

Article

Adaptive PIF Control for Permanent Magnet Synchronous Motors Based on GPC

Shaowu Lu, Xiaoqi Tang and Bao Song *

School of Mechanical Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China; E-Mails: guoting518@163.com (S.L.); xqtang@hust.edu.cn (X.T.)

* Author to whom correspondence should be addressed; E-Mail: songbao@hust.edu.cn; Tel.: +86-27-8754-4383; Fax: +86-27-8754-0024.

Received: 7 November 2012; in revised form: 10 December 2012 / Accepted: 18 December 2012 / Published: 24 December 2012

Abstract: To enhance the control performance of permanent magnet synchronous motors (PMSMs), a generalized predictive control (GPC)-based proportional integral feedforward (PIF) controller is proposed for the speed control system. In this new approach, firstly, based on the online identification of controlled model parameters, a simplified GPC law supplies the PIF controller with suitable control parameters according to the uncertainties in the operating conditions. Secondly, the speed reference curve for PMSMs is usually required to be continuous and continuously differentiable according to the general servo system design requirements, so the adaptation of the speed reference is discussed in details in this paper. Hence, the performance of the speed control system using a GPC-based PIF controller is improved for tracking some specified signals. The main motivation of this paper is the extension of GPC law to replace the traditional PI or PIF controllers in industrial applications. The efficacy and usefulness of the proposed controller are verified through experimental results.

Keywords: permanent magnet synchronous motor (PMSM); proportional integral feedforward (PIF) controller; speed reference; speed control system

1. Introduction

High-performance servo systems for permanent magnet synchronous motors (PMSMs) are essential in many applications such as precision engineering and industrial automation because of their

advantages of high efficiency, high power factor and high power density [1–3]. As the control core of the speed control system for a PMSM, the servo drive generally uses a fixed proportional integral (PI) controller to complete the adjustment process. Although the PI controller has the advantages of simple algorithms and high reliability, the control performance is closely related to its structure and control parameters. Nevertheless, as a one-degree-of-freedom (1Dof) controller, even if a traditional PI controller is tuned to be optimal, it is difficult for the servo system to satisfy good load-disturbance rejection and good transient response simultaneously. Recently, controllers with two degrees of freedom (2Dof) are being used more and more frequently [4–6]. These controllers are aimed at responding swiftly to speed command changes and offering enough robustness against the uncertainties of the servo system, but more controller parameters must be tuned for the additional flexibility. Therefore, in order to meet the development requirements of high speed and high precision, it is thus desirable to have an intelligent 2Dof controller that can self-tune its control parameters according to the operating condition uncertainties.

Self-tuning strategies for industrial applications have been also widely investigated. Roughly speaking, there are two lines of thought to self-tune the controller parameters so that the controllers can adapt to the varying conditions: model-based methods and rule-based methods [7]. Fuzzy logic theory and neural network (NN) techniques were developed to provide rule-based methods that incorporate self-tuning capability to implement nonlinear algorithms [8–10]. They are characterized by a series of linguistic or weight statements in the controller, but there are obstacles to these methods. The learning cost of these methods is considerably large, and these control parameters cannot be adequately adjusted due to the nonlinear properties [11]. For the simplicity and feasibility of self-tuning strategies, model-based methods appear to be more suitable for adoption in real-time self-tuning situations. Alfaro *et al.* [7] proposed a robust tuning method for a 2Dof PI controller based on the model reference optimization procedure, but its effectiveness depended on the accuracy of the model, and the method might not sound solid when the estimated model conditions were violated in real applications.

In order to solve the problem considering the accuracy of the controlled model, many PID controllers based on a generalized predictive control (GPC) law have been proposed to improve the performance of the control system [12,13]. A GPC law is a model-based controller not requiring extra compensators, and has been successfully implemented in many industrial applications [14–17]. An important reason for the growing number of applications is the fact that its linear model can be obtained by identification techniques. Another important reason is its control performance deterioration caused by the identification errors of the controlled model can be prevented [18]. Therefore, by using GPC to tune 2Dof PI controller parameters online in this paper, not only can the stability of the speed control process be guaranteed, but control parameters can also be tuned automatically according to the changing characteristics of the controlled model.

In this paper, a PI plus feedforward (PIF) controller, in the form of a commercially available 2Dof PI controller, is used to improve the regulatory control performance and the closed-loop control system robustness simultaneously. The main motivation of this paper is to further advance the application of GPC in adaptive control for PMSM systems. There are possibly two contributions in design of a GPC-based PIF controller.

First, the mainly disadvantage of GPC law is that the computational expense to obtain the solution of an optimization process is considerable at each sampling time [19–21]. To solve the problem,

a first-order controlled model is employed as the process object for the speed control system, and the parameters of this controlled model can be obtained directly by using a recursive least squares (RLS) algorithm with a low online computation burden. Moreover, Diophantine equations based on a first-order controlled model will be simplified obviously, so real-time performance of the GPC law can be guaranteed. Finally, the reference trajectory of a conventional GPC law was usually integrated as a smooth transition toward the desired trajectory or set point. However, in real industrial applications, this trajectory may not only cause a degradation of the closed-loop control performance, but increase the complexity of the GPC law for a PMSM system. In this paper, the trajectory will be ignored and a simplified cost function only requiring a future speed tracking error item and a penalization item on the torque current command is used.

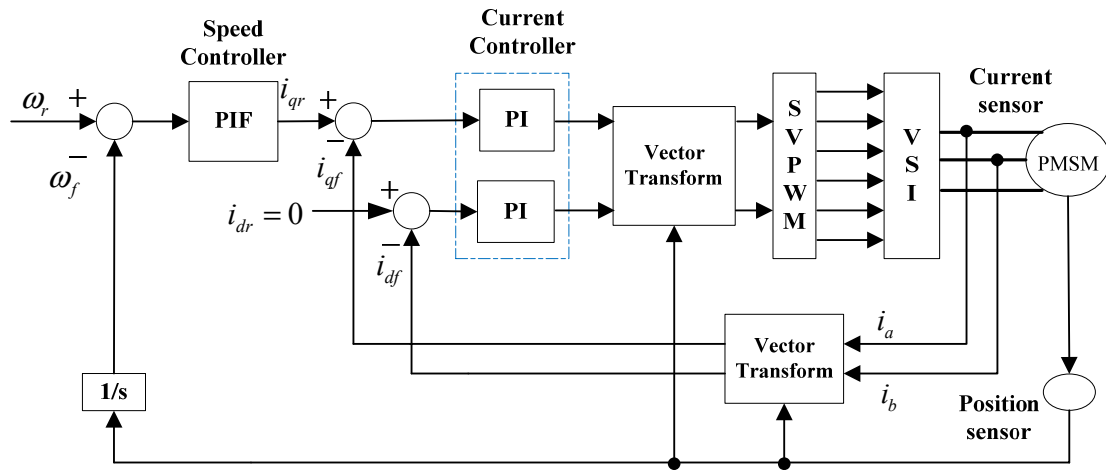
Second, conventional GPC-based PID controllers use the future reference trajectory to obtain control performance as good as that of the GPC law, and have been designed to follow a step-type or a ramp-type reference command [18,22]. Frequent step-type speed commands for a PMSM will severely impact the electrical and mechanical agencies of the servo system, and a ramp-type command is discontinuous and just a special application, but the speed command for a commercial servo drive is usually required to be continuous and at least first-order differentiable. Meanwhile, due to the structural differences between a PIF controller and a PID controller, the coefficients of the GPC law need to be rearranged and keep in consonance with those of the PIF controller. In this paper, a GPC law can be straightforwardly replaced by a PIF controller, and the GPC-based PIF controller will be designed to follow many general reference curves for speed commands. However, it is noted that an arbitrary reference command does not necessarily satisfy the design requirements for the GPC-based PIF controller.

The paper is organized as follows: in Section 2, the system model of the PMSM is built in detail. In Section 3, the model parameters of a dynamic process object for a PMSM will be obtained in real time by using an RLS method, then, based on the controlled model and the future speed reference, a GPC law will supply a PIF controller with the suitable controller parameters to ensure good control performance. Experimental results are presented in Section 4 and conclusions are drawn in the final section.

2. System Model of PMSM

2.1. Speed Control System of PMSM

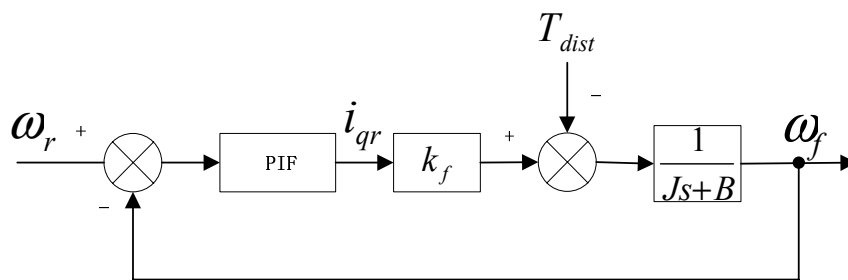
Under perfect field orientation and sensing technique conditions, the complicated coupled nonlinear dynamic performance of a PMSM can be significantly improved, whereby torque and flux can be tuned separately by two closed loops [23,24]. Thus accurate current feedback and position feedback are important factors in the realization of a high-precision and highly responsive closed control system [25]. A system configuration of a vector-controlled PMSM servo system is shown in Figure 1. In this vector control scheme, three-phase current is often detected by using a Hall current sensor, and position information is often obtained by using a high-resolution optical encoder.

Figure 1. Vector-controlled PMSM servo system.

In the high-performance speed control system, a considerable high-bandwidth current loop is designed to ensure accurate current tracking and act as a current source amplifier within the current loop bandwidth [26]. Assuming the perfect current tracking and ignoring dead time in this paper, an average controlled model of a PMSM is shown in Figure 2. Accordingly, the mechanical dynamics can be reasonably described as:

$$\omega_f(s) = \frac{k_f i_{qr}(s) - T_{dist}(s)}{Js + B} \quad (1)$$

where ω_f is feedback speed; i_{qr} is expected torque current; k_f is the torque coefficient related with flux linkage of PMSM; J is the system inertia including the load inertia; T_{dist} is the external disturbances; B is the frictional torque coefficient.

Figure 2. Average model of PMSM.

The uncertainties of the servo system mainly include the load inertia and load torque. In the running of a servo system, system inertia may change. When the system inertia increases, the response of the servo system will speed up with system overshoot, which is likely to cause system instability. On the contrary, when the system inertia decreases, dynamic response will occur with system oscillation as well as turbulence. Meanwhile, the main role of the servo system is to drive the load operation, but in many industries, the load carried by servo system is not constant. Changes in the load torque will have a significant impact on servo control performance: in the running of a servo system, the sudden increase or reduce of load torque would result in fluctuations in servo speed control, affecting the accuracy of positioning and control performance.

Aside from the above reasons, the true controlled model may be more complex and of high-order since the current sensor and position sensor are nonlinear systems [27]. Actually, in order to better apply a GPC law to servo control, the effect of uncertainties is considered as the variation of model parameters. So the dynamic model of PMSM is rewritten as:

$$\omega_f(s) = \frac{k_f i_{qr}(s)}{Js + B} \quad (2)$$

2.2. Discrete-Time Model and Controller

Based on Equation (2), with the zero-order-hold (ZOH) conversion, the mathematical expression of controlled model can be described with the following discrete difference equation:

$$\begin{cases} A(z^{-1})\omega_f(k) = B(z^{-1})i_{qr}(k-1) + \xi(k)/\Delta \\ A(z^{-1}) = 1 - e^{-T_s B/J} z^{-1} = 1 + a_1 z^{-1} \\ B(z^{-1}) = k_f(1 - e^{-T_s B/J})/B = b_0 \end{cases} \quad (3)$$

where T_s is the sampling time of servo system; a_1 and b_0 are the estimated parameters; $\xi(k)$ is white noise; Δ is the differential operator, $\Delta = 1 - z^{-1}$.

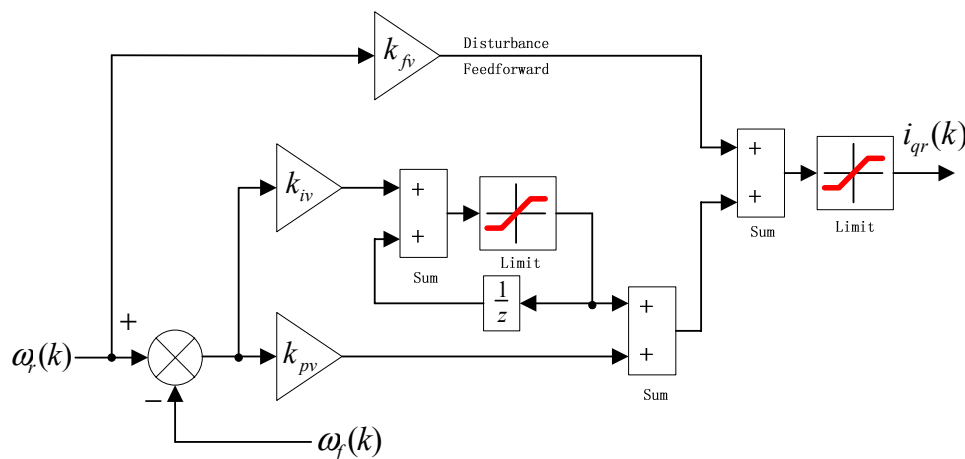
The control structure of PIF controller is shown in Figure 3. In order to use GPC to tune the control parameters, the PIF controller can be described as:

$$\Delta i_{qr}(k) = (k_{pv}\Delta + k_{iv} + k_{fv}\Delta)\omega_r(k) - (k_{pv}\Delta + k_{iv})\omega_f(k) \quad (4)$$

$$i_{qr}(k) = i_{qr}(k-1) + \Delta i_{qr}(k) \quad (5)$$

where k_{pv} is velocity loop proportional gain; k_{iv} is velocity loop integral gain; k_{fv} is velocity loop feedforward gain. i_{qr} usually need to be limited, and Δi_{qr} is used for GPC to replace PIF controller.

Figure 3. The structure of PIF controller.

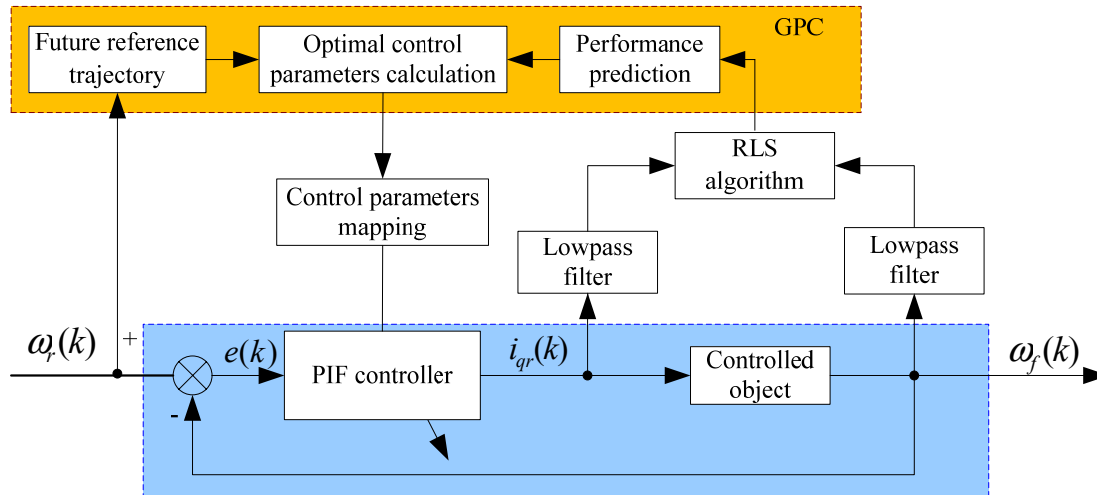


3. The design of Adaptive PIF Controller

The basic function of the GPC law is to calculate a sequence of future control signals in such a way that it minimizes a cost function defined over a prediction horizon [28,29]. In this paper, the GPC-based PIF controller mainly includes performance prediction, future reference trajectory and control

parameters calculation. The structure of a GPC-based PIF controller is presented in Figure 4. Firstly, torque current and actual speed are sampled for obtaining a controlled model in every control period, and a recursive least squares (RLS) algorithm is used to update the model parameters. Then, based on the controlled model and future reference trajectory, the predicted output can be obtained and the calculation of future control values can be achieved. Finally, future control values will supply the PIF controller with suitable control parameters online.

Figure 4. The principle diagram of a PIF control-based GPC.



3.1. The Identification of Model Parameters

In order to obtain predictions of the speed control performance, continuous recognition should be applied to the model parameters of the controlled object. The model parameters can be generally obtained by a RLS algorithm, so the parameters in the polynomial of $A(z^{-1})$ and $B(z^{-1})$ in Equation (3) are calculated using the following recursive least squares method [30]:

$$\hat{\theta}(k) = \hat{\theta}(k-1) + K(k)[\omega_f(k) - \hat{\theta}(k-1)\varphi(k-1)] \quad (6)$$

$$K(k) = \frac{X(k-1)\varphi(k-1)}{\alpha^* + \varphi^T(k-1)X(k-1)\varphi(k-1)} \quad (7)$$

$$X(k) = \frac{1}{\alpha^*} \left[X(k-1) - \frac{X(k-1)\varphi^T(k-1)X(k-1)\varphi(k-1)}{\alpha^* + \varphi^T(k-1)X(k-1)\varphi(k-1)} \right] \quad (8)$$

$$\begin{cases} \hat{\theta}(k) = [\hat{a}_1(k), \hat{b}_0(k)] \\ \varphi(k-1) = [-\omega_f(k-1), i_{qr}(k-1)] \end{cases} \quad (9)$$

where α^* is the forgetting factor; $X(0) = \alpha_I I$ ($0 < \alpha_I < \infty$).

3.2. Performance Prediction

To simplify the GPC law, the trajectory will be ignored, and a cost function of the speed control system is designed by using output error and quadratic indicators form of weighted control increments:

$$J = \sum_{j=N_1}^{N_2} [\omega_f(k+j) - \omega_r(k+j)]^2 + \lambda \sum_{j=1}^{N_u} \Delta i_{qr}(k+j-1)^2 \quad (10)$$

where N_1 is minimum predictive horizon, usually choose $N_1 = 1$; N_2 is maximum predictive horizon; N_u is control horizon, generally $N_u \leq N_2$; λ is the control increment weighting factor.

Because $B(z^{-1})$ is zero order polynomial, simplified Diophantine Equations (11) and (12) are solved to obtain $\omega_f(k+j)$:

$$1 = E_j(z^{-1})A(z^{-1})\Delta + z^{-j}F_j(z^{-1}) \quad (11)$$

$$E_j(z^{-1})B(z^{-1}) = G_j(z^{-1}) \quad (12)$$

$$\begin{cases} E_j(z^{-1}) = e_0 + e_1 z^{-1} + \dots + e_j z^{-j+1} \\ F_j(z^{-1}) = f_0^j + f_1^j z^{-1} \\ G_j(z^{-1}) = g_0 + g_1 z^{-1} + \dots + g_j z^{-j+1} \end{cases} \quad (13)$$

Then the j steps ahead predictive output is obtained by following equation:

$$\omega_f(k+j) = G_j \Delta i_{qr}(k+j-1) + F_j \omega_f(k) + E_j \xi(k+j) \quad (14)$$

Obviously, in the right hand part of Equation (14), the first two terms and the third are irrelevant. If the first two are regarded as the optimal prediction, then the third is the prediction error, that is, the optimal prediction controlled object can be expressed as:

$$\hat{\omega}_f(k+j) = G_j(z^{-1})\Delta i_{qr}(k+j-1) + F_j(z^{-1})\omega_f(k) \quad (15)$$

The vector form of Equation (14) is expressed as:

$$\boldsymbol{\omega}_f = \mathbf{G} \mathbf{i}_{qr} + \mathbf{F} \omega_f(k) + \mathbf{E} \quad (16)$$

$$\boldsymbol{\omega}_f^T = [\omega_f(k+1), \dots, \omega_f(k+N_2)] \quad (17)$$

$$\mathbf{i}_{qr}^T = [\Delta i_{qr}(k), \dots, \Delta i_{qr}(k+N_u-1)] \quad (18)$$

$$\mathbf{F}^T = [F_1(z^{-1}), \dots, F_{N_2}(z^{-1})] \quad (19)$$

$$\mathbf{E}^T = [E_1 \xi(k+1), \dots, E_{N_2} \xi(k+N_2)] \quad (20)$$

$$\mathbf{G} = \begin{bmatrix} g_0 & & & \\ g_1 & g_0 & & \\ \vdots & & & \\ g_{N_u-1} & g_{N_u-1} & \cdots & g_0 \\ \vdots & & & \\ g_{N_2-1} & g_{N_2-1} & \cdots & g_{N_2-N_u} \end{bmatrix} \quad (21)$$

The vector form of the cost Function in Equation (10) can be obtained:

$$J = (\boldsymbol{\omega}_f - \boldsymbol{\omega})^T (\boldsymbol{\omega}_f - \boldsymbol{\omega}) + \lambda \mathbf{i}_{qr}^T \mathbf{i}_{qr} \quad (22)$$

$$\boldsymbol{\omega}^T = [\omega_r(k+1), \dots, \omega_r(k+N_2)] \quad (23)$$

Since Equation (22) is quadratic in i_{qr} , the optimal solution with respect to i_{qr} is obtained by:

$$\mathbf{i}_{qr} = (\mathbf{G}^T \mathbf{G} + \lambda I)^{-1} \mathbf{G}^T [\boldsymbol{\omega} - \mathbf{F} \boldsymbol{\omega}_f] \quad (24)$$

The use of the first element of i_{qr} gives the following control law:

$$\Delta i_{qr}(k) = P(z^{-1})\omega_r(k + N_2) - F(z^{-1})\omega_f(k) \quad (25)$$

$$\begin{cases} [p_{N_1} \cdots p_{N_2}] = [1 \ 0 \cdots 0](\mathbf{G}^T \mathbf{G} + \lambda I)^{-1} \mathbf{G}^T \\ P(z^{-1}) = p_{N_2} + p_{N_2-1}z^{-1} + \cdots + p_{N_1}z^{-(N_2-N_1)} \\ F(z^{-1}) = p_{N_1}F_{N_1}(z^{-1}) + p_{N_1+1}F_{N_1+1}(z^{-1}) + \cdots + p_{N_2}F_{N_2}(z^{-1}) = f_0 + f_1z^{-1} \end{cases} \quad (26)$$

3.3. Future Reference Trajectory

Remark 1: Compared with the increment control output in [18,22], Equation (25) is more simple, only requiring a future speed command item and a feedback speed item. However, the future speed command item cannot be straightforwardly achieved in the PIF controller. To apply the GPC law to the PIF controller, assuming the speed command curve for PMSM is arbitrary-order continuously differentiable at some points, the future reference trajectory can be expressed by the following equation by using Taylor series expansion:

$$\omega_r(k + j) = \omega_r(k) + g(k, j)\Delta\omega_r(k) \quad (27)$$

where $g(k, j)$ is related to k and j . Detailed expressions of Equation (27) are given in Appendix A.

Remark 2: If the speed command cannot be expressed as Equation (27), the polynomial of GPC will not match with the polynomial of the PIF controller, and the GPC law will not supply the PIF controller with the suitable control parameters. When a PMSM runs at constant velocity, the servo drive is required to set the rise time and the drop time. Therefore, the rise stage and the drop stage of speed command are usually set as a continuously differentiable curve, e.g., trapezium-type or exponent-type curve, in this case, $g(j)$ is only related to j and Equation (27) can be simplified as:

$$\omega_r(k + j) = \omega_r(k) + g(j)\Delta\omega_r(k) \quad (28)$$

3.4. Optimal Control Parameters Calculating

Using Equation (28), the first term on the right hand side of Equation (25) is rewritten as:

$$\begin{cases} P(z^{-1})\omega_r(k + N_2) = p_r\omega_r(k) + p_s\Delta\omega_r(k) \\ p_r = \sum_{j=N_1}^{N_2} p_j \\ p_s = \sum_{j=N_1}^{N_2} p_j g(j) \end{cases} \quad (29)$$

Using Equation (29), the GPC law Equation (25) is rewritten as:

$$\Delta i_{qr}(k) = (p_r + p_s\Delta)\omega_r(k) - (f_0 + f_1 - f_1\Delta)\omega_f(k) \quad (30)$$

Based on $f_0^j + f_1^j = 1$, so the following equations can be obtained:

$$\begin{cases} f_0 + f_1 = \sum_{j=N_1}^{N_2} p_j \\ p_r = f_0 + f_1 \end{cases} \quad (31)$$

Then comparing Equation (30) with Equation (4), the PIF controller gain parameters are calculated by the GPC law:

$$\begin{cases} k_{pv} = -f_1 \\ k_{rv} = f_0 + f_1 \\ k_{fv} = p_s + f_1 \end{cases} \quad (32)$$

With the above discussion, the implementation of a GPC-based PIF controller design is summarized as follows:

Step 1: Given the related parameters for an RLS law and a GPC algorithm.

Step 2: The controlled model parameters a_1 and b_0 are updated online by the RLS law.

Step 3: Based on the controlled model, Diophantine Equations (11) and (12) are solved to obtain $\omega_r(k+j)$ and $\Delta i_{qr}(k)$ is obtained by Equation (25).

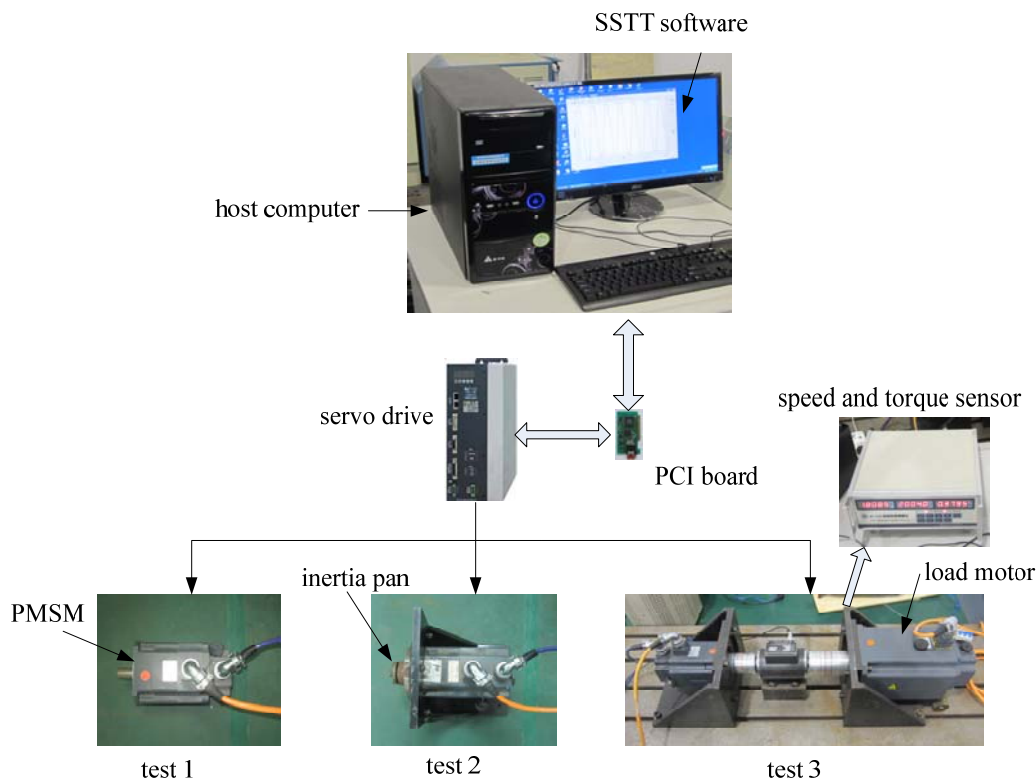
Step 4: Speed command can be expressed as Equation (28), then, the PIF controller gain parameters can be tuned by Equation (32).

Step 5: Set $k = k + 1$ and go to step 2.

4. Experiment

The apparatus for the experimental platform contains three major parts and some data transferring buses, as shown in Figure 5. These three major parts are: (1) PC-based PCI motion controller with sampling time equal to 5 ms; (2) AC servo drive with position, velocity and torque control modes using a DSP plus a FPGA, where DSP TMS320F2812 mainly accomplishes in real-time control algorithms during motion control, and FPGA EP2C8Q208C8N is responsible for the analysis and realization of absolute ruler and NCUC-Bus protocols; (3) PMSM with the parameters described in Table 1, increment encoder with 2,500 pulse/revolution is fixed in the PMSM. Through the PCI controller, the PC sends the speed command and control parameters to the servo drive, receives expected torque current and feedback velocity from the servo drive for the identification of model parameters, and implements the RLS algorithm and proposed GPC-based PIF design every 5 ms by a commercial Servo Self-adapted Turning Tool (SSTT) software based on the VC environment.

In the experimental tests, the maximum value of the speed command is set to 1,200 r/min, its rise time and drop time are set to 1 s. Finally, all experimental data from the SSTT are plotted by using Matlab 6.5. The related parameters for the RLS algorithm and the GPC law are: $N_1 = 1$, $N_2 = 10$, $N_u = 2$, $\lambda = 0.01$, and $\alpha^* = 0.94$. To evaluate the control performance and confirm the effectiveness of the proposed strategy, three different tests, which all consider a traditional PI controller and a traditional PIF controller, have been taken on the experiment platform.

Figure 5. The apparatus for the experimental platform.**Table 1.** Specification of the PMSM.

Name	Values
Phase current	3.7 A
The number of poles	3
Rated torque	4.5 Nm
Rotor inertia	0.00067 kgm ²
Rated speed	1,200 r/min

4.1. Test 1 without Applied Load

Figures 6 and 7 show the speed response and speed error for the traditional PI controller, traditional PIF controller and GPC-based PIF controller. It is clear that traditional PI controller and traditional PIF controller do not meet the precision control requirement, and a large speed error exists in the whole control process, especially in the rise stage and the drop stage. Meanwhile, the performance of the speed response based on the adaptive PIF controller is obviously improved in the whole process, except in the initial stage, because model parameters are not completely convergent. PIF controller gains are shown in Figure 8, because more torque is required in the rise stage and the drop stage, controller gains of the rise stage and the drop stage is larger than those of the steady stage. The identification result for model parameters using the RLS algorithm is shown in Figure 9, it can be seen that model parameters in uniform velocity stage are relatively stable without applied load.

Figure 6. Speed response (test 1).

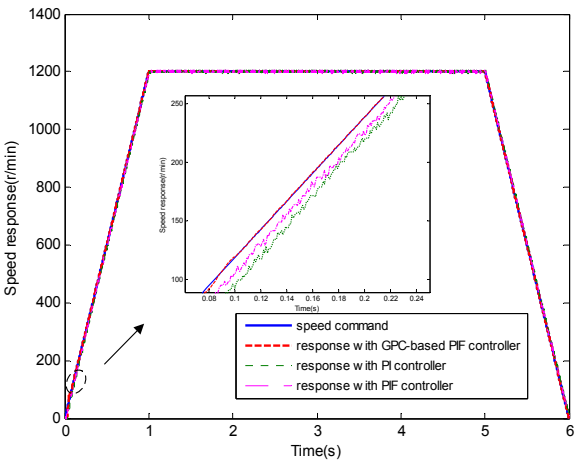


Figure 7. Speed error (test 1).

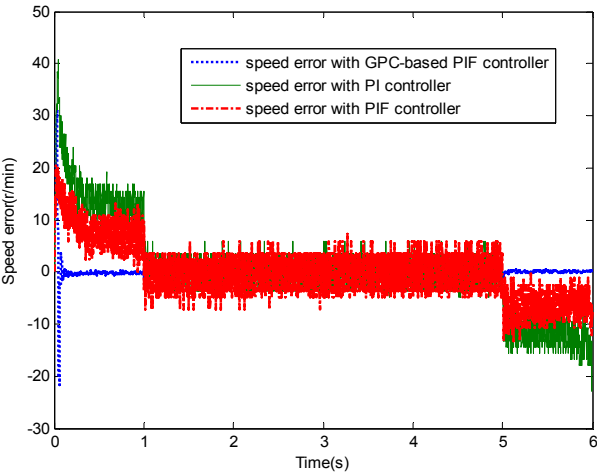


Figure 8. Control parameters (test 1).

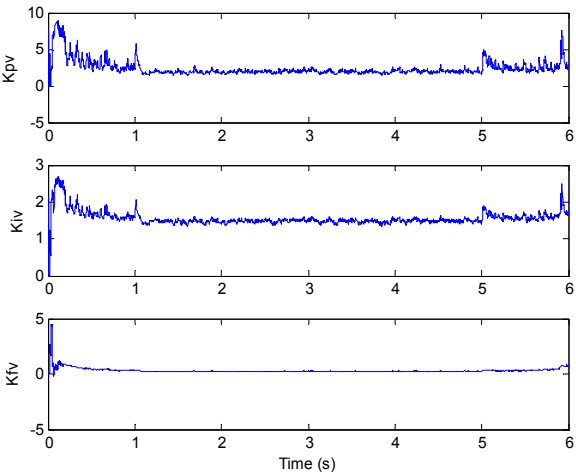
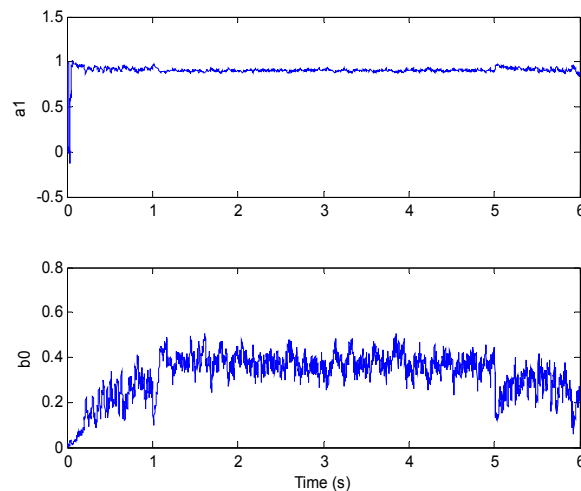


Figure 9. Model parameters (test 1).

4.2. Test 2 with Applied Load Inertia

In this test, the load inertia is a triple rotor inertia, and is applied to confirm the robustness and the effectiveness of the proposed controller. It is noted that the control gains of traditional PI controller and traditional PIF controller are the same as in test 1. Figures 10 and 11 show the speed response and speed error for a traditional PI controller, traditional PIF controller and GPC-based PIF controller. It is clear that the traditional PI controller and traditional PIF controller both lead to large speed errors and system overshoot in the control process. Meanwhile, the adaptive PIF controller can continue to maintain fairly good control performance in the whole control process, but the initial model mismatch still exists. GPC-based PIF controller gains and model parameters are shown in Figures 12 and 13, because system inertia becomes larger and leads to speed overshoot in the transition point between the rise stage and the steady stage, controller gains of the transition stage are required to become larger to overcome the larger inertia.

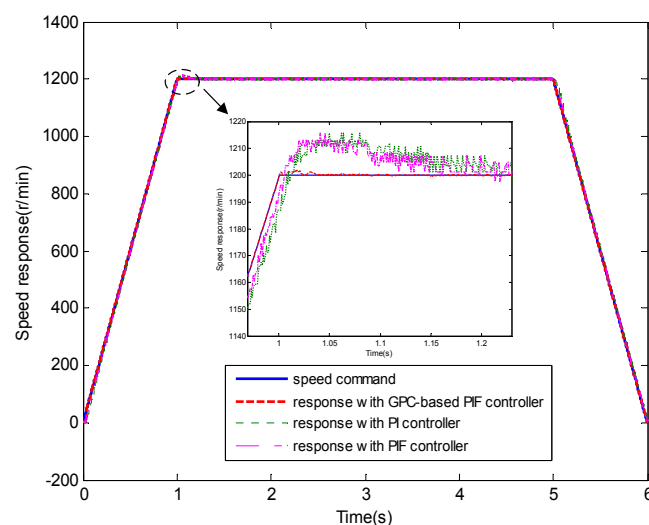
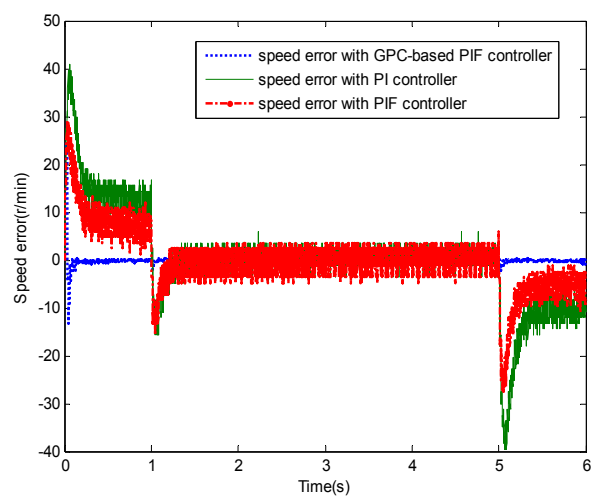
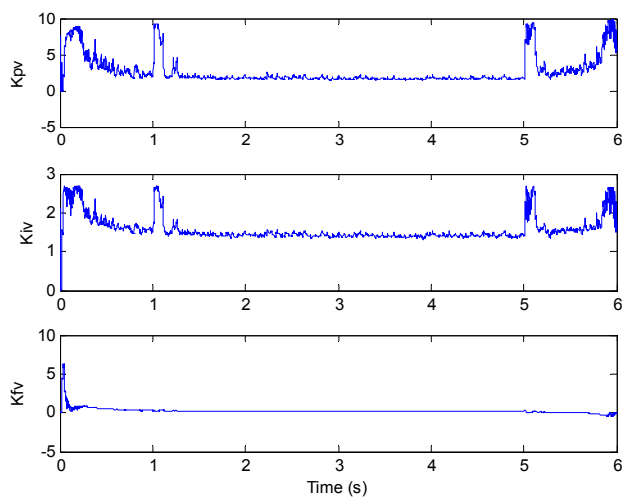
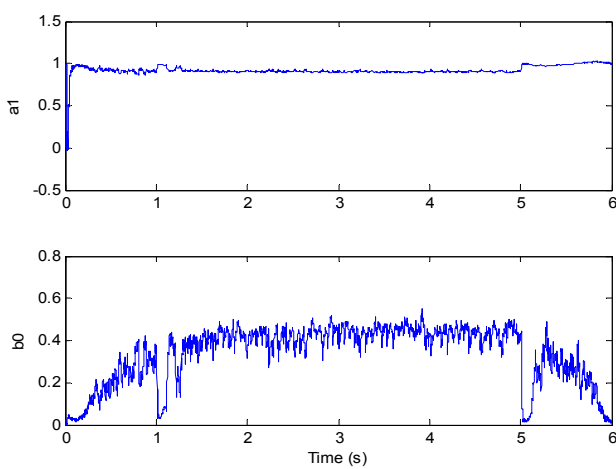
Figure 10. Speed response (test 2).

Figure 11. Speed error (test 2).**Figure 12.** Control parameters (test 2).**Figure 13.** Model parameters (test 2).

4.3. Test 3 with Applied Load Torque

In this test, load torque is applied to confirm the robustness and the effectiveness of the proposed controller. The initial load torque is set to 2 Nm, and became to be 9 Nm when $t = 3$ s. It is noted that the control gains of the traditional PI controller and traditional PIF controller are still the same as in test 1. Figures 14 and 15 show the speed response and speed error for the three controllers. When $0 < t \leq 3$ s, it is clear that traditional PI controller and traditional PIF controller both lead to larger speed errors and system overshoot than those in test 2. When $t \geq 3$ s, because of the variation of load torque, the actual speed drops and begins to keep speed tracking after 1 s. Meanwhile, the adaptive PIF controller can continue to maintain fairly good control performance in the whole control process especially when $t \geq 3$ s. The initial model mismatch still exists, but the control performance is superior for the adaptive PIF controller. PIF controller gains and model parameters are shown in Figures 16 and 17, because load torque becomes larger and makes servo speed drops when $t \geq 3$ s, controller gains of the transition stage is required to become larger to overcome the load torque.

Figure 14. Speed response (test 3).

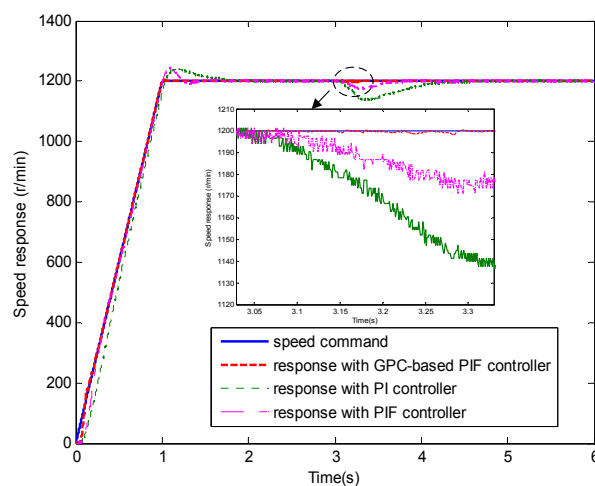


Figure 15. Speed error (test 3).

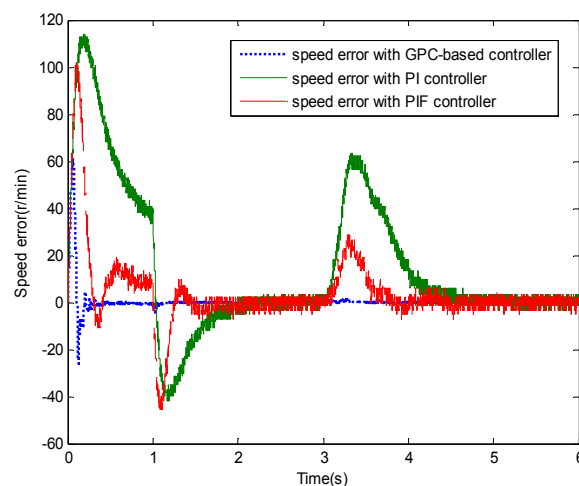
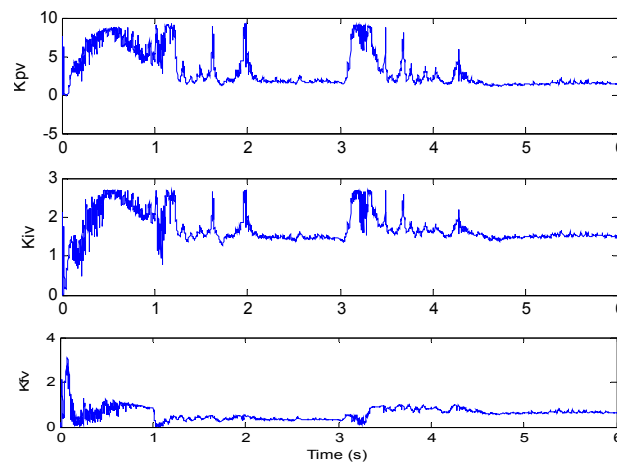
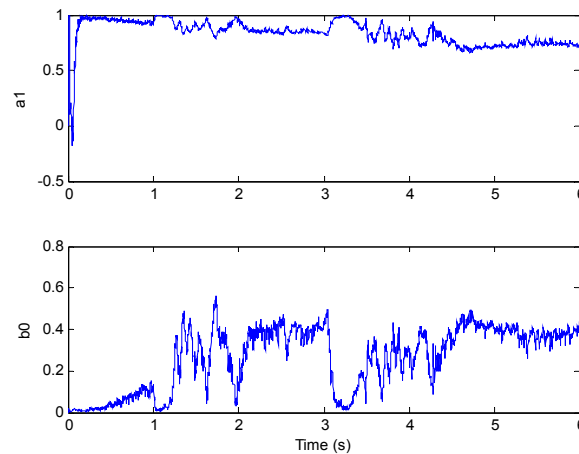


Figure 16. Control parameters (test 3).**Figure 17.** Model parameters (test 3).

5. Conclusions

The speed control systems of the PMSMs employed in various industries are almost always controlled by traditional PI and PIF controllers. In order to meet the development requirements of high performance servo systems, this paper has proposed a new GPC-based PIF controller to increase the robustness of servo systems. In this method, controlled model parameters can be obtained online by an RLS algorithm. Based on the controlled model, a high performance and simplified GPC supplies the PIF controller with suitable control parameters according to the uncertainties in the operating conditions. The experimental results show a good control performance and strong robustness of the speed control system can be maintained.

Acknowledgments

The work is supported by the National Natural Science Foundation of China (NSFC) (Grant NO. 50905069), the National Science and Technology Major Project (Grant NO. 2012ZX04001012), and the National Science and Technology Major Project (Grant NO. 2012ZX04001022).

References

1. Cheng, S.; Li, C.-W. Fuzzy PDFFF-IIR controller for PMSM drive systems. *Cont. Eng. Pract.* **2011**, *19*, 828–835.
2. Errouissi, R.; Ouhrouche, M.; Chen, W.H.; Trzynadlowski, A.M. Robust nonlinear predictive controller for permanent-magnet synchronous motors with an optimized cost function. *IEEE Trans. Indust. Electron.* **2012**, *59*, 2849–2858.
3. Choi, H.H.; Vu, N.T.-T.; Jung, J.-W. Digital implementation of an adaptive speed regulator for a PMSM. *IEEE Trans. Power Electron.* **2011**, *26*, 3–8.
4. Claveau, F.; Chevrel, P.; Knittel, K. A 2DOF gain-scheduled controller design methodology for a multi-motor web transport system. *Contr. Eng. Pract.* **2008**, *16*, 609–622.
5. Sanchez, J.; Visioli, A.; Dormido, S. A two-degree-of-freedom PI controller based on events. *J. Process Contr.* **2011**, *21*, 639–651.
6. Krebs, G.; Tounzi, A.; Pauwels, B.; Willemot, D. Modeling and control of a 2Dof PM synchronous actuator using Hall effect sensors. *Mechatronics* **2010**, *20*, 153–161.
7. Alfaro, V.M.; Vilanova, R. Model-reference robust tuning of 2Dof PI controllers for first- and second-order plus dead-time controlled processes. *J. Process Contr.* **2012**, *22*, 359–374.
8. Pan, I.; Das, S.; Gupta, A. Tuning of an optimal fuzzy PID controller with stochastic algorithms for networked control systems with random time delay. *ISA Trans.* **2011**, *50*, 28–36.
9. Mohan, B.M.; Sinha, A. Mathematical models of the simplest fuzzy PI/PD controllers with skewed input and output fuzzy sets. *ISA Trans.* **2008**, *47*, 300–310.
10. Zeng, S.W.; Hu, H.G.; Xu, L.H.; Li, G.H. Nonlinear adaptive PID control for greenhouse environment based on RBF network. *Sensors* **2012**, *16*, 365–375.
11. Yamamoto, T.; Takao, K.; Yamada, T. Design of a data-driven PID controller. *IEEE Tran. Contr. Syst. Technol.* **2009**, *17*, 29–39.
12. Asano, M.; Yamamoto, T.; Oki, T.; Kaneda, M. A Design of neural-net based predictive PID controllers. *Proc. IEEE Conf. Syst. Man Cyber.* **1999**, *4*, 1113–1118.
13. Miller, R.M.; Shah, S.L.; Wood, R.K.; Kwok, E.K. Predictive PID. *ISA Trans.* **1999**, *38*, 11–23.
14. Cortes, P.; Kazmierkowski, M.P.; Kennel, R.M.; Quevedo, D.E.; Rodriguez, J. Predictive control in power electronics and drives. *IEEE Trans. Indust. Electron.* **2008**, *55*, 4312–4324.
15. Beerten, J.; Vervekken, J.; Driesen, J. Predictive direct torque control for flux and torque ripple reduction. *IEEE Trans. Indust. Electron.* **2010**, *57*, 404–412.
16. Nemec, M.; Nedeljkovic, D.; Ambrozic, V. Predictive torque control of induction machines using immediate flux control. *IEEE Trans. Indust. Electron.* **2007**, *54*, 2009–2017.
17. Miranda, H.; Cortes, P.; Yuz, J.I.; Rodriguez, J. Predictive torque control of induction machines based on state-space models. *IEEE Trans. Indust. Electron.* **2009**, *56*, 1916–1924.
18. Sato, T. Design of a GPC-based PID controller for controlling a weigh feeder. *Control Eng. Pract.* **2010**, *18*, 105–113.
19. Papafotiou, G.; Kley, J.; Papadopoulos, K.G.; Bohren, P.; Morari, M. Model predictive direct torque control-part II: Implementation and experimental evaluation. *IEEE Trans. Indust. Electron.* **2009**, *56*, 1906–1915.

20. Bolognani, S.; Bolognani, S.; Peretti, L.; Zigliotto, M. Design and implementation of model predictive control for electrical motor drives. *IEEE Trans. Indust. Electron.* **2009**, *56*, 1925–1936.
21. Liu, H.; Li, S. Speed control for PMSM servo system using predictive functional control and extended state observer. *IEEE Trans. Indust. Electron.* **2012**, *59*, 1171–1183.
22. Sato, T.; Inoue, A.; Yamamoto, T. Improvement of tracking performance in designing a GPC-based PID controller using a time-varying proportional gain. *Int. J. Innov. Comput. Inform. Contr.* **2006**, *2*, 491–503.
23. Errouissi, R.; Ouhrouche, M.; Chen, W.H.; Trzynadlowski, A.M. Robust cascaded nonlinear predictive control of a permanent magnet synchronous motor with an antiwindup compensator. *IEEE Trans. Indust. Electron.* **2012**, *59*, 3078–3088.
24. Gamazo-Real, J.C.; Vazquez-Sanchez, E.; Gomez-Gil, J. Position and speed control of brushless DC motors using sensorless techniques and application trends. *Sensors* **2010**, *10*, 6901–6947.
25. Gaeid, K.S.; Ping, H.W.; Khalid, M. Sensor and sensorless fault tolerant control for induction motors using a wavelet index. *Sensors* **2012**, *12*, 4031–4050.
26. Yasser, A.-R.; Mohamed, I. Adaptive self-tuning speed control for permanent-magnet synchronous motor drive with dead time. *IEEE Trans. Energy Convers.* **2006**, *21*, 855–862.
27. Kuo, C.-F.J.; Hsu, C.-H. Precise speed control of a permanent magnet synchronous motor. *Int. J. Adv. Manufact. Technol.* **2006**, *28*, 942–949.
28. Correa, P.; Pacas, M.; Rodriguez, J. Predictive torque control for inverter fed induction machines. *IEEE Trans. Indust. Electron.* **2007**, *54*, 1073–1079.
29. Morel, F.; Lin-Shi, X.F.; Retif, J.M.; Allard, B.; Buttay, C. A comparative study of predictive current control schemes for a permanent-magnet synchronous machine drive. *IEEE Trans. Indust. Electron.* **2009**, *56*, 2715–2728.
30. Wang, D.Q.; Ding F. Input-output data filtering based recursive least squares identification for CARARMA systems. *Digit. Signal Process.* **2010**, *20*, 991–999.

Appendix A

Assuming the speed command curve for a PMSM is arbitrary-order continuously differentiable at $k = 0$, Taylor series expansion can be expressed as following equation:

$$\omega_r(k) = \sum_{n=0}^{\infty} \frac{\omega_r^{(n)}(0)}{n!} k^n \quad (\text{A.1})$$

So, Taylor series expansions of $\omega_r(k+j)$ and $\Delta\omega_r(k)$ can be obtained:

$$\omega_r(k+j) = \sum_{n=0}^{\infty} \frac{\omega_r^{(n)}(0)}{n!} (k+j)^n \quad (\text{A.2})$$

$$\Delta\omega_r(k) = \omega_r(k) - \omega_r(k-1) = \sum_{n=0}^{\infty} \frac{\omega_r^{(n)}(0)}{n!} [k^n - (k-1)^n] \quad (\text{A.3})$$

Substituting Equations (A.1), (A.2) and (A.3) into Equation (27), $g(k,j)$ can be obtained:

$$g(k, j) = \frac{\sum_{n=0}^{\infty} \frac{\omega_r^{(n)}(0)}{n!} [(k+j)^n - k^n]}{\sum_{n=0}^{\infty} \frac{\omega_r^{(n)}(0)}{n!} [k^n - (k-1)^n]} \quad (\text{A.4})$$

For example, a three-order polynomial function is chosen as the rise stage curve for speed command and can be expressed as following equation:

$$\omega_r(k) = ck^3 \quad (\text{A.5})$$

where, c is a constant. So, substituting Equation (A.5) into Equation (A.4), the corresponding $g(k, j)$ can be obtained:

$$g(k, j) = \frac{(k+j)^3 - k^3}{k^3 - (k-1)^3} \quad (\text{A.6})$$

© 2013 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).