

PI Adaptive Neuro-Fuzzy and Receding Horizon Position Control for Intelligent Pneumatic Actuator

Omer Faris Hikmat^{a*}, Ahmad 'Athif Mohd Faudzi^{a,b}, Mohamed Omer Elnimair^{a,d}, Khairuddin Osman^{a,c}

^aDepartment of Control and Mechatronics Engineering, Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia

^bCentre for Artificial Intelligence and Robotics (CAIRO), Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia

^cDepartment of Industrial Electronics, Faculty of Electrical and Electronics, Universiti Teknikal Malaysia, Melaka, Malaysia

^dAlhsour Mining, Khartoum, Sudan

*Corresponding author: omer_faris2002@yahoo.com

Article history

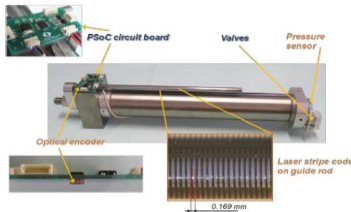
Received :23 October 2013

Received in revised form :

14 December 2013

Accepted :10 January 2014

Graphical abstract



Abstract

Pneumatic systems are widely used in automation industries and in the field of automatic control. Intelligent Pneumatic Actuators (IPA) is a new generation of actuators designed and developed for research and development (R&D) purposes. This work proposes two control approaches, Proportional Integral Adaptive Neuro-Fuzzy (PI-ANFIS) controller and Receding Horizon Controller (RHC), for IPA position control. The design steps of the controllers are presented. MATLAB/SIMULINK is used as a tool to implement the controllers. The design is based on a position identification model of the IPA. The simulation results are analyzed and compared with previous work on the IPA to illustrate the performance of the proposed controllers. The comparison shows a significant improvement in IPA position control after using the new controllers.

Keywords: Intelligent pneumatic actuator; position control; neuro-fuzzy; receding horizon control

Abstrak

Sistem pneumatik digunakan secara meluas di dalam industri automasi dan dalam bidang kawalan automatik. Penggerak Pintar Pneumatik (IPA) ialah generasi terkini penggerak yang direka dan dibangunkan bagi tujuan penyelidikan dan pembangunan. Kerja ini mencadangkan dua pendekatan kawalan, iaitu Penyesuaian Berkadar Integral Neuro-Fuzzy (PI-ANFIS) dan Kawalan Surut Ufuk (RHC), untuk kawalan kedudukan IPA. Langkah-langkah bagi merekabentuk pengawal ditampilkan. Matlab/Simulink digunakan sebagai alat untuk mengadaptasi pengawal terbabit. Rekabentuk ini adalah berdasarkan model pengenalan kedudukan IPA. Keputusan simulasi di analisis dan dibandingkan dengan kerja-kerja terdahulu terhadap IPA untuk menggambarkan prestasi pengawal yang dicadangkan. Perbandingan terbabit menunjukkan peningkatan yang ketara didalam kawalan kedudukan IPA selepas menggunakan pengawal yang baru..

Kata kunci: Penggerak pintar automatik; kawal kedudukan; neoru-fuzzy; kawalan surut ufuk

© 2014 Penerbit UTM Press. All rights reserved.

1.0 INTRODUCTION

Pneumatic systems are widely used in automation industries and in the field of automatic controllers. Pneumatic actuators are safe and reliable. They have relatively small size compared to hydraulic actuators. Moreover, they have fast response, and at high temperatures or in nuclear environments, they have the advantages over hydraulic actuators because gases are not subjected to temperature limitations.¹

The difficulties of controlling pneumatic actuators are mostly because of the nonlinearities existed. The high frictional

forces, which the pneumatic actuator is subjected to, the compressibility of air, the valve dead zone, etc are all sources of these nonlinearities. As a result, these nonlinearities had made achieving accurate position control of the pneumatic actuators become such a difficult task.

These merits and challenges have motivated many researchers among the years to propose and apply different control approaches to achieve higher accuracy and better dynamic performance. Their main interest is to control the position, but due to different industry and automation

requirements, the interests of researchers extended to control the force, stiffness and viscosity of the pneumatic actuators.²

Based on the historical development, pneumatic systems were created since the 16th century.³ There are mainly two types of pneumatic actuators, the piston-cylinder type and the rotary type. Many developments has been done on pneumatic actuators to suit different automation and industry requirements according to the desired accuracy and performance and to the amount of force that is needed for each particular application. In the 20th century, more complex and intelligent pneumatic systems were developed. The intelligent pneumatic actuator (IPA) system, on which the two proposed controllers are applied, is developed by A. A. M. Faudzi *et al.*⁴⁻⁶ in which they developed intelligent actuators for a Pneumatic Actuator Seating System (PASS).

The IPA plant structure is briefly explained in section 2. In section 3, two control approaches to control the IPA position namely PI Adaptive Neuro-Fuzzy controller and Receding horizon predictive controller (RHC) are presented. The results of these controllers are presented, analyzed and compared. The last section addresses the conclusion and the future work.

2.0 THE IPA PLANT

The actuator is equipped with five main components; laser strip on rod, optical encoder, pressure sensor, valves and PSoC microcontroller (Figure 1–shows all these components). There are three elements of the optical encoder; an LED light source, a photo detector IC and optical lenses. The lenses role is to focus an LED light onto the code strips. This light will be reflected and received by the photo detector IC. The encoder, which is used as position sensor, is mounted at bottom side of which is used as position sensor, is mounted at bottom side of the PSoC board (see Figure 1).

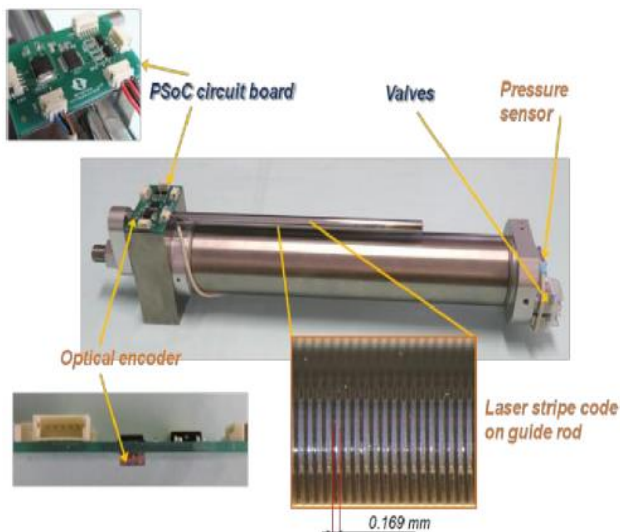


Figure 1 Intelligent pneumatic actuator and its components²

There are two chambers available in IPA. By manipulating the pressure in chamber 1, right and left movements of the actuator can be controlled. The method of controlling the actuator movements is by supplying constant air pressure to chamber 2 at 0.6 MPa (P_1) while regulating air inside chamber 1 from (0-0.6) Mpa (P_2). Right and left movements depend on the algorithm to drive the valve using PsoC PWM duty cycle in chamber 1. Pressure sensor is connected to PsoC for pressure

data reading. The chamber pressure is the input for the control action of the cylinder. The pressure sensor reads the pressure in chamber 1 and can be used to calculate force, F_d using equation below:

$$F_d = P_2A_2 - P_1A_1$$

where P_1 and P_2 are pressure data, A_1 and A_2 are cross-sectional areas in chamber 1 and 2. Assume that P_1 (constant 0.6Mpa), A_1 , A_2 are known values. By reading the pressure in chamber 2 (P_2), force data, F_d can be known.

The actuator applies 2 valves, KOGANEI (EB10ES1-PS-6W) (two ports two positions) to drive the actuator. The valves are attached at the end of the actuator. By controlling only air inlet in chamber 1, the control mechanism will be easier compared to control both chambers. Valve 1 will control the air inlet while valve 2 will control the air exhaust. The method of controlling the valves is by using PWM duty cycle driven by PsoC (Figure 2–shows the IPA schematic operations, valve connection and airflow to the cylinder). Below are the possible movements of the actuator, which depend on the valves operation.

- 1) Valve 1-OFF, Valve 2-OFF–Cylinder stops
- 2) Valve 1-OFF, Valve 2-ON–actuator moves left direction
- 3) Valve 1-ON, Valve 2-OFF–actuator moves right direction
- 4) Valve 1-ON, Valve 2-ON–no operation

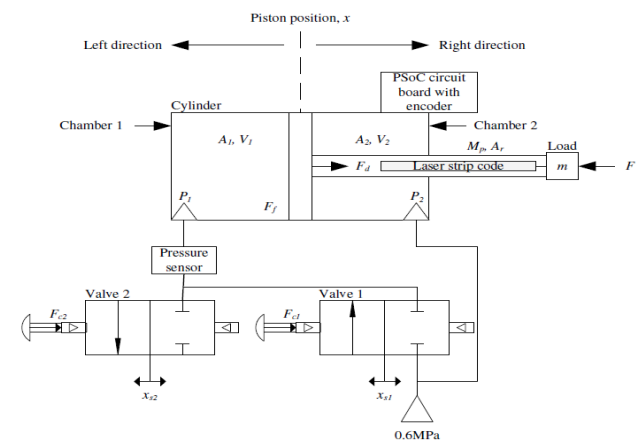


Figure 2 IPA schematic operations⁷

The PSoC board attached to the actuator plays an important role in control and communication of the actuator. There are two inputs signal; encoder and pressure sensor for PSoC and one output signal to control the valve.

A position model of the IPA used in this study has been previously obtained using system identification technique.⁸ The model was approximated using MATLAB System Identification Toolbox from open-loop input-output experimental data. For experimental setup, the hardware and Personal Computer (PC) is connected using Data Acquisition (DAQ) card through MATLAB software.

From several methods used in generating the signals such as PRBS (Pseudo-Random Binary Sequences), sinusoidal, step etc., the step signal was selected and was specially designed for the on/off valve of the cylinder system. This signal has been injected to valve and the output of the system was recorded. Several sets of input and output data sampled at 0.1s were collected for model estimation and validation. Each data contains 1000 samples. Details of the SI technique used are described in the references.^{2,8}

The system identification resulted in an Auto-Regressive Moving Average with Exogenous Input (ARMAX) model in the

form of discrete-time open-loop transfer function. The model obtained is a linear third order system as in Equation (1),

$$\frac{B_0(z^{-1})}{A_0(z^{-1})} = \frac{0.3033z^{-1} + 0.04125z^{-2} + 0.2108z^{-3}}{1 - 1.147z^{-1} + 0.9434z^{-2} - 0.5826z^{-3}} \quad (1)$$

This discrete model is then converted to continuous transfer function for ANFIS controller design and to discrete state space model for RHC controller design.

3.0 CONTROLLERS DESIGN

This work proposes two control approaches, Proportional Integral Adaptive Neuro-Fuzzy (PI-ANFIS) controller and Receding Horizon Controller (RHC), for IPA position control. The design steps of the controllers are presented in the following subsections.

3.1 Adaptive Pneuuro-Fuzzy (Anfis)

Classical control theory is based on the mathematical models that describe the physical plant under consideration. The essence of fuzzy control is to build a model of human expert who is capable of controlling the plant without thinking in terms of mathematical model. The transformation of expert's knowledge in terms of control rules to fuzzy frame work has not been formalized and arbitrary choices concerning, for example, the shape of membership functions have to be made. The quality of fuzzy controller can be drastically affected by the choice of membership functions. Thus, methods for tuning the fuzzy logic controllers are needed. In this work, neural networks are used to solve the problem of tuning a fuzzy logic controller. The neuro fuzzy controller uses the neural network learning techniques to tune the membership functions while keeping the semantics of the fuzzy logic controller intact.⁹

ANFIS architecture contain five layers, a circle represents the fixed node, while a square represents an adaptive node. To explain the ANFIS principle, two inputs x, y and one output z will be considered. Among many FIS models, the Sugeno fuzzy model is commonly used due to its high interpretability and computational efficiency, and built-in optimal and adaptive techniques.¹⁰ The fuzzy models use if-then principle for the rules. The rules for a first order Sugeno fuzzy model can be expressed as:

Rule1 : if x is A_1 and y is B_1 , then $f_1 = p_1x + q_1y + r_1$
 Rule2 : if x is A_2 and y is B_2 , then $f_2 = p_2x + q_2y + r_2$ (2)

where A_i and B_i are the fuzzy sets in the antecedent, and p_i, q_i and r_i are the design parameters that are determined during the training process.¹¹ The ANFIS consists of five layers (Fig. 3):

Layer 1: Generate the membership grades

$$O_1^1 = \mu_{A_i}(x), i = 1,2$$

$$O_1^1 = \mu_{B_i=2}(y), i = 3,4 \quad (3)$$

where μ_{A_i} and μ_{B_i} can adopt any fuzzy membership function (MF).

Layer 2: Every node in this layer calculates the firing strength of a rule via multiplication

$$O_1^2 = w_i = \mu_{A_i}(x)\mu_{B_i}(y), i = 1,2 \quad (4)$$

Layer 3: Normalize the firing strengths

$$O_1^3 = \bar{w}_i = \frac{w_i}{w_1 + w_2}, i = 1,2 \quad (5)$$

Layer 4: In this layer, every node, i , has the following function:

$$O_1^4 = \bar{w}_i f_i = \bar{w}_i(p_i x + q_i y + r_i), i = 1,2 \quad (6)$$

where w_i is the output of layer 3, and $\{ p_i, q_i, r_i \}$ are the parameters to be set. The parameters in this layer are referred to as the consequent parameters.

Layer 5: Computes the overall output as the summation of all incoming signals, which is expressed as:

$$O_1^5 = \sum_{i=1}^2 \bar{w}_i f_i = \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2} \quad (7)$$

The output z in Figure 3 can be rewritten as,¹²⁻¹⁵

$$f = (\bar{w}_1 x)p_1 + (\bar{w}_1 y)q_1 + (\bar{w}_1)r_1 + (\bar{w}_2 x)p_2 + (\bar{w}_2 y)q_2 + (\bar{w}_2)r_2 \quad (8)$$

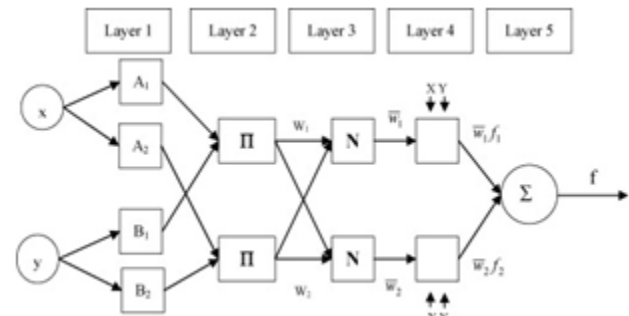


Figure 3 ANFIS Architecture

The ANFIS structure in this study is based on:

- 1) The consequent part of fuzzy if-then rules is a linear equation by choosing a first order Sugeno model.
- 2) Algebraic product is used as the T-norms operator to performs fuzzy AND.
- 3) The training is done by using a sinusoidal wave as input signal to the transfer function model as shown in Figure 4
- 4) The generalized bell functions are used as the input membership functions (MF) which can be expressed as:

$$\mu_{A_i}(x) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}} \quad (9)$$

where a is half the width of the (MF), b (together with a) controls the slopes at the crossover points (where the MF value is 0.5) and c determines the center of the MF.

The computational time is reduced by using only one input and three rules is used, so that Equation (7) becomes

$$f = (\bar{w}_1 x)p_1 + (\bar{w}_1 y)q_1 + (\bar{w}_1)r_1 + (\bar{w}_2 x)p_2 + (\bar{w}_2 y)q_2 + (\bar{w}_2)r_2 + (\bar{w}_3 x)p_3 + (\bar{w}_3 y)q_3 + (\bar{w}_3)r_3 \quad (10)$$

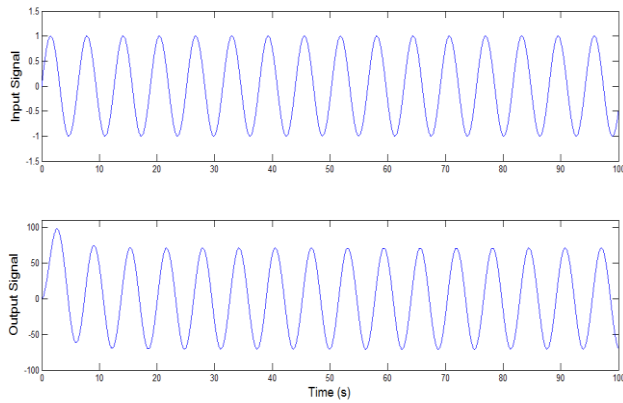


Figure 4 Training data

The training algorithm requires a training set defined between inputs and output. Several inputs are used to get the suitable signal for the system training. Among which, the sine wave, in this case, is the best signal in order to get the training data (Figure 4). The parameters to be trained are a , b , and c of the premise parameters and p , q , and r of the consequent parameters (Figure 5—shows the resulted input membership functions from the training process, which have three memberships negative (N), zero (Z) and positive (P)).

The training data are used to train the ANFIS controller, as mentioned before. ANFIS toolbox in MATLAB/SIMULINK is used as the tool to design the controller. At first, the data is received from the workspace in MATLAB, then, the generalised bell membership function (MF) is used as the input MF type after examining different types such as triangular and trapezoidal MF. The output MF is Sugeno since it is the only type that ANFIS deals with. Three MFs are used for both the input and the output and they were optimized (The results are shown in Figure 5 and Figure 6 respectively).

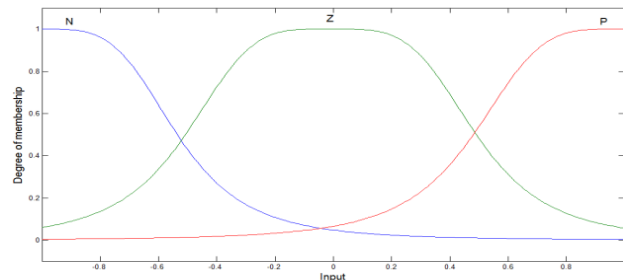


Figure 5 Input membership functions

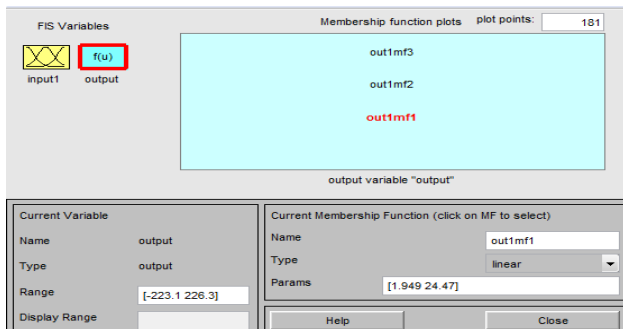


Figure 6 Output membership functions

3.2 Receding Horizon Controller

The receding horizon control is a model predictive control approach. In this type of control, the control law is calculated by solving an open-loop optimization problem for a fixed optimization window (prediction length), providing that the current states of the plant, $x(k_i)$, are available. This procedure is carried out for all iteration (for each sampling instant). Based on the plant model, the controller is able to predict the output for P_h (prediction horizon) steps in the future, and calculate the control trajectory for C_h (control horizon) steps in the future. The control horizon must be less than the prediction horizon because the current output is independent of the current control signal; that is the current control signal results in the next output (Figure 7—illustrates the different signals and labels that are dealt with when using a discrete RHC). In other words, at time instant k , the output is predicted till $(k + P_h)$ steps providing that the optimal control signal is calculated for $(k + C_h)$ steps.

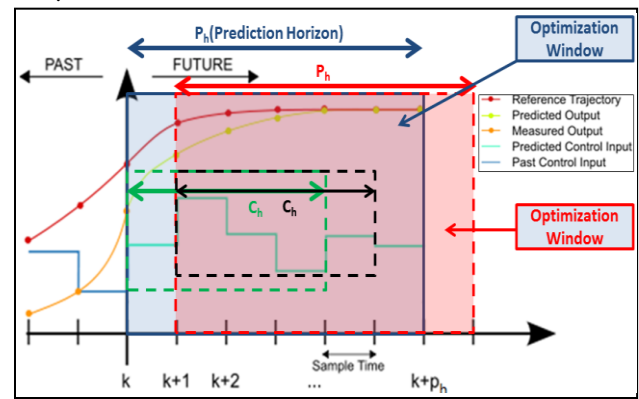


Figure 7 A discrete RHC scheme

The principle of receding horizon states that even though the control trajectory is calculated for C_h steps in the future, only the first part of this trajectory is applied to the plant.¹⁶ At the next time instant $(K+1)$, the output is predicted again for $(k + P_h)$ steps in the future, i.e. until $(k + P_h + 1)$ and another optimization window is formed (The red-color window in Fig. 7). The control trajectory is calculated as before for $(k + C_h)$, i.e. until $(k + C_h + 1)$. This procedure is repeated for all coming time instants, and that is why it is called the receding Horizon Principle

There are many formulations for RHC, which can be a continuous-time or a discrete-time formulation for either linear or nonlinear systems. In this study, a linear discrete-time receding horizon controller is chosen since the transfer function of the system is linear. The formulation used for this controller is based on the formulation presented in L. Wang.¹⁷ The following is a guidance of the control law formulation.

The discrete-time state space model of the system is presented in (11),

$$\begin{aligned} x_m(k + 1) &= A_m x_m(k) + B_m u(k), \\ y(k) &= C_m x_m(k), \end{aligned} \tag{11}$$

By modifying the state space model, yields the following model in (12) which is to be used in the design of RHC controller.

$$\begin{bmatrix} x(k+1) \\ \Delta x_m(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} A \\ A_m & o_m^T \\ C_m & A_m & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ \Delta x_m(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} B \\ B_m \\ C_m B_m \end{bmatrix} \Delta u(k)$$

$$y(k) = \begin{bmatrix} o_m^T & 1 \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} \quad (12)$$

where,
 $\Delta x_m(k) = x_m(k) - x_m(k - 1)$;
 $\Delta x_m(k + 1) = x_m(k + 1) - x_m(k)$;
 $o_m = [0 \ 0 \ \dots \ 0]$; n is the order of the system.

Let $Y = Fx(k_i) + \Phi \Delta U$ where,

$$Y = \begin{bmatrix} y(k_i + 1 | k_i) \\ y(k_i + 2 | k_i) \\ y(k_i + 3 | k_i) \\ \vdots \\ y(k_i + P_h | k_i) \end{bmatrix}; \Delta U = \begin{bmatrix} \Delta u(k_i) \\ \Delta u(k_i + 1) \\ \Delta u(k_i + 2) \\ \vdots \\ \Delta u(k_i + C_h - 1) \end{bmatrix}; F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{P_h} \end{bmatrix};$$

$$\Phi = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{P_h-1}B & CA^{P_h-2}B & CA^{P_h-3}B & \dots & CA^{P_h-C_h}B \end{bmatrix}$$

where $\Delta u(k_i + j)$ is the future control movement and $j = 0, 1, \dots, C_h$.

Assuming that set-point is $R_s^T = \overbrace{[1 \ 1 \ \dots \ 1]}^{P_h} r(k_i)$, then the cost function J for this control objective is defined as,

$$J = (R_s - Y)^T (R_s - Y) + \Delta U^T \bar{R} \Delta U \quad (13)$$

where $\bar{R} = r_w I_{C_h \times C_h}$ and r_w is used as tuning parameter by which the control signal is constrained more as it is increased. Minimizing the cost function, $\frac{\partial J}{\partial \Delta U} = 0$, yields the optimal control movement, ΔU , which is to be added to the previous control signal. Equation (14) represents the control law for the RHC controller.

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (\bar{R}_s r(k_i) - Fx(k_i)),^{17} \quad (14)$$

From (14), the matrices F and Φ must be calculated so that the control movement is calculated after. Although ΔU is a vector that contains the future control movement, only the first element of this vector is applied to the plant. This is illustrated in the RHC algorithm flowchart (Figure 8).

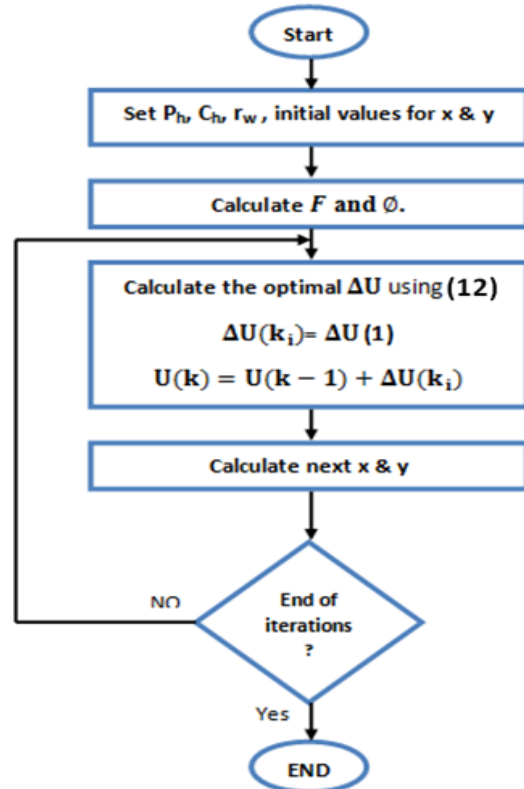


Figure 8 Flowchart of the RHC algorithm

In the design of this particular controller, the prediction horizon, P_h , is set to 4, the control horizon, C_h , is set to 3 and the tuning parameter, r_w , is set to 1. If the performance is not enhanced a lot, it is not recommended to choose larger prediction horizon or control horizon as the size of both matrices F and Φ will be increased and this will cost more time for the calculations and thus slower down the algorithm.

The position transfer function in Equation (1) is directly converted to state-space model as in Equation (15),

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} = \begin{bmatrix} 1.1470 & -0.9434 & 0.5826 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} [U(t)]$$

$$y(k) = [0.0330 \ 0.0413 \ 0.2105] \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + [0] [U(t)] \quad (15)$$

By applying the receding horizon algorithm, the matrices $\Phi^T \Phi$, $\Phi^T R$ and $\Phi^T F$ are calculated as detailed above. Next, the control signal movement trajectory (a vector with the size of C_h) is calculated using Equation (14). Only the first element of this vector is then added to the previous control signal and then applied to the plant at the current time instant. This procedure is repeated at each time instant.

To this point, the design of both controllers was covered. In the next section the results of both controllers are presented and compared.

4.0 RESULTS AND DISCUSSION

In this section, the results of ANFIS, PI-ANFIS and the RHC controllers are presented, discussed and compared with PI and Pole-placement controllers in the work of A. A. M. Faudzi.⁸ ANFIS position controller is implemented in MATLAB/SIMULINK (Figure 9). As seen, till now the ANFIS controller is applied to model without adding the proportional integral gain *PI* to test the exclusive response when using this controller. The input and the output membership functions (in Figure 5 and Figure 6) are loaded to the Neuro-fuzzy controller in the SIMULINK circuit (in Figure 9). The controller has one input which is the error and one output which is the resulted control signal to be sent to the plant directly.

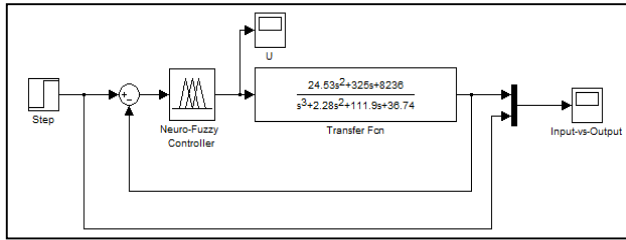


Figure 9 SIMULINK diagram for ANFIS position controller

The response of this controller (Figure 10) has a good settling time and a very small steady-state error. However, the overshoot percentage is significantly high; about 30%.

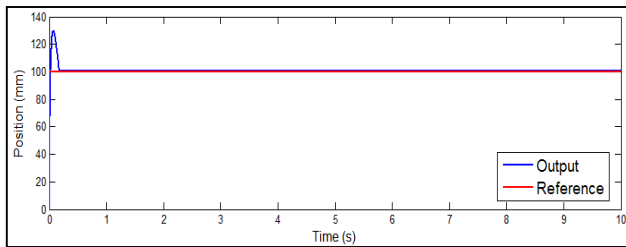
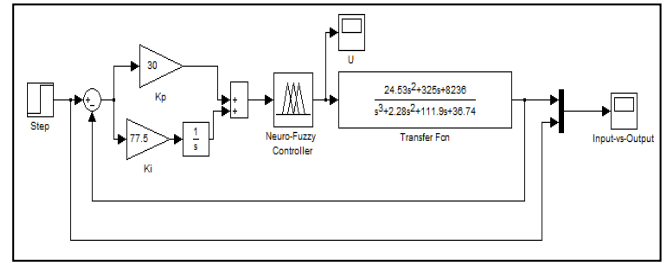
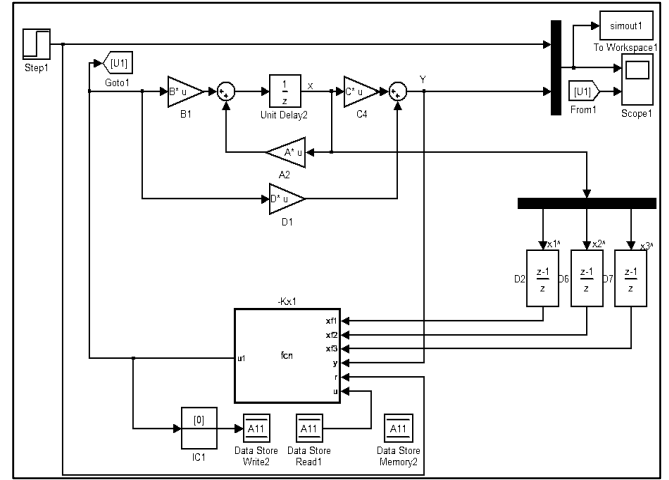


Figure 10 ANFIS controller results

To improve the response of the ANFIS controller, a proportional integral *PI* controller has been added to the ANFIS controller (Figure 11(a)).



(a)

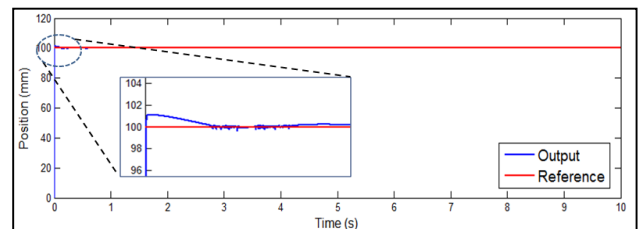


(b)

Figure 11 SIMULINK diagram for (a) PI-ANFIS; (b) RHC

Likewise, the receding horizon controller is also implemented via SIMULINK (Figure 11(b)). As seen, the plant model is implemented in its discrete state-space form to have a direct feedback from the three states of the system. The controller's inputs are the three states, the output signal, the reference signal and the previous control signal (to be added to the following control signal). This very MATLAB embedded function block contains the RHC algorithm and is executed at each time instant to calculate the current control signal and then send it to the plant.

The step response for PI-ANFIS and RHC controllers are shown in Figure 12. From Figure 12(a), adding the PI controller to the ANFIS controller significantly reduces the overshoot. Moreover, PI-ANFIS has faster response with settling time of 0.15 s compared to RHC, which has a settling time of 0.25 s



(a)

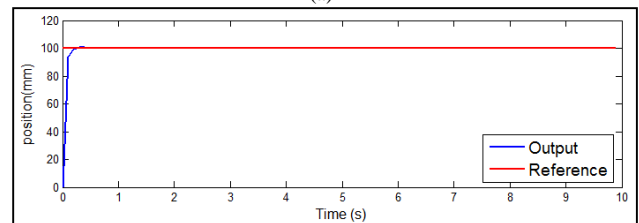


Figure 12 Step response for (a) PI-ANFIS; (b) RHC

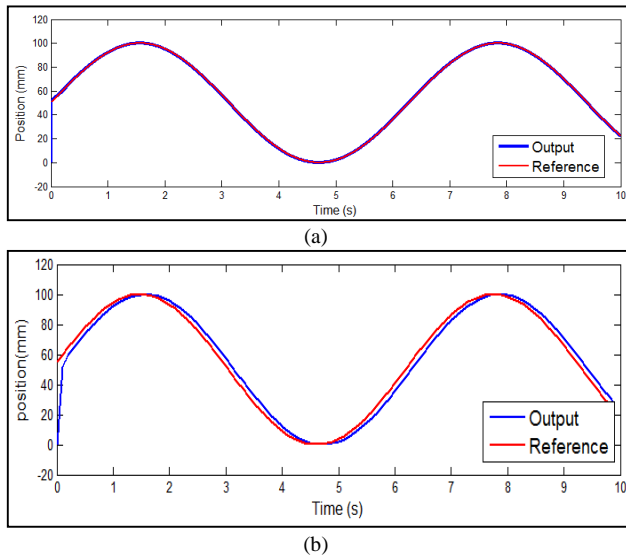


Figure 13 Sinusoidal response for (a) PI-ANFIS; (b) RHC

The controller's abilities to track sinusoidal wave are shown in Figure 13. In this case, the PI-ANFIS perfectly tracks the reference compared to RHC whose response has a small delay between the output and the reference.

The controllers were further tested with multistep reference (Figure 14). Both controllers are able to track the reference within the operating range of the IPA. Still, the PI-ANFIS controller has better response than the RHC controller.

Finally, the step responses of the PI-ANFIS and the RHC controllers are further compared with the work of A. A. M. Faudzi⁸ in which PI and feedback controllers has been applied to

control the position of the same plant (the IPA). (Table 1 - shows the comparison for step response for the four controllers). Although PI and feedback controllers shows 0% overshoot while this study shows 0.6% and 1.1% for RHC and PI-ANFIS controllers respectively, but this amount of overshoot is insignificant especially with the very short settling and rising time, and also with the very small percentage of the steady state error compared to PI and feedback controllers.

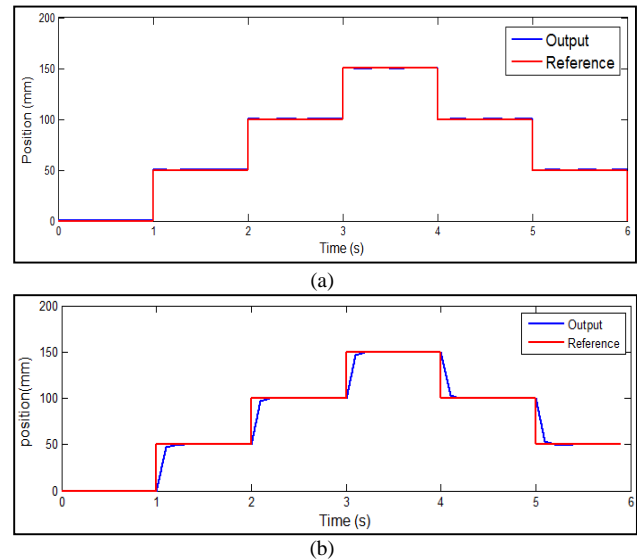


Figure 14 Multistep response for (a) PI-ANFIS; (b) RHC

Table 1 Comparison for step response position tracking

Analysis	PI Controller	Feedback Controller	RHC Controller	PI-ANFIS Controller
Overshoot (%OS)	0%	0%	0.6%	1.1%
Settling time	4s	1.25s	0.25s	0.14s
Rise time	2.05s	0.8s	0.085s	0.01s
Steady state error (%e _{ss})	0.01%	0.01%	0%	0.003%

5.0 CONCLUSION

In this paper, PI-ANFIS and RHC controllers has been designed and analyzed for IPA position control. Unlike common fuzzy and Neuro-fuzzy controllers that usually comprise at least two inputs, the proposed Neuro-fuzzy controller has only one input, which is the error, and that reduce the computational time, which yields faster response. A significant amount of overshoot occurred as result of using single input and it was eliminated by adding PI controller to the ANFIS controller and resulted faster response as well.

PI-ANFIS is better in terms of settling and rise time. In the other hand, RHC has no steady state error and less overshoot.

The results of both proposed controllers show significant improvement in the response over the widely used PI controller and also over the feedback controller.

This study was conducted by MATLAB/SIMULINK. As a future work, real time controller will be conducted with the real IPA plant using the two proposed controllers.

References

- [1] H. I. Ali, S. B. B. M. Noor, S. M. Bashi, M. H. Marhaban. 2009. A Review of Pneumatic Actuators (Modeling and Control). *Australian Journal of Basic and Applied Sciences*. 2: 440–454.

- [2] A. A. M. Faudzi, K. Suzumori, S. Wakimoto. 2009. Development of an Intelligent Pneumatic Cylinder for Distributed Physical Human-Machine Interaction. *Advanced Robotics*. 23: 203–225.
- [3] Khairuddin Osman. 2012. *Member, IEEE*, Ahmad 'Athif Mohd Faudzi, *Member, IEEE*, M.F. Rahmat, Nu'man Din Mustafa, M. Asyraf Azman, Koichi Suzumori. *Member, IEEE. System Identification Model for an Intelligent Pneumatic Actuator (IPA) System*. IROS 2012
- [4] A. A. M. Faudzi, K. Suzumori, S. Wakimoto. 2010. Development of an Intelligent Chair Tool System Applying New Intelligent Pneumatic Actuators. *Advanced Robotics*. 24: 1503–1528.
- [5] A. A. M. Faudzi, K. Suzumori. 2010. Programmable System on Chip Distributed Communication and Control Approach for Human Adaptive Mechanical System. *Journal of Computer Science*. 6(8): 852–861.
- [6] A. A. M. Faudzi. 2010. Development of Intelligent Pneumatic Actuators and Their Applications to Physical Human-Mechine Interaction System. Ph.D. thesis, The Graduate School of Natural Science and Technology, Okayama University, Japan.
- [7] A. A. M. Faudzi. 2012. *Member*. Khairuddin Osman, M.F. Rahmat, Nu'man Din Mustafa, M. Asyraf Azman, Koichi Suzumori. 2012. *Nonlinear Mathematical Model of an Intelligent Pneumatic Actuator (IPA) Systems: Position and Force Controls*, AIM 2012.
- [8] A. A. M. Faudzi, Khairuddin bin Osman, M. F. Rahmat, Nu'man Din Mustafa, M. Asyraf Azman, Koichi Suzumori. 2012. Controller Design for Simulation Control of Intelligent Pneumatic Actuators (IPA) System. *Procedia Engineering*. 41: 593–599.
- [9] J-SR, Jang. 1993. ANFIS: Adaptive-Network-based Fuzzy Inference System. *Systems, Man and Cybernetics, IEEE Transactions on*. 23(3): 665–685.
- [10] Tahour, H. Ahmed, A. Hamza, A. A. Ghani. 2007. Adaptive Neuro-Fuzzy Controller of Switched Reluctance Motor. *Serbian Journal of Electrical Engineering*. 4(1): 23–34.
- [11] Denal, A. Mouloud, P. Frank, Z. Abdelhafid. 2004. ANFIS Based Modelling and Control of Non-linear Systems: A Tutorial. *Systems, Man and Cybernetics, 2004 IEEE International Conference* vol. 4.
- [12] V. A. Constantin. 1995. *Fuzzy Logic and Neuro-Fuzzy Applications Explained*. Englewood Cliffs, Prentice-Hall.
- [13] C. T. Lin, C. S. G. Lee. 1996. *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*. Upper Saddle River, Prentice-Hall.
- [14] N. K. Kim. 1999. HyFIS Adaptive Neuro-fuzzy Inference Systems and Their Application to Nonlinear Dynamical Systems. *Neural Networks*. 12(9): 1301–19.
- [15] D. D. Popa, Aurelian Craciunescu, Liviu Kreindler. 2008. A PI-Fuzzy Controller Designated for Industrial Motor Control Applications. *Industrial Electronics. ISIE 2008. IEEE International Symposium*. IEEE.
- [16] C. E Garcia, D. M. Prett, M. Morari. 1989. Model Predictive Control: Theory and Practice-a Survey. *Automatica*. 25(3): 335–348.
- [17] L. Wang. 2009. *Model Predictive Control System Design and Implementation Using MATLAB*. Springer books. 1: 40.