Investigation on Advanced Control of A Linear Switched Reluctance Motor

Zou Y Cheng K. W. E Cheung N. C

Abstract–Advanced control systems are increasingly employed for intelligent factories. Fuzzy logic control (FLC) and backward propagation neural network (BPNN) control are investigated in this paper to realize position control for a linear switched reluctance motor (LSRM) against its nonlinear characteristics. Principles for FLC and BPNN control are introduced elaborately. Simulation results via BPNN show that dynamic position errors for the LSRM can be limited to 0.1 mm. Experimental results on FLC suggest that point-to-point position tracking for the motor can achieve 0.01 mm, constraining dynamic position error in 0.1 mm. By experiments, FLC for the LSRM performs better than traditional proportional-integral-derivative (PID) control, proving the effectiveness of the alleviation of the nonlinearity for the LSRM.

Keywords–Fuzzy logic control, backward propagation neural network, LSRM, position tracking.

I. INTRODUCTION

Intelligent control algorithms are introduced to improve the performance of linear motors for precisely positioning trackers with high stability, because of uncertainties on the motors caused by their nonlinear characteristics and interferences from circumstance [1]. A liner switched reluctance motor (LSRM) as one of the linear motors is the control plant in this paper. The nonlinearity of the motor exists due to saturated magnetic cores, nonlinear thrust outputs and variety between static fraction coefficients and moving fraction coefficients [2]. Meanwhile, interference caused by varied loads, coupled flux lines and drifted parameters may significantly deteriorate the performance of the motor. Employing traditional PID controller is unable to fulfill the requirements of good performance under these uncertainties. Therefore, advance control approaches are increasingly investigated to address those problems mentioned. To solve the deteriorated performance stemmed from nonlinearity and altered operation conditions of the motor, fuzzy logic control (FLC), neural network control (NNC), adaptive control and sliding mode control (SMC) are used to control the position of the motor [3-6]. These control methods can imitate expert's experience, compute and exert adjusted control commands for the motor to get a satisfactory performance. After using these advanced controllers, not only will the overall hardware cost on the motion system reduce, but also the motor performance becomes better. Among them, FLC can transform linguistic control rules from experts' knowledge to regulate behaviors of mechanical devices, regardless of their accurate mathematic models [3]. Consequently, the controller would become simple and easy to design and implement. For industry control, most processes are suited to use linguistic rules, especially for those complex systems that their dynamic models are difficult to obtain or precisely predict. After imitating what the experts' programed, the controller would become intelligent. BP NNC is a popular intelligent control method. This method is capable of addressing nonlinear issues for the whole system because of its multiple layer mapping transformations with nonlinear characteristics [4]. Also, self-teaching and self-adaptation can optimize the weights of nodes, thus improving the performance of the motor and enhancing the stability of the system [5]. Except for these merits, BP NNC has high fault-tolerant capability so that the whole system would become more robust. Adaptive control is an effective approach to avoid the influences of altered environment or parameters. The designed zeros or poles for the motor will be regulated after identifying the parameters of the controlled subject. For the motor, in this paper, the position of the motor can be controlled and regulated after identifying the dynamic parameters of the operated motor [5]. Accordingly, the controlled position will be modified with more accuracy even though the parameters of the motor vary with the time. Sliding mode control (SMC) is another advanced control to address the nonlinear behavior of the control plant [6]. The core to design the SMC is to find a suitable sliding mode surface for the motion system.

In this paper, fuzzy logic control and BP neural network control approached mentioned are designed for the motor. Simulation and experimental results are obtained. After comparing the controlled results, advantages from each advanced controller are yielded, with elaborated analysis for the obtained results.

II. INTRODUCTION FOR THE LSRM

A. Configuration of LSRM

The basic mechanical structure of the LSRM is shown in Fig.1. This motor consists of three phase windings and each winding possesses a coil wrapped around a steel stacked core. Three cores fixed on an aluminum plate constitute the mover of the motor. Their section views are shown in Fig.1. The mover connected with four slippery wheels that belong to the components of two linear guides. The linear guides as well as the stator that is also comprised of stacked steel plates are fixed on a basement, which is the stator base. A 1-μm-resolution linear optical encoder is integrated into the LSRM system to observe the motion profile of the moving platform and provide the feedback position information. The main electrical and mechanical parameters of the LSRM are listed in Table I.

Power Electronics Research Center, Department of Electrical Engineering,
The Hong Kong Polytechnic University, Hong Kong.
Email: yy.zou@connect.polyu.hk eeecheng@polyu.edu.hk norbert.cheung@polyu.edu.hk

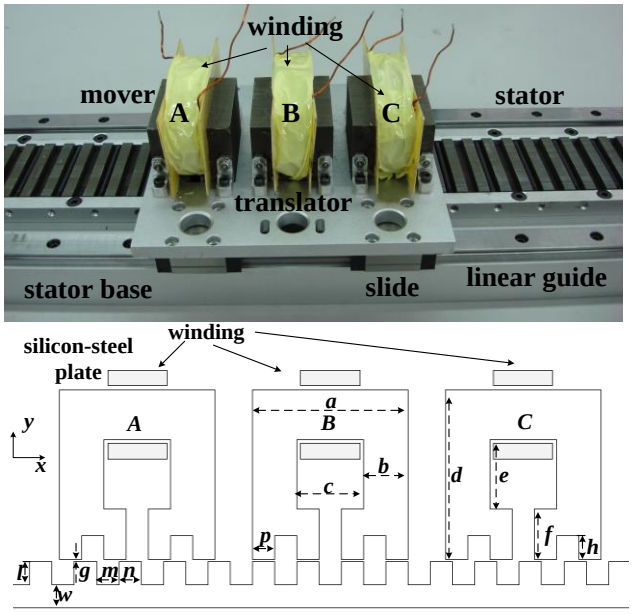


Fig. 1: The sections views of the motor.

Table 1: Specifications of the LSRM

Parameters	value
Rated current	4A
Mass of moving platform	1.5kg
Mass of stator	2kg
Pole width	6mm
Pole pitch	12mm
Phase division	10mm
Phase resistance	2ohm
Air gap length	0.3mm
Number of turns	160
Stack length	25mm
Encoder resolution	1μm
Rated current	4A
Mass of moving platform	1.5kg

B. Dynamic model of LSRM

The electrical terminal for any one phase can be characterized as the voltage balancing equation as follows:

$$u = R \times i + \frac{d l(i,x)}{dt} \tag{1}$$

where R, u and i represent phase resistance, terminal voltage and current, respectively. x is displacement and $l(i,x)$ denotes flux-linkage. From the mechanical side

$$F = M\frac{d^2x}{dt^2} + B\frac{dx}{dt} + f_L = \frac{1}{2} \times i^2 \times \frac{\partial L(i,x)}{\partial x} \tag{2}$$

where F is electromagnetic force, f_L, M and B are load force, mass of moving platform and friction coefficient, respectively.

C. Nonlinearity of the LSRM

As the saturation phenomenon of the magnetic steel materials exists and nonlinearity for the motor will occur. All motors cannot avoid the nonlinearity. The LSRM, as one of the motors, has a highly nonlinear characteristic. According to equation (2), if the input command is a force reference, this force command can be transformed as a current reference for drivers of the motor. Also, to improve the performance of the motor, force distribution function (FDF) is an effective way to alleviate the nonlinear force outputs excited by phase currents [5]. The overall FDF scheme is shown in Fig.2. Force ripples and acoustic noise generated by the motor can also be mitigated via a reasonable FDF that is designed to match the electromagnetic behavior of the motor. Apart from this approach, the advanced control method introduced as the following can also address the nonlinear problem. However, advanced control algorithms are usually employed for the motor to get a better performance against the nonlinearity.

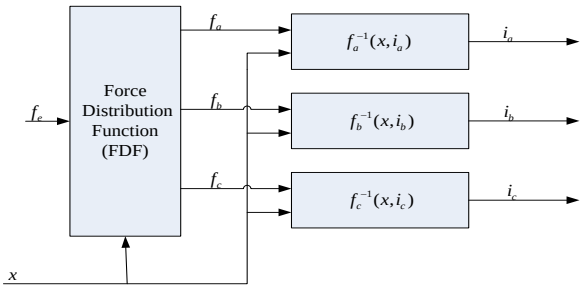


Fig. 2: The structure of the FDF.

III. Advanced Control For The LSRM

A. Configuration of control

This motor is suitable for a fuzzy PD controller. This controller can adjust control parameters according to the status of the motor via fuzzy logic approach. The fuzzy logic mainly consists of fuzzification, rule bases, fuzzy reasoning and defuzzifier. K_p and K_d are the proportional gain and the differential gain of the PD controller, respectively. The control diagram is shown in Fig. 3.

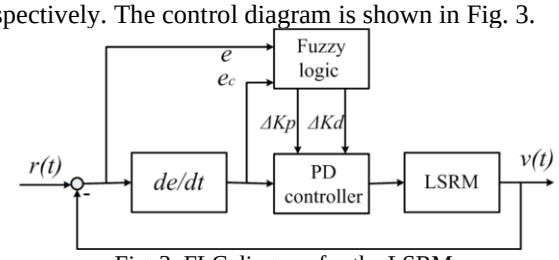


Fig. 3: FLC diagram for the LSRM.

The proposed fuzzy PD controller takes the position error and its change as the inputs and outputs the values of K_p and K_d. The two coefficients can be formulated as

$$K_d = K_{d0} + \Delta K_d \tag{3}$$

$$K_p = K_{p0} + \Delta K_p \tag{4}$$

where K_{p0} and K_{d0} are the initial values ,ΔK_p and ΔK_d are the increases of K_p and K_d.

By using the feedback signal from a position sensor, position error *E* and its change rate *Ec* can be obtained. The fuzzy logic controller will employ the two values. In this paper, the regions of the fuzzy sets for the two parameters are {-6 -4 -2 0 2 4 6}. After fuzzification, the fuzzy set values will be {NB NM NS ZO PS PM PB} and the corresponding linguistic variable fuzzy set is [negative large, negative middle, negative small zero, positive small, positive middle, positive big], as shown in Fig.4. Similarly, the discussion region for K_d and K_p are given in Fig.5. The rule bases for K_p and K_d are listed in Table II and III.

In this paper, we choose the triangle membership function for *E*, *Ec*, K_p and K_d, Mamdani inference method for fuzzy reasoning, and the center of gravity method for defuzzification.

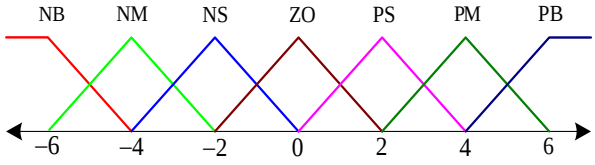


Fig.4: Membership functions of *E* and *Ec*.

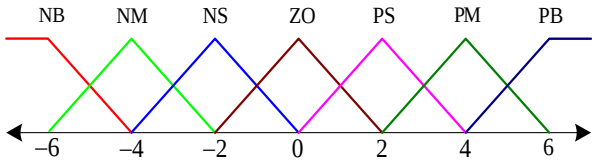


Fig.5: Membership functions of K_p and K_d.

The basic control rule for the fuzzy logic controller mainly includes:

(1) When |*E*| is large, to achieve a stronger dynamic response performance, we can set larger K_p and smaller K_d.

(2) When |*E*| is medium size, in order to obtain a smaller overshoot of the system response, K_p can be set smaller. At this time, K_d can be set larger.

(3) When |*E*| is small, to make the system stable, the value of K_p should be set smaller. To avoid the system oscillation near the set value, the value of K_d should refer to the value of |*Ec*|. When the value of |*Ec*| is small, the value of K_d can be larger. In contrast, when the value of |*Ec*| is larger, the value of K_d can be small.

Table II: Rule base of the K_p

E \ EC	NB	NM	NS	ZO	PS	PM	PB
NB	PB	PB	PM	PM	PS	ZO	ZO
NM	PB	PB	PM	PS	PS	ZO	NS
NS	PM	PM	PM	PS	ZO	NS	NS
ZO	PM	PM	PS	ZO	NS	NM	NM
PS	PS	PS	ZO	NS	NS	NM	NM
PM	PS	ZO	NS	NM	NM	NM	NB
PB	ZO	ZO	NM	NM	NM	NB	NB

B. Artificial neural network

Neuron here is the basic processing unit for the artificial neural network. Generally, it consists of multi inputs and a single output, processing nonlinear signals. A neuron includes a connector with a weight value, a sum unit and

Table III: Rule base of the K_d

E \ EC	NB	NM	NS	ZO	PS	PM	PB
NB	PS	NS	NB	NB	NB	NM	PS
NM	PS	NS	NB	NM	NM	NS	ZO
NS	ZO	NS	NM	NM	NS	NS	ZO
ZO	ZO	NS	NS	NS	NS	NS	ZO
PS	ZO	ZO	ZO	ZO	ZO	ZO	ZO
PM	PB	NS	PS	PS	PS	PS	PB
PB	PB	PM	PM	PM	PS	PS	PB

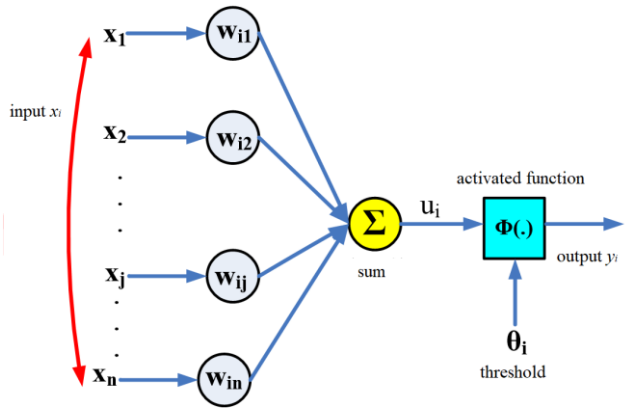


Fig.6: Model of an artificial neuron.

an activated function. A neuron with a series of inputs x_1, $x_2, \ldots, x_n$ of connectors with weights $w_1, w_2, \ldots, w_n$ is shown in Fig.6. y_i is the output of the neuron i. After summing, a linear integrity output u_i is yielded through these y_i. $y_i = j(x)$ is the activated function with a threshold θ_i to constrain the output scope of the designed neuron. The mathematic model for the neuron can be expressed as

$$u_i = \sum_{j=1}^{n} w_{ij} x_j \ , \qquad (5)$$

$$net_i = u_i - \theta_i \ , \qquad (6)$$

$$y_i = \varphi(net_i) . \qquad (7)$$

where net_i denotes a computing method to combine these inputs.

a) Backward Propagation neural network

Back propagation (BP) neural network is one kind of neural network and it can be also named as a backward propagation network or a multi-level forward network.

BP algorithm is an optimal method based on gradient downward approach, adjusting the weight to minimize the overall error of the network. It comprises forward network computation and error inverse propagation. These weights for neurons will be optimized after backward transferring errors to minimize the overall error of the controlled system. BP network usually involves an input layer, a hiding layer and an output layer. A BP network is shown in Fig. 7, with the numbers of the inputs, the outputs and the implicit node *m*, *n* and *q*, respectively.

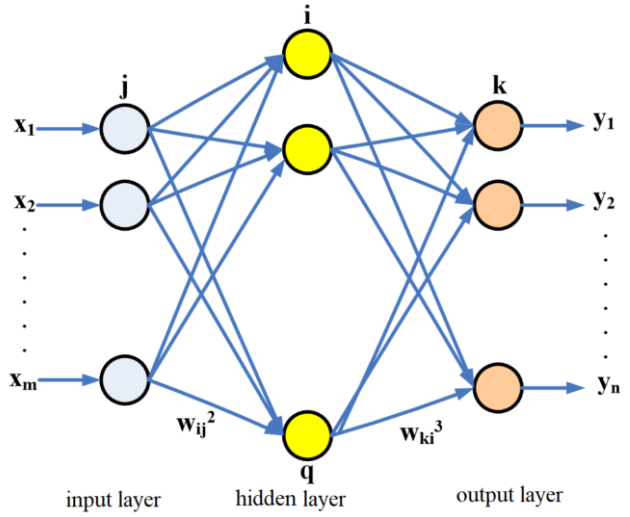


Fig.7: The structure of the BP neural network.

b) BP forward network design

The outputs of the input layer of the BP network can be expressed as

$$O_j^1 = x_j, (j = 1, 2, ..., m) \tag{8}$$

The inputs and outputs of the implicit layer are

$$net_i^{\ 2} = \sum_{j=1}^{n} w_{ij}^{\ 2} O_j^1 - \theta_i, (i = 1, 2, ..., q) \tag{9}$$

$$O_j^2 = \varphi(net_i^{\ 2}) \tag{10}$$

Inputs and outputs of the output layer can be formulated as

$$net_k^{\ 3} = \sum_{i=1}^{q} w_{ki}^{\ 3} O_i^1 - \theta_k, (k = 1, 2, ..., n) \tag{11}$$

$$O_j^3 = \varphi(net_i^{\ 3}) \tag{12}$$

where O_j^1, O_i^2, O_k^3 are outputs of the input layer，and $net_i^{\ 2}$, $net_i^{\ 3}$ are inputs of the complicit layer and the output layer. $w_{ij}^{\ 2}$ and $w_{ij}^{\ 3}$ are connecting weights between the input layer, the implicit layer and the output layer. θ_i, θ_k are thresholds of the implicit layer and the output layer. $y_i = j(x)$ is the activated function expressed by

$$\varphi(x) = \frac{1}{1 + e^{-x}} \tag{13}$$

$$\varphi(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \tag{14}$$

c) Regulation of PB neural network via error

The output error of the system is

$$J = \frac{1}{2} \sum_{k=1}^{n} (r_k - y_k)^2 \tag{15}$$

where r is the reference of the system and y is the output.

d) Weight regulation of output layer

Weights, as weighting factors of the output layer, will adjust according to the negative value of the gradient to function J

$$\Delta w_{ki}^3 = -\eta \frac{\partial J}{\partial w_{ki}^3} = -\eta \frac{\partial J}{\partial net_{ki}^3} \cdot \frac{\partial net_{ki}^3}{\partial w_{ki}^3} \tag{16}$$

where η is a learning rate, $\eta > 0$.

And

$$\frac{\partial net_{ki}^3}{\partial w_{ki}^3} = O_i^2 \tag{17}$$

$$\frac{\partial J}{\partial net_k^3} = \frac{\partial J}{\partial O_k^3} \cdot \frac{\partial O_k^3}{\partial net_k^3} = -(r_k - y_k) \cdot \varphi'(net_i^{\ 3}) \tag{18}$$

assuming that

$$\delta_k = (r_k - y_k) \cdot \varphi'(net_i^{\ 3}) \tag{19}$$

Then

$$\Delta w_{ki}^3 = \eta (r_k - y_k) \cdot \varphi'(net_i^{\ 3}) \cdot O_i^2 = \eta \delta_k O_i^2 \tag{20}$$

it can be rewritten as

$$w_{ki}^3(k+1) = w_{ki}^3(k) + \Delta w_{ki}^3 = w_{ki}^3(k) + \eta \delta_k O_i^2 \tag{21}$$

to improve the convergent rate of the computing results, the weights can be modified as

$$\begin{aligned} w_{ki}^3(k+1) &= w_{ki}^3(k) + \eta \delta_k O_i^2 \\ &+ \alpha (w_{ki}^3(k) - w_{ki}^3(k-1)) \end{aligned} \tag{22}$$

where α is an inertia coefficient and $0 < \alpha < 1$.

e) Weight regulation of implicit layer

According to the gradient of J, the weights for the implicit layer can be calculated as

$$\Delta w_{ki}^2 = -\eta \frac{\partial J}{\partial w_{ki}^2} = -\eta \frac{\partial J}{\partial net_{ki}^2} \cdot \frac{\partial net_{ki}^2}{\partial w_{ki}^2} \tag{23}$$

where η is a learning rate, $\eta > 0$.

And

$$\frac{\partial net_{ki}^2}{\partial w_{ki}^2} = O_i^1 \tag{24}$$

$$\frac{\partial J}{\partial net_k^2}=\frac{\partial J}{\partial O_k^2}\cdot\frac{\partial O_k^2}{\partial net_k^2}=-(r_k-y_k)\cdot\varphi'(net_i^{\,2}) \tag{25}$$

the output of the implicit layer will put an impact on all units connected with the unit from the layer

$$\begin{aligned}\frac{\partial J}{\partial O_k^2}&=\sum_{k=1}^{n}\frac{\partial J}{\partial net_k^3}\cdot\frac{\partial net_k^3}{\partial O_k^2}\\&=\sum_{k=1}^{n}\frac{\partial J}{\partial net_k^3}\cdot w_{ki}^3=-\sum_{k=1}^{n}\delta_k\end{aligned} \tag{26}$$

given that

$$\delta_i=-\sum_{k=1}^{n}(\delta_k\cdot w_{ki}^3)\cdot\varphi'(net_i^{\,3}) \tag{27}$$

then

$$\Delta w_{ki}^2=\eta\sum_{k=1}^{n}(\delta_k\cdot w_{ki}^3)\cdot\varphi'(net_i^{\,2})\cdot O_i^1=\eta\delta_k O_i^1 \tag{28}$$

it can be rewritten as

$$w_{ki}^2(k+1)=w_{ki}^2(k)+\Delta w_{ki}^2=w_{ki}^2(k)+\eta\delta_k O_i^1 \tag{29}$$

likewise, to improve the convergent rate of the computing results, the weights can be modified as

$$\begin{aligned}w_{ki}^2(k+1)&=w_{ki}^2(k)+\eta\delta_k O_i^1\\&+\alpha(w_{ki}^2(k)-w_{ki}^2(k-1))\end{aligned} \tag{30}$$

where α an inertia coefficient and $0<\alpha<1$.

f) BP neural network PID control

BP network is capable of self-learning and nonlinear adaption. BP network PID control is to seek the optimal control parameters for the PID controller by self-learning of the BP network. The whole control block can be plotted as in Fig. 3. *r* is the reference and y is the output. *e* is the error of the system. u is the output of the system controller. *Kp*, *Ki* and *Kd* are the control parameters for the PID controller, obtained from the BP network. These parameters for the PID controller and the BP network are the two key parts for the whole control of the system. According to the status of the system, BP network can enter a self-learning to adjust the parameters for the PID controller so that the adaptive and nonlinear control will be realized.

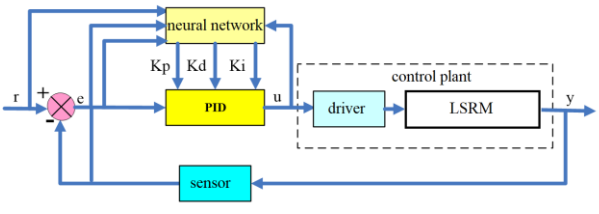
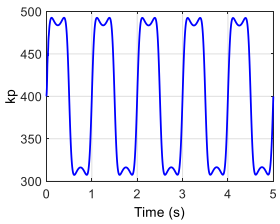


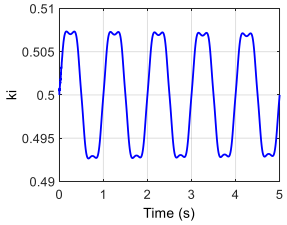
Fig.8: PID control scheme with neural network for the LSRM.

IV. Verification of The Advanced Control

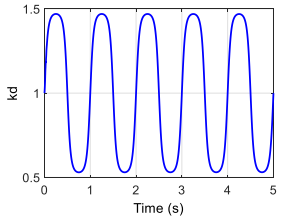
To verify the effectiveness of the mentioned advanced control algorithms, simulation software MATLAB/Simulink is introduced to build the controllers including fuzzy logic control and BP neural network control. After building the simulation block via Simulink, simulation results are obtained for neural network PID controller, as shown in Fig. 9. Fig. 9 (a), (b) and (c) show the values of the coefficients K_p, K_i and K_d, respectively. Fig.9 (d) and (e) illuminate the position tracking and its errors. Position tracking errors corresponding to a sinusoidal reference can be limited less than 0.1 mm under the steady status of the motor. Meanwhile, by programing the Simulink control blocks of fuzzy logic control into dSPACE card, a hardware controller for the fuzzy logic control of the linear motor is established. Experimental results are shown in Fig. 10. Position tracking errors are constrained in 0.1 mm for sinusoidal reference and 0.01 mm for square wave reference, respectively. Both sinusoidal and square wave references validate the effectiveness of the designed fuzzy logic controller.



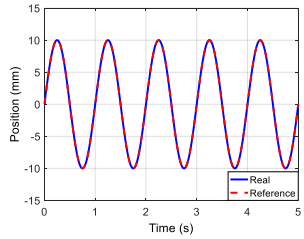
(a) Values of K_p for BPNN



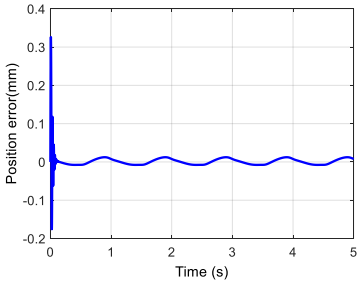
(b) Values of K_i for BPNN



(c) Values of K_d for BPNN



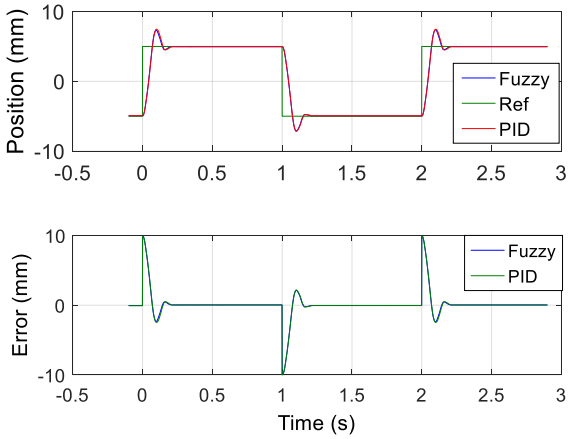
(d) Position response controller



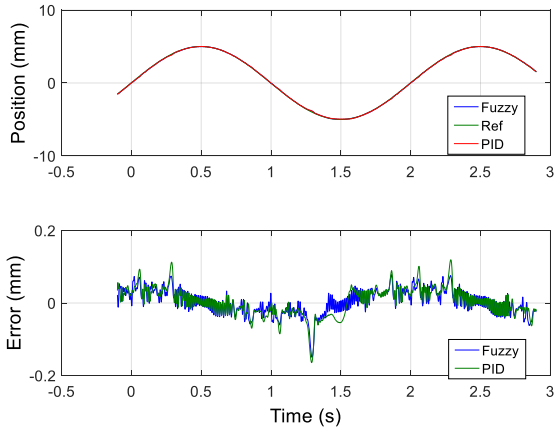
(e) position error of the BPNNC control system

Fig. 9: Coefficients (a), (b), (c) and (d) position response and errors (e) for the designed BPNN control.

The comparison of PID controller and fuzzy logic PID controller are shown in Fig.10. From the experimental results especially under a sinusoidal reference, position errors from the fuzzy logic controller are less than that from a PID controller, which obviously shows that advanced control can improve the performance of the LSRM, particular under a varied reference. Although the performance of the LSRM under the reference could be worse as different operation status with nonlinear factors, advanced control motion system outweigh traditional equipment and can perform better in industry applications.



(a) Position response of square wave under fuzzy logic controller.



(b) Position response of sinusoidal wave under fuzzy logic controller.

Fig. 10: Position response for sinusoidal and the square wave reference (a) and the sinusoidal wave, under PID and fuzzy logic PD controllers.

IV. Conclusion

Fuzzy logic PD control and BP neural network PID control are advanced control algorithms to better performance of complex equipment and nonlinear machines. They inherit the advantages of PID controller, regulating behaviors of control subjects without accurate mathematic modes. Apart from this merit, these advanced control systems can adjust their control parameter according to the variations of the status of the entire control system, overcoming shortcomings of traditional PID control that fails to conquer the influences from varied references and interferences of the environment. Advanced control systems could dominate industrial equipment for intelligence factories and smart mobility.

References

[1] Wen Yu, *Recent Advances in Intelligent Control Systems*, Springer Dordrecht Heidelberg London New York, 2009.

[2] J. F. Pan, N.C. Cheung, Yu. Zou, “An Improved Force Distribution Function for Linear Switched Reluctance Motor on Force Ripple Minimization with Nonlinear Inductance Modeling", IEEE Trans. Magn., Vol. 48, no.11, 2012, pp. 3064-3067.

[3] G. El-Saady, El-Nobi A. Ibrahim, M. Abuelhamd, “Hybrid PD-Fuzzy controller for high performance linear switched reluctance motor under different operating conditions”, 2016 Eighteenth International Middle East Power Systems Conference (MEPCON), 2016, pp. 437-444.

[4] R. Zhong, Y. B. Wang, Y. Z. Xu, “Position sensorless control of switched reluctance motors based on improved neural network,” IET Electric Power Applications, Vol.6, no.2, 2012, pp.111-121.

[5] Shi Wei Zhao, Norbert C. Cheung, Wai-Chuen Gan, Jin Ming Yang, Jian Fei Pan,"A Self-Tuning Regulator for the High-Precision Position Control of a Linear Switched Reluctance Motor” , IEEE Ind. Electron., vol. 54,no. 5, Oct. 2007.

[6] Jianfei Pan, Siu Wing Or, Yu Zou, Norbert C. Cheung, “Sliding-mode position control of medium-stroke voice coil motor based on system identification observer,” IET Electric Power Applications, Vol.9, no.9, 2015, pp.620-627.

Acknowledgment

The author gratefully acknowledge of the financial support of the Research Office, The Hong Kong Polytechnic University under the Project G-YBLH.