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. Due to this difficulty, off-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 neuro-fuzzy 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]. GAs 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 derivation 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 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 optimizer 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

\[
\begin{bmatrix}
    p_{lsd} \\
p_{lsq} \\
p_{lsq}
\end{bmatrix} = \frac{1}{\sigma L_L L_T}
\begin{bmatrix}
    -R_s L_T & L_m R_T & L_m L_T \\
    -L_m R_T & -R_m L_T & L_m L_T \\
    L_m L_T & L_m L_T & \sigma L_L L_T
\end{bmatrix}
\begin{bmatrix}
i_d \\
i_q \\
i_d
\end{bmatrix} + \frac{1}{\sigma L_L L_T}
\begin{bmatrix}
    L_e & 0 & 0 \\
    0 & L_e & 0 \\
    -L_m & 0 & -L_e
\end{bmatrix}
\begin{bmatrix}
v_d \\
v_q \\
v_d
\end{bmatrix}
\tag{1}
\]

The mechanical equation can be written as

\[
T_{em} - T_l = \frac{d}{dt}\left(\frac{1}{p} L_{1} \frac{\omega_r}{\omega_r}ight) + \frac{B}{p} \omega_r
\tag{2}
\]

and the electromagnetic torque:

\[
T_{em} = \frac{3}{2} P L_m (i_d i_q - i_d i_q)
\tag{3}
\]

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, $\omega_r$ is the rotor speed in electric rad/s, $L_s$, $L_r$ and $L_m$ are the stator, rotor and mutual inductances, respectively, $i_d$, $i_q$, $i_d$, $i_q$ are the stator and rotor $d$-axis and $q$-axis current components and $\sigma$ is the leakage coefficient given by

\[
\sigma = 1 - \frac{L_q}{L_d}
\tag{4}
\]

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.

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].

**Table 1**

<table>
<thead>
<tr>
<th>GA property</th>
<th>Value/method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of generations</td>
<td>10</td>
</tr>
<tr>
<td>Number of chromosomes in each generation</td>
<td>8</td>
</tr>
<tr>
<td>Number of genes in each chromosome</td>
<td>2</td>
</tr>
<tr>
<td>Chromosome length</td>
<td>40 bit</td>
</tr>
<tr>
<td>Selection method</td>
<td>Stochastic Universal Selection (SUS)</td>
</tr>
<tr>
<td>Crossover method</td>
<td>Double-point</td>
</tr>
<tr>
<td>Crossover probability</td>
<td>0.7</td>
</tr>
<tr>
<td>Mutation rate</td>
<td>0.05</td>
</tr>
</tbody>
</table>

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].

### 4. PI tuned by fuzzy logic

The control strategy presented in Section 3 suffers from an inability to cope with on-line 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 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.
Table 2
Fuzzy rules for updating the gain $\Delta K_p$

<table>
<thead>
<tr>
<th>$\Delta \omega$</th>
<th>NB</th>
<th>NS</th>
<th>ZE</th>
<th>PS</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>–</td>
<td>PB</td>
<td>PB</td>
<td>PB</td>
<td>–</td>
</tr>
<tr>
<td>NS</td>
<td>–</td>
<td>PB</td>
<td>PS</td>
<td>ZE</td>
<td>–</td>
</tr>
<tr>
<td>ZE</td>
<td>–</td>
<td>PB</td>
<td>ZE</td>
<td>PB</td>
<td>–</td>
</tr>
<tr>
<td>PS</td>
<td>–</td>
<td>ZE</td>
<td>PS</td>
<td>PB</td>
<td>–</td>
</tr>
<tr>
<td>PB</td>
<td>–</td>
<td>PB</td>
<td>PB</td>
<td>PB</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 3
Fuzzy rules for updating the gain $\Delta K_i$

<table>
<thead>
<tr>
<th>$\Delta \omega$</th>
<th>NB</th>
<th>NS</th>
<th>ZE</th>
<th>PS</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>ZE</td>
<td>NS</td>
<td>NB</td>
<td>NS</td>
<td>ZE</td>
</tr>
<tr>
<td>NS</td>
<td>PS</td>
<td>ZE</td>
<td>NS</td>
<td>ZE</td>
<td>PS</td>
</tr>
<tr>
<td>ZE</td>
<td>PB</td>
<td>PS</td>
<td>ZE</td>
<td>PS</td>
<td>PB</td>
</tr>
<tr>
<td>PS</td>
<td>PS</td>
<td>ZE</td>
<td>NS</td>
<td>ZE</td>
<td>PS</td>
</tr>
<tr>
<td>PB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NS</td>
</tr>
</tbody>
</table>

Table 4
FLC parameters

<table>
<thead>
<tr>
<th>Variables</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_1$, $K_2$</td>
<td>1.1, 0.1</td>
</tr>
<tr>
<td>$K_3$, $K_4$</td>
<td>0.2, 1.1</td>
</tr>
<tr>
<td>Defuzzification method</td>
<td>Centre of gravity</td>
</tr>
<tr>
<td>$K_p$ initial</td>
<td>10</td>
</tr>
<tr>
<td>$K_i$ initial</td>
<td>1.2</td>
</tr>
</tbody>
</table>

Table 5
PI-Type fuzzy logic controller rules

<table>
<thead>
<tr>
<th>$\Delta \omega$</th>
<th>NB</th>
<th>NM</th>
<th>NS</th>
<th>EZ</th>
<th>PS</th>
<th>PM</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>PB</td>
<td>EZ</td>
<td>PS</td>
<td>PM</td>
<td>PB</td>
<td>PB</td>
<td>PB</td>
<td></td>
</tr>
<tr>
<td>PM</td>
<td>NS</td>
<td>EZ</td>
<td>PS</td>
<td>PM</td>
<td>PB</td>
<td>PB</td>
<td>PB</td>
</tr>
<tr>
<td>PS</td>
<td>NM</td>
<td>NS</td>
<td>EZ</td>
<td>PS</td>
<td>PM</td>
<td>PB</td>
<td>PB</td>
</tr>
<tr>
<td>PM</td>
<td>NS</td>
<td>NB</td>
<td>NB</td>
<td>NM</td>
<td>NS</td>
<td>EZ</td>
<td>PS</td>
</tr>
<tr>
<td>PB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NM</td>
<td>NS</td>
<td>EZ</td>
<td>PS</td>
</tr>
<tr>
<td>PB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NS</td>
<td>EZ</td>
</tr>
</tbody>
</table>

Fig. 4. Flow chart of fuzzy self-tuning PI speed controller.

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 [1,5,15]. 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 MEDIUM, NS = NEGATIVE SMALL, EZ = ZERO, PS = POSITIVE SMALL, PM = POSITIVE MEDIUM, 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.

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]

\[ e(t) = \omega_r - \omega_r^* \]

Defining the attractive switching surface as

\[ s(t) = e(t) - \int_0^t k e(\tau) \, d\tau \]

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

\[ \dot{e}(t) = ke(t); \quad k < 0 \]

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

\[ u^* = u_{eq} + u_s \]

\[ u_s = -k_1 \, \text{sign}(s) \]  

The sign function is given by

\[ \text{sign}(s) = \begin{cases} -1 & \text{for } s < 0 \\ +1 & \text{for } s > 0 \end{cases} \]

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]. \( x(t) \) 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

\[ u_s = -k_1 \, \text{sat}\left(\frac{s}{\phi}\right) \]

where \( \phi \) represents the thickness of the boundary layer. The saturation function is defined as [13]

\[ \text{sat}\left(\frac{s}{\phi}\right) = \begin{cases} \left(\frac{s}{\phi}\right) & \text{for } \left|\frac{s}{\phi}\right| \leq 1 \\ \text{sign}\left(\frac{s}{\phi}\right) & \text{for } \left|\frac{s}{\phi}\right| > 1 \end{cases} \]

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 \(-k_1 \, \text{sat}\left(\frac{s}{\phi}\right)\) is replaced by a fuzzy inference system as shown in Fig. 7(c) in order to smooth the control action [15]. The choice of \( \phi \) is crucial; small values of \( \phi \) 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 and 9 [15,16]. The If-Then rules of the fuzzy logic controller can be written as [15,16]
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

---

### Table 6

<table>
<thead>
<tr>
<th>Machine parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rated power (kW)</td>
<td>7.5</td>
</tr>
<tr>
<td>Rated voltage (V)</td>
<td>220</td>
</tr>
<tr>
<td>Rated torque (N m)</td>
<td>40</td>
</tr>
<tr>
<td>Rated frequency (Hz)</td>
<td>60</td>
</tr>
<tr>
<td>( R_s ) (( \Omega ))</td>
<td>0.15</td>
</tr>
<tr>
<td>( R_r ) (( \Omega ))</td>
<td>0.17</td>
</tr>
<tr>
<td>( L_s ) (mH)</td>
<td>0.035</td>
</tr>
<tr>
<td>( L_m ) (mH)</td>
<td>0.0338</td>
</tr>
<tr>
<td>( L_r ) (mH)</td>
<td>0.035</td>
</tr>
<tr>
<td>( J ) (kg m(^2))</td>
<td>0.14</td>
</tr>
<tr>
<td>Pole number</td>
<td>4</td>
</tr>
</tbody>
</table>

---

Fig. 7. Switching functions (a) sliding mode, (b) sliding mode with boundary layer and (c) fuzzy sliding mode.

---

Fig. 8. Fuzzy sliding mode speed controller.

---

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

- 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.
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 $t = 0.5$ s. 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 $t = 0.5$ s. Stator resistance is chosen because the performance of DTC drive is greatly affected by the variation of this parameter especially at low speed. At $t = 1$ s, 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 on-line 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 $K_1$; however, its value should be selected high enough to make the manifold $s = 0$ in (7) attractive [15]. The value of $\phi$ 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 0.1 s without overshoot. PI-FL has an overdamped response where the motor speed builds in 0.115 s 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 off-line tuning process. Due to their adaptive features, the three other control strategies show fast disturbance rejection.

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.
Table 7
Summary of results

<table>
<thead>
<tr>
<th></th>
<th>PI-GA $K_p = 127$, $K_i = 4$</th>
<th>PI-FL variable gains</th>
<th>FLC</th>
<th>FSM</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITAE ($t \in [0 \text{ s}, 0.5 \text{ s}]$)</td>
<td>0.0811</td>
<td>0.0835</td>
<td>0.0863</td>
<td>0.0813</td>
</tr>
<tr>
<td>ITAE ($t \in [0.5 \text{ s}, 1 \text{ s}]$)</td>
<td>0.0014</td>
<td>0.0028</td>
<td>0.0054</td>
<td>0.0006</td>
</tr>
<tr>
<td>ITAE ($t \in [1 \text{ s}, 1.5 \text{ s}]$)</td>
<td>0.0532</td>
<td>0.0306</td>
<td>0.0166</td>
<td>0.0016</td>
</tr>
<tr>
<td>Total ITAE ($t \in [0 \text{ s}, 0.5 \text{ s}]$)</td>
<td>0.136</td>
<td>0.117</td>
<td>0.108</td>
<td>0.083</td>
</tr>
<tr>
<td>Speed overshoot</td>
<td>1%</td>
<td>0%</td>
<td>0%</td>
<td>1.4%</td>
</tr>
<tr>
<td>Torque appl. Initial drop, steady state error</td>
<td>49.92 rad/s, 0.16%</td>
<td>49.6 rad/s, 0 after 0.2 s</td>
<td>49.8 rad/s, 0 after 0.1 s</td>
<td>49.98 rad/s, 0 after 1 ms</td>
</tr>
</tbody>
</table>

Fig. 18. Speed response using conventional sliding mode controller.

Fig. 19. Speed response using fuzzy sliding mode controller.

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 1 ms as shown in Fig. 12(b). The FLC and PI-FL controllers show speed drops to 49.8 and 49.6 rad/s but are corrected back after 0.1 and 0.2 s, respectively, as shown in Figs. 12(c) and (d). The load torque disturbance rejection property of the different controllers is shown together in Fig. 13.

### 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 $t = 1 \text{ s}$, 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.

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.

### 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 off-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 conver-

Table 8
Comparison among controllers

<table>
<thead>
<tr>
<th>Method</th>
<th>PI-GA</th>
<th>PI-FL</th>
<th>FLC</th>
<th>FSM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property</td>
<td>Good</td>
<td>Good</td>
<td>Very good</td>
<td>Good</td>
</tr>
<tr>
<td>Starting transient performance</td>
<td>Good</td>
<td>Very good</td>
<td>Good</td>
<td>Excellent</td>
</tr>
<tr>
<td>Robustness</td>
<td>Very good</td>
<td>Good</td>
<td>Very good</td>
<td>Excellent</td>
</tr>
<tr>
<td>Disturbance rejection</td>
<td>Poor</td>
<td>Good</td>
<td>Very good</td>
<td>Good</td>
</tr>
<tr>
<td>Steady-state performance</td>
<td>Poor</td>
<td>Very good</td>
<td>Moderate</td>
<td>Good</td>
</tr>
<tr>
<td>Computational effort</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td></td>
</tr>
</tbody>
</table>
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.

Acknowledgement

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