.itl.nist.gov/div898/handbook/ [Accessed: 05.01.2017].

[20] Lin M., Carlsson J. G., Ge D., Shi J., Tsai J. A review of piecewise linearization methods. Mathematical Problems in Engineering. 2013;**2013**:1–8. DOI: 10.1155/2013/101376

[21] Hillier F. S., Lieberman G. J. Introduction to Operations. 7th ed. New York, NY, USA: McGraw-Hill; 1995. 1237 p. ISBN 0-07-232169-5

[22] Taha H. A. Operations Research. 8th ed. Upper Saddle River, New Jersey, NJ, USA: Pearson, Prentice Hall; 2007. 838 p. ISBN 0-13-188923-0

Chapter 6

Mechatronic Design of a Lower Limb Exoskeleton

Luis I. Minchala, Fabián Astudillo-Salinas,
Kenneth Palacio-Baus and Andrés Vazquez-Rodas

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/67460

Abstract

This chapter presents a lower limb exoskeleton mechatronic design. The design aims to be used as a walking support device focused on patients who suffer of partial lower body paralysis due to spine injuries or caused by a stroke. First, the mechanical design is presented and the results are validated through dynamical simulations performed in Autodesk Inventor and MATLAB. Second, a communication network design is proposed in order to establish a secure and fast data link between sensors, actuators, and microprocessors. Finally, patient-exoskeleton system interaction is presented and detailed. Movement generation is performed by means of digital signal processing techniques applied to electromyography (EMG) and electrocardiography (EEG) signals. Such interaction system design is tested and evaluated in MATLAB whose results are presented and explained. A proposal of real-time supervisory control is also presented as a part of the integration of every component of the exoskeleton.

Keywords: lower limb, exoskeleton, control, design, kinematics, dynamics, analysis

1. Introduction

Lately, exoskeletons are designed to provide strength in gait and to transport heavy loads. There are also designs for assisting people with disorders in motion or elderly people. Gait rehabilitation is one of the greatest challenges for society in the upcoming years due to population aging and the increase of diseases affecting motion. A partial or total paralysis of one side of the body due to injuries in the motor centers of the brain is called Hemiplegia. Hemiplegia is a disorder that causes one half of the human body to fail to perform its functions. This disorder is caused mainly due to stroke and in many cases it is hereditary. Recovery from the stroke is difficult and the treatment is prolonged. Generally, an injury to the right side of the brain will cause a left-sided hemiplegia while an injury to the left side of the brain will cause a right-sided hemiplegia [1].

Wearable robotics is an area that provides solutions for such problems. A wearable robot extends, complements, or fully substitutes human function or empowers the human limb where it is worn. These kinds of robots are classified according to the function they perform [2]:

- *Empowering robotic exoskeletons*: These kinds of robots are known as extenders since they extend the strength of the human hand beyond its natural ability while maintaining human control of the robot.
- *Orthotic robots*: An orthosis maps the anatomy of a limb to restore lost functions. The robotic counterpart of orthosis is robotic exoskeletons that complement the ability of the limbs. Exoskeletons are also capable of restoring handicapped functions.
- *Prosthetic robots*: These robots are devices that fully substitute lost limbs.

Figure 1 shows two examples of wearable robots. Scientific community differentiates exoskeletons from orthosis by defining the former as the devices that enhance physical capabilities of wholesome users, and the latter as the devices that assist persons with impairments in the limbs [3]. In spite of their differences, both devices act in parallel with the limb. The applications of exoskeleton robots are varied, ranging from military applications to the medical field. In the medical field, exoskeletons in conjunction with rehabilitation therapies can assist patients with spinal cord injury, stroke and lower limb paralysis, caused by hemiplegia, to perform gait training [4].

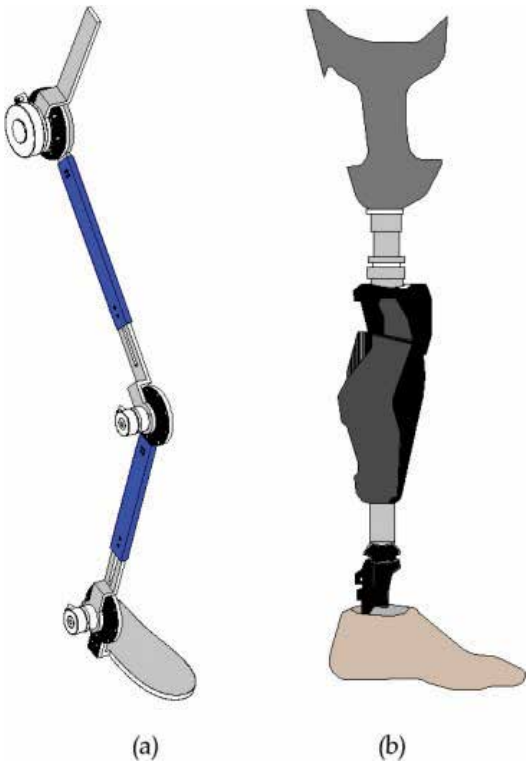

Figure 1. (a) Lower limb orthotic exoskeleton; (b) lower limb prosthetic robot.

1.1. Lower limb exoskeleton prototypes around the world

There are many designs of robotic exoskeletons for different purposes around the world. **Table 1** shows a summary of some of the main active projects in the scientific community which have been reported in journal and conference papers.

1.2. Lower limb exoskeleton features

Exoskeletons are defined as anthropomorphic mechanical devices worn by an operator that fit closely to the body anatomy and work in coordination with the movements of the user. Among the main features of an exoskeleton to be considered during the design, we have:

- *Design must be anthropomorphic*: Current designs are unnatural in shape. Another limitation of exoskeletons is the lack of direct information exchange between the human nervous system and the wearable robot. Motion intentions originate with the user whose physiological state and desires must be discerned and interpreted.
- *Design must be flexible*: The length of the thigh, shank and waist should be adjustable [5]. The length variation of thigh and shank is about 6 cm for average people with height from 1.60 m

No.	Institution	Name	Purpose	Technical data	Control method

1	Yonsei University South Korea	Walking assistance lower limb exoskeleton	Patients with lower limb paralysis	• 200W brushless dc motor
				• Harmonic drives
				• Torques:
				  • Hip: 79.3 nm
				  • Knee: 42.2 nm
					• Active kinematic control
					• Stability control

2	University of Tsukuba Japan	HAL-3	Patients with lower limb paralysis	• DC servo motors • Harmonic reducers in joints	• Conscious recognition based on plantar pressure and torso angle
3	Centre for Automation and Robotics Spain	ATLAS	Quadriplegic patients	• Brushless motors (Maxon) • Harmonic reducers • Peak torque of 57 nm (hip)	• CoP stability control • Conscious recognition
4	Berkeley Robotics and Human Engineering Laboratory USA	Lower extremity exoskeleton (BLEEX)	Allow personnel the ability to carry major loads such as food, rescue equipment, first-aid supplies, etc.	• DC brushless motors • Harmonic reducers	• CoP stability control • Conscious recognition

Table 1. Robotic exoskeletons around the world.

to 1.80 m. The shank length is about 0.246 times the stature, and thigh length is about 0.245 times the stature.

- *Increase joint strength*: Exoskeletons do not transfer substantial load to the ground, but augments joint torque. This consideration might be used to reduce joint pain or increase joint strength in paralyzed or weak joints.
- *Selection of the degrees of freedom* (*DOF*). The exoskeleton must be compliant to the freedom of motion of joints. **Table 2** shows the DOF of a lower limb exoskeleton.
- The actuator of exoskeleton robot should have a large output power-to-weight ratio and characteristics as low inertia, fast response, high precision, etc.

DOF of joints	Joints	DOF	Range of freedom	Driving force needed
	Hip	Flexion/extension	$-120° \le \theta \le 65°$	80–100 N/m
		Adduction/abduction	$-30° \le \theta \le 40°$	Spring
		Rotation	$-30° \le \theta \le 30°$	Spring
	Knee	Flexion/extension	$-120° \le \theta \le 0°$	45–70 N/m
	Ankle	Pronation/rotation	$-15° \le \theta \le 30°$	Spring
		Dorsiflexion/toe flexion	$-20° \le \theta \le 50°$	Spring
		Varus/valgus	$-30° \le \theta \le 20°$	Spring

Table 2. Design of DOF for a lower limb exoskeleton.

2. Lower limb exoskeleton mechanical design

2.1. Kinematic analysis

Kinematics gives the relationship between the articular space and the end effector of the robot. This approach is useful to generate trajectories, and joint actuators set points for controlling them. **Figure 2** shows the kinematic analysis results.

Section 2.1.1 shows direct kinematics analysis of the exoskeleton and Section 2.1.2 shows inverse kinematics of the exoskeleton.

2.1.1. Direct kinematics

The main objective of the direct kinematics analysis is to find the position and orientation of the last link of the robot from the values of the articular coordinates. Direct kinematics equations are defined by

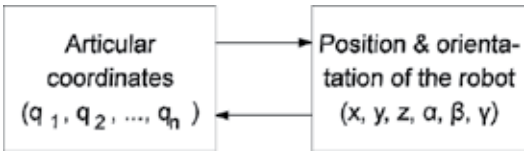

Figure 2. Kinematics of a robot.

http://dx.doi.org/10.5772/67460

$$
\begin{aligned}
x &= f_x(q_1, q_2, \ldots, q_n) \\
y &= f_y(q_1, q_2, \ldots, q_n) \\
z &= f_z(q_1, q_2, \ldots, q_n) \\
\alpha &= f_\alpha(q_1, q_2, \ldots, q_n) \\
\beta &= f_\beta(q_1, q_2, \ldots, q_n) \\
\gamma &= f_\gamma(q_1, q_2, \ldots, q_n)
\end{aligned} \tag{1}
$$

Figure 3 shows reference axes placed on each joint of the exoskeleton by using the Denavit-Hartemberg (DH) algorithm [6]. **Table 3** shows the DH parameters for each link of the exoskeleton.

The exoskeleton has six DOFs. Every joint is revolute, and the distribution of the DOF considering displacement of the exoskeleton in the sagittal plane is the following:

- Two DOFs in each side of the hip
- Two DOFs in the knees, one per each joint
- Two DOFs in the ankles, one per each joint

The analysis to be performed, due to symmetry, is presented in one right limb which has the articular variables q_1, q_2, and q_3. The transformation matrix that relates the position of the third link of the right limb is:

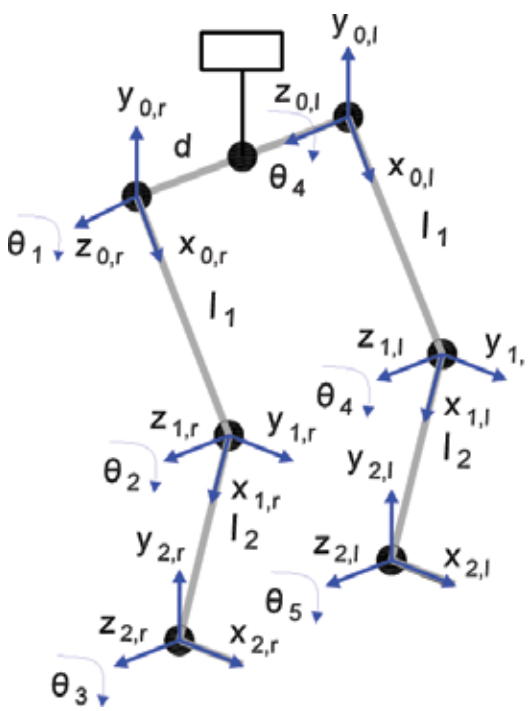

Figure 3. Allocation of reference axes on the lower limb exoskeleton.

Right limb	a_i	α_i	d_i	θ_i	Left limb	a_i	α_i	d_i	θ_i
1	l_1	0	0	θ_1	4	l_1	0	0	θ_4
2	l_2	0	0	θ_2	5	l_2	0	0	θ_5
3	l_3	0	0	θ_3	6	l_3	0	0	θ_6

Table 3. DH parameters for the exoskeleton links.

$$\mathbf{T} = \begin{bmatrix} C(\theta_1+\theta_2+\theta_3) & -S(\theta_1+\theta_2+\theta_3) & 0 & l_1C\theta_1 + l_2C(\theta_1+\theta_2) + l_3C(\theta_1+\theta_2+\theta_3) \\ S(\theta_1+\theta_2+\theta_3) & C(\theta_1+\theta_2+\theta_3) & 0 & l_1S\theta_1 + l_2S(\theta_1+\theta_2) + l_3S(\theta_1+\theta_2+\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where $C(\theta_i)$ represents the cosine of the articular variable i, $S(\theta_i)$ represents the sine of the articular variable i.

The forth column of matrix **T** in Eq. (2) represents the position of the third link of the right limb. The analysis for obtaining Eq. (2) is performed in the plane xy (sagittal plane). Therefore, the direct kinematic equations of the right limb are:

$$\begin{aligned} x_d &= l_1C(\theta_1) + l_2C(\theta_1+\theta_2) + l_3C(\theta_1+\theta_2+\theta_3) \\ y_d &= l_1S(\theta_1) + l_2S(\theta_1+\theta_2) + l_3S(\theta_1+\theta_2+\theta_3) \\ z_d &= -d \end{aligned} \tag{3}$$

2.1.2. Inverse kinematics

Inverse kinematics provides the values of the articular variables ($\boldsymbol{q} = [q_1,\ q_2, \ldots, q_n]^T$) for a specific location in the space of the last link of a kinematic chain in a robot $[x, y, z, \alpha, \beta, \gamma]^T$. Inverse kinematics equations are defined by

$$\begin{aligned} q_1 &= f_1(x, y, z, \alpha, \beta, \gamma) \\ q_2 &= f_2(x, y, z, \alpha, \beta, \gamma) \\ &\vdots \\ q_n &= f_n(x, y, z, \alpha, \beta, \gamma) \end{aligned} \tag{4}$$

The inverse kinematics equations for the lower limb exoskeleton are to be obtained by the geometric method. It is convenient to perform the analysis in the sagittal plane, as shown in **Figure 4**.

Considering **Figure 4**, the following variables are defined:

$$\begin{aligned} \text{Final link position} &= [x, y] \\ \text{Third link angle} &= \delta \\ \text{Links lenghts} &= l_1, l_2, l_3 \end{aligned}$$

http://dx.doi.org/10.5772/67460

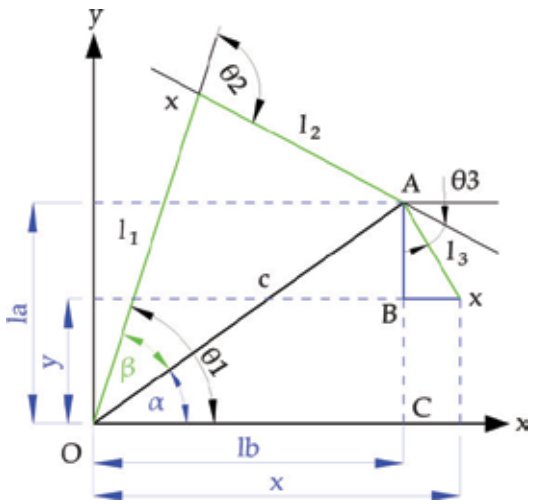

Figure 4. Geometric diagram to perform the inverse kinematics analysis.

By applying triangle's laws and proportions, we obtain:

$$\begin{aligned} LA &= y - l_3\cos(\delta) \\ LB &= x - l_3\sin(\delta) \\ c &= \sqrt{LA^2 + LB^2} \\ \alpha &= \tan^{-1}\left(\frac{LA}{LB}\right) \end{aligned} \tag{5}$$

$$\begin{aligned} \theta_1 &= \alpha + \beta \\ \gamma &= \cos^{-1}\left(\frac{l_1^2 + l_2^2 - c^2}{2l_1l_2}\right) \\ \theta_2 &= \gamma - \pi \\ \theta_3 &= \delta - \theta_1 - \theta_2 \end{aligned} \tag{6}$$

Finally, by substituting variables, we obtain:

$$\begin{aligned} \theta_1 &= \tan^{-1}\left(\frac{y - l_3\sin(\delta)}{x - l_3\cos(\delta)}\right) + \cos^{-1}\left(\frac{l_1^2 - l_2^2 + \left(y - l_3\sin(\delta)\right)^2 + \left(x - l_3\cos(\delta)\right)^2}{2l_1\sqrt{\left(y - l_3\sin(\delta)\right)^2 + \left(x - l_3\cos(\delta)\right)^2}}\right) \\ \theta_2 &= \cos^{-1}\left(\frac{l_1^2 - l_2^2 - \left(y - l_3\sin(\delta)\right)^2 + \left(x - l_3\cos(\delta)\right)^2}{2l_1l_2}\right) \\ \theta_3 &= \delta - \theta_1 - \theta_2 \end{aligned} \tag{7}$$

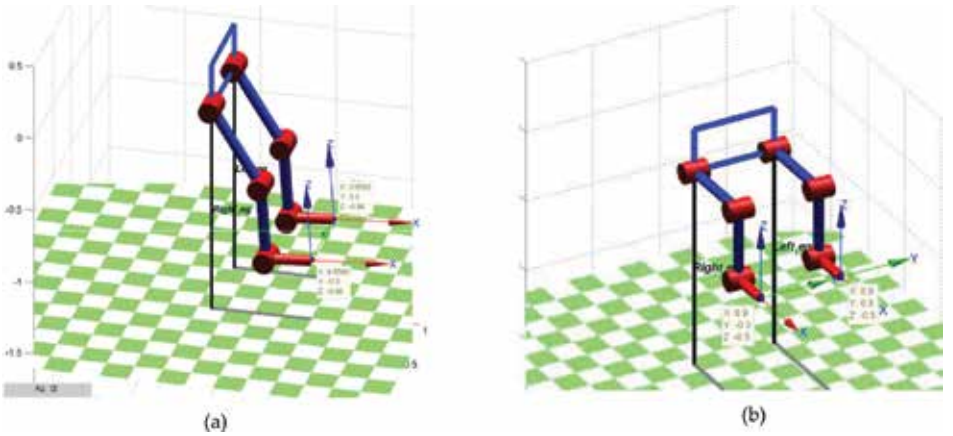

Figure 5. (a) Direct kinematics simulation; (b) inverse kinematics simulation.

2.1.3. Kinematics simulation

The Robotics toolbox for MATLAB is used for the kinematic simulation. **Figure 5a** shows the result of applying $\left[-\frac{\pi}{2}, 0, \frac{\pi}{2}\right]$ as articular variables values. For the inverse kinematics simulation, the third link is located at

$$\begin{aligned} \text{Left limb} &= [0.9, 0.3, -0.5] \\ \text{Right limb} &= [0.9, -0.3, -0.5] \end{aligned}$$

The joint values are:

$$\begin{array}{cc} \theta_1 = 0 & \theta_4 = 0 \\ \theta_2 = -\pi/2 & \theta_5 = -\pi/2 \\ \theta_3 = \pi/2 & \theta_6 = \pi/2 \end{array}$$

2.2. Dynamic analysis

There are two well-known methodologies for obtaining the dynamic model of any mechanism, the Euler-Lagrange method which is an energy method, and the Newton-Euler method which is based on the equilibrium of forces and torques. Hereafter, the Euler-Lagrange method is used for obtaining the dynamic equations of the exoskeleton, whose equations are:

$$\begin{aligned} \boldsymbol{\tau} &= \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} \\ L &= K - P \end{aligned} \tag{8}$$

where i represents the DOF, q_i represents the generalized coordinates (articular coordinates θi), τ represents the force and torque vector applied to joint i, L is the Lagrangian function, K is the kinetic energy of the system, and P is the potential energy of the system.

According to [6], a dynamic model is based on:

- The spatial location of a robot is defined by its articular variables or by the coordinates of its last link, as well as their derivatives;
- The forces and torques applied in the joints or in the last link of the robot; and
- The robot dimensions.

There are two types of dynamic models:

- Direct dynamic model: Express the transient evolution of the articular coordinates of the robot, according with the forces and torques involved in the system.
- Inverse dynamic model: Express the forces and torques in every junction of the system, according with the transient evolution of the articular coordinates and their derivatives.

The direct dynamic model is obtained in the following, starting with the kinetic energy calculation,

$$K_i = \frac{1}{2} m_i \dot{\mathbf{s}}_i^T \dot{\mathbf{s}}_i + \frac{1}{2} \boldsymbol{\omega}_i^T \mathbf{I}_i \boldsymbol{\omega}_i \tag{9}$$

where m_i is the mass of the i-link, $\dot{\mathbf{s}}_i$ is the translational speed of the center of gravity G of the i-link, $\mathbf{I}_i$ is the inertia matrix of the link, and $\boldsymbol{\omega}_i$ is the angular speed.

The potential energy of an i-link is defined by,

$$P_i = m_i g h_{ci} \tag{10}$$

where m_i is the mass of the link, g is the gravity acceleration, h_{ci} is the distance between the origin of the link, and the center of gravity of the i-link is parallel to the gravity vector.

Therefore, the following Lagrangian function is obtained:

$$L_i = \sum_{i=1}^{n} K_i - \sum_{i=1}^{n} P_i \tag{11}$$

By applying Eqs. (9)–(11) with the consideration shown in **Figure 6**, we obtain the dynamic equations:

$$\begin{aligned}
L &= K_1 + K_2 + K_3 - P_1 - P_2 - P_3 \\
L &= \frac{1}{2} m_1 l_{g1}^2 \dot{\theta}_1^2 + \frac{1}{2} I_1 \dot{\theta}_1^2 + \frac{1}{2} m_2 [l_1^2 \dot{\theta}_1^2 + l_{g2}^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + 2 l_1 l_{g2} \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) C_2] + \frac{1}{2} I_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 \\
&\quad + \frac{1}{2} m_3 [l_1^2 \dot{\theta}_1^2 + l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + l_{g3}^2 (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)^2 + 2 l_1 l_2 \dot{\theta}_1 (\dot{\theta}_1 + I \dot{\theta}_2) C_2 \\
&\quad + 2 l_1 l_{g3} \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) C_{23} + 2 l_2 l_{g3} (\dot{\theta}_1 + \dot{\theta}_2)(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) C_3] + \frac{1}{2} I_3 (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)^2 \\
&\quad - m_1 g l_{g1} S_1 - m_2 g (l_1 S_1 + l_{g2} S_{12}) - m_3 g (l_1 S_1 + l_2 S_{12} + l_{g3} S_{123})
\end{aligned} \tag{12}$$

The torque for joint 1, τ_1, is obtained by deriving Eq. (12) with respect to θ_1:

$$\begin{aligned}\frac{\partial L}{\partial \dot{\theta}_1} &= m_1 l_{g1}^2 \dot{\theta}_1 + I_1 \dot{\theta}_1 + m_2[l_1^2 \dot{\theta}_1 + l_{g2}^2(\dot{\theta}_1 + \dot{\theta}_2) + l_1 l_{g2} C_2(2\dot{\theta}_1 + \dot{\theta}_2)] \\ &+ I_2(\dot{\theta}_1 + \dot{\theta}_2) + m_3[l_1^2 \dot{\theta}_1 + l_2^2(\dot{\theta}_1 + \dot{\theta}_2) + l_{g3}^2(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) \\ &+ l_1 l_2 C_2(2\dot{\theta}_1 + \dot{\theta}_2) + l_1 l_{g3} C_{23}(2\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) + l_2 l_{g3} C_3(2\dot{\theta}_1 + 2\dot{\theta}_2 + \dot{\theta}_3)] \\ &+ I_3(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)\end{aligned} \tag{13}$$

The time derivative of Eq. (13) is:

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) &= \ddot{\theta}_1[m_1 l_{g1}^2 + I_1 + m_2(l_1^2 + l_{g2}^2 + 2l_1 l_{g2} C_2) + I_2 + m_3[l_1^2 + l_2^2 + l_{g3}^2 + 2l_1 l_2 C_2 \\ &+ 2l_1 l_{g3} C_{23} + 2l_2 l_{g3} C_3] + I_3] + \ddot{\theta}_2[m_2(l_{g2}^2 + l_1 l_{g2} C_2) + I_2 + m_3(l_2^2 + l_{g3}^2 + l_1 l_2 C_2 \\ &+ l_1 l_{g3} C_{23} + 2l_2 l_{g3} C_3) + I_3] + \ddot{\theta}_3[m_3(l_{g3}^2 + l_1 l_{g3} C_{23} + l_2 l_{g3} C_3) + I_3] \\ &- m_2 l_1 l_{g2} S_2(2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) - m_3 l_1 l_2 S_2(2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) - m_3 l_1 l_{g3} S_{23}(2\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)(\dot{\theta}_2 + \dot{\theta}_3) \\ &- m_3 l_2 l_{g3} S_3(2\dot{\theta}_1 \dot{\theta}_3 + 2\dot{\theta}_2 \dot{\theta}_3 + \dot{\theta}_3^2)\end{aligned} \tag{14}$$

The partial derivative $\frac{\partial L}{\partial \theta_1}$ is:

$$\frac{\partial L}{\partial \theta_1} = -m_1 g l_{g1} C_1 - m_2 g(l_1 C_1 + l_{g2} C_{12}) - m_3 g(l_1 C_1 + l_2 C_{12} + l_{g3} C_{123}) \tag{15}$$

Finally, τ_1 is calculated:

$$\tau_1 = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) - \frac{\partial L}{\partial \theta_1}$$

$$\begin{aligned}\tau_1 &= \ddot{\theta}_1[m_1 l_{g1}^2 + I_1 + m_2(l_1^2 + l_{g2}^2 + 2l_1 l_{g2} C_2) + I_2 + m_3[l_1^2 + l_2^2 + l_{g3}^2 + 2l_1 l_2 C_2 \\ &+ 2l_1 l_{g3} C_{23}] + I_3] + \ddot{\theta}_2[m_2(l_{g2}^2 + l_1 l_{g2} C_2) + I_2 + m_3(l_2^2 + l_{g3}^2 + l_1 l_2 C_2 + l_1 l_{g3} C_{23} \\ &+ 2l_2 l_{g3} C_3) + I_3] + \ddot{\theta}_3[m_3(l_{g3}^2 + l_1 l_{g3} C_{23} + l_2 l_{g3} C_3) + I_3] \\ &- m_2 l_1 l_{g2} S_2(2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) - m_3 l_1 l_2 S_2(2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) - m_3 l_1 l_{g3} S_{23}(2\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)(\dot{\theta}_2 + \dot{\theta}_3) \\ &- m_3 l_2 l_{g3} S_3(2\dot{\theta}_1 \dot{\theta}_3 + 2\dot{\theta}_2 \dot{\theta}_3 + \dot{\theta}_3^2) + m_1 g l_{g1} C_1 + m_2 g(l_1 C_1 + l_{g2} C_{12}) \\ &+ m_3 g(l_1 C_1 + l_2 C_{12} + l_{g3} C_{123})\end{aligned} \tag{16}$$

The same procedure is applied for joints 2 and 3:

$$\tau_2 = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_2}\right) - \frac{\partial L}{\partial \theta_2}$$

$$\begin{aligned}\tau_2 &= \ddot{\theta}_1[m_2(l_{g2}^2 + l_1 l_{g2} C_2) + I_2 + m_3(l_2^2 + l_{g3}^2 + l_1 l_2 C_2 + l_1 l_{g3} C_{23} + 2l_2 l_{g3} C_3) + I_3] \\ &+ \ddot{\theta}_2[m_2 l_{g2}^2 + I_2 + m_3(l_2^2 + l_{g3}^2 + 2l_2 l_{g3} C_3) + I_3] + \ddot{\theta}_3[m_3(l_{g3}^2 + l_2 l_{g3} C_3) + I_3] \\ &- m_2 l_1 l_{g2} S_2 \dot{\theta}_1 \dot{\theta}_2 - m_3 l_1 l_2 S_2 \dot{\theta}_1 \dot{\theta}_2 - m_3 l_1 l_{g3} S_{23} \dot{\theta}_1(\dot{\theta}_2 + \dot{\theta}_3) \\ &- m_3 l_2 l_{g3} S_3(2\dot{\theta}_1 + 2\dot{\theta}_2 + \dot{\theta}_3)(\dot{\theta}_3) + m_2 l_1 l_{g2} S_2 \dot{\theta}_1(\dot{\theta}_1 + \dot{\theta}_2) + m_3 l_1 l_2 S_2 \dot{\theta}_1(\dot{\theta}_1 + \dot{\theta}_2) \\ &+ m_3 l_1 l_{g3} S_{23} \dot{\theta}_1(\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) + m_2 g l_{g2} C_{12} + m_3 g(l_2 C_{12} + l_{g3} C_{123})\end{aligned} \tag{17}$$

http://dx.doi.org/10.5772/67460

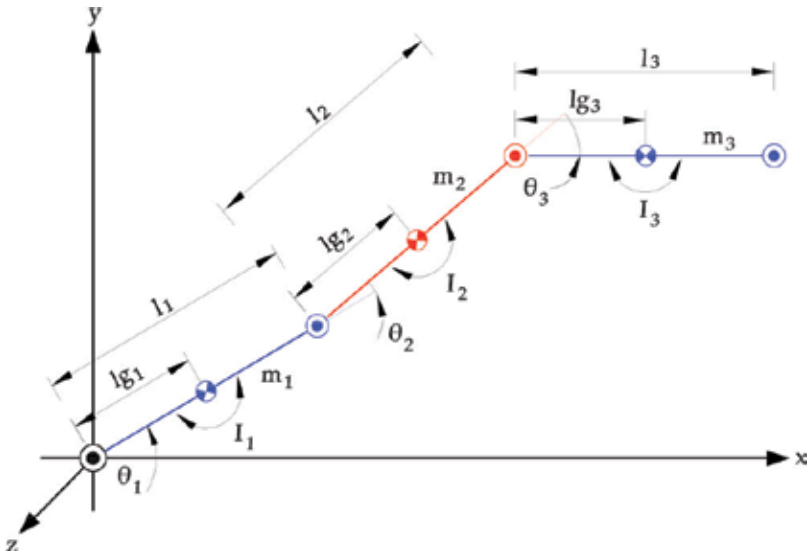

Figure 6. *i*-link parameters for dynamic modeling.

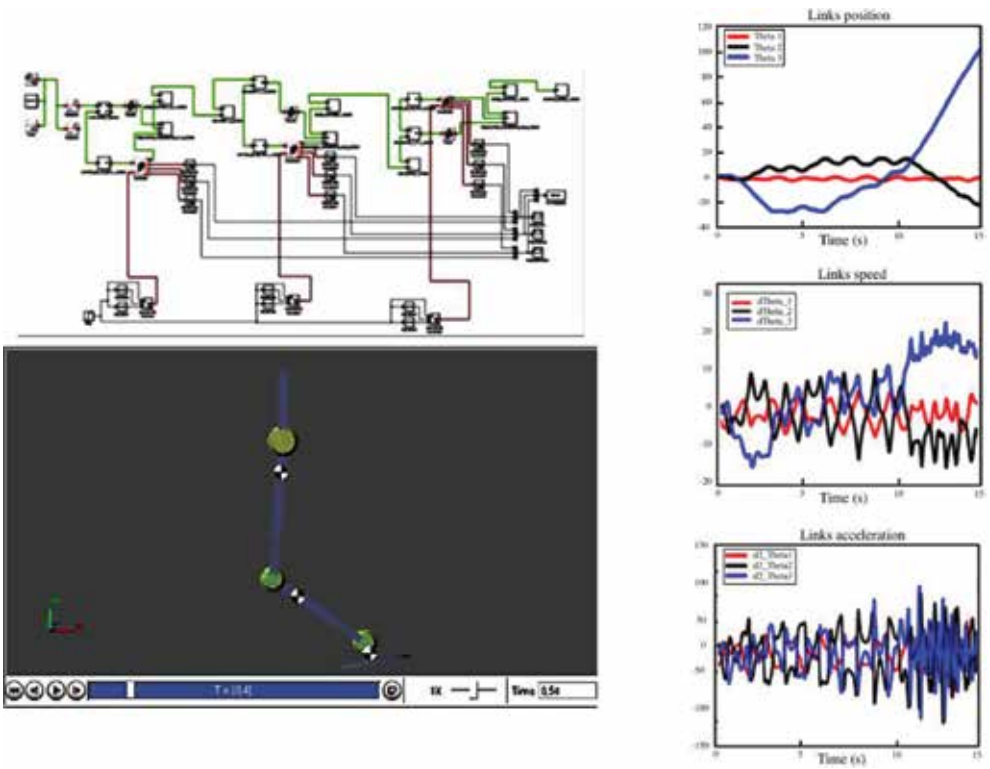

Figure 7. Dynamic simulation of the exoskeleton in simulink.

$$\tau_3 = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_3}\right) - \frac{\partial L}{\partial \theta_3}$$
$$\tau_3 = \ddot{\theta}_1[m_3(l_{g3}^2 + l_1 l_{g3} C_{23} + l_2 l_{g3} C_3) + I_3] + \ddot{\theta}_2[m_3(l_{g3}^2 + l_2 l_{g3} C_3) + I_3] + \ddot{\theta}_3[m_3 l_{g3}^2 + I_3] + m_3 l_1 l_{g3} S_{23} \dot{\theta}_1^2 + m_3 l_2 l_{g3} S_3 (\dot{\theta}_1 + \dot{\theta}_2)^2 + m_3 g l_{g3} C_{123} \tag{18}$$

2.3. Dynamic simulation

The Simulink graphical programming environment is used for simulating the dynamic response of the exoskeleton when a variable torque is applied to every joint. **Figure 7** shows the transient response of the system for this simulation.

3. Communication networks for robotics

The exoskeleton communication network is one of the key design issues to be solved. The main goal of such a network is the proper integration of the exoskeleton mechanism, sensors, actuators, and MPU (Main Process Unit). Several state-of-the-art works describe some details regarding the communication system. The following section presents a review of some relevant and recent published works.

3.1. Related work

CORBYS is a cognitive robotic system architecture proposed for a robot-assisted gait rehabilitation device [7]. Its purpose is to allow the integration of high-level cognitive control modules, a semantically driven selfawareness module, and a cognitive framework for anticipation of human behavior based on biologically inspired information theoretic principles. The main focus of the work is on the low-level real-time physical layer control system. The proposed CORBYS architecture consists of a cognitive, an executive, a control, and physical layers. At the cognitive level, the decision on actuation is carried out evaluating the human state by means of sensor data processing. This level is also responsible for the system model learning and the generation of a reference trajectory. The main objective of this layer is to anticipate human behavior and adapt accordingly the commands sent to the real-time control module. The executive layer is responsible of the communication between all layers and the coordination for the general system functionality. Heartbeat messages are used to verify the proper functionality of each module. The CORBYS modules are deployed on individual computers connected to a general-purpose network. A parameter server is required for storing and distributing application parameters. A real-time control system executes high-level commands from cognitive layer and controls the physical layer actuators. Sensors data are gathered and centralized by a real time data server. These four layers are integrated using a robot operating system (ROS). At the physical layer, smart actuators consist of a brushless DC motor, gearbox, sensors, logic, and power electronics. Additionally, they have three integrated communication protocols: RS-232, CANopen, and profibus DB. The last two are suitable for real-time operation. Regarding smart sensors, authors consider two categories: one for non-real-time applications intended for fatigue, intention/attention to motion sensing. And another one for the real-time control subsystem. Sensors are processed by a local microcontroller and their data is provided digitally through the communication network.

http://dx.doi.org/10.5772/67460

Sensors include magnetic encoders at the orthosis joints and 6-DOF force/torque sensors with data processed by the self-contained hardware. The communication network for the cognitive and executive layers is based on Ethernet, while the control and physical layers use EtherCAT (Ethernet for Control Automation Technology) protocol. This is a real-time industrial Ethernet fieldbus with a data rate of 100 Mbps [8], optimized for short cyclic process data (≤10 μs), low jitter for accurate synchronization (≤1 μs) and low hardware costs [9]. Since the sensors and actuators of the system are not EtherCAT enabled, the authors of CORBYS have designed a dedicated low level control module to act as a gateway and as a local controller at the same time.

Authors in [10] propose a lower limb exoskeleton for physical assistance and rehabilitation. The purpose of the system is to provide powered assistance in the sagittal plane at both hip and knee joints. It essentially consists of a motorized leg device with four degrees of freedom including hip, knee, and ankle. The system is a combination of an exoskeleton robot for the legs and an external supporting end-effector robot for the pelvis. The exoskeleton sensors are Hall effect and high-resolution encoders of 2048 pulses per cycle in each joint. It also includes a custom distributed embedded driven system. The actuation elements are Maxon dc flat brushless motors. The proposed architecture includes a four-layer control scheme as exoskeleton, distributed embedded system, servomotor, and control computer. Specifically, the robot control architecture which interfaces the control algorithm with the robot, consists of the exoskeleton, sensors, and actuators; a control unit for real-time algorithm execution, which includes the acquisition card for interface with sensors, actuators, and the controller; and a host computer, which runs an application for doctor to user interface. The main functions of the control system are, in first place, to gather the position, angular velocity, and acceleration signals needed for control, data collection, and evaluation. This information is used to generate the required signal to activate the driven motors. A low-level driver that is connected to a Windows XP PC controls the non–real-time tasks. The interface between this WinXP controller and the active orthosis is based on a controller area network (CAN). Specifically, the communication is done through a wired CAN bus using CANOpen protocol and a dynamic link library developed by the authors. During the execution of the control strategy, this interface is used for real-time signals monitoring and controller parameters tuning. The main objective is that the user should be able to track any continuous desired trajectory assisting the knee and angle movement. For this, sensors attached to the limb measure its motion, an error-canceling algorithm performs a real-time discrimination of the undesired motion components, and error information is the input for the controller in order to generate the desired actuator action to minimize the error.

In [11], with the purpose of enhancing the real-time movement detection of a lower limb exoskeleton, authors propose an inertial sensor network based on CAN. In order to acquire the inertial signals, these sensor nodes are distributed over the lower limb, trunk and waist of the exoskeleton. All the inertial sensors include three-axis gyro, accelerometer, and magnetometer together with an ARM microprocessor and the CAN communication module. The microprocessor is responsible for signal conversion and filtering, and data merging. The complete body area network is built by means of the CAN network to transfer the sensors data to a central perception node. A work station for data analysis receives the data from the central perception node and records the exoskeleton postures. In more detail, the CAN bus network gather and integrate the data from eight sensor nodes placed on each left and right foot, calf,

thigh, on the waist, and on the back of the exoskeleton system. The communication protocol is based on CAN 2.0a, which frames a structure that defines an 11-bit identifier. Among other, messages could be remote frames, data, or error frames. In order to improve the real-time performance, the medium access control for the shared bus used in this work is a collision free one. Specifically, it is based on time division multiplexing (TDM), i.e. tCAN, and so, each sensor has available an exclusive time slot for data transmission. The required synchronization is carried out by means of a special synchronous message and alternatively by the local clock in each sensor node. Authors define the perception cycle to be 10 ms divided into 10 time slots (one for synchronization message, one for back time slot, and eight for each sensor). Authors claim that using this protocol reduces the data load by 16.7% regarding a system that transmits the original raw sensor data.

A lower limb rehabilitation exoskeleton suit which includes biomimetic framework, perception sub-system, controller, electrical motor, driving system, and power supply subsystem is presented in [12]. The perception system includes sensors (pressure, gyroscopes, encoders, and pressure switches) and a CAN-based communication network. Data from the sensors is gathered by any of the seven sensor node microsystems and are fundamental to obtain movement position and perceive movement intention. The CAN communication bus uses a single-line topology. The master node consists of an embedded controller that uses a CAN bus adapter for network connection, while sensors transmit their information directly to the bus.

A bed-type lower limb rehabilitation training robot consisting of a robot bed and an exoskeleton is presented in [13]. The perceptual system is a multisource information perception system, which includes the detection methods of bioelectric (EEG, EMG, oxygen) and physical (joint angles, interaction forces) signals and the integration of a sensor network based on a multiagent system. The integration between the several control softwares and the variety of data and different communication protocols is done by means of a multi-agent software system. A modular design has been selected for the subsystems and the communication between them is based on TCP/IP. Data communication is done in an asynchronous way and the heterogeneous devices are integrated by means of an Ethernet protocol.

As an alternative to a wired-based exoskeleton communication network, authors in [14] propose a wireless remote control arm exoskeleton intended for upper limb rehabilitation. The robot system can be remotely controlled by means of a ZigBee-based network [15] which is implemented with low energy consumption XBee modules. The physiotherapist controls and receives feedback information by means of a LabVIEW-based human machine interface.

Under these premises, and regarding the exoskeleton communication network, our future line of work will focus on the performance comparison between wired and wireless solutions.

3.2. Architecture

Figure 8 shows the architecture of the communication system. It has one main process unit (MPU) and six sensor and actuator units (SAU). The MPU components are a microcomputer, an accelerometer, a module of biological signal acquisition (EEG and EMG), and a communication device. A motor, an encoder, a communication device compose the SAU, and the SAU+, additionally, has two strain gauges at the ankles.

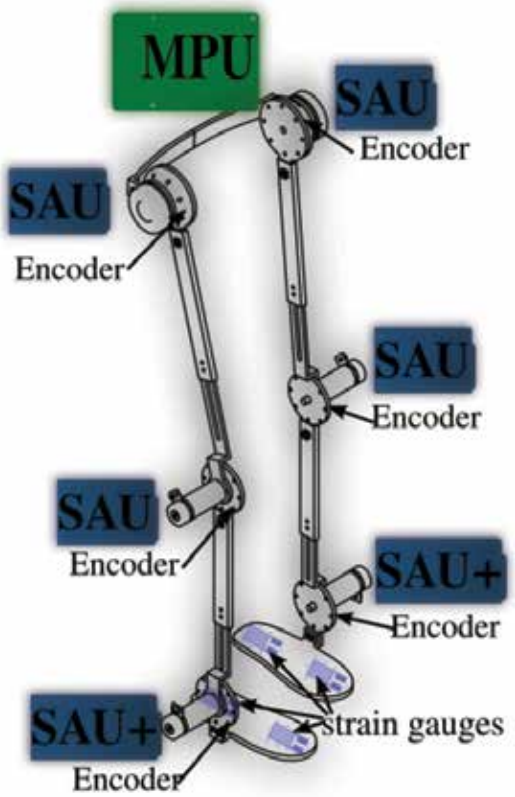

Figure 8. Architecture of the exoskeleton system.

The MPU performs the control of the exoskeleton using the data generated by SAUs. The SAUs implements the signal acquisition of the encoder and the strain gauges (if SAU is located at the ankle), the communication (transmission and reception), and the actuation of the control performed by the MPU.

The microcomputer used in the MPU is an embedded device with enough processing power to carry out the control task. Different options available commercially are Raspberry Pi, BeagleBone, Odroid, etc., but in the market, some other options exist.

The use of communication devices is to reduce the wiring in the exoskeleton. The communication device in the MPU has two interfaces and in the SAU has one interface. The bus network topology is used for the communication. Each communication interface in the MPU operates one leg.

3.3. Communication protocol

The selected protocol for the required real time communications is CAN. It is a serial communications protocol. The advantage of this protocol is the support of real time control with high level of reliability that is a necessary feature for the exoskeleton system. It supports the real time communications offering a guaranteed maximum latency.

There exist two versions of CAN, and the last one was standardized in 2015 in ISO 11898-1. The chosen version is the newest one, version 2.0. CAN standard defines a multicast-based communication protocol originally intended for automotive industry. Currently, its applications include factory and plant controls, robotics, medical devices, and avionics systems [16]. CAN is a broadcast digital bus that supports data rates from 20 kbps to 1 Mbps depending on the bus length and transceiver speed. The longest bus length of the exoskeleton would be 40 meters, for whose case a speed of 1 Mbps is selected. The selection of a CAN-based communication network was mainly motivated by its reliability and deterministic operation. Such behavior is based on node priorities, which depends on the node identification (Node ID). The node with the lowest ID value is the one with the highest priority. If two or more nodes attempt to simultaneously transmit over the shared bus, the node with the lesser ID will automatically get the access to the bus. In this way, there is no need of master or slave CAN devices. The version CAN 2.0a uses identifiers of 11 bits (standard frame) and the version CAN 2.0b uses identifiers of 29 bits (extended frame). The exoskeleton system requires seven units then the version CAN 2.0a can be used.

3.4. Data types

The three types of units (MPU, SAU, and SAU+) generate data. MPU send actuation data about the position in degrees of the motors, and SAUs and SAUs+ send sensor information from encoders and strain gauges. All units send information in the *data* field of a CAN 2.0a frame, shown in **Figure 9**. The *data* field in the CAN frame can be from 0 to 64 bits, which means a maximum of 8 bytes can be send by frame.

The data transmitted by the MPU is basically the position in degrees of the motors located at the two SAUs and the SAU+ of each leg. The MPU transmits two MPU frames (**Figure 10**), one by each leg. The total size of the MPU frame is 24 bits (3 bytes). This frame is received by the SAU and SAU+. MPU frame is broadcast by the MPU in the bus. SAU and SAU+ read the same frame. The first eight bits are processed by SAU located at the hip, the next eight bits are processed by the SAU located at the knee, and the last eight bits are processed by the SAU+, which is located at the ankle.

SAUs send a data field of eight bits, which has been called SAU frame. The SAU frame (**Figure 11**) sends information about position of motor in degrees, which it gets from encoders. This data is collected by an MPU to perform the control. SAUs+ frame (**Figure 11**) has, additionally, one byte by each strain gauge; there are two gauges by leg. The total size of data field in this unit is 24 bits. Each unit has a slot time of the bus to transmit their data.

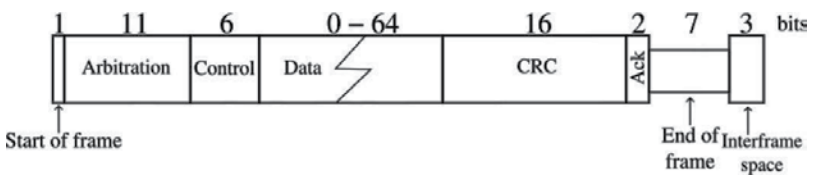

Figure 9. CAN 2.0a frame.

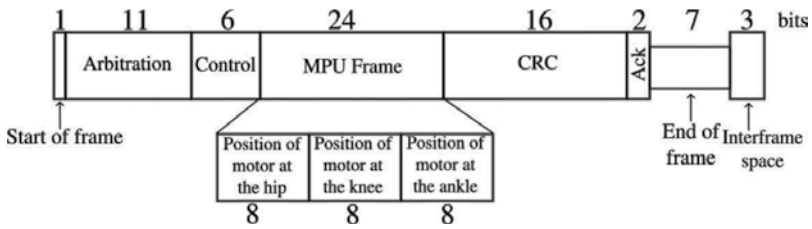

Figure 10. MPU frame.

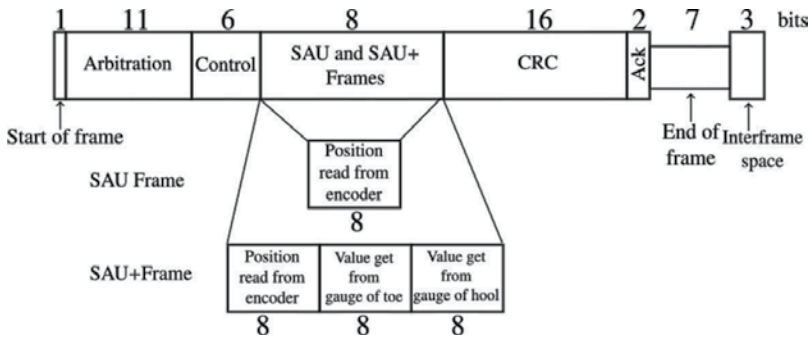

Figure 11. SAU frame.

4. Patient-exoskeleton interaction system design

Since a robotic structure such as an exoskeleton comprises a human component, the amount of interaction the individual may have with it is directly related to his impairment degree. A patient wearing an exoskeleton becomes part of it, and the way he interacts with such device must be considered as a nuclear part of the whole system. Since the ability of walk depends on the impairment degree of the subject, exoskeletons are built so that they may work independently as isolated structures with autoequilibrium mechanisms where the patient becomes only a passenger, or they can be built as a complementary structure where the patient controls how the exoskeleton must behave. We consider a patient having partial leg impairment, such that he can participate his motion intention to the exoskeleton by trying to walk.

4.1. Signal acquisition and feedback

An exoskeleton structure comprises many input signals related to two main features: first, *structural feedback*, which is mainly related to the actual movement and relative position of the mobile parts of the exoskeleton; and secondly, *human feedback*, which is related to the biological electrical signals generated by the nervous system because of the exoskeleton action on human body and due to the subject intention to move. While the first set of signals can be obtained using rotation encoders, torque sensors, force sensors, etc., the latter set of signals must be acquired directly from the subject who wears the exoskeleton. For instance, to register the electrical activity generated within muscles, electromyographic signals (EMG)

are retrieved by implementing wired electrodes located directly on the subject's skin, specifically, on the muscles whose parts are related to the exoskeleton movement such as laps, hips, etc. Movement intention may be guessed by analyzing electroencephalogram signals (EEG). Brain-computer interfaces are regularly used to allow computers to acquire brain activity related to actual body translation or motion intention. Some external interfaces may also be used to recognize human motion intention because of their suitability in being quickly to put on and the easiness to get used to control them. Among many proposals found in the literature, joystick-like devices that are easy to move stand out as the most relevant. Normally, these kinds of devices take advantage of the body parts that are fully controllable by partially impaired persons, such as the eyes blinking, tongue movement, hands, arms, shoulders, etc. These devices usually take advantage of electromyographic signals that can be captured from muscles to generate linearly separable patterns that are usually associated to a limited set of motion functions. Basic motion function commands are: go forward, go backward, sit down, stand up, stop, etc.

There are many other sources of information that can be leveraged towards helping impaired persons to regain the walking independence ability. For instance, the use of cameras and ultrasonic sensors has become more frequent among researchers and computer vision enthusiasts mainly due to the development of high computational power reached in recent years. Although their study has gained attention in recent years, their analysis goes beyond the scope of this book chapter. The interested reader, however, may want to look to [17–20].

4.2. EMG interface and signal conditioning

Electromyographic signals (EMG) represent the electrical activity found in body muscles related to motions and sensations. These signals travel from the brain to muscles and vice versa through the spinal cord to control body movement. Normally, acquired EMG signals are converted into muscle forces variables that are introduced into the total exoskeleton control strategy.

Several authors have proposed the use of EMG signal in the implementation of lower body exoskeletons. Human motion intention is clearly reflected on the legs muscles.

EMG signals normally vary very slowly with frequency, and then, it is possible to acquire them using sampling rates of 512–1000 samples per second. As usual, these signals are also low-pass filtered to remove high frequency and noise components. More details in conditioning EMG signals can be found at [21].

The use of EMG signals in controlling exoskeleton structures emerged as the natural intuition of having signals that although being present in a muscle (the actuator) are not able to generate movement. Then, the basic idea is to take advantage of these residuary motion electrical signals to ignite external mechanisms based on electrical motors, pneumatic pistons, etc.

In [22], EMG signals are used to control a lower body exoskeleton and test their system on the right leg of a subject. The authors located differential electrodes on the rectus femoris, vastus lateralis, and semitendinosus muscles to extract information related to knee flexion. Signals are acquired with a 12 bit resolution analog to digital converter (ADC). Moreover, they explore two different control methods: one based on a human body model that is used to determine actuator controllers' inputs; and a second method in which they convert EMG signals into forces that are

compared with forces present in the exoskeleton such that a difference can be found and used as the controller input. The work presented in [23] describes the use of EMG signals and Bayesian information criteria (BIC) combined with well-known feature extraction techniques aimed to feed a linear discriminant function that can effectively classify eight distinct stages of gait.

In [24] authors use surface electromyography (sEMG) signals and a least squares support vector machine classifier in order to recognize six different motion patterns. The feature extraction phase involves the use of the wavelet transform and two different decomposition substages that yielded results with very high accuracy. Once sEMG signals are classified, they can be easily linked to a specific motion pattern of the exoskeleton. Another control strategy based on the use of EMG signals is presented in [25].

4.3. EEG interface and signal conditioning

As many other functions of the human body, motion originates in the brain. Electroencephalogram signals (EEG) reflect brain activity; therefore, they are normally considered whenever it is necessary to associate some human experience with a determined pattern. Although thoughts of moving a specific body part for trivial activities such as shaking hands or playing video games are caused by human will, there are many activities that imply body motion that seem to be generated unconsciously. Nobody thinks about how to walk unless he may seem to be interested in doing so in a very specific way other than the one he does on a regular basis. Nonetheless, muscles react to an order that comes from the brain, which indicates that the human structural system cinematic components must work harmonically together to move the body to another location. We may want to think about taking a glass from a table; however, we do not really care about the actual path of our hand heading from wherever it is located to the actual position of the glass. Moreover, certain translation activities rely heavily on our natural feedback sensors of sight and equilibrium.

EEG signals are retrieved by brain-computer interfaces, which can use invasive and non-invasive methods to register electrical brain activity. The latter methods are preferred since they do not require to introduce estrange objects into the human body such as electrodes. Tests in animals have proven that although the insertion of physical electrodes into the brain may be a suitable way to obtain and to introduce electrical signals in the brain, the involved risks are still too high, mainly since long term effects and possible brain damage in humans are still unknown factors. Therefore, non-invasive methods involving devices that require only installing a set of small electrodes on the head have gained the attention of researchers and have reported preliminary satisfactory results that did not require surgery of physical invasion of the brain. Devices such as the Mindwave from Neurosky, or the Emotiv Epoc have shown to be effective ways to acquire human brain activity [26]. As the complete understanding of the brain functioning is still in development, we are currently transiting a path of discoveries and research guided by experimentation.

4.3.1. EEG signals

Human brain activity can be acquired through an EEG device. These signals comprise a set of signals in different frequency bands, ranging from 0 to 30 Hz. This bandwidth is separated in five well-known kinds of waves called correspondingly α, β, γ, Δ, and θ. Due to the very low frequency range, these waves exhibit sampling rates of brain-computer interfaces. BCI varies

from 512 to 1000 samples per second. Normally, a raw signal is acquired and passed through a filter bank that separates the waves of interest.

4.4. Related work

Current brain-computer interfaces require subjects to wear a cap-like device that includes a set of several electrodes located at different positions aiming to capture the activity generated in the brain in relation of what subjects are indeed experiencing. It is well known that specific parts of the head reflect brain activity related to specific activities such as concentration, movement, speech, etc.

Authors in [27] make use of the device Emotiv EPOC to control a robotic arm. They point out that body motion is well known to be reflected on the standardized 10–20 brain-computer interface (BCI) electrodes location; for instance, the left front side of the head with electrode locations AF3, F3, F7, and FC5 effectively reflect brain activity related to the right side of body movement. The general strategy considered here is to use three different feature extraction methods from brain signals: the wavelet packet decomposition transforms to separate the acquired signal into a vector space; the use of the first 10 coefficients of the fast Fourier transform as the relevant components of each EEG derivation and lately, the use of principal components analysis (PCA) to reduce space dimensionality. Later, a classifier based on neural networks is used to determine different subject movement intentions comparing the obtained results given the use of the three feature extraction methods mentioned.

In [28], the authors propose the use of a commercial and auto-balanced exoskeleton that is tested under the use of a BCI as an acquisition device that feeds an artificial neural-based decoder that will determine the user motion intention. They also propose the use of intracortical electrodes inserted into the cranium to study how intracortical networks evolve with the extended use of the exoskeleton.

Authors of [29] reach a 98% of accuracy in predicting motion intention using a BCI of 64 channels, using the standard 10–20 norm. The interface provides signals sampled at 100 Hz, which is wirelessly sent to a computer. They make use of digital filters to obtain the Δ wave using a 200 ms sliding window. Later, they use a local fisher linear discriminant (FLD) classifier.

4.5. General approach

Analyzing a general way to approach the use of EEG and EMG signals in predicting the human intention to move, we may require first to determine the best BCI device considering how invasive it could be, and how easy will it be to wear in a regular basis; secondly, the set of EMG electrodes and their relative location. Most BCI devices will return a raw EEG signal which needs to be filtered to extract the waves of interest. Normally, the use of IIR digital filters is adopted because of its flexibility and the easiness for them to be implemented in software only.

The feature extraction stage is one of the most important phases in the design of a classifier. Current research is focused on the use of domain change strategies such as the use of Fourier, wavelets, discrete cosine transforms, etc. As for the classifier part, neural networks have shown to exhibit good results; nonetheless, researchers are now using modern techniques such as support vector machines (SVMs).

http://dx.doi.org/10.5772/67460

5. Control design

Figure 12 shows hierarchical control architecture for efficient management/interaction of the lower limb exoskeleton with the user. The control architecture proposes three layers of execution, each for specific functionalities:

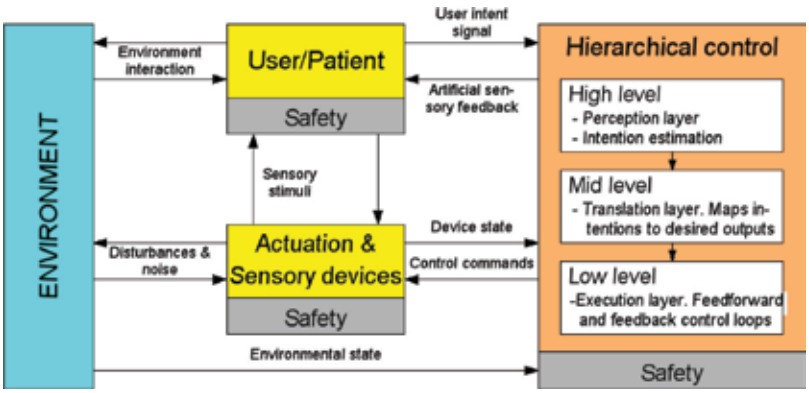

Figure 12. Hierarchical control architecture for efficient control of lower limb exoskeleton.

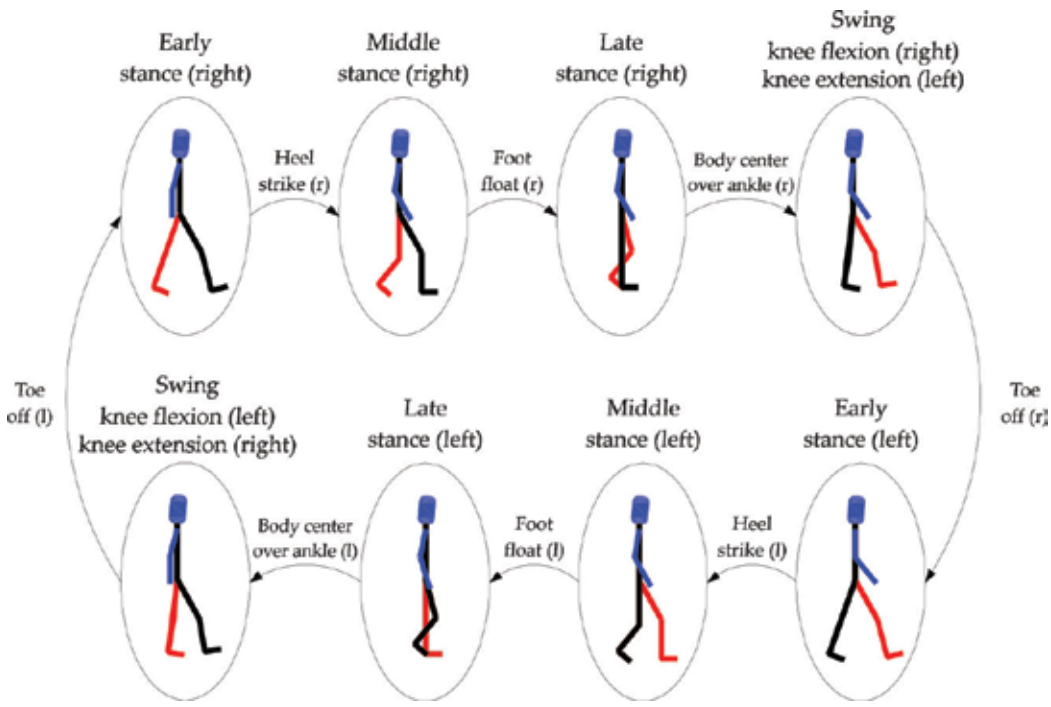

Figure 13. Transition states for gait control in the exoskeleton.

- *High level*: This layer performs an activity mode recognition, which enables the controller to switch between mid-level controllers that are appropriate for different locomotive tasks, such as level walking, stair ascent, standing, hopping, etc.
- *Mid level*: This layer maps the user intentions into orders/setpoints to be sent to local controllers, each installed at every joint of the exoskeleton.
- *Low level*: This layer performs real-time control in each joint by executing feed forward and feed backward control loops.

Several approaches can perform the switching selection among midlevel controllers. The easiest and safe selection for this layer is a discrete-event controller. **Figure 13** shows proposed transition states for such controllers when performing gait routines.

Author details

Luis I. Minchala*, Fabián Astudillo-Salinas, Kenneth Palacio-Baus and Andrés Vazquez-Rodas

*Address all correspondence to: ismael.minchala@ucuenca.edu.ec

Department of Electrical, Electronics and Telecommunications Engineering, Universidad de Cuenca, Cuenca, AZ, Ecuador

References

[1] S. M. Rajesh, "Design of human exo-skeleton suit for rehabilitation of hemiplegic people," Procedia Engineering, vol. 51, pp. 544–553, 2013.

[2] Pons, José L. Wearable Robots: Biomechatronic Exoskeletons, doi 10.1002/9780470987667, 9780470512944, 1729-8806, 1–338, Wiley, 2008.

[3] M. R. Tucker et al., "Control strategies for active lower extremity prosthetics and orthotics: a review," Journal of NeuroEngineering and Rehabiliation, vol. 12, pp. 1, 2015.

[4] N. Li, L. Yan, H. Qian, H. Wu, J. Wu, and S. Men, "Review on lower extremity exoskeleton robot," Open Automation Control Systems Journal, vol. 7, pp. 441–453, 2015.

[5] U. Onen, F. M. Botsali, M. Kalyoncu, M. Tinkir, N. Yilmaz, and Y. Sahin, "Design and actuator selection of a lower extremity exoskeleton," IEEE/ASME Transactions Mechatronics, vol. 19, no. 2, pp. 623–632, Apr. 2014.

[6] T. Yoshikawa, Foundations of robotics: analysis and control. MIT Press, Cambridge, MA, USA, 1990.

[7] V. Grosu et al., "Real-time physical layer architecture for CORBYS gait rehabilitation robot," IEEE International Conference on Rehabilitation Robotics (ICORR), 2015, pp. 606–611.

[8] I. K. Jung and S. Lim, "An EtherCAT based control system for human-robot cooperation," 16th International Conference on Methods Models in Automation Robotics, 2011, pp. 341–344.

[9] EtherCAT, "EtherCAT- the Ethernet Fieldbus," EtherCAT - the Ethernet Fieldbus. [Online]. Available from: https://www.ethercat.org/en/technology.html. [Accessed: 29-Dec-2016].

[10] R. Lu, Z. Li, C. Y. Su, and A. Xue, "Development and learning control of a human limb with a rehabilitation exoskeleton," IEEE Transactions on Industrial Electronics, vol. 61, no. 7, pp. 3776–3785, 2014.

[11] X. Zhang, Y. Shi, and Y. Zhang, "A CAN-based inertial sensor network for lower limb exoskeleton," IEEE International Conference on Communiction Problem-solving, 2014, pp. 473–476.

[12] X. Zhang, Z. Xiang, Q. Lin, and Q. Zhou, "The design and development of a lower limbs rehabilitation exoskeleton suit," ICME International Conference on Complex Medical Engineering, 2013, pp. 307–312.

[13] J. Chen, X. Zhang, and R. Li, "A novel design approach for lower limb rehabilitation training robot," IEEE International Conference on Automation Science and Engineering (CASE), 2013, pp. 554–557.

[14] Y. Bouteraa and I. Ben Abdallah, "Exoskeleton robots for upper-limb rehabilitation," 13th International Multi-Conference on Systems, Signals Devices (SSD), 2016, pp. 1–6.

[15] ZigBee-Alliance, "ZigBee Alliance," [Online]. Available: http://www.zigbee.org/. [Accessed: 29-Dec-2016].

[16] M. Di. Natale, H. Zeng, P. Giusto, and A. Ghosal, Understanding and using the controller area network communication protocol: theory and practice. Springer Science & Business Media, New York, 2012.

[17] S. Qiu, Z. Li, W. He, L. Zhang, C. Yang and C. Y. Su, "Brain–Machine Interface and Visual Compressive Sensing-Based Teleoperation Control of an Exoskeleton Robot," in *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 1, pp. 58–69, Feb. 2017.

[18] K. Kiguchi and M. Liyanage, "A study of a 4DOF upper-limb power-assist intelligent exoskeleton with visual information for perception-assist," IEEE International Conference on Robotics and Automation, 2008, pp. 3666–3671.

[19] T. Tanaka, S. Hori, R. Yamaguchi, M. Q. Feng, and S. Moromugi, "Ultrasonic sensor disk for detecting muscular force," The 12th IEEE International Workshop on Robot and Human Interactive Communication, 2003. Proceedings. ROMAN, 2003, pp. 291–295.

[20] G. S. Sawicki and N. S. Khan, "A simple model to estimate plantarflexor muscle–tendon mechanics and energetics during walking with elastic ankle exoskeletons," IEEE Transactions on Biomedical Engineering, vol. 63, no. 5, pp. 914–923, May 2016.

[21] G. De Luca, Fundamental Concepts in EMG Signal Acquisition. DelSys Inc., 2003.

[22] C. Fleischer, A. Wege, K. Kondak, and G. Hommel, "Application of EMG signals for controlling exoskeleton robots," Biomedical Techiques, vol. 51, no. 5/6, pp. 314–319, 2006.

[23] C. D. Joshi, U. Lahiri, and N. V. Thakor, "Classification of gait phases from lower limb EMG: Application to exoskeleton orthosis," IEEE Point-of-Care Healthcare Technologies (PHT), 2013, pp. 228–231.

[24] H. Wang, T. Mu, H. Li, and X. Zhang, "Research on surface EMG based accurate perception method for exoskeleton robot control," IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2015, pp. 1900–1905.

[25] D. Chen, M. Ning, B. Zhang, and G. Yang, "Control strategy of the lower-limb exoskeleton based on the EMG signals," IEEE International Conference on Robotics and Biomimetics (ROBIO 2014), 2014, pp. 2416–2420.

[26] F. Torres et al., "Adquisición y análisis de señales cerebrales utilizando el dispositivo MindWave," Maskana. Revista Científica, 2014.

[27] H. A. Shedeed, M. F. Issa, and S. M. El-sayed, "Brain EEG signal processing for controlling a robotic arm," 8th International Conference on Computer Engineering Systems (ICCES), 2013, pp. 152–157.

[28] J. L. Contreras-Vidal and R. G. Grossman, "NeuroRex: A clinical neural interface roadmap for EEG-based brain machine interfaces to a lower body robotic exoskeleton," in 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2013, pp. 1579–1582.

[29] A. Kilicarslan, S. Prasad, R. G. Grossman, and J. L. Contreras-Vidal, "High accuracy decoding of user intentions using EEG to control a lower-body exoskeleton," in 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2013, pp. 5606–5609.

Chapter 7

Design and Motion Control of a Lower Limb Robotic Exoskeleton

Ümit Önen, Fatih M. Botsalı, Mete Kalyoncu,
Yusuf Şahin and Mustafa Tınkır

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/67458

Abstract

This chapter presents the results of research work on design, actuator selection and motion control of a lower extremity exoskeleton developed to provide legged mobility to spinal cord injured (SCI) individuals. The exoskeleton has two degrees of freedom per leg. Hip and knee joints are actuated in the sagittal plane by using DC servomotors. Additional effort supplied by user's arms through crutches is defined as user support rate (USR). Experimentally determined USR values are considered in actuator torque computations for achieving a realistic actuator selection. A custom-embedded system is used to control exoskeleton. Reference joint trajectories are determined by using clinical gait analysis (CGA). Three-loop cascade controllers with current, velocity and position feedback are designed for controlling the joint motions of the exoskeleton. A non-linear ARX model is used to determine controller parameters. Overall performance and an assistive effect of WSE-2 are experimentally investigated by conducting tests with a paraplegic patient with T10 complete injury.

Keywords: exoskeleton, legged locomotion, motion control, wearable robot

1. Introduction

Paraplegia is impairment in motor or sensory function of the lower extremities. One of the most significant impairments resulting from paraplegia is the loss of mobility. In addition to impaired mobility, the inability to stand and walk entails severe physiological effects, including muscular atrophy, loss of bone mineral content, frequent skin breakdown problems, increased incidence of urinary tract infection, muscle spasticity, impaired lymphatic and vascular circulation [1]. Spinal cord injury (SCI) is the most important reason of paraplegia and

commonly referred to as either complete or incomplete. In a "complete" spinal cord injury, all functions below the injured area are lost, and patient has little hope of functional recovery. An "incomplete" spinal cord injury involves preservation of motor or sensory function below the level of injury in the spinal cord [2]. Most patients with incomplete injuries can recover some functions after successful therapies. Using robots in a rehabilitation process is quite reasonable since the physical therapy consists of time-consuming repetitive movements. Robots can record quantitative measures of improvements in addition to great motion repeatability. Using robotic exoskeletons in physical therapy can provide further motivation to patients by offering more realistic movements. Furthermore, powered exoskeletons can be used as a movement assistant for subjects affected by permanent movement disorders such as complete SCI. Several exoskeletons have been developed for rehabilitation of paraplegic patients. In general, they can be divided into two categories: the first type exoskeletons are based on a gait orthosis and a weight support system in combination with a treadmill [3]. The total weight of orthosis and user is carried by a weight support system. Powered orthosis only applies the force required to complete movements of patient on impaired limbs and move the patient's leg in a normal gait pattern. These systems have some drawbacks. They have a limited workspace and can be used only in clinical environments. Furthermore, these systems hold the patient's pelvis fixed, and this causes the changes in gait kinematics. Although these systems are appropriate for strengthening exercises they do not have any contribution to balancing problem. Lokomat [4–7], LOPES [8–10] and ALEX [11–13] all fall into this category. The second type exoskeletons are ambulatory devices. These systems carry their own weight in addition to user's weight. They offer users limitless workspace. Furthermore, they can be used as a movement assistant in daily activities for subjects affected by permanent movement disorders. These exoskeletons can help to patients for improving necessary motions in order to maintain balance besides the strengthening exercises. Actuator selection and controller design of these exoskeletons are more challenging due to compact design, less power consumption and safety requirements. The following literature review focuses on the second type exoskeletons. Hybrid assistive limb (HAL) [14–16] is developed for the purposes of rehabilitation or living support of people who have disorders in the lower limb and whose legs are weakening. Full body versions of HAL are also developed for heavy labour and rescue support. The lower body model of the device weighs about 12 kg and the full body model weighs about 23 kg. The batteries of HAL's rehabilitation purposed model can provide the necessary power for 60–90 min in normal operation. Hip and knee joints of HAL are powered in sagittal plane Actuators are composed of DC servomotors and harmonic drive gears. A hybrid control system of HAL consists of an autonomous posture controller and a power-assisted controller. The intended motion of the user is determined by using electromyography (EMG) sensors and ground reaction force sensors. The drawback of this control system is that it requires a process of adaptation and adjustment to a specific user. ReWalk [3, 17] has two different models developed for rehabilitation and life support of SCI patients. Hip and knee joints of ReWalk are powered in the sagittal plane by using DC motors. It comprises a wearable brace support suit, which integrates DC motors, rechargeable batteries, sensors and a computer control system. The device is about 20 kg weight. Pre-programmed motion control strategy is used to control of ReWalk. Changes in the user's centre of gravity are used to initiate and maintain walking processes. The user also has a remote control placed in his/her arm for

http://dx.doi.org/10.5772/67458

selecting different tasks, such as sit-to-stand or climbing stairs. The Vanderbilt exoskeleton [18, 19] is developed to provide gait assistance to patients with spinal cord injuries. The device weights about 12 kg. Hip and knee joints are actuated by brushless DC motors. Control is based on postural information measured on the device. A lithium polymer battery of 29.6 V and 3.9 Ah brings 1 h of autonomy for continuous walk with the device at a speed of 0.8 km/h. Ekso is a gait training exoskeleton intended for medically supervised use by individuals with various levels of paralysis or hemiparesis. Ekso weights approximately 20 kg and has a maximum speed of 3.2 km/h with a battery life of 6 h. It can execute sit-to-stand and stand-to-sit operations and walk in a straight line. Ekso uses a gesture-based human-machine interface to determine the user's gestural intentions and then acts accordingly. No studies have been published about Ekso for discussing its efficacy. WSE [20] is developed to support walking of partially or entirely disabled individuals. The total weight of WSE is about 18.5 kg. A 24V DC motors powered by Li-Po battery pack are used in actuation of WSE. The pre-programmed motion control strategy is used to control of WSE. Adaptive network-based fuzzy logic controllers are used to control of joint motions of WSE. Changes in the user's centre of gravity are used to initiate and maintain walking processes. The Li-Po battery pack of WSE can provide the necessary power for about 3 h in normal operation.

In this chapter, mechanical design, actuator selection and controller design of the second generation prototype of WSE (WSE-2) is described. Additional effort supplied by user's arms through crutches is considered in required torque computations in order to realize realistic actuator selection. A non-linear ARX model of the exoskeleton is created and used in order to determine the best controller parameters. The assistive effect of WSE is experimentally investigated. WSE was worn by a paraplegic patient with T10 complete injury during the experiments. The 78 kg weight patient was successfully walked with the speed of 0.5 cycle/s.

2. Mechanical design

Basic working principle of a lower extremity exoskeleton is transferring the user's weight to the ground by creating a force path between the user and the ground. Thus, an exoskeleton eliminates the effects of gravity on user. So, the body of an exoskeleton must be strong enough to carry both its own weight and the weight of the user. But at the same time, it should be as light as possible due to ergonomics, small actuator usage and low power consumption requirements. In addition, critical parts of an exoskeleton should be easily adjusting for adapting to different sized users. The second critical point in the design of an exoskeleton is the choice of actuators. The selected actuators must be capable of meeting the speed and moment requirements for the targeted motions, as well as being compact and lightweight as possible. In addition, an exoskeleton should be able to perform all the targeted movements in a manner that will provide the least inconvenience to the user. In order to achieve this, degree of freedom and range of motion of the joints must be selected properly. In fact, increasing the joint's degree of freedom and range of motion provides more comfortable using. However, releasing all degrees of freedom (DOF) of joints is not safe for the users who have lost their muscular activity in their legs since it causes involuntary movements. For devices in contact

with the user, safety is a very important criterion. Safety measures are more important for exoskeleton applications where the user does not have sufficient mobility. Therefore, all electrical and mechanical safety precautions should be taken into consideration in exoskeleton design for preventing the user from damaging. In this study, it was tried to design a lightweight, ergonomic and safe exoskeleton in accordance with the above-mentioned design criteria.

Design requirements of an exoskeleton can be determined by a prior analysis of human motion since they are required to perform similar tasks with human body. Clinical gait analysis (CGA) is one of the best tools for determining the required degrees of freedom (DOF), joint motions and joint torques of lower extremity exoskeletons. In clinical gait analysis, motions of specific points on the limbs are collected in the form of CGA data via video motion capture. So, all the joint kinematics during a walking cycle can be obtained in the form of CGA data. Then, joint torques which required for a walking cycle can be determined via dynamical equations that include joint kinematics, limb masses and inertias. In this study, CGA data obtained from CGA normative gait database of Hong Kong Polytechnic University [21] is used in mechanical design and an actuator selection process of WSE.

Degrees of freedoms and motions of human extremities are generally defined in standard anatomical planes, as shown in **Figure 1**. Generally, human lower extremities are modelled with 7 degrees of freedom (DOF) (3 DOF at the hip, 1 DOF at the knee and 3 DOF at the ankle) in standard anatomical planes, as shown in **Figure 2**.

The events occurring in each anatomical plane during the walking cycle can be briefly summarized as follows:

Sagittal plane: Extension of the hip joint ensures to advance the body. Flexion of the knee joint ensures the shock absorption at heel contact. Flexion of the knee joint in swing phase shortens the leg length and allows the foot clearance. Dorsiflexion and plantarflexion of the ankle joint during the stance phase ensure a more comfortable walking by moving the foot pressure point from heel to toe.

Coronal plane: Projection of the body's centre of gravity must be stay inside the footprint for a stable walking. In the coronal plane, abduction and adduction of the hip joint slide the centre of gravity for ensuring body to stay inside the footprint. Abduction and adduction of the ankle joint are necessary for adapting foot to rugged terrains.

Transverse plane: Medial and lateral rotation of the hip and the ankle joints ensure body to change the direction of movement.

In brief, movements in the sagittal plane ensure body to move forward, movements in the coronal plane provide to balance body and movements in the transverse plane ensure to change the direction of motion.

The degrees of freedom of an exoskeleton must be in accordance with human anatomy for ensuring a comfortable use. Increasing the degrees of freedom also increases the comfort. But disabled individuals cannot prevent some unintended motions during walking since they have a weak control on leg muscles. So, some degrees of freedom must be locked for the exoskeletons designed the use of disabled users. Although this approach reduces the users

http://dx.doi.org/10.5772/67458

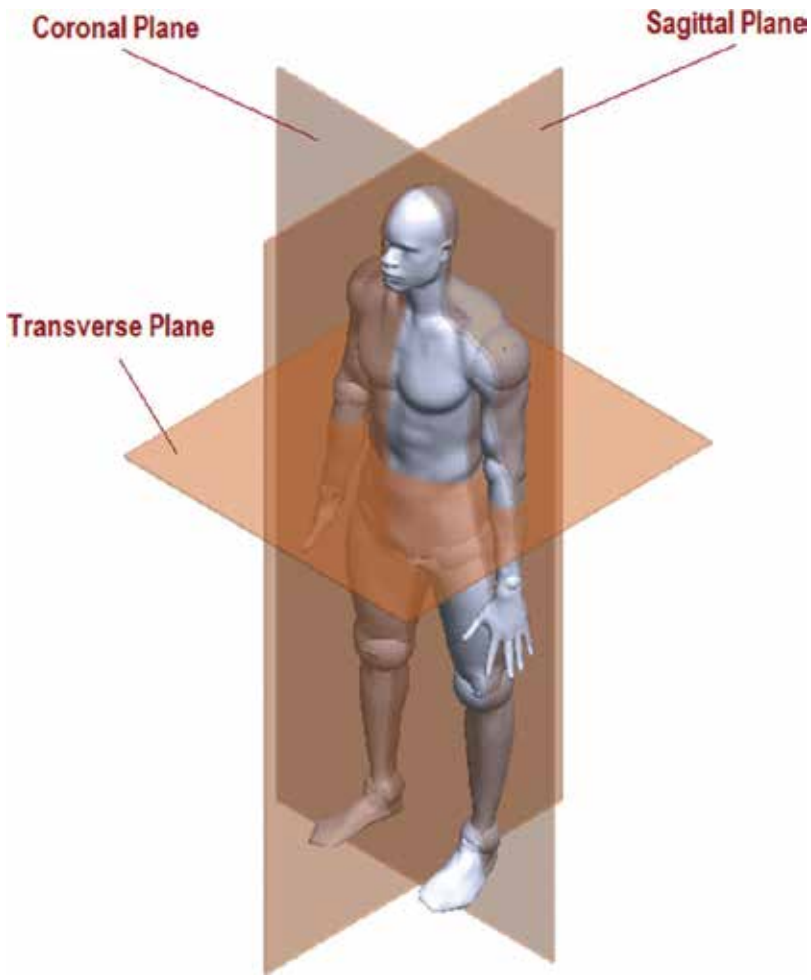

Figure 1. Standard anatomical planes.

comfort, it is necessary to ensure a safe and stable walking. For the user safety, all degrees of freedom of WSE-2 in coronal and transverse planes are locked. But, the degrees of freedom in sagittal plane, which are required to advance the body, are released. In addition, degree of freedom of the ankle joint in the sagittal plane was limited within a certain range. Under these conditions, users are required to use crutches to control the body balance and walking direction since the degrees of freedom of WSE-2 in coronal and transverse planes are locked.

Another point that should be considered in design of an exoskeleton is the determination of proper motion ranges according to targeted movements. Walking is the fundamental movement selected for WSE-2. In addition, WSE-2 should be proper to sitting and standing up motions for daily use. In accordance with these targeted movements, motion ranges of the hip joints are selected as 100° in flexion and 17° in extension, while the motion ranges of the knee joints are selected as 100° in flexion and 0 in hyperextension.

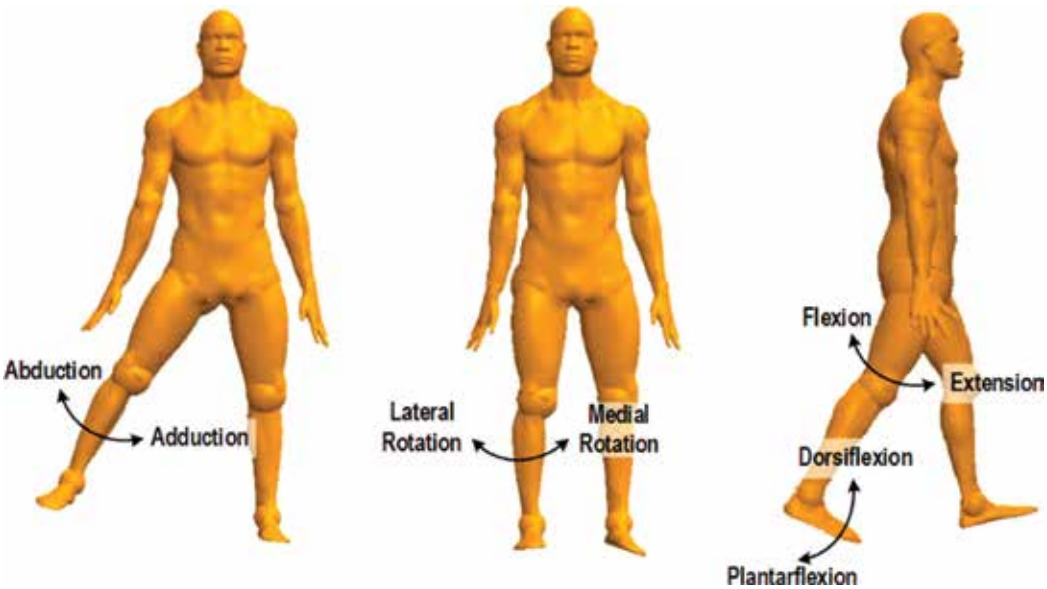

Figure 2. Human degrees of freedom in standard anatomical planes.

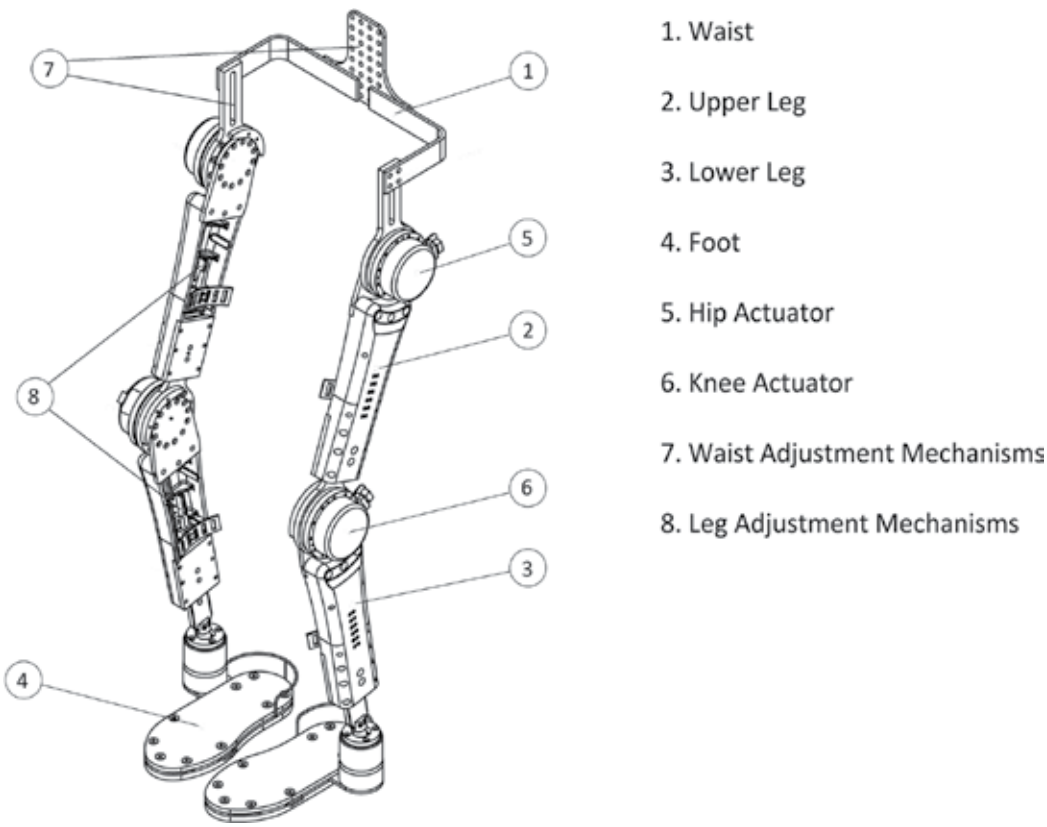

Figure 3. Mechanical design of WSE-2.

Final design of WSE-2 is consisted of waist, upper and lower legs, hip and knee joints and feet, as shown in **Figure 3**. Length adjustment mechanisms of WSE-2 ensure the adjustment of upper legs, lower legs and waist so as to fit to users with different body sizes. In this way, WSE-2 can be comfortably used by the male users whose height is between 1.70 and 1.86 m and by the female users whose height is between 1.67 and 1.86 m. Furthermore, waist adjustment mechanism makes WSE-2 usable for the users of almost all size whatever their weights are. Parts of WSE-2 exposed to light loads were made of polyamide (P6), while the other parts exposed to high loads were made of aluminium 7075. The total weight of WSE-2 excluding the actuators is 12 kg. WSE-2 is designed to have a two-step safety system, both mechanical and electronic for granted the safety of its users. First, proximity limit switches which cut off the power of the actuators in the case of excessive rotation are placed in the joints. Furthermore, mechanical safety apparatuses are designed and mounted on the joints to prevent excessive rotation, in the event of a malfunction in the limit switches.

3. Actuator selection

Actuators used in exoskeleton applications are required to provide high moments while operating in high speeds. This requires the use of larger and heavier actuators. However, the available area around the joint is too limited for connecting the actuators. So, it is necessary to select the most compact actuators that can provide the required moment-speed values for the targeted movements. Velocity-moment characteristics of the joints for targeted movements should be known for the selection of proper actuators. The moment-velocity values of the joints can be determined from the CGA for walking, which is the fundamental targeted movement of WSE. Variation of joint angles and normalized joint moments during a gait cycle is available in CGA data format. Joint velocities appropriated to targeted walking velocity can be computed from the joint angle variations obtained from CGA. Required joint moments for targeted movements can be calculated from the normalized joint moments obtained from CGA by using the relation

$$T_R = T_N . m_T \tag{1}$$

where T_N is the normalized joint torque and m_T is the total mass (user + exoskeleton) carried by the actuators. In case the user supplies an additional effort by using crutches, Eq. (1) should be modified since the user transfers a certain amount of total weight to the ground through the crutches and the total weight carried by the exoskeleton is reduced by a certain ratio during the stance phase. This ratio was defined as "user support rate" (USR) in Ref. [20]. In the presence of additional effort, Eq. (1) can be rewritten as follows:

$$T_R = T_N(m_T . k_s) \text{ Stance Phase} \tag{2}$$

$$T_R = T_N . m_T \text{ Swing Phase} \tag{3}$$

where k_s stands for USR. An experimental study performed for the determination of a user support rate and the calculation of k_s coefficient is given in Ref. [20]. According to the experimental results, the total load carried by the exoskeleton is reduced about 47.7% in the case of

users use crutches. Comparisons of computed angle, torque and power characteristics of hip and knee joints for a walking cycle in cases of USR = 0 and USR = 0.45 are given in **Figure 4**. Masses of the user and the exoskeleton used in calculations are 78 and 18 kg, respectively, and the normalized joint moments are obtained from CGA.

Actuators of WSE-2 are required to provide moment, velocity and power requirements presented in **Figure 4**. Moreover, it should be as compact and lightweight as possible. Maxon EC 90 flat servomotor coupled by CSD series harmonic reducer is selected for the actuation of WSE-2. Technical specifications of selected actuators are given in **Table 1**.

Moment-velocity characteristics of the actuators are compared with the required moment-velocity characteristics of the targeted movements in order to evaluate the suitability of the selected actuator. Required moment-velocity characteristics for two different USR values (0 and 0.45) are determined by using the angle and moment characteristics given in **Figure 4**. Characteristic curves of the selected actuator are obtained from the manufacturer's catalogue. Comparison of the required moment-velocity curves with the characteristic curve of the

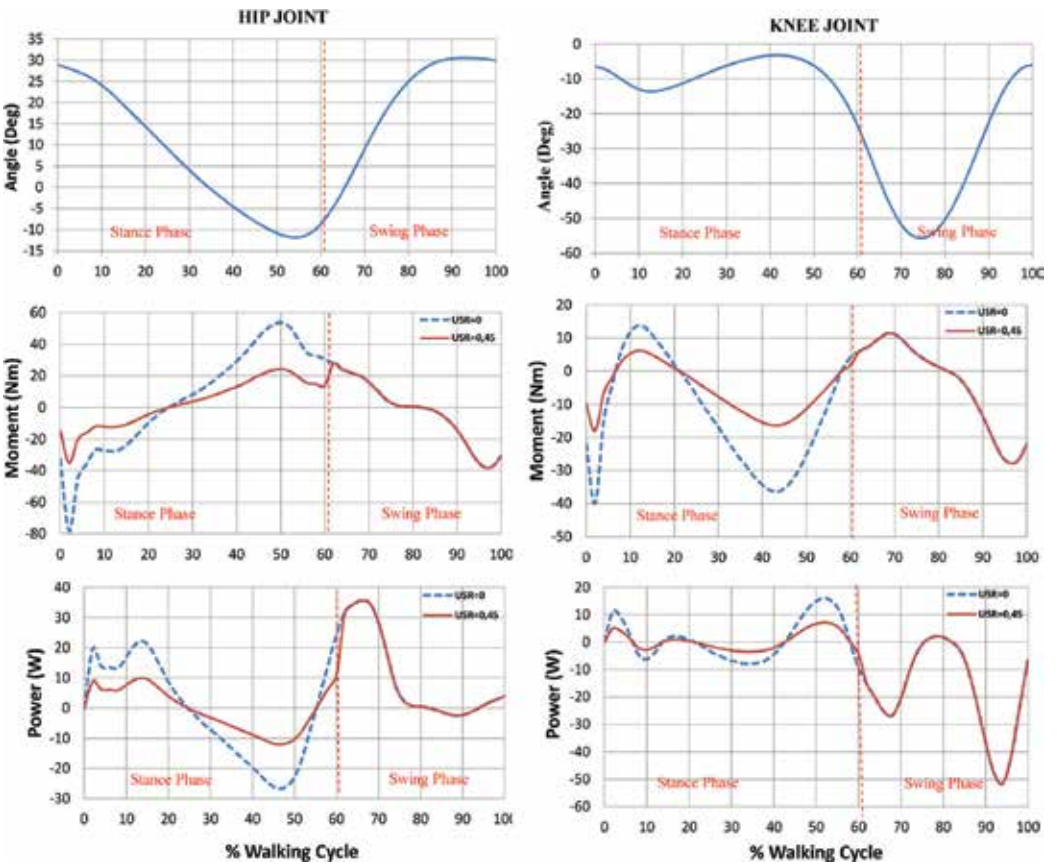

Figure 4. Comparisons of computed angle, torque and power characteristics of hip and knee joints for a walking cycle in the cases of USR = 0 and USR = 0.45.

http://dx.doi.org/10.5772/67458

Servomotor	
Nominal output power (W)	90
Nominal voltage (V)	24
Nominal current (A)	5.39
Nominal torque (Nm)	0.387
Stall torque (Nm)	4.67
Nominal speed (rpm)	2650
Maximum speed (rpm)	3190
Weight (kg)	0.6
Gear	
Gear ratio	100
Moment of inertia (kg m^2)	0.282 (10^{-4})
Weight (kg)	0.91

Table 1. Specifications of selected actuators.

actuator is given in **Figure 5**. The required moment-velocity characteristics are determined by assuming the total weight carried by WSE is 94 kg and the walking speed is 0.5 cycle/s.

As shown in **Figure 5**, while a required moment-velocity curve for the hip joint exceeds the short-term operation limit of the actuators for USR = 0, it stays in the continuous operation region of the actuators for USR = 0.45. On the other hand, a required moment-velocity curve for the knee joint stays in the continuous operation region of the actuators for both USR = 0 and USR = 0.45. Furthermore, maximum moment requirement is decreased about 45% for hip joints and 30% for knee joints in the case of USR = 0.45. Consequently, it is verified that selected actuators can meet moment-velocity requirements of WSE up to 78 kg user weight and 0.5 cycle/s walking speed.

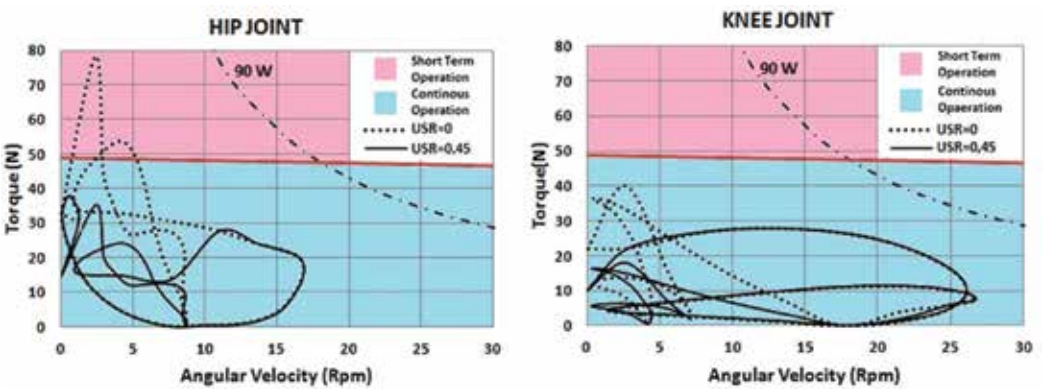

Figure 5. Comparison of the required moment-velocity curves with the characteristic curve of the actuator.

4. Controller design

4.1. Pre-defined motion control (PMC)

Control techniques to be used in control of exoskeletons are generally characterized by the method used in the determination of motion intended by the user. Most of the control techniques used in exoskeletons need interaction signals between the user and device for determining the intention of user. However, majority of paraplegic patients are not capable of generating an effective user-exoskeleton interaction. Another method that can be used in the determination of user's intentions is EMG signals. However, there are technical difficulties in implementing EMG-based control techniques since the EMG signals are extremely noisy and necessitate extensive signal conditioning. A pre-defined motion control (PMC) technique is an alternative method for controlling exoskeletons. In PMC control, the user itself selects the intended motion. The PMC technique bears the advantages of reduced computational complexity, hardware complexity and sensor requirements. So, the PMC technique is selected to be used in control of WSE considering the advantages it offers. In the implementation of PMC, reference motion database which includes the information of sitting, standing up and walking motions is created by using CGA. Cascade PID controllers are designed for motion control and tracking error compensation of hip and knee joints.

4.2. Three-loop cascade control

A control system scheme of WSE-2 is shown in **Figure 6**. Maxon EC 90 flat servomotor coupled by a harmonic drive CSD series harmonic reducer is used in actuation of hip and knee joints. Athena ATHM 800 xPC target compatible PC 104 expandable single board computer combining high integration CPU and high accuracy data acquisition is used in control of WSE. 25.9 V 10 Ah Li-Po battery pack is used as power supply. Limit switches placed at the joints are used to prevent excessive rotations.

Three-loop cascade control structure used in WSE-2 is shown in **Figure 7**. The cascade controller comprises three feedback loops: a current loop, a velocity loop and a position loop. Both the current loop and the velocity loop are the inner sub-control loops, and the position loop is the primary control loop. The current loop is used to limit the current of actuator by keeping it constant under the maximum allowed value at the start and stop, and it optimizes the variation of current. The velocity loop is used to enhance the ability to resist disturbances in load and to suppress fluctuations in velocity. PI type controllers are used in both the current and the velocity loop. The position loop is used to ensure good dynamic tracking performance and static position accuracy. The PD-type controller is used in the position loop.

The choice of controller parameters for a non-linear system is rather complicated. Linearization of a system is the best way of determining controller parameters. But, it is quite difficult to linearize a highly non-linear system. For this reason, it has been decided to use a linearizable non-linear autoregressive exogenous (NARX) model instead of the

http://dx.doi.org/10.5772/67458

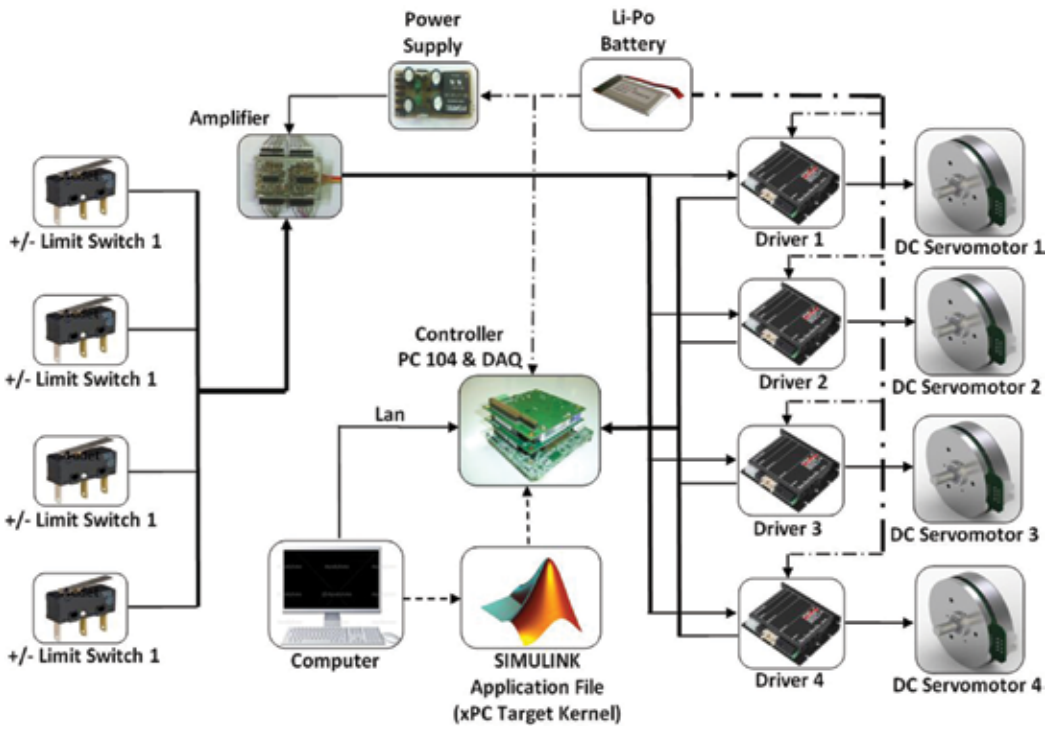

Figure 6. Control scheme of WSE-2.

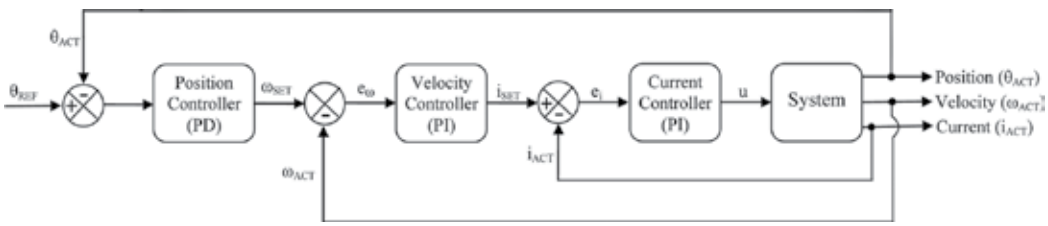

Figure 7. Three-loop cascade control structure of WSE-2.

non-linear model in determination of the controller parameters of WSE-2. Dynamics of hip and knee joints are estimated as an NARX model by using MATLAB system identification toolbox. The NARX model structure given the best performance for system identification of WSE-2 is shown in **Figure 8**. Two-time delayed inputs and outputs are selected as the regressors of NARX models of both hip and knee joints. Sigmoidnet non-linearity estimator with 10 units is used for the hip model while the sigmoidnet non-linearity estimator with six units is used for the knee model. Convergence between the NARX model and the experimental results is 96.58 and 96.78% for hip and knee joints, respectively. Comparisons

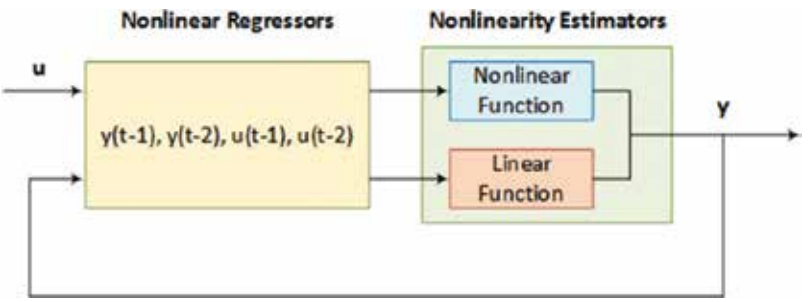

Figure 8. NARX structure of WSE-2.

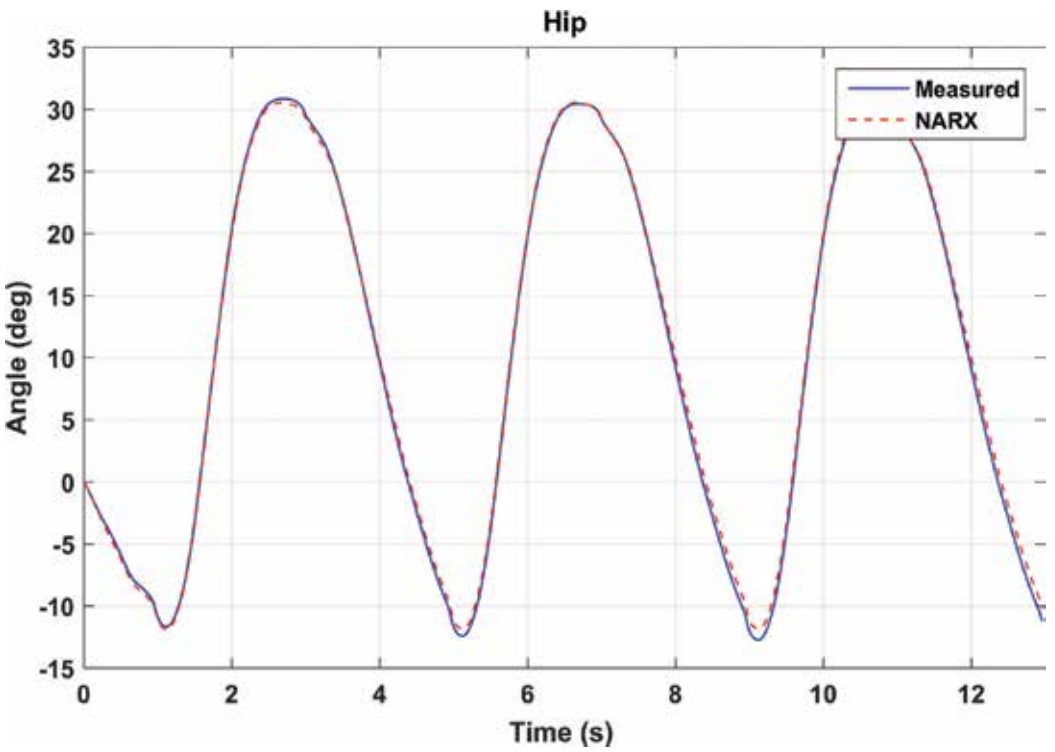

Figure 9. Comparison of the experimental results with the results of NARX model of hip.

of the experimental results with the results of NARX models created for hip and knee joints are given in **Figures 9** and **10**.

Created NARX models are used to determine the parameters of cascade controllers. Controller parameters which given the best simulation results for the hip and knee controllers are presented in **Tables 2** and **3**, respectively.

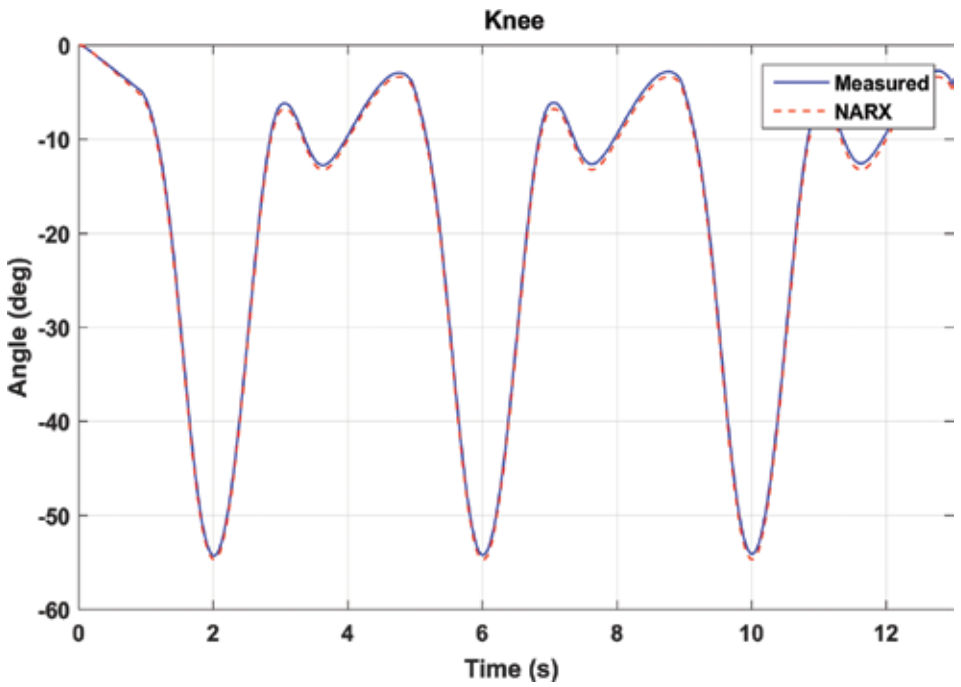

Figure 10. Comparison of the experimental results with the results of NARX model of knee.

Current controller P-gain	215
Current controller I-gain	43
Velocity controller P-gain	13,317
Velocity controller I-gain	1174
Position controller P-gain	0.48553
Current controller sampling period	10^{-6} s
Velocity controller sampling period	10^{-3} s
Position controller sampling period	10^{-3} s

Table 2. Controller parameters for hip actuators.

Current controller P-gain	239
Current controller I-gain	42
Velocity controller P-gain	14,986
Velocity controller I-gain	1217
Position controller P-gain	0.50786
Current controller sampling period	10^{-6} s
Velocity controller sampling period	10^{-3} s
Position controller sampling period	10^{-3} s

Table 3. Controller parameters for knee actuators.

5. Experimental results

Overall performances of WSE-2 and cascade controllers are investigated experimentally as shown in **Figure 11**. WSE-2 is worn by paraplegic patient with a T10 complete injury user. During the experiments, a 78-kg weight paraplegic user is walked with a velocity of 0.5 cycle/s by using underarm crutches.

Comparisons of reference and actual joint angles for hip and knee joints are presented in **Figure 12**. Only two strides are given in the figures since the walking cycles repeated itself in the same manner. Measured joint angles are found to be in good conformance with the reference joint angles of corresponding joints. Designed cascade controllers are evaluated to ensure smooth and stable motion despite the disturbances induced by the user.

Figure 11. Performance tests of WSE-2.

http://dx.doi.org/10.5772/67458

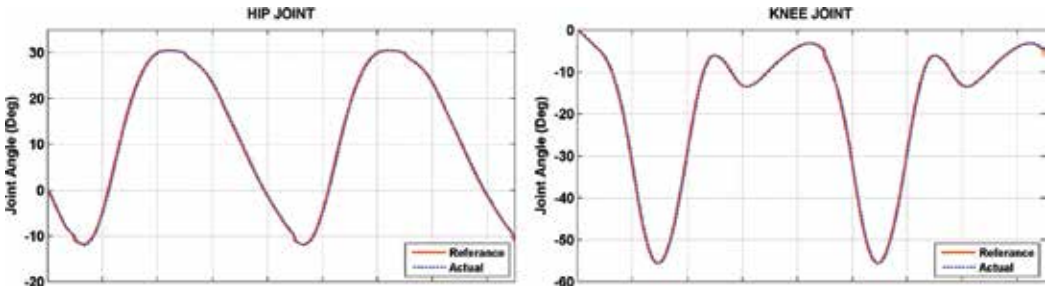

Figure 12. Comparison of reference and actual joint angles.

The experimental results imply that (1) selected actuators are capable of providing the necessary torque and velocity required for a 78 kg weighing complete SCI user walking with a velocity of 0.5 cycle/s, (2) designed cascade controllers provide satisfactory performance in joint tracking control and (3) the assumption that kinematics and dynamics of the exoskeleton is analogous to that of human leg works for WSE-2. Power consumptions of actuators in hip and knee joints are measured during the experiments; power signal is filtered by using a zero-phase digital filter in order to discard possible noise. Plots of filtered and unfiltered actuator power consumption values versus time are presented in **Figure 13**. As seen from figure, the maximum power consumption for hip and knee joints are about 30 and 33 W, respectively. The results show that selected actuators are capable of providing the power requirement for a 78-kg weight paraplegic user walking with a velocity of 0.5 cycle/s.

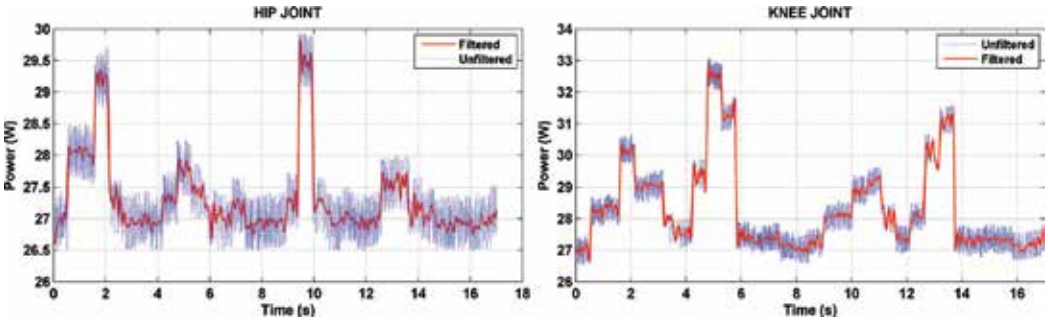

Figure 13. Power consumptions of the hip and knee actuators.

6. Conclusion

Majority of complete SCI patients are not capable of generating sufficient user-exoskeleton interaction required for control action. Subsequently, many control techniques that require user-exoskeleton interaction forces or displacements cannot be used in control of exoskeletons developed for paraplegic individuals. A pre-defined motion control architecture is selected for controlling WSE-2 since it does not require interaction forces or displacements between the user and the exoskeleton. Three-loop cascade-type controllers are designed and used in joint motion control of WSE-2. A non-linear ARX model of WSE-2 is constructed and used to

determine controller parameters. Designed controllers provided good joint angle tracking performance despite disturbances. Consequently, experimental studies conducted with the second generation WSE prototype used by a 78-kg weighing paraplegic user with T10 complete injury showed satisfactory performance in hip and knee joint angle tracking.

Author details

Ümit Önen[1*], Fatih M. Botsalı[2], Mete Kalyoncu[2], Yusuf Şahin[3] and Mustafa Tınkır[1]

*Address all correspondence to: uonen@konya.edu.tr

1 Mechatronics Engineering Department, Faculty of Engineering and Architecture, Necmettin Erbakan University, Konya, Turkey

2 Mechanical Engineering Department, Faculty of Engineering, Selçuk University, Konya, Turkey

3 Mechanical Engineering Department, Faculty of Engineering, Ömer Halisdemir University, Niğde, Turkey

References

[1] Phillips L, Ozer M, Axelson P, and Fonseca J, editors. Spinal cord injury: A guide for patient and family. San Diego, CA: Raven Press; 1987.

[2] Ho CH, Wuermser LA, Priebe MM, Chiodo AE, Scelza WM, Kirshblum SC. Spinal cord injury medicine. 1. Epidemiology and classification. Archives of Physical Medicine and Rehabilitation. 2007;**88**(3):49–54.

[3] Díaz I, Gil JJ, Sánchez E. Lower-limb robotic rehabilitation: Literature review and challenges. Journal of Robotics. 2011;**2011**:1–11/759764. DOI: 10.1155/2011/759764

[4] Hidler J, Nichols D, Pelliccio M, Brady K, Campbell DD, Kahn JH, Hornby TG. Multicentre randomized clinical trial evaluating the effectiveness of the Lokomat in subacute stroke. Neurorehabilitation NeuralRepair. 2009;**23**(1):5–13.

[5] Hidler J, Wisman W, Neckel N. Kinematic trajectories while walking within the Lokomat robotic gait-orthosis. Clinical Biomechanics. 2008;**23**(10):1251–1259. DOI: 10.1016/j.clinbiomech.2008.08.004

[6] Colombo G, Joerg M, Schreier R, Dietz V. Treadmill training of paraplegic patients using a robotic orthosis. Journal of Rehabilitation Research and Development. 2000;**37**(6):693–700.

[7] Colombo G, Joerg M, Dietz V. Driven gait orthosis to do locomotor training of paraplegic patients. Engineering in Medicine and Biology Society. 2000;**4**:3159–3163.

[8] Ekkelenkamp R, Veneman J, Kooij HVD. LOPES: A lower extremity powered exoskeleton. In: 2007 IEEE International Conference on Robotics and Automation ICRA 2007; 10–14 April 2007; Roma, Italy. pp. 3132–3133. DOI: 10.1109/ROBOT.2007.363952

[9] Veneman J, Kruidhof R, Hekman EEG, Ekkelenkamp R, Asseldonk EV, Kooij HVD. Design and evaluation of the LOPES exoskeleton robot for interactive gait rehabilitation. IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2007;**15**(3):379–386.

[10] Veneman J, Asseldonk EV, Ekkelenkamp R, Helm FVD, and Kooij HVD. Evaluation of the effect on walking of balance-related degrees-of-freedom in a robotic gait training device. In: IEEE 10th International Conference on Rehabilitation Robotics; 13–15 June 2007; Noordwijk, Netherlands. pp. 868–875. DOI: 10.1109/ICORR.2007.4428526

[11] Banala SK, Agrawal SK, Kim SH, Scholz JP. Nowel gait adaptation and neuromotor training results using an active leg exoskeleton. IEEE/ASME Transactions on Mechatronics. 2010;**15**(2):216–225.

[12] Winfree KN, Stegall P, Agrawal SK. Design of a minimally constraining, passively supported gait training exoskeleton: Alex II. In: IEEE 12th International Conference on Rehabilitation Robotics, editor. 29 June–1 July 2011; Zurich, Switzerland. pp. 1–6.

[13] Zanotto D, Rosati G, Spagnol S, Stegall P, Agrawal S. Effects of complementary auditory feedback in robot-assisted lower extremity motor adaptation. IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2013;**21**(5):775–786.

[14] Kawamoto H, Sankai Y. Power assist method based on phase sequence and muscle force condition for HAL. Advanced Robotics. 2005;**19**(7):717–734.

[15] Lee S, Sankai Y. Power assist control for walking aid with HAL-3 based on EMG and impedance adjustment around knee joint. In: IEEE/RSJ International Conference on Intelligent Robots and Systems; 30 September–4 October 2002; Lausanne, Switzerland. pp. 1499–1504.

[16] Tsukahara A, Hasegawa Y, Sankai Y. Standing-up motion support for paraplegic patient with robot suit HAL. In: IEEE 11th International Conference on Rehabilitation Robotics; 23–26 June 2009; Kyoto, Japan. pp. 211–217.

[17] Esquenazi A, Talaty M., Packel A, Saulino M. The rewalk powered exoskeleton to restore ambulatory function to individuals with thoracic-level motor-complete spinal cord injury. American Journal of Physical Medicine & Rehabilitation. 2012;**91**(11):911–921.

[18] Farris RJ, Quintero HA, Goldfarb M. Preliminary evaluation of a powered lower limb orthosis to aid walking in paraplegic individuals. IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2011;**19**(6):652–659.

[19] Farris RJ, Quintero HA, Murray SA, Ha KH, Hartigan C, Goldfarb M. Preliminary assessment of legged mobility provided by a lower limb exoskeleton for persons with paraplegia. IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2014;**22**(3):482–490.

[20] Onen U, Botsali FM. Kalyoncu M, Tinkir M, Yilmaz N, Sahin Y. Design and actuator selection of a lower extremity exoskeleton. IEEE/ASME Transactions on Mechatronics. 2013;**19**(2):623–632.

[21] C. Kirtley, CGA Normative Gait Database, Hong Kong Polytechnic University. Curtin University [Internet]. Available from: http://guardian.curtin.edu.au/cga/data/ [Accessed: 2008]

Edited by Sahin Yildirim

This book can serve as a reference resource for those very same design and control engineers who help connect their everyday experience in design with the control field of mechatronics. This book also consists of basic and main mechatronic system's laboratory applications for use in research and development departments in academia, government, and industry, and it can be used as a reference source in university libraries. It can also be used as a resource for scholars interested in understanding and explaining the engineering design and control process and for engineering students studying within the traditional structure of most engineering departments and colleges. It is evident that there is an expansion of mechatronics laboratories and classes in the university environment worldwide.

ISBN 978-953-51-4853-1
9 789535 148531

Photo by AntonMatveev / iStock

IntechOpen