Sensors & Transducers, Vol. 158, Issue 11, November 2013, pp. 302-311

www.sensorsportal.com

Sensors & Transducers

http://www.sensorsportal.com

Speed Sensorless Control with ANN and Fuzzy PI Adaptation Mechanism for Induction Motor Drive

[1] Kai XU, [2] Shanchao LIU
College of Information Science and Engineering, Chongqing Jiaotong University
Chongqing 400074, P. R. China
Tel.:[1] 15123139839, [2] 13618348922
E-mail: xkxjxwx@hotmail.com, 676053114@qq.com

Received: 9 September 2013 /Accepted: 25 October 2013 /Published: 30 November 2013

Abstract: In the speed sensorless induction motor drives system, the Rotor Flux based Model Reference Adaptive System (RF-MRAS) is the most common strategy. It suffers from parameter sensitivity and flux pure integration problems. As a result, it leads to the deterioration of speed estimation. Simultaneously, the traditional PI parameters design may cause speed estimation instability or have gross errors in the regenerative mode. To overcome above-mentioned problems, a suitable Artificial Neural Networks (ANN) based on Ant Colony Optimization (ACO) is presented to replace the reference model of the RF-MRAS. Furthermore, the ANN learning by the modified ACO can enhance the ANN convergence speed and avoids the trap of local minimum value of algorithm. In the meantime, a fuzzy PI adaptation mechanism is also put forward, so the proportional coefficient k_p and the integral coefficient k_i can be adjusted dynamically to adapt the speed variations. Finally, the simulation results suggest that the speed estimation is more accurate in both the dynamic and static process, and the stability of speed estimation in regenerative mode was improved. *Copyright © 2013 IFSA.*

Keywords: Vector control, Speed sensorless, ANN, Modified ACO, Fuzzy PI adaptation mechanism.

1. Introduction

Nowadays, Field-Oriented Control (FOC) is used for high performance variable speed drives. Implementation of FOC technique requires the motor speed information. Tachogenerators, resolvers, incremental or optic encoders are usually used to detect the rotor speed. However, these sensors affect on reliability, simplicity and ruggedness of induction motor drive. The other requirements of these sensors are careful mounting and alignment, and special attention with electrical noises. Additional space for mounting and maintenance is required for speed sensor and hence cost and size of the drive system will be increased [1]. Therefore, elimination of speed sensor reduces the total cost of the drive system. Also, in emergency applications the induction motor drive must be able to continue its work in the failure of speed sensor. Sensorless drive system is more versatile due to the absence of the numerous problems associated with the speed sensor as some of them discussed previously. Therefore, it is encouraged to use the sensorless drive where the speed is estimated using a control algorithm instead of measuring. It should be noted that the speed sensor elimination degrades performance of drive system. The stability at the low speed operation range and the parameter sensitivity can be the main drawbacks of sensorless control [2].

There are several speed estimation methods that are based on different algorithms and most of them use dynamic equations of the induction motor.

Article number P_1502

Among several sensorless speed estimation strategies [3-5], Model Reference Adaptive System (MRAS) has been chosen to be evaluated in this research. The main reason behind this choice is so obvious because MRAS has been proclaimed as one of the best available methods, especially when the motor parameters are poorly known or have large variations [6-7]. Rotor Flux based Model Reference Adaptive System (RF-MRAS) scheme suffers from parameter sensitivity and flux pure integration problems which may cause DC drift. It leads to the deterioration of estimation at low speed [1]. Y. Yusof et al. [8] proposed Low-Pass Filters (LPF) with very low cut-off frequency to replace the pure integrator, but it introduces phase and gain errors due to its natural delay. Nonlinear feedback integrators for drift and DC offset compensation have been proposed in [9].

Currently, an integrated research on Artificial Neural Networks (ANN) RF-MRAS is still further explored [10-11]. The ANN may be used to improve the performance of the sensorless drives. The applications of ANN in sensorless drives have been reviewed in [12]. Generally, the Grads Descend Method (GDM) learning algorithms is used to adjust the neural network weights and threshold values. In [13], the training is performed using the Levevberg-Marquardt algorithms. An algorithm based on total least square for the training of linear neuron has been reported in [14] to extract better performance from the RF-MRAS. All the methods mentioned above have the defects of low convergence of rate, plunging into the local minimum easily and bad generalization. In this paper, a modified Ant Colony Optimization (ACO) training method is proposed. The modified ACO algorithms change the neural network learning process to stochastic optimization problems. It increases the convergence rate, enhance the precision, and avoid the local minimum.

On the other hand, The RF-MRAS make a problem related to the strong dependence of the rotor time constant T_r which causes a problem of adjustment of the adaptation mechanism gains, especially with a classical PI [15]. The performance of speed estimation is greatly depends on PI block. If traditional PI parameters were used, speed estimator may become unstable or may have gross errors in regenerative mode. It is because of abrupt and uncontrolled variation of rotor flux magnitude [13]. The instability a phenomenon under regeneration is generally found in speed adaptive observers based sensorless drive [16-17]. E. Babaei et al. [18] proposed a new predictive and adaptive method for design of PI block. In [19], the PI controller in the adaptation mechanism is replaced by a radial basis function neural network. A fictitious quantity is used in [20] to overcome instability problem under regeneration. In a different approach, Hinkkanen et al. [16] modifies the error signal to improve the stability in the regenerating mode. The conventional PI adaptation mechanism can not provide perfect speed estimation because of the fixed parameters. In this work we presented a fuzzy PI controller for adaptation mechanism in RF-MRAS observer. The fuzzy logic controller is used for the on-line tuning of the PI adaptation mechanism, then the proportional coefficient k_p and the integral coefficient k_i can be adjusted dynamically to adapt the speed variations. It improves stability of speed estimation in the regenerative mode, as well as enhances the robustness under a load torque vibrations.

In this thesis, firstly, we use a well trained ANN based on ACO to replace the reference model of the RF-MRAS. The ANN generates accurate rotor flux in different command speed. So, the proposed method has good speed identification in both the dynamic and static respond process. Secondly, a fuzzy PI adaptation mechanism is used to advanced the stability of speed estimation in regenerative mode, and enhances the robustness in the sensorless indirect vector control system. And finally, the effectiveness of this method is proved by the simulation experiments.

2. The Principle of RF-MRAS Speed Observer

The block diagram of a model based sensorless indirect vector control scheme is shown in Fig. 1. Based on terminal voltages and currents, the rotor speed is estimated through the proposed RF-MRAS. The estimated rotor speed is used for coordinate transformation and to close the speed loop.

The basic concept of MRAS is the presence of a reference model which determines the desired states and an adjustable model which generates the estimated values of the states. The error between these states is fed to an adaptation mechanism to generate an estimated value of the rotor speed which is used to adjust the adaptive model. This process continues till the error between two outputs tends to zero. The structure of the RF-MRAS is shown in Fig. 2.

The building blocks of such RF-MRAS are as follows.

Reference model: The reference model computes rotor flux in stationary reference frame using (1).

$$\begin{cases} \dfrac{d\psi_{r\alpha}}{dt} = \dfrac{L_r}{L_m}(u_{s\alpha} - R_s i_{s\alpha} - \sigma L_s \dfrac{di_{s\alpha}}{dt}) \\ \dfrac{d\psi_{r\beta}}{dt} = \dfrac{L_r}{L_m}(u_{s\beta} - R_s i_{s\beta} - \sigma L_s \dfrac{di_{s\beta}}{dt}) \end{cases}, \quad (1)$$

where $u_{s\alpha}$ and $u_{s\beta}$ are the α and β components of the stator voltage vector; $i_{s\alpha}$ and $i_{s\beta}$ are the α and β components of the stator current vector; R_s is the stator resistance; L_s, L_m and L_r are the stator, mutual and rotor inductances respectively; σ is the leakage coefficient.

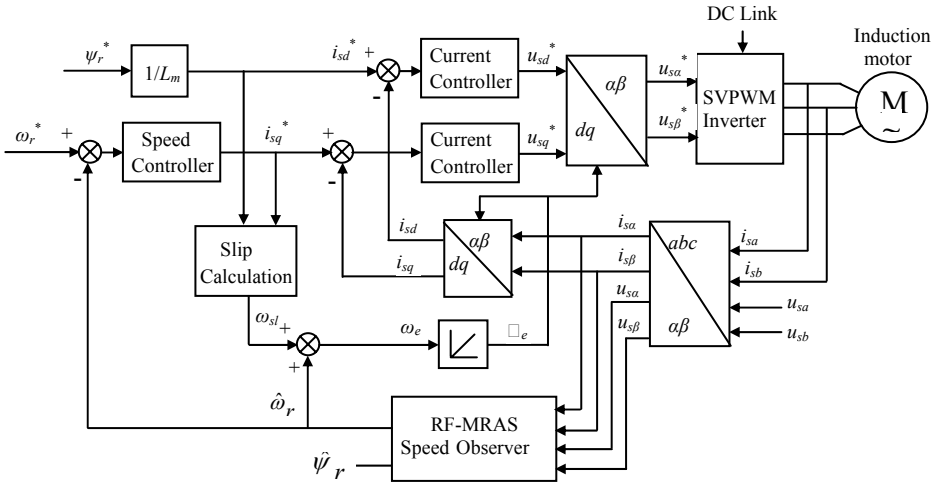


Fig. 1. The schematic diagram of sensorless indirect vector control system.

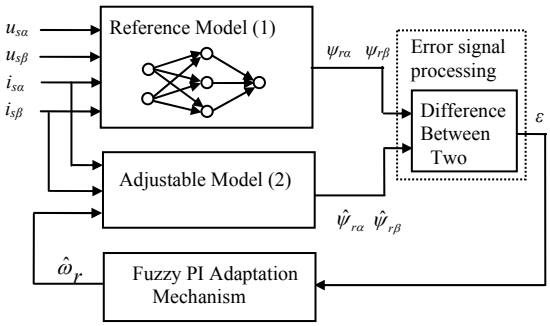


Fig. 2. RF-MRAS speed observer.

The reference model (1) is based on stator equations and is therefore independent of the motor speed. Such RF-MRAS is influenced by the variation of induction motor parameters (e.g., stator resistance). When working at low or zero speed operation range the observer performance deteriorates due to the integrator drift and initial condition problems and sensitivity to current measurement noise. Therefore an artificial neural network is proposed to replace the conventional voltage model flux observer (1) to improve the RF-MRAS scheme performance.

Adjustable model: The adjustable model computes rotor flux in stationary reference frame using (2).

$$\begin{cases} \dfrac{d\hat{\psi}_{r\alpha}}{dt} = -\dfrac{1}{T_r}\hat{\psi}_{r\alpha} - \hat{\omega}_r\hat{\psi}_{r\beta} + \dfrac{L_m}{T_r}i_{s\alpha} \\ \dfrac{d\hat{\psi}_{r\beta}}{dt} = -\dfrac{1}{T_r}\hat{\psi}_{r\beta} + \hat{\omega}_r\hat{\psi}_{r\alpha} + \dfrac{L_m}{T_r}i_{s\beta} \end{cases}, \quad (2)$$

where T_r is the rotor time constant, $\hat{\omega}_r$ is the estimated rotor speed.

The adaptive model (2) is speed-dependant since it is derived from the rotor equation in the stationary reference frame.

Error signal processing: The error between the two models is:

$$\varepsilon = \hat{\psi}_r \times \psi_r = \hat{\psi}_{r\alpha}\psi_{r\beta} - \hat{\psi}_{r\beta}\psi_{r\alpha} \quad (3)$$

The error signal ε is used to drive an adaptation mechanism.

Fuzzy PI adaptation mechanism: In Fig. 2, to obtain a stable nonlinear feedback system, the error signal ε and a Fuzzy PI controller are used in the adaptation mechanism to generate the estimated speed $\hat{\omega}_r$. The estimated speed is used as feedback for control system. The aim of this mechanism is to improve the control and sensorless speed estimation of induction motor in high, low speed and also under a load torque vibrations.

3. ANN for the Computation of Reference Model

3.1. The ANN Structure Design and Signal Computations

As expressed in (1), it is observed that ψ_r is a function of $u_{s\alpha}, u_{s\beta}, i_{s\alpha}$, and $i_{s\beta}$.

$$\psi_r = f(u_{s\alpha}, u_{s\beta}, i_{s\alpha}, i_{s\beta}) \quad (4)$$

The nonlinear function $f(\cdot)$ is unknown to the users. However, the inputs and the corresponding outputs are known. So, the problem relies on the realization of the nonlinear function $f(\cdot)$ through input-output mapping. ANN is a well-known tool, which may be used for such purpose. To estimate the rotor flux in the reference model, a 8-17-2 multilayer feedforward neural network is proposed as shown in Fig. 3.

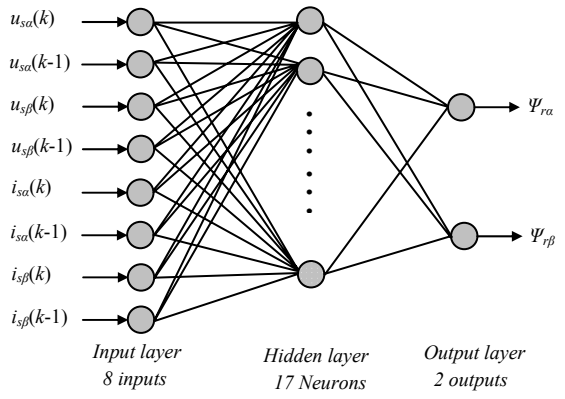


Fig. 3. Structure of the ANN used in the reference model.

To obtain good estimation accuracy, the inputs to the network are the present and past values of the α-β components of the stator voltage and current in the stationary reference frame. This is different from [21] which uses only the present samples. Better performance can be obtained by increasing the number of inputs to include voltage and current samples from more than one time step in the past. Variable values like present $u_{s\alpha}(k)$, $u_{s\beta}(k)$, $i_{s\alpha}(k)$, $i_{s\beta}(k)$ and one time step in the past $u_{s\alpha}(k-1)$, $u_{s\beta}(k-1)$, $i_{s\alpha}(k-1)$, $i_{s\beta}(k-1)$ are treated as the inputs of the ANN.

Generally speaking, the initial value of q (the number of nodes of hidden layer) can be obtained according to the empirical formula: $q \geq 2p+1$ (p is the number of nodes of input layer). Finally we make q as 17 through the experiment. The output layer consists of two neurons representing the rotor flux components.

The various signals at different stages of the network are computed as follows.

Consider a neuron j in a layer m with n inputs in the (m-1) layer, the net input to the neuron is given by:

$$\begin{aligned} net_j &= \sum_{k=1}^{n} w_{jk} x_k + b_j \\ &= w_{j1}x_1 + w_{j2}x_2 + ... + w_{jn}x_n + b_j, \end{aligned} \tag{5}$$

where w_{jk} is the weight from n inputs in the (m-1) layer to the neuron j, x_k is the k^{th} output signal in the (m-1) layer to the neuron j, b_j is the threshold of the neuron j.

The neuron output is given by:

$$y_j = g(net_j) = g(\sum_{k=1}^{n} w_{jk} x_k + b_j), \tag{6}$$

where $g(\cdot)$ is the activation function or the neuron transfer function. Here, we used sigmoidal function in the hidden layer neuron, the neuron transfer function can be written as:

$$y_j = \frac{1}{1+\exp(-net_j)} \tag{7}$$

Whereas, linear function is considered for the output layer neuron.

In this paper, a modified ACO training method is proposed, because the Grads Descend Method (GDM) learning algorithm has the defects of low convergence rate, plunging into the local minimum easily and bad generalization. The modified ACO algorithms changes the neural network learning process to stochastic optimization problems. It increases the convergence rate, enhance the precision, and avoid the local minimum.

3.2. Rank-Weight-Based Ant Colony Optimization (ACOrw)

The basic thought of ACOrw is: ranking the routes got after each ant achieved once circulation according to their length. The contribution of each ant regenerating the pheromone depends on the length of the routes got in the process of circulation. The shorter of the route, the more contribution of the ant makes. This is to regenerate the weight coefficient according to the changes of the pheromone concentration of the best No.k, basing on ant circumference system model. In this way, each ant makes contribution in the global regenerate strategy. After less iterative times, the same better solution can be found. This method can save the computation time greatly, and this is very advantageous for searching the solutions of large-scale problem.

The basic formulas of ACO are shown as follow:

$$P_r^k(t) = \frac{\tau_j(t)}{\sum_{g=1}^{N} \tau_g(t)} \tag{8}$$

$$\tau_j(t) = (1-\rho)\tau(t) + \rho\Delta\tau_j(t) \tag{9}$$

$$\Delta\tau_j(t) = \sum_{k=1}^{h} \Delta\tau_j^k(t) \tag{10}$$

$$\Delta\tau_j^k(t) = Q / L_N \tag{11}$$

where $P_r^k(t)$ is the transfer probability of ant No. k at t time at route L_N; $\tau_j(t)$ is the pheromone quantity released by ant at t time at route L_N; $\Delta\tau_j^k(t)$ is the pheromone variety of ant No. k at t time at route L_N; L_N is the is the distance between

two points; ρ is pheromone volatility coefficient; Q is constant, which is used to the adjustment speed of pheromone.

Based on the basic ACO, the modified formulas of ACOrw is:

$$\Delta\tau_j^k(t) = \lambda^k \cdot Q / L_N \tag{12}$$

Here, λ^k is the weight coefficient of ant No. *k*.

3.3. Neural Network of ACOrw

The ACOrw is a global optimization algorithm. It used the thought of global regenerate and the weight coefficient. So we use it to train the weights and thresholds value of neural networks to avoid some shortcomings of ANN algorithm. Its basic thought is: supposing that there are *m* parameters in ANN, that mean they contain all possible weights and thresholds value, and there are *M* ants in ant colony. First, set up q_i ($1 \le i \le m$) (to be optimized), the parameter of ANN, to be *m* random nonzero value, and form set I_{qi}. Each ant chooses a group of weights and the threshold value in the set, supposing that the ant *k* of the group chooses the parameters independently under the prompting of pheromone concentration and the transfer probability. After all of the ants achieved choosing the parameters, they reach the food source. Then, regenerate the pheromone concentration and the parameters of the set. The modified formulas are shown as follow:

$$\Delta\tau_j^k(I_{qi}) = \lambda^k \cdot Q / e^k \tag{13}$$

$$e^k = (O_s - O_l)^2 / 2 \tag{14}$$

where $\Delta\tau_j^k(I_{qi})$ is the change of the pheromone concentration that ant *k* chooses No. *j* element in set I_{qi}; Q is constant, which is used to the adjustment speed of pheromone; e^k is the output error of the ANN (here it replaces the routes passed by the ants in ACO), O_s is the actual output of the ANN, O_l is the expectation output of the ANN; λ^k is the weight coefficient of ant No. *k*.

The realization procedures of the algorithm were shown as below:

Step 1. Initialize BP network structure : deciding the number of neurons in the input layer, hidden layer and output layer.

Step 2. Initialize pheromone concentration, individual optimum value and global optimum value.

Step 3. Compute the transfer probability of each ant by the optimization function.

Step 4. According to the transfer probability replace individual optimum value and global optimum value.

Step 5. Sort all of the routes and replace each ant pheromone concentration.

Step 6. Whether the algorithm error reach supposed precision or max iterations?

When the algorithm error achieves anticipated precision, all of the ants converge on the same route, or reach the assigned iterative times, thus the searching is finished; Otherwise returns to step (3), and the iterative process may be continued.

To generate the input and output training datas, the vector controlled induction motor running at different speed commands and subjected to various load torques is simulated. 4800 groups of data were obtained and are used to train the ANN. When the largest training times is 2000, and the smallest allowable error is 0.005. The training is performed off-line using GDM learning algorithms and modified ACO algorithms.

The experimental result shows that GDM can not reach the smallest allowable error 0.005 when the networks are trained 2000 times. However, modified ACO meets the error requirement after training 258 times. Experiments show the quick convergent rate and good precision of modified ACO algorithms.

The well-trained ANN is used in the reference model of the RF-MRAS to compute accurate rotor flux under all operating conditions. This will benefit from the following advantages: fault tolerance, noise immunity and fast processing speed.

4. Fuzzy PI Adaptation Mechanism

4.1. Structure of the Fuzzy PI Adaptation Mechanism

In the previous Fig. 2, the error between two models is the input of a Fuzzy PI adaptation mechanism which output is the estimated rotor speed. This estimated speed is used to adjust the adaptive model.

In the induction motor drives system, the stator resistance of motor may change with the temperature variation up to 50 %, motor magnetizing inductance varies with the magnetic operating point and becomes nonlinear near the saturation level. Furthermore, the load torque and inertia may change due to mechanical disturbances. Nonlinearity also arises in the drive system due to voltage and current limits of the power converter.

For above reasons, it is difficult to build an accurate mathematical model of induction motor. So the conventional PI adaptation mechanism can not provide perfect speed estimation performance because of the fixed parameters. In this article, a fuzzy logic controller is used for the on-line tuning of the PI adaptation mechanism shown in Fig. 4.

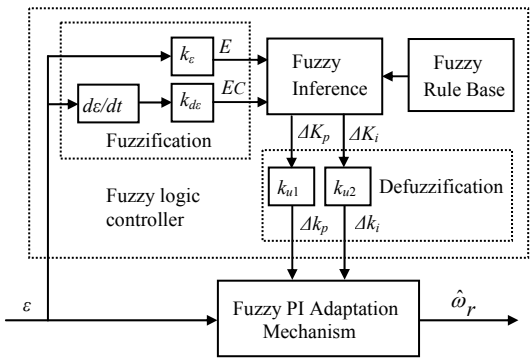


Fig. 4. Fuzzy PI adaptation mechanism.

In the induction motor drives system, the discrete expression of the PI adaptation mechanism is shown as in:

$$\hat{\omega}_r(k) = k_p\varepsilon(k) + k_i T\sum_{j=1}^{k}\varepsilon(j), \tag{15}$$

where T is the sampling period, $\varepsilon(k)$ is the error signal between two models, k_p is the proportional coefficient, k_i is the integral coefficient, $\hat{\omega}_r$ is the estimated speed. Equation (15) shows that the $\hat{\omega}_r$ value can be easily acquired, by dynamically adjusting k_p and k_i according to the input error signal.

The fuzzy logic control process consists of three different stages: fuzzification, fuzzy inference and defuzzification. During the fuzzification stage, definite values of the input variables are converted through the membership functions and fuzzy sets into fuzzy values. Then, the fuzzified input values are evaluated through the control IF-THEN rules and the control output is generated. fuzzy output is then converted back to definite values through defuzzification method.

In this work the fuzzy logic controller is a Mamdani type, the error between the two models ε and rate change of error $d\varepsilon/dt$ are regarded as two input variables. The gain of proportional and integral coefficient Δk_p, Δk_i as output variables. The adequate choice of k_ε, $k_{d\varepsilon}$, k_{u1} and k_{u2} permits to adapt the estimation.

The Fuzzy inference is a system with 2-input, 2-output. The input variables of Fuzzy inference are the error E and the rate change of error EC. And the output variables ΔK_p and ΔK_i are corrected value based on a set of rules. Then the proportional coefficient k_p and integral coefficient k_i can be obtained by the following formula:

$$k_p = k_{p0} + \Delta k_p \tag{16}$$

$$k_i = k_{i0} + \Delta k_i, \tag{17}$$

where the k_{p0} and k_{i0} are the reference values which already adjusted. Through experiments, the PI parameters of the adaptation mechanism obtained: k_{p0}=4997, k_{i0}=1.98.

As showing in Fig. 4, the error signal $\varepsilon(k)$ is fuzzified into E, the rate change of the error signal $d\varepsilon(k)/dt$ is fuzzified into EC. In Fig. 5, seven membership functions were used to describe both of the inputs, and the corresponding fuzzy subsets are: NB (negative big), NM (negative medium), NS (negative small), ZE (zero), PS (positive small), PM (positive medium) and PB (positive big).

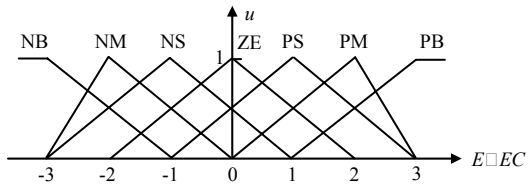
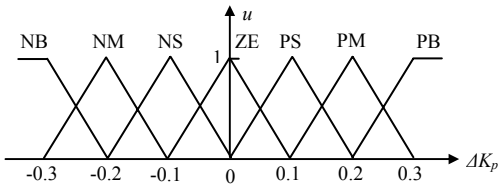
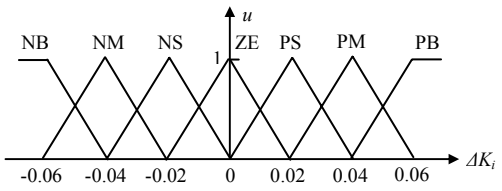


Fig. 5. Membership function of the inputs**.**

The output variables Δk_p and Δk_i are fuzzified into ΔK_p and ΔK_i respectively. They both include 7 fuzzy subsets: NB (negative big), NM (negative medium), NS (negative small), ZE (zero), PS (positive small), PM (positive medium) and PB (positive big), and their membership functions are shown as the Fig. 6, respectively.



(a) membership function of ΔK_p



(b) membership function of ΔK_i

Fig. 6. Membership function of the outputs.

4.2. Rules of the Fuzzy Logic Controller

Fuzzy control rules are regarded as the core of the whole fuzzy control. According to the error signal E

and its rate of change EC, the principles about the fuzzy PI adaptation mechanism are list as follows:

1) When E takes a relatively big number, a big number should also be assigned to ΔK_p in order to accelerate the system response, but ΔK_i must take a minimal number or even zero in order to prevent integral saturation and distinct speed overshooting;

2) When E takes a moderate number, ΔK_p must take a relatively small number and ΔK_i take a moderate one, in order to decrease the overshoot and ensure the swift speed response;

3) When E takes a relatively small number, the system usually run in steady state, thus a moderate ΔK_p and a big ΔK_i should be assigned to decrease static error and ensure the stability of the system.

According to the error signal E and its rate of change EC, fuzzy control rules of ΔK_p and ΔK_i can be acquired as shown in Table 1 and Table 2.

Table 1. Control rules of ΔK_p.

EC	***E***						
	NB	NM	NS	ZE	PS	PM	PB
NB	PB	PB	PM	PM	PS	ZE	ZE
NM	PB	PB	PM	PS	PS	ZE	NS
NS	PM	PM	PM	PS	ZE	NS	NS
ZE	PM	PM	PS	ZE	NS	NM	NM
PS	PS	PS	ZE	NS	NS	NM	NM
PM	PS	ZE	NS	NM	NM	NM	NB
PB	ZE	ZE	NM	NM	NM	NB	NB

Table 2. Control rules of ΔK_i.

EC	***E***						
	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NM	NM	NS	ZE	ZE
NM	NB	NB	NM	NS	NS	ZE	ZE
NS	NB	NM	NS	NS	ZE	PS	PS
ZE	NM	NM	NS	ZE	PS	PM	PM
PS	NM	NS	ZE	PS	PS	PM	PB
PM	ZE	ZE	PS	PS	PM	PB	PB
PB	ZE	ZE	PS	PM	PM	PB	PB

5. System Simulation Results

In this section, the software Matlab is used to simulate the whole system to examine the performance of speed identification with the proposed method. Through a lot of experiments, the parameters of modified ACO used in the simulation experiments are as followings:

$M = 40$, $\rho = 0.6$, $\lambda = 0.5$, with 0.005 supposed precision.

For a induction motor, with the following parameters:

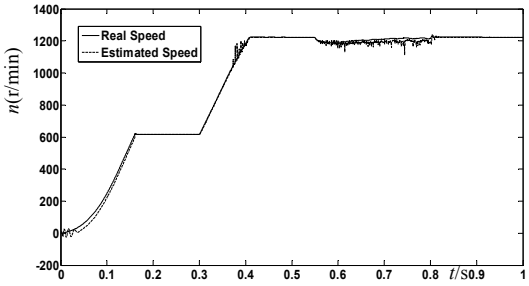
$P_n = 2.2\ kW$, $U_n = 380\ V$, $f_n = 50\ Hz$, $R_s = 0.435\ \Omega$, $R_r = 0.816\ \Omega$, $L_s = L_r = 2.08\ mH$, $L_m = 69.4\ mH$, $J = 0.18\ kg \cdot m^2$, $p = 2$.

The following several cases include different speed command, operating in the regenerating mode. Simultaneously, the disturbance of the load is considered. The estimated speed and real speed curves of the method are given, and the electromagnetic torque and rotor flux curves are also presented.

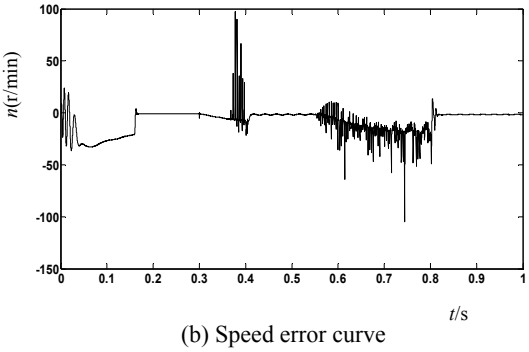
5.1. Dynamic and Static Characteristics

What is shown from Fig. 7 to Fig. 9 are the simulation results of a staircase type speed command. The command speed ω_r^* is changed from 0 to 1200 r/min through 0→600→1200 r/min. Induction motor is suddenly loaded with 90 N·m at t=0.55 *sec*, and unloading at t=0.8 *sec*.

Fig. 7(a) is the comparison between real speed and estimated speed. It is noticed that both of them are closely following. Fig. 7(b) is the speed error curve.



(a) Real and estimated speed



(b) Speed error curve

Fig. 7. The estimated speed with the proposed method.

By analyzing the waveforms, although there are errors always in the dynamic process, the maximal speed error does not exceed 8.2 %, and the estimated speed can rapidly track the real speed. When reached

a speed of 600 r/min or 1200 r/min, the estimated speed is very precise in steady state. Small deviation between the real speed and estimated speed is noticed only in the instant loaded at t=0.55 *sec*. When removing the load t=0.8 *sec*, the estimated speed is in good accordance with the real speed. The dynamic and static performance of the RF-MRAS is improved because of the reference model is replaced by the well trained ANN. The well trained ANN can generate more accurate rotor flux in different command speed. As a result, the proposed method has high accurate in both the dynamic and static operating. It also has good robustness under a load torque vibrations, because of using the fuzzy PI controller for adaptation mechanism.

Fig. 8 is the electromagnetic torque curve. In the dynamic and loaded process, the torque keeps balance. Although there are tiny amplitude vibrates in the process, the electromagnetic torque is matched with the load torque. Fig. 9 is the variations of rotor flux magnitude in stationary reference frame. It shows that when motor is loaded, the rotor flux value varies in the stable states.

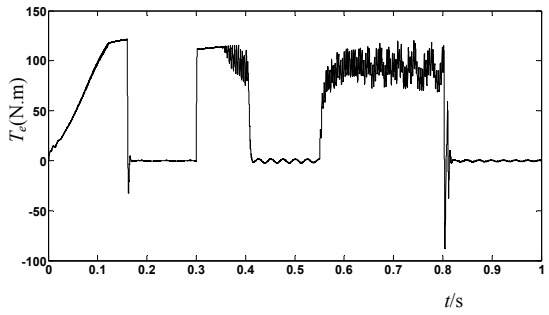


Fig. 8. Electromagnetic torque.

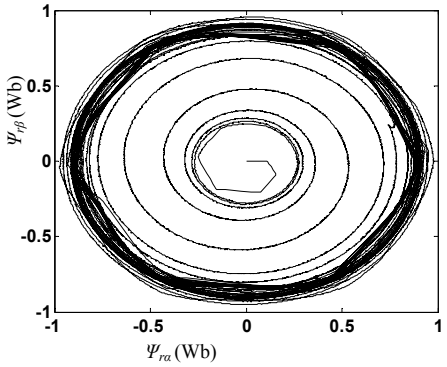
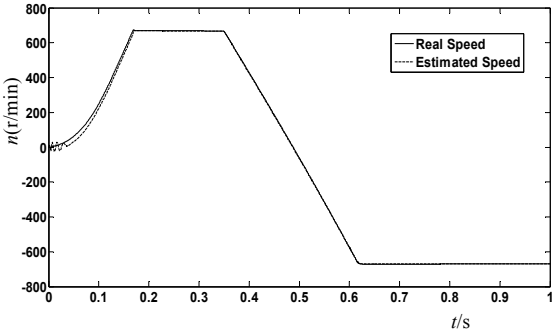


Fig. 9. Variations of rotor flux.

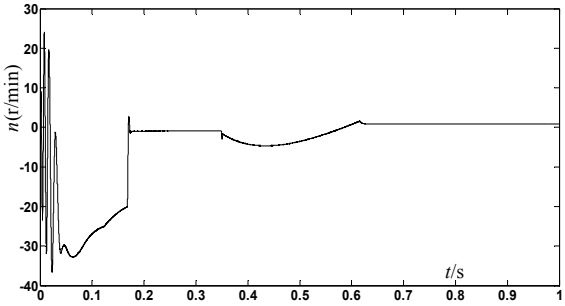
5.2. Stability in the Regenerative Mode

What is shown from Fig. 10 to Fig. 12 are the simulation results of command speed ω_r^* change from 650 r/min to -650 r/min at t=0.35 *sec*.

Fig. 10(a) is the comparison between the real speed and estimated speed. Fig. 10(b) is the speed error curve. It is observed that the estimated speed follows the real speed with good accuracy. In the dynamic process, the zero crossing has taken place at t=0.48 *sec*, and the estimated speed error does not exceed -5 r/min. The results reveal satisfactory performance around zero crossing. The performance is improved because of the fuzzy PI adaptation mechanism is used, and the well trained ANN can generate more accurate rotor flux. As a result, the proposed method has stable speed estimation performance in the regenerative mode.



(a) Real and estimated speed



(b) Speed error curve

Fig. 10. The estimated speed with the proposed method**.**

Fig. 11 is the electromagnetic torque curve. When the command speed ω_r^* change from positive to negative, the electromagnetic torque is matched with it. Fig. 12 is the variations of rotor flux magnitude curve. The curve reveals that the drive enters into the regenerative mode stability.

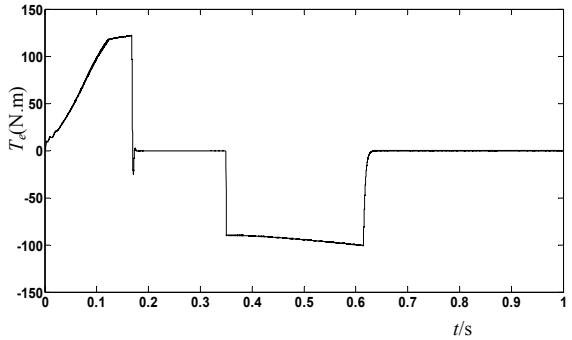


Fig. 11. Electromagnetic torque.

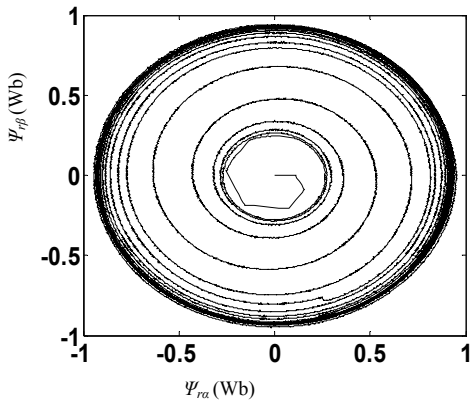


Fig. 12. Variations of rotor flux.

6. Conclusions

In the speed-sensorless induction motor drives system, the RF-MRAS is suffers from parameter sensitivity and flux pure integration problems. At the same time, the traditional PI parameters design may cause speed estimation instability especially in the regenerative mode. In this paper, a suitable ANN based on ACO is presented to replace the reference model of the RF-MRAS. As a result, the speed estimation is more accurate in both the dynamic and static process. Furthermore, a fuzzy PI adaptation mechanism is used to advance the stability of speed estimation in the regenerative mode, as well as enhances the robustness of the system.

Further research should focus on some new evolutionary computing approaches to optimize the ANN parameters, such as artificial fish swarm algorithm, particle swarm optimization, and so on. Simultaneously, the evolutionary computation should be combined with other intelligent control method to solve some practical problems.

Acknowledgements

This work is supported by the Major Project of Postgraduate Education Reform of Chongqing (No. yjg131001) and supported by Natural Science Foundation Projects of CQ CSTC (No. cstc2012jjA40019).

References

[1]. P. Vas, Sensorless vector and direct torque control, *Oxford University Press*, New York, 1998.

[2]. J. Holtz, J. Quan, Drift and parameter compensated flux estimator for persistent zero stator frequency operation of sensorless controlled induction motors, *IEEE Transactions on Industry Applications*, Vol. 39, No. 4, July/August 2003, pp. 1052-1060.

[3]. G. Foo, M. F. Rahman, Sensorless direct torque and flux controlled IPM synchronous motor drive at very low speed without signal injection, *IEEE Transactions on Industrial Electronics*, Vol. 57, No. 1, January 2010, pp. 395-403.

[4]. J. L. Chen, T. H. Liu, C. L. Chen, Implementation of a novel high-performance sensorless IPMSM control system, in *Proceedings of the IEEE International Conference on Industrial Technology (ICIT' 2010)*, Viña del Mar, Chile, 14-17 March 2010, pp. 361-366.

[5]. J. L. Chen, T. H. Liu, C. L. Chen, Design and implementation of a novel high-performance sensorless control system for interior permanent magnet synchronous motors, *IET Electric Power Applications*, Vol. 4, No. 4, April 2010, pp. 226-240.

[6]. A. Paladugu, B. H. Chowdhory, Sensorless control of inverterfed induction motor drives, *Electric Power Systems Research*, Vol. 77, 2007, pp. 619-629.

[7]. M. Dybkowski, T. Orlowska-Kowalska, Self-tuning adaptive sensorless induction motor drive with the stator current-based MRAS speed estimator, in *Proceedings of the IEEE Region 8 EUROCON 2009*, St. Petersburg, 18-23 May 2009, pp. 804-810.

[8]. Y. Yusof, A. H. M. Yatim, Simulation and modelling of stator flux estimator for induction motor using artificial neural network technique, in *Proceedings of the National Power and Energy Conference (PECon)*, 2003, pp. 11-15.

[9]. Q. Gao, C. S. Staines, G. M. Asher, M. Sumner, Sensorless speed operation of cage induction motor using zero drift feedback integration with MRAS observer, in *Proceedings of the European Conference on Power Electronics and Applications*, Dresden, Germany: EPE, 2005, pp. P1-P9.

[10]. Suman Maiti, Yoichi Hori, An adaptive speed sensorless induction motor drive with artificial neural network for stability enhancement, *IEEE Transactions on Industrial Informatics*, Vol. 8, No. 4, 2012, pp. 757-766.

[11]. T. Orlowska-Kowalska, M. Kaminski, FPGA implementation of the multilayer neural network for the speed estimation of the two-mass drive system, *IEEE Transactions on Industrial Informatics*, Vol. 7, No. 3, 2012, pp. 436–445.

[12]. B. K. Bose, Neural network applications in power electronics and motor drives – an introduction and perspectives, *IEEE Transactions on Industrial Electronics*, Vol. 54, No. 1, January 2007, pp. 14–33.

[13]. S. M. Gaduoue, D. Giaouris, J. W. Finch, A neural network based stator current MRAS observer for speed sensorless induction motor drives, in *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE' 08)*, Cambridge, United Kingdom, 2008, pp. 650-655.

[14]. M. Cirrincione, M. Pucci, Sensorless control of induction machines by a new neural algorithm: the TLS exin neuron, *IEEE Transactions on Industrial. Electronics*, Vol. 54, No. 1, January 2007, pp. 127–148.

[15]. A. Abbou, Speed sensorless simulation and experiment of asynchronous motor control strategy, Doctoral dissertation, Morocco, Rabat: College of Mohammadia des Ingénieurs, *University Mohammed 5 Agdal,* 2009.

[16]. M. Hinkkanen, L. Harnefors, J. Luomi, Reduced-order flux observers with stator-resistance adaptation for speed-sensorless induction motor drives, *IEEE Transactions on Power Electronics*, Vol. 25, No. 5, May 2010, pp. 1173-1183.

[17]. M. S. Zaky, Stability analysis of speed and stator resistance estimators for sensorless induction motor

drives, *IEEE Transactions on Industrial Electronics*, Vol. 59, No. 2, February 2012, pp. 858-870.

[18]. E. Babaei, M. B. B. Sharifian, R. A. Farshbaf, S. H. Hosseini, Verification of a new method for PI block design of MRAS-based sensorless speed estimators, in *Proceedings of the IEEE International Conference on Electrical Machines and Systems (ICEMS'2011)*, Beijing, China, 20-23 August 2011, pp. 1–6.

[19]. Wei Gao, Zhirong Guo, Speed sensorless control of PMSM using model reference adaptive system and RBFN, *Journal of Networks*, Vol. 8, No. 1, January 2013, pp. 213–220.

[20]. A. V. R. Teja, C. Chakraborty, S. Maiti, Y. Hori, A new model reference adaptive controller for four quadrant vector controlled induction motor drives, *IEEE Transactions on Industrial Electronics*, Vol. 59, No. 10, October 2012, pp. 3757-3767.

[21]. L. M. Grzesiak, B. Ufnalski, Neural stator flux estimator with dynamical signal preprocessing, in *Proceedings of the 7th IEEE Africon Conference in Africa*, Vol. 2, 2004, pp. 1137-1142.
