Artificial intelligence based speed control of DTC induction motor drives - a comparative study

S.M. Gadoue, D. Giaouris, J.W. Finch *

School of Electrical, Electronic & Computer Engineering, Newcastle University, Newcastle upon Tyne, NE1 7RU, UK

Abstract

The design of the speed controller greatly affects the performance of an electric drive. A common strategy to control an induction machine is to use direct torque control combined with a PI speed controller. These schemes require proper and continuous tuning and therefore adaptive controllers are proposed to replace conventional PI controllers to improve the drive's performance. This paper presents a comparison between four different speed controller design strategies based on artificial intelligence techniques; two are based on tuning of conventional PI controllers, the third makes use of a fuzzy logic controller and the last is based on hybrid fuzzy sliding mode control theory. To provide a numerical comparison between different controllers, a performance index based on speed error is assigned. All methods are applied to the direct torque control scheme and each control strategy has been tested for its robustness and disturbance rejection ability.

Key words:

Direct torque control, PI control, Artificial intelligence, Sliding mode control, Induction motor drives

---

* Corresponding Author. Tel: +441912227326; fax:+441912228180

Email address: J.W.Finch@ncl.ac.uk (J.W. Finch).

1 Introduction

PI controllers are widely used in industrial control systems applications. They have a simple structure and can offer a satisfactory performance over a wide range of operation. Due to the continuous variation in the plant parameters and the nonlinear operating conditions, fixed gain PI controllers may become unable to provide the required control performance [1-3]. A typical example where nonlinearities or changing parameters occurs is that of a modern brushless electrical drive involving an induction machine being fed by an inverter. It is very difficult for of-line tuning algorithms to cope with the continuous variations in the induction motor parameters as well as the nonlinearities present in inverter, motor and controller [1, 4-6]. Therefore on-line controller tuning becomes desirable when high performance is required from the drive scheme. A lot of strategies have been proposed to tune the PI controller parameters. The most famous method, which is frequently used in industrial applications, is the Ziegler-Nichols method. Frequency response methods based on specified phase and gain margins as well as crossover frequency have also been used to improve the behaviour of the system. Root locus and pole assignment design techniques are also used, in addition to transient response specifications [6, 7]. These inherent disadvantages of the PI controller have encouraged the replacement of the conventional PI controller with adaptive control techniques, such as Sliding Mode Control (SMC), Model Reference Adaptive Control (MRAC), self tuning PI controllers and other Artificial Intelligence (AI) based controllers such as the Fuzzy Logic Controller (FLC), neural network, fuzzy neural network and Genetic Algorithms (GA) [1, 3-5, 8-12].

Fuzzy logic strategy has been proposed for speed control in vector control induction motor drives [1, 3, 5, 12]. Combined with neural networks, a hybrid adaptive neuro-fuzzy controller has also been presented for speed control [13], with an on-line and off-line memetic control design being applied to permanent magnet drives [14]. Fuzzy logic strategy can cope with parameter uncertainties, imprecision and does not rely on any mathematical models based on human knowledge. Difficult tuning of fuzzy logic parameters and stability are its main problems [2, 15]. Genetic Algorithms (GA) are adaptive search techniques based on the "survival of the fittest" biological aspect. They have shown an efficient and effective way for optimization applications by searching global minimal without needing the derivative of the cost function. [11]. However most GA based strategies are not real time [6, 9]. GA strategy is proposed to optimize the performance of the Fuzzy and adaptive sliding mode controllers [3, 9]. Fuzzy logic and GAs are also proposed for tuning the conventional speed controller for a vector control induction motor drive [4, 6]. Conventional and Fuzzy sliding mode strategies have been also presented as controllers for induction motor drives [2, 9, 10, 16]. Such strategies show robustness against motor parameter variation, better external disturbance rejection and stability and fast dynamic response [2, 9, 10]. Chattering in the steady state is the main drawback of the conventional strategy which may be solved by using the fuzzy

sliding mode technique [2, 15, 18].

Despite much research on the design of speed controllers based on either pure or hybrid Artificial Intelligence (AI), these techniques have been separately studied and some papers in the literature provide a comparative study of these different controllers [2, 12, 17]. Usually each new controller design is just compared with the conventional PI controller and not with other new designs [2, 4, 12]. Moreover, the majority of these designs are applied to vector control drives. One of the main suggestions from a recently published survey paper [19], by two of the present authors, is that progress in the electrical drives control area is hampered by lack of agreed standard tests and infrequent comparisons of revised algorithms with previous standards.

This paper provides a comprehensive analysis and comparison between three of these different controller designs with a GA optimized PI controller. Transient response, robustness, and disturbance rejection capability based on assigning a performance index in terms of speed error provide a numerical comparison of performances when applied to a DTC induction motor drive.

2 Machine model and DTC strategy

The induction motor state space model with stator and rotor currents as state variables can be written in d-q coordinates fixed to the stator as:

$$\begin{bmatrix} pi_{sd} \\ pi_{sq} \\ pi_{rd} \\ pi_{rq} \end{bmatrix} = \frac{1}{\sigma L_s L_r} \begin{bmatrix} -R_s L_r & L_m^2 \omega_r & L_m R_r & L_m L_r \omega_r \\ -L_m^2 \omega_r & -R_s L_r & -L_m L_r \omega_r & L_m R_r \\ L_m R_s & -L_m L_s \omega_r & -L_s R_r & -L_s L_r \omega_r \\ L_m L_s \omega_r & L_m R_s & L_s L_r \omega_r & -L_s R_r \end{bmatrix} \begin{bmatrix} i_{sd} \\ i_{sq} \\ i_{rd} \\ i_{rq} \end{bmatrix} + \frac{1}{\sigma L_s L_r} \begin{bmatrix} L_r & 0 \\ 0 & L_r \\ -L_m & 0 \\ 0 & -L_m \end{bmatrix} \begin{bmatrix} v_{sd} \\ v_{sq} \end{bmatrix} \tag{1}$$

The mechanical equation can be written as:

$$T_{em} - T_l = \frac{J}{P} \frac{d\omega_r}{dt} + \frac{B}{P} \omega_r \tag{2}$$

and the electromagnetic torque:

$$T_{em} = \frac{3}{2} P L_m \left(i_{rd} i_{sq} - i_{sd} i_{rq} \right) \tag{3}$$

where T_L is the load torque, J is the combined motor and load inertia, B is the friction coefficient, P is the number of motor pole pairs, ω_r is the rotor speed in electric rad/s, L_s, L_r and L_m are the stator, rotor and mutual inductances respectively, i_{sd}, i_{sq}, i_{rd}, i_{rq} are the

stator and rotor d-axis and q-axis current components and σ is the leakage coefficient given by:

$$\sigma = 1 - \frac{L_m^2}{L_s L_r} \tag{4}$$

In principle DTC is a direct hysteresis stator flux and electromagnetic torque control which triggers one of the eight available discrete space voltage vectors generated by a Voltage Source Inverter (VSI), to keep stator flux and motor torque within the limits of two hysteresis bands. Correct application of this principle allows a decoupled control of flux and torque. The DTC block diagram is shown in Fig. 1.

Figure 1.

3 PI controller tuning using GA

GA is a stochastic global search optimization technique based on the mechanisms of natural selection. GA has been recognized as an effective technique to solve optimization problems [8, 9]. Compared with other optimization techniques GA is superior in avoiding local minima which is a common aspect of nonlinear systems. Furthermore GA is a derivative-free optimization technique which makes it more attractive for applications that involve nonsmooth or noisy signals. Generally GA consists of three main stages; selection, crossover and mutation [3, 6, 8]:

• Selection stage

In this stage individuals of the initial population are selected for reproduction with probability proportional to their fitness value. The purpose of this operation is to obtain a mating pool with the fittest individuals selected according to a probabilistic rule that allows these individuals to be mated into the new population.

• Crossover stage

After the selection stage the genetic crossover operation is then applied between parent pairs from the mating pool to generate new individuals or offsprings which acquire good features from their parents. This crossover operation is performed with a crossover probability (P_c). The crossover operation can be a one-point or a double-point operation.

• Mutation stage

The last operation is genetic mutation which takes place after the crossover operation. The application of genetic mutation introduces a change in the offspring bit string to produce new chromosomes which may represent a solution of the problem and at the same time avoid the population falling into a local optimal point. The mutation operation is performed with a

mutation probability (P_m) which is usually low to preserve good chromosomes and to mimic real life where mutation rarely happens. The application of these three basic operations allows the creation of new individuals which may be better than their parents. This algorithm is repeated for many generations and finally stops when reaching individuals which provide an optimum solution to the problem [3, 7-9]. The GA architecture is shown in Fig. 2 [3, 7].

Figure 2.

Due to its effectiveness in searching nonlinear, multi-dimensional search spaces, GA can be applied to the tuning of PI speed controller gains to cope with the nonlinearities existing in the inverter and the machine. In this case the fitness function used to evaluate the individuals of each generation can be chosen to be the reciprocal of Integral with Time of Absolute Error (ITAE). The mathematical expression of this cost function, which is the function minimized by the GA, can be written as:

$$ITAE = \int_0^t t\,|e(t)|\,dt \tag{5}$$

During the search process the GA looks for the optimal setting of the PI speed controller gains which minimize the cost function (ITAE). Individuals with low ITAE are considered as the fittest. This function is used as the GA evolution criteria and has the advantage of avoiding cancellation of positive and negative errors. Each chromosome represents a solution of the problem and hence it consists of two genes: the first one is the K_p value and the second is the K_i value: So the chromosome vector is [K_p K_i]. The range of each gain must be specified. The genetic algorithm parameters chosen for the tuning purpose are shown in Table 1, [7].

Table 1.

4 PI tuned by fuzzy logic

The control strategy presented in section 3 suffers from an inability to cope with online changes of the system's parameters as well as disturbance rejection, even though it provides optimum gains for a specific operating condition. On-line GA strategies have been proposed but they require significant processing power and hence may be unattractive for real drives applications [6]. A solution to this problem is to change on-line the gains of the PI compensator by using a Fuzzy Logic Controller (FLC) [4]. Tuning methods based on fuzzy

logic have been found to offer advantage in dealing with systems that are imprecise, nonlinear, or time varying with uncertain or unknown parameter and structure variation. This makes the application of the FLC ideal for the tuning of a speed controller in a DTC scheme.

Furthermore a FLC is relatively easy to implement and does not need a mathematical model of the controlled system [1, 3]. FLC has become popular in the field of industrial control applications. When fuzzy logic is used for the on-line tuning of the PI speed controller it receives the scaled values of the speed error and the change in speed error. Its output is the updating of the PI controller gains (ΔK_p and ΔK_i) based on a set of rules to maintain excellent control performance even in the presence of parameter variation and drive nonlinearity [4, 7]. The block diagram of the control system is shown in Fig. 3.

Figure 3.

Each input of the FLC has 5 triangular membership functions with equal width and overlap. The first output (ΔK_p) has 3 triangular membership functions; whereas the second output (ΔK_i) has 5 membership functions. The inference rules base has 25 rules [4, 7]. The parameters of the FLC are obtained by trial error to ensure optimal performance. The fuzzy inference rules used for the on-line tuning of the PI controller gains are given in Tables 2 and 3 [4, 7]. The flow chart of this self tuning (ST) controller is given in Fig. 4 [7].

Tables 2 - 3

Figure 4.

The parameters of the FL ST algorithm are listed in Table 4.

Table 4.

5 Fuzzy logic speed controller

Since FLC can cope with the nonlinearities, load disturbances and the uncertainties of the DTC it has also been used to entirely replace the traditional PI controller [1, 3, 12]. For the

proposed FL speed controller, the inputs are the normalized values of the speed error and the rate of change to remain between ±1 limits of speed error [1]. Two scaling factors (K_e and K_d) are used to normalize the actual speed error and its rate of change. The output of the controller is the normalized change of the motor torque command which when multiplied by a third scaling factor (K_u) generates the actual value of the rate of change of the motor torque demand.

Finally, a discrete integration is performed to get the value of the electromagnetic torque command. Hence a PI-Type FLC is created [5, 17]. The FLC structure is shown in Fig. 5 [1, 15], [5]. Table 5 shows the fuzzy rule base with 49 rules which can be obtained from observation of the drive performance at different operating points, [5, 17].

The following fuzzy sets are used: NB= NEGATIVE BIG, NM= NEGATIVE MEDUIM, NS= NEGATIVE SMALL, EZ= ZERO, PS= POSITIVE SMALL, PM= POSITIVE MEDUIM, PB= POSITIVE BIG, [5, 17]. The membership functions of the FLC shown in Fig. 6 are obtained by a trial and error technique where the EZ fuzzy set has a narrow shape different from other fuzzy sets to improve the controller steady state performance.

Figure 5.

Table 5.

Figure 6.

6 Fuzzy sliding mode controller

Another method of replacing the PI controller is the use of a Sliding Mode Control (SMC) [2, 10]. This is a Variable Structure Control (VSC) strategy with high frequency switched feedback control which forces the states of the system to slide on a predefined hypersurface. The plant states are mapped into a control surface using different continuous functions [18]. The discontinuous control action switches between these several functions according to plant state value at each instant to achieve the desired trajectory. SMC is known for its capability to cope with bounded disturbances as well as model imprecision which makes it ideal for the robust nonlinear control of induction motor drives [2, 6, 10]. Designing a sliding mode speed controller for the induction motor DTC drive starts by defining the speed error as [10]:

$$e(t) = \omega_r - \omega_r^* \tag{6}$$

Defining the attractive switching surface as:

$$s(t) = e(t) - \int_0^t k\, e(\tau) d\tau \quad (6) \qquad (7)$$

such that the error behaviour at the sliding surface at $s = 0$ will be represented by a homogeneous differential equation and hence it will be forced to exponentially decay to zero. The error dynamics equation can be written as:

$$\dot{e}(t) = ke(t); k < 0 \qquad (8)$$

The structure of the sliding mode controller can be written as [15]:

$$u^* = u_{eq} + u_s \qquad (9)$$

$$u_s = -k_1 sign(s) \qquad (10)$$

The sign function is given by:

$$sign(s) = \begin{cases} -1 & for\ s < 0 \\ +1 & for\ s > 0 \end{cases} \qquad (11)$$

where u_{eq} is the equivalent control which defines the control action when the system is on the sliding mode [16], and k_1 is a constant which represents the maximum value of the controller output. This constant is selected to be large enough to reduce the effect of any external disturbances [15]. $s(t)$ is the switching function because the control action switches its sign according to its value. The switching control action is shown in Fig. 7(a). Unfortunately the use of the sign function causes high frequency chattering due to the discontinuous control action, which represents a severe problem when the system state is close to the sliding surface [15]. This problem is more severe when a SMC is used in a DTC scheme which already includes many switching operations to achieve the desired values of the electromagnetic torque and the stator flux. To overcome this problem a boundary layer is introduced around the switching surface as shown in Fig. 7(b), [15, 16, 18]. The switching part of the control law is now written as:

$$u_s = -k_1 sat\left(\frac{s}{\phi}\right) \qquad (12)$$

where Φ represents the thickness of the boundary layer. The saturation function is defined as [13]:

$$sat\left(\frac{s}{\phi}\right) = \begin{cases} \left(\frac{s}{\phi}\right) & for\ \left|\frac{s}{\phi}\right| \le 1 \\ sign\left(\frac{s}{\phi}\right) & for\ \left|\frac{s}{\phi}\right| > 1 \end{cases} \qquad (13)$$

The introduction of the saturation function represents the continuous approximation of the discrete relay action by the sign function. The system robustness becomes highly dependent on the boundary layer thickness [16].

Another approach to reduce the chattering phenomenon is to combine a FLC with a SMC [2,

15]. Hence a new Fuzzy Sliding Mode Controller (FSMC) is formed with the robustness of the SMC and the smoothness of a FLC. In this technique the term $-k_1 sat(s/\Phi)$ is replaced by a fuzzy inference system as shown in Fig. 7(c) in order to smooth the control action [15]. The choice of Φ is crucial; small values of Φ may not solve the chattering problem and large values may increase the steady state error [15], requiring a compromise choice when selecting the boundary layer thickness The block diagram of the control system and the input-output membership functions of the fuzzy logic controller are shown in Figs. 8-9 [15, 16]. The If-Then rules of the fuzzy logic controller can be written as [15, 16]:

If s is BN then u_S is BIGGER

If s is MN then u_S is BIG

If s is JZ then u_S is MEDIUM

If s is MP then u_S is SMALL

If s is BP then u_S is SMALLER

Figures 7-9

7 Simulation results

To compare the different speed controller design strategies a DTC of a 7.5 kW squirrel cage induction motor shown in Fig. 1 is simulated using Matlab-Simulink software using the well established two-axis machine model, which includes the main speed dependant terms. Very low speed behaviour will be affected by power electronic nonlinearities such as device voltage drop, although attention has been given to reducing these effects on behaviour [19]. Experimental investigation of the most promising schemes is being sort, but the comparison based on these simulations is invaluable in establishing priorities. The induction motor parameters are given in Table 6. The motor is started under 25% rated load with a speed command of 50 electrical rad/s and is running under normal operating conditions from t=0 to t=0.5s. To study the effect of parameter variation on the performance of the different controllers, a 20% step increase in the motor stator resistance is applied at t=0.5s. Stator resistance is chosen because the performance of DTC drive is greatly affected by the variation of this parameter especially at low speed. At t=1s, a 100% sudden load increase is applied to the motor.

The simulation is performed for the four different speed controller strategies:

PI-GA: Using the GA parameters given in Table 1, the optimal PI controller gains with 25% rated torque applied to the motor during the tuning process are found to be $K_p = 127$, $K_i = 4$.

PI-FL: The PI speed controller is tuned online using fuzzy logic as shown in Fig. 3 with the parameters given in Table 4. These parameters are obtained by trial error to ensure optimal performance.

FLC: The PI speed controller is entirely replaced by a FLC as shown in Fig. 5. The controller parameters are chosen based on the guidelines reported in [1] as: K_e =0.007, K_d=0.5, K_u=17.3.

FSM: The PI controller is replaced by a fuzzy SMC as shown in Fig. 8. The controller coefficients used in the simulation are: $k = -10^{-5}$, $k_1 = 300$ and $\Phi = 1$. The value of K is obtained based on required error dynamic performance. No design criterion is assigned to design the value of K1; however, its value should be selected high enough to make the manifold $s = 0$ in (7) attractive [15]. The value of Φ is obtained as a compromise between chattering reduction and steady state error requirements.

7.1 Speed response

The starting transient performance of the induction motor under the different control strategies is shown in Fig.10. The FLC has the best transient response where the motor speed is approximately built up in less than 0.1s without overshoot. PI-FL has an over-damped response where the motor speed builds in 0.115s without overshoot. PI-GA and FSM have a speed overshoot of 1% and 1.4% respectively which are still very small values.

Fig.11 shows the speed response of the different techniques when the stator resistance changes abruptly. Both PI-GA and FSM show more robustness against stator resistance variation compared to FLC and PI-FL. When the 100% load change is applied to the motor, the rotor speed with the PI-GA strategy drops to 49.92 rad/s with a steady state error of 0.16% as shown in Fig.12 (a). This is due to the variation of the operating conditions from those used during the of-line tuning process. Due to their adaptive features, the three other control strategies show fast disturbance rejection. FSM is the most robust controller where the speed drops initially to 49.98 rad/s and then is adjusted back to its demanded value in 1ms as shown in Fig.12 (b). The FLC and PI-FL controllers show speed drops to 49.8 rad/s and 49.6 rad/s but are corrected back after 0.1 and 0.2 s respectively, as shown in Figs.12(c)-(d). The load torque disturbance rejection property of the different controllers is shown together in Fig. 13.

Figure 10-13

Table 6.

7.2 ITAE

To give a clear idea of the performance of the different controllers, the ITAE using each technique is calculated during these three stages: normal operating conditions, stator resistance variation and load torque change, as shown in Figs.14-17. During normal operating conditions, PI-GA shows the lowest ITAE since it uses the optimal PI controller gains for normal operating conditions. FSM has an ITAE near to that of PI-GA. Compared to FLC, PI-FL shows a lower ITAE when the drive is working under normal operating conditions. During stator resistance variations, FSM has the lowest ITAE. PI-GA has the lowest ITAE after the FSM technique. PI-FL performance is still better compared to FLC for motor parameter variations. When the sudden load change takes place at t=1s, PI-GA gives the highest ITAE whereas FSM shows excellent robustness with the lowest ITAE. For this load variation, FLC shows better disturbance rejection capability compared to PI-FL. The simulation results are summarized in Table 7.

Figures 14-17

Fuzzy logic has now been combined with a conventional sliding mode controller with switching function based on (10), simulated under the same conditions. The total ITAE obtained is 0.293, which is very large compared to the value of 0.083 obtained from FSM. The considerable chattering in the speed response obtained from the conventional technique is shown in Fig. 18. This chattering is reduced dramatically when fuzzy logic is combined with sliding mode as shown in Fig. 19.

Figures 18-19

Table 7.

8 Discussion

PI-GA works well under normal operating conditions, giving small drift but has a low torque disturbance rejection capability due to the fixed gain controller. Generally the GA of-line tuning process is simple but may need a lot of time to converge to the optimal solution, depending on the complexity of the drive system and as the choice of the GA parameters. To decrease the convergence time, GA parameters such as crossover and mutation rate can be varied based on statistics of the population at each generation to form an adaptive genetic algorithm. Furthermore, GA can be implemented to tune the PI controller gains on-line,

however the updating time will be highly dependent on the convergence speed of the algorithm.

Due to its variable gains, PI-FL performs better than fixed gain PI-GA during a load torque disturbance. Compared to FLC, PI-FL has better robustness against motor parameter variation as well as better steady state performance since the gain updating stops after a given limit of speed accuracy. PI-FL also has better steady state performance compared to FSM which is affected by the chattering in the steady state. FLC has a better disturbance rejection capability compared to PI-FL and a better transient response during starting. It does require on-line tuning of its parameters: scaling factors, membership functions and rules during drive operation to form an adaptive fuzzy logic controller to improve its steady state performance. This will increase the scheme complexity and computational effort. Results obtained from FSM look promising: during normal operating conditions its performance is very close to PI-GA. Furthermore it shows good robustness against motor parameter variation with good, fast load disturbance rejection capability related to proper selection of the attractive switching surface with minimum hitting time. However, FSM still needs some improvement to reduce the chattering phenomenon which directly affects the steady state performance of the controller. A comparison between the four controllers is given in Table 8.

Table 8.

9 Conclusion

In this paper four design strategies for the speed controller in DTC of induction motor are presented: PI controller tuned by a genetic algorithm and fuzzy logic, fuzzy sliding mode and fuzzy logic controllers. These design techniques are based on artificial intelligence techniques which do not require any mathematical modelling. All these techniques work well under normal operating conditions. Adaptive structure controllers show more robustness against motor parameter variations as well as high disturbance rejection capability compared to fixed structure techniques. The fuzzy logic speed controller needs some modifications to improve its steady state performance. The fuzzy sliding mode controller seems the best choice for the controller design in terms of robustness and disturbance rejection capability, but still needs modifications to reduce the chattering phenomenon in the steady state.

Acknowledgements

The authors would like to gratefully acknowledge The Ministry of Higher Education, Arab Republic of Egypt for the financial support of this research project.

References

[1] M. N. Uddin, T. S. Radwan, M. Rahman, Performance of fuzzy-logic-based indirect vector control for induction motor drive, IEEE Trans. on Ind. Appl. 38 (5) (2002) 1219-1225.

[2] F. Barrero, A. Gonzalez, A. Torralba, E. Galvan, L. G. Franquelo, Speed control of induction motors using a novel fuzzy sliding mode structure, IEEE Trans. Fuzzy Systems, 10 (2002) 375-383.

[3] W. Oh, Y. Kim, C. Kim, T. Kwon, H. Kim, Speed control of induction motor using genetic algorithm based fuzzy controller, in: Proc. IECON'99, Vol. 2, 1999, pp. 625-629.

[4] L. Mokrani, R. Abdessemed, A fuzzy self-tuning PI controller for speed control of induction motor drive, in: IEEE Conference on Control Applications 2003, Piscataway, NJ, USA, 2003, pp. 785-790.

[5] Y. Lai, J. Lin, New hybrid fuzzy controller for direct torque control induction motor drives, IEEE Trans. Power Electronics 18 (2003) 1211-1219.

[6] F. Lin, H. Shieh, K. Shyu, P. Huang, Online gain tuning IP controller using real-coded genetic algorithm, Journal of Electric Power Systems Research 72 (2004) 157-169.

[7] S. M. Gadoue, D. Giaouris, J. W. Finch, Tuning of PI speed controller in DTC of induction motor based on genetic algorithms and fuzzy logic schemes, in: 5th International Conference on Technology and Automation 2005, Thessaloniki - Greece, 2005, pp. 85-90.

[8] P. Vas, Artificial-Intelligence-Based Electrical Machines and Drives, Oxford University Press, New York, USA, 1999.

[9] F. Lin, W. Chou, P. Huang, Adaptive sliding mode controller based on real time genetic algorithm for induction motor servo drive, IEE Proc. Electr. Power Appl. 150 (2003) 1-13.

[10] O. Barambones, A. Garrido, F. Maseda, P. Alkorta, An adaptive sliding mode control law for induction motor using field oriented control theory, in: Proc. IEEE Int. Conf. Control Appls., year =2006, pages=1008-1013.

[11] Fleming P.J.; Purshouse R.C., "Evolutionary algorithms in control systems engineering: a survey", Control Engineering Practice, vol. 10, no. 11, Nov. 2002, pp. 1223-1241

[12] Z. Ibrahim, E. Levi, A comparative analysis of fuzzy logic and PI speed control in high performance ac drives using experimental approach, IEEE Trans. Industry Appl. 38 (2002) 1210-1218.

[13] M. N. Uddin, H. Wen, Development of a self tuned neuro-fuzzy controller for induction motor drives, in: Proc. Industry Applications Conf., 39th IAS annual meeting, 2004, pp.

2630-2636.

[14] A. Caponio, G. L. Cascella, F. Neri, N. Salvatore, M. Sumner, "A Fast Adaptive Memetic Algorithm for On-line and Off-line Control Design of PMSM Drives," IEEE Trans. Systems, Man and Cybernetics - Part B: vol. 37, no. 1, , Feb. 2007, pp. 28 - 41

[15] J. Lo, Y. Kuo, Decoupled fuzzy sliding mode control, IEEE Trans. Fuzzy Systems, 6 (1998) 426-435.

[16] A. Hazzab, I. K. Bousserhane, M. Kamli, Design of fuzzy sliding mode controller by genetic algorithms for induction machine speed control, International Journal of Emerging Electric Power Systems 1 (2004) 1016-1027.

[17] M. A. Denai, S. A. Attia, Intelligent control of an induction motor, Electric Power Components and Systems 30 (2002) 409-427.

[18] W. S. Levine (Ed.), The control handbook, CRC Press, 1996.

[19] J. W. Finch and D. Giaouris, "Controlled AC Electrical Drives," IEEE Transactions on Industrial Electronics, vol. 55, no. 1, Feb. 2008, pp. 481-491.

List of Figs.

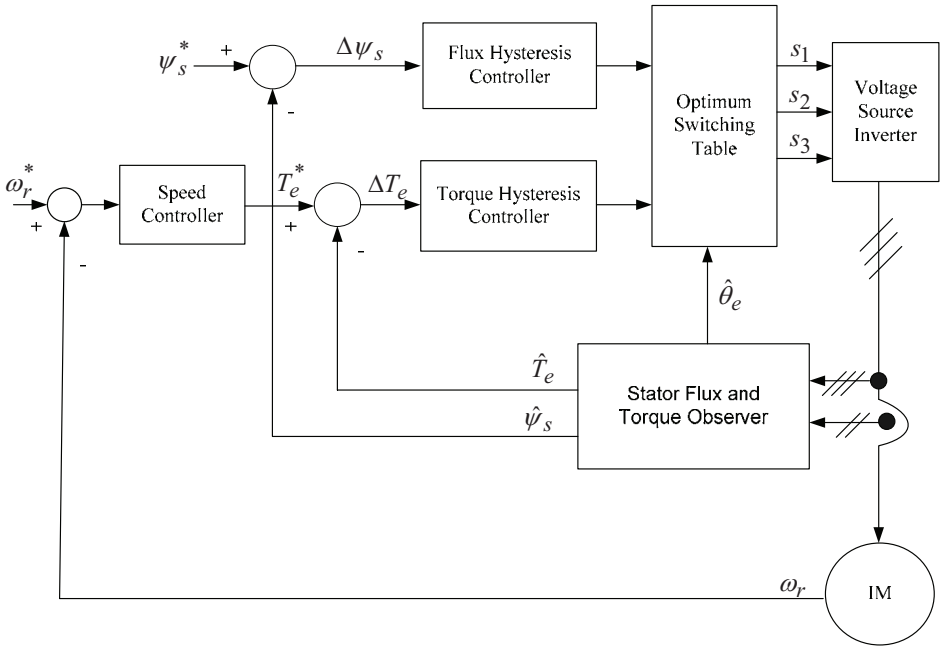


Fig. 1 Block diagram of DTC with speed control loop

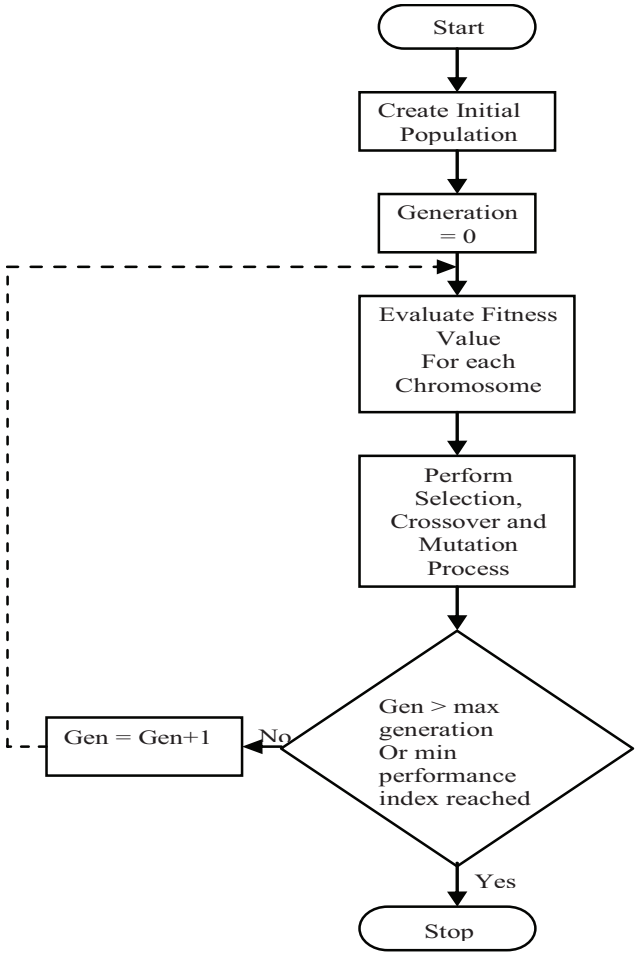


Fig. 2 Genetic Algorithm Architecture

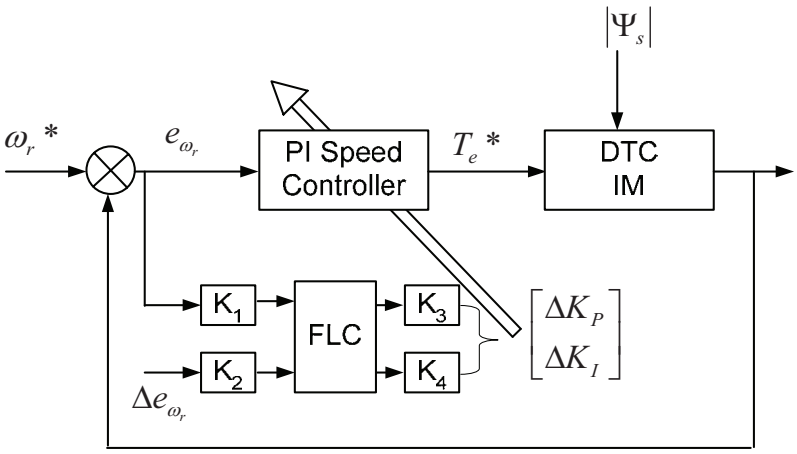


Fig. 3 Fuzzy self tuning PI speed controller

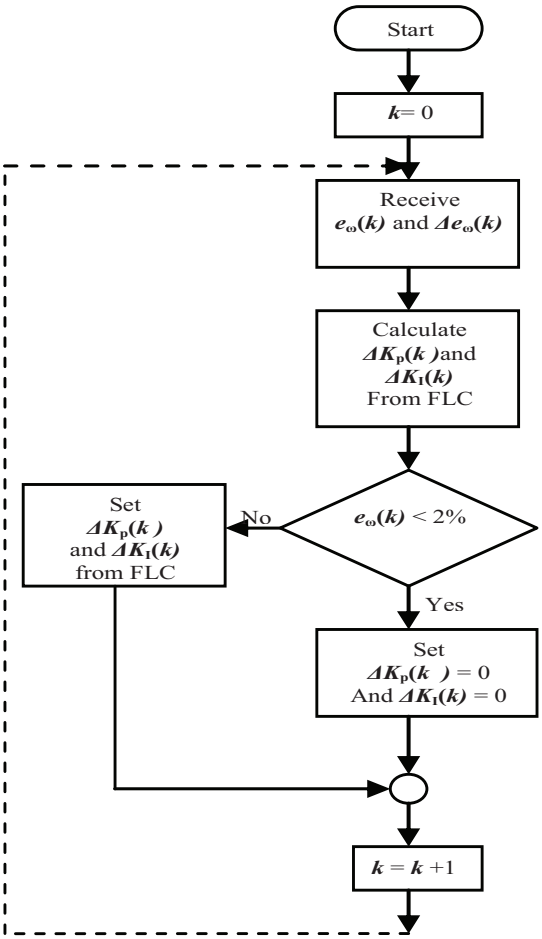


Fig. 4 Flow chart of Fuzzy self tuning PI speed controller

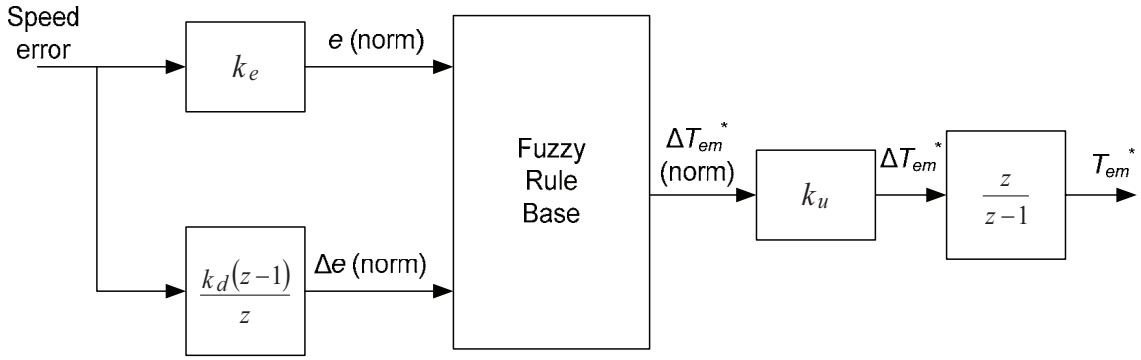
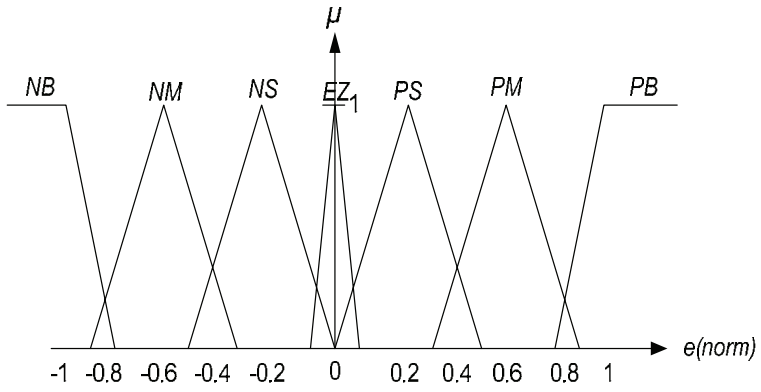
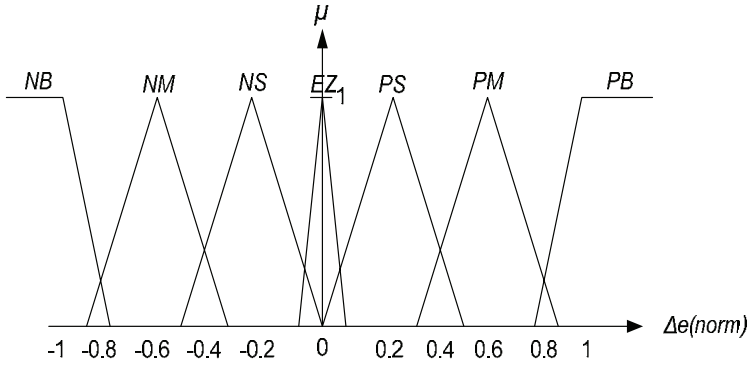


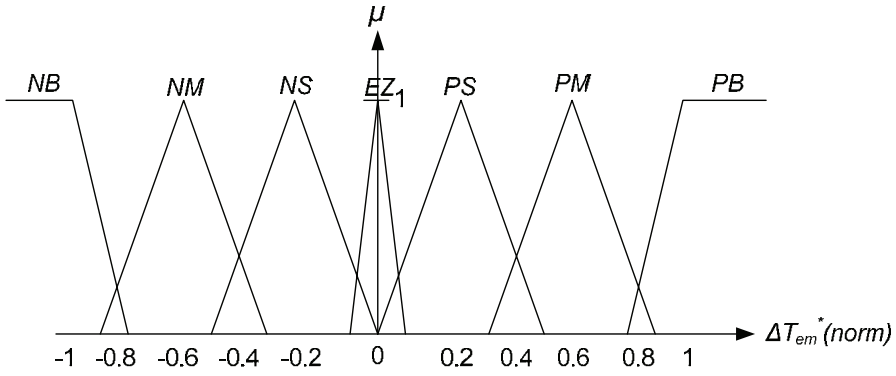
Fig. 5 Block diagram of PI-Type Fuzzy logic controller



(a)

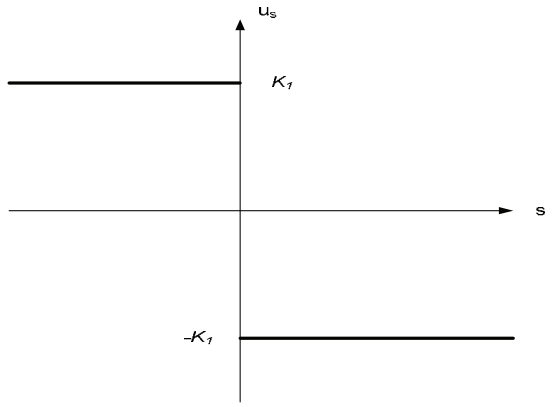


(b)

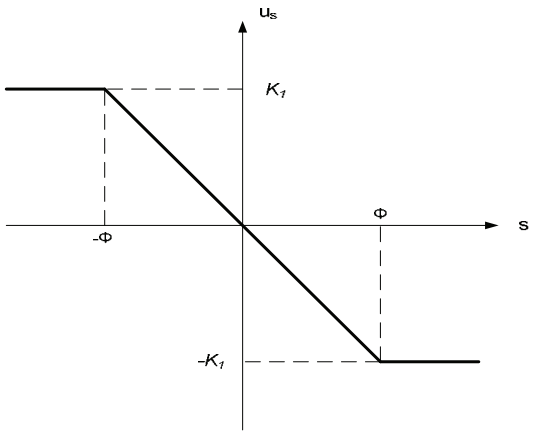


(c)

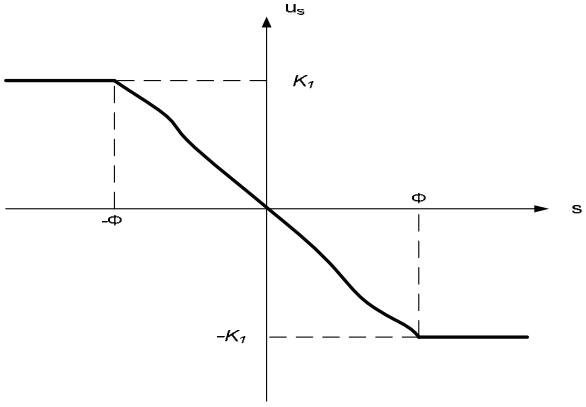
Fig. 6 Fuzzy controller input and output membership functions (a) speed error (b) change in speed error (c) change in the torque command



(a)



(b)



(c)

Fig. 7 Switching functions (a) Sliding mode (b) Sliding mode with boundary layer (c) Fuzzy sliding mode

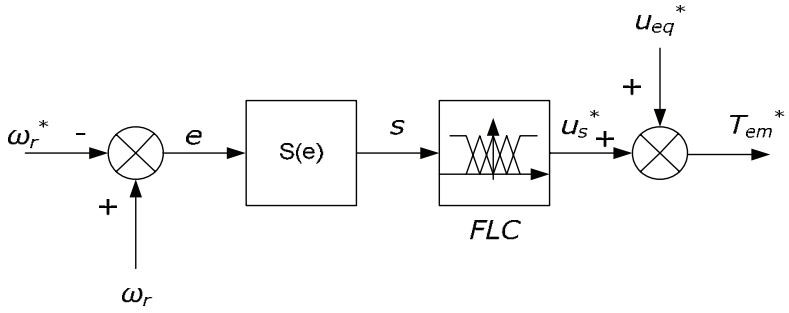
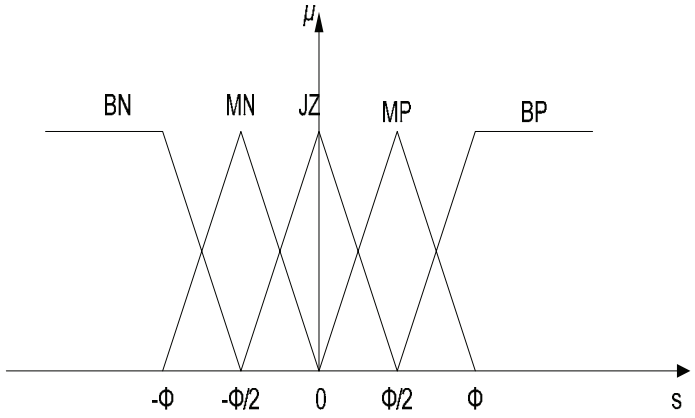
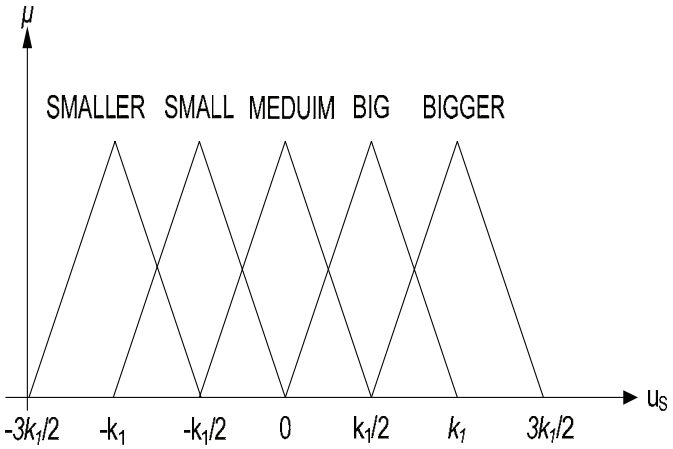


Fig.8 Fuzzy sliding mode speed controller



(a)



(b)

Fig.9 Fuzzy logic membership functions (a) input (b) output

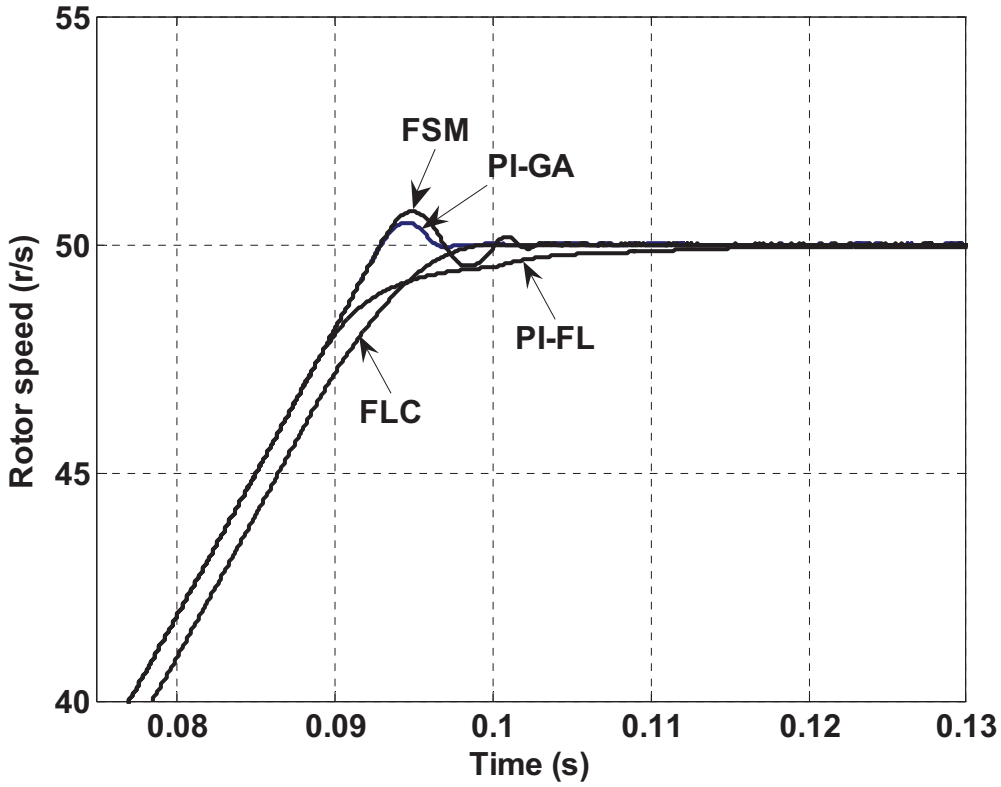


Fig.10 Starting transient performance

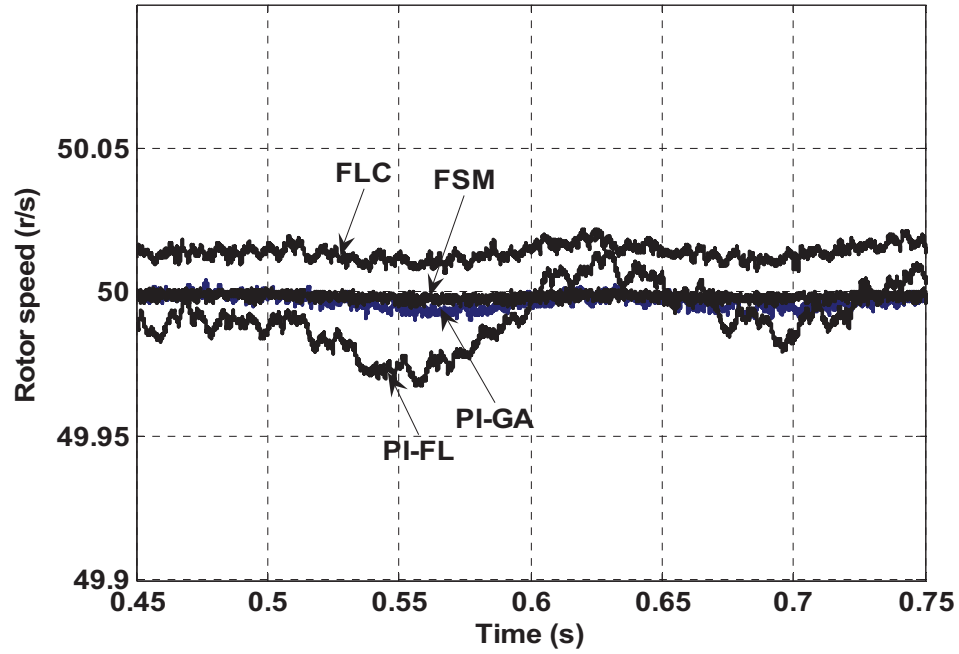
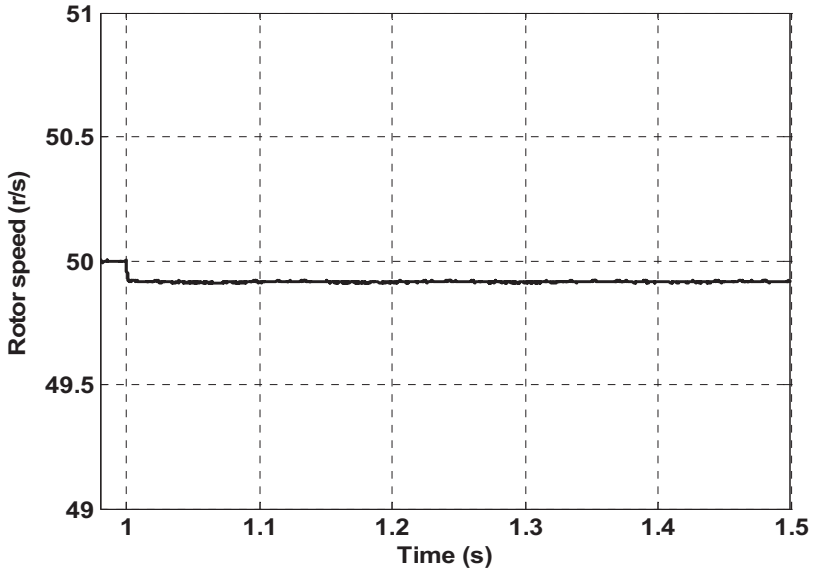
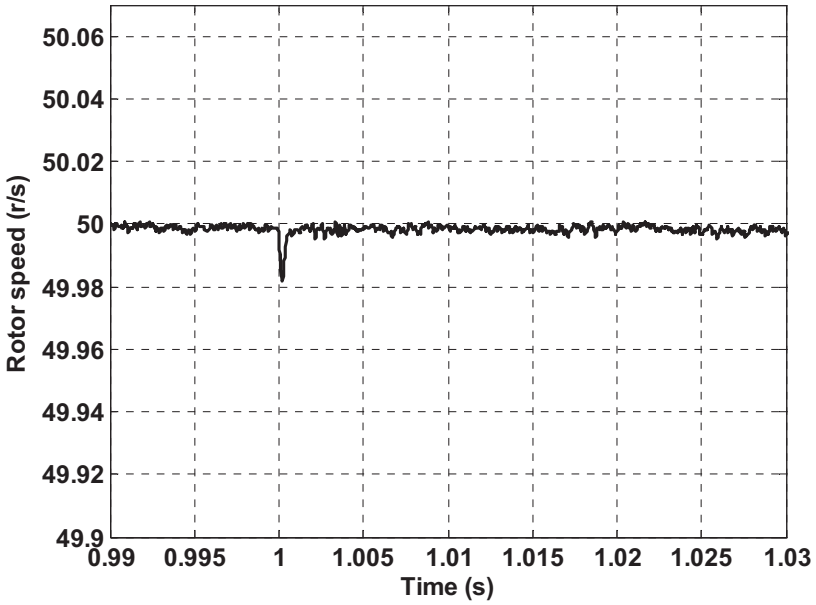


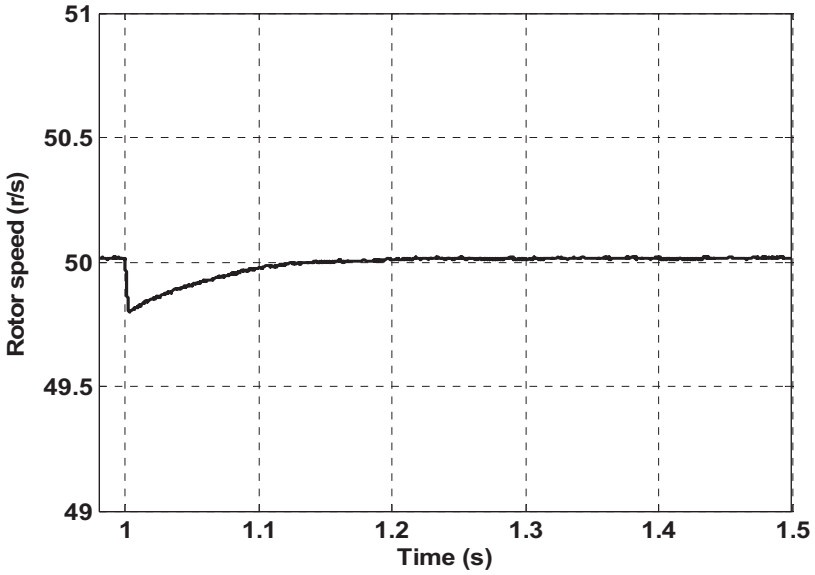
Fig.11 Speed response during R_s variation


51
50.5
50
49.5
49
Rotor speed (r/s)
1
1.1
1.2
1.3
1.4
1.5
Time (s)

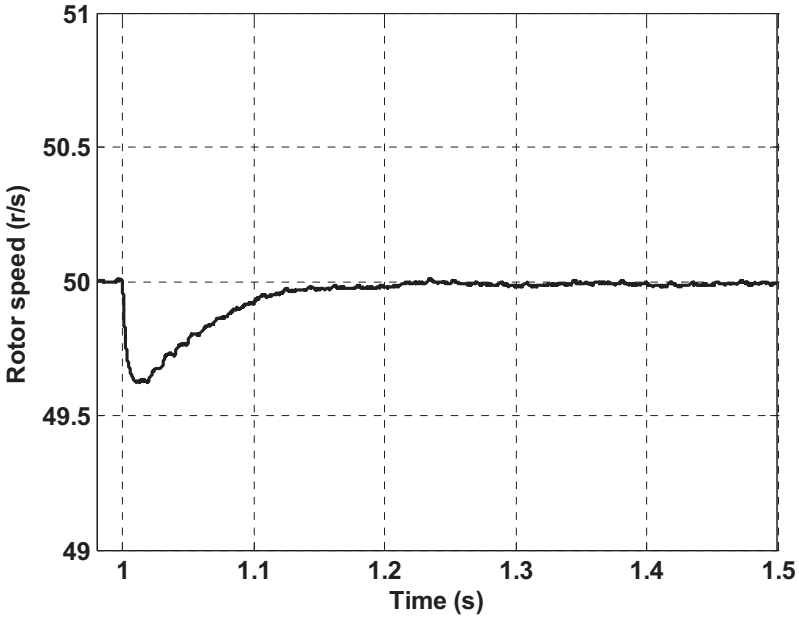
(a)


50.06
50.04
50.02
50
49.98
49.96
49.94
49.92
49.9
Rotor speed (r/s)
0.99
0.995
1
1.005
1.01
1.015
1.02
1.025
1.03
Time (s)

(b)



(c)



(d)

Fig. 12 Speed response due to load change (a) PI-GA (b) FSM (c) FLC (d) PI-FL

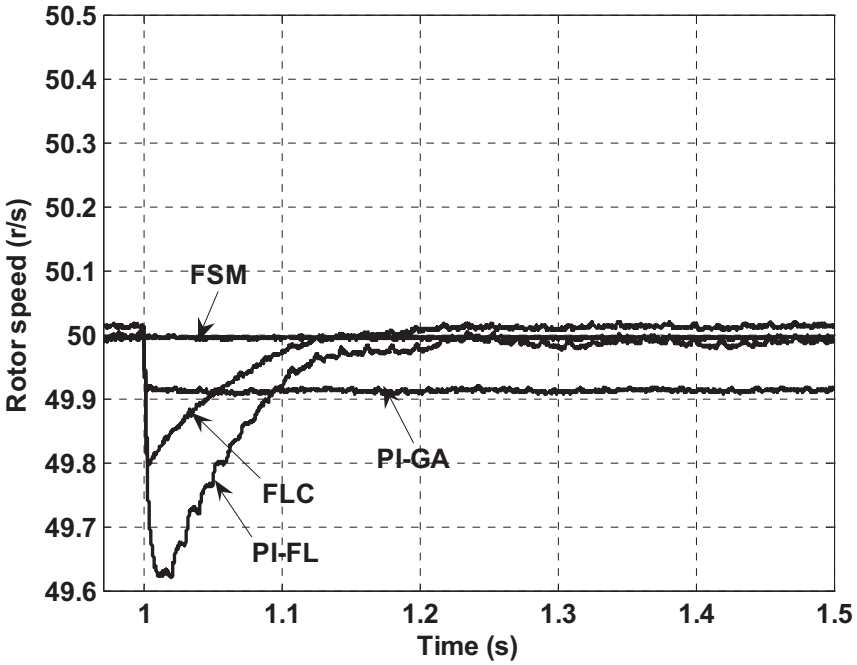


Fig. 13 Disturbance rejection property for different controllers

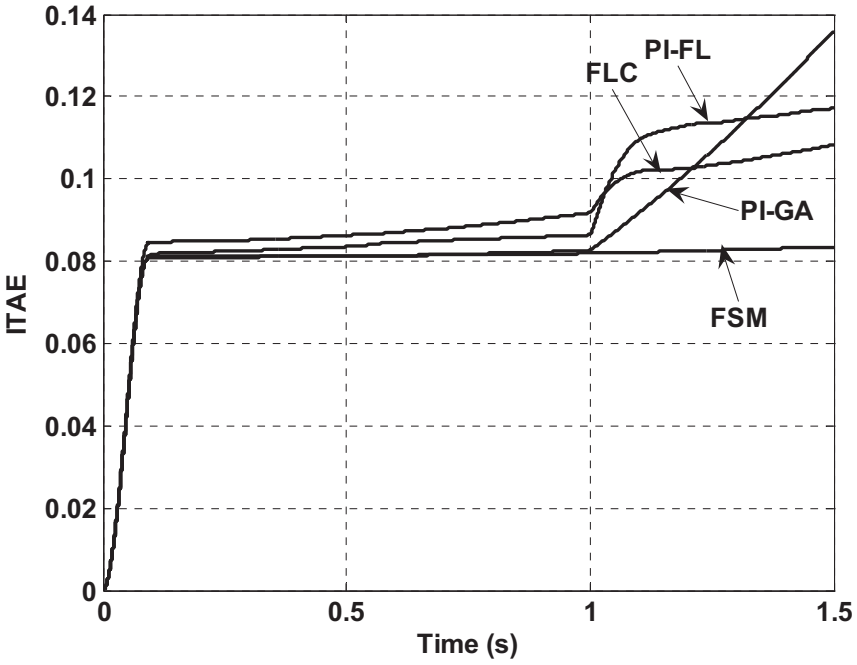


Fig.14 Total ITAE

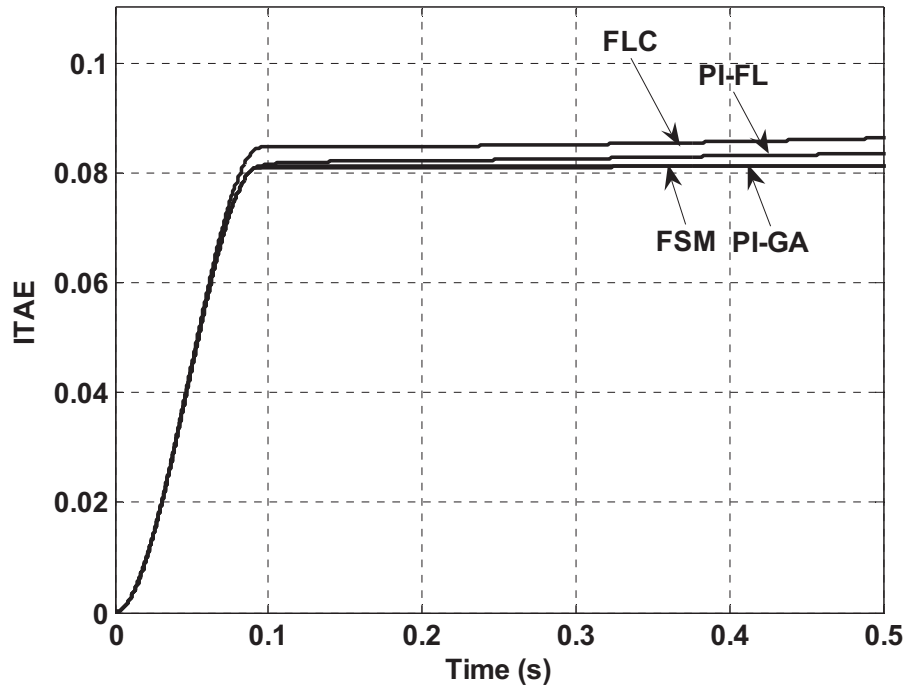


Fig.15 ITAE with normal operating conditions

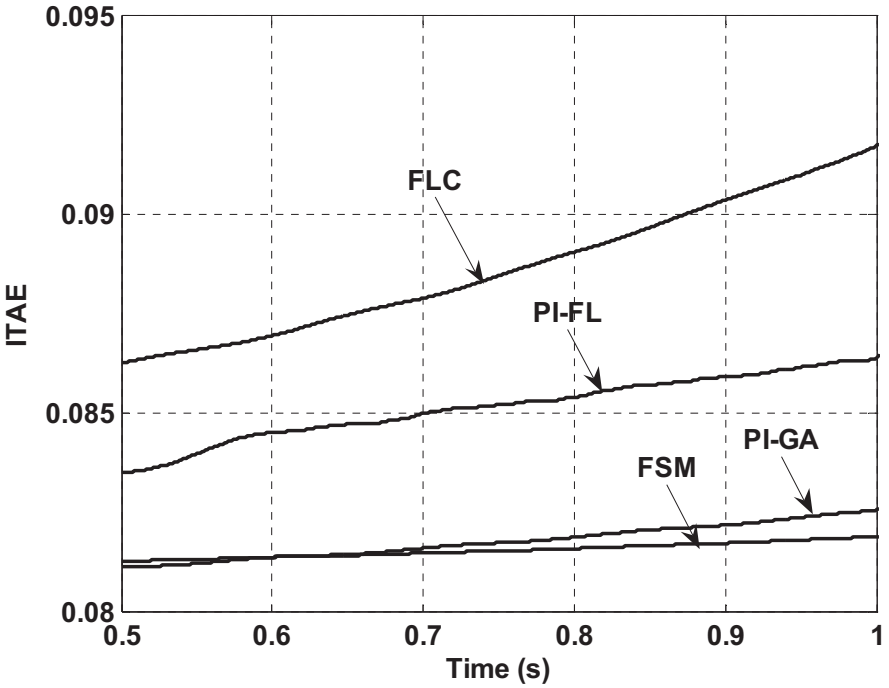


Fig.16 ITAE with stator resistance variation

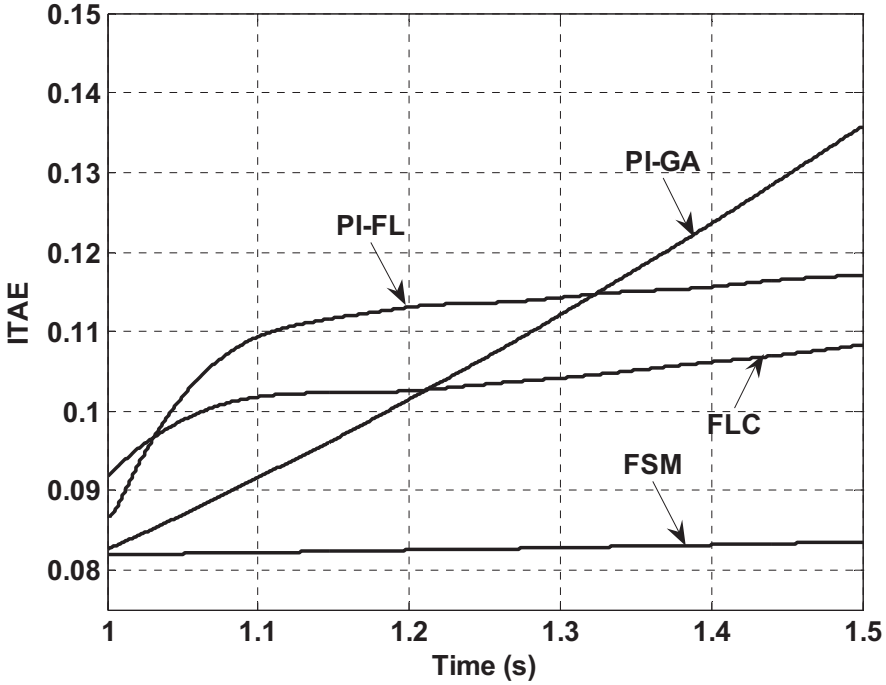


Fig.17 ITAE with load torque change

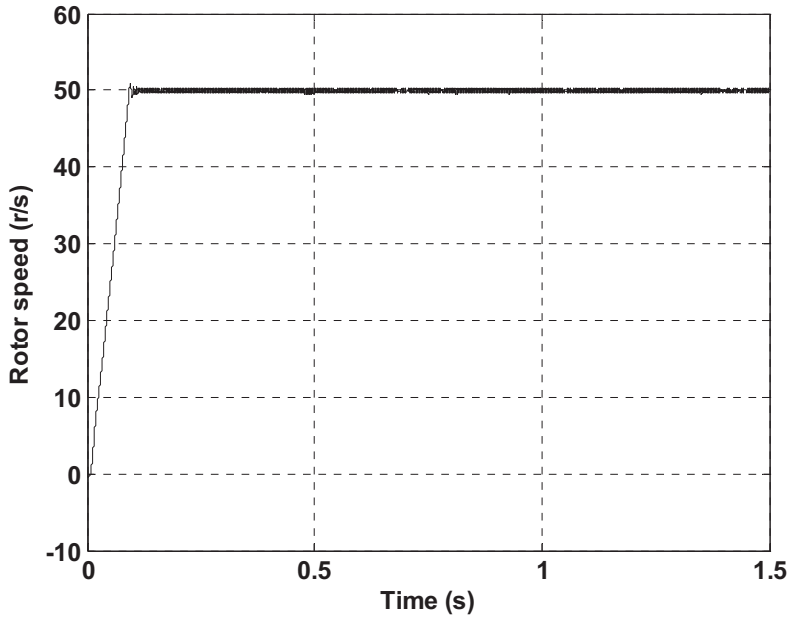


Fig.18 Speed response using conventional sliding mode controller

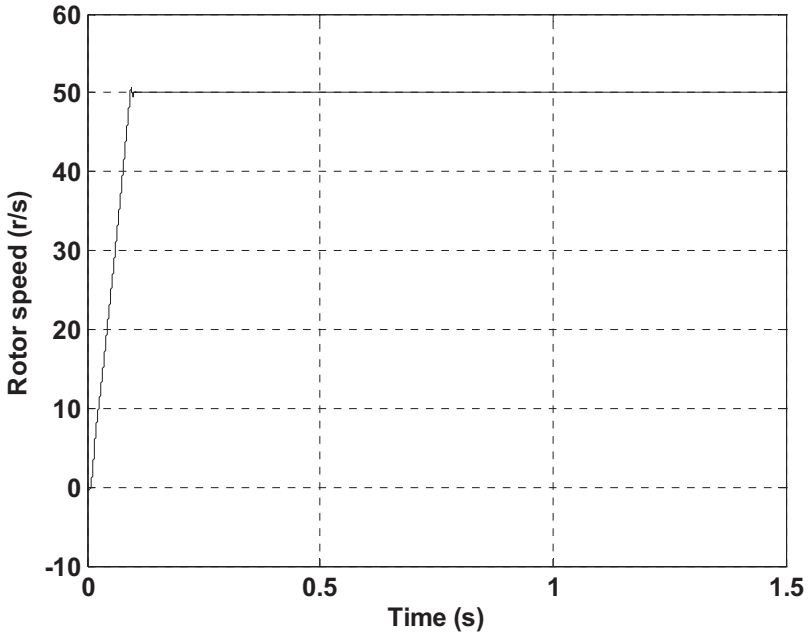


Fig.19 Speed response using fuzzy sliding mode controller

Tables

Table 1
Genetic algorithm parameters

GA property	Value/ Method
Number of generations	10
No of chromosomes in each generation	8
No of genes in each chromosome	2
Chromosome length	40 bit
Selection method	Stochastic Universal Selection (SUS)
Crossover method	Double-point
Crossover probability	0.7
Mutation rate	0.05

Table 2
Fuzzy rules for updating the gain ΔK_p

Δe_ω \ e_ω	NB	NS	ZE	PS	PB
NB	—	PB	PB	PB	—
NS	—	PB	PS	ZE	—
ZE	—	PB	ZE	PB	—
PS	—	ZE	PS	PB	—
PB	—	PB	PB	PB	—

Table 3
Fuzzy rules for updating the gain ΔK_i

Δe_ω \ e_ω	NB	NS	ZE	PS	PB
NB	ZE	NS	NB	NS	ZE
NS	PS	ZE	NS	ZE	PS
ZE	PB	PS	ZE	PS	PB
PS	PS	ZE	NS	ZE	PS
PB	ZE	NS	NB	NS	ZE

Table 4
FLC parameters

Variables	Value
Input scaling factors K_1, K_2	1.1, 0.1
Output scaling factors K_3, K_4	0.2, 1.1
Defuzzification method	Centre of gravity
K_p initial	10
K_i initial	1.2

Table 5
PI-Type fuzzy logic controller rules

Δe_ω \ e_ω	NB	NM	NS	EZ	PS	PM	PB
PB	EZ	PS	PM	PB	PB	PB	PB
PM	NS	EZ	PS	PM	PB	PB	PB
PS	NM	NS	EZ	PS	PM	PB	PB
EZ	NB	NM	NS	EZ	PS	PM	PB
NS	NB	NB	NM	NS	EZ	PS	PM
NM	NB	NB	NB	NM	NS	EZ	PS
NB	NB	NB	NB	NB	NM	NS	EZ

Table 6
Induction motor parameters

Machine parameter	Value
Rated power, [kW]	7.5
Rated voltage, [V]	220
Rated torque, [Nm]	40
Rated frequency, [Hz]	60
R_s, [Ω]	0.15
R_r, [Ω]	0.17
L_s, [mH]	0.035
L_m, [mH]	0.0338
L_r, [mH]	0.035
J [Kg / m^2]	0.14
Pole number	4

Table 7
Summary of results

	PI-GA $K_p = 127$ $K_i = 4$	PI-FL Variable Gains	FLC	FSM
ITAE $(t \in [0s, 0.5s])$	0.0811	0.0835	0.0863	0.0813
ITAE $(t \in [0.5s, 1s])$	0.0014	0.0028	0.0054	0.0006
ITAE $(t \in [1s, 1.5s])$	0.0532	0.0306	0.0166	0.0016
Total ITAE $(t \in [0s, 1.5s])$	0.136	0.117	0.108	0.083
Speed overshoot	1%	0%	0%	1.4%
Torque appl. Initial drop St. st. error	49.92 r/s 0.16%	49.6 r/s 0 after 0.2s	49.8 r/s 0 after 0.1s	49.98 r/s 0 after 1ms

Table 8
Comparison among controllers

Method / Property	PI-GA	PI-FL	FLC	FSM
Starting transient performance	Good	Good	Very good	Good
Robustness	Very good	Very good	Good	Excellent
Disturbance rejection	Poor	Good	Very good	Excellent
St. state performance	Poor	Very good	Moderate	Good
Computational effort	High during tuning and low during drive operation	High	High	Low