Sādhanā Vol. 35, Part 1, February 2010, pp. 75–86. © Indian Academy of Sciences

Pneumatic motor speed control by trajectory tracking fuzzy logic controller

CENGIZ SAFAK, VEDAT TOPUZ and A FEVZI BABA

Department of Electronics and Computer Education, Marmara University, Istanbul 34046, Turkey
e-mail: censaf@yahoo.com; vtopuz@marmara.edu.tr; fbaba@marmara.edu.tr

MS received 2 August 2008; revised 21 October 2009; accepted 5 November 2009

Abstract. In this study, trajectory tracking fuzzy logic controller (TTFLC) is proposed for the speed control of a pneumatic motor (PM). A third order trajectory is defined to determine the trajectory function that has to be tracked by the PM speed. Genetic algorithm (GA) is used to find the TTFLC boundary values of membership functions (MF) and weights of control rules. In addition, artificial neural networks (ANN) modelled dynamic behaviour of PM is given. This ANN model is used to find the optimal TTFLC parameters by offline GA approach. The experimental results show that designed TTFLC successfully enables the PM speed track the given trajectory under various working conditions. The proposed approach is superior to PID controller. It also provides simple and easy design procedure for the PM speed control problem.

Keywords. Pneumatic motor; fuzzy logic controller; trajectory tracking.

1. Introduction

PMs are widely used in industry due to their special advantages. The motors are of simple construction, widely sourced, and easily maintained making them low in cost. They have a high power-to-weight ratio, and unlike electric motors, can apply a force at a fixed position over a prolonged period with no ill effects. Like electric motors, PMs operate cleanly; like hydraulic motors, they may act directly on a payload. They operate at relatively high speeds in industrial and spark-prohibited applications. They can be regulated easily for speed and torque, and can stop and reverse very quickly. They play a very significant part as prime movers because they are relatively cheap, easy to maintain, and have the versatility of variable speed, high starting torque. They are also intrinsically safe in hazardous areas and can operate in exceptionally harsh environments (Anonymous 1999).

Although PMs are very often used in industrial applications, PM speed control problem has been considered very difficult especially in low speed values due to its nonlinear dynamics and compressibility of air (Jihong *et al* 1998). Many control strategies such as, PID controller, adaptive controller, FLC and ANN controller have been tried to solve this problem (Chillari *et al* 2001; Shibata *et al* 2000; Marumo & Tokhi 2004). In this study, trajectory tracking fuzzy

logic controller (TTFLC) is designed and applied for the speed control of PM for any given trajectory under various working conditions. Although TTFLC produce superior results to that of classical FLC controllers, this approach have difficulties for the designers to get an insight on how the designed membership functions and fuzzy rule affect the performance of the system (Baba 2004; Ruihua *et al* 2004). These difficulties can be eliminated by using GA to find optimal TTFLC parameters (Herera *et al* 1994; Linkens & Nyongesa 1995). Offline GA learning approach requires the usage of a dynamic PM model. The mathematical model of PM is nonlinear and depends on their physical parameters which are very hard to measure. Therefore, it is difficult to obtain the mathematical model of the PM (Pandian *et al* 1999). These difficulties forced us to use ANN to model the dynamic behaviours of PM. In this study, we have chosen the nonlinear Auto Regressive (NARX) approach for ANN model. Optimal ANN parameters such as number of hidden layer, neuron number in hidden layers and learning rate are found because these values greatly affect the performance of ANN (Zhang *et al* 1998). To demonstrate the effectiveness of this approach, a series of experimental studies were carried out and results were compared to that of PID controller. The experiments show that TTFLC is superior to PID controller and it provides a simple and easy design procedure for the speed control of PM. TTFLC allow the user to maintain the desired motor speed by defining the desired trajectory profile easily.

This paper is organized as follows. Section 2 deals with, realized pneumatic system, its trajectory and PM ANN models. The designed TTFLC and GA, FLC parameters are given in section 3. Section 4 presents experimental results. Finally, conclusion is provided in section 5.

2. System description and analysis

Experimental structure of the pneumatic and realized systems are consecutively presented in figures 1a and 1b. The pneumatic system is supplied with a constant air pressure (6 bars) in order to decrease fluctuation, and high pressure receiver tank is added to the pneumatic system. The pneumatic motor (Gant 2AM-NCW-7B) speed is controlled by 3-way proportional solenoid valve, FESTO MPYE–5, (Gast air motor manual). In addition, motor speed is measured by linear tachogenerator, and data acquisition board (Adventech PCI 1711) is used for control signal generation.

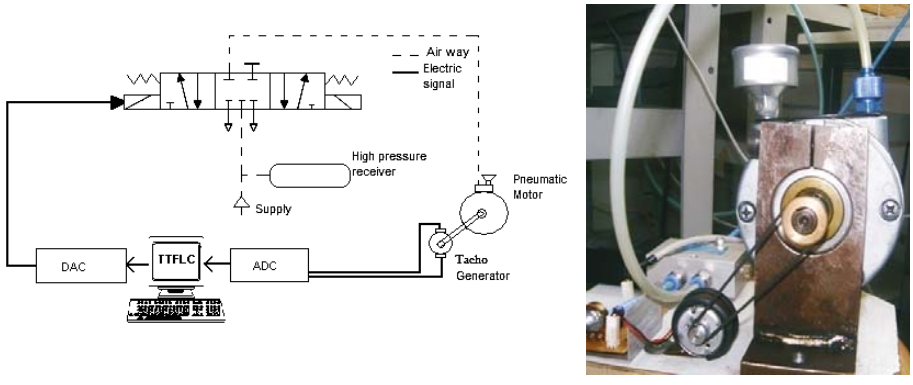

Figure 1. **(a)** Block diagram of the pneumatic system. **(b)** Realized system.

2.1 *Pneumatic motor ANN models*

Generally, system modelling means measuring the inputs, outputs and finding the physical system state variables with appropriate learning approach. The input–output data set forming a system model is a data group that contains values sampled in sufficient numbers. ANN can be used to model any linear or nonlinear system structures using these data sets. We obtain these input–output data sets using open loop behaviours of PM. This modelling scheme is usually based on supervised learning and by this way the ANN can be interpreted as a NARX model as follows (Ronco & Gawthrop 1997).

$$\hat{y}_{(k)} = f(y_{(k-1)}, \ldots, y_{(k-n)}; u_{(k-d)}, \ldots, u_{(k-m)}) + e_{(k)}, \tag{1}$$

where $\hat{y}_{(k)}$ is approximated system output, $y_{(k)}$ and $u_{(k)}$are the sampled process output and input at sample k. n, m denote the number of past inputs and outputs used in the model respectively, also d is a transport lag. f is a nonlinear function describing the system behaviour and $e_{(k)}$ is the approximation error.

Most ANN applications involve multi layer perceptron (MLP) model. The basic MLP building unit is a simple model of artificial neurons. Typical MLP network is arranged in layers of neurons, where each neuron in a layer computes the sum of its inputs and passes this sum through an activation function (ϕ). The output of the network is defined as a matrix form;

$$\boldsymbol{\hat{y} = \phi^2(W^2\phi^1(W^1x + b^1) + b^2)}, \tag{2}$$

where; superscript defines the layer number, $S0$ is input number, $S1$ is hidden layer neuron number, exp is base of natural logarithm, n is activation function input, W is weight matrices;

$$\boldsymbol{W^1} = \begin{bmatrix} w_{1,1} & w_{1,2} & \ldots & w_{1,S0} \\ w_{2,1} & w_{2,2} & \ldots & . \\ . & . & \ldots & w_{2,S0} \\ w_{S1,1} & w_{S1,2} & \ldots & w_{S1,S0} \end{bmatrix}, \quad \boldsymbol{W^2} = [w_{1,1} w_{1,2} \ldots w_{1,S1}],$$

b is bias vector; $b^1 = [b_1 b_2 \ldots b_{S1}]^T$, $b^2 = [b_1]$ subscript defines the neuron number, superscript defines the layer number, ϕ is activation functions; $\phi^1 = \frac{\exp^n - \exp^{-n}}{\exp^n + \exp^{-n}}$, $\phi^2 = n$, and $\boldsymbol{x}$ is input vector; $\boldsymbol{x} = [u_{(k)} u_{(k-1)} y_{(k-1)} y_{(k-2)} y_{(k-3)}]^T$.

Realized MLP network is shown in figure 2. MLP networks learn any input–output relations by adjusting the weights using backpropagation approach. The backpropagation algorithm is the generalization of the least mean square (LMS) algorithm (Hagan *et al* 1996). This algorithm adjusts the weights in order to minimize the mean square error;

$$e = \frac{1}{2} \sum_{q=1}^{Q} (y_q - \hat{y}_q)^2, \tag{3}$$

where; Q is the sample size.

The steepest descent algorithm iteratively decreases network error during learning phase at each epoch as given below;

$$w(z+1) = w(z) - \eta \frac{\partial e}{\partial w}; \quad b(z+1) = b(z) - \eta \frac{\partial e}{\partial b}, \tag{4}$$

where; η is the learning rate, z is the epoch number.

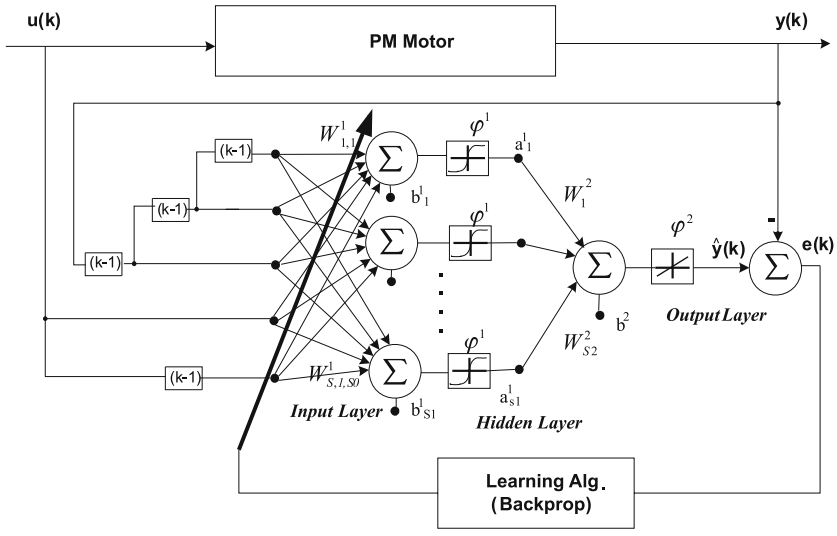

Figure 2. Modelling of the PM dynamic behaviour using MLP-based NARX structure.

The number of hidden layer and neurons in the hidden layer(s) play very important roles for ANNs and choice of these numbers depends on the application. Theoretical works have proved that a single hidden layer is sufficient for ANNs to approximate any complex nonlinear function with any desired accuracy (Zhang *et al* 1998). Determining the optimal number of hidden neurons is also a question to deal with. Although there is no theoretical basis for selecting these parameters, a few systematic approaches are also reported but the most common way of determining the number of hidden neurons is still trial and error approach. Therefore, five neurons in the input layer and 15, 20, 25, 30, 35 neurons in hidden layer are selected to find optimal MLP model. Designed MLP network is trained using backpropagation learning algorithm, which is described above.

Once trained, the ANN can be used for modelling the system. This is called the 'generalization property' of the network. Generalization property of trained MLP is performed by using the testing data set which is selected randomly. The obtained RMS error and correlation coefficient (R) in the different networks are given in table 1.

Table 1. Different MLP models training and testing RMS error and regression coefficients.

	Learning phase		Testing phase	
ANN models	RMS error %	Correlation coefficient (R)	RMS error %	Correlation coefficient (R)
MLP-15	16·5	0·684	42·7	0·589
MLP-20	2·14	0·891	2·70	0·809
MLP-25	1·72	0·917	2·09	0·826
MLP-30	1·63	0·976	1·84	0·956
MLP-35	1·50	0·984	1·67	0·974

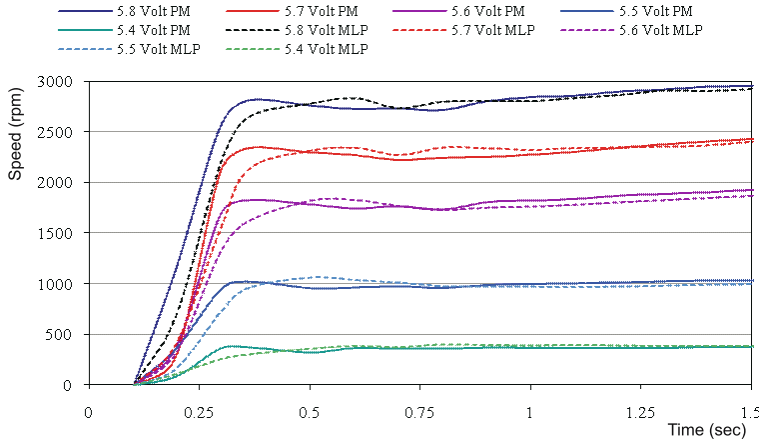

Figure 3. PM behaviour and MLP-35 models at the different control voltages.

MLP-35 network can perform a desired correlation (R = 0·974) between input and output. That means the system can be substituted by MLP-35 at the TTFLC design phase. Figure 3 clearly shows ANNs testing results with a comparison of actual PM output at different control voltage.

2.2 *Trajectory*

The pneumatic motor speed trajectory is defined as a third order function as follows:

$$\begin{aligned} p_y &= p_0 && t_0 < t \leq t_1 \\ p_y &= f_3(t-t_1)^3 + f_2(t-t_1)^2 + f_1(t-t_1) + f_0 && t_1 < t \leq t_2 \\ p_y &= p_s && t_2 < t \leq t_3, \end{aligned} \tag{5}$$

where, p_o, p_s are initial and desired speed values of the trajectory. f_{0-3} are trajectory coefficients. These can be calculated from the initials speed values, desired speed values and slope (θ) of trajectory (Baba 2004).

3. Designing of a GA-based trajectory tracking fuzzy logic control

Several studies have shown that FLC is an appropriate method for the control of complex or partially identified processes, many of which cannot easily be modelled in a mathematical way. Unlike a conventional controller, no rigorous mathematical model is required to design a FLC and in many cases, they can be implemented easily. However, this simplicity also presents a bottleneck in their design. FLC relies on heuristic knowledge that is subject to designer's interpretation and choice. The traditional approach to fuzzy design is laborious, time consuming and in most cases specific to each application. Optimal search algorithms such as GAS can solve some of these problems (Herrera *et al* 1994; Linkens & Nyongesa 1995).

GA is the process of searching the most suitable chromosomes that build the population in the potential solutions space. GA starts parallel searching from independent points of search

Table 2. The best values of SGA parameters.

SGA parameters	Value	SGA parameters	Value
Population size (μ)	30	Mutation operator (ω_m)	Bit mutation
Crossover operator (ω_c)	Single point	Mutation probability (PM)	0·01%
Crossover probability (PC)	90 %	Selection operator (S)	Tournament

space in which the solution knowledge is poor or not available. The solution depends on interaction of the surroundings and genetic operators. For that reason, obtaining the suboptimal solutions of GA is a small probability. It is particularly presented as an alternative for the traditional optimal search approaching in which it is hard to find the global optimum point in nonlinear and multi-modal optimizations problems. GA provides solutions using randomly generated bit strings for different types of problems. Simple GA (SGA) is defined by 11 parameters given as below.

$$\mathrm{GA}(\Sigma, G, \rho, \phi, \mu, I, S, Gap, \Omega, R, \tau), \tag{6}$$

where, Σ represents the search space, G is the coding space, ρ is the coding function ($\rho : \Sigma \rightarrow G$), ϕ is the fitness function ($\phi : \Sigma \rightarrow R^{+}$), μ is the population size, I is the initialization function ($I : \mu \rightarrow P(G)$), ($P(G)$) is the population, S is the selection type, Gap is the percentage of the population that is conveyed from old generation to new generation, Ω is the genetic operator (crossover operator: ωc and mutation operator: ωm), R is the replacement policy, and τ is the termination criterion.

Selection of the parameters depends on the type of the problem. In literature, there are many studies to find the best fit of the parameters (Bäck 1993). However, most of the studies depend on trial and error approach. The realized SGA optimal parameter values are found at the end of the numerous experiments, which are given in table 2.

The block diagram of the TTFLC of PM is shown in figure 4. In the proposed architecture zero order Sugeno type FLC are designed. The FLC has 2 input variables (error, change in error) and both of them have 5 and 3 trapezoid MFs trapezoid membership functions (MF).

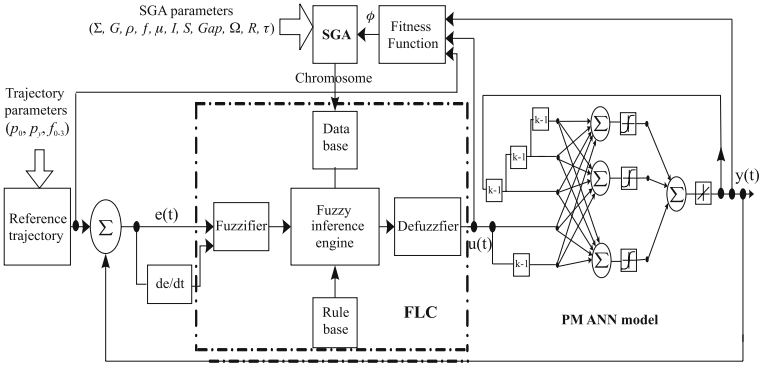

Figure 4. Designed TTFLC block diagram.

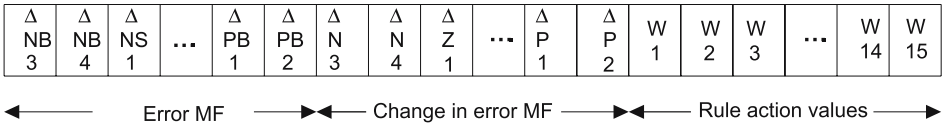

Figure 5. Realized chromosome structure.

Therefore, rule base consists of fifteen rules as follows:

$$R_{(1)} : \text{If (Error} = \text{NB) and (Change\_in\_error} = \text{N) then } (u = W_1)$$

$$R_{(2)} : \text{If (Error} = \text{NB) and (Change\_in\_error} = \text{Z) then } (u = W_2)$$

$$\cdots\cdots$$

$$R_{(15)} : \text{If (Error} = \text{PB) and (Change\_in\_error} = \text{P) then } (u = W_{15}).$$

The boundary values of membership functions in the antecedent part of fuzzy rules are determined by expert knowledge. Consequent part of fuzzy rule base which we call weight coefficients are obtained randomly. Variations of MFs (Δ MF) and rule weight coefficients (W) are coded as a 10 bit weighted binary code in a single chromosome which is shown in figure 5.

Another important feature of GA-based FLC is to transform the system output to fitness value (ϕ). Designed trajectory has different crossing points (t_1, t_2, t_3), and therefore fitness function has different weight factors ($\alpha = 0{\cdot}1, \beta = 0{\cdot}3, \delta = 0{\cdot}6$) for each trajectory region. To avoid excessive variation of control signal in the pneumatic motor, constrained fitness function is designed. This constrained fitness function is able to minimize RMS error in all trajectory regions;

$$\text{minimize } \phi = \left[\sum_{t0}^{t1} \alpha \sqrt{(p_{y(t)} - y_{(t)})^2} + \sum_{t1}^{t2} \beta \sqrt{(p_{y(t)} - y_{(t)})^2} + \sum_{t2}^{t3} \delta \sqrt{(p_{y(t)} - y_{(t)})^2} \right]$$

$$\text{subject to } (u_{(t-n)} - u_{(t-n-1)}) \leq \delta u \tag{7}$$

where ϕ is the fitness value, δu is the allowance limit on control signal variation, $y_{(t)}$, $p_{y(t)}$ are the systems and desire trajectory outputs respectively.

Defined fitness function is solved by penalty function approach as follows:

$$\begin{aligned} &\phi = \phi - \varepsilon;\ (u_{(t-n)} - u_{(t-n-1)}) > \delta u \\ &\phi = \phi;\ (u_{(t-n)} - u_{(t-n-1)}) \leq \delta u, \end{aligned} \tag{8}$$

where ε is penalty value.

Optimal values of TTFLC parameters are found by SGA at the end of 200 generations. Boundary values of membership functions are given in table 3. The output of a zero-order Sugeno model is a smooth function of its input variables as long as the neighbouring MFs in the antecedent have enough overlap (Jang *et al* 1997). From this point of view, we can conclude that we have a smooth control surface using MFs obtained by SGA that is given in figure 6.

Table 3. Obtained MFs boundary values by SGA.

	NB		NB2 = −3000	NB3 = −105·3	NB4 = −38·3
	NS	NS1 = −128	NS2 = −87·4	NS3 = −57	NS4 = −12·8
	ZE	ZE1 = −23·7	ZE2 = 5·2	ZE3 = 33·4	ZE4 = 62·5
	PS	PS1 = 0·5	PS2 = 58·9	PS3 = 120·3	PS4 = 177·2
Error	PB	PB1 = 157·4	PB2 = 133·4	PB3 = 3000	
Change in error	N		N2 = −50	N3 = −16·4	N4 = 1·3
	Z	Z1 = −16·3	Z2 = 6·2	Z3 = 21·5	Z4 = 40·2
	P	P1 = 10·3	P2 = 25·4	P3 = 50	

Obtained rule weights are given in table 4. Since zero order Sugeno type FLC rules have a crisp output, the overall output is obtained via weighted average as follows:

$$u = \frac{\sum_i c_i W_i}{\sum_i c_i} \tag{9}$$

where; c_i is the firing strength of each rule using min operator.

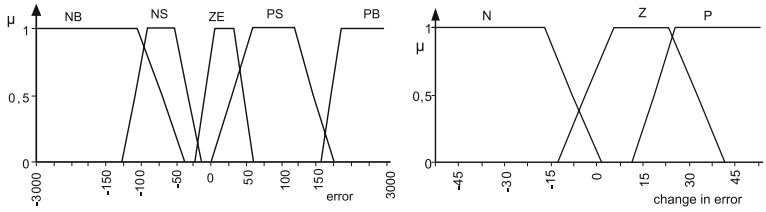

Figure 6. Figures of the error and change in error MFs.

Table 4. TTFLC rule weight table.

Rule	Error	Change in error	Rule weight (W)
1	NB	NE	−0·13585
2	NB	ZE	−0·86321
3	NB	PO	0·14803
4	NS	NE	−0·00684
5	NS	ZE	−0·18753
6	NS	PO	0·00206
7	ZE	NE	−0·00603
8	ZE	ZE	0
9	ZE	PO	0·00037
10	PS	NE	−0·00145
11	PS	ZE	0·21879
12	PS	PO	0·00940
13	PB	NE	−0·17836
14	PB	ZE	0·74936
15	PB	PO	0·27205

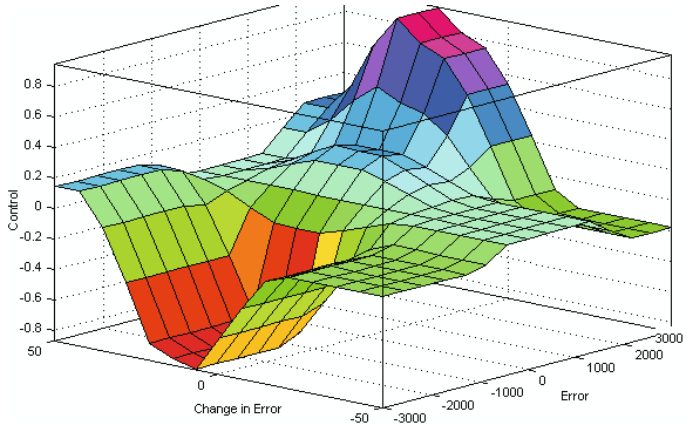

Figure 7. Related control surface of designed TTFLC.

Consequently, related control surface is given in figure 7. This control surface controls the pneumatic motor with satisfying results, which is described in section 4.

4. Experimental results

In order to test the performance and the robustness of the proposed TTFLC, a series of experiments are carried out under various working conditions. The results show that TTFLC is able to attain the PM to state error. We have also designed a PID controller, whose parameters are found by using Zeigler–Nichols (Z–N) method, to compare its results by those of TTFLC. After system open loop response is analysed, delay time (τd) and time constant (τ) are measured as 0·2 sec. and 0·4 sec. respectively. Then PID parameters $K_p = 2{\cdot}4$, $K_i = 0{\cdot}01$, $K_d = 0{\cdot}4$ are calculated using Z–N table.

The TTFLC and PID controller are compared using three different set points (750 rpm, 1250 rpm, and 2000 rpm.) with the slope ($\theta = 20$) and results are given in figure 8.

To compare the obtained results clearly, % overshoot and steady state error values are also given in table 5. These results show that TTFLC have small overshoot and steady state errors in every set point comparing to PID controller. Obtained results also show that it is very hard to control PM at low speed values.

To test the performance of designed TTFLC for effects of different slopes, PM is started up for 2000 rpm with 10, 20 and 30 sec. slope times. Designed TTFLC have satisfactory results at all slope times which are given in figure 9.

Finally, we have also tested the performance of TTFLC for changing slope and set points at the same time. It can be observed from figure 10 that TTFLC is able to bring the PM speed to the desired level following the given trajectory with small overshoot and steady state error.

5. Conclusion

In this study, a GA-based TTFLC is designed to control the speed of the pneumatic motor (PM). To demonstrate the effectiveness of TTFLC, a series of experiments were carried out. In addition to results discussed above, the following conclusions can be summarized:

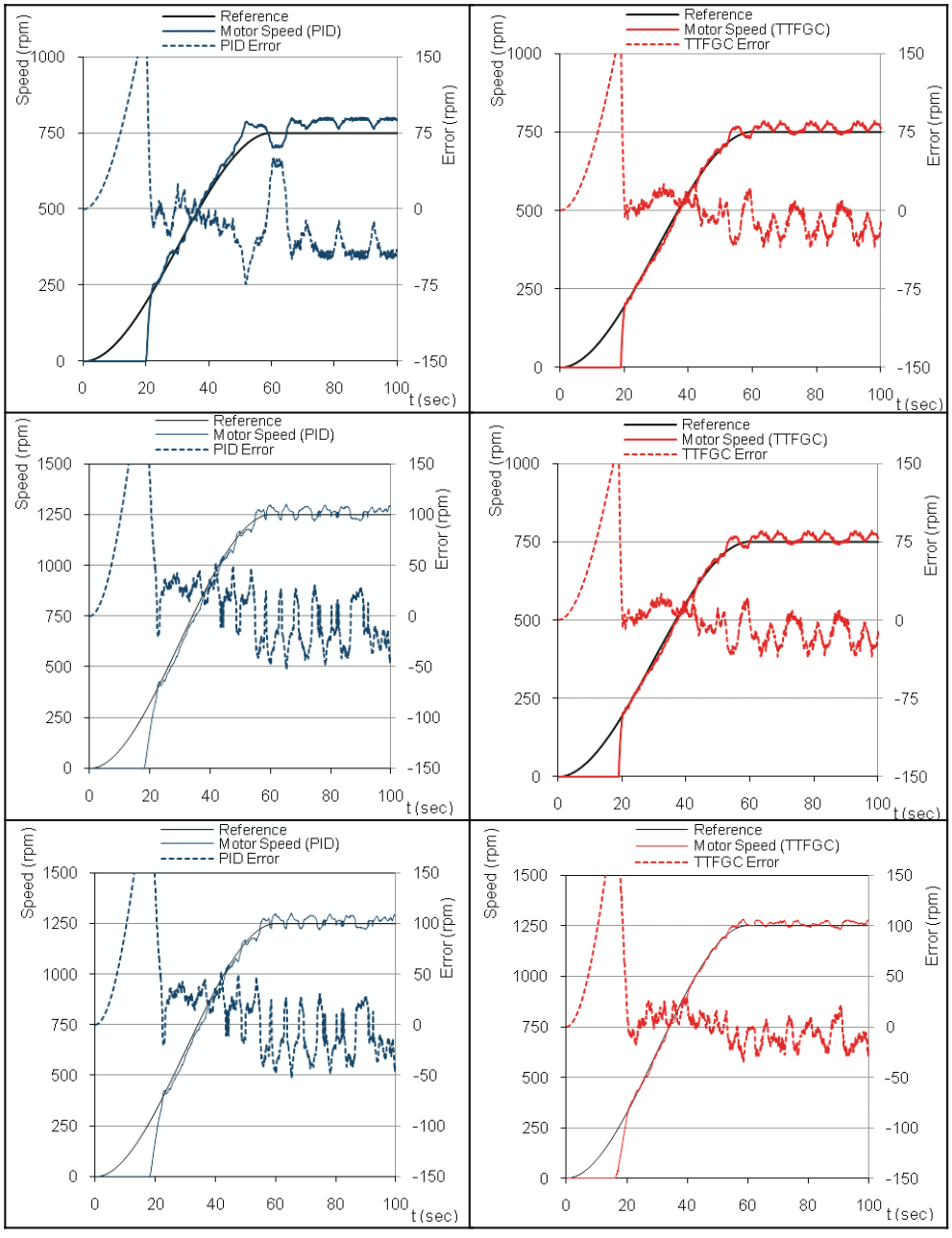

Figure 8. Pneumatic motor speed at different (750, 1250, 2000 rpm) set points.

(i) GA-based FLC is able to reduce the complex efforts which are used to find optimal FLC parameters.

(ii) When using a standard FLC, slope time of motor speed is defined by control rules. However, in TTFLC, the user determines the slope time values. Therefore, TTFLC

Table 5. TTFLC and PID controller overshoot and steady state error at different set points.

	750 rpm		1250 rpm		2000 rpm	
	Overshoot (%)	Steady state error (rpm)	Overshoot (%)	Steady state error (rpm)	Overshoot (%)	Steady state error (rpm)
TTFGC	3·6	$\sim +6-35$	1·8	$\sim +6-15$	0	$\sim +10-10$
PID	9·4	$\sim -22-49$	3·7	$\sim +22-33$	2·6	$\sim -27-52$

provides an easier way to operate the pneumatic motor at different slope times and working conditions. This method also allows the rule base to be composed of fewer rules.

(iii) TTFLC in the speed control of PM provides an accurate speed control. The speed follows the predetermined trajectory under various working conditions with small overshoot and steady state error.

(iv) TTFLC also provides a nonchattering control action that prolongs the PM lifetime and reduces system maintenance cost.

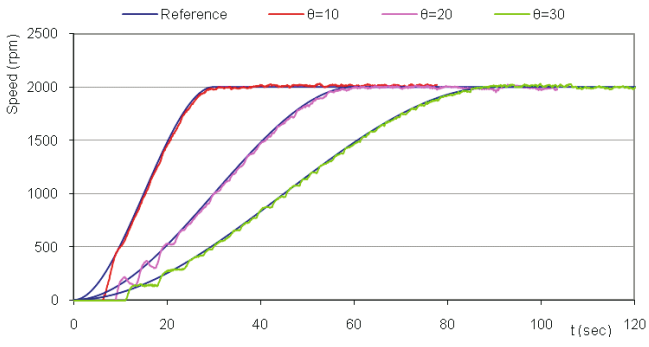

Figure 9. TTFLC performance at the different trajectory slope time.

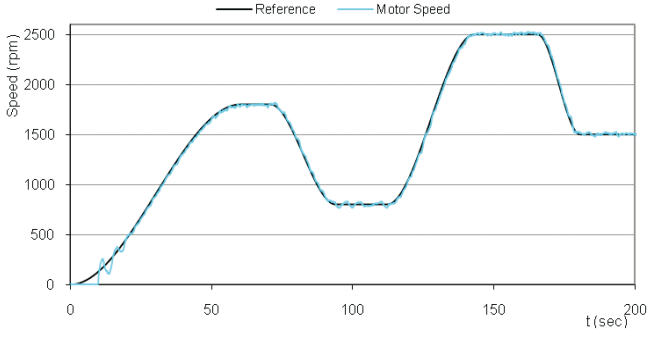

Figure 10. TTFLC performance at the changing trajectory.

References

Anonymous 1999 Air-motor advantage. Machine design, Nov 18, 124

Baba A F 2004 Fuzzy logic controller. *Nuclear Eng. Inter.* 49: 36–38

Bäck T 1993 Optimal mutation rates in genetic search. *Proceeding of the Fifth International Conference on Genetic Algorithms* (CL: Morgan Kaufmann)

Chillari S, Guccione S, Muscato G 2001 An experimental comparison between several pneumatic position control methods. *Proceeding of the 40th IEEE Conference on Decision and Control Orlando* FL

GAST *Air Motors manual for 2AM series* http://www.gastmfg.com

Hagan M T, Demuth H B, Beale M 1996 Neural network design (Boston, MA: PWS Publishing) 11·2–11·25

Herrera F, Lozano M, Verdegay J L 1994 Applying genetic algorithms in fuzzy optimization problems. *Fuzzy Systems and A.I. Reports and Letters* 3(1): 39–52

Jang J S R, Sun C T, Mizutani E 1997 Neuro-Fuzzy and Soft Computing (NJ: Prentice Hall) 82

Jihong W, Junsheng P, Philip R M, Zongmao Z 1998 Modelling study and servo-control of air motor systems. *Int. J. Control* 71(3): 459–476

Linkens D A, Nyongesa M 1995 Genetic algorithms for fuzzy control Part 1: Offline systems development and applications. *IEE Proc. Control Theory Applications* 142

Marumo R, Tokhi M O 2004 Neural-model reference control of an air motor. *IEEE Africon* 467

Marumo R, Tokhi M O 2004 Intelligent modelling and control of a pneumatic motor. CCECE 2004-CCGEI 2004, Niagara Falls

Pandian R S, Takemura F, Hayakawa Y, Kawamura S 1999 Control performance of an air motor. *Proceeding of the 1999 IEEE International Conference on Robotics and Automation*, Detroit, MI 518–524

Ronco E, Gawthrop P J 1997 Neural networks for modelling and control. Centre for System and Control Department of Mechanical Engineering University of Glasgow, Technical Report csc97008

Ruihua L, Weixiang S, Qingyu Y 2004 Multi-region fuzzy tracking control for a pneumatic servo squeezing forces system. *Proceedings of the 5 World Congress on Intelligent Control and Automation*, June 15–19, Hangzhou, P.R. China

Shibata S, Jindai M, Shimizu A 2000 Neuro-Fuzzy control for pneumatic servo system. IECON 2000 *26th Annual Conference of the IEEE* 3: 1761–1766

Zhang G, Patuwo B E, Hu M Y 1998 Forecasting with artificial neural networks: The state-of-the-art. *Int. J. Forecasting* 14: 35–62