NEW CONTROL APPROACH FOR PERMANENT MAGNET SYNCHRONOUS MOTOR

A. LOUKDACHE*, J. ALAMI, M. EL BELKACEMI AND A. EL IMRANI
Département de physique, Laboratoire Conception & Systèmes
Université Mohammed V, Faculté des Sciences- Agdal.
4, Avenue Ibn Battouta, B.P.1014 RP, Rabat.
MAROC

Abstract:- Permanent magnet synchronous motors are often used in electrical drives because of their simple structures, ease of maintenance and high efficiency. However, these motors have a nonlinear characteristic arisen from motor dynamics and load characteristics. To overcome this problem, a new speed controller based on genetic algorithms for the drive system is proposed in this paper. To illustrate the performance of the proposed controller, a conventional proportional integral controller is used as well to the speed control of permanent magnet synchronous motors. Simulations are realized by both control strategies and simulation results are presented and compared with the conventional control.

Key-Words :- Genetic algorithms, Permanent magnet synchronous motors, Optimal control, Adaptive control.

1. Introduction

Permanent magnet synchronous motors (PMSM) are of great interest especially for industrial applications in low-medium power range, since they have superior features such as compact size, high torque/weight ratio, high torque/inertia ratio and absence of rotor losses [1]. However, the performance of the PMSM is very sensitive to parameter variations and is spoiled due to external load disturbances in the system. The conventional controller design, i.e., Proportional-Integrator, is based on mathematical model of the plant, which may often be unknown, less defined, nonlinear, complex and multivariable with parameter variation. Thus, the conventional PI controller is not an all-purpose solution for any motor drive applications.

To overcome these problems, several control strategies such as fuzzy logic control [2], sliding mode control [3] and artificial neural network [4] have been proposed for speed and position control of PMSM. Recent literature has also explored the potentials of the Genetic Algorithms (GA) for motor drive applications [3], [4], [5] and [6].

As an intelligent control technology, the GA can give robust adaptive response of a drive with nonlinearity, parameter variation and load disturbance effect; where an exact mathematical model of system cannot be obtained at all [5].

In this paper, a new Proportional-Integrator corrector based on Genetic Algorithms (PIGA) is designed for speed control of PMSM. In fact, the PIGA speed controller is applied to the speed loop by replacing the conventional PI speed controller. An automatic tuning process using GA is used to optimize the conventional PI parameters. Only two parameters are sought but tuning is complicated by a significant nonlinearity caused by saturation of the speed controller. This means that optimum controller settings depend on the form of the required speed demand.

This paper is organized as follows. The section 2 gives an overview of PMSM modelling. The genetic algorithms are presented in section 3. Then, the principles of the proposed speed controller based on GA are described in section 4. Finally, the section 5 displays the simulation results obtained using the improved speed controller which is compared with results obtained by the conventional one.

2. Modeling of PMSM drive system

The configuration of PMSM drive system is given in Figure 1. The drive system is composed of speed controller (PIGA or conventional PI), a current regulator, a hysteresis band current controller, a three phase PWM inverter and a position encoder.

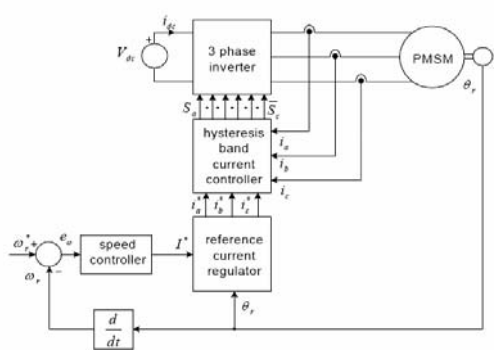


Fig.1. Block diagram of PMSM speed drive system.

Figure 2 presents an equivalent circuit of PMSM and the 3-phase inverter.

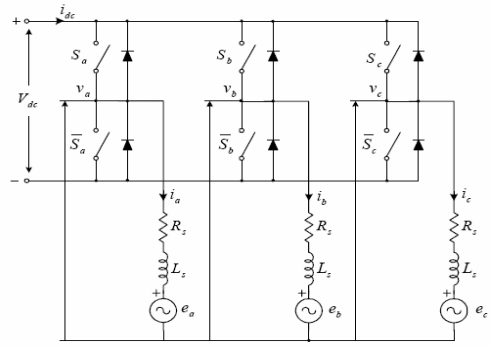


Fig. 2. Equivalent circuit of PMSM and inverter.

Where θ_r is rotor position, ω_r actual speed, i_a^*, i_b^*, i_c^* reference phase currents and e_w speed error. e_w is the difference between reference speed ω_r and actual speed ω_r. Using the speed error e_w, the speed controller generates I^* called reference current or control current.

The stator voltage equations of PMSM in matrix form can be represented as the statements of phase currents in (1). This equation can be shown in state-space form as in (2).

$$(1)\quad \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} L_s & 0 & 0 \\ 0 & L_s & 0 \\ 0 & 0 & L_s \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix}$$

$$(2)\quad \frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \begin{bmatrix} L_s & 0 & 0 \\ 0 & L_s & 0 \\ 0 & 0 & L_s \end{bmatrix}^{-1} \left\{ \begin{bmatrix} -R_s & 0 & 0 \\ 0 & -R_s & 0 \\ 0 & 0 & -R_s \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} - \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} + \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} \right\}$$

The rotor speed and electrical torque can be written

$$(3)\quad \frac{d}{dt}\omega_r = \frac{p}{2}\left(T_e - T_L - B\left(\frac{2}{p}\right)\omega_r \right) \Big/ J$$

$$(4)\quad T_e = K_t I^*$$

Where, $K = \frac{-3p}{4}\lambda_f$, and λ_f is the flux due to the permanent magnet rotor. The function of hysteresis band current controller is given in (5).

$$(5)\quad h_x = \begin{cases} 1 & if \quad i_x^* - i_x \le 0.5 h_{rb} \\ 0 & if \quad i_x^* - i_x \ge -0.5 h_{rb} \end{cases}$$

Where x represents respectively a, b and c. hx represents the functions of hysteresis band current controller h_a, h_b and h_c. h_{rb} is the range of hysteresis band current controller. Using the obtained functions of hysteresis band current controller, (6) is obtained as follows:

$$(6)\quad \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \begin{bmatrix} L_s & 0 & 0 \\ 0 & L_s & 0 \\ 0 & 0 & L_s \end{bmatrix}^{-1} \left\{ \begin{bmatrix} -R_s & 0 & 0 \\ 0 & -R_s & 0 \\ 0 & 0 & -R_s \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} - \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} + \begin{bmatrix} \frac{(2h_a - h_b - h_c)}{3} \\ \frac{(-h_a + 2h_b - h_c)}{3} \\ \frac{(-h_a - h_b + 2h_c)}{3} \end{bmatrix} [V_{dc}] \right\}$$

3. Genetic algorithms (GA)

GA are computational models that emulate evolutionary behaviour of biological systems to solve optimization problems. The GA comprise a set of individual elements (population) and a set of biologically inspired operators. According to the evolutionary theory, the population evolves towards increasingly better regions of the search space by means of selection, crossover and mutation processes [7] and [8].

The GA are generally initialized with a randomly generated population (P(t=0)) of chromosomes. At each iteration t, the selection process is applied to the current population (P(t)) to create an intermediate one. Pairs of parents are then randomly chosen for reproduction via a crossover procedure that mimics biological mating. Information between the two parents is exchanged and swapped to create two new offspring. To ensure that all points in the search space can be reached, the mutation operator is used to randomly alter the values of the new offspring by adding some small perturbations [9] and [10].

The GA principles are given in figure 3.

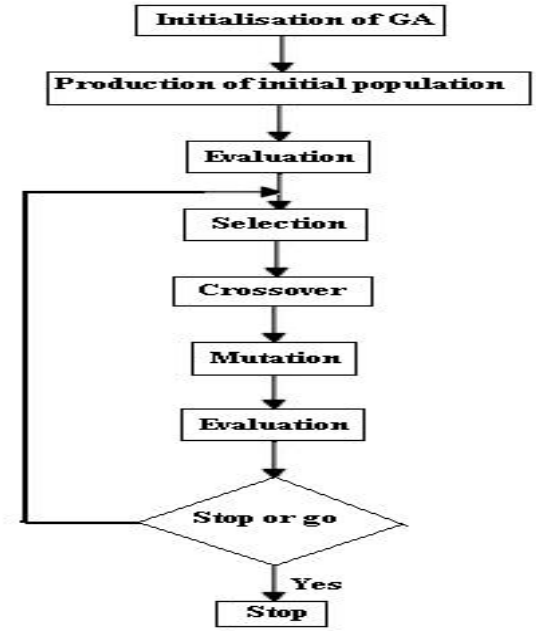


Fig. 3. Genetic Algorithm process

The GA performs a parallel search of a parameter space by using genetic operators to manipulate a set of encoded chromosomes which represents system parameters. The operation of the GA changes slightly depending on the base of the numbers to apply the genetic operators (crossover, mutation, reproduction, elitism). Traditionally GA's have been designed to operate over binary numbers (0, 1) and more recently there have been several decimal numbers (0,..,9). The fitness of each of the members of the population is calculated using a fitness function that characterizes how well each particular member solves the given problem [11].

The GA have found application in the area of the automatic tuning process for conventional and intelligent controllers. Same research has been conducted using genetic algorithms to help online or off-line control systems [12]. It has primarily been utilized as an off-line technique for performing a directed search for the optimal solution to a problem. In this paper, the GA is used on-line in real-time controller implementation to adaptively search through a population of controllers and determine the member most fit to be implemented over a given sampling period.

4. The improved PI controller based on GA (PIGA) for the PMSM drive

4.1 PMSM drive bloc scheme

The control strategy of PMSM using PIGA controller is shown in Fig. 4, where:

ω^* is the reference speed, $\omega(k)$ is the measured speed,

and $e_\omega(k) = \omega^* - \omega(k)$ is the speed error.

The control law is defined by the following equation:

$$F(K_p, K_i) = \alpha_1 \cdot e_w^2(k) + \alpha_2 \cdot (K_p \cdot e_w(k) + K_i \cdot e_w(k) \cdot T)^2 \quad (7)$$

Where α_1, α_2 represent the importance weight of the first and the second terms of (7) respectively, T is the sample time, K_p and K_i are the gains of the PI controller.

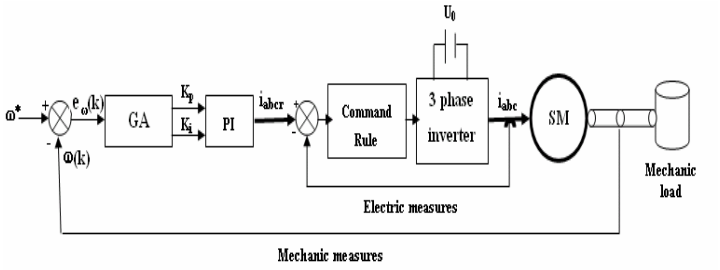


Fig. 4. The PIGA controller for the PMSM drive.

This strategy shows that the GA bloc receives the speed error e_ω and provides the optimal parameters (K_p and K_i) for the PI bloc. This bloc exploits these parameters in order to generate the optimal reference currents i_{abcr}. Then, the currents loop, composed of a command rule and a 3-phase inverter, provides the optimal currents i_{abc} that will be used by the SM bloc to reach the required speed ω^*.
In this application, feedback signals are the position θ and the phase currents. The position signal is used to calculate the speed. The switching signal generator is used to control turn-on angle θ_{on}, turn-off angle θ_{off} and pulse width modulation duty cycle.

4.2 Implementation of GA to the PMSM drive

The PIGA and the conventional PI controller are simulated using Matlab R2006a and simulation results are compared with each other.

In the first time, the reference electric speed is defined by a trapezoidal repeating sequence with a mechanical load torque constant (0 N.m and then 4Nm). In the second case, the reference electric

speed is constant and equal to 700 rd/s; the mechanical load torque varied between 0 and 8Nm.

The next section presents the simulation results related to both cases.

- Stator resistance: Rs= 2.875 Ω

- Inductance Ld= Lq= 8.5 e-3H.

- Flux induced by magnets= 0.175wb.

- Inertia= 0.8 e-3kg.m2, friction factor= 0 N.m.s and pair of poles= 4.

The configuration of GA parameters is given as follow:

1- Population size: the first step of genetic algorithm is to create a population. The population size used is 80. Individuals of the population are represented by a real valued number.

2- Variable bounds: the genetic algorithms are used to optimise the gains (K_p *and* K_i) of the PI controller. There are going to be two strings assigned to each member of the population, these members will be composed of a string that will be evaluated throughout the evolution of the GA. The initial range of the first parameter is [50, 80] and the second one belongs to [1, 10].

3- Select function: tournament selection operator is used to select the fittest individuals, which minimise the performance function given in (7), to form the next generation.

4- Crossover operator: arithmetic crossover was chosen as the crossover procedure. Single point crossover is too simplistic to work effectively on a chromosome with two alleles; a more uniform crossover procedure throughout the chromosome is required. The arithmetic crossover procedure is specifically used, with a rate of 0.08, for floating point numbers and is the ideal crossover option for use in this work.

5- Mutation operator: the uniform mutation function is used as the mutation operator, and the mutation rate is 0.01.

5. Simulation results and discussion

The PIGA and the conventional PI controller are simulated using Matlab R2006a and simulation results are compared with each other.

In the first time, the reference electric speed is defined by a trapezoidal repeating sequence (figure 5) with a mechanical load torque constant (0 N.m and then 4Nm). In the second case, the reference electric speed is constant and equal to 700 rd/s; the mechanical load torque varied between 0 and 8Nm (figure 6).

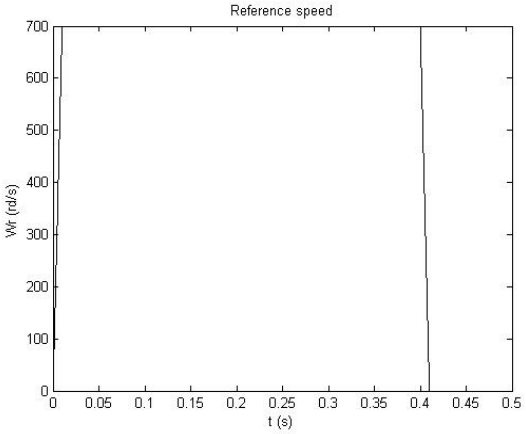


Fig 5. Electric speed reference

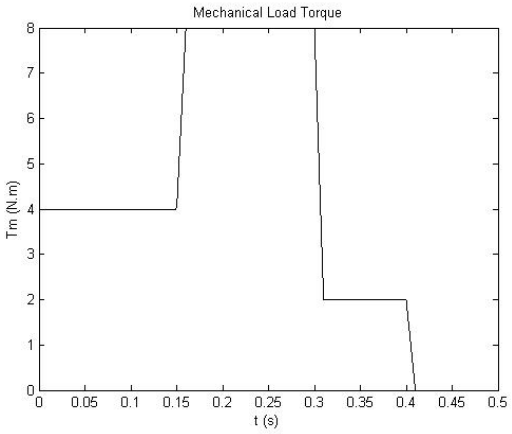


Fig 6. Mechanical load torque

The next section presents the simulation results related to both cases.

5.1 Case 1

In this study case, the reference electric speed is defined by a trapezoidal repeating sequence (figure 5) with a mechanical load torque constant (0 N.m and then 4Nm).

Figure 7 and 8 present the simulation results related to both control strategies (for Tm=0Nm and Tm=4Nm respectively), i.e., conventional PI (figure 7.a, figure 8.a) and PIGA (figure 7.b, figure 8.b). These results display the electric speed response time.

Figure 7.a shows that the electric speed response time for Tm=0Nm, related to the conventional PI, is $t_S \approx 0.025$s while at $t_S \approx 0.018$s , the speed response time is reached using PIGA (figure 7.b). Hence, the PIGA is 139% faster than conventional PI.

It is obvious that the performance of PIGA in electric speed response is better than the conventional PI controller in this case.

In addition, to illustrate the performance and the efficiency of the proposed controller, figures 9 and 10 present the electromagnetic torque response; figures 11 and 12 present the phases currents response provided by the two controllers.

Electromagnetic torque response (for Tm=0Nm) depicted in figure 9.a, given by the conventional controller, shows that oscillations are reduced at 0.025s, whereas with PIGA, oscillations are attenuated at 0.018s (figure 9.b). In this case, the time rate conventional PI / PIGA is 139%. For a mechanical load torque Tm=4Nm, the time rate conventional PI/GA is 166%.

Note that the phase's currents response reflects the electromagnetic torque response and vice versa (Figure 11 and 12).

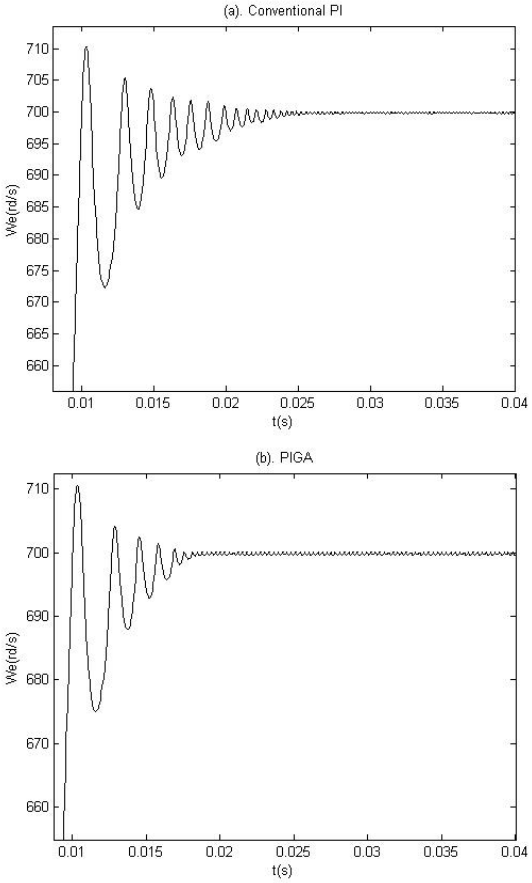


Fig 7. Electric speed response for Tm=0Nm

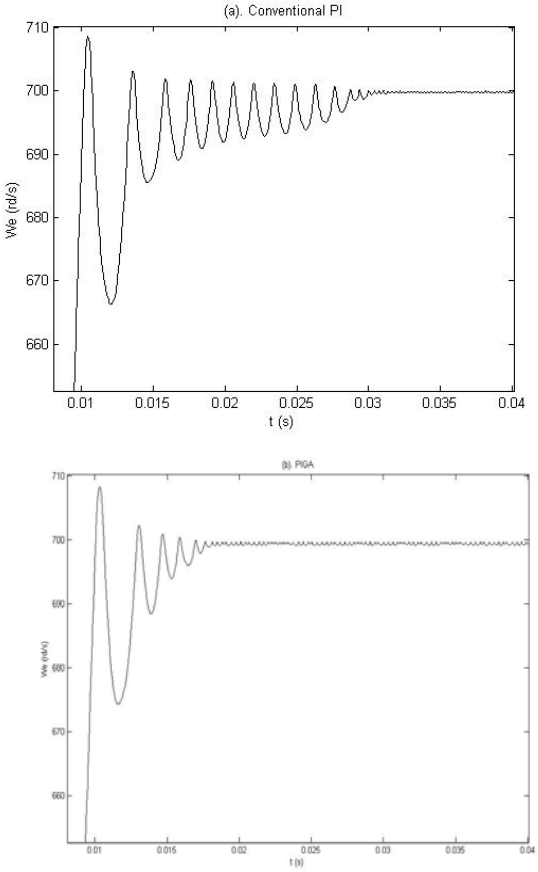


Fig 8. Electric speed response for Tm=4Nm

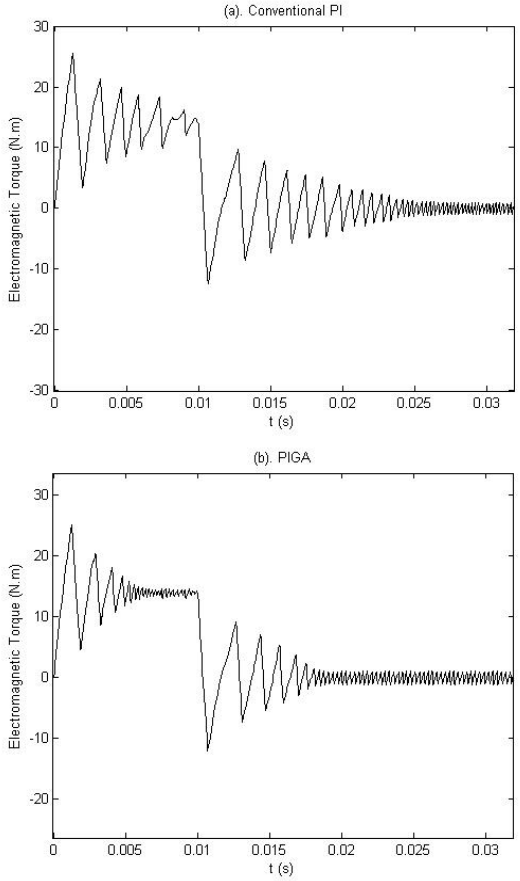


Fig 9. Electromagnetic Torque response for Tm=0Nm

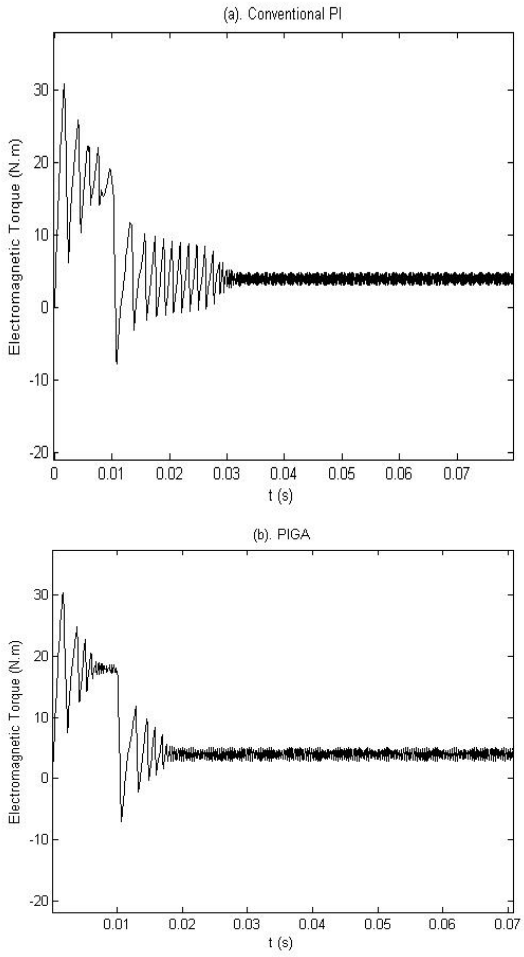
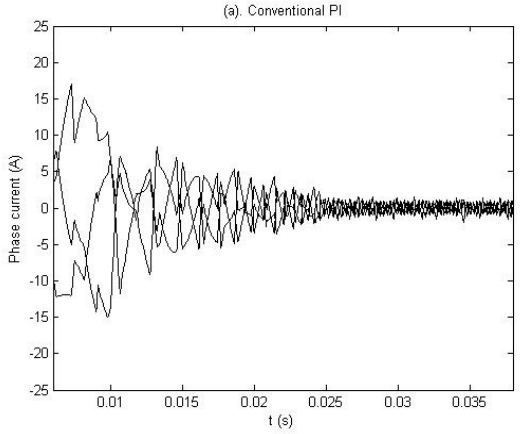


Fig 10. Electromagnetic Torque response for Tm=4Nm



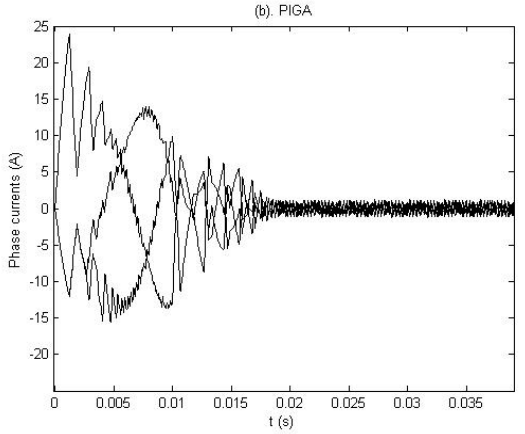


Fig 11. Phase's currents response for Tm=0Nm

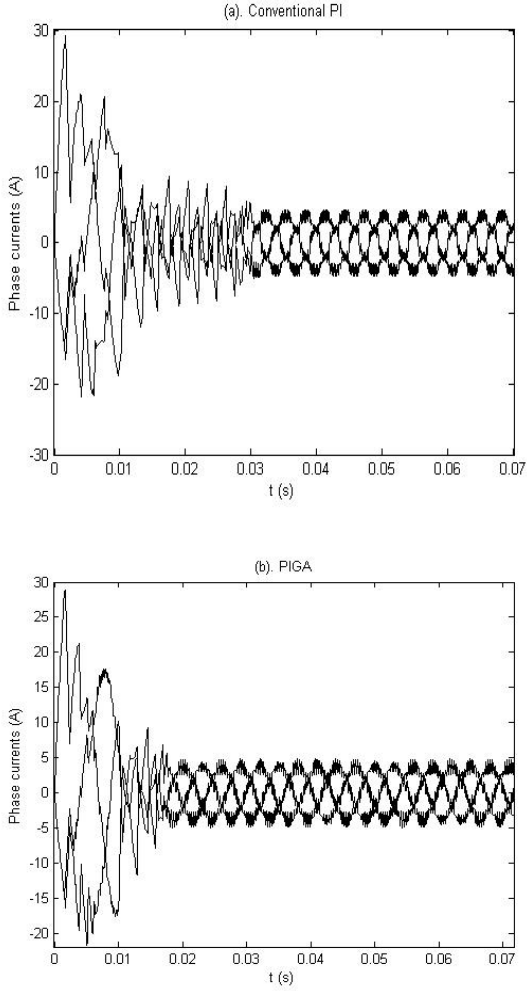


Fig 12. Phase's currents response for Tm=4Nm

5.2 Case 2

For this case, the reference electric speed is fixed to 700 rd/s with a mechanical load torque varied between 0 and 8Nm (figure 6).

Figure 13 represents the speed response obtained using both controllers ((a) conventional and (b) PIGA).

These figures show that PIGA is more appropriate than conventional PI, especially when starting the MSAP ($0 \leq t \leq 0.02s$), in terms of stability and response time.

Figure 13.a show that the electric speed response time related the conventional PI is $ts \approx 0.022s$ while at $ts \approx 0.014s$, the electric speed response time is reached using PIGA (figure 13.b). Hence, the PIGA is 157% faster than conventional PI.

Figure 14 and 15 illustrate the electromagnetic torque response and the phase's currents response respectively. These figures confirm the results mentioned above.

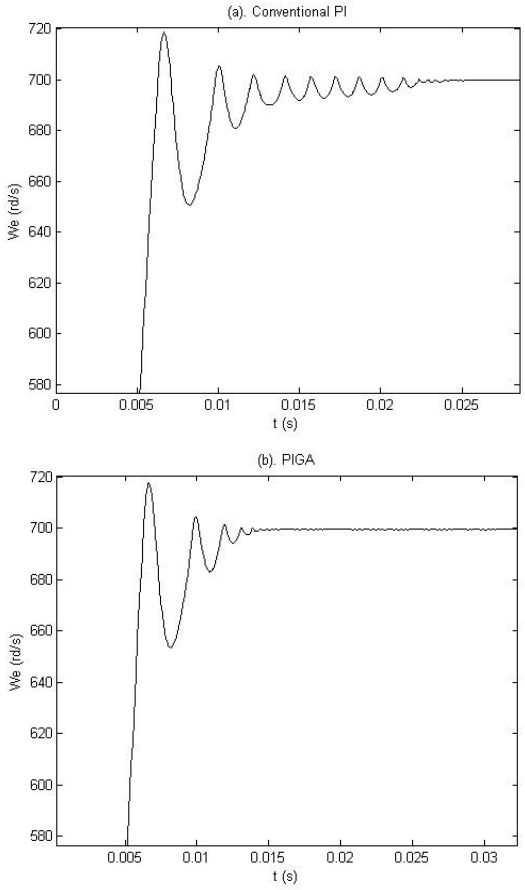


Fig 13. Electric speed response

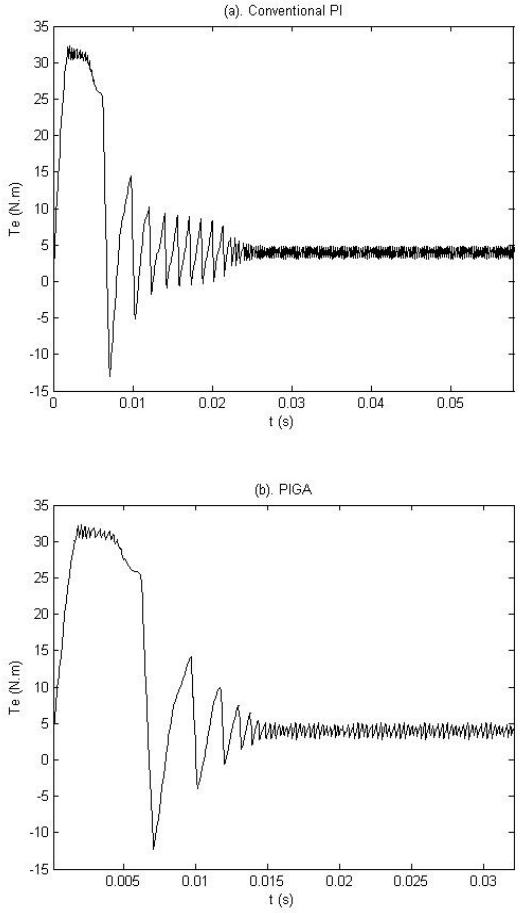
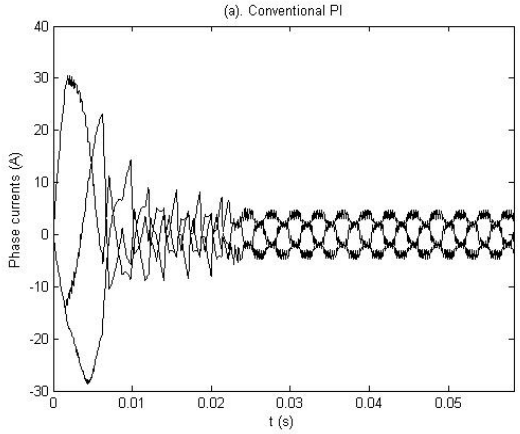


Fig 14. Electromagnetic Torque response



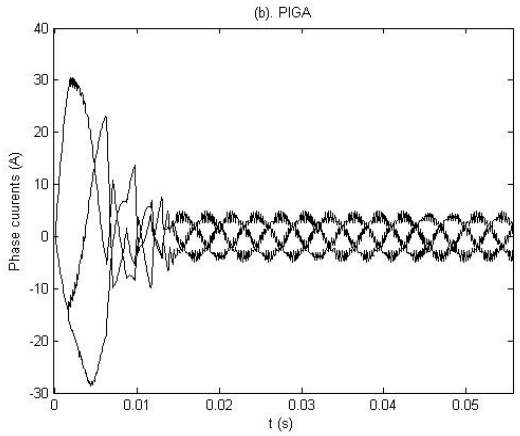


Fig 15. Phase's currents response

6. Conclusion

In this paper the speed control problem of Permanent magnet synchronous drive is studied. The nonlinear behaviour of the system, non-matched perturbations, parameter variations and load torque disturbance limit the performances of classical linear controllers used for this purpose. To overcome this problem, a new control strategy based on genetic algorithm is proposed.

As an intelligent control technology, the GA can give robust adaptive response of a drive with nonlinearity, parameter variation and load disturbance effect. In this study, a PIGA speed controller for PMSM drive system is presented. A mathematical model of a PMSM fed by three phase PWM inverter is implemented. Then the PIGA and conventional PI controller are simulated and results are given.

Results obtained using the PIGA controller to the PMSM are compared to those obtained by the conventional PI controller. With PIGA control, convergence is certainly and rapidly obtained in real time without insensitivity to perturbations applied to the PMSM. However, using the conventional PI, convergence is occasionally reached and depends generally on a fine tuning of parameters; therefore, this controller is not efficient in real time.

This work might validate the adaptation of GA to PMSM drive with low power. It will be interesting to extend it for alternative current machines (synchronous and asynchronous) with high power.

References

[1] Slemon G. R., 'Electrical machines for variable-frequency drives' , *Proceeding of IEEE,* Vol.82, No.8, pp. 1123-1139, 1994.

[2] Akcayol M. A., Cetin A., and Elmas C., 'An Educational Tool for Fuzzy Logic-Controlled BDCM', *IEEE Transactions on Education*, Vol. 45, No. 1, pp. 33-42, 2002.

[3] Karunadasa J. P. and Renfrew A.C., 'Design and implementation of microprocessor based sliding mode controller for brushless servomotor ', *IEE Proceedings-B*, Vol.138, No.6, pp.345-363, 1991.

[4] Rahman M.A. and Hoque M. A. 'On-line adaptive artificial neural network based vector control of permanent magnet synchronous motors ', *IEEE Trans. On Energy Conversion*, Vol.13, No. 4, pp. 311-318, 1998.

[5] Lee C.C., ' Fuzzy logic in control systems: Fuzzy logic controller', Part I, Part II, *IEEE Trans. on Syst., Man, and Cyber.*, Vol. 20, No.2, pp. 404- 435, 1990.

[6] Lin C.T. and Lee C.S.G., 'Neural-network-based Fuzzy logic control and decision system', *IEEE Trans. On Comp.*, Vol.40, No. 12, pp.1320-1336, 1991.

[7] El Imrani A. A., A. Bouroumi, M. Limouri and A. Essaid, 'A Coevolutionary genetic algorithms using fuzzy clustering', *Intelligent Data Analysis* 4 (3-4), 183-193, 2000a.

[8] Goldberg, D. E., 'Genetic algorithms in search, optimization and Machine learning' *Addison-Wesley Publishing* Inc, 1989.

[9] El Imrani A. A., A. Bouroumi, H. Zine El Abidine, M. Limouri and A. Essaid, 'A fuzzy clustering-based niching approach to multimodal function optimization', *Cognitive Systems Research*, vol. 1, Issue 2, 119-133, 2000b.

[10] Michalewicz, Z., 'Genetic Algorithms + Data Structures = Evolution Programs '. *Springer-Verlag,* New York, 1993.

[11] Lennon W.K., and Passino K.M., 'Genetic adaptive identification and control', *Engineering Applications of Artificial Intelligence*, Vol.12, No.2, pp.185-200., 1999.

[12] Texas Instruments, TMS320F240 DSP Controllers Evaluation Module, July, 1999.