

## Research Article

# System Identification and Embedded Controller Design for Pneumatic Actuator with Stiffness Characteristic

**Khairuddin Osman,<sup>1,2</sup> Ahmad 'Athif Mohd Faudzi,<sup>2,3</sup> M. F. Rahmat,<sup>2</sup> and Koichi Suzumori<sup>4</sup>**

<sup>1</sup> Department of Industrial Electronics, Faculty of Electronic and Computer Engineering, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia

<sup>2</sup> Department of Control and Mechatronics Engineering, Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 Skudai, Johor Bahru, Malaysia

<sup>3</sup> Centre for Artificial Intelligence and Robotics, Universiti Teknologi Malaysia, 81310 Skudai, Johor Bahru, Malaysia

<sup>4</sup> Graduate School of Natural Science and Technology, Okayama University, Okayama, Japan

Correspondence should be addressed to Ahmad 'Athif Mohd Faudzi; athif@fke.utm.my

Received 19 December 2013; Revised 6 March 2014; Accepted 28 March 2014; Published 2 June 2014

Academic Editor: ShengJun Wen

Copyright © 2014 Khairuddin Osman et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents model and controller design applications to pneumatic actuator embedded system. Two model strategies of position and force are proposed to realize compliance control for stiffness characteristic. Model of the pneumatic actuator system (transfer function) is obtained from system identification (SI) method. Next, combination of predictive functional control with observer (PFC-O) design is selected as a new control strategy for pneumatic system. Performance assessment of the controller is performed in MATLAB and validated through real-time experiments using national instrument (NI) devices and programmable system on chip (PSoC) microcontroller. Result shows that the new controller is adapted to the system and able to successfully control both simulation and real-time experiments.

## 1. Introduction

Pneumatic control systems are widely explored in research and development (R&D) activities by researchers and industry as they offer advantages such as easy and simple maintenance, relatively low cost, self-cooling properties, good power density (power/dimension rate), fast acting with high accelerations, and installation flexibility [1]. The purpose of a pneumatic control system is to solve problems of many nonlinear characteristics such as valve dead zone problems, mass flow rate parameters, and compliance variation. However, pneumatic system is getting more complex, intelligent, and difficult to model especially using mathematical calculation for validation process. System identification (SI) is used to solve system modelling and unknown parameters and linearizes the system from mathematical model drawback. System identification also can obtain the linear mathematical model (transfer function) of the plant system from the measured experimental data. Zadeh presented that there are

a multitude of identification process techniques that can be utilized [2]. The application area of transfer function has become widespread to cover areas such as engineering, computer science, financial sector, industrial applications, and many others [3].

Various researchers proposed modeling and controller design in pneumatic system including system identification model. System identification not only can model the plant but also can realize online identification and control of pneumatic actuator in a real-time environment [4]. Reference [5] presented a method of identification and controlling electropneumatic servo drives via a mixed-reality environment (MRE). To precisely obtain the system's transfer function can be difficult for nonlinear systems. This causes a great difficulty in servopneumatic system modeling and control. In order to avoid the complexity associated with nonlinear system modeling, a MRE is employed to identify the transfer function of the system using a recursive least squares (RLS) algorithm based on the autoregressive moving-average (ARMA) model.

Online system identification can be conducted effectively and efficiently using the proposed method. Next, the research realized online identification of the pneumatic positional servomechanism necessary to determine the order of the numerator and denominator of the system transfer [6]. This research briefly describes two main parts that constitute an adaptive control. The first part describes the design of an optimal structure of mathematical model that allows for continuous identification. The second part describes the design of an adaptive state-space controller whereby the adaptive control is implemented. Another review on modeling, controller design, and implement system identification to pneumatic actuator has been proposed by various researchers as presented in [7, 8]. The implication of this research is to further improve the performance of existing pneumatic actuators.

Controller design for pneumatic system to control the position, force, compliance, viscosity, and so forth is a challenging issue for improving its tracking performance. Many controller designs were proposed to control pneumatic system such as proportional-integral-derivative (PID), artificial intelligence, and robust controller. Model predictive control (MPC) is one of the controllers that have been successfully used in both industry and academia for the control of large-scale installations, which are typically described by large-scale models with relatively slow dynamics. The key element in MPC is to repeatedly solve an optimization problem based on available measurements of the current state of the process. The advantages of MPC over classic PID control are its ability to steer the process in an optimal approach while taking proactively desired future behavior into account, to tackle multiple inputs and outputs simultaneously and to incorporate constraints. Among the most popular MPC algorithms are dynamic matrix control (DMC), model algorithm control (MAC), generalized predictive control (GPC), predictive functional control (PFC), and so forth [9]. Each controller has their own strategies, advantages, and their specific applications which assured good results. Usually, GPC is widely used in pneumatic systems. However, the system suffers from instability and it is difficult to be implemented on this research real-time embedded system. Industrial applications of PFC can be found in the defense sector, automotive, metallurgical industries, miscellaneous process (chemical reactors and distillation excluded), and so forth [10]. PFC is based on the same approach as all MPC strategies, that is, prediction of the future outputs and calculation of the manipulated variables for optimal control using a simple algorithm. Therefore, PFC is also based on the same principle which uses an internal model, specification of a reference trajectory, and determination of the control law [11]. The research is motivated by the PFC's high-quality control performance with improved rise time, precise tracking, robust stabilization, fast response, and an algorithm that is easy to understand for implementation on a real-time embedded system.

In recent years, interest in exploiting several controller designs on embedded systems has grown. Examples of implementing the controller design for embedded system and their advantage were presented by [9, 12, 13]. The embedded

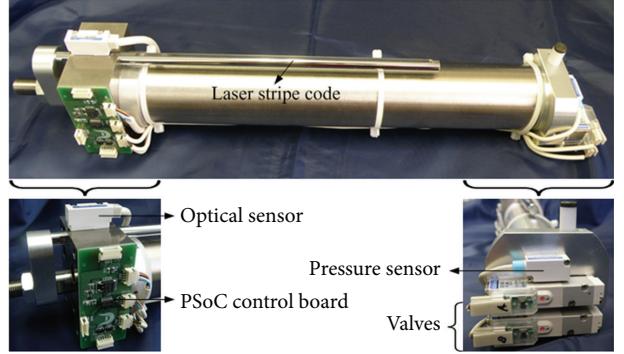


FIGURE 1: Pneumatic system and its parts.

system has been widely applied to manufacturing industry, processing control, communication, instrumentation, vehicle, weapon system, and so forth. In addition, the embedded system is referred to as a dedicated computer application system, which can be adapted into some specialized rigorous requirements to function and power the application system forward. That is to say, it can be centered on the engineering application system, based on computer technology, and its hardware and software can be easily clipped. This allows real-time application by using a chip microprocessor, dSPACE, PC, DSP, and so forth. In these cases the controller designs are not a supervisory controller anymore but directly steer the actuators and as such also the process itself [14, 15].

The related development of the pneumatic system used in this research is presented in [16–19]. Pneumatic system can be further divided into two types of actuator specifically with position accuracy of 0.169 mm and position accuracy of 0.01 mm. The design of actuator with position accuracy of 0.01 mm was enhanced from position accuracy of 0.169 mm for better performance during experimentation on the applications but the system operation for both actuators is still the same. Design with position accuracy of 0.01 mm will have a new position sensor with higher accuracy, new tape type stripe code for better durability, and new enhanced circuit design and will not have been implemented as yet in any application. Figure 1 shows all the parts of the pneumatic actuator at position accuracy of 0.01 mm used in this research. The actuator has 200 mm stroke and can deliver maximum force up to 120 N. KOGANEI-ZMAIR optical sensor is used where smaller pitch of 0.01 mm can be detected. The pneumatic system presents the next generation of actuator development with new features that provide better control, higher position and speed, force accuracy, communication ability, and all-in-one mechanism for compact system design. The pneumatic actuator is equipped with programmable system on chip (PSoC) microcontroller, which acts as the brain for the system and performs the local control to suit the requirements of any related applications. Contraction and extension movements depend on the algorithm to drive the valve using pulse width modulation (PWM) duty cycle. In addition, the pneumatic actuator is able to analyze position, force, stiffness, and viscosity. However, the existing system implements only a simple proportional-integral (PI)

controller design [16–19]. Moreover, there are other disadvantages such as slow response, time delay, overshoot issues and stretch-back may not function with lower stiffness parameters. The main contribution of this paper is to model a pneumatic system by using SI. Of special importance is that the substitute PI controller used and improves the system with the new controller algorithm in embedded system where the accuracy in position, force, and compliance for stiffness characteristics is the main control objective.

The rest of this paper is organized as follows. In Section 2, the model identification technique for this research is described. Section 3 describes the control strategy. Section 4 briefly explains stiffness characteristic. Then, Section 5 describes the embedded controller development. Moreover, Section 6 describes the experimental setup. After that Section 7 presents the analysis of data collection and discussion about system performance. Finally, conclusions and future work are given in Section 8.

## 2. Model Identification

System Identification (SI) technique is proposed to obtain real-time model of the pneumatic system. Two models are proposed, position model, and force model to realize the stiffness characteristic. The plant mathematical models are developed using MATLAB System Identification Toolbox from open-loop input-output experimental data. Through experimental setup, the hardware and Personal Computer (PC) communicate using Data Acquisition (DAQ) card over the MATLAB software. During experimental setup, data will be gathered and analyzed to support system identification model and to observe the system dynamic. The system identification model will go through model estimation, structure selection, and validation for three models. Good parameters identification requires the usage of input signals that are rich in frequencies. There are several methods of generating the signals such as Pseudo Random Binary Sequence (PRBS), sinusoidal, step multi-sine and so forth. For model estimation in position model, square wave input signal is used while pseudorandom binary sequence (PRBS) input signal is used in force models. In this research, a lower sampling time of  $t_s = 0.01$  s is used. It is identified that a smaller sampling time could improve controller performance and more samples can also be taken for system identification process. The PWM generator is designed to mimic the 8-bit PWM modules and the signal amplitude is set to 255 and -255 on the PSoC microcontroller to ease implementation on this platform in the future.

There are few structures of parametric model that can be used to represent certain system. An example are Auto-Regressive with Exogenous Input (ARX) model, Auto-Regressive Moving Average with Exogenous Input (ARMAX) model, Output-Error (OE) model, and Box-Jenkins (BJ) model [20]. There are also other models that are not mentioned in this paper. The plant model is derived from the measured input and output signals of a real plant that needs to be identified. The ARX parametric model structure is chosen for its good result which fulfills the criteria for SI model

after comparison with other model structures. Assuming that noise is zero, the following equation can be derived:

$$\begin{aligned} y(k) + a_1 y(k-1) + \cdots + a_{na} y(k-na) \\ = b_1 u(k-d) + b_2 u(k-d-1) \\ + \cdots + b_{nb} u(k-d-nb+1), \\ \frac{Y(z^{-1})}{U(z^{-1})} = z^{-d} \frac{B(z^{-1})}{A(z^{-1})}, \end{aligned} \quad (1)$$

where  $na \geq nb$ ,  $d$  is time delay,  $na$  is number of poles,  $nb$  is number of zeros,  $u(k)$  is input, and  $y(k)$  is output. A minimum phase model can be obtained using large sampling time whereas the nonminimum phase model can be obtained using small value sampling time [20]. Basically, the models obtained are limited to second and third order only. For example, ARX model will have different structures from lower degree of 2-2-1 structure to high degree of 3-3-1. Higher-order models may produce unstable output. In this case, the third-order model will represent the nearest model of the true plant. After suitable model estimation and structure have been selected, the next procedure is validation. Model validation is to check the validity between the measured data and the desired data under a validation requirement. The simplest validity check is by observing convergence of training errors and assessing the prediction errors for test data. Using part of experimental data that was not used and reserved for model validation purposes, the acceptance or rejection of certain obtained model can be done based on the following criteria using Akaike's final prediction error (FPE) [20, 21]:

$$FPE = V \cdot \frac{(1 + n_a/N)}{(1 - n_a/N)}, \quad (2)$$

where

$$\begin{aligned} V &= \frac{e^T(k)}{N} = \frac{e^T(k) \cdot e(k)}{N} \\ e(k) &= [e_k \ e_{k-1} \ \cdots \ e_{k-N}]^T, \end{aligned} \quad (3)$$

where  $V$  is loss function,  $n_a$  is the number of approximated parameters,  $N$  is the number of samples, and  $e(k)$  is error vector. According to Akaike, the selection of model from various orders can be done based on the smallest value of FPE or Akaike's information criteria (AIC):

$$AIC = \log \left[ V \cdot \left( 1 + \frac{2n_a}{N} \right) \right]. \quad (4)$$

Other than FPE or AIC criterion, best fitting criteria can also be used. These criterions show the preciseness of the approximate model as compared to the true model. This best fit criterion is explained by [20, 21] where model selection will be based on the highest percentage value:

$$\text{fit} = 100 \cdot \left[ 1 - \frac{\text{norm}(\hat{y} - y)}{\text{norm}(y - \bar{y})} \right] \%, \quad (5)$$

where  $y$  is true value,  $\hat{y}$  is approximate value, and  $\bar{y}$  is mean value.

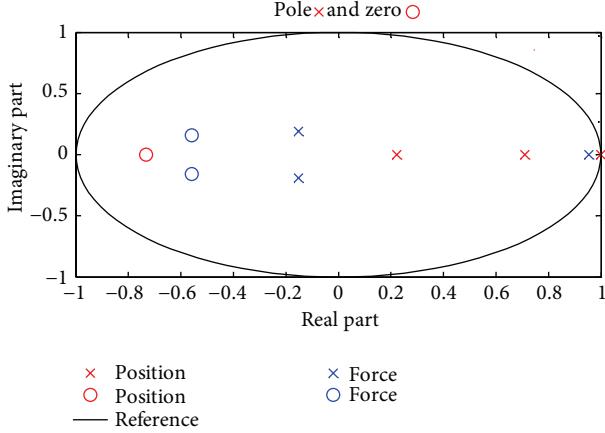


FIGURE 2: Pole-zero plot for the models.

All these processes are done through the System Identification Toolbox in MATLAB. The following discrete-time open-loop transfer functions for position model shown in (6) and force model shown in (7) were identified for third-order system as follows:

$$\frac{B_{\text{Position}}(z^{-1})}{A_{\text{Position}}(z^{-1})} = \frac{0.001269z^{-1} + 0.0004517z^{-2} - 0.0003498z^{-3}}{1 - 1.932z^{-1} + 1.09z^{-2} - 0.1577z^{-3}}, \quad (6)$$

$$\frac{B_{\text{Force}}(z^{-1})}{A_{\text{Force}}(z^{-1})} = \frac{0.04097z^{-1} + 0.04577z^{-2} + 0.01379z^{-3}}{1 - 0.6513z^{-1} - 0.2287z^{-2} - 0.05708z^{-3}}. \quad (7)$$

The criteria for both model validation processes are stable because all the poles of the open-loop discrete transfer function lie within the unit circle of the  $z$ -plane as in Figure 2. Based on the smallest values criteria of AIC and FPE, both models can be accepted. In addition, percentage of best fitting (fit) is more than 90% where the balances are losses because of nonlinear factor such as dead zone, friction, and air leakage.

### 3. Control Strategy

This research proposed the predictive functional control with observer (PFC-O) design for pneumatic system. The formulation of PFC can handle linear and nonlinear processes [10]. Observer design is essential in order to estimate the state of the pneumatic system model. A state observer will provide estimation of the internal state of a given system, from measurements of the input and output of the system.

**3.1. Predictive Functional Control (PFC).** Many literature approaches of PFC and other MPC algorithms are designed

based on the state-space (matrix) form of the plant. The state-space form is preferable for several reasons, easy generalization to multivariable systems and easy analysis of the closed-loop properties, and allows online computation [10, 22]. In this section, the pneumatic model as in (6) or (7) can be converted into state-space form. The PFC algorithm below will explain the main algorithm behind the controller. The general state-space model can be written as in (8):

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k, \\ y_k &= Cx_k + Du_k. \end{aligned} \quad (8)$$

For a prediction with a strictly proper system,  $D = [0]$ :

$$\begin{aligned} x_{k+2} &= Ax_{k+1} + Bu_{k+1}, \\ y_{k+2} &= Cx_{k+2}. \end{aligned} \quad (9)$$

By substituting (8) into (9), the state-space model is written as follows:

$$\begin{aligned} x_{k+2} &= A^2x_k + ABu_k + Bu_{k+1}, \\ y_{k+2} &= Cx_{k+2}, \\ x_{k+3} &= A^2[Ax_k + Bu_k] + ABu_{k+1} + Bu_{k+2}, \\ y_{k+3} &= Cx_{k+3}. \end{aligned} \quad (10)$$

This process is simply an iteration of a one-step-ahead prediction and repeated substitution results can be generalized to

$$\begin{aligned} x_{k+n} &= A^n x_k + A^{n-1}Bu_k + A^{n-2}Bu_{k+1} + \cdots + Bu_{k+n-1}, \\ y_{k+n} &= C[A^n x_k + A^{n-1}Bu_k + A^{n-2}Bu_{k+1} \\ &\quad + \cdots + Bu_{k+n-1}]. \end{aligned} \quad (11)$$

It is clearly seen from (11) that it is possible to convert the state-space model to state prediction equation

$$\begin{aligned} \begin{bmatrix} x_{k+1} \\ x_{k+2} \\ x_{k+3} \\ \vdots \\ x_{k+n} \end{bmatrix} &= \begin{bmatrix} A \\ A^2 \\ A^3 \\ \vdots \\ A^n \end{bmatrix} x_k + \begin{bmatrix} B & 0 & 0 & \cdots & 0 \\ AB & B & 0 & \cdots & 0 \\ A^2B & AB & B & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ A^{n-1}B & A^{n-2}B & A^{n-3}B & \cdots & B \end{bmatrix}_{P_{xx}} \\ &\times \begin{bmatrix} u_k \\ u_{k+1} \\ u_{k+2} \\ \vdots \\ u_{k+n-1} \end{bmatrix} \end{aligned} \quad (12)$$

and output prediction equation

$$\begin{aligned} \begin{bmatrix} y_{k+1} \\ y_{k+2} \\ y_{k+3} \\ \vdots \\ y_{k+n} \\ y_k \end{bmatrix} &= \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^n \\ P \end{bmatrix} x_k \\ + \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ CA^2B & CAB & CB & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ CA^{n-1}B & CA^{n-2}B & CA^{n-3}B & \cdots & CB \end{bmatrix} & (13) \\ &\times \begin{bmatrix} u_k \\ u_{k+1} \\ u_{k+2} \\ \vdots \\ u_{k+n-1} \\ u_{k-1} \end{bmatrix}. \end{aligned}$$

This can be achieved by introducing the prediction matrices,  $P$  and  $H$ . Therefore, the model used is a linear one that can be obtained as shown in

$$\begin{aligned} x_k &= P_{xx}x_k + H_{xx}u_{k-1}, \\ y_k &= Px_k + Hu_{k-1}, \end{aligned} \quad (14)$$

where  $x_k$  is the state model,  $u_k$  is the input model, and  $y_k$  is the measured output model.  $P_{xx}$ ,  $H_{xx}$ ,  $P$ , and  $H$  are matrices and vectors of the right dimension, respectively. The starting point in formulating PFC control law is developing the reference trajectory equation. This can be done by placing the desired closed-loop dynamic into the reference trajectory. Given the actual set point is  $r$ , and the loop set point,  $w$ , is a first-order lag:

$$w_{k+i/k} = r_k - (r_k - y_k)\psi^i, \quad (15)$$

where  $i$  is value of  $n$ ,  $y_k$  is the most recent measured output, and  $\Psi$  ( $0 < \Psi < 1$ ) is scalar time constant and a tuning parameter setting the desired closed-loop poles. Equation (15) is the predictive essence of control strategy. Indeed, the aim is to have the set point trajectory closely follow the reference desired closed-loop behavior. In addition, it must also deal with the set of coincidence points. This can be achieved by using the degree of freedom (DOF) to force the equality of the prediction and the reference trajectory at a number of points. Therefore, solving the control moves such that

$$y_{k+n} = w_{k+n}, \quad (16)$$

where  $n = n_1, n_2, \dots$ . These equalities are called coincidence points. In usual cases, there are no more than two coincidence points. In this paper, we will only focus on only one coincidence point,  $n_1$ . Thus, at a single coincidence point and using (15) and (16), the control law is determined by

$$y_{k+n} = w_{k+n} = r_k - (r_k - y_k)\psi^i. \quad (17)$$

Hence, substituting (14) into (17),

$$y_{k+n} = Px_k + Hu_{k-1} = r_k - (r_k - y_k)\psi^i. \quad (18)$$

Assuming that  $u_{k+i} = u_k$ , thus the control law can be formulated by rewriting (18) and obtain

$$\begin{aligned} u_k &= -H^{-1} [Px_k + (r_k - (r_k - y_k)\psi^i)], \\ u_k &= -K_c x_k + P_c r_k, \end{aligned} \quad (19)$$

where  $K_c = -H^{-1}(P - \Psi^i y_k)$  and  $P_c = -H^{-1}(1 - \Psi^i)$ . Now, the prediction algorithm can easily be recognized from the fixed linear feedback law. Thus, the typical posterior stability and sensitivity analysis can be easily achieved in a straightforward manner.

As stated earlier, there is only one coincidence point. According to [22], the typical procedure with one coincidence point would be as follows.

- (1) Choose the desired time constant,  $\Psi$ .
- (2) Do a search for coincidence horizon,  $n_1 = 1, 2, \dots$  large and find the associated control law for each  $n_1$ .
- (3) Select the  $n_1$ , which gives closed-loop dynamics closest to the chosen  $\Psi$ .
- (4) Simulate the proposed law. Otherwise, reselect  $\Psi$  and go to step 2.

Optimal parameter tuning is an optimization problem, which requires implementation of global optimization strategy such as particle swarm optimization (PSO).

**3.2. Observer.** The model states are not related to physical parameters. In such cases and for the real implementation of PFC, an observer must be designed as the state variable  $x(k_i)$  at time  $k_i$  is not measurable [23]. The function of the observer is to calculate the future state by using the values of the current output of the plant  $y(k_i)$  and the current value of the control signal  $u(k_i)$ . For the system in this study, the observer is designed using the pole-assignment method to calculate the gain  $K_{ob}$ . The following equation is used to estimate the state variable  $x(k_i)$  in each time instant:

$$\hat{x}(k_i + 1) = A\hat{x}(k_i) + Bu(k_i) + K_{ob}(y(k_i) - C\hat{x}(k_i)), \quad (20)$$

where  $u(k_i)$  at time  $k_i$  is as expressed in (19). The closed-loop observer error equation is

$$\tilde{x}(k+1) = (A - K_{ob}C)\tilde{x}(k), \quad (21)$$

where  $\tilde{x}(k) = x(k) - \hat{x}(k)$ . It is important to have all eigenvalues of  $A$  matrix inside the unit circle for the observer error converge to zero. Therefore, the closed-loop observer poles are selected to be inside the unit circle, which gives the observer the fast dynamic response required. The closed-loop PFC system with state estimate has two independent characteristic equations:

$$\det(\lambda I - (A - K_{ob}C)) = 0, \quad (22)$$

$$\det(\lambda I - (A - BK_{PFC})) = 0. \quad (23)$$

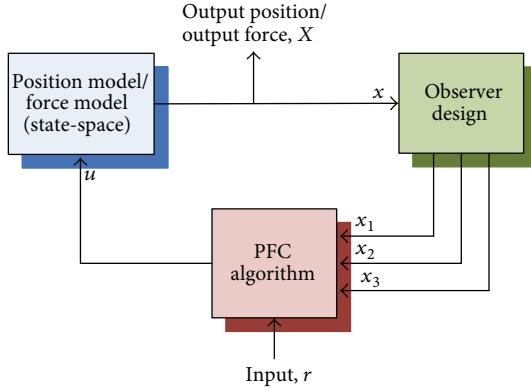


FIGURE 3: Block diagram of PFC-O for plant model.

Adding a sufficiently fast observer will not affect the performance of the PFC controller; (22) represents the eigenvalues of the PFC control loop while (23) represents the eigenvalues of the observer loop. This shows that two sets of eigenvalues are independent of each other. Hence, the design of the observer will not affect the design of the PFC controller or vice versa. The PFC-O design structure is illustrated in Figure 3. The plant models are obtained by using system identification technique (as discussed in Section 2).

The stability test method for this research is done by testing the locations of the closed-loop poles. The stability performance of the closed-loop feedback system is determined primarily by the location of the poles (eigenvalues) of the matrix  $(A - BK_{\text{PFC}})$ . Since  $A$  and  $B * K_{\text{PFC}}$  are both 3 by 3 matrices, there will be 3 poles for the closed-loop system. By using the MATLAB function  $\text{eig}(A - BK_{\text{PFC}})$ , the desired poles for position model and force model are stable because all the poles of the closed-loop system lie within the unit circle of the  $z$ -plane.

#### 4. Stiffness Characteristic

The relationship between deflection and force is known as the stiffness or can be assumed as a spring rate. The greater the stiffness, the less the deflection for a given force,  $F$ , and high stiffness springs are hard and low stiffness springs are soft. With an ideal spring system with spring constant, the stiffness characteristic is achieved using compliance control as in

$$F = k_s e(k), \quad (24)$$

where  $F$ ,  $k_s$ , and  $e(k)$  represent the force reference, coefficient of stiffness, and position error from the optical sensor, respectively.

Within its elastic (flexibility) limit, the deflection,  $\delta$ , of a spring is linearly proportional to the force applied to that spring. The coil spring phenomena are illustrated in Figure 4. Stiffness coefficient of the spring can be calculated as

$$F = \delta k_s. \quad (25)$$

When weight,  $W$ , is the force exerted on a body by gravity,

$$F = W = mg, \quad (26)$$

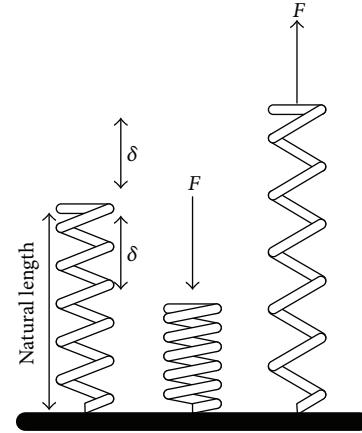


FIGURE 4: Coil spring illustration.

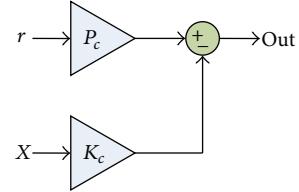


FIGURE 5: PFC controller stage.

the deflection is

$$\delta = \frac{mg}{k_s}, \quad (27)$$

where free gravitational acceleration,  $g$ , is  $9.8 \text{ m/s}^2$ .

#### 5. Embedded Controller Development

Before applying to embed algorithm in PSoC programming, all equations need specific data, a simple equation, and rewriting for easier coding. Consider a PFC controller with the following fundamental matrices:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (28)$$

$$B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},$$

$$C = [c_{11} \ c_{12} \ c_{13}],$$

where pole is  $1 \times 3$  matrix of constants.

Figure 5 shows the controller stage of a PFC controller from (19) which can be represented by

$$\text{Out} = P_c r - K_c X, \quad (29)$$

where  $r$  is reference input,  $X$  is  $3 \times 1$  matrix representing the system states, Out is control signal,  $P_c$  is constant gain and coincidence horizon, and  $n_1$  is 2 while the matrix  $K_c$  is given by

$$K_c = [k_{c11} \ k_{c12} \ k_{c13}] = [(CAB + CB)(CA^2 - C\Psi^2)]^{-1}, \quad (30)$$

where  $\Psi$  is a given constant. Expanding (29) yields

$$\begin{aligned} \text{Out} &= P_c r - [k_{c11} \ k_{c12} \ k_{c13}] \begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \end{bmatrix} \\ &= P_c r - (k_{c11}x_{11} + k_{c12}x_{21} + k_{c13}x_{31}). \end{aligned} \quad (31)$$

Figure 6 shows the observer stage of a PFC controller from (20) where the output signal of the rightmost summing junction,  $M_p$ , is represented by

$$\begin{aligned} M_p(k) &= \\ &= \begin{bmatrix} p_{11}(k) \\ p_{21}(k) \\ p_{31}(k) \end{bmatrix} = K(Y - C \cdot M_p(k-1)) + AM_p(k-1), \end{aligned} \quad (32)$$

where matrix  $K$  as  $K_{ob}$  and derived using the MATLAB function  $K = \text{place}(A', C', \text{pole})$  yielding

$$K = [k_{11} \ k_{12} \ k_{13}] \quad (33)$$

and  $Y$  and  $X$  are the plant output signal and estimated state output, respectively. The value of  $X$  is given by

$$X(k) = \begin{bmatrix} x_{11}(k) \\ x_{21}(k) \\ x_{31}(k) \end{bmatrix} = M_p(k-1). \quad (34)$$

The value of  $M_p(k)$  is obtained by expanding (32) yielding

$$\begin{aligned} M_p(k) &= \begin{bmatrix} p_{11}(k) \\ p_{21}(k) \\ p_{31}(k) \end{bmatrix} = [k_{11} \ k_{12} \ k_{13}] \\ &\quad \times \begin{bmatrix} Y(k) - c_{11}p_{11}(k-1) \\ Y(k) - c_{12}p_{21}(k-1) \\ Y(k) - c_{13}p_{31}(k-1) \end{bmatrix} \\ &\quad + \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} p_{11}(k) \\ p_{21}(k) \\ p_{31}(k) \end{bmatrix} \\ &= (k_{11} + k_{12} + k_{13})Y(k) \\ &\quad - (k_{11}c_{11}p_{11}(k-1) + k_{12}c_{12}p_{21}(k-1) \\ &\quad + k_{13}c_{13}p_{31}(k-1)) \\ &\quad + \begin{bmatrix} a_{11}p_{11}(k-1) + a_{12}p_{21}(k-1) + a_{13}p_{31}(k-1) \\ p_{11}(k-1) \\ p_{12}(k-1) \end{bmatrix}. \end{aligned} \quad (35)$$

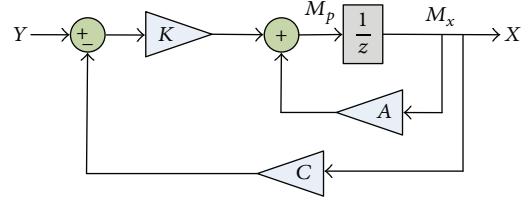


FIGURE 6: Observer stage.

To rewrite the equations for easier coding, the equation for PFC controller and observer stage is now reduced to (31), (34), and (35), which are still in their matrix form. To ease coding, the equations are rewritten as single-line expressions, thus yielding the following equations. The observer stage is written as

$$\text{Out} = P_c r(k) - (k_{c11}x_{11}(k) + k_{c12}x_{21}(k) + k_{c13}x_{31}(k)), \quad (36)$$

where  $\text{Out}(k)$  and  $r(k)$  are the controller output and controller reference signal, respectively. The value of  $x_{n1}(k)$  is provided by the observer stage written as

$$\begin{aligned} x_{11}(k) &= p_{11}(k-1), \\ x_{21}(k) &= p_{21}(k-1), \\ x_{31}(k) &= p_{31}(k-1) \end{aligned} \quad (37)$$

and the value of  $p_{n1}(k)$  is provided by

$$\begin{aligned} p_{11}(k) &= H + a_{11}p_{11}(k-1) + a_{12}p_{21}(k-1) \\ &\quad + a_{13}p_{31}(k-1), \\ p_{21}(k) &= H + p_{11}(k-1), \\ p_{31}(k) &= H + p_{12}(k-1). \end{aligned} \quad (38)$$

The value of  $H$  is given by

$$H = I - J, \quad (39)$$

where  $I = (k_{11} + k_{12} + k_{13})Y(k)$ ,  $J = (k_{11}c_{11}p_{11}(k-1) + k_{12}c_{12}p_{21}(k-1) + k_{13}c_{13}p_{31}(k-1))$ , and  $Y(k)$  is the general feedback signal from the plant.

In this research, the control methodology contains force inner loop and position outer loop to obtain the stiffness characteristic objective. By controlling the difference of both sides of the pneumatic actuator, the inner loop enforces the natural stiffness characteristic of the pneumatic actuator. The working function of stiffness characteristic is shown in Figure 7 where the system tried to achieve the target position by giving the appropriate value of force. The error in force value reading will be eliminated using PFC-O control that adjusts the duty cycle of PWM signal for actuator stroke force. Meanwhile integral gain,  $k_I$ , as a compensator is added to solve the problem of stretch-back not functioning with lower stiffness parameter.

The feedback of output force (signal inner loop) to observer is

$$Y(k) = y_F(k) = 104.9074 - 0.8437n_p, \quad (40)$$

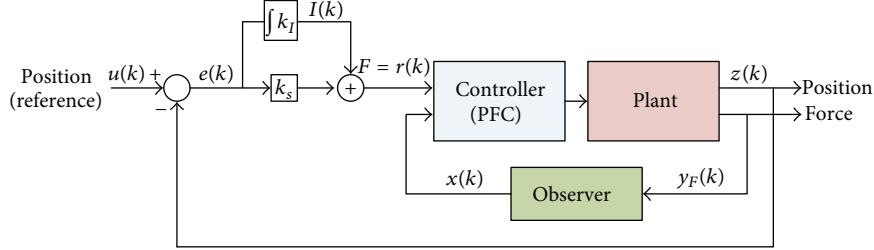


FIGURE 7: Block diagram for control system with stiffness characteristic.

where  $n_p$  is PSoC 11-bit delta-sigma ADC raw conversion result by calculation and the PFC force controller reference signal is

$$r(k) = k_s e(k) + I(k), \quad (41)$$

where  $k_s$  is a coefficient of stiffness,  $e(k)$  is position error,  $I(k)$  is compensator output, and the equation for the compensator is given by

$$I(k) = k_I \tau e(k) + I(k - 1), \quad (42)$$

where  $k_I$  is integral gain and  $\tau$  is discrete integrator sampling time.

## 6. Experimental Setup

The experimental setup for these researches consists of simulation and real-time analysis. The simulation data is acquired using MATLAB Simulink, where (6) and (7) are directly applied and tested with the close-loop controller design using MATLAB Simulink. Meanwhile, the real-time experimental data are acquired using national instrument (NI) devices and programmable system on chip (PSoC) microcontroller. The experimental setup for the real-time using national instrument (NI) devices is the same as that described in Section 2 but the input-output connection is directly tested with the close-loop controller design using MATLAB Simulink [4]. Therefore, the technical merit of this work consists of modified new wiring and communication figure with an online system in a real-time environment. The data acquisition (DAQ) card PCI/PXI-6221 (68-Pin) board connected is used for interfacing the plant with a computer. From the communication diagram in Figure 8, the signal emitted from the circuit board consists of an analog signal output for valves, an analog signal input for pressure, and a signal counter input for the encoder. Experiment for position control or compliance control is in normal movement of the actuator as in Figure 9(a). However, experiments on force control will be in static position movement as in Figure 9(b).

Next, experimental setup to implement the real-time environment using embedded system is a continuation of previous work using the PSoC control board [16–19]. There are 5 connectors attached on the board connected to valves, pressure sensor, 2 for power supply and  $I^2C$  communication, and 1 for reburning programs. From this board, other parts are controlled by reading pressure sensor

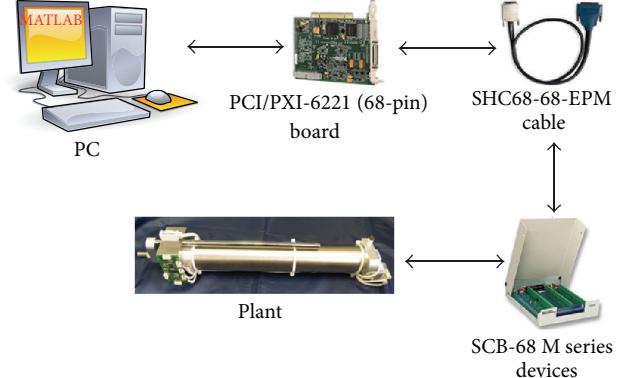


FIGURE 8: National instrument (NI) devices connection.

data and detecting actuator strokes from the optical sensor. CY8C27443 chip and C programming were used for easier implementation and fast execution. PSoC represents a whole new concept in microcontroller development. By having an easy-to-use development tool, PSoC enables user to select desired peripherals including analogue function (amplifiers, ADCs, DACs, filters, and comparators) and digital functions (timers, counters, PWMs, SPI, and UARTs) making PSoC different from other microcontrollers. Additionally, a fast CPU of 24 MHz, 16 kb of Flash program memory, SRAM data memory with 256 bytes, and configurable input/output (I/O) is included in a range of pin outs. The distributed architecture applying several PSoCs enables multitasking and parallel processing of the microcontroller. This will increase efficiency of the data processing and give shorter access time. In this distributed approach, the PSoC has its own private memory and information is exchanged by passing data between the microcontrollers.

By applying this methodology, the overall system will be enhanced with the new controller coding such as PFC-O algorithm, simpler connections, and reduced numbers of wires between PC and the actuator. Furthermore, the communication protocol between PC and  $I^2C$  communication board applies USB to UART converter protocol. For better response, the actuator will give different output characteristics (position and stiffness parameter) from the input given and monitor using MATLAB M-File (position and force) as an online communication. Figure 10 shows the PSoC control board and experiment setup to be applied to the embedded system. In addition, the payload as a mass

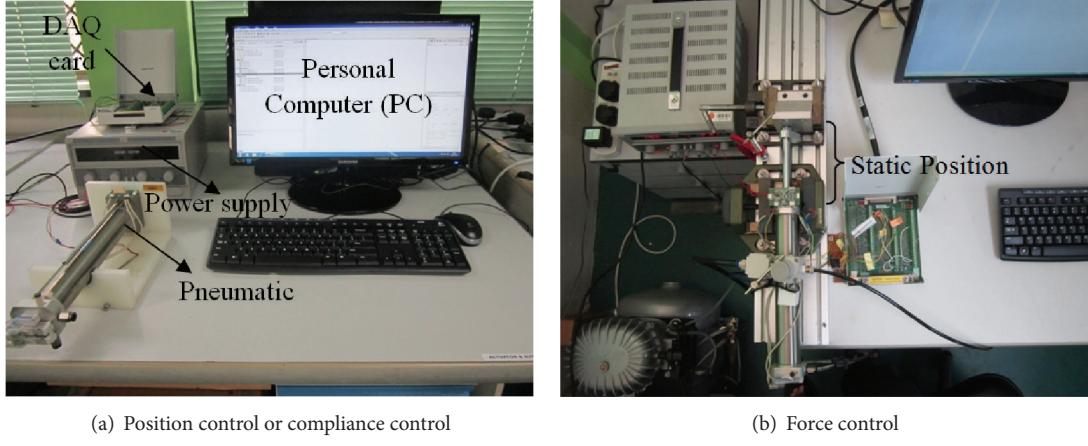


FIGURE 9: Real experiment setup using NI devices.

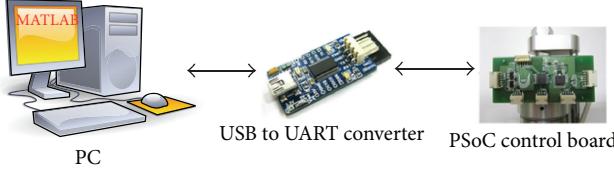


FIGURE 10: Embedded system connection.

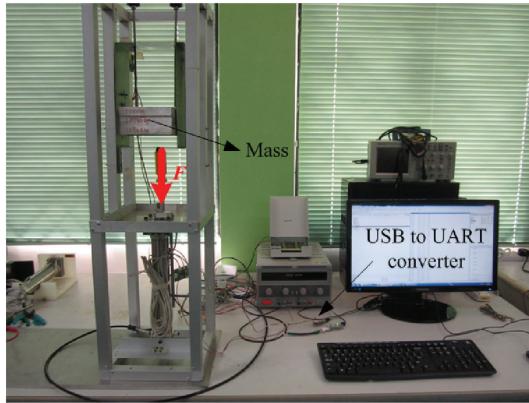


FIGURE 11: Pneumatic actuator with mass.

is attached to the pneumatic plant in vertical direction to test the system capability with fixed position and different stiffness parameters. The purpose of this experiment was to compare the theoretical data analysis for mass-spring mechanical system method and real-time experiment using NI devices. Figure 11 shows the real experiment setup for pneumatic actuator with mass.

## 7. Result and Discussion

In this research, a new model and a novel embedded process control strategy to design the controller for real-time pneumatic system have been proposed. This section shows the results of model validation and application to embedded

TABLE 1: Comparison of simulated and experimental performance for position control.

Specifications	Simulation	Experiment
Settling time ( $T_S$ )	0.79 s	1.12 s
Rise time ( $T_R$ )	0.57 s	0.80 s
Percent steady state error ( $\%e_{ss}$ )	0.01%	0.03%

%: percent, s: second.

system are analyzed and discussed to further evaluate the controller.

**7.1. Model Validation Analysis.** Simulation and real-time experiment using national instrument (NI) device analysis were carried out to validate the controller performance. This analysis of the actual situation pneumatic actuator where force maximum and without stiffness characteristics. The results of position control and force control are analyzed before applying the PFC-O controller algorithm to embedded system.

**7.1.1. Position Control.** Comparison between the simulation and the experiment result for position step and multistep responses including control signals within 18 s is shown in Figures ?? and 13. For this control position, the PFC control law and the prediction model of the system are developed using the following parameters: desired time constant,  $\Psi$ , is 0.95 and coincidence horizon,  $n_1$ , is 2 while the closed-loop observer poles are selected to be inside the unit circle, that is, 0.05, 0.04, and 0.001. The performances index of the simulation and experiment for position step responses is summarized in Table 1.

**7.1.2. Force Control.** Figures 14 and 15 show the force responses of the system for step and multistep responses including control signals within 18 s. Force readings were obtained from the mathematical derivation of the pressure sensor data. The high negative force reading during the initial

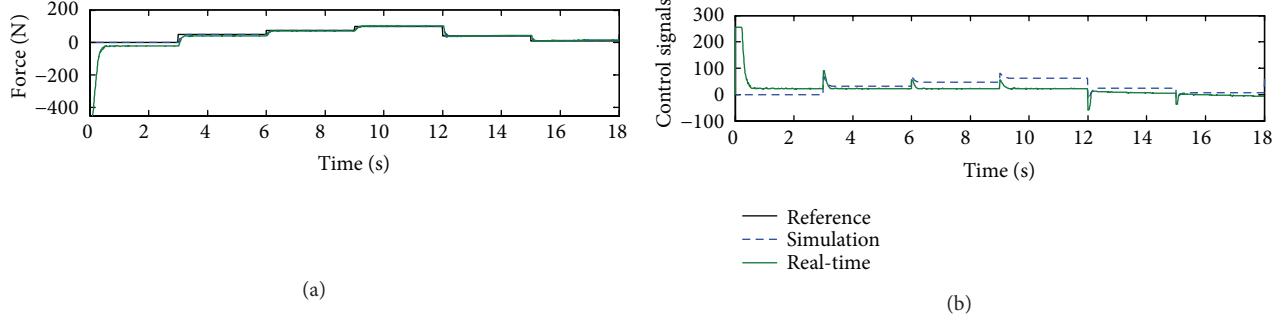


FIGURE 12: Force multistep responses.

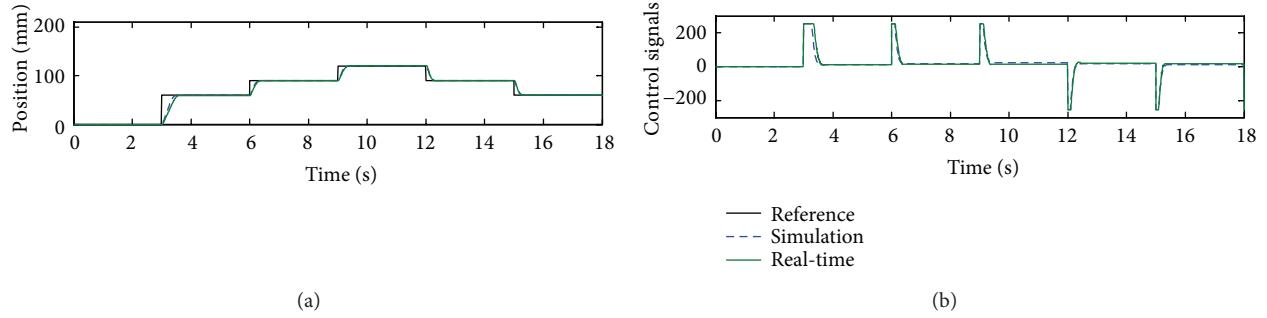


FIGURE 13: Position multistep responses.

TABLE 2: Comparison of simulated and experimental performance for force control.

Specifications	Simulation	Experiment
Settling time ( $T_S$ )	0.2788 s	0.935 s
Rise time ( $T_R$ )	0.1601 s	0.3616 s
Percent steady state error (% $e_{ss}$ )	0.01%	0.08%

%: percent, s: second.

stage of the experiment is due to the double acting nature of the cylinder, where one chamber is filled with compressed air driving the cylinder in the negative direction when the other chamber is emptied. The parameters of PFC desired time constant,  $\Psi$ , are 0.92, coincidence horizon,  $n_1$ , is 2, and closed-loop observer poles are 0.05, 0.23, and 0.01. The performance index of the simulation and experiment for force step responses is summarized in Table 2.

The results for model validation analysis and the position control and force control analysis provide good performance in terms of no overshoot, faster settling time ( $T_S$ ), rise time ( $T_R$ ), and smaller percent steady-state error (% $e_{ss}$ ). PFC-O control signal for both models is stable because the signal amplitude is between 255 and -255. Not to mention the zero also, to force the valve to fully open in their periods an 8-bit PWM generator is used, so the maximum value for the signal is 255, which forces the first valve to have a full open period, whereas when the signal is -255, it forces the second valve to have a full open period instead. The simulated result is better because the transfer function used for simulation is linear, which does not contain the nonlinearities found on actual

systems. However, due to the involvement of nonlinearities, the transient response experiment shows slower response but can be considered fast enough for a pneumatic system.

**7.2. Embedded System Analysis.** The control analysis will be done in the simulation and real-time experiment using National Instrument (NI) device environment. Next, all coding in Section 5 is converted to C programming and burned to PSoC. The aim controller performances proceed to apply an embedded system and realized compliance control for stiffness characteristic. Different stiffness coefficients will be tested and the position value will be fixed based on experimental work. Two methods to get the position value are examined, first by controlling the pneumatic actuator position and sending the value for force control loop and second by getting the values from mass attached and sending them to the force control loop.

**7.2.1. Compliance Control without Mass.** The basic compliance control is presented in (24). In this control, the target position was set to origin position at 100 mm within 18 s. Three different  $k_s$  inputs of 2 N/mm, 1 N/mm, and 0.5 N/mm are plotted in Figure 16. From the results, feedback force from the actuator gives positive force. In addition, when the value of  $k_s$  is small, position during the experiment is slightly different and rise time is slow to achieve the target compared to the simulation. This is because the actuator becomes soft at low force. This can be seen more clearly in Figure 17 for position step response. PFC parameters were not much different with force control where desired time constant,  $\Psi$ , is

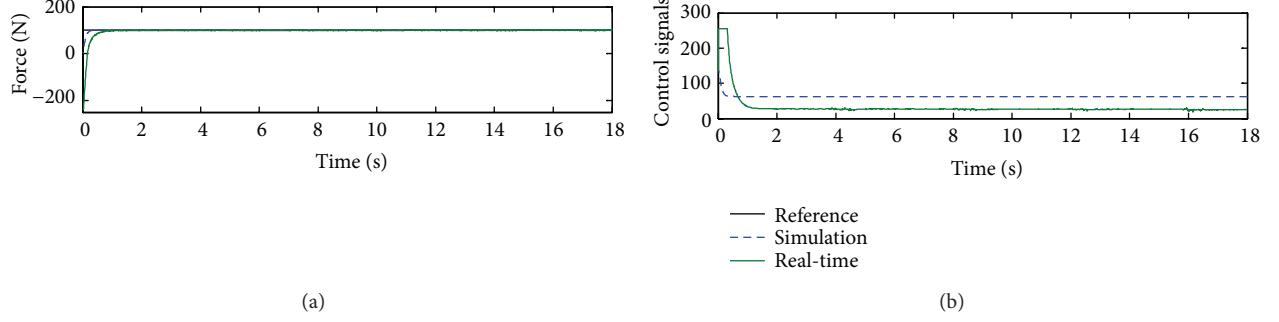


FIGURE 14: Force step responses.

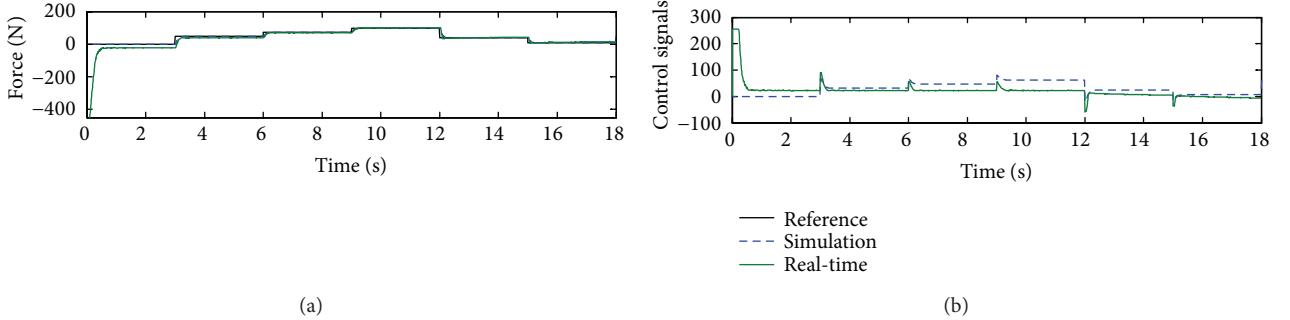


FIGURE 15: Force multistep responses.

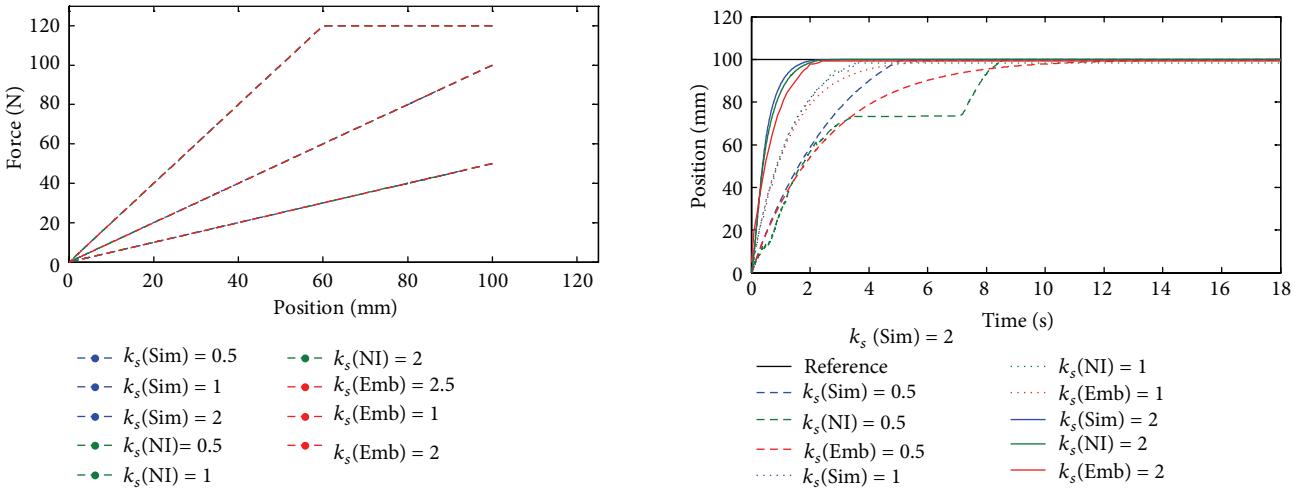


FIGURE 16: Stiffness characteristic responses.

0.92, coincidence horizon,  $n_1$ , is 2, observer poles are 0.05, 0.2, and 0.01, and integral gain,  $k_I$ , is 0.1. These parameters are also the same as with simulations, real-time experiments using national Instrument (NI) devices, and real-time embedded system.

**7.2.2. Compliance Control with Mass.** The second method for compliance control was referred to in (27). The mass 3 kg is attached and raised at certain times to the pneumatic plant in vertical direction to test the system capability. In this control

with fixed 100 mm position and different stiffness parameters such as  $k_s$  input of 2 N/mm, 1 N/mm, and 0.5 N/mm, results of the analysis found that the deflection value in experiment using NI devices is almost the same and better than embedded systems. This is because embedded systems follow the behavior that has been set through MATLAB and time taken to calculate the algorithm. Furthermore, when a mass applied on the value of  $k_s$  is small, the time decreases faster but rise time is slow to achieve the target because the actuator becomes soft. A comparison between the theoretical

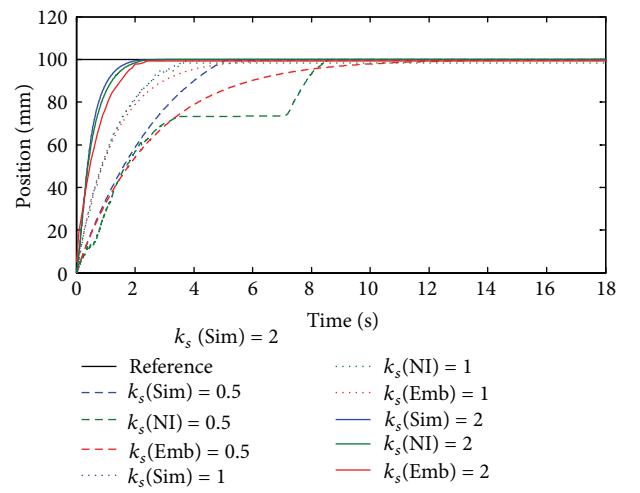


FIGURE 17: Position step responses for difference stiffness.

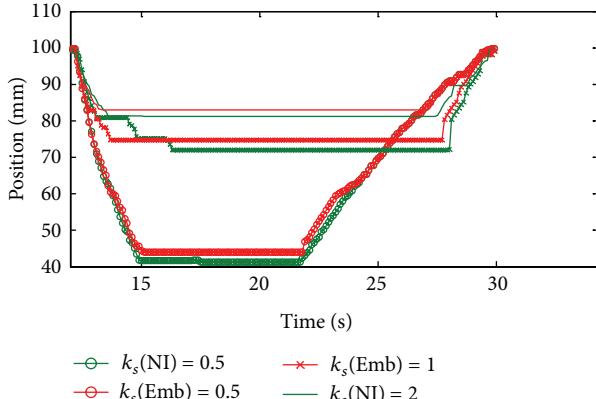


FIGURE 18: Deflection analysis responses.

TABLE 3: Comparison of deflection results.

Stiffness parameters, $k_s$	Theory	Real-time: NI devices	Real-time: embedded system
0.5 N/mm	58.86 mm	58.23 mm	57.50 mm
1 N/mm	29.43 mm	29.01 mm	28.17 mm
2 N/mm	14.72 mm	16.80 mm	15.94 mm

N: newton, mm: millimeter.

calculations with both real-time results is shown in Figure 18 and Table 3 where the controller parameters are set up as control with mass.

In the analysis of compliance control for an embedded system, stiffness characteristics were successfully applied to give the spring effect. The experimental  $k_s$  data are identical to the input data giving minimum error. This is due to hardware friction inside the actuator. Despite the observed differences, the model is considered acceptable because of the similarities between simulation, theoretical calculation, and both real-time experiments except for rise time and target achieved.

## 8. Conclusion

Considering the nonlinear characteristics of the pneumatic system for this research scope, the results from the simulation and the both real-time experiments matched closely, and this is considered as a validation of the obtained mathematical model. Controller design for pneumatic actuator is done using PFC-O. Stiffness characteristic is realized using the compliance control. To compare the performance of the PFC-O analysis, several parameters have been identified. The results obtained from the simulation and experiment show that the developed real-time model could be used for various research bases, such as improvement of the controller performance and implementation on embedded systems. Furthermore, this pneumatic system can work well as a robust system and that makes it a suitable controller with good control performance. This research will provide greater opportunities for future work such as development of graphic

user interface (GUI) to enhance online communication with more than one actuator and to apply the pneumatic actuator to related applications such as rehabilitation device.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

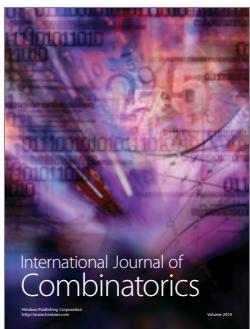
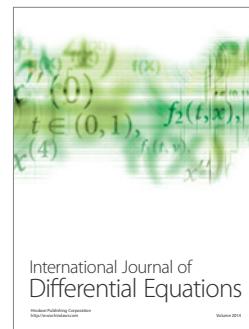
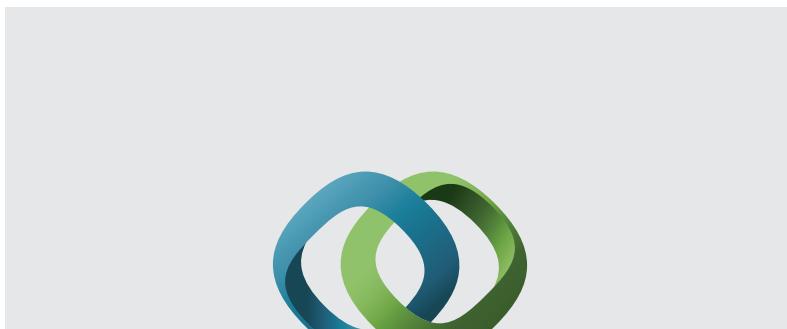
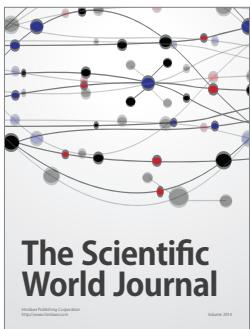
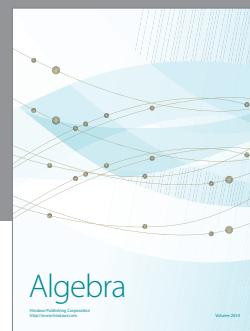
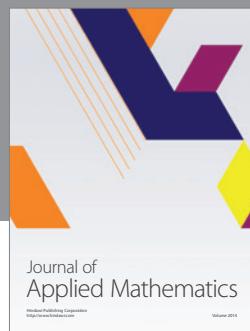
## Acknowledgments

The authors would like to thank the Universiti Teknologi Malaysia (UTM), Ministry of Higher Education (MOHE), Malaysia, under Exploratory Research Grant Scheme (ERGS) no. R.J130000.7823.4L070, Universiti Teknikal Malaysia Melaka (UTeM), and the Okayama University for their support.

## References

- [1] A. C. Valdiero, C. S. Ritter, C. F. Rios, and M. Rafikov, "Non-linear mathematical modeling in pneumatic servo position applications," *Mathematical Problems in Engineering*, vol. 2011, Article ID 472903, 16 pages, 2011.
- [2] L. A. Zadeh, "From circuit theory to system theory," *Proceedings of the IRE*, vol. 50, no. 5, pp. 856–865, 1962.
- [3] J. A. Corrales, G. J. G. Gómez, F. Torres, and V. Perdereau, "Cooperative tasks between humans and robots in industrial environments," *International Journal of Advanced Robotic Systems*, vol. 9, article 92, 2012.
- [4] A. A. M. Faudzi, K. Osman, M. F. Rahmat, K. Suzumori, N. D. Mustafa, and M. A. Azman, "Real-time position control of intelligent pneumatic actuator (IPA) system using optical encoder and pressure sensor," *Sensor Review*, vol. 33, no. 4, Article ID 17094939, pp. 341–351, 2013.
- [5] A. Saleem, S. Abdrabbo, and T. Tutunji, "On-line identification and control of pneumatic servo drives via a mixed-reality environment," *International Journal of Advanced Manufacturing Technology*, vol. 40, no. 5-6, pp. 518–530, 2009.
- [6] P. Matoušek, "Adaptive control of pneumatic servomechanism," *ANNALS of Faculty Engineering Hunedoara*, vol. 2, pp. 73–78, 2011.
- [7] M. F. Rahmat, N. H. Sunar, S. N. S. Salim, M. S. Z. Abidin, A. A. M. Fauzi, and Z. H. Ismail, "Review on modeling and controller design in pneumatic actuator control system," *International Journal on Smart Sensing and Intelligent Systems*, vol. 4, no. 4, pp. 630–661, 2011.
- [8] M. F. Rahmat, S. N. S. Salim, N. H. Sunar, A. A. M. Faudzi, Z. H. Ismail, and K. Huda, "Identification and non-linear control strategy for industrial pneumatic actuator," *International Journal of the Physical Sciences*, vol. 7, no. 17, pp. 2565–2579, 2012.
- [9] B. Huyck, H. J. Ferreau, M. Diehl et al., "Towards online model predictive control on a programmable logic controller: practical considerations," *Mathematical Problems in Engineering*, vol. 2012, Article ID 912603, 20 pages, 2012.
- [10] J. A. Rossiter, *Model-Based Predictive Control: A Practical Approach*, CRC Press, 2003.
- [11] A. I. Maalouf, "Improving the robustness of a parallel robot using Predictive Functional Control (PFC) tools," in *Proceedings of the 45th IEEE Conference on Decision and Control (CDC '06)*, pp. 6468–6473, December 2006.

- [12] M. Aliff, S. Dohtaa, T. Akagi, and H. Li, "Development of a simple-structured pneumatic robot arm and its control using low-cost embedded controller," in *Proceedings of the International Symposium on Robotics and Intelligent Sensors (IRIS '12)*, vol. 41 of *Procedia Engineering*, pp. 134–142, 2012.
- [13] R. Isermann, "Mechatronic systems—innovative products with embedded control," *Control Engineering Practice*, vol. 16, no. 1, pp. 14–29, 2008.
- [14] J. Wang, H. Hu, J. Wang, Z. Li, and Z. Gong, "Development of an embedded control system for magnetorheological fluid damper under impact load," in *Proceedings of the 9th International Conference on Electronic Measurement and Instruments (ICEMI '09)*, pp. 717–722, August 2009.
- [15] K.-P. Kang, M. Moallem, and R. V. Patel, "Embedded controller design for an active damper system," in *Proceedings of IEEE Conference on Control Applications (CCA '03)*, pp. 526–531, June 2003.
- [16] A. A. M. Faudzi, K. Suzumori, and S. Wakimoto, "Development of an intelligent pneumatic cylinder for distributed physical human-machine interaction," *Advanced Robotics*, vol. 23, no. 1–2, pp. 203–225, 2009.
- [17] A. A. M. Faudzi, K. Suzumori, and S. Wakimoto, "Development of an intelligent chair tool system applying new intelligent pneumatic actuators," *Advanced Robotics*, vol. 24, no. 10, pp. 1503–1528, 2010.
- [18] A. A. M. Faudzi and K. Suzumori, "Programmable system on chip distributed communication and control approach for human adaptive mechanical system," *Journal of Computer Science*, vol. 6, no. 8, pp. 852–861, 2010.
- [19] A. A. M. Faudzi, K. Suzumori, and S. Wakimoto, "Development of pneumatic actuated seating system to aid chair design," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM '10)*, pp. 1035–1040, July 2010.
- [20] L. Ljung, *Identification Toolbox for Use with MATLAB*, The MathWorks, 2002, <http://www.mathworks.com/products/sysid>.
- [21] K. Osman, A. A. M. Faudzi, M. F. Rahmat, N. D. Mustafa, M. A. Azman, and K. Suzumori, "System identification model for an Intelligent Pneumatic Actuator (IPA) system," in *Proceedings of the 25th IEEE/RSJ International Conference on Robotics and Intelligent Systems (IROS '12)*, pp. 628–633, October 2012.
- [22] J. A. Rossiter and J. Richalet, "Handling constraints with predictive functional control of unstable processes," in *Proceedings of the American Control Conference (ACC '02)*, vol. 6, pp. 4746–4751, May 2002.
- [23] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB*, Springer, 2009.



Submit your manuscripts at  
<http://www.hindawi.com>

