Electric Power Systems Research 119 (2015) 293–303

Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr

# A self-tuning PI controller for the speed control of electrical motor drives

Mohamed S. Zaky*

*Electrical Engineering Department, Faculty of Engineering, Minoufiya University, Shebin El-Kom (32511), Minoufiya, Egypt*

ARTICLE INFO

*Article history:*
Received 4 April 2014
Received in revised form 6 September 2014
Accepted 5 October 2014
Available online 4 November 2014

*Keywords:*
Self-tuning PI controller
DSP-based control
Hybrid stepper motor
Field oriented control
Adaptive control
Speed control

ABSTRACT

This paper proposes a self-tuning PI controller for the speed control of electrical motor drives. The proposed PI controller gains are adjustable parameters and will be updated online depending on the speed error. They are designed to vary within a pre-determined range to eliminate the problems faced by the conventional fixed PI controller. The performance of the proposed self-tuning PI controller using a field oriented control (FOC) of hybrid stepper motor (HSM) is simulated and compared with the conventional PI controller under tracking performance, parameters variation and load torque disturbances. The proposed controller is implemented with DSP-DS1102 control board and evaluated through laboratory experiments for the HSM drive. Simulation and experimental results show the effectiveness of the proposed approach. It is very simple and has given improved results when compared with previous approaches.



## 1. Introduction

The control of stepper motors has attracted much attention over the past few years, due to the developments in control theory and the availability of low-cost digital hardware [1–4]. The hybrid stepper motors (HSMs) have been widely known for their high efficiency and torque capability as compared to other stepper motor types [5]. HSMs are generally operated in open loop due to their special structure. They are mainly used for simple point-to-point positioning tasks in which they were open loop controlled, and no information on the motor shaft position or speed was used [6].

In the open loop control, there is no feedback of load position to the controller, however the motor must respond to each excitation change. This introduces large overshoot, resonance and torque ripple problems which degrade the dynamic performance of HSMs. Besides, if fast excitation changes are applied, the stepper motor can lose steps and therefore, it fails to move the rotor to the new demanded position quickly and accurately. This would result a permanent error between the actual load position and the required position and, consequently, lose its stability and synchronization [6,7]. Therefore, open loop control of HSMs is insufficient, so closed-loop control is essentially required for high performance applications under parameters variation and load torque disturbances.

The closed-loop principle has been introduced to stepper motors in order to increase positioning accuracy and reduce their sensitivity to load disturbances. Nowadays, stepper motors are more often closed-loop controlled, in particular, for machine tools and robotic manipulators. They have used in applications which require to perform high-precision operation in spite of mechanical configuration changes such as load torque disturbances and inertia variations. The use of classic closed-loop algorithms such as proportional–integral-derivative (PID) control is inadequate since these algorithms are often sensitive to mechanical configuration changes [8]. Classic control methods that make use of linear models for designing controllers are valid only for small variation around an operating point. This problem can be solved by applying advanced closed-loop control techniques such as self-tuning regulation (STR) [8] or nonlinear feedback control [9] where the controller is enforced to adapt itself to the motor operating conditions. Applied to stepper motors, STR gives better performance than PID controller because this technique is adaptive to system variations [8]. Nevertheless, this kind of control strategy is difficult to be implemented due to the large amount of floating-point computation, which necessarily increases the sampling period. In adaptive methods, the control laws such as model reference adaptive control and STR are nonlinear control laws which are difficult to derive. Furthermore, they require accurate parameters tuning.

Various feedback control methods have been developed to improve the performance of microstepping for permanent magnet stepper motors [10–12]. An optimal algorithm for closed-loop control of HSM drives has been presented in [13]. A third order

* Tel.: +966 500791353; fax: +966 14 6620709.
*E-mail address:* mszaky78@yahoo.com

http://dx.doi.org/10.1016/j.epsr.2014.10.004
0378-7796/

sliding mode controller (SMC) for a stepper motor has been proposed in [14]. A Lyapunov-based control with a nonlinear observer in microstepping for permanent magnet stepper motors has been proposed in [15]. In this work, the stability of the closed-loop system has been analyzed. The application of the conditional servo-compensator technique to position control of a permanent magnet stepper motor, which incorporates a servocompensator as part of a robust SMC design, has been presented in [16].

Recently, artificial intelligence control techniques have been presented to represent motion control systems which are difficult to be represented by mathematical models [17–21]. A laboratory experiment of fuzzy logic controller using a 16-b microcontroller for a stepper motor drive has been described in [8,17]. An artificial neural network (ANN) control scheme has been proposed in [18,19]. The ANN shows good results in stepper motor speed trajectory tracking [18]. However, when the system is subjected to a sudden load disturbance, the ANN takes a long recovery time to cope the changes because it requires large amount of computations for learning and adaptation. In [19], the controller achieves the tracking process with a high degree of accuracy; even in the presence of external disturbance. A low-speed-damping controller for a stepper motor using ANN, to improve the performance at less than the resonance speed, has been presented in [20]. An intelligent control scheme using online growing radial basis neural network controller in parallel with an implicit model adaptive controller has been proposed in [21].

It is well known that a conventional PI controller is most widely used in industry due to its simple control structure, ease of design and low cost. However, the PI controller is unable of simultaneously meeting good step reference tracking and good load torque rejection. Moreover, it suffers from disadvantages of slow response, large overshoots, and oscillations. Because of the widespread use of PI control it is highly desirable to tune this controller gains [22–25].

During the past decades, a lot of attention has been given to the problem of designing adaptive controllers and self-tuning regulators. Various types of self-tuning PID controllers have been proposed in [26]. Different approaches have been adopted for pole-assignment regulators, where the controllers are chosen in such a way that the closed-loop-system poles are placed at prespecified positions depending on the required system performance. Pole and/or zero assignment approach for the purpose of tuning a PID controller has been presented in [27–31]. Gain scheduled scheme for PI controller, as it is simple to implement and provides good performance, has been proposed in [32,33].

The PI controller gains of this control scheme are tuned such that the drive system exhibits satisfactory transient and steady state responses under varying operating conditions. Therefore, in order to improve the shortcomings of dynamic performance associated with the conventional fixed PI controller and at the same time maintains its advantages, a self-tuning of its gains becomes an effective choice.

The main contributions of this paper are as follows:

- The controller gains of the self-tuning PI speed controller are adapted online with the speed error unlike many of the previously fixed PI controllers.
- The self-tuning PI speed controller is designed to give better performance and to avoid the overshoots and undershoots under change of step speed reference and load torque disturbance.
- The effectiveness of the proposed self-tuning PI controller is verified using computer simulation and experiments throughout trajectory tracking accuracy and robustness properties against load torque disturbances and parameters variation.
- The proposed self-tuning PI controller is compared with the conventional One-Degree-Of-Freedom (1DOF) PI control and the Two-Degrees-Of-Freedom (2DOF) PI control.

This paper is organized as follows: Section 2 presents the dynamic modeling of HSM. Section 3 describes the mathematical model of FOC. In Section 4, the 1DOF PI controller, the 2DOF PI controller and the proposed self-tuning PI controller for speed control of HSM are presented and compared. Sections 5 and 6 present the simulation and experimental results. Finally, Section 7 gives the conclusion of the current paper.

## 2. Dynamic modeling of HSM

The mathematical model of a HSM assumes that the magnetic circuit is linear (no saturation) and the mutual inductance between phases is negligible. So, the motor can be represented by the following electric equations;

$$v_a = R_s i_a + L_s \frac{di_a}{dt} - N_r \psi_m \omega_r \sin(N_r \theta_r) \tag{1}$$

$$v_b = R_s i_b + L_s \frac{di_b}{dt} - N_r \psi_m \omega_r \sin(N_r \theta_r - \pi/2) \tag{2}$$

where $i_a$ and $i_b$ are the currents of phases A and B, $v_a$ and $v_b$ are the phase voltages, $R_s$ is the phase resistance, $L_s$ is the phase inductance, $\omega_r$ is the angular velocity, $\theta_r$ is the mechanical rotor position, $\psi_m$ is the motor maximum magnetic flux, and $N_r$ is the rotor teeth number.

The electromagnetic torque produced by a two-phase HSM is equal to the sum of the torque resulting from the interaction of the phase currents with magnetic fluxes created by the magnets and the detent torque, which results from the magnets of the rotor. The total torque equals

$$T_e = -N_r \psi_m i_a \sin(N_r \theta_r) - N_r \psi_m i_b \sin(N_r \theta_r - \pi/2) - T_{dm} \sin(2N_r \theta_r) \tag{3}$$

where $T_{dm}$ is the detent torque of HSM.

## 3. Field oriented control of HSM

### 3.1. Motor d–q model

In order to have linear and decoupling terms, the model of the HSM in the d–q rotor reference frame, by applying the Park transformation to Eqs. (1) and (2), can be expressed as follows [34,35]:

$$\frac{di_d}{dt} = -\frac{R_s}{L_d} i_d + N_r \omega_r i_q + \frac{v_d}{L_d} \tag{4}$$

$$\frac{di_q}{dt} = -\frac{R_s}{L_q} i_q - N_r \omega_r i_d - \frac{K_m}{L_q} \omega_r + \frac{v_q}{L_q} \tag{5}$$

$$\frac{d\omega_r}{dt} = \frac{T_e}{J} - \frac{B}{J} \omega_r - \frac{T_L}{J} \tag{6}$$

$$\frac{d\theta_r}{dt} = \omega_r \tag{7}$$

where $i_d$ and $i_q$ are the direct and quadrature currents. $v_d$ and $v_q$ are the direct and quadrature voltages. $L_d$ and $L_q$ are the direct and quadrature components of inductance. $K_m$ is the torque constant. $J$ is the coefficient of moment of inertia. $B$ is the coefficient of viscous friction.

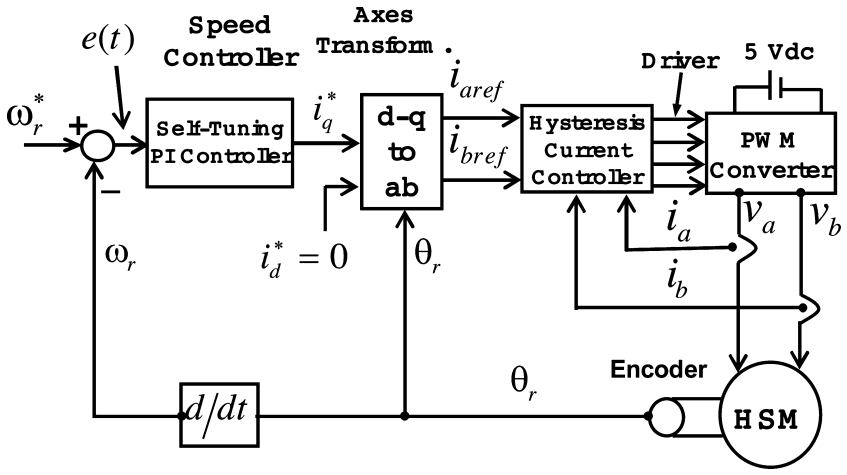


**Fig. 1.** Block diagram of a field oriented controlled HSM drive system using the proposed self-tuning PI controller.

### 3.2. Field oriented control principles

In order to simplify the control, direct current $i_d$ is set to zero. Then, the developed torque can be expressed as:

$$T_e = K_m i_q^*, \quad i_d = 0 \tag{8}$$

The electromechanical equation becomes

$$\frac{d\omega_r}{dt} = \frac{K_m}{J} i_q^* - \frac{B}{J}\omega_r - \frac{T_L}{J} \tag{9}$$

The overall block diagram control structure for the speed control of a HSM drive system can be constructed as shown in Fig. 1.

## 4. Speed control schemes of HSM

### 4.1. Design of the conventional PI controller

The conventional fixed PI controller can be constructed as follows:

$$i_q^* = K_{pc} e(t) + K_{ic} \int e(t)dt \tag{10}$$

The conventional PI controller gains ($K_{pc}$ and $K_{ic}$) are tuned at rated conditions based on Ziegler–Nichols tuning rules and have these fixed values along with the different operating conditions.

The Ziegler–Nichols tuning rules consist of determining the point where the Root-locus plot of the open loop system intersects the imaginary axis. This point is obtained by connecting a purely proportional controller to the system with zero integral gain, and by increasing the controller gain until the closed-loop system reaches the stability limit, at which sustained oscillations occur. The oscillation period is denoted $P_{cr}$ and the corresponding critical gain by $K_{cr}$. The Ziegler–Nichols choice for the PI gains is then specified as follows [22–24]:

$$\left.\begin{aligned} K_p &= 0.45\ K_{cr} \\ K_i &= 1.2\,K_p/P_{cr} \end{aligned}\right\} \tag{11}$$

### 4.2. Design of the self-tuning PI controller

The key feature of the proposed scheme of a self-tuning PI controller is that the gains are varied over a pre-determined range for varying operating conditions. It is well known that the proportional term ($K_p$) is responsible for improving overshoots, rise time response and the integral ($K_i$) term reduces steady state error. When the speed error is large, a high value of proportional gain is necessary for better control effort. Similarly, when the speed error is small a high value of integral gain is necessary to overcome steady state error.

The self-tuning PI controller output, which is considered as the reference torque component of the motor, can be described as:

$$i_q^* = K_p(t)e(t) + K_i(t) \int e(t)dt \tag{12}$$

where $e(t) = \omega_r^*(t) - \omega_r(t)$, $K_p(t)$ is the proportional gain and $K_i(t)$ is the integral gain. These gains are functions of the speed error $e(t)$. The gain $K_p(t)$ is expressed as a function of speed error as follows:

$$K_p(t) = K_{p(\max)} - (K_{p(\max)} - K_{p(\min)})e^{-[ke(t)]} \tag{13}$$

where $k$ is a constant which decides the rate at which $K_p(t)$ varies between maximum and minimum values of the proportional gain. A large proportional gain, $K_{p(\max)}$ is used to speed up the transient response when the speed error $e(t)$ is large and when the error $e(t)$ becomes small, a minimum proportional gain $K_{p(\min)}$ is used to eliminate overshoots and oscillations. The integral gain $K_i(t)$ is expressed as a function of speed error signal $e(t)$ as:

$$K_i(t) = K_{i(\max)}e^{-[ke(t)]} \tag{14}$$

Under steady state condition when speed error $e(t)$ is small, large integral gain is used to overcome the steady state error. When the error is large, a small integral gain is used in order to eliminate the undesirable oscillations and overshoot. In transient condition, a large control signal is used to accelerate or decelerate the motor to the reference value within smallest possible time. During this period, $K_p(t)$ is at its maximum value and $K_i(t)$ is maintained at its

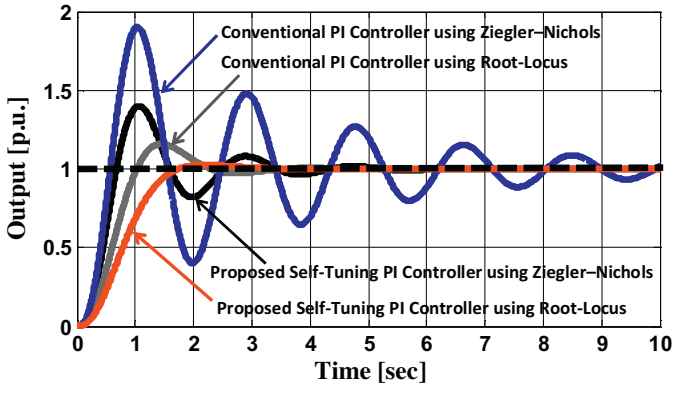


**Fig. 2.** Simulated step responses of both the conventional fixed PI controller and the proposed self-tuning PI controller for a highly under-damped control system.

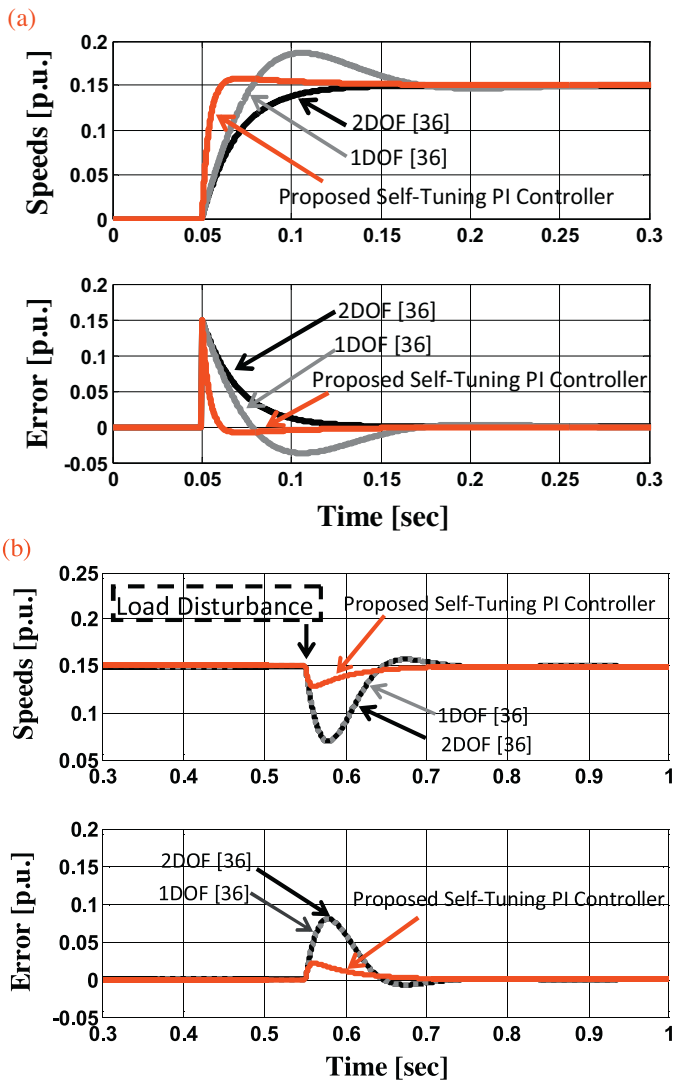


**Fig. 3.** Simulated step responses of the 1DOF, the 2DOF PI control and the proposed self-tuning PI control during, (a) reference tracking, and (b) load torque disturbance.

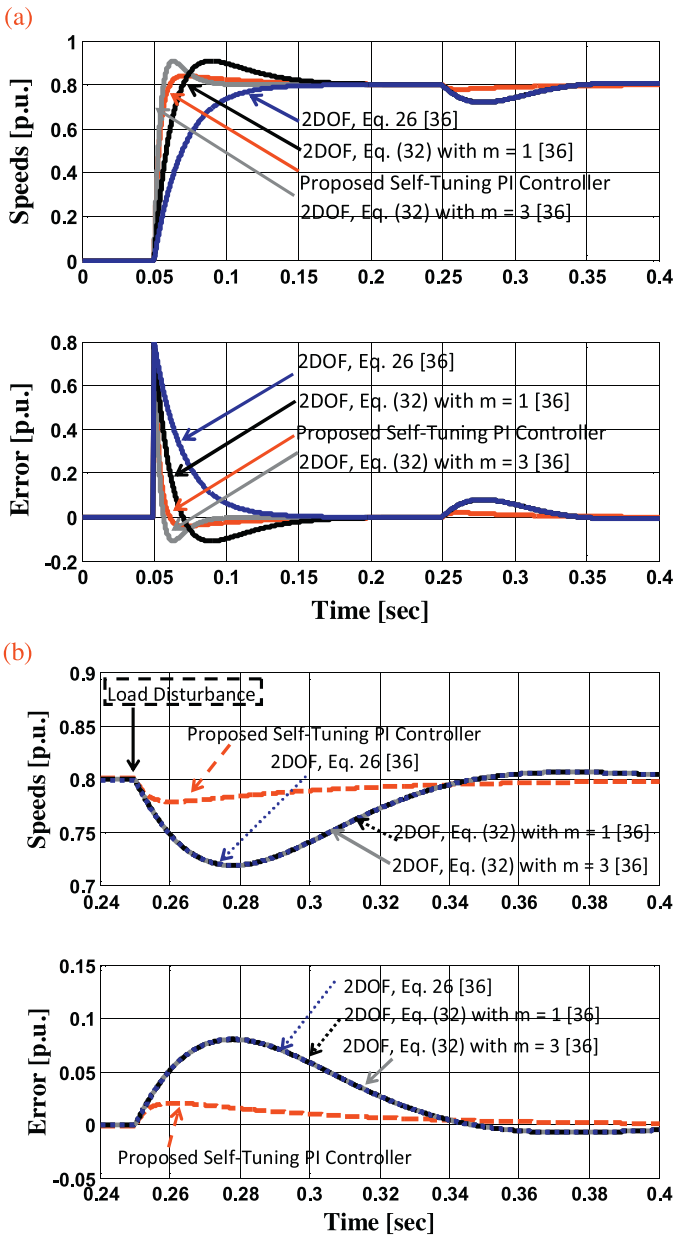


**Fig. 4.** Simulated step responses of all design types of the 2DOF control method as [36] and the proposed self-tuning PI control during, (a) reference tracking and (b) zoomed of load torque disturbance.

minimum value. Under steady state operating condition, the integral gain $K_i(t)$ is increased to its maximum value. These two gains are varied online as a function of speed error $e(t)$.

From Eq. (13), it can be obvious that when the error $e(t)$ is large the exponential term approaches zero ($e^{-[ke(t)]} \to 0$) and therefore, we have $K_p(t) = K_{p(\max)}$. Similarly, when $e(t)$ is small the exponential term approaches 1 ($e^{-[ke(t)]} \to 1$) and hence $K_p(t) = K_{p(\min)}$. The integral gain $K_i(t)$ in Eq. (14) varies in the range of $0 \le K_i(t) \le K_{i(\max)}$.

### 4.3. Example 1: A highly under-damped system

Consider the transfer function of a highly under-damped control system is $G_p(s) = 1/s\left(s^2 + 9s + 25\right)$ with $\xi = 0.9$, $\omega_n = 5$ rad/s.

The control system response is examined using the conventional PI controller. The gains of the PI controller are chosen using Ziegler–Nichols and Root-locus methods. Using Ziegler–Nichols tuning method, it has been found that $P_{cr} = 1.25$ and $K_{cr} = 225$. Then, the gains of the conventional PI controller using Eq. (11) become $K_p = 0.45K_{cr} = 101.25$ and $K_i = 1.2\,K_p/P_{cr} = 97.2$. In case of using Root-locus plot, the gains of the conventional PI controller become $K_p = 50$ and $K_i = 0.01$.

The self-tuning PI controller is designed also using Ziegler–Nichols and Root-locus methods. Using Ziegler–Nichols tuning method, it has been found that $K_{p(\max)} = 2K_{cr} = 450$, $K_{p(\min)} = 0.45K_p = 101.25$, $K_{i(\max)} = 0.05$ and $k = 0.1$. In case of using Root-locus method, it has been found that the PI controller zero $K_i/K_p$ should be 0.005 and the proportional gain $K_p$ should be large as 25 to guarantee the stability and give good dynamic performance. The self-tuning PI controller is designed to have the following gains based on Root-locus plot, $K_{p(\min)} = 25$, $K_{p(\max)} = (1.5:2)K_{p(\min)} = 37.5:50$, $K_{i(\max)} = 0.125:0.01$. and $k = 0.01$.

The simulation results for the output response of the above system to a unit step reference are shown in Fig. 2. It is observed that the proposed self-tuning PI controller using Root-locus method gives a better performance in comparison to the other control methods. This would show how the proposed control respond in a stepper motor system with little mechanical damping. As clear, the speed error decays quickly to zero. Also, the proposed controller has minimum overshoots and small settling time when compared with the conventional PI controller. Moreover, it has been justified the

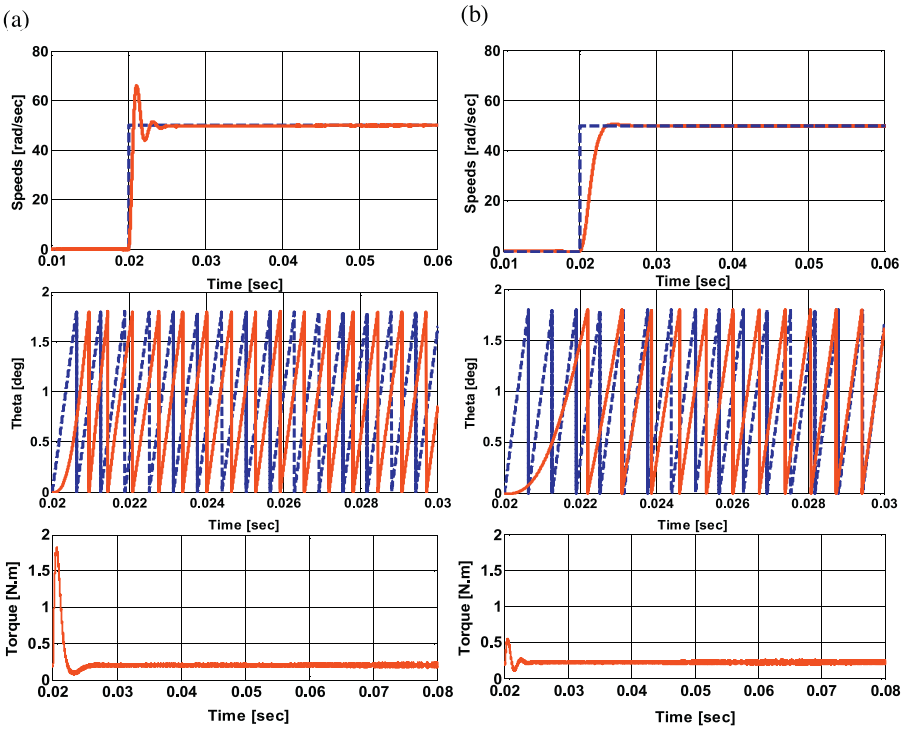


**Fig. 5.** Simulated responses of reference and rotor speeds, reference and actual rotor position signal, and developed torque during starting operation using, (a) the conventional fixed PI controller, and (b) the proposed self-tuning PI controller.

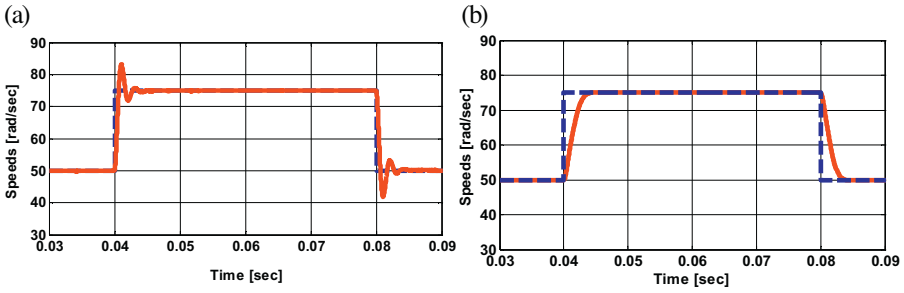


**Fig. 6.** Simulated speed responses during sudden change of speed using, (a) the conventional fixed PI controller, and (b) the proposed self-tuning PI controller.

fact that Ziegler–Nichols tuning method for tuning a PI controller is obsolete and gives too small stability margins.

### 4.4. Example 2: Comparison with 1DOF and 2DOF control methods [36]

The simulation results for comparing the proposed self-tuning PI controller with the 1DOF and 2DOF control methods of [36] are shown in Fig. 3. The step reference input and the load disturbance are applied at $t = 0.05$ s and $t = 0.55$ s, respectively. It has been observed that the 1DOF PI control is inherently incapable of simultaneously meeting good step reference tracking and good load torque rejection during transients [36]. So, to reduce or eliminate the overshoot appearing for the 1DOF PI control, 2DOF control has been used {Eq. (26) with $m = 1$ in [36]}. However, the speed response is slow. Alternatively, the self-tuning PI controller gives a

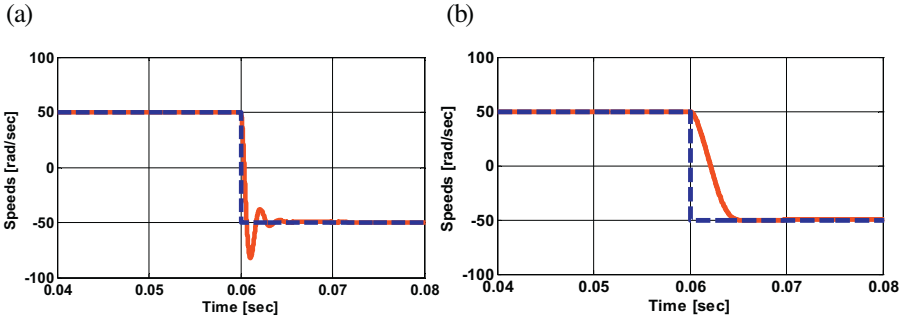


**Fig. 7.** Simulated speed responses during speed reversal using, (a) the conventional fixed PI controller, and (b) the proposed self-tuning PI controller.

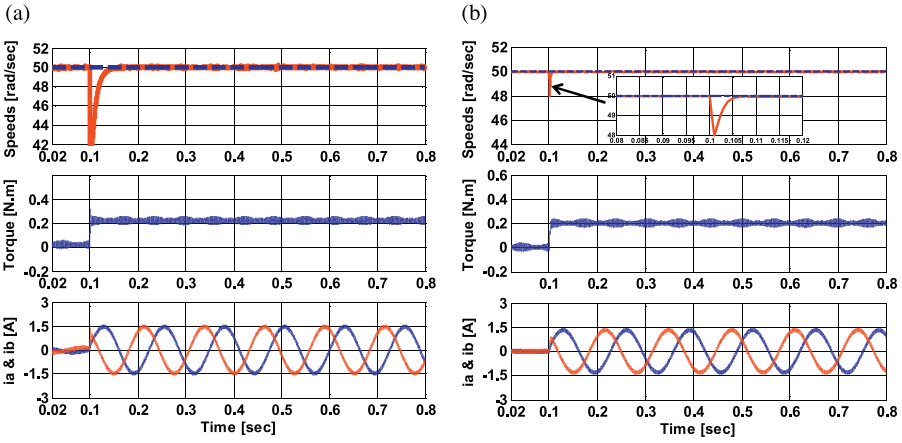


**Fig. 8.** Simulated responses of reference and rotor speeds, developed torque, and stator currents of phases a and b during sudden load change using, (a) the conventional fixed PI controller, and (b) the proposed self-tuning PI controller.

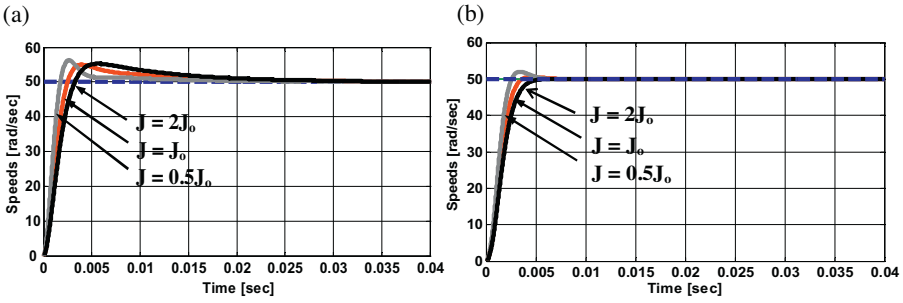


**Fig. 9.** Simulated speed responses under load inertia variations using (a) the conventional fixed PI controller, and (b) the proposed self-tuning PI controller.

significant performance in comparison to the 1DOF and the 2DOF control methods under speed reference tracking and load torque disturbance.

The simulation results for comparing the proposed self-tuning PI controller with all design types of the 2DOF control methods presented in [36] are shown in Fig. 4. The step reference input and the load disturbance are applied at $t = 0.05$ s and $t = 0.25$ s, respectively. As obvious, the 2DOF using Eq. (32) with $m = 3$ in [36] gives an excellent performance when compared with the other two design types of the 2DOF in terms of fast response [36]. However, the speed response suffers from overshoot and the speed dip is the same in all types of 2DOF control methods [36]. On the other hand, the proposed self-tuning PI controller ensures a superior performance during speed reference tracking and load torque disturbance in comparison to all types of the 2DOF control methods presented in [36] with low overshoot and minimum speed dip during torque disturbance.

## 5. Simulation results

Simulation model is established in Matlab/Simulink environment based on the introduced mathematical model. The performance of the HSM drive system is tested under different operating conditions. Simulation results are presented under starting operation, step speed command, speed reversal, parameters variation and load torque disturbances. Performance of the self-tuning PI controller in comparison to the conventional PI controller is examined and assessed. The gains of the conventional PI and the self-tuning PI controllers are given in Tables 1 and 2, respectively. The rating and parameters of the HSM are given in Table 3.

### 5.1. Reference tracking performance

The HSM drive system is tested using the two speed controllers for trajectory tracking performance. Fig. 5(a) and (b) shows the simulated speed responses, the developed torque and the rotor position signal during staring operation at 50 rad/s using the conventional fixed PI and the self-tuning PI controllers. It is obvious that the speed response and the developed torque with the fixed PI controller suffer from overshoots in comparison to the self-tuning PI controller. Moreover, the rotor position signal has an error and takes a long time to reach the steady state value. The same conclusion is observed with step speed reference change from 50 to 75 rad/s as shown in Fig. 6(a) and (b).

The stability and synchronization of the drive system with the two controllers are examined during speed reversal as shown in Fig. 7(a) and (b). It is evident that the proposed self-tuning PI controller gives a good performance in comparison to the conventional fixed PI controller in terms of overshoot.

**Table 1**
Gains of conventional PI speed controller.

| | |
|---|---|
| Proportional gain | 0.6 |
| Integral gain | 0.01 |

**Table 2**
Gains of the proposed self-tuning PI speed controller.

| | |
|---|---|
| Proportional gain max | 0.8 |
| Proportional gain min | 0.6 |
| Integral gain | 0.01 |
| $k$ | 0.1 |

**Table 3**
Parameters of HSM.

| | | | |
|---|---|---|---|
| Rated voltage (V) | 5 | Coefficient of viscous friction [N m s/rad] | 0.00047 |
| Phase resistance (Ohm) | 0.37 | Torque constant | 0.153 |
| Phase inductance (mH) | 0.9 | Rotor inertia (kg m$^2$) | 15.62e−5 |
| Number of rotor teeth | 50 | The step angle | 1.8° |
| Number of phases | 2 | Detent torque (N m) | 0.002 |
| Maximum flux linkage (Wb) | 0.005 | Maximum speed (rpm) | 440 |
| Rated speed (rpm) | 314 | | |

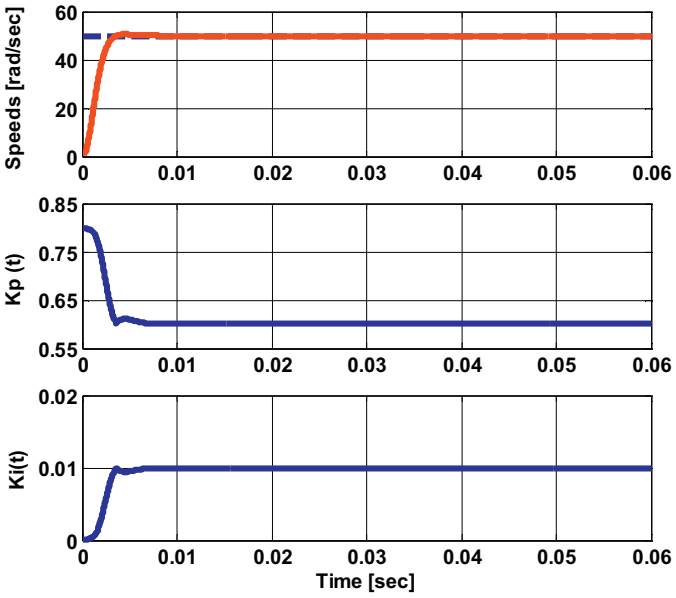


**Fig. 10.** Simulated responses of speeds and variations of $K_p(t)$ and $K_i(t)$ for starting operation using the proposed self-tuning PI controller.

### 5.2. Robustness against load torque disturbances and parameters variation

The robustness of the self-tuning PI controller is proved and confirmed under load torque disturbances. Fig. 8(a) and (b) show the speed response, the developed torque and the stator currents of the HSM drive system with the two speed controllers at 50 rad/s speed reference. It is observed that the speed recovers quickly and the speed dip is low using the self-tuning PI controller which is not the case with the conventional fixed PI controller.

The sensitivity of the proposed self-tuning PI controller is also examined under load inertia variations. Fig. 9(a) shows the speed responses at 50 rad/s with different load inertias ($J=2J_o$, $J=J_o$, $J=0.5J_o$) using the conventional fixed PI controller. The same simulation responses are taken using the self-tuning PI controller as shown in Fig. 9(b). These results show that the speed response with the proposed controller is better than the corresponding response with the fixed PI controller in terms of overshoots and settling time. However, the fixed PI controller is better in its rising time compared to the proposed controller. The main reason of this performance for the conventional fixed PI controller reveals to that it is designed at certain operating point. If this operating point is changed the performance of PI controller is degraded.

The variations of $K_p(t)$ and $K_i(t)$ of the proposed self-tuning PI controller during transient conditions are shown in Figs. 10–12. The simulated speed responses and both $K_p(t)$ and $K_i(t)$ variations during starting condition are shown in Fig. 10. The same responses are taken under the repetitive operation and the sudden load change as shown in Figs. 11 and 12, respectively. As obvious, the self-tuning PI controller gains are updated online depending on the speed error. Therefore, the proposed controller has maximum capabilities at large speed error. However, if the speed error decays to zero, the controller has the minimum gains which they are the gains of the conventional PI one. This guarantees that the proposed self-tuning PI controller achieves simultaneously good reference tracking and good load torque rejection.

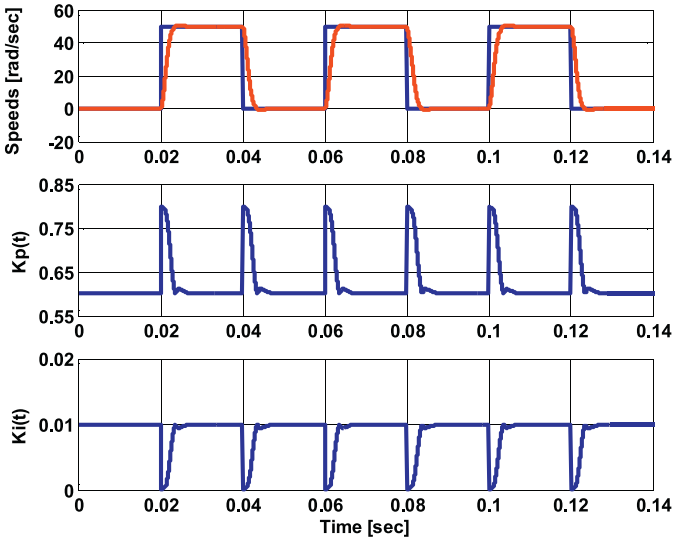


**Fig. 11.** Simulated responses of speeds and variations of $K_p(t)$ and $K_i(t)$ under repetitive operation using the proposed self-tuning PI controller.

## 6. Experimental results

The basic configuration of the experimental system is shown in Fig. 13. It consists of a HSM interfaced with a digital control board based on a Texas Instruments TMS320C31 Digital Signal

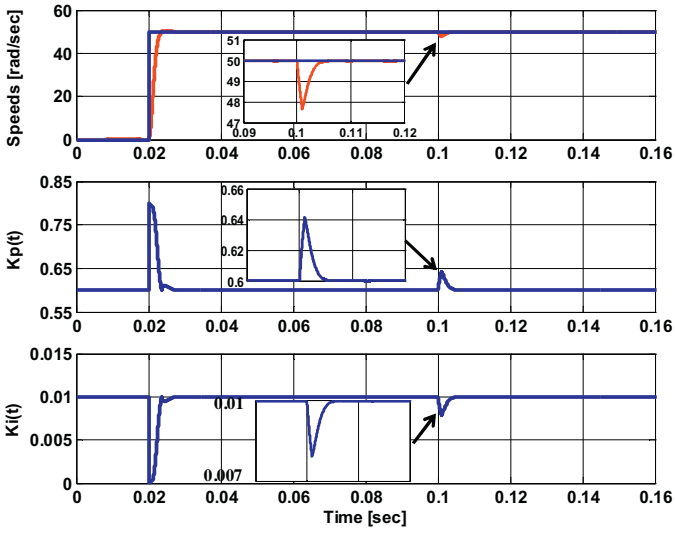


**Fig. 12.** Simulated responses of speeds and variations of $K_p(t)$ and $K_i(t)$ under load torque disturbance using the proposed self-tuning PI controller.

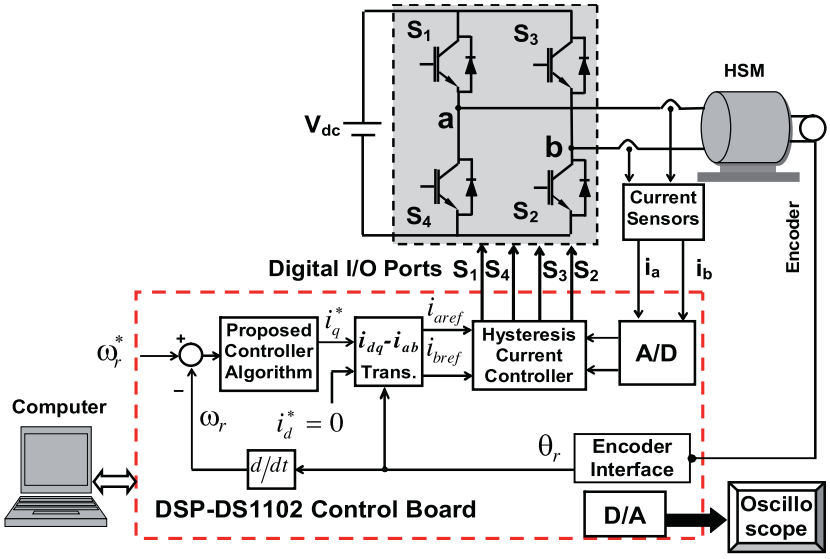


**Fig. 13.** Block diagram of DSP-based real-time implementation of the proposed self-tuning PI controller for field oriented control of HSM drive system.

Processor for real-time control. Stator currents are measured using Hall-effect sensors. The measured current signals are acquired by the A/D input ports of the DSP control board. This board is hosted by a personal computer on which mathematical control algorithms are programmed and downloaded for real-time control.

The motor phases are fed by two H-bridge MOSFET power converter connected to a 5 V DC voltage source. The motor phase currents are controlled by a hysteresis controller which generates the MOSFET gate drive signals by comparing the measured currents with their corresponding references. The output switching commands of the DSP control board are obtained via its digital ports and interfaced with the converter through an opto-isolated gate drive circuit. A picture of the overall experimental system of HSM using DSP control board and the associated interface circuits is shown in Fig. 14.

Figs. 15–17 show the reference and rotor speeds, the rotor position signal and the absolute rotor position under step speed change using the self-tuning PI controller. This includes step change of reference speed at different reference values 314, 130, and 60 rpm, respectively. It is obvious that the rotor speed, and consequently, the rotor position reaches the steady state value smoothly without overshoot or undershoot. Moreover, the drive system has a fast dynamic response and takes a minimum rise time to reach the steady state value. However, the rotor speed contains ripples which increase at low speeds as observed from Fig. 17.

Experimental results are presented also during forward and reverse motoring operation at 314 rpm as shown in Fig. 18. It is clear that the rotor speed follows the reference speed smoothly. Moreover, experimental results are presented with repetitive operation to test the precise operation of the drive system and the fast

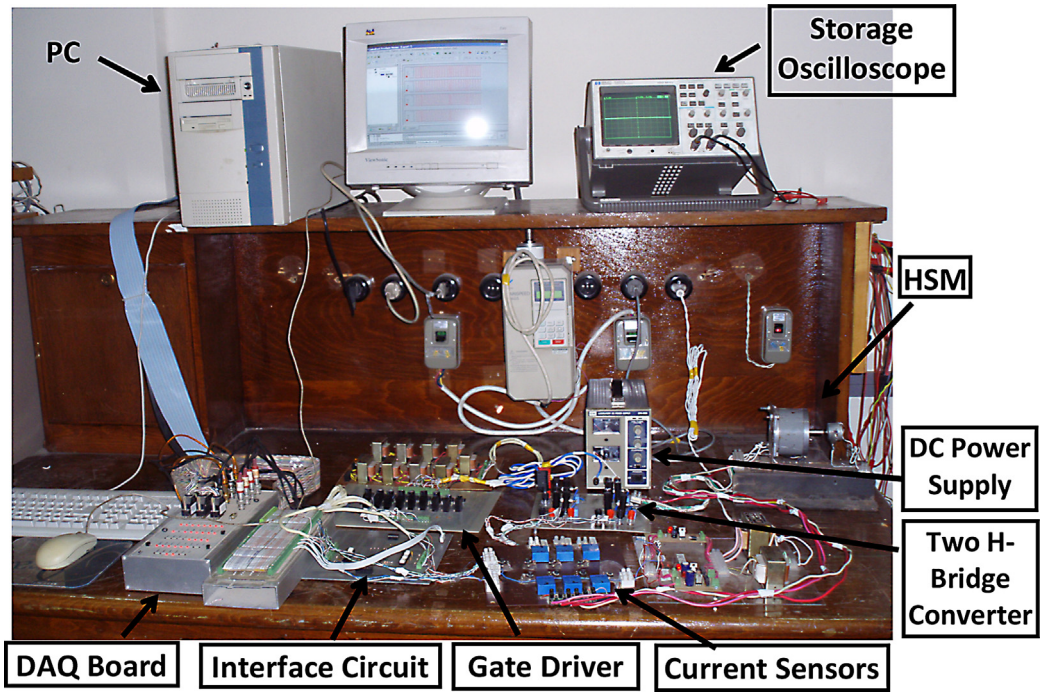


**Fig. 14.** A picture of the overall experimental system of HSM using DSP-DS1102 control board.

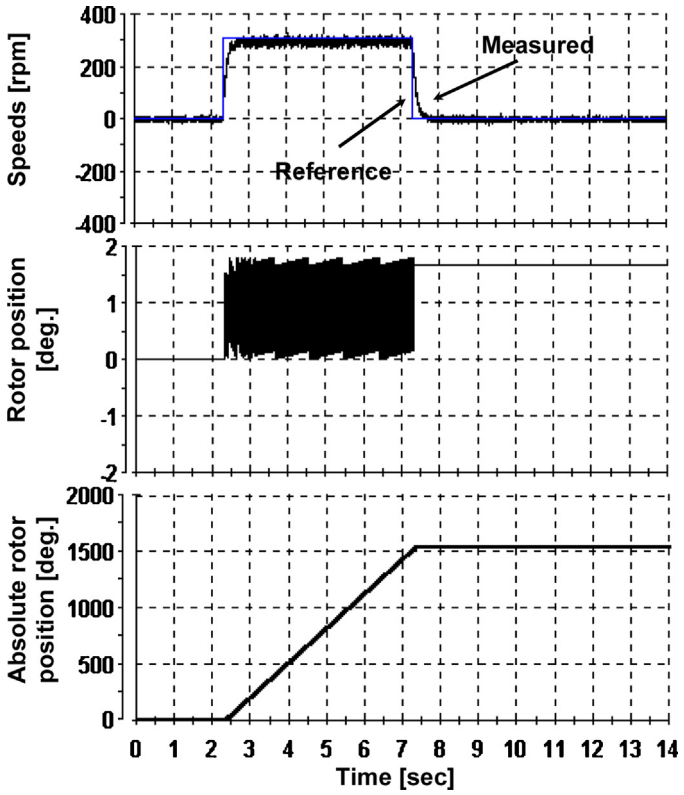


**Fig. 15.** Experimental results of reference and actual speeds, rotor position signal, and absolute rotor position during step change of speed reference at 314 rpm using the proposed self-tuning PI controller.

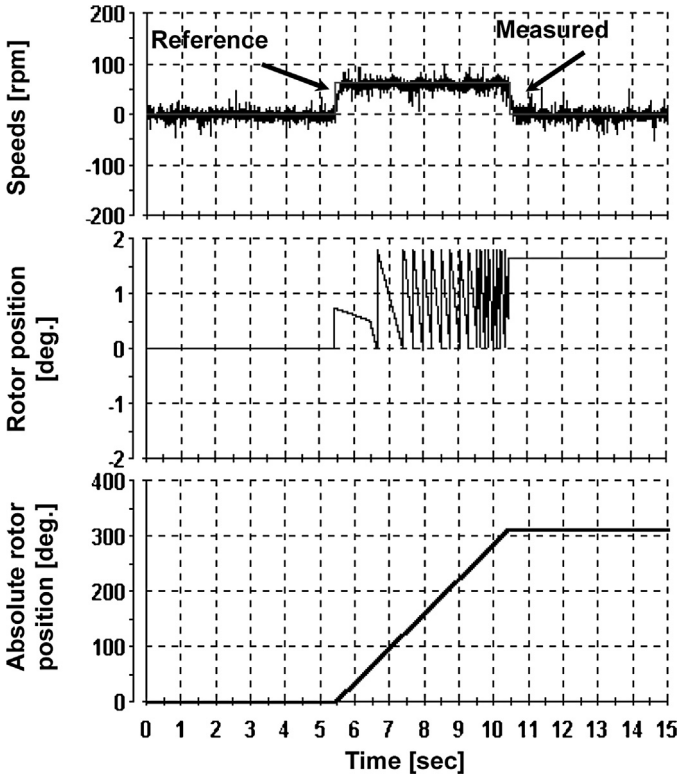


**Fig. 17.** Experimental results of reference and actual speeds, rotor position signal, and absolute rotor position during step change of speed reference at 60 rpm using the proposed self-tuning PI controller.

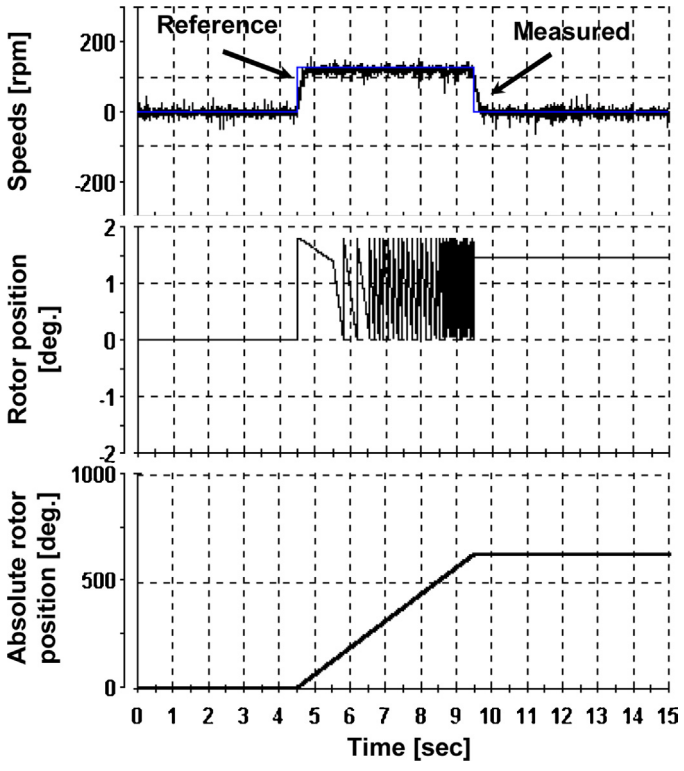


**Fig. 16.** Experimental results of reference and actual speeds, rotor position signal, and absolute rotor position during step change of speed reference at 130 rpm using the proposed self-tuning PI controller.

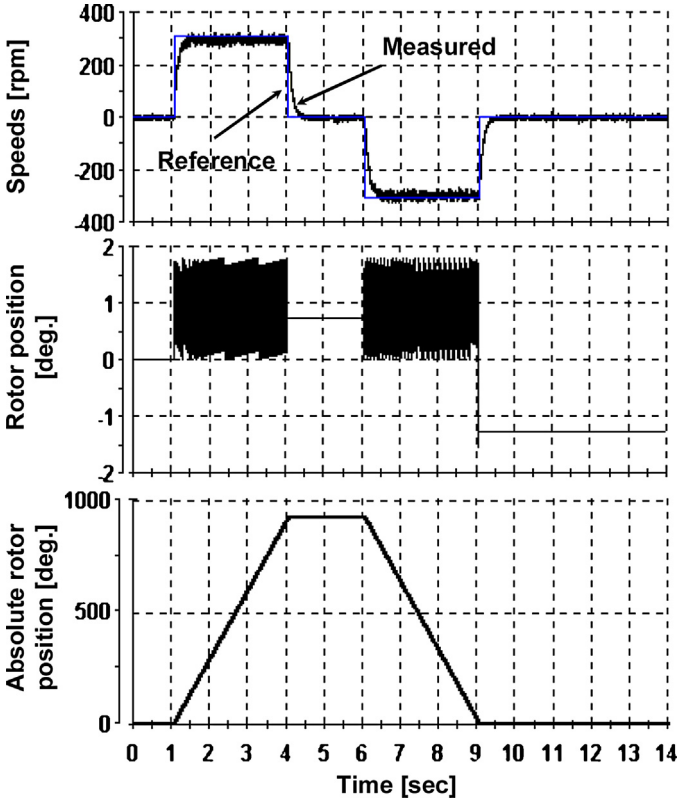


**Fig. 18.** Experimental results of reference and actual speeds, rotor position signal, and absolute rotor position during forward and reverse motoring operation at 314 rpm using the proposed self-tuning PI controller.

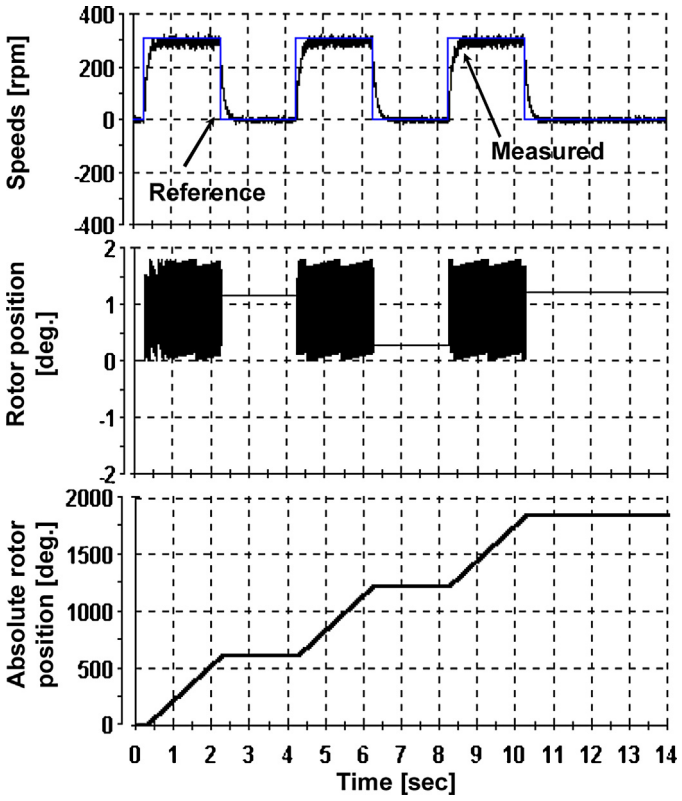


**Fig. 19.** Experimental results of reference and actual speeds, rotor position signal, and absolute rotor position during repetitive operation at 314 rpm using the proposed self-tuning PI controller.

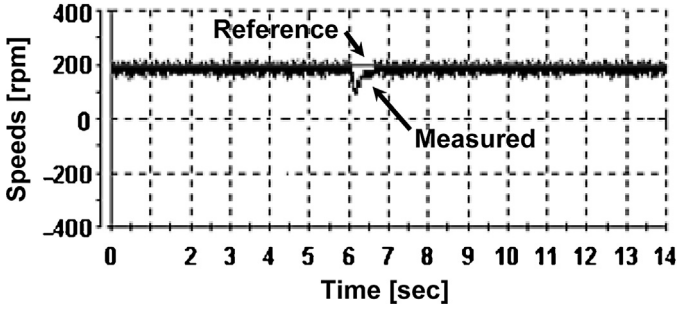


**Fig. 20.** Experimental results of reference and actual speeds under sudden change of load torque of 0.2 Nm at 200 rpm using the proposed self-tuning PI controller.

dynamic response as shown in Fig. 19. It is observed that the drive system maintains its stability and synchronization with fast start and stop operations. This proves the effectiveness of the proposed closed-loop control of HSM drive system.

The HSM drive system is also tested and examined experimentally under sudden load change from no-load to full-load of 0.2 N m. It is obvious that the speed recovers quickly to the steady state value as shown in Fig. 20.

In experimental system, the speed sensor is followed by a low pass filter to alleviate the high speed ripples. This is the main reason for the delay time in the measured speed compared to the simulation one. Moreover, in real-time control using DSP-DS1102, the systems exhibits some delay due to data acquisition ports like A/D converter, computation time burden due to field oriented control. This makes a delay in the control effort.

The self-tuning PI controller gives a good performance in comparison to PID control and STR [8], and Online Random Training of Neural Networks [18] and exhibits a similar performance as multilayer neural networks [19]. However, it is simple when compared with these approaches and no need for adding sensors.

## 7. Conclusion

In this paper, the self-tuning PI speed controller for electrical motor drives has been presented. The HSM drive system as a test bed has been examined experimentally and by computer simulations using the proposed self-tuning PI controller and the conventional fixed PI one. The gains of the proposed controller have been varied and tuned such that the drive system exhibits satisfactory transient and steady state response under varying operating conditions. Experimental and simulation results show the effectiveness of this approach, and demonstrate the usefulness of the proposed controller for high performance drives. The proposed controller method has shown a good performance in comparison to the conventional fixed PI controller in terms of trajectory tracking, load inertia variations and load torque disturbances. Furthermore, the proposed method has characterized by its simplicity, low computation time, ease of implementation and improved dynamic performance under parameters variation and load torque disturbances. It has been concluded that the proposed controller is very simple and has given improved results when compared with previous approaches such as IDOF and 2DOF control methods.

## Acknowledgment

The author would like to thank the reviewers for their professional work and helpful suggestions.

## References

[1] P. Krishnamurthy, F. Khorrami, Voltage-fed permanent-magnet stepper motor control via position-only feedback, IEE Proc. Control Theory Appl. 151 (4) (2004) 499–510.
[2] P. Krishnamurthy, F. Khorrami, Robust adaptive voltage-fed permanent magnet step motor control without current measurements, IEEE Trans. Control Syst. Technol. 11 (3) (2003) 415–425.
[3] H. Melkote, F. Khorrami, Robust nonlinear control and torque ripple reduction for permanent magnet stepper motors, IEE Proc. Control Theory Appl. 146 (2) (1999) 186–196.
[4] H. Melkote, F. Khorrami, S. Jain, M.S. Mattice, Robust adaptive control of variable reluctance stepper motors, IEEE Trans. Control Syst. Technol. 7 (2) (1999) 212–221.
[5] M. Butcher, A. Masi, R. Picatoste, A. Giustiniani, Hybrid stepper motor electrical model extensions for use in intelligent drives, IEEE Trans. Ind. Electron. 61 (2) (2014) 917–929.
[6] D. Carrica, M.A. Funes, S.A. González, Novel stepper motor controller based on FPGA hardware implementation, IEEE/ASME Trans. Mechatron. 8 (1) (2003) 120–124.
[7] J.B. Grimbleby, Simple algorithm for closed-loop control of stepping motors, IEE Proc. Electr. Power Appl. 142 (I) (1995) 5–13.
[8] F. Betin, M. Deloizy, C. Goeldel, Closed loop control of stepping motor drive: comparison between PID control, self tuning regulation and fuzzy logic control, Eur. Power Electron. J. 8 (1–2) (1999) 33–39.
[9] M. Bodson, J.N. Chiasson, R.T. Novotnak, R.B. Rekowski, High performance nonlinear feedback control of permanent-magnet stepper motor, IEEE Trans. Control Syst. Technol. 1 (1) (1993) 5–14.
[10] W. Kim, D. Shin, C.C. Chung, Microstepping with nonlinear torque modulation for permanent magnet stepper motors, IEEE Trans. Control Syst. Technol. 21 (5) (2013) 1971–1979.
[11] D. Shin, W. Kim, Y. Lee, C.C. Chung, Phase-compensated microstepping for permanent-magnet stepper motors, IEEE Trans. Ind. Electron. 60 (12) (2013) 5773–5780.
[12] W. Kim, D. Shin, C.C. Chung, Microstepping using a disturbance observer and a variable structure controller for permanent-magnet stepper motors, IEEE Trans. Ind. Electron. 60 (7) (2013) 2689–2699.
[13] P. Crnosija, B. Kuzmanovic, S. Ajdukovic, Microcomputer implementation of optimal algorithms for closed-loop control of hybrid stepper motor drives, IEEE Trans. Ind. Electron. 47 (6) (2000) 1319–1325.
[14] M. Defoort, F. Nollet, T. Floquet, W. Perruquetti, A third-order sliding-mode controller for a stepper motor, IEEE Trans. Ind. Electron. 56 (9) (2009) 3337–3346.
[15] W. Kim, I. Choi, C.C. Chung, Lyapunov-based control in microstepping with a nonlinear observer for permanent magnet stepper motors, Am. Control Conf. (ACC) (2010) 4313–4318.

[16] S. Seshagiri, Position control of permanent magnet stepper motors using conditional servo compensators, IET Control Theory Appl. 3 (9) (2009) 1196–1208.
[17] F. Betin, D. Pinchon, G.-A. Capolino, Fuzzy logic applied to speed control of stepping motor drive, IEEE Trans. Ind. Electron. 47 (3) (2000) 610–622.
[18] A. Rubaai, R. Kotaru, Adaptation learning control scheme for a high-performance permanent-magnet stepper motor using online random training of neural networks, IEEE Trans. Ind. Appl. 37 (2) (2001) 495–502.
[19] A. Rubaai, M.J. Castro-Sitiriche, M. Garuba, L. Burge, Implementation of artificial neural network-based tracking controller for high-performance stepper motor drives, IEEE Trans. Ind. Electron. 54 (1) (2007) 218–227.
[20] Q.N. Le, J.W. Jeon, Neural-network-based low-speed-damping controller for stepper motor with an FPGA, IEEE Trans. Ind. Electron. 57 (9) (2010) 3167–3180.
[21] S. Kamalasadan, A new intelligent controller for the precision tracking of permanent magnet stepper motor, IEEE Power Eng. Soc. Gen. Meet. (2007) 1–7.
[22] O. Lequin, M. Gevers, M. Mossberg, E. Bosmans, L. Triest, Iterative feedback tuning of PID parameters: comparison with classical tuning rules, Control Eng. Pract. 11 (2003) 1023–1033.
[23] K.J. Astrom, T. Hagglund, Revisiting the Ziegler–Nichols step response method for PID control, J. Process Control 14 (2004) 635–650.
[24] B. Kristiansson, B. Lennartson, Evaluation and simple tuning of PID controllers with high-frequency robustness, J. Process Control 16 (2006) 91–102.
[25] W.D. Chang, R.C. Hwang, J.G. Hsieh, A self-tuning PID control for a class of nonlinear systems based on the Lyapunov approach, J. Process Control 12 (2002) 233–242.
[26] R. Ortega, R. Kelly, PID self-tuners: some theoretical and practical aspects, IEEE Trans. Ind. Electron. 31 (4) (1984) 332–338.
[27] K.J. Astrom, B. Wittenmark, Self-tuning controllers based on pole-zero placement, Proc. IEE 127 (3) (1980) 120–130.
[28] D.W. Clarke, P.J. Gawthrop, Self-tuning controller, Proc. IEE 122 (9) (1975) 929–934.
[29] S.C. Puthenpura, J.F. MacGregor, Pole-zero placement controllers and self-tuning regulators with better set-point tracking, Proc. IEE 134 (1) (1987) 26–30.
[30] T.F. Chwee, H.R. Sirisena, Self-tuning PID controllers for dead time processes, IEEE Trans. Ind. Electron. 35 (1) (1988) 119–125.
[31] A.Y. Allidina, F.M. Hughes, Generalised self-tuning controller with pole assignment, Proc. IEE 127 (1) (1980) 13–18.
[32] S.K. Panda, J.M.S. Lim, P.K. Bash, K.S. Lock, Gain scheduled PI speed controller for PMSM drive, in: IEEE Industrial Electronics Society International Conference IECON 97, 1997, pp. 925–930.
[33] B. Singh, A.H.N. Reddy, S.S. Murthy, Gain scheduling control of permanent magnet brushless DC motor, IE (I) J.-EL 84 (2003) 52–62.
[34] W. Kim, C. Yang, C.C. Chung, Design and implementation of simple field-oriented control for permanent magnet stepper motors without DQ transformation, IEEE Trans. Magn. 47 (10) (2011) 4231–4234.
[35] M. Bendjedia, Y. Ait-Amirat, B. Walther, A. Berthon, Position control of a sensorless stepper motor, IEEE Trans. Power Electron. 27 (2) (2012) 578–587.
[36] L. Harnefors, S.E. Saarakkala, M. Hinkkanen, Speed control of electrical drives using classical control methods, IEEE Trans. Ind. Appl. 49 (2) (2013) 889–898.