Journal of Cybernetics and Informatics

published by

Slovak Society for
Cybernetics and Informatics

Volume 13, 2012

http://www.sski.sk/casopis/index.php (home page)

ISSN: 1336-4774

Journal of Cybernetics and Informatics 13 (2012)
http://www.sski.sk/casopis/index.php

PI CONTROLLER BASED ON GENETIC ALGORITHM FOR PMSM DRIVE SYSTEM

Faeka Khater*, Adel Shaltout, Essam Hendawi*, and Mohamed Abu El-sebah***

* Electronics Research Institute, Dokki, Giza, Egypt. ** Cairo University, Giza, Egypt.
Tel:+20 2 33310554 Fax :+20 2 33351631
khater@eri.sci.eg, aashaltout@yahoo.com, esam_hendawi@hotmail.com,
mohamedibrahi32@hotmail.com

Abstract

This paper introduces a genetic-algorithm-based PI controller for position control of permanent magnet synchronous motor. The algorithm is proposed for optimizing the PI controller gains in the position control. Different controllers' strategies are applied for the cascaded-loop position controller, speed controller and current controllers. The controllers are compared together to select the best one. The objective target, which has been used for comparison, is the rise time, settling time, steady state error. In addition, the response of the developed torque is investigated. Simulation results show that using genetic-algorithm-based PI controller gives the best performance.

Keyword: PI controller, PMSM control, Genetic Algorithm

1 INTRODUCTION

Permanent magnet synchronous motors (PMSM) have gained an increasing popularity where they are widely used in high performance drive applications such as servo systems, machine tool drives, computer peripherals, industrial robotics, electric vehicles and other industry applications [1-3]. This is because PMSMs have advantages as compared with induction motors such as the absence of rotor copper losses which heats the rotor and decreases the efficiency, high torque to inertia ratio, high efficiency and high power factor. These advantages make PMSMs competitive over induction motors. Also PMSMs are preferred to dc motors in applications that require variable speed drives because of the high torque-to-inertia ratio, an excellent power factor close to unity, high acceleration and high efficiency of PMSMs.

PMSMs are used in many applications that require fast and accurate torque response in particular servo system with position control [4-6]. Position control system belongs to cascade control systems which require several control loops. In this paper, different combinations of controllers are applied to the system using conventional PI controllers, fuzzy logic controller with conventional PI controller and PI controller optimized by Genetic algorithm. Simulation is carried out to predict the performance of the drive system for all combinations. The objective target which has been used for comparison is the rise time, settling time, steady state error. In addition, the response of the developed torque is investigated. It will be shown that applying genetic algorithm optimization to the PI controller in position control gives the best performance.

This paper is organized as follows. Section 1 presents the introduction. In sections 2 and 3, PMSM drive system description and motor model are given respectively. Fuzzy logic controller is reviewed in section 4. The proposed technique using genetic algorithm is applied to optimize the PI controller parameters during system operation using triangular distribution of controller parameters in sections 5 and 6. In section 7, simulation results will be shown. Section 8 gives conclusion.

2 SYSTEM DESCRIPTION

The block diagram of the PMSM drive system is shown in Fig. 1. The outer loop is the position controller. The intermediate loop is the speed controller. The control action of this controller is the reference value of the q-axis current. The reference value of the d-axis current is set to zero. The inner loops are the current controllers in which the control actions of these loops are the reference value of the q-axis voltage and the reference value of the d-axis voltage. These reference values of the voltage vector (in the rotating reference frame) are transformed finally to stator three-phase reference frame.

The main advantage of conventional PI controllers is that they are easy to implement. However, conventional PI controllers suffer from problems due to changes in system dynamics or variation in operating points as a result of parameter variations. These problems may affect the system performance using controllers that have fixed parameters. To overcome such deficiency, more efficient controllers such as fuzzy logic controllers may be used or by on-line tuning of the parameters of the PI controller.

Since the effects of the outer loop are dominant compared with the other loops, the inner loop (current controllers), are selected to be conventional PI controller in all cases. The following combinations of controllers are to be considered for investigation:

Case 1: all the controllers are conventional PI controller

Case 2: position controller is fuzzy logic controller and PI controller for speed and current controllers

Case 3: position and speed controllers are fuzzy logic controllers and PI controller for current controllers

Case 4: genetic algorithm-based PI controller (proposed method) for position controller and PI controller for speed and current controllers.

For all controllers, the goal is keeping the absolute value of steady state error to be minimum.

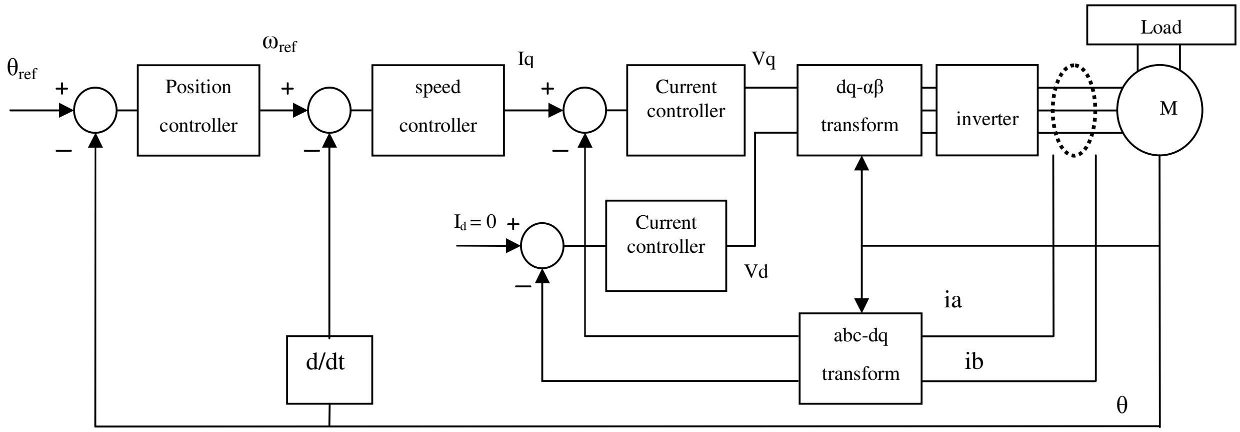

Fig. 1 Block diagram of the PMSM drive system.

3 MOTOR MODEL

The stator d, q axes voltage equations of the PMSM in the synchronous rotating reference frame are given by:

$$V_q = r_s i_q + L_q \frac{d}{dt} i_q + L_d \omega_r i_d + \omega_r \lambda_m \tag{1}$$

$$V_d = r_s i_d + L_d \frac{d}{dt} i_d - L_q \omega_r i_q \tag{2}$$

For a uniform air gap motor (surface mounted PMSM), as in the case under study, the d-axis and q-axis inductance L_d and L_q are equal and can be denoted as L_s

$$V_q = r_s i_q + L_s \frac{d}{dt} i_q + L_s \omega_r i_d + \omega_r \lambda_m \tag{3}$$

$$V_d = r_s i_d + L_s \frac{d}{dt} i_d - L_s \omega_r i_q \tag{4}$$

Where

V_q and V_d are the q and d axis voltages (v)

i_q and i_d are the q and d axis currents (A)

r_s and L_s are the resistance and inductance per phase of stator winding respectively (Ω,H)

λ_m is the flux linkage established by the permanent magnet as viewed from the stator windings (wb)

ω_r is the rotor speed (rad/sec).

Using the technique of field orientation control of the PMSM, the d-axis current i_d is controlled to be zero to maximize the output torque [3, 7]. Thus, the motor torque expression can be given by the following equation:

$$T_e = \frac{3}{2} * \frac{P}{2} \lambda_m i_q \tag{5}$$

The relation between the electromagnetic torque T_e and the rotor speed ω_r is expressed as the following:

$$T_e = J \frac{d\omega_r}{dt} + B_m \omega_r + T_L \tag{6}$$

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

Where:

J	is the inertia of the rotor (kg.m^2)
B_m	is the damping coefficient (N.m.s/rad)
T_L	is the load torque (N.m)
θ_r	is the rotor speed (rad)
P	is the number of poles

4 FUZZY LOGIC CONTROLLER

The fuzzy logic is considered as a mathematical theory combining multi-valued logic, probability theory, and artificial intelligence to simulate the human approach when solving various problems by relating different data sets to make decisions. It has been reported that fuzzy controllers are more robust to parameter changes than conventional PI or PID controllers and have better noise rejection capabilities [8].

A standard FLC is usually defined by a set of fuzzy parameters which specify any control action to be taken for a given process state. The FLC has two crisp inputs, error (e) and change of error (ce), and one crisp output, (out). FLC consists of three stages: fuzzification, rule base execution and defuzzification. In the first stage, the crisp variables (e), (ce) and (out) are converted into fuzzy variables E, CE and OUT. Each universe of discourse is divided into odd number of fuzzy sets; seven fuzzy sets are used in this paper.

Each fuzzy variable is a member of the subsets with a degree of membership ranging from 0 (non-member) to 1 (full-member). In the rule base execution stage, the fuzzy variables E and CE are processed by an inference engine that executes a set of control rules contained in the rule bases. Different inference algorithms can be used to produce the fuzzy set values for the output fuzzy variable OUT. The min-max inference algorithm is used in this investigation. In the last stage, defuzzification, the inference engine output variable is converted into a crisp value (out). Different defuzzification algorithms can be applied. The centroid defuzzification algorithm is used in this work. Due to its simplicity, the Mamdani type fuzzy systems are used in the design of FLC.

5 GENETIC ALGORITHM

The Genetic algorithm is a method for solving optimization problems that are based on natural selection (the process that drives biological evolution). The main advantages of the GA over other conventional optimization techniques are summarized as follows [9-13]:

1) GA technique uses a population of trials representing possible solutions of the problem, not a single point. As a result, the GA will be less susceptible to getting trapped on local minima.

2) GAs use probabilistic rules to make decisions when solving problems

3) GAs apply a performance index assessment to guide the search in the problem space.

A flow-chart of the GA algorithm optimization procedure is given in Fig. 2.

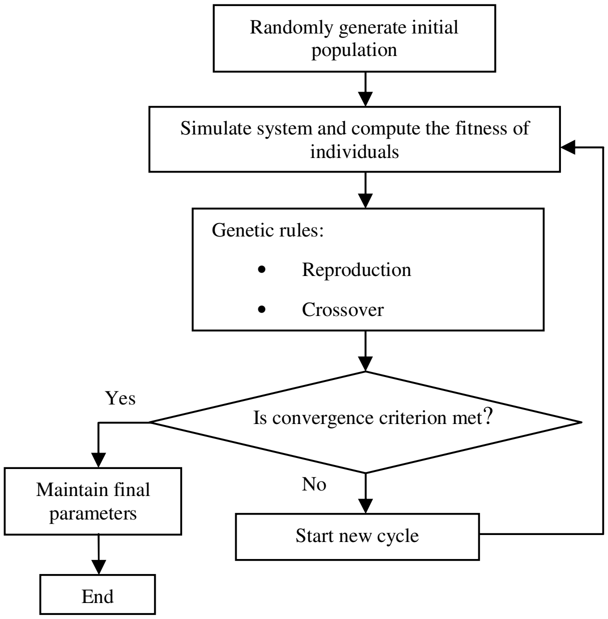

Fig. 2 Flow-chart of the general GA

Individual solutions are initialized, then, the algorithm repeatedly modifies these solutions until a predefined convergence criterion is met. The convergence criterion of a genetic algorithm is a user-specified condition. In general, the genetic algorithm uses three types of rules at each step to create the next generation from the current population; reproduction, crossover, and mutation. Details of these rules are listed in [10, 12].

The individuals are encoded in either binary or real numbers. Binary encoded numbers have many drawbacks. They take long time of calculation since they need the real values to be converted to binary and at the end of each genetic algorithm cycle the binary numbers are converted back to real ones. In addition, dealing with binary numbers may affect the precision and conversion process. These problems affect the accuracy of the algorithm and as a result, real numbers are chosen in this work.

6 APPLICATION OF GENETIC ALGORITHM FOR TUNING PI CONTROLLER

The described genetic algorithm that was applied successfully in [12] is used as a method of tuning PI controller parameters of the position controller in this paper. The individuals are the parameter gains of the PI controller that are selected in a certain interval. The gains are updated during each generation achieving adaptive control which is the main difference as compared to conventional controller that has constant gains. The population size, number of individuals is chosen as ten. Although bigger population size results in more accurate but this need more time of calculations. Therefore, we see that population size of ten is a suitable choice in our study. Notice that random distribution of individuals is not suitable when applying them to the system because changing the controller parameters of the outer loop randomly will result in random changes in the reference speed. As a result this paper proposes a triangular distribution of controller parameters to be applied to the system in successive short time intervals. The objective (fitness) function is selected as the absolute of error and the target is to minimize this function.

The operation of genetic algorithm based PI controller is explained by the following steps:

1- Initialize a population of K_p and K_i having a size of '10'. i.e. K_p (n) and K_i (n) for n = 1 to n = 10. Define a period of time equals 'ts', an objective function as absolute of error abs (error), and a constant $R \in]0,1[$

2- Set n = 1 and apply K_p (n) and K_i (n) to the system for a period of 'ts'

3- At the end of 'ts' catch the value of error and hold it as abs (error). Then increment 'n' by one and apply K_p (n) and K_i (n) to the system for a period of 'ts'.

4- Repeat step '3' until 'n' reaches 10.

5- Search for the minimum value of the objective function for n = 1 to n = 10 and catch the corresponding values of K_p and K_i. For example if the objective function is minimum when n = m then catch $K_p(m)$ and $K_i(m)$ and use them in the next generation. This is the reproduction stage.

6- Apply the crossover operation to the rest of $K_p(n)$ and $K_i(n)$. Again if the objective function is minimum when n = m, then apply the following formula [10, 12] to calculate the crossover individuals (new modified $K_p(n)$ and $K_i(n)$ that will be used in the next generation for n = 1 to n = 10 and $n \neq m$) .

$$K_p(n) = R * K_p(m) + (1-R) * K_p(n) \qquad (8)$$

$$K_i(n) = R * K_i(m) + (1-R) * K_i(n) \qquad (9)$$

7- Check meeting the conversion criterion then if it is not met repeat steps 2-6 otherwise stop genetic algorithm and maintain the final results of the fittest values of K_p and K_i and apply them continuously to the system. In the system under study, the range of PI parameters is well known, therefore mutation rule may be eliminated.

7 SIMULATION RESULTS

The combinations of controllers are applied to a loaded PM drive system as a plant (given in section 2). The reference position is a step function having a final value of 6 rad and the load torque is 1 N.m. The motor parameters are given in the Appendix. The sampling time of the inner loop is set to 200 μ sec, that of the intermediate to 1 m sec, and of the outer loop to 5 m sec. Simulation has been carried out using MATLAB Simulink.

Case 1: Design of the conventional PI controllers is carried out using Ziegler-Nichols method [12, 14]. Figures 3-5 show the simulation results, notice that there is a torque ripple in T_e, and the settling time is longer compared to the results of the later cases. The outer PI controller parameters have fixed values of 10 and 0.1 for K_p and K_i respectively.

Case 2: FLC gives very good performance compared with PI controllers but this controller suffers from the large calculations that have to be carried out. Therefore, only the position controller is chosen as FLC and the speed and current controllers are selected to be conventional PI controllers. Figures 6-8 show simulation results in this case. The steady state error of the output position, the percentage toque ripples and the settling time are reduced compared with case one.

Case 3: Better performance is achieved as illustrated in Figures 9-11 but larger time of implementation of FLCs (both speed and position loops) slows on-line control. To overcome this problem, the fuzzy sets may be transformed to real values € [-1,1] as shown in Table I. This transformation decreases the calculation requirements but on the other hand will increase rise and settling time. In case of using fixed values as in Table I, those values have to be on-line tuned but will result in excessive execution time [15].

Case 4: The proposed controller is simple to implement and gives good performance at the same time. The proposed controller is applied to the outer loop (position controller) since this controller has the dominant effects on the performance and the periods between successive samples is long enough for the optimization process to be carried out. Transfer function of the controller is basically PI form with adaptive parameters through GA. Vector of initial population for K_p and K_i are given in Table II. Speed and current controllers are kept as conventional PI controller for the aim of simplicity.

TABLE I.

TRANSFORMATION FROM FUZZY SETS TO REAL VALUES

NB	NM	NS	ZE	PS	PM	PB
-0.9	-0.6	-0.3	0.0	0.3	0.6	0.9

TABLE II.

INITIAL VALUES OF K_P AND K_I

K_p	19	21	23	25	27	26	24	21	18	16
K_i	0.12	0.14	0.16	0.18	0.2	0.19	0.17	0.15	0.13	0.11

Figures 12-14 show simulation results in case 4. It is noticed that applying genetic algorithm modifies the system behavior by reducing the rise time and settling time as compared to case 1 and very close to that of cases 2 and 3 as shown in Fig. 15. The results show the superiority of the proposed method in the reduction of the algorithm and consequently the execution time. Steady state error is zero using the proposed controller. Variations of proportional and integral gains are illustrated in Fig. 16 and Fig. 17 which result due to triangular variations of the position controller gains to avoid any sudden changes in the reference speed. The developed torque has a percentage of torque ripples which can be minimized in a future study.

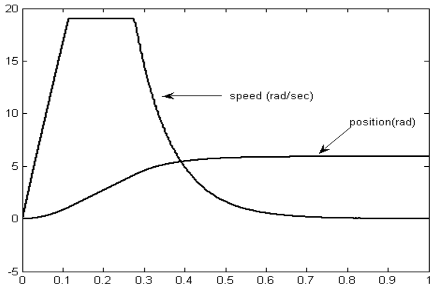

Fig. 3 case 1: Rotor position and speed, t (sec).

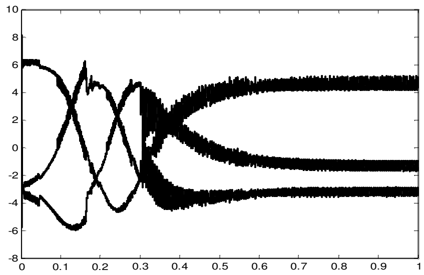

Fig. 4 case 1: Phase currents (A) –t (sec).

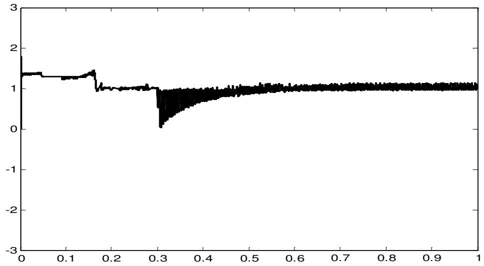

Fig. 5 case 1: Developed torque (N.m) – t (sec).

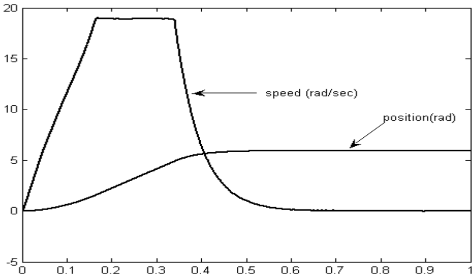

Fig. 6 case 2: Rotor position and speed, t (sec).

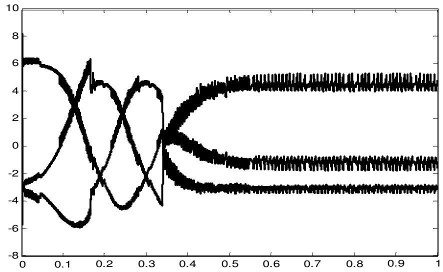

Fig. 7 case 2: Phase currents (A) – t (sec).

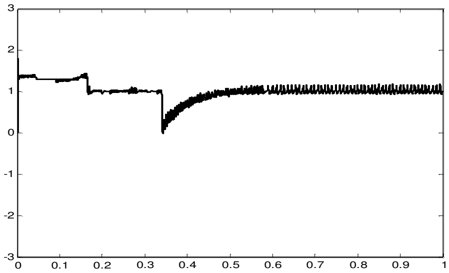

Fig. 8 case 2: Developed torque (N.m) – t (sec).

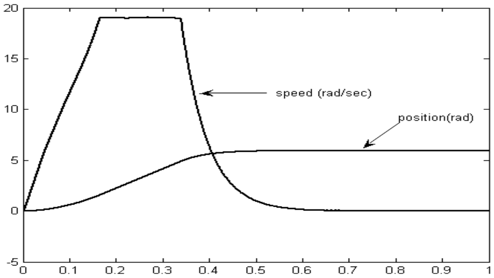

Fig. 9 case 3: Rotor position and speed, t (sec).

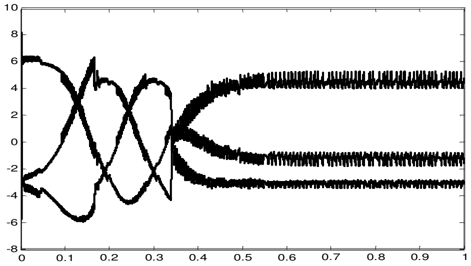

Fig. 10 case 3: Phase currents (A) –t (sec).

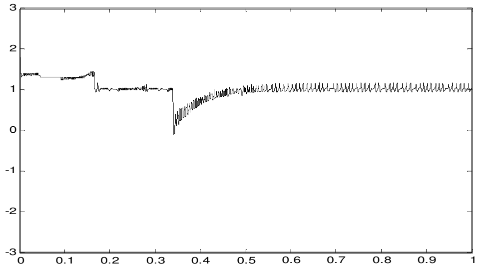

Fig. 11 case 3: Developed torque (N.m) – t (sec).

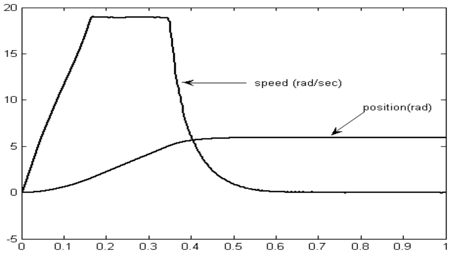

Fig. 12 case 4: Rotor position and speed, t (sec).

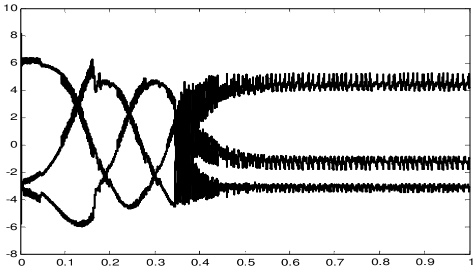

Fig. 13 case 4: Phase currents (A) –t (sec).

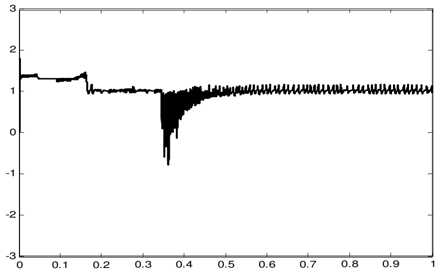

Figure 14 case 4: Developed torque (N.m) – t (sec).

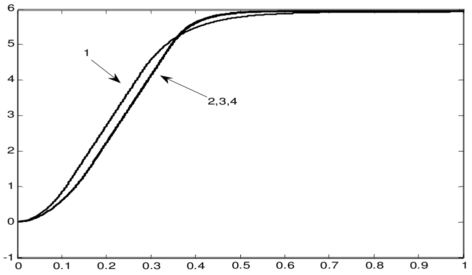

Fig. 15 cases 1, 2, 3, and 4: Position θ (rad) – t (sec).

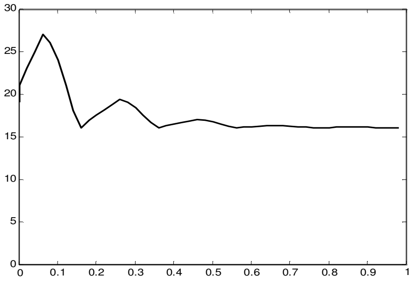

Fig. 16 Variation of proportional gain k_p during operation.

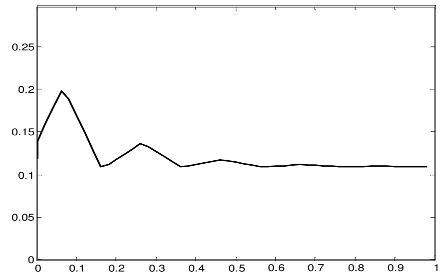

Fig. 17 Variation of integral gain k_i during operation.

8 CONCLUSIONS

Combinations of controllers are applied to cascaded control system to drive a permanent magnet synchronous motor in servo application. The controllers include conventional PI controller, fuzzy logic controller and PI controller with on-line tuning using genetic algorithm. Tuning of PI controller parameters based on genetic algorithm is proposed and applied to the position controller of a PMSM. Triangular distribution of controller gains is introduced to avoid any sudden changes in the reference speed which is the output of this controller. The controller is simple to implement compared with FLC. Simulation results show good performance using the proposed algorithm in rise time, settling time and steady state error.

APPENDIX

Motors parameters

Power: 1.7 kW Frequency: 150 Hz

Line Voltage: 380 V Line Current: 3.4 A

No. of Poles: 6 Rated Speed: 3000 rpm

Stall Torque: 5.4 Nm Torque Constant: 1.6 N.m/A

Rs: 2 Ω /phase Ls: 7.75 mh/phase

REFERENCES

[1] Krause P. C. and Wasynczuk O.: Electromechanical motion devices, McGraw-Hill, 1989.

[2] Rahman A. and Zhou M. A: Analysis of brushless permanent magnet synchronous motors, IEEE Trans. Ind. Elect., vol. 43, no. 2, April 1996, 256-267.

3] Khater F.M.H., Ahmed F.I., and Abu-Elsebah M.I.: Design and implementation of advanced PMSM drive system, in Proc. of Int. Symposium on Flexible Automation, Osaka, Japan, July 10-12, 2006, 367-374.

[4] Lin F. J.: Real-Time IP position controller design with torque feedforward control for PM synchronous motor, IEEE Trans. Ind. Elect., vol. 44, no. 3, June 1997, 398-407

[5] Su W. T. and Liaw C. M. : Adaptive positioning control for a LPMSM drive based on adapted inverse model and robust disturbance observer, IEEE Trans. Power Elect., vol. 21, no. 2, March 2006, 505-517.

[6] Ogasawara S., and Akagi H.: Implementation and position control performance of a position-sensorless IPM motor drive system based on magnetic saliency, IEEE Trans. Ind. Appl., vol. 34, no. 4, July/August 1998,806-812.

[7] Vas P.: Sensorless vector and d-axis torque control, Oxford University Press, May 1998.

[8] Reznik L.: Fuzzy controllers", Biddles Ltd, 1997.

[9] Genetic algorithm and d-axis search toolbox 2 user's guide, Copyright by the MathWorks, Inc, 2004.

[10] Man K. F., Tang K. S**.**, and Kwong S**.**: Genetic algorithms: concepts and applications, IEEE Trans. Ind. Elect., vol. 43, no. 5, October 1996, 519-534.

[11] Kristinsson K., and Dumont G. A.: System identification and control using genetic algorithms, IEEE Trans. Syst., Man, and Cybernetics., vol. 22, no. 5, Septemper/October 1992, 1033-1046.

[12] Khater F., Shaltout A., and Hendawi E.: Tuning of PI controller based on genetic algorithm, in Proc. Of the 2^{nd} International Conference on Advanced Control Circuits & Systems (ACCS'08), Cairo, Egypt, March 30 – April 1 2, 2008.

[13] Silva W. G., Acarnley P. P., and Finch J. W.: Application of Genetic Algorithms to the Online Tuning of Electric Drive Speed Controllers", IEEE Trans. Ind. Elect., vol. 47, no. 1,Febr uary 2000, 217-219.

[14] Jantzen J.: Tuning of fuzzy PID controller, Technical University of Denemark, Department of Automation, Tech. report no. 98-H 871, 1998.

[15] Chan P. T., Rad A. B., and Tsang K. M.: Optimization of Fused Fuzzy Systems via Genetic Algorithms , IEEE Trans. Ind. Elect., vol. 49, no. 3, June 2002,685-692.