Eng. & Tech. Journal ,Vol. 29, No.14, 2011

Particle Swarm Optimization for Adapting Fuzzy Logic Controller of SPWM Inverter Fed 3-Phase I.M.

Fadhil A. Hassan*& Lina J. Rashad*

Received on: 14 /2/ 2011
Accepted on: 8/9/ 2011

Abstract

According to the high performance demand of speed control of an induction motor, Fuzzy Logic Controller (FLC) gives superior behavior over wide range of speed variation. Fuzzy logic is a robust controller for linear and non-linear system, even if good mathematical representation of the system is not available. But, adapting fuzzy controller parameters is very complex and depends on operator experience. Particle Swarm Optimization (PSO) algorithm is proposed in this paper as an optimization technique for adapting centers and width of triangle inputs membership functions. The ordinary adapting method of FLC is represented too. Meanwhile, based on the concept of optimization, ways of defining the fitness function of the PSO including different performance criteria are also illustrated. The complete mathematical model and simulation of an induction motor and inverter are represented in this paper. The simulation results demonstrate that the proposed PSO-FL speed controller realizes a good dynamic behavior of the I.M with very good speed tracking.

Keywords: Particle Swarm Optimization, Fuzzy Logic Control, PSO Adapting of Fuzzy Parameters, Speed Control of Induction Motor.

أمثلية الحشد الجزيئي لتكييف المسيطر المضبب لعاكس تضمين عرض النبضة الجيبي المغذي للمحرك الحثي ثلاثي الطور

الخلاصة

وفقاً للأداء العالي المطلوب في السيطرة على سرعة المحرك الحثي ، يقدم المسيطر الضبابي المنطقي سلوكاً فائقاً في نطاق واسع من تغيير السرع. حيث أنه مسيطر متين للأنظمة الخطية واللاخطية حتى وإن لم يتوفر النموذج الرياضي الجيد للنظام. ولكن عملية تكييف براميترات المسيطر الضبابي معقدة جداً وتعتمد على خبرة المشغل. قـُدِّمت في هذا البحث خوارزمية تحقيق الأمثلية لأسراب الجزيئات كتقنية للتكييف الأمثل لمراكز وعرض دوال العضوية المثلثة لمدخلات المسيطر الضبابي، كما وأن طريقة التكييف العادية ممثلة أيضاً. وفي نفس الوقت وأستنادا على متطلبات تحقيق الأمثلية، تعريف دالة الكفاءه وعدد من معايير الأداء قد وُضِّحت أيضاً. إنّ النموذج الرياضي الكامل ومحاكاة المحرك الحثّي والعاكس مُمَثَّلةٌ في هذا البحث. تُبيّنُ نَتائِجُ المحاكاةَ بأنَّ تحقيقُ أمثلية أسراب الجزيئات المُقتَرَحه يحقق سلوكاً ديناميكياً جيداً في السيطره على سرعة المحرك الحثي مع تعقب جيد جداً للسرعة.

***Electrical Engineering Department, University of Technology/Baghdad**

1. Introduction

The design of a conventional control system is normally based on the mathematical model of a plant. If an accurate mathematical model is available with known parameters, it can be analyzed, for example, by a Bode or Nyquist plot, and a controller can be designed for the specified performance. Such procedures are tedious and time-consuming, and in addition of that for complex or non-linear system are difficult to obtain good mathematical model [1].

Fuzzy logic (FL) or Fuzzy Inference System (FIS) was invented in 1970s, which was powerful tool in the field of control applications. The fuzzy controller gives superior performance over the conventional control methods such as PID controller, even if the accurate mathematical model is not available [1, 2].

Unfortunately, adapting and choosing of the fuzzy controller parameters such as: number of membership functions, type of membership functions, number of rules, and formulating rules, are strongly depend on the experience of the operator and designer. In spite of experience, adapting position and width of the input and output memberships are very difficult and dependant parameters [2, 3].

Nowadays, several new intelligent optimization techniques have been emerged like: Genetic Algorithm (GA), Ant Colony Optimization (ACO), Simulated Annealing (SA), Bacterial Foraging (BF), and Particle Swarm Optimization (PSO) [3, 4]. Due to its high potential for global optimization, PSO has received great attention in control system such as the search of optimal PID controller and fuzzy controller parameters. PSO is one of the modern heuristics algorithms; it was developed through simulation of a simplified social system, and has been found to be robust in solving continuous non-linear optimization problems [4, 5].

In this paper, a tradition method for adapting fuzzy logic controller of non-linear three phase induction motor speed control is represented. Then, the PSO based method for adapting fuzzy controller parameters are proposed as a modern intelligent optimization algorithm.

Also, this paper represents the Sinusoidal Pulse Width Modulation (SPWM) inverter as a voltage fed inverter. Which is used as volt/Hz control drive.

2. Sinusoidal PWM Inverter

The pulse width modulation techniques have wide simple and complex types, the Sinusoidal Pulse Width Modulation (SPWM) is the most popular used method of A.C. drives, but it's not the efficient method [1, 6]. Voltage source inverter (VSI) should have a stiff source at the input [1], that is, its Thevenin impedance ideally is zero. Thus, a large capacitor can be connected at the input if the voltage source is not stiff. A practical (VSI) consist of power bridge devices with three output legs, each consist of two power switches and two freewheeling diodes, the inverter is supplied from D.C. voltage source via LC or C filter. In sinusoidal PWM the three output legs considered as three independent push-pull amplifiers [6] as shown in figure (1). The gating signals of each push-pull stage generated by comparing a constant level triangle signal of frequency (f_c) called "carrier signal", with 3-phase sinusoidal signals of frequency (f_r) called "reference signals", which has variable amplitude to get the desired output voltage, this comparison leads to generate a sequence of variable width pulses used to gating each switch in the push-pull stage. Figure (2) illustrates the principles of SPWM, The output phase voltage:

V_{ao} = is the output phase voltage measured to the center of the input D.C. voltage.

V_{an} = is the output phase voltage measured to isolated neutral of three-phase load such as induction motor.

Where:

$$\begin{bmatrix} V_{an} \\ V_{bn} \\ V_{cn} \end{bmatrix} = \tfrac{1}{3}\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} V_{ao} \\ V_{bo} \\ V_{co} \end{bmatrix} \ldots..(1)$$

$V_{ao} = 0.5mV_d \sin(\omega t)$+high-freq. ($M\,\omega_c \pm N\omega$) [1] ………… (2)

Where: V_d is the D.C. voltage, ω = fundamental frequency; ω_c=carrier frequency; M and N are integers and M+N =odd, m = modulation index is defined as [1, 6]:

$$m = \frac{V_r}{V_c} \qquad \ldots\ldots\ldots\ldots..(3)$$

Where: V_r = peak of reference signal, V_c = peak of carrier signal. At m= 1, the maximum value of fundamental peak = $0.5V_d$ which is 78.54% of the peak fundamental voltage of the square-wave ($2V_d/\pi$) which called the linear modulation region. To further increase the amplitude of the output voltage, the amplitude of the modulating signals exceeds the amplitude of the carrier signal which leads to enter into quasi-PWM region called "over modulating region" causing increase in the low order harmonics. Further increasing modulation index tends to obtain square wave at maximum possible output fundamental ($2V_d/\pi$) [1, 6].

3. SPWM Inverter Simulation

By using MATLAB/SIMULINK PROGRAM, the SPWM inverter can be simulated, firstly generation of the carrier triangle signal and the three modulating signals by using internal timer and the rated frequency (50 H_Z) to obtain the angular speed ($\omega_e t$), then multiplying the angular speed and the amplitude of the signal by the frequency command (f^*_{com}) and voltage command (V^*_{com}) respectively. Secondly, compared the two signal sets to generate the switching signals of three switches used as three push-pull devices. The output of the switches gives (V_{ao}, V_{bo}, V_{co}) then the three phase to load neutral (V_{an}, V_{bn}, V_{cn}) can be achieved by implementing equation (1). Figure (3) illustrated the complete simulation of SPWM inverter, and the output phase voltage can be shown in figure (4).

4. Modeling and Simulation of Three Phase I.M.

The synchronously rotating reference frame model for an induction motor is the most used mathematical representation, in which the induction motor can be looked on as transformer with moving secondary winding, where the coupling coefficients between the stator and rotor phases change continuously with the change of rotor position [1, 6]. The machine model can be described by differential equation with time varying mutual inductances, but such model tends to be very complex. Therefore, axis transformation is applied to transfer the three phase parameters (voltage, current and flux) to two–axis frame called (dq-axis stationary frame or park transformation). Park transformation is applied to refer the stator variables to a synchronously rotating reference frame fixed in the rotor, by such transformation the stator and rotor parameters rotate in synchronous speed and all simulated variables in the stationary frame appear as D.C. quantities in the synchronously rotating reference frame [1, 6].

The per-phase equivalent circuit diagrams of an I.M. in two- axis synchronously rotating reference frame are illustrated in figure (5). From the circuit diagram the following equations can be written [1]:

- Stator equation:

$$V_{qs}^{e}=R_s i_{qs}^{e}+\frac{d\Psi_{qs}}{dt}+w_e\Psi_{ds}.............(4)$$

$$V_{ds}^{e}=R_s i_{ds}^{e}+\frac{d\Psi_{ds}}{dt}-w_e\Psi_{qs}.............(5)$$

- Rotor equation:

$$V_{qr}^{e}=R_r i_{qr}^{e}+\frac{d\Psi_{qr}}{dt}+(w_e-w_r)\Psi_{dr}...(6)$$

$$V_{dr}^{e}=R_r i_{dr}^{e}+\frac{d\Psi_{dr}}{dt}-(w_e-w_r)\Psi_{qr}...(7)$$

Where:

V_{qs}^{e}, V_{ds}^{e}, V_{qr}^{e}, V_{dr}^{e} are the stator and rotor dq-axis voltages.

i_{qs}^{e}, i_{ds}^{e}, i_{qr}^{e}, i_{dr}^{e} are the stator and rotor dq-axis currents.

Ψ_{qs}^{e}, Ψ_{ds}^{e}, Ψ_{qr}^{e}, Ψ_{dr}^{e} are the stator and rotor dq-axis fluxs.

The superscript notation "*e*" referred to the synchronously rotating reference frame quantities.

It's obviously that in squirrel cage I.M V_{qdr}=0, then the pervious equation can be rewritten:

$$\frac{d\Psi_{qs}}{dt}=V_{qs}^{e}-R_s i_{qs}^{e}-w_e\Psi_{ds}....(8)$$

$$\frac{d\Psi_{ds}^{e}}{dt}=V_{ds}^{e}-R_s i_{ds}^{e}+w_e\Psi_{qs}....(9)$$

$$\frac{d\Psi_{qr}}{dt}=-R_r i_{qr}^{e}-(w_e-w_r)\Psi_{dr}....(10)$$

$$\frac{d\Psi_{dr}}{dt}=-R_r i_{dr}^{e}+(w_e-w_r)\Psi_{qr}....(11)$$

The development torque by interaction of air gap flux and rotor current can be found as:

$$T_e=(3/2)(P/2)\overline{\Psi_m}\ X\ \overline{I_r} \qquad ....(12)$$

Where: T_e is the electromagnetic torque.

By resolving the variables into d^e-q^e components:

$$T_e=(3/2)(P/2)(\Psi_{ds} i_{qs}^{e}-\Psi_{qs} i_{ds}^{e})$$

$$i_{ds}^{e}=\frac{\Psi_{ds}-\Psi_{qm}}{L_s} \qquad .....(15)$$

.....(13)

The dynamic torque equation of the rotor:

$$T_e=T_L+(\frac{2}{P})J\frac{dw_r}{dt} \qquad ......(14)$$

Where: ω_r = is the rotor speed; P: no. of poles; J= rotor inertia; T_L= load torque.

The stator current can be found by:

$$i_{qs}^{\ e} = \frac{\Psi_{qs} - \Psi_{dm}}{L_s} \qquad .....(16)$$

The air gap flux:

$$\Psi_{qm} = \frac{L_{m1}}{L_s}\Psi_{qs} + \frac{L_{m1}}{L_r}\Psi_{qr} \qquad ...(17)$$

$$\Psi_{dm} = \frac{L_{m1}}{L_s}\Psi_{ds} + \frac{L_{m1}}{L_r}\Psi_{dr} \qquad ....(18)$$

Where:

$$L_{m1} = \frac{1}{\left(\frac{1}{L_m} + \frac{1}{L_s} + \frac{1}{L_r}\right)} \qquad ....(19)$$

From the previous equations the dynamic model of an induction motor is simulated as shown in figure (6).

5. Speed Control of I.M

The closed loop control of A.C. drives is somewhat complex, and the complexity increases if high performance is demanded. There are wide studies and researches concern with several types of I.M. speed control. The main control techniques including scalar control, vector or field oriented control, direct torque and flux control, adaptive control, and intelligent control. The simplest and popular used method is scalar slip regulation of rotor speed, in which the actual rotor speed measured by means of tacho-meter [6, 7]. The difference between the command speed $(w_r^{\ *})$ and the actual rotor speed (w_r) gives the slip speed (w_{sl}) which added to the rotor speed through proportional-integral (PI) controller to obtain new speed command fed to the inverter. With step-up speed command the motor accelerates freely with a slip limit that corresponds to the torque limit, and then settles down to steady value. With step-down speed command the drive goes into regenerative or dynamic breaking mode and decelerates with constant negative slip $(-w_{sl})$.

6. Fuzzy Logic Control

Since fuzzy logic (FL) was invented, it had many successful applications mostly in control. One of the main advantages of fuzzy logic system is the design on the basis of incomplete and approximate information, thus providing simple and fast approximations of the unknown or too complicated models [1, 2].

The main idea of fuzzy control, which had proved to be a very successful method, is to build a model of human control expert who is capable of controlling the plant without thinking in terms of mathematical model. Usually the Mamdani method is used in adaptive fuzzy logic controller system. For example: if X & Y are the inputs of the fuzzy system, and "*F*" is the output signal:

IF X is A_1 *AND* Y is B_1 *THEN* $z=f_1$
IF X is A_2 *AND* Y is B_2 *THEN* $z=f_2$

The output "F" can be constructed as:

$$F = \frac{W_1}{W_1+W_2} f_1 + \frac{W_2}{W_1+W_2} f_2 \qquad ...... (20)$$

Where: A_1, A2, B1, B2 are the input membership functions, f_1 and f_2 are the output singleton membership functions, and W_1 and W_2 are the Degree Of Fulfillments (DOF) of rule 1 & 2, which can be adaptive to satisfied the input/output data [1, 2].

As mentioned before the selection of Membership Functions

(MFs) for the input and output variables and the determination of fuzzy rules are not always easy. There is no formal framework for the choice of the parameters of FLC and hence the means of tuning them and learning models in general have become an important subject of fuzzy control.

7. Design of FL Controller:

The function of the fuzzy controller is to observe the pattern of the speed loop error signal and correspondingly updates the control signal so that the actual speed (w_r) matches the command speed (w_r^*) . There are two input signals to the fuzzy controller: the error $E = w_r^* - w_r$, and the change of error *CE* (also it known as the future of error), which is related to the derivative of error *dE/dt*. A simple fuzzy logic controller of two inputs and one output can be used in this application, for each input three of triangle memberships was used, and five triangle memberships for the output. For such FLC nine (3*3) of *"If"* statement rules was used. Adapting of the input/output memberships positions are very difficult, especially for one don't have experience of the system behavior. Trial and error method can be used in such situations, in this work the final suitable distribution of the input/output memberships after several trial and error iterations can be shown in figure (7). The overall system simulation is illustrated in figure (8). System performance of different command speed, with fuzzy logic controller, under full-load condition is shown in figure (9).

Where, the used squirrel cage induction motor name plate is:

3-ph I.M, 380 v., 2.2 kw, 2 poles, 50 Hz, L_s= 13.6 mH, L_r= 11.4 mH, R_s=2.3 Ω, R_r= 3.4 Ω, rotor inertia= $4.5*10^{-3}$ kg/m^2.

8. Particle Swarm Optimization

PSO is one of the optimization techniques first proposed by Eberhart and Colleagues [8, 9]. This method has been found to be robust in solving problems featuring non-linearity and non-differentiability, which is derived from the social-psychological theory. The technique is derived from research on swarm such as fish schooling and bird flocking. In the PSO algorithm, instead of using evolutionary operators such as mutation and crossover to manipulate algorithms, the population dynamics simulates a "bird flocks" behavior, where social sharing of information takes place and individuals can profit from the discoveries and previous experience of all the other companions during the search for food. Thus, each companion, called particle, in the population, which is called swarm, is assumed to " *fly* " in many direction over the search space in order to meet the demand fitness function [4, 5, 9, 10].

For n-variables optimization problem a flock of particles are put into the n-dimensional search space with randomly chosen velocities and positions knowing their best values, so far (P_{best}) and the best position in the n-dimensional space [8, 10]. The velocity of each particle, adjusted accordingly to its own flying experience and the other particles flying experience. For example, the i_{th} particle is represented, as:,

$$x_i = (x_{i,1}, x_{i,2}, \ldots\ldots x_{i,n}) \quad \ldots.. (21)$$

In n-dimensional space, the best previous position of the i_{th} particle is recorded as:

$$P_{best_i} = (P_{best_{i,1}}, P_{best_{i,2}}, \dots \dots P_{best_{i,n}}) \quad ..(22)$$

The modified velocity and position of each particle can be calculated using the current velocity and distance from ($P_{best_{i,d}}$) to (g_{best_d}) as shown in the following formula [4, 8, 9, 10]:

$$v_{i,m}^{(It+1)} = W * v_{i,m}^{(It)} + c_1 * rand * (P_{best_{i,m}} - x_{i,m}^{(It)}) + c_2 * rand * (g_{best_m} - x_{i,m}^{(It)}) \quad \dots (23)$$

$$x_{i,m}^{(It+1)} = x_{i,m}^{(It)} + v_{i,m}^{(It)} \quad \dots (24)$$

i=1,2,…..,n

m=1,2,…..,d

Where;

n = Number of particles.

d = Dimension.

It. = Iterations pointer.

$V_{i,m}^{(It)}$ = Velocity of particle no. *i* at iteration It.

W = Inertia weight factor.

c1,c2 = Acceleration constant.

rand = Random number between 0-1.

$x_{i,m}^{(It)}$ = Current position of particle *i* at iteration It.

P_{best_i} = Best previous position of i_{th} particle.

g_{best_m} = Global best particle among all the particles in the population.

9. Implementing PSO for Adapting FL Controller

The implementation of particle swarm optimization in this work is same what complex, because the performance of the system must be examined in each iteration and particles position during the optimization algorithm. Therefore, the optimization algorithm is implemented by using MATLAB m-file program and linked with the system simulation program in MATLAB SIMULINK, to check the system performance in each iteration.

In this work, the problem is to optimize the position of the two inputs (*E*, *CE*) memberships. Each triangle membership is recognized by three parameters: left corner, right corner, and center. Therefore, for six memberships (of the two inputs) eighteen (6*3) parameters must be adapted. Thus, the particles (birds) have eighteen dimensions, or in other words particles must 'fly' in eighteen dimensional spaces. A random of 15 particles positions is assumed (for each dimension 15 birds), and optimization algorithm of 20 iteration is used to estimate the optimal positions of the two inputs memberships parameters.

In most intelligent optimization algorithms, there are commonly performance criteria such as: Integrated Absolute Error (IAE), the Integrated of Square Error (ISE), and Integrated of Time weight Square Error (ITSE). That can be evaluated analytically in frequency domain [4, 8].

Each criterion has its own advantage and disadvantage. For example, disadvantage of IAE and ISE criteria is that its minimization can result in a response with relatively small overshot but a long settling time, because the ISE performance criteria weights all errors equally independent of time. Although, the ITSE performance criterion can overcome the disadvantage of ISE criterion. The IAE, ISE, and ITSE performance criterion formulas are as follows:

$$IAE = \int_0^t |r(t) - y(t)|\,dt = \int_0^\infty |e(t)|\,dt \quad ..(25)$$

$$ISE = \int_0^t e^2(t)\,dt \quad ..(26)$$

$$ITSE = \int_0^t t * e^2(t)\,dt \quad ..(27)$$

In this paper, the integrated of time weight square error ITSE is used for evaluating the accuracy performance of the fuzzy controller. A set of good control parameters can yield a good step response that will result in performance criteria minimization in the time domain, this performance criterion is called Fitness Function (FF) which can be formulated as follows [5, 8]:

$$FF = (M_P + E_{ss}) * \beta + ITSE \quad .....(28)$$

Where:

M_P is maximum overshot.

E_{ss} is steady state error.

β is the weight factor can set to be larger than 0.7 to reduce the overshot and steady state error, also can be smaller than 0.7 to reduce the rise time and settling time.

The objective function is to minimize the fitness function FF. the PSO algorithm process can be summarized in the flowchart shown in figure (10). And the final obtained positions of the membership functions from the particle swarm optimization algorithm are illustrated in figure (11). System performance of PSO-based for different command speed is shown in figure (12).

10. Conclusions

A comparison performance between the proposed PSO method and the ordinary adapting method of the fuzzy controller membership functions is illustrated in figure (13), from which the following tips can be concluded:

i. The proposed PSO adapting method can be considered as: very computationally efficient in eliminate computing time, easy to implement, and simple concept. Unlike, the ordinary adapting method which needs long adapting time and complex procedure.

ii. In the ordinary adapting method, the fuzzy controller gives a satisfactory system performance: max. overshot = 4% in low seed & 0% in high speed, rise time = 6 sec, settling time = 6.5 sec, and steady state error = 0%.

iii. The proposed PSO adapting method gives a superior performance, in which: max. overshot = 0% in low and high speed variation, rise time = 3 sec, settling time = 3 sec, and steady state error = 0%.

iv. The proposed PSO adapting method gives very good speed tracking with non-sluggish performance and high robust controller than those obtained by the ordinary adapting of fuzzy logic controllers.

11. References

[1] Bimal K. Bose, “ *Modern Power Electronic and AC Drives*”, Prentice Hall, 2002.

[2] K. M. Passino and S. Yurkovich, *"Fuzzy control"*, Addison – Wesley Longman, 1998.

[3] T. Senjyu, A. Miyazato and K. Uezato, " *Enhancement of Transient Stability of Multi-Machine Power Systems by Using Fuzzy-Genetic Controller"*,

Journal of Intelligent and Fuzzy Systems, Vol. 8, 2000.

[4] A. Soundarrajan, Dr.S.Sumathi, C.Sundar *"Particle Swarm Optimization Based LFC and AVR of Autonomous Power Generating System"*, IAENG International Journal of Computer Science, 37:1, February, 2010.

[5] C. Thanga, S.P. Snvastava, and P. Agarwal, *"Particle Swarm and Fuzzy Logic Based Optimal Energy Control of Induction Motor for a Mine Hoist Load Diagram"*, IAENG International Journal of Computer Science, 36:1, 2009.

[6] Fadhil A. Hassan, Lina J. Rashad, *"Artificial Neural Control of 3-Phase Induction Motor Slip Regulation Using SPWM Voltage Source Inverter"*, Engineering and Technology Journal, Vol. 28, No. 12, 2010.

[7] Wang Dazhi, Wang Zhenlei, Gu Shusheng, "*Identification and Control of Induction Motor Using Artificial Neural Network*", IEEE, Page: 751-754, 2005.

[8] B. Allaoua, B. Gasbaoui, B. Mebarki, *"Setting Up PID DC Motor Speed Control Alteration Parameters Using Particle Swarm Optimization Strategy"*, Leonardo Electronic Journal of Practices and Technologies, Issue 14, P 19-32, 2009.

[9] B. Nagaraj, N. Murugananth, *"Soft Computing-Based Optimum Design of PID Controller for a Position Control of DC Motor"*, Mediamira Science Publisher, Vol. 51, No. 1, 2010.

[10] M. Nasri, H. Nezamabadi, M. Maghfoori, *"A PSO-Based Optimum Design of PID Controller for a Linear Brushless DC Motor", World Academy of Science, Engineering and Technology 26, 2007.*

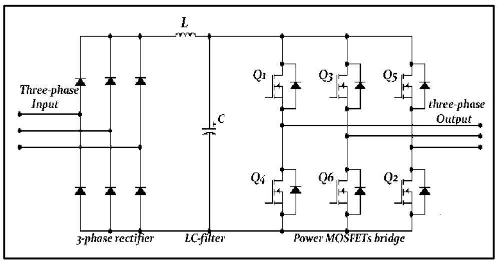

Figure (1) ***3-phase VS Inverter.***

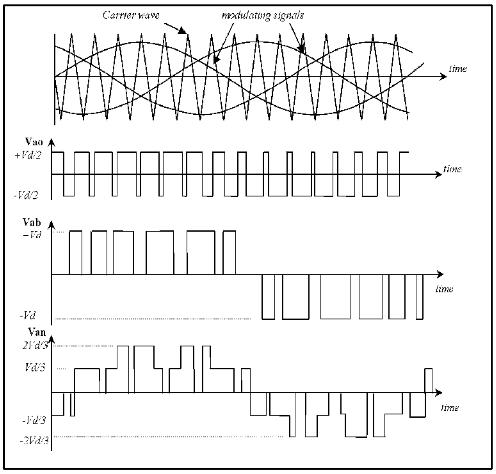

Figure (2) ***SPWM Principles.***

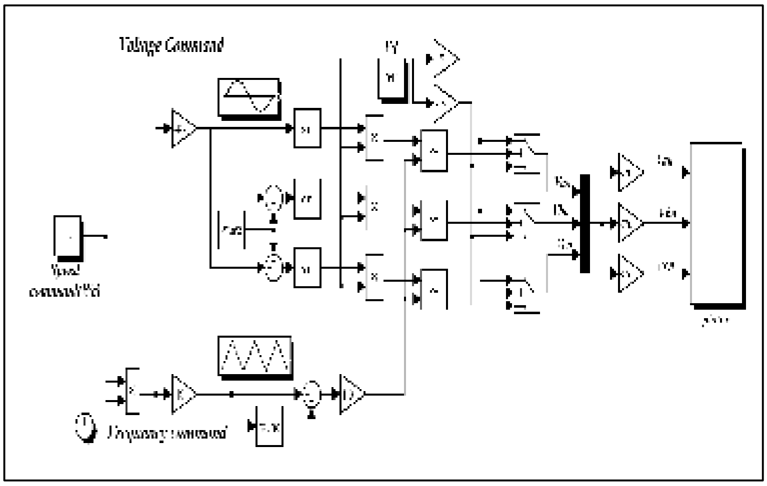

Figure (3) *SPWM Simulation*

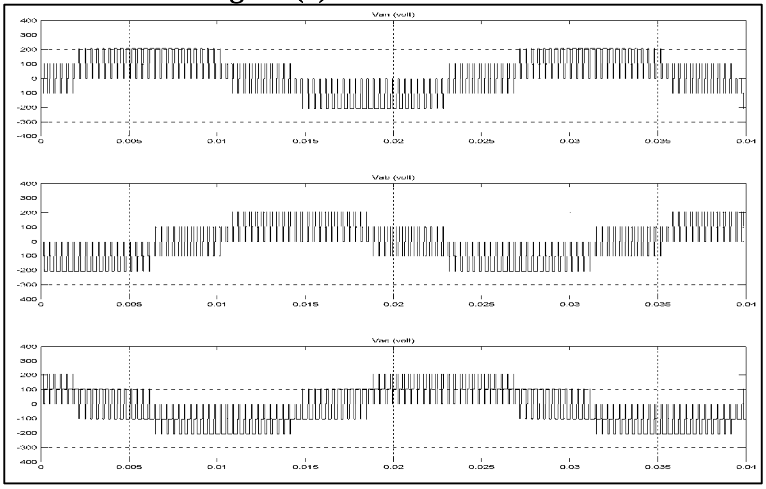

Figure (4) Linear Region Output Phase Voltages.

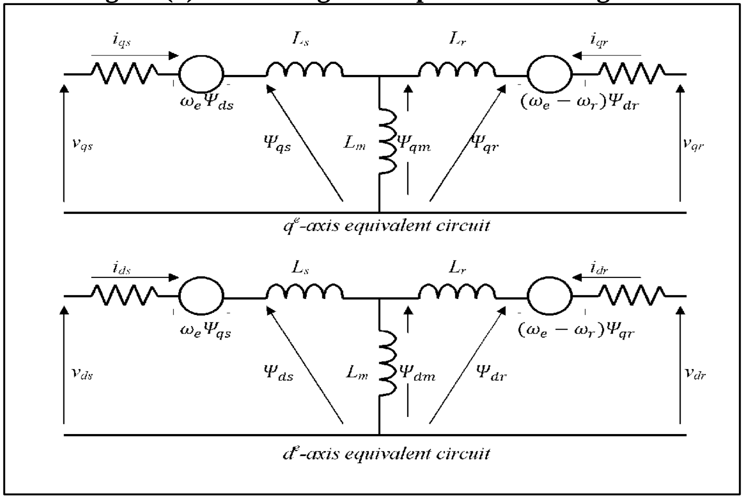

Figure (5) q^e-d^e *I.M Equivalent Circuit.*

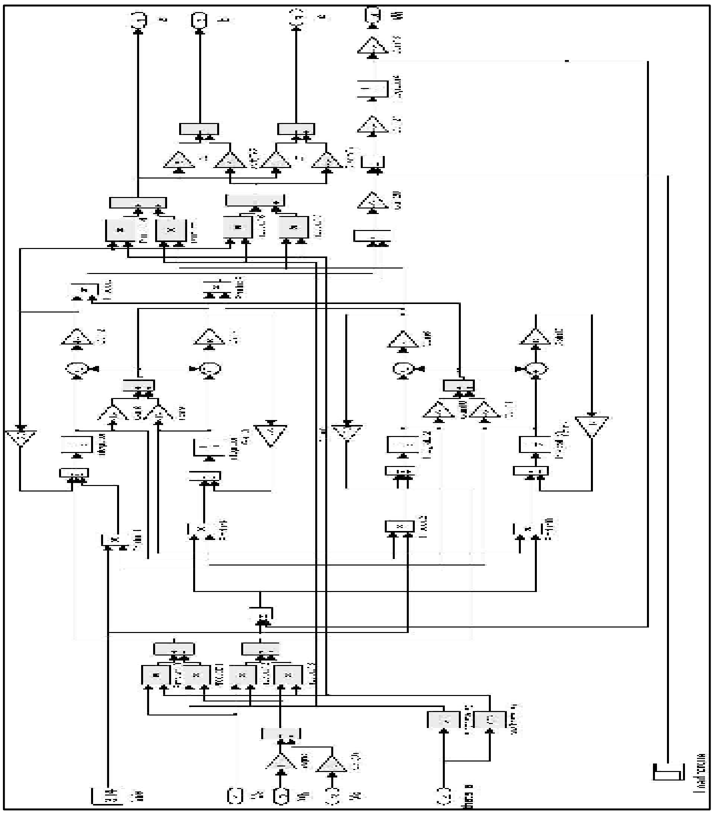

Figure (6) *I.M Simulation.*

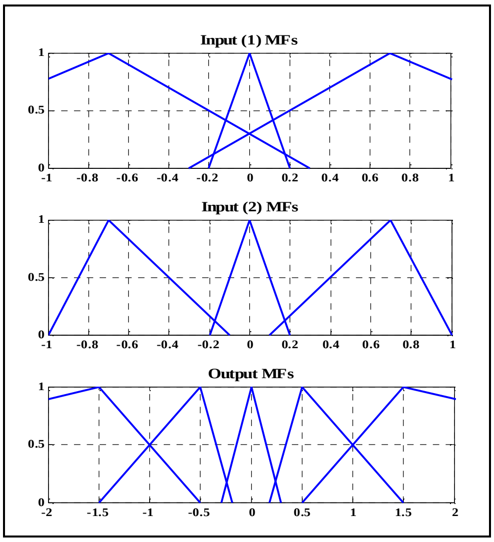

Figure (7) Ordinary Adapting of Input/Output MFs

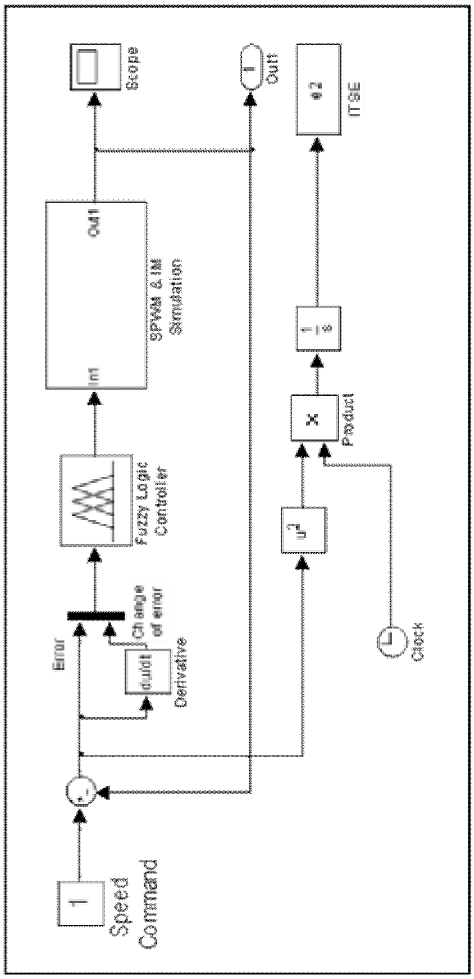

Figure (8) *Overall System Simulation*

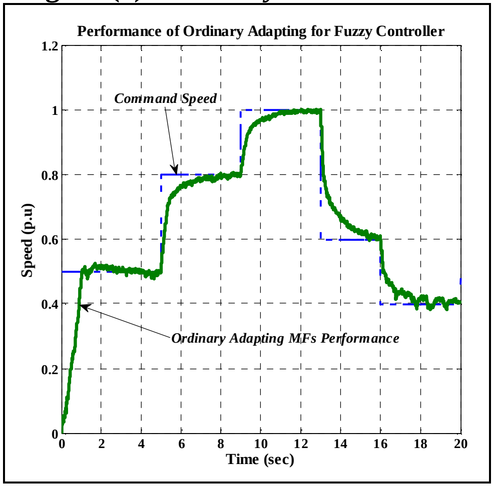

Figure (9) *System Performance of Ordinary Adapting FLC*

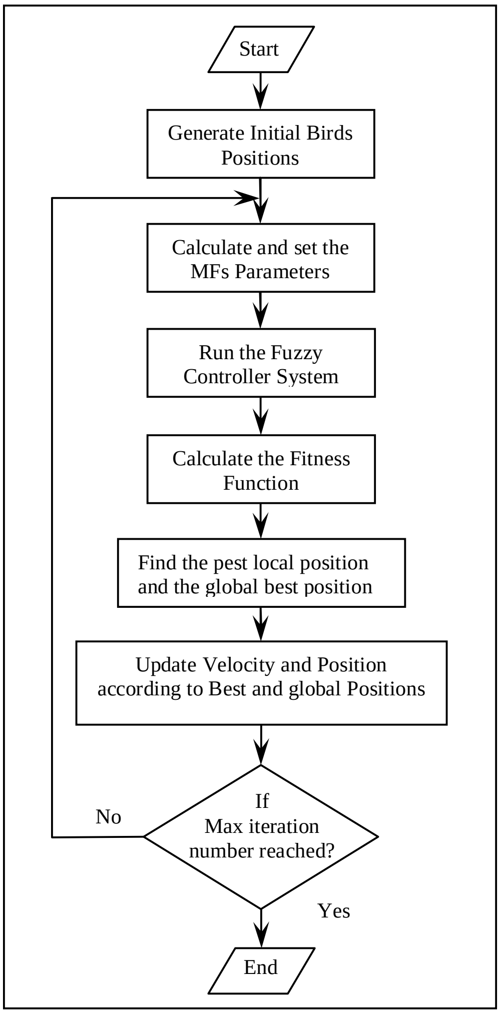

Figure (10) Flowchart of PSO Algorithm

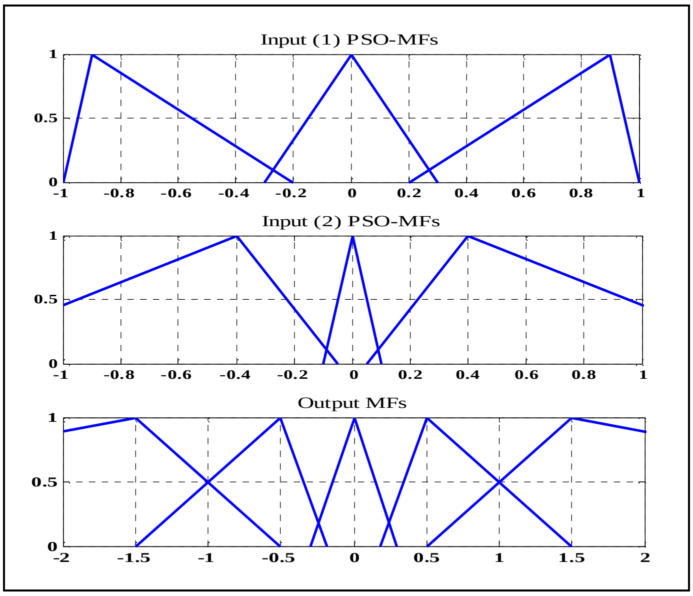

Figure (11) PSO Adapting of Input MFs

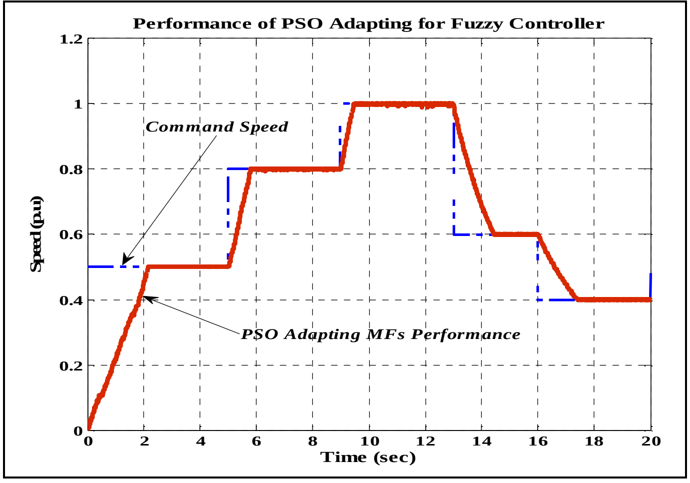

Figure (12) System Performance of PSO Adapting FLC

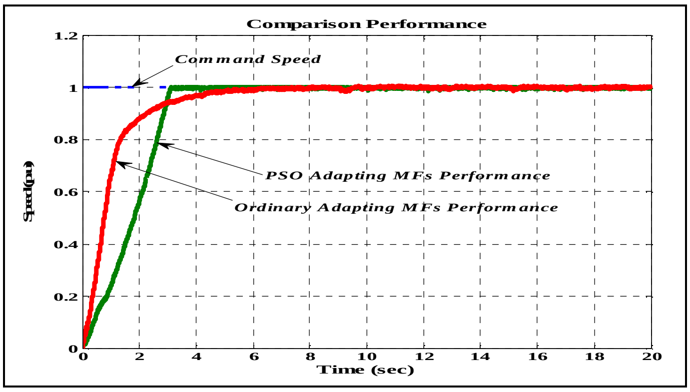

Figure (13) Comparison Performance