

# Self-Tuning PI Controller Based on Neural Network for Permanent Magnet Synchronous Motor

Jianguang Zhu, Zhifeng Zhang, Renyuan Tang

National Engineering Research Center for Rare-earth Permanent Magnet Machines

Shenyang University of Technology

Shenyang, Liaoning Province, China 110023

Jianguang\_zhu@sohu.com

## Abstract

*In servo motor drive applications, the variation of load inertia will degrade drive performance severely. Good dynamic and static performance of servo system requires controlling inertia robustly. In order to get the moment of rotational inertia, online identification methods based on model reference adaptive identification (MRAI) were developed in this paper. Then a well-trained neural network supplies the PI controller with suitable gain according to each operating condition pair (inertia, angular velocity error, and angular velocity) detected. To demonstrate the advantages of the proposed self-tuning PI control technique based on neural network, the simulation was executed in this research. The simulation results show that the method not only enhances the fast tracking performance, but also increases the robustness of the synchronous motor drive.*

## 1. Introduction

With progress in power electronics, microprocessors, and control theory, permanent magnet synchronous motors (PMSM), which possess high power/weight ratio and inherent maintenance-free capability, have been recognized as one of the key components in factory automation. But the system performance of the PMSM is influenced by uncertainties such as unpredictable plant parameter variations, external load disturbances and modeled nonlinear dynamics of the plant. In order to get high performance, accurate values of the machine's parameters are necessary [1]. Drive moment of inertia mainly affects the mechanical response of the system. When the moment of inertia increases, the system response will be slower. Therefore, it is necessary to identify the moment of inertia of the drive. Once the drive inertia is identified accurately, the parameters of the controller can be adjusted to compensate the inertia variation. System performance can be improved in this way.

Conventionally, PI control schemes dominate the controller market for PMSM. However, proportional-plus-integral (PI)-type control methods are not robust

enough to accommodate the variations of the external disturbances (torque load) and parameters and structural perturbations (inertia) during operation. Numerous methods have been proposed to replace PI-type control schemes. The methods include the model reference adaptive control [2], sliding mode, neural/fuzzy input-output linearization [3,4], and so on. The PI-type controller is still the most widely used for PMSM due to its relatively simple implementation. It is thus desired to have an intelligent PI-type controller that can self-tune its control gains according to the variations in operating conditions. This will take advantage of its simplicity and feasibility.

In this paper, the moment of rotational inertia is online identified based on model reference adaptive identification (MRAI) first. The self-tuning PI control system based on neural network to overcome the robustness problem of the fixed-gain PI-type controller is then developed. In this new approach, a controller based on neural network is trained first, and the trained neural network then serves as the dynamic gain supplier that will output suitable control gains according to the operating conditions which include the moment of rotational identified inertia. This realizes that the parameters of the controller can be adjusted to compensate the inertia variation. Analysis based on the simulation shows that the proposed scheme has good performance even with the dramatic variation of load inertia.

## 2. The inertia identification based on MRAI

### 2.1. Mathematical Model of the PMSM

The governing equation of an ac servo motor consists of two parts, electrical and mechanical.

1) Electrical Governing Equation:

With reference to synchronous rotating reference frame, the dynamic equations of IPM machine are given in

$$\begin{aligned} u_d &= p\psi_d - \psi_q\omega_r + R_s i_d \\ u_q &= p\psi_q + \psi_d\omega_r + R_s i_q \end{aligned} \quad (1)$$

$$\begin{aligned}\psi_d &= L_d i_d + \psi_f \\ \psi_q &= L_q i_q\end{aligned}\quad (2)$$

$$T_e = p_n (i_q \psi_d - i_d \psi_q) = p_n [\psi_f i_q + (L_d - L_q) i_d i_q] \quad (3)$$

## 2) Mechanical Governing Equation:

The torque created by the energy conversion process is then used to drive mechanical loads. Its expression is related to mechanical parameters via the fundamental law of the dynamics as follows

$$T_e = T_L + B\omega + J \frac{d\omega}{dt} \quad (4)$$

where

- $R_s$  motor phase resistance, [ $\Omega$ ];
- $L_d$  d-axis inductance, [H];
- $L_q$  q-axis inductance, [H];
- $\omega_r$  motor electrical angular velocity, [Rad/s];
- $\omega$  the machine angle velocity of rotor [Rad/s];
- $p_n$  number of magnetic poles;
- $T_e$  electromagnetic torque, [Nm];
- $T_L$  load torque, [Nm];
- $B$  viscous damping coefficient, [Nm.s];
- $J$  moment of inertia of the motor [ $\text{kg/m}^2$ ];
- $i_d$  d-axis current in synchronous frame, [A];
- $i_q$  q-axis current in synchronous frame, [A];
- $u_d$  d-axis voltage in synchronous frame, [V];
- $u_q$  q-axis voltage in synchronous frame, [V];
- $\psi_d$  d-axis flux linkage in synchronous frame, [Wb];
- $\psi_q$  q-axis flux linkage in synchronous frame, [Wb];
- $\psi_f$  PM flux linkage in synchronous frame, [Wb].

## 2.2 The inertia identification based on MRAI

Ignoring the effect of viscous damping coefficient  $B$ , the discrete form of mechanical movement equation of the PMSM drive system can be described as:

$$\omega[k] = \omega[k-1] + \frac{T}{J} (T_e[k-1] - T_L[k-1]) \quad (5)$$

where,  $T$  is the sampling period.

We can get (6) from (5) very easily,

$$\omega[k-1] = \omega[k-2] + \frac{T}{J} (T_e[k-2] - T_L[k-2]) \quad (6)$$

The load torque in (5) and (6) is assumed to change slowly, i.e.  $T_L[k-1] = T_L[k-2]$ . In servo system that requires a fast response, sampling period tends to be shorter. Therefore, this assumption is reasonable [5]. This leads to the following equation:

$$\begin{aligned}\omega[k] &= 2\omega[k-1] - \omega[k-2] \\ &+ \frac{T}{J} (T_e[k-1] - T_e[k-2])\end{aligned}\quad (7)$$

Equation (7) can be regarded as the difference equation of the reference mode. Using Landau's discrete time-recursive parameter identification [6], the MRAI algorithm is designed as:

$$\hat{\omega}[k] = 2\omega[k-1] - \omega[k-2] + \hat{b}[k-1]\Delta T_e[k-1] \quad (8)$$

$$\hat{b}[k] = \hat{b}[k-1] + \beta \frac{\Delta T_e[k-1]}{1 + \beta \Delta T_e[k-1]} \Delta \omega[k] \quad (9)$$

$$\begin{aligned}\Delta T_e[k-1] &= T_e[k-1] - T_e[k-2] \\ &= T_e[k] \cdot (Z^{-1} - Z^{-2})\end{aligned}\quad (10)$$

$$\Delta \omega[k] = \omega[k] - \hat{\omega}[k] \quad (11)$$

Whereas,  $\hat{\omega}$  is estimated angle velocity,  $\hat{b} = T/J$  is the parameter to be identified,  $\beta$  is adaptation gain.

Equation (8) is regarded as the difference equation for an adjustable model. Equation (9) is regarded as the adaptation mechanism. The block diagram of the proposed MRAI algorithm is illustrated in Fig. 1. The moment of inertia can be obtained very easily from (12).

$$\hat{J} = T / \hat{b} \quad (12)$$

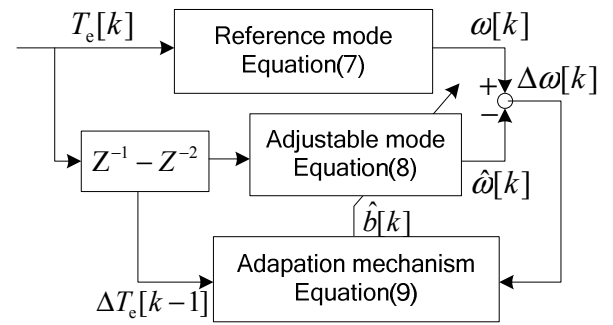


Fig. 1 MRAI algorithm block diagram

## 3. Self-tuning PI Control Based on Neural Network

### 3.1 Controller structure

Most of the PMSM controllers for industrial applications adopt a fixed-gain PI scheme. This fixed-gain scheme may work fine under certain operating conditions, but degrades its performance under other operating conditions. Moreover, suitable PI gains are usually obtained using time-consuming trial-and-error methods. Artificial neural networks, with their high learning and nonlinear mapping essences, have been successfully applied to many system identification and control problems [7]. To increase the robustness of conventional

PI control schemes, a self-tuning PI control system based on neural network that can self-tune its control gains according to various operating conditions is proposed. The block diagram of the proposed self-tuning PI control system based on neural network is illustrated in Fig.2. ANN denotes the neural network based parameter tuner

To enable the ANN to produce suitable gain according to the operating conditions, the neural network input patterns should contain suitable variables that can well represent the motor's operating conditions. In the proposed system, three variables  $e_\omega$ ,  $\omega$ , and  $\hat{J}$  are used. The ANN outputs are  $k_p$  and  $k_i$ .

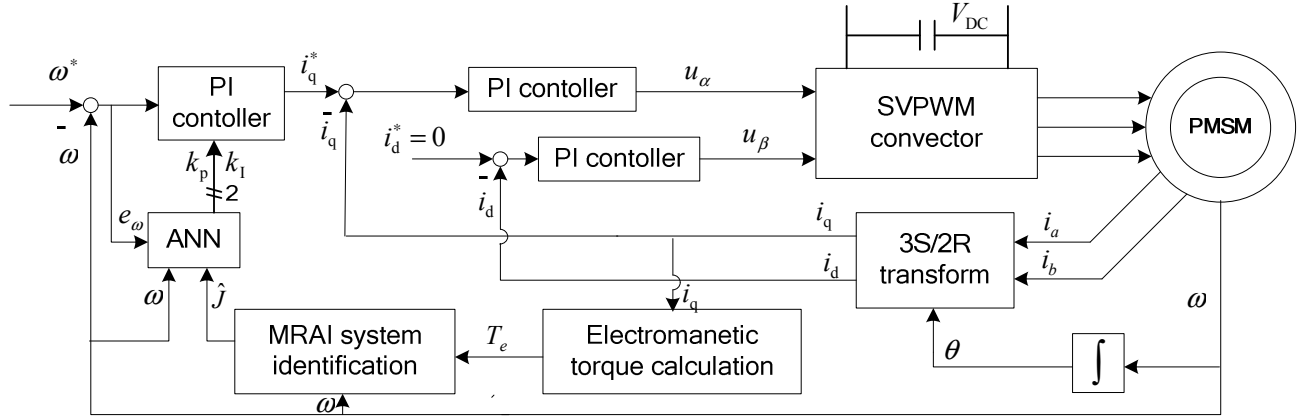


Fig. 2 Block diagram of the self-tuning PI control system based on neural network

Before the ANN can be used to serve as an intelligent parameter tuner, a training (or learning) process is required. Effective learning means that the network outputs will match the desired output for each training pattern selected from the application domain. In the proposed system, each training pattern is designed to contain five elements  $e_\omega$ ,  $\omega$ ,  $\hat{J}$ ,  $k_p$  and  $k_i$ . Where  $e_\omega$ ,  $\omega$ , and  $\hat{J}$  and are the input variables, and  $k_p$  and  $k_i$  are the relative desired outputs. Combinations of  $e_\omega$ ,  $\omega$ , and  $\hat{J}$  are as follows:

$e_\omega$ : from 0 to 150 rad/s, total is five equal intervals;

$\omega$ : from 0 to 150 rad/s, total is five equal intervals;

$\hat{J}$ : from  $J - 10J$ , total is five equal intervals.

To find the optimal desired outputs and for each input vector ( $e_\omega$ ,  $\omega$ , and  $\hat{J}$ ), a performance index is defined as:

$$F = F(\sigma, t_r, e_\omega) \\ = k_1 \sigma + k_2 t_r + k_3 e_\omega \quad (13)$$

where  $k_1$ ,  $k_2$  and  $k_3$  and are weighting factors chosen to be 100, 5, and 100, respectively. In this performance index, overshoot, rise time, and steady-state error represent the three cost variables. For each input vector or initial condition ( $e_\omega$ ,  $\omega$ , and  $\hat{J}$ ), several different ( $k_p$ ,  $k_i$ ) pairs are assigned to drive the PMSM from its initial state to the desired target. The ( $k_p$ ,  $k_i$ ) pair that

makes the minimum is the optimal PI gain with respect to each input vector ( $e_\omega$ ,  $\omega$ , and  $\hat{J}$ ). According to different combinations of  $e_\omega$ ,  $\omega$ , and  $\hat{J}$ , 125 total training patterns are thus obtained.

Instead of a brute-force search method, a more efficient method is proposed for determining the optimal pair of ( $k_p$ ,  $k_i$ ) with respect to each input vector ( $e_\omega$ ,  $\omega$ , and  $\hat{J}$ ). The ( $k_p$ ,  $k_i$ ) determination procedures are listed below.

Step 1) Divide  $k_p$  and  $k_i$  into several equal intervals, respectively.

Step 2) The procedure for Step 2) is diagrammed in Fig.3. For each  $k_p$ ,

a) Gradually increasing the value of  $k_i$  one interval per time while  $F(t+1) < F(t)$ ;

b) If  $F(t+1) > F(t)$  is detected, decrease the value of  $k_i$  by half the interval gradually until  $F(t+1) < F(t)$ ;

c) Increasing  $k_i$  by 1/4 the interval gradually until  $F(t+1) > F(t)$  is detected again;

d)  $k_{i\text{opt}}$  with respect to the selected  $k_p$  is  $k_{i\text{opt}} = (k_i(t) + k_i(t+1))/2$ .  $F_{\text{opt}}$  with respect to the selected ( $k_p$ ,  $k_{i\text{opt}}$ ) pair is  $F_{\text{opt}} = (F(t) + F(t+1))/2$ .

Step 3) Among all  $(k_p, k_{\text{iopt}})$  pairs obtained in Step 2), find the optimal pair that possesses the minimum  $F_{\text{opt}}$ .

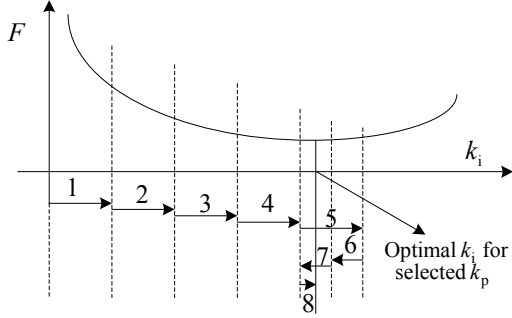


Fig.3 Sketch map of  $k_i$  determination procedures

The structure of the ANN is a three-layer feedforward neural network with input, hidden, and output nodes of 3, 40, and 2, respectively. The reason for using the feedforward type neural network is that it is simple to implement and is the most widely used neural network model.

The robust parameters self-tuning learning algorithm (RSTL), that can automatically adjust its learning parameters according to learning trajectories, is adopted to train the ANN for fast convergence.

### 3.2. Robust parameters self-tuning learning algorithm

Consider a multilayer feedforward neural-network. Assume  $Q$  is the number of training patterns;  $K$  is the number of nodes of the output layer. Energy function for any training pattern  $p$  can be expressed as

$$E_p = \frac{1}{2} \sum_{k=1}^K (d_{pk} - o_{pk})^2 \quad \text{for } p=1, \dots, Q \quad (14)$$

where  $d_{pk}$  and  $o_{pk}$  denote the desired output and network output of the  $k$ th node in the output layer for the training pattern  $p$ , respectively.

Assume  $N$  is the number of weights in the network considered. When  $\Delta w_i$  ( $i=1, \dots, N$ ) are small,  $\Delta E_p$ , the amount of change of  $E_p$ , can be approximated by the first-order Taylor series as

$$\Delta E_p \approx \sum_{i=1}^N \frac{\partial E_p}{\partial w_i} \Delta w_i \quad (15)$$

If the error back propagation is the selected learning rule, weights update by means of

$$\Delta w_i = -\eta \frac{\partial E_p}{\partial w_i} + \alpha \Delta w_i(t-1) \quad (16)$$

where  $\eta$  and  $\alpha$  denote the learning constant and

momentum term, respectively. Substitute Equation (16) into Equation (15),

$$\begin{aligned} \Delta E_p &\approx \sum_{i=1}^N \frac{\partial E_p}{\partial w_i} \left( -\eta \frac{\partial E_p}{\partial w_i} + \alpha \Delta w_i(t-1) \right) \\ &= -\eta \sum_{i=1}^N \left( \frac{\partial E_p}{\partial w_i} \right)^2 + \alpha \sum_{i=1}^N \frac{\partial E_p}{\partial w_i} \Delta w_i(t-1) \quad (17) \\ &= -\eta A(p) + \alpha B(p) \end{aligned}$$

Define a performance index as:

$$J = \frac{1}{2} (E_p + \Delta E_p)^2 + \frac{1}{2} \sum_{i=1}^N \Delta w_i(t)^2 \quad (18)$$

The purpose of term  $(E_p + \Delta E_p)$  is to have  $-\Delta E_p$  as close to  $\Delta E_p$  as possible. And the role of the summation of  $\Delta w_i$  is like a penalty function defined on  $\Delta w_i$  to prevent  $\Delta w_i$  from being too large. Substitute  $\Delta E_p$  (Equation (17)) and  $\Delta w_i$  (Equation (16)) into Equation (18), we find

$$\begin{aligned} J &= \frac{1}{2} (E_p - (\eta A(p) - \alpha B(p)))^2 \\ &\quad + \frac{1}{2} \sum_{i=1}^N \left( -\eta \frac{\partial E_p}{\partial w_i} + \alpha \Delta w_i(t-1) \right)^2 \quad (19) \end{aligned}$$

To obtain an efficient way that can adaptively adjust  $\eta$  and  $\alpha$ , let us rewrite Equation (16) in the vector form as

$$\Delta W(t) = -\eta \nabla(t) + \alpha \Delta W(t-1) \quad (20)$$

where

$$\Delta W(t) = [\Delta w_1(t) \dots \Delta w_N(t)], \quad \nabla(t) = \left[ \frac{\partial E_p}{\partial w_1} \dots \frac{\partial E_p}{\partial w_N} \right]$$

where vector  $-\eta \nabla(t)$  denotes the contribution of the current pattern and vector  $\alpha \Delta W(t-1)$  represents the contribution from patterns already trained.

For efficient training, it is desired that the length of vector  $-\eta \nabla(t)$  shall be longer than that of vector  $\alpha \Delta W(t-1)$ . Let

$$\gamma \|-\eta \nabla(t)\| = \|\alpha \Delta W(t-1)\|, \quad 0 < \gamma \leq 1 \quad (21)$$

Since both  $\eta$  and  $\alpha$  are required to be positive, Equation (21) can be rewritten as

$$\alpha = \frac{\|\nabla(t)\|}{\|\Delta W(t-1)\|} \gamma \eta = K \gamma \eta \quad (22)$$

where  $K$  is defined as

$$K \equiv \frac{\|\nabla(t)\|}{\|\Delta W(t-1)\|} \quad (23)$$

Substitute Equation (22) into Equation (19),  $J$  can be expressed as

$$J = \frac{1}{2}(E_p - (\eta A(p) - \alpha B(p)))^2 + \frac{1}{2} \sum_{i=1}^N \left( -\eta \frac{\partial E_p}{\partial w_i} + \gamma K \eta \Delta w_i(t-1) \right)^2 \quad (24)$$

For the current pattern,  $A(p)$ ,  $B(p)$ ,  $K$  and  $\alpha$  in Equation (24) are constants. Equation (24) is then a quadratic function of  $\eta$ . The optimal value of  $\eta$  can be obtained by taking the derivative of  $J(\eta)$  and setting the result to zero, i.e.,

$$\begin{aligned} \frac{\partial J}{\partial \eta} = & -(A(p) - \gamma KB(p))(E_p - \eta(A(p) - \gamma KB(p))) \\ & + \eta \left( \sum_{i=1}^N \left( \frac{\partial E_p}{\partial w_i} \right)^2 - 2\gamma K \sum_{i=1}^N \frac{\partial E_p}{\partial w_i} \Delta w_i(t-1) \right. \\ & \left. + \gamma^2 \sum_{i=1}^N K^2 \Delta w_i^2(t-1) \right) \\ = & -(A(p) - \gamma KB(p))(E_p - \eta(A(p) - \gamma KB(p))) \\ & + \eta \left( A(p) - 2\gamma KB(p) + \gamma^2 \sum_{i=1}^N K^2 \Delta w_i^2(t-1) \right) \end{aligned} \quad (25)$$

whereas

$$\begin{aligned} \gamma^2 \sum_{i=1}^N K^2 \Delta w_i^2(t-1) &= \gamma^2 \frac{\|\nabla(t)\|^2}{\|\Delta W(t-1)\|^2} \|\Delta w(t-1)\|^2 \\ &= \gamma^2 \|\nabla(t)\|^2 = \gamma^2 \sum_{i=1}^N \left( \frac{\partial E_p}{\partial w_i} \right)^2 \quad (26) \\ &= \gamma^2 A(p) \end{aligned}$$

Substituting Equation (26) into Equation (25), we find

$$\begin{aligned} \frac{\partial J}{\partial \eta} = & -(A(p) - \gamma KB(p))(E_p - \eta(A(p) - \gamma KB(p))) \\ & + \eta(A(p) - 2\gamma KB(p) + \gamma^2 A(p)) \\ = & -E_p(A(p) - \gamma KB(p)) + \eta(A(p) - \gamma KB(p))^2 \\ & + \eta(A(p) - 2\gamma KB(p) + \gamma^2 A(p)) \\ = & 0 \end{aligned} \quad (27)$$

Rearranging Equation (26),  $\eta$  and  $\alpha$  can be obtained as

$$\eta = \frac{E_p(A(p) - \gamma KB(p))}{E_p(A(p) - \gamma KB(p)) + A(p) - 2\gamma KB(p) + \gamma^2 A(p)} \quad (28)$$

$$\alpha = \frac{\|\nabla(t)\|}{\|\Delta W(t-1)\|} \eta \quad (2)$$

One  $\eta$  and  $\alpha$  are found, weights are updated by using Equation (20).

The  $\eta$  obtained from Eq. (28) is greater than zero, which is proofed in [8].

## 4. Simulations and Discussions

In this section, computer simulations of the PMSM servo system using the proposed self-tuning PI control scheme based on neural-network are performed. Performance comparisons between the proposed method and the conventional fixed gain PI control scheme are executed. The parameters of the PMSM used in simulation research are:

Tab.1. parameters of PMSM

Rated power $P_N$ (kW)	1
Rated voltage $U_N$ (V)	150
magnetic pole pairs $p$	4
Rated speed(r/min)	2000
Inertia (kg/m <sup>2</sup> )	0.0087
Stator resistance $R_s$ ( $\Omega$ )	0.53
d-axis stator inductance $L_d$ (mH)	2.25
q-axis stator inductance $L_q$ (mH)	2.25
Rotor flux linkage $\Psi_f$ (Wb)	0.175

The input speed of the PMSM servo system is  $\omega^* = 100$  rad/s. The simulation results of the PMSM servo system is in nominal condition and the inertia variation are shown in Fig. 4 and Fig.5. In Fig. 4 and Fig.5, dashed curve A and solid curve B represent the angular velocity step response curves of the PMSM servo system based on conventional fixed gain PI control scheme and proposed method.

From Fig.4, we can find that the PMSM servo system based on the proposed self-tuning PI control scheme has good static and dynamic performance and only little more slow than that based on traditional fixed gain PI control scheme when the system is in nominal condition. But from Fig.5, we can easily get that the response curves of the PMSM servo system based on traditional fixed gain PI control scheme has bigish overshoot when the inertia variation. Whereas, the response curves of the PMSM servo system based on proposed self-tuning PI control scheme not only hasn't overshoot but also has more quickly response speed than that based on traditional fixed gain PI control scheme when inertia variation.

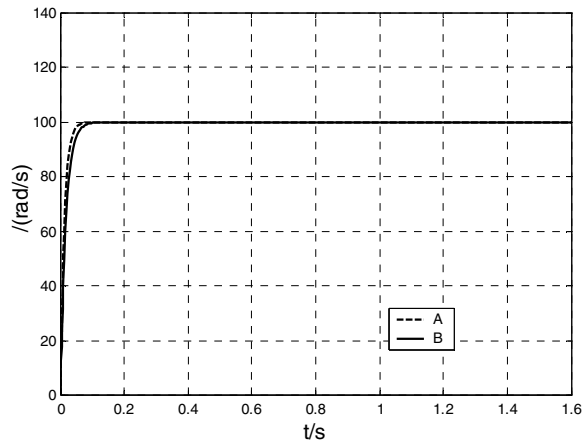


Fig.4 The angular velocity step response curves of PMSM servo system in nominal condition

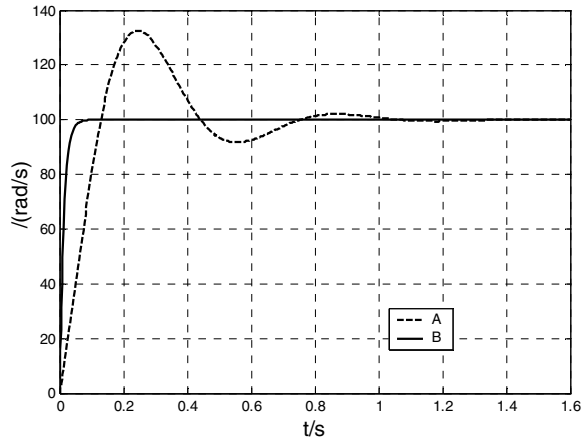


Fig.5 The angular velocity step response curve of PMSM servo system when  $J = 10J_0$

## 5. Conclusion

In this paper, the moment of rotational inertia is online identified based on model reference adaptive identification (MRAI) firstly. Then a well-trained three layer neural network supplies the PI controller with suitable gain according to each operating condition pair (identified inertia, angular velocity error, and angular velocity) detected. Result of simulations show that the proposed self-tuning PI control scheme outperforms the fixed PI scheme in rise time, precise track of angular velocity, and robustness when the inertia varied.

## 6. References

- [1]. Holtz Joachim, "Identification of the Machine Parameters in a Vector-Controlled Induction Motor Drive", *IEEE Transactions on Industry Applications*, vol.27, Dec./Nov. 1991, pp. 1111-1118
- [2]. K. Ohnishi, Y. Ueda, and K. Miyachi, "Model reference adaptive system against rotor resistance variation in induction motor drive," *IEEE Trans. Ind. Electron.*, vol. 4, Aug. 1986, pp. 217-223
- [3]. R. Marino, S. Peresada, and P.Valigi, "Adaptive input-output linearizing control of induction motors," *IEEE Trans. Automat. Contr.*, vol. 38, Feb. 1993, pp. 208-221.
- [4]. B. Grcar, P. Cafuta, M. Znidaric, and F. Gausch, "Nonlinear control of synchronous servo drive," *IEEE Trans. Contr. Syst. Technol.*, vol. 4, Mar. 1996, pp.177-184.
- [5]. Yujie GUO, Lipei HUANG, Yang QIU, "Inertia identification and auto-tuning of speed controller for AC servo system." *Tsinghua Univ (Sci & Tech)*, vol. 42, Sep. 2002, pp. 1180-1183.
- [6]. I. D. Landau, et al., "Adaptive Control", New York : Springer, 1998
- [7]. Gou-Jen Wang, Chuan-Tzueng Fong, and Kang J. Chang. "Neural-Network- Based self-tuning PI controller precise motion control of PMAC motors", *IEEE Ind. Electron*, vol.48, Feb. 2001, pp.408-415
- [8]. G. J. Wang and T. C. Chen. "A robust parameters self-tuning learning algorithm for multilayer feedforward neural network", *Neurocomputing*, Vol.25. 1999, pp. 167-189