Journal of Computers Vol. 31 No. 2, 2020, pp. 1-11
doi:10.3966/199115992020043102001

MRAS Speed Observer for Permanent Magnet Linear Synchronous Motor Based on RBF Neural Networks

Qi-Yong Chen[1], Rong-Kun Wang[1*], Fei Man[1], Kai Peng[2], Bing-Tao Hu[1], Dong Yan[1]

[1] College of Information Science and Engineering, Huaqiao University, Jimei 668, Xiamen, China
wangrongkun@hqu.edu.cn

[2] College of Engineering, Huaqiao University, Huabei 269, Quanzhou, China
pkbupt@gmail.com

Received 13 January 2019; Revised 20 April 2019; Accepted 20 April 2019

Abstract. In order to realize speed sensorless control of permanent magnet synchronous linear motors (PMLSMs) in the whole speed range, a model reference adaptive system (MRAS) speed observer based on radial basis function neural network (RBFNN) is proposed to observe the speed information of PMLSMs in this paper. The speed observer uses a RBFNN as its adaptive mechanism and achieves the speed accurate estimation by taking advantage of strong nonlinear approximation ability of the RBFNN. In addition, a RBFNN identifier is designed to provide gradient information for the RBFNN in the speed observer. Simulation and experiment indicate that the speed observer can accurately observe the speed in the whole speed range for PMLSMs and has good dynamic response characteristics.

Keywords: model reference adaptive system, permanent magnet linear synchronous motor, radial basis function neural network, speed sensorless control

1 Introduction

With the advantages of high power density, precision and efficiency, PMLSMs are widely used in industrial, aerospace, military and other fields. For the PMLSM control system, the installation of the mechanical sensor increases the system cost and volume, which impedes the high precision development of the PMLSM. Therefore, speed sensorless control technology has become a research hotspot in recent years [1-4].

Compared to rotary motors, the linear motors have the characteristics of frequent changing directions and larger speed range. Thus, the study of the speed sensorless control for linear motors is relatively complicated. It is hard for using single traditional methods to reach accurate estimation of the speed in the whole speed range. A composite combination algorithm that adopted the high-frequency voltage signal injection method at the low speed and the state augmented extended Kalman filtering method at high speed can realize the position sensorless control of the motor in a wide speed range, while this algorithm is relatively complex and has a large amount of computation [5]. Ref [6] combines the advantages of the sliding mode control and neural network, and proposes a sliding mode controller based on the neural network. This sliding mode controller can effectively reduce the chattering phenomenon, but at low speed, the speed estimation ability is not ideal. Expanded state observer (ESO) can also be used to identify the speed. It can effectively improve the anti-interference ability, while the adjustment time of the system is too long and can not realize the real-time estimation of the speed [7].

The Radial Basis Function (RBF) neural network has strong adaptive and nonlinear approximation abilities and can approximate any nonlinear function with arbitrary precision, which is more suitable for solving the control problems of nonlinear and uncertain systems. Therefore, this paper proposes a MRAS speed observer based on RBF neural networks to realize the speed estimation of PMLSM in whole speed

* Corresponding Author

range.

2 Speed Sensorless Control System of PMLSM

The Akribis's PMLSM named AUM2-S1 is used for research in this paper. The parameters of this motor are shown in Table 1. The stator of the motor installed an incremental encoder with model Mercury II 1600.The encoder does not participate in motor control, and the motor speed measured by the encoder is only used for comparative test with the estimated speed. The experimental platform is shown in Fig. 1. The experimental platform includes the self-designed driver with TMS320F28335 as the core controller. It can be used to verify the speed sensorless control method proposed in this paper.

Table 1. AUM2-S1 linear motor parameters

Parameters	Value
Rated current I_R (A)	2
Peak current I_{pk} (A)	8
Continuous power P_R (w)	13
Stator phase resistance R_s (Ω)	1.625
d、q axis inductance $L_{d,q}$ (mH)	0.375
Back EMF constant K_e (V/m/s)	6.4
Motor pole distance τ_p (mm)	15
Continuous force F_N (N)	11
Molar mass m_r (kg)	1.35

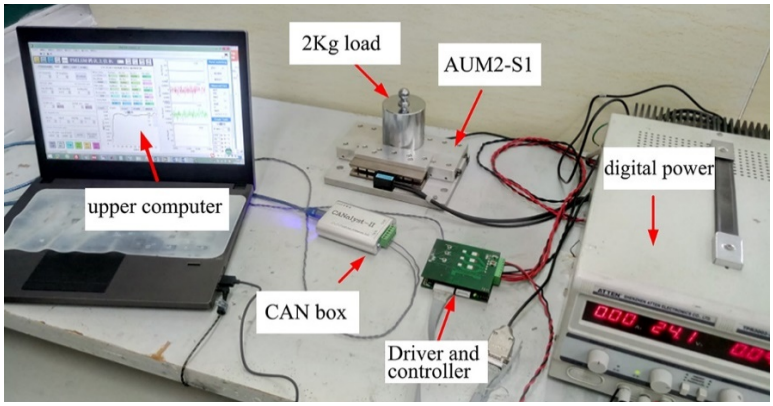


Fig. 1. Experimental platform

2.1 Mathematical Model of PMLSM

The MRAS system needs to select the mathematical model of the motor as the adjustable model. The mathematical model of PMLSM in synchronous rotating coordinate system is as follows:

$$\begin{cases} u_d = R_s i_d + L_d \dfrac{\mathrm{d}i_d}{\mathrm{d}t} - \omega_e L_q i_q \ . \\ u_q = R_s i_q + L_q \dfrac{\mathrm{d}i_q}{\mathrm{d}t} + \omega_e \psi_m + \omega_e L_d i_d \ . \end{cases} \tag{1}$$

Where u_d , u_q , i_d , i_q , L_d , L_q are the voltage, current, inductance of the d/q winding of the stator respectively. ψ_m is the flux linkage produced by the permanent magnets. ω_e is synchronous rotational electrical angular velocity, and R_s is the phase resistance of the stator winding.

2.2 Speed Sensorless Control Strategy of PMLSM

As shown in Fig. 2, $i_d = 0$ vector control mode and speed-current loop dual loop control strategy is applied in the PMLSM speed sensorless control system. The MRAS based on RBFNN (RBFNN MRAS) speed observer proposed in this paper replaces the mechanical sensor and provides speed and phase information for the motor system. The work principle of the speed sensorless control system is: the speed observer obtains the angular velocity estimation $\hat{\omega}_e$ and phase estimation $\hat{\theta}_e$ by collecting and processing the signals of q-axis current i_d and d/q-axis voltage u_d and u_q. The angular velocity $\hat{\omega}_e$ is linearly transformed to obtain the linear velocity $\hat{V}$, which is fed back to the velocity loop as a velocity feedback value, and the phase estimation $\hat{\theta}_e$ is used for the coordinate transformation of the system, thereby realizing the speed sensorless control of the PMLSM.

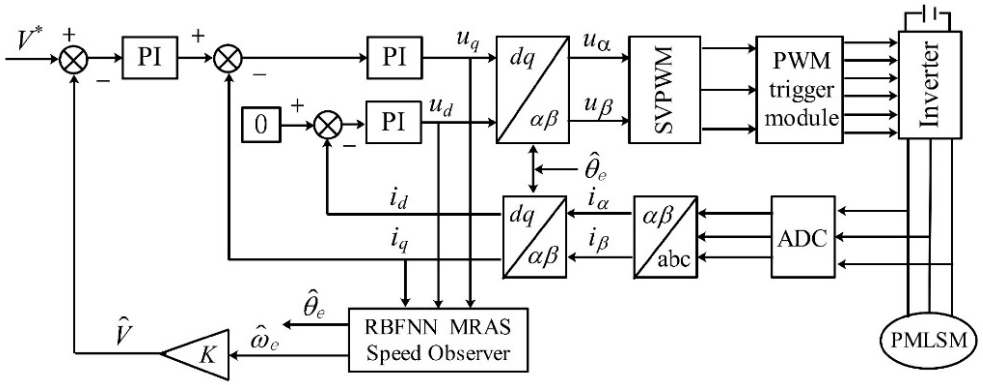


Fig. 2. Speed sensorless control system diagram of the PMLSM

3 The Design of RBFNN MRAS Speed Observer

3.1 RBF Neural Network

RBF neural network is a feedforward local approximation neural network and the structure is shown in Fig. 3. The structure of RBF neural network consists of three layers. The first layer is the input layer, which plays the role of direct signal transmission; the second one is the hidden layer, in which the activation function of the neuron is a radial basis function (usually using a Gaussian radial basis function); the third one is the output layer, whose neuron activation function is a linear function, and its output is the result of linearly weighted summation of the output of each unit node of the hidden layer.

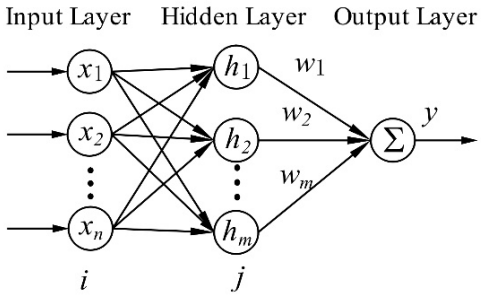


Fig. 3. The structure of RBF neural network

The RBF neural network maps the low-dimensional input signals of the input layer into the high-dimensional space through the hidden layer through a simple three-layer structure, which makes the indivisibility problem of low-dimensional linear is linearly separable in the high-dimensional space, and at the same time, the learning of weights can also adopt a faster linear optimization strategy. Therefore, the RBF neural network has the characteristics of simple structure, fast learning convergence, and ability

to approximate any nonlinear function. This is the reason why RBF neural networks are increasingly used in adaptive control [8-12].

Suppose the number of neurons in the input layer is *n*, the number of neurons in the hidden layer is *m*, and the number of output layers is 1, $x=[x_1,\ldots,x_n]^T$, which is the input vector of the network. When the radial basis function in the RBF neural network adopts a Gaussian function, the output of jth hidden layer nodes is:

$$h_j=\exp\left(-\frac{\left\|x-c_j\right\|^2}{2b_j^2}\right)\left(1\le j\le m\right). \tag{2}$$

In formula (2), $\|x\|$ is the Euclidean distance; b_j is the width of the jth Gaussian basis function of the network; c_j is the central position vector of the jth hidden layer node of the network, and $c_j=[c_1,\ldots,c_{nj}]$; $w=[w_1,\ldots,w_m]^T$ is the weight vector between the hidden layer and the output layer. At this time the output of RBF network is:

$$y=h_1w_1+\cdots+h_jw_j+\cdots+h_mw_m. \tag{3}$$

3.2 RBFNN MRAS Speed Observer

The block diagram of RBFNN MRAS speed observer for PMLSM is shown in Fig. 4. The reference model is the PMLSM control system, and the plant model is the mathematical model of the PMLSM, as shown in formula (1). This RBFNN MRAS speed observer includes a RBF neural network controller (RBFNNC), which is used as the adaptive mechanism of the MRAS system. The input signals are the q-axis estimated current $\hat{i}_q$ from the plant model and the q-axis actual current i_q from the reference model and the difference e between $\hat{i}_q$ and i_q. The output is the angular velocity estimation $\hat{\omega}_e$. The RBFNNC utilizes the nonlinear function approximation ability of the RBF neural network to compensate the non-modeling factor of the system. The learning process of the RBFNNC is the process of speed estimation by RBFNN MRAS speed observer. In addition, a RBF Neural Network Identifier (RBFNNI) for parameter identification is designed in this speed observer to provide gradient information for the RBFNNC. The input of the RBFNNI is angular velocity estimation $\hat{\omega}_e$ and the q-axis estimated current $\hat{i}_q$. e^* is the difference between $\hat{i}_q$ and the output of the RBFNNI $\hat{i}_q$. The correction rule of the RBFNNI is to make the error e^* as small as possible (the theoretical value is zero).

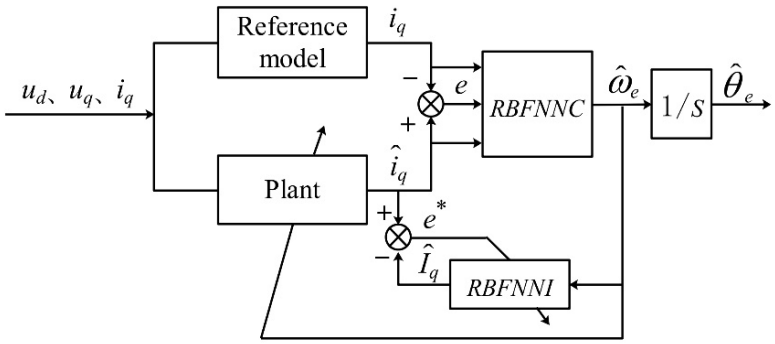


Fig. 4. The structure of RBFNN MRAS speed observer

3.3 Off-Line Training

The center and width of the radial basis function, as well as the connection weights, are three important parameters of the RBF neural network. The setting of these parameters is completed by training the

neural network. The training of the neural network is mainly divided into two steps: off-line training and on-line training.

In this paper, self-organizing center selection method is used to select the center of the basis function in off-line training session. After analyzing the sample data extracted from the full speed range of PMLSM, selected the ones with representative distribution as the center. In this case, the width b of the Gaussian function can be set to:

$$b = \frac{d_m}{\sqrt{2m}} . \tag{4}$$

Where m is the number of hidden layer nodes; d_m is the maximum distance between the selected centers. This method for selecting the width b can make the mapping range of the Gaussian function suitable.

Then, the weight w can be calculated by the following formula:

$$w = N^{-1}Y . \tag{5}$$

Where Y is the expected response, N^{-1} is the pseudo-inverse of matrix N, $N = \{h_j\}$, and h_j is the output of the Gaussian function.

While adjusting the three parameters above, it is also necessary to adjust the number of hidden layer nodes according to the training effect to build a simple and compact neural network structure. This is an effective way to avoid the neural network calculation occupying too much mechanical period on the serial computing DSP28335 controller. At the beginning of training, the initial number of hidden layers used in this paper is equal to the number of input signals of RBF neural network, that is, three. In the training process, add or delete the hidden layer nodes on the basis of the "novelty" according to the adaptive algorithm from Ref [13].

The specific algorithm is shown in Fig. 5:

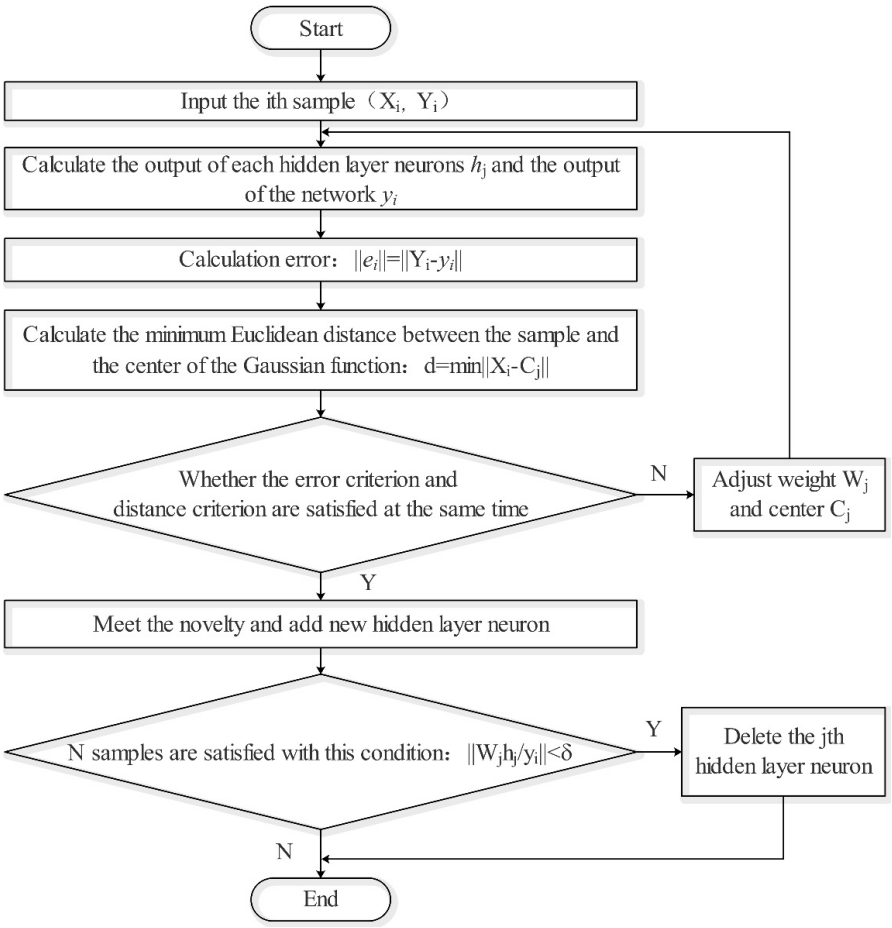


Fig. 5. Procedure for adjusting the number of hidden layer nodes in off-line training

After offline training, this paper designed the structure of RBFNNC and RBFNNI. As shown in Fig. 6, the structure of the RBFNNC is 3-9-1, the RBFNNI is a structure of 2-7-1.

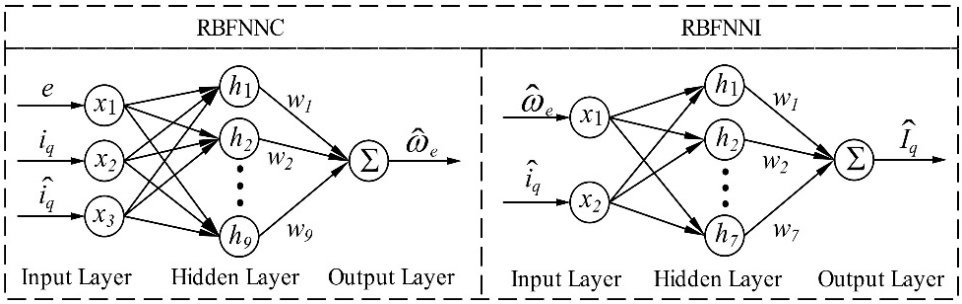


Fig. 6. The neural network structure of RBFNNC and RBFNNI

3.4 On-Line Training

The learning algorithm uses the gradient descent method, and corrects the center c_j and width b of the Gaussian function and the weight w that between the hidden layer and the output layer according to the negative gradient direction of the error function, and increases a momentum term at the same time to improve the network learning efficiency. The specific method is as follows:

For the RBFNNC, the tracking error is defined as:

$$e(k) = i_q(k) - \hat{i}_q(k). \tag{6}$$

The objective function of the Network parameter learning error is:

$$E(k) = \frac{1}{2} e(k)^2 \cdot \tag{7}$$

The modified parameters are obtained by gradient descent method:

$$\begin{cases} \Delta w_j(k) = -\eta \dfrac{\partial E(k)}{\partial \omega(k)} = \eta e(k) \dfrac{\partial i_q(k)}{\partial \hat{\omega}_e(k)} h_j \cdot \\ \Delta b_j(k) = -\eta \dfrac{\partial E(k)}{\partial b_j(k)} = \eta e(k) \dfrac{\partial i_q(k)}{\partial \hat{\omega}_e(k)} w_j h_j \dfrac{\left\| x - c_{ij} \right\|^2}{b_j^3} \cdot \\ \Delta c_{ij}(k) = -\eta \dfrac{\partial E(k)}{\partial c_{ij}(k)} = \eta e(k) \dfrac{\partial i_q(k)}{\partial \hat{\omega}_e(k)} w_j h_j \dfrac{x_i - c_{ij}}{b_j^2} \cdot \end{cases} \tag{8}$$

Then update the parameters:

$$\begin{cases} w_j(k) = w_j(k-1) + \Delta w_j(k) + \alpha \left(w_j(k-1) - w_j(k-2) \right) \cdot \\ b_j(k) = b_j(k-1) + \Delta b_j(k) + \alpha \left(b_j(k-1) - b_j(k-2) \right) \cdot \\ c_j(k) = c_j(k-1) + \Delta c_j(k) + \alpha \left(c_j(k-1) - c_j(k-2) \right) \cdot \end{cases} \tag{9}$$

Where η is the learning rate, $\eta \in [0,1]$; α is a momentum factor, $\alpha \in [0,1]$; and k represents the kth moment; In the formula (8), $\dfrac{\partial i_q(k)}{\partial \hat{\omega}_e(k)}$ is the Jacobian matrix, also called the gradient information. However, this value cannot be obtained directly, so this paper uses the RBFNNI to provide gradient information for the RBFNNC. For the RBFNNI, the tracking error is defined as:

$$e^*(k) = i_q - \hat{I}_q(k). \tag{10}$$

The objective function of the Network parameter learning error is:

$$E^*(k) = \frac{1}{2} e^*(k)^2 \cdot \tag{11}$$

In order to distinguish the network parameters of RBFNNC, the center, width and weight of RBFNNI are named as $c_j(k)$, $B_j(k)$ and $W_j(k)$ respectively. The output of the hidden layer neuron is H_j, and the modified equation for each parameter can be obtained according to the gradient descent method as follows:

$$\begin{cases} W_j(k) = W_j(k-1) + \Delta W_j(k) + \alpha\left(W_j(k-1) - W_j(k-2)\right). \\ B_j(k) = B_j(k-1) + \Delta B_j(k) + \alpha\left(B_j(k-1) - B_j(k-2)\right). \\ C_j(k) = C_j(k-1) + \Delta C_j(k) + \alpha\left(C_j(k-1) - C_j(k-2)\right). \end{cases} \tag{12}$$

The RBFNNI network recognizer output is:

$$\hat{I}_q(k) = H_1 W_1 + \cdots H_j W_j + \cdots + H_7 W_7. \tag{13}$$

Then, the gradient information of the RBFNNC can be calculated by the RBFNNI:

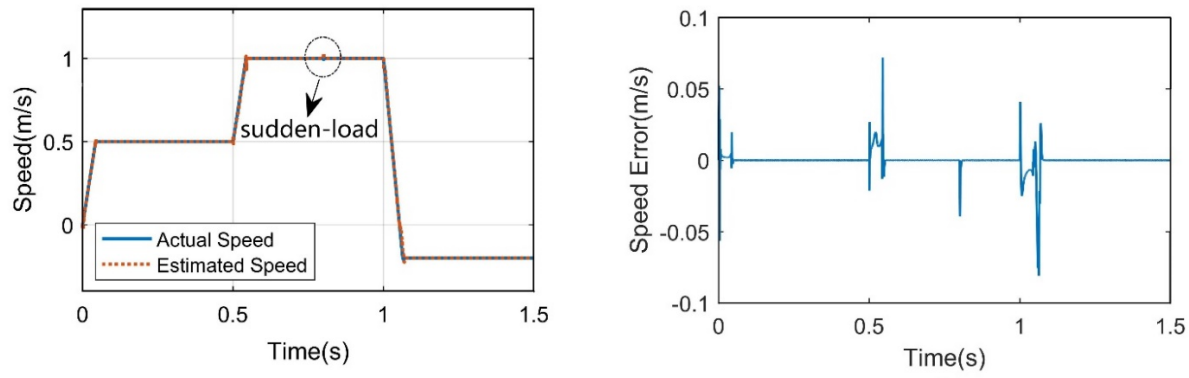
$$\frac{\partial i_q(k)}{\partial \hat{\omega}_e(k)} \approx \frac{\partial \hat{I}_q(k)}{\partial \hat{\omega}_e(k)} = \Sigma_{j=1}^{7} W_j H_j \frac{c_{ij} - \hat{\omega}_e(k)}{B_j^2}. \tag{14}$$

4 Simulation and Experiment

4.1 Simulation Results

In order to verify the effectiveness of the method, this paper uses MATLAB/Simulink simulation software to build simulation model. The motor parameters of the simulation model are shown in Table 1. In order to verify the control performance of the speed sensorless control method in various speed ranges, a simulation experiment under the condition of acceleration and deceleration, sudden load and round-trip movement at medium and high speeds is performed, and simulation experiments on sudden load and round-trip movement at low speed is performed.

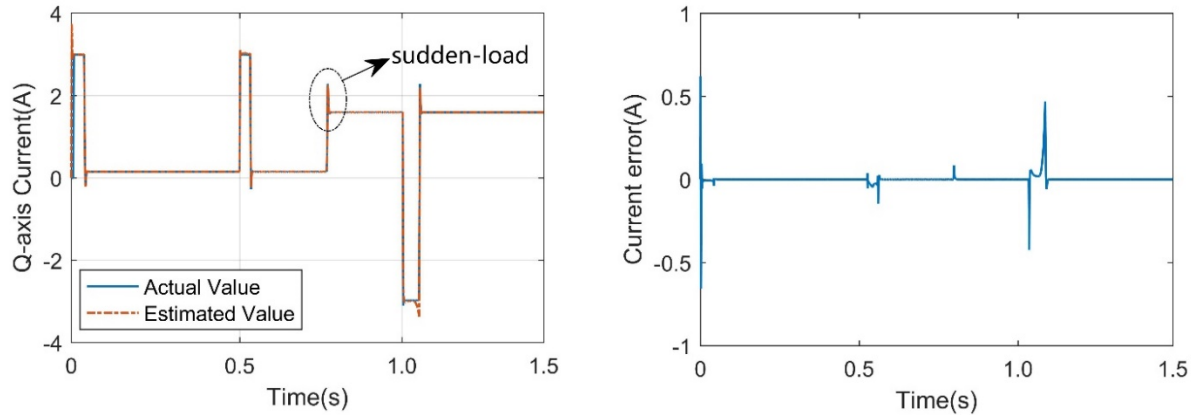
Fig. 7 to Fig. 9 show the simulation results of the linear motor under medium and high speeds. In the experiment, the motor starts at a speed of 0.5m/s at zero time, increases to 1m/s at 0.5s, then sudden load 10kg at 0.8s, and finally runs in reverse direction at a speed of 0.2m/s at 1s.The waveform of actual speed, estimated speed and speed error are shown in Fig. 7. The figure shows that the estimated speed can follow the actual speed very well throughout the all process, and the error is very small and tends to 0 when the motor is running in steady state, and the instantaneous error is slightly larger with the maximum value of nearly 7% in the speed change and commutation interval range. Adding a load at 0.8s, a slight disturbance is produced with a speed error of approximately 4%.



(a) (b)

Fig. 7. Two kinds of speeds and their error waveforms at medium and high speeds

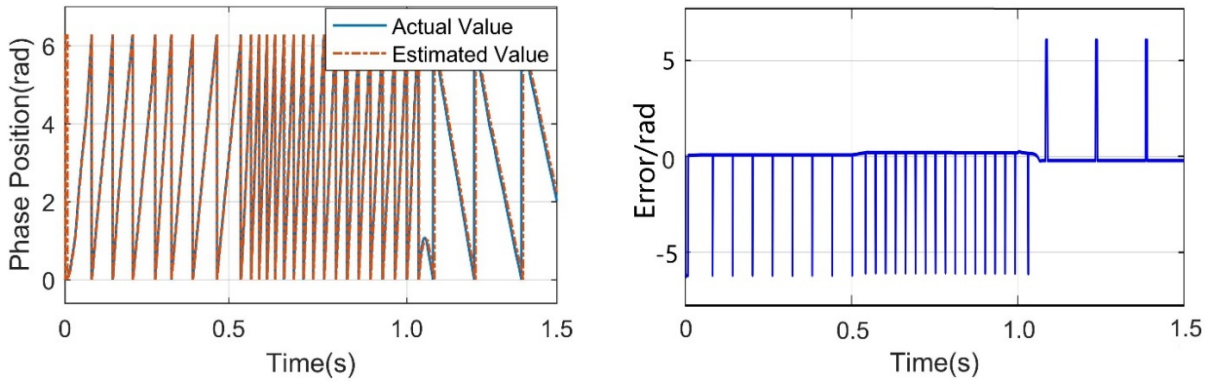
Fig. 8 shows the waveforms of the q-axis current actual value i_q and the estimated value $\hat{i}_q$ and their errors. From Fig. 8(a), it can be seen that the waveform of the estimated value $\hat{i}_q$ was basically identical with the actual value i_q, and the slightly larger error occurs only when sudden load and the speed is changed, and the maximum value of the error nearly 18%. The error waveform is shown in Fig. 8(b).



(a) (b)

Fig. 8. Two kinds of Q-axis currents and their error waveforms at medium and high speeds

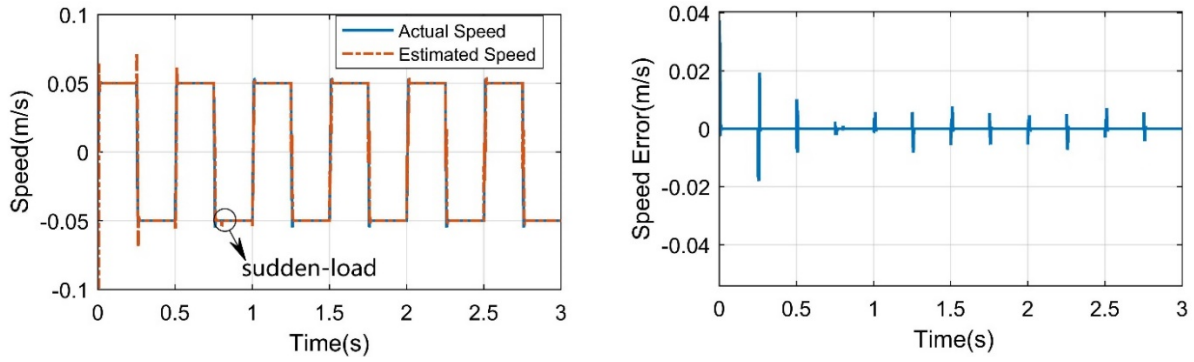
Fig. 9 shows the waveforms of the actual phase, estimated phase and their errors. In Fig. 9, there is a larger error between the phase estimation value and the actual phase value when the motor starts up. The error gradually decrease after start-up and the errors slightly increase when the motor is in reverse operation.



(a) (b)

Fig. 9. Two kinds of phase positions and their error waveforms at medium and high speeds

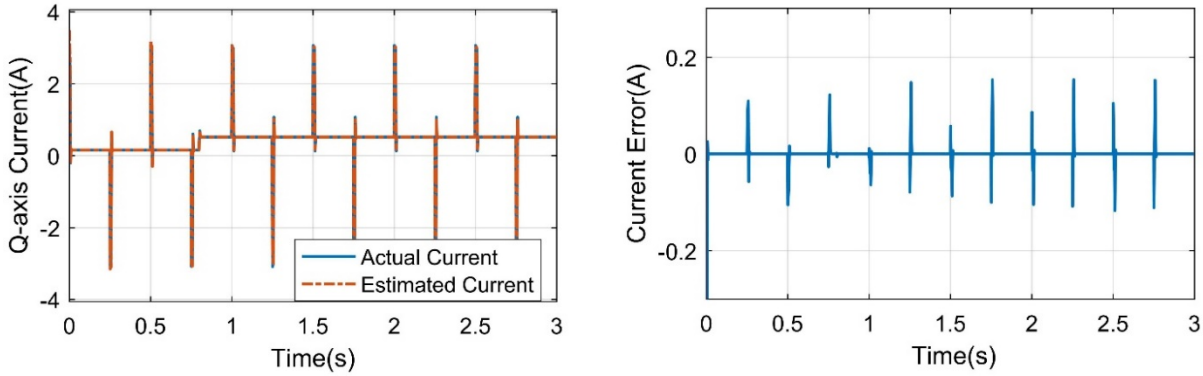
Fig. 10 to Fig. 12 show the simulation results at low speed. In the simulation experiment, the speed is given to 0.5 m/s, and make the motor do round trip motion at low speed, then 5kg load is added at 0.8s. From the Fig. 10, it can be seen that the estimated speed can follow the actual speed well. At startup, commutation and load, a slightly larger instantaneous error occurs with a maximum error of 40%.



(a) (b)

Fig. 10. Two kinds of speeds and their error waveforms at low speed

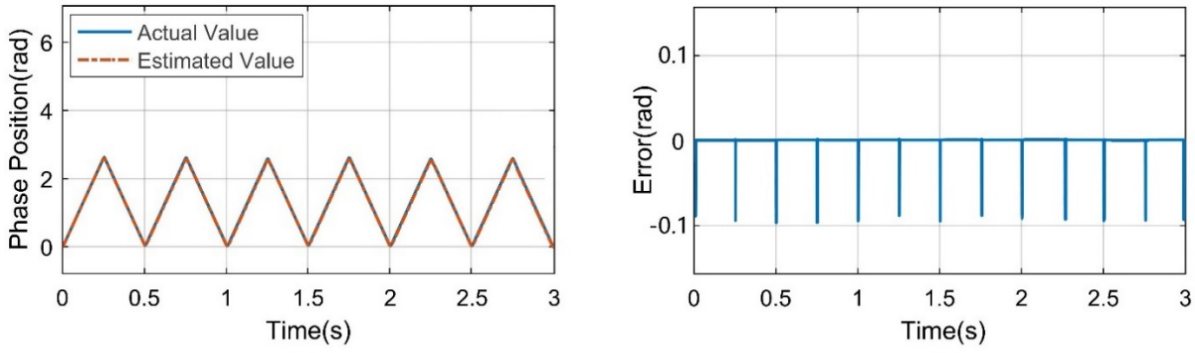
From the Fig. 11, it can be seen that the estimated Q-axis current can follow the actual Q-axis current well. When the motor starts, switches direction and sudden load, a slightly larger error occurs with a maximum error of 6%.



(a) (b)

Fig. 11. Two kinds of Q-axis currents and their error waveforms at low speed

In Fig. 12, there are waveforms of actual phase, estimated phase and their error, it can be seen that the waveform of the estimated phase was basically identical with the actual phase, and there is a slighter large error when switching direction.



(a) (b)

Fig. 12. Two kinds of phase positions and their error waveforms at low speed

4.2 Experiment Results

Due to the PMLSM used in the experiment has a short stroke and is unable to do the speed sensorless control experiment at high speed, this experiment make the PMLSM do round-trip motion at a speed of 0.05m/s, the same as the simulation experiment at low speed. In the experiment, taking the counting value of the encoder as the actual speed, which is not performed in motor control and is only used to compare with the speed estimation. Fig. 13(a) shows the curve of the actual speed and estimated speed. From the figure, it can be seen that the waveform of estimated value is basically consistent with the waveform of the actual speed, but the error is relatively large and the maximum error occurs in the commutation, the maximum error reaches 40%. The error waveform is as shown in Fig. 13(b).

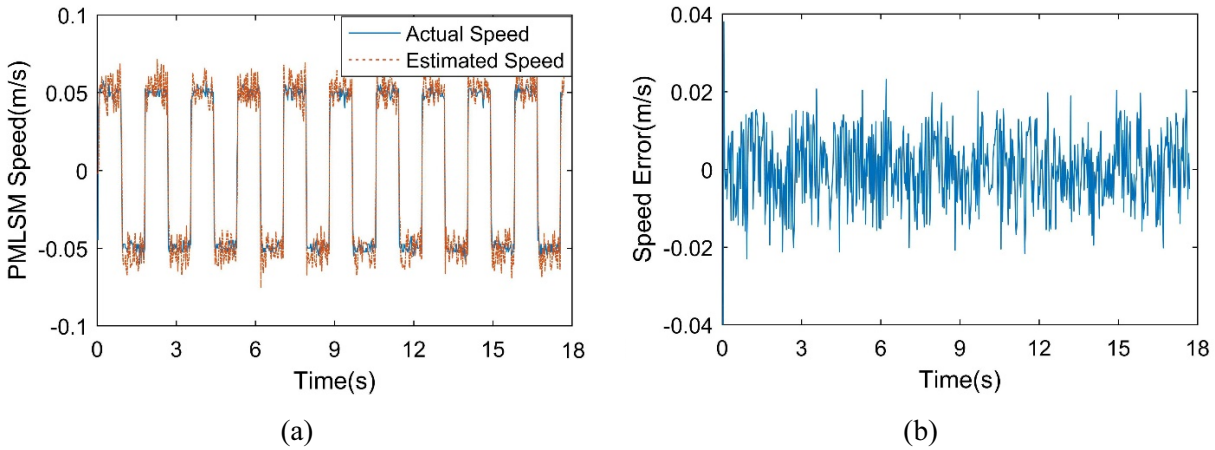


Fig. 13. Two kinds of speeds and their error waveforms at low speed

Compared with the simulation results, the experimental result is not good enough. There are three main reasons for this result. Firstly, the DSP28335 controller used in the experiment is a serial computing controller, which can not realize the parallel calculation of the neural network. Secondly, the interrupt frequency of the DSP28335 controller is only 5kHz, which is relatively low, so the control signal of the neural network has low timeliness. Thirdly, input signals of the neural network at low speed have large noise, and will change in a wide range when the motor changes direction, at this time, the neural network generates large-value updates and corrections, causing the estimated speed to deviate from the actual speed.

5 Conclusion and Future Work

RBF neural network has strong nonlinear function approximation ability. MRAC speed observer based on RBF neural network can accurately observe the motor speed in the whole speed range, so as to realize speed sensorless control of PMLSM. Simulation results show that the RFFNN MRAS speed observer can provide accurate speed information for the motor under the conditions of acceleration and deceleration, sudden load, round-trip movement in the whole speed range. The low-speed experiment also proves that the method can basically complete the speed estimation at low speed, but the experimental results are not good enough due to the limitations of the experimental platform.

In the future research, the linear motor with long stroke will be replaced to complete the performance test under high-middle speed operation, and the FPGA control chip with parallel computing capability will be selected to complete the calculation of RBF neural network to improve the calculation accuracy and experiment performance.

Acknowledgements

The authors would like to thank their laboratory team member's assistance. This work is supported by the National Natural Science Foundation of China [grant number 51707068 & 51477058], the Natural

Science Foundation of Fujian Province [grant number 2017J01097], the Subsidized Project for Postgraduates' Innovative Fund in Scientific Research of Huaqiao University [17014082002].

References

[1] H.-C. Lu, J. Ti, L.-L. Sun, A new sliding mode observer for the sensorless control of a PMLSM, Applied Mechanics and Materials 494-495(2014) 1401-1404.

[2] G.-R. Wang, J.-Y. Li, Application of modified UKF algorithm in PMLSM sensorless control, Transducer & Microsystem Technologies 36(2)(2017) 158-160.

[3] Y.-Y. Chao, R.-H. Zhang, Y.-M. Du, Speed sensorless control of maglev permanent magnet linear synchronous motor based on sliding mode observer, in: Proc. 2017 International Conference on Electrical Machines & Systems IEEE, 2017.

[4] L.-H. Quan, Z.-S. Wang, X.-C. Liu, Sensorless PMSM speed control based on NN adaptive observer, in: Proc. 2014 International Symposium on Neural Networks, 2014.

[5] H.-C. Lu, X.-Y. Yao, J. Ti, Combined back-EMF estimator with high-frequency signal injection for wide speed range sensorless control of PMLSM, Advanced Materials Research 753-755(2013) 1405-1408.

[6] W.-M. Yang, L.-J. Pan, H.-C. Lu, Neural-network-based adaptive sliding mode control for PMLSM, Electrical Automation 31(3)(2009) 9-11.

[7] X.-Z. Liu, J. Wu, C.-L Lei, Speed identification and control of PMSLM based on ESO, Control and Decision 20(3)(2005) 308-311.

[8] Z.-Y. Tang, M.-Y. Yang, Z.-C. Pei, Self-adaptive PID control strategy based on RBF neural network for robot manipulator, in: Proc. 2010 International Conference on Pervasive Computing, 2010.

[9] C.-L. Liu, G.-F. Ma, J.-C. Ma, Adaptive PID control strategy for nonlinear model based on RBF neural network, in: Proc. 2012 Affective Computing and Intelligent Interaction, 2012.

[10] Kilic, Erdal, O.-H. Riza, Y. Saban, Efficient speed control of induction motor using RBF based model reference adaptive control method, Automatika 57(3)(2016) 714-723.

[11] C.-L. Xia, D. Wen, J. Fan, Based on RBF neural network position sensorless control for brushless DC motors, Transactions of China Electrotechnical Society 17(3)(2002) 26-29.

[12] Y.-Y. He, Q.-F. Xu, S.-L. Yang, A power load probability density forecasting method based on RBF neural network quantile regression, Proceedings of the CSEE 33(1)(2013) 93-98.

[13] C.-L. Xia, M.-C. Wang, T.-N. Shi, Position sensorless control for switched reluctance motors using neural network, Proceedings of the CSEE 25(13)(2005) 123-128.