Abstract - In this paper a self-tuning fuzzy logic speed controller is proposed for the high performance drives of induction motor. Here, the output gain of the controller is adjusted on-line by fuzzy rules according to the current trend of the controlled process. Tuning of the output gain has been given the highest priority because of its strong influence on the performance and stability of the system. The ability of the proposed controller is verified in the presence of external load variations using MATLAB simulation with the comparison of conventional PI controller. Simulation results showed the effectiveness and robustness of the proposed approach.

I. INTRODUCTION

This scheme is based on the fact that irrespective of the nature of the process to be controlled and the control policy to be adopted, a skilled human operator always tries to manipulate the process input, usually by adjusting the controller gain based on the controller process states (generally $e$ and $\Delta e$) to get the process optimally controlled [1-2]. The exact manipulation strategy of an operator is quite complex in nature and possibly no mathematical model can replace it accurately [3]-[5].

Standards regulators with fixed parameters may be insufficient in controlling systems, such as the robotic arms, that are subject to large variations of inertia and load during their normal operating cycles. However, more sophisticated controllers are required, such as adaptive regulators, self-tuning regulators, which in presence of variations of plant parameters, are able to modify their features in order to maintain the desired dynamic behavior of the system [6]. Different type of adaptive FLC’s (Fuzzy Logic Controller) have been developed and proposed in the last years. In [7] a simple algorithm for modifying triangular input membership functions has been used. Another approach to adaptation described in [8], [9] has involved modification of the whole fuzzy rule base.

In this paper we propose a simple but robust model independent self-tuning scheme, where the controller gain is adjusted continuously with the help of fuzzy rules. Here, our objective is to adapt only the output SF (Scaling Factor) for given input SF’s. Tuning of the output SF has been given the highest priority because of its strong influence on the performance and stability of the system [10]. The proposed scheme is applied to the speed control of an IFOC. The simulation results show its effectiveness in case of parameter variation of the system.

II. DYNAMIC MODEL OF THE INDUCTION MACHINE

The model of the squirrel-cage induction machine can be expressed in terms of d- and q-axes quantities resulting in the following equations:

$$X = AX + BU$$  \hspace{1cm} (1)

where definitions are given in “(2)-(5)”. The electromagnetic torque and the mechanical equations can be written as follows:

$$T_e = \frac{3}{2} p L_m (i_{d1} i_{q1} - i_{q1} i_{d1})$$  \hspace{1cm} (2)

$$J \frac{d\Omega}{dt} + f \Omega = T_e - T_L$$  \hspace{1cm} (3)

where $J$ is the moment of inertia, $f$ the viscous friction coefficient and $T_L$ the load torque. A simulation model of the induction machine has been built using top and bottom equations.
The system, which is presented in “Fig. 1”, is an indirect field oriented control (IFOC)-based induction motor drive. It consists mainly of a squirrel-cage induction motor, a voltage-regulated pulse width modulated inverter, fuzzy speed controller and fuzzy rotor resistance estimator. The induction motor is a three phase, Y connected, four pole, 1.5 Kw, 1420tr/min 220/380V, 50Hz and 6.4/3.7A.

Under field orientation condition, the d-q equations of the motor in the synchronous reference frame are:

\[
0 = \frac{d}{dt} i_{qs} + \frac{L_m}{L_r} i_{ds} + R_r i_{qs} + \omega_s \psi_{ds}
\]

\[
0 = \frac{d}{dt} i_{dr} + \omega_s i_{qs} + \frac{L_m}{L_r} i_{qs} + \frac{L_m}{L_s} i_{ds} + R_r i_{dr}
\]

\[
0 = L_m i_{qs} + L_r i_{qr}
\]

\[
0 = L_m i_{ds} + L_r i_{dr} = \psi_{ds}
\]

where \( R_n, R_r, L_n, L_m, L_{dr}, L_{ds} \) are motor parameters, \( i_{ds}, i_{qs}, i_{dr}, i_{qr}, \psi_{dr}, \psi_{ds} \) are motor currents and fluxes, and \( \omega_s \) is slip frequency. The equations describing the motor operation in decoupling mode are deduced from \( (2), (7)-(10) \):

\[
\omega_s = \frac{L_m}{\psi_r} \left( \frac{R_r}{L_r} \right) i_{qs}
\]

\[
T_e = \frac{3}{2} \frac{L_m}{L_r} \psi_r i_{qs}
\]

\[
\left( \frac{L_r}{R_r} \right) \frac{d\psi_r}{dt} + \psi_r = L_m i_{ds}
\]

In literature fuzzy logic algorithms with adaptive characteristics can be found under various names: self-tuning, self-organizing, self-learning, adaptive and expert algorithms or fuzzy logic algorithms with a varying rule base. Our proposed FLC is tuned by modifying the output SF of an existing FLC so we describe it as a self-tuning FLC.

The block diagram of the proposed self-tuning FLC is shown in “Fig. 3”. The output SF (gain) of the controller is modified by a self-tuning mechanism, which is shown by the dotted boundary.

In order to design a self-tuning fuzzy logic controller, the following steps must be performed:

1) development of a suitable rule set;
2) selection of input/output variables and their quantization in fuzzy sets;
3) definition of membership functions to be associated to the input/output variables;
4) Selection of the inference method;
5) Selection of the defuzzification technique.

### 1. Membership Functions

All membership functions (MF’s) for: 1) controller inputs, i.e., error (e) and change of error (\( \Delta e \)) and 2) incremental change in controller output (\( \Delta T^* \)), are defined on the common interval \([-1,1]\); \([-10,10]\) respectively, whereas the MF’s for the gain updating factor (\( \alpha \)) is defined on \([0,10]\). We use symmetric triangles (except the two MF’s at the extreme ends which are trapezoidal) as shown in “Fig. 2”. These input membership functions are used to transfer crisp inputs into fuzzy sets.
2. Scaling Factors

The values of the actual inputs $e$ and $\Delta e$ are mapped onto $[-1,1]$ by the input SF’s $G_e$ and $G_{\Delta e}$, respectively. On the other hand, the actual output of the self-tuning FLC is obtained by using the effective SF $(\alpha G_{\Delta T^*})$ as shown in “Fig. 3”. Selection of suitable values for $G_e$, $G_{\Delta e}$, and $G_{\Delta T^*}$ are made based on the knowledge about the process to be controlled and sometimes through trial and error to achieve the best possible control performance.

We propose to compute $\alpha$ on-line using a model independent fuzzy rule base defined in terms of $e$ and $\Delta e$. The relationships between the SF’s and the input and output variables of the self-tuning FLC are as follows:

\[ e_N = G_e e \] (14)
\[ \Delta e_N = G_{\Delta e} \Delta e \] (15)
\[ \Delta T^* = \alpha G_{\Delta T^*} \] (16)

The value of $G_{\Delta T^*}$ is constant for a particular type of conventional FLC. But the gain of our self-tuning FLC does not remain fixed while the controller is in operation, rather it is modified in each sampling time by the gain updating factor $\alpha$, depending on the trend of the controlled process output. The reason behind this on-line gain variation is to make the controller respond according to the desired performance specifications.

3. The Rule Bases

The expert experience has been incorporated into a knowledge base with 49 rules (7x7). Then, the inference engine based on the input fuzzy sets, uses appropriate IF-THEN rules in the knowledge base to imply the final output fuzzy sets as shown in the “Fig. 4”, where NB, NM, NS, ZE, PVS, PS, PM, PB, PMB, PVB correspond to Negative Big, Negative Medium, Negative Small, Zero, Positive Very Small, Positive Small, Positive Medium, Positive Big, Positive Medium Big, Positive Very Big, respectively.

Some of the important considerations that have been taken into account for determining the rules are as follow:

1. To make the controller produce a lower overshoot and reduce the settling time (but not at the cost of increased rise time) the controller gain is set at a small value when the error is medium big (it may be +ve or -ve), but $e$ and $\Delta e$ are of opposite signs. For example, if $e$ is PM and $\Delta e$ is NS THEN $\alpha$ is PS or if $e$ is NM and $\Delta e$ is PS THEN $\alpha$ is PS. Now if the error is big but both $e$ and $\Delta e$ are of the same sign (i.e., the process is now not only far away from the set point but also it is...
moving farther away from it), the gain should be made very large to prevent from further worsening the situation. This has been realized by the rules of the form: if $e$ is PB and $\Delta e$ is PS THEN $\alpha$ is PVB or if $e$ is NB and $\Delta e$ is NS THEN $\alpha$ is PVB.

2) Depending on the process trend, there should be a wide variation of the gain around the set point (i.e., when $e$ is small) to avoid large overshoot and undershoot. For example, overshoot will be reduced by the rule IF $e$ is ZE and $\Delta e$ is NM THEN $\alpha$ is PM. This rule indicates that the process has just reached the set point but it is moving away upward from the set point rapidly. Similarly, a large undershoot can be avoided using the rules of the form: IF $e$ is NS and $\Delta e$ is PS THEN $\alpha$ is ZE. This type of gain variation around the set point will also prevent excessive oscillation and as a result the convergence rate of the process to the set point will be increased.

3) To improve the control performance under load disturbance, the gain should be sufficiently large around the steady-state condition. IF $e$ is PM and $\Delta e$ is PS THEN $\alpha$ is PVB or IF $e$ is NM and $\Delta e$ is NS THEN $\alpha$ is PMB. At steady state (i.e., $e=0$ and $\Delta e \approx 0$) controller gain should be very small (e.g., IF $e$ is ZE and $\Delta e$ is ZE THEN $\alpha$ is ZE) to avoid chattering problem around the set point. It is very important to note that the rule base for the computation of $\alpha$ will always be dependent on the choice of the rule base for the controller. Any significant change in the controller rule base may call for changes in the rule base for $\alpha$ accordingly.

Once all the rules are established, the implied fuzzy set is transformed to a crisp output by the center of gravity defuzzification technique as given by the formula (12), $z_i$ is the numerical output at the $i$th number of rules and $\mu(z_i)$ corresponds to the value of fuzzy membership function at the $i$th number of rules. The summation is from one to $n$, where $n$ is the number of rules that apply for the given fuzzy inputs, [1]-[2].

$$z = \frac{\sum_{i=1}^{n} z_i \cdot \mu(z_i)}{\sum_{i=1}^{n} \mu(z_i)} \quad (17)$$

The crisp output $\Delta T^*$ is multiplied by the gain factor $\alpha.G_{\Delta T}$ and then integrated to give:

$$T_e(k) = T_e(k-1) + \Delta T^* \cdot \alpha.G_{\Delta T} \quad (18)$$

This torque component command is used as an input to the F.O.C. block of “Fig. 1”.

### IV. RESULTS

In order to verify the validity of the proposed self-tuning fuzzy logic controller, several simulations are carried out using MATLAB and Simulink software. The configuration of the overall control system is shown in “Fig. 1”.

Simulations are based on the facts that whether the proposed controller is better and more robust than the PI linear controller or not. For the comparison, simulations of the speed response were performed according to the speed command variation, the load variation, inertia variation, viscous friction variation, rotor resistance variation and speed tracking variation of the induction motor.

Table 1 shows the parameter of the used induction motor for simulation whose general specifications are
1.5 Kw, 1420 rev/min, 380V, 50 Hz, 4 pole. The sampling time of the controller is 100 μs.

TABLE I.

<table>
<thead>
<tr>
<th>Induction Motor Parameters Used For Simulation</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Stator resistance</td>
<td>4.85 Ω</td>
</tr>
<tr>
<td>Rotor resistance</td>
<td>3.805 Ω</td>
</tr>
<tr>
<td>Stator inductance</td>
<td>274mH</td>
</tr>
<tr>
<td>Rotor inductance</td>
<td>274mH</td>
</tr>
<tr>
<td>Magnetising inductance</td>
<td>258mH</td>
</tr>
<tr>
<td>Rotor inertia</td>
<td>0.031 Kg.m² 0.00114</td>
</tr>
<tr>
<td>Viscous friction</td>
<td>Kg.m²/s</td>
</tr>
<tr>
<td>Rated torque</td>
<td>10 Nm</td>
</tr>
</tbody>
</table>

“Fig. 5,6”show speed response waveforms when applying full load. In all point view such as rejection of load disturbance and speed recovery time, proposed self-tuning fuzzy controller is better than conventional PI controller. “Fig.9,10”show speed response with inertia variation, where the load moment of inertia J_L is changed to four times of the nominal value J_LN and full load is applied at 6 sec. We can see that proposed fuzzy self-tuning controller is more robust to the inertia variation.
“Fig.11,12” show speed response with viscous friction variation at nominal value and “Fig.13,14” show speed response with viscous variation, where the load viscous friction $f_L$ is changed to ten times of its nominal value $f_{LN}$ and full load is applied at 6 sec. Here again, the performance of proposed controller is much better than the PI controller. One can observe as well that varying the viscous friction from its nominal value to ten times by using the same controller has no effect on the speed response.

“Fig.15,16” show speed tracking performance under no load. The proposed controller reacts perfectly and tracks the command speed with almost no steady state error. The PI controller is less performing especially at the starting and when changing speed from directs to reverse. A slight overshoot appears on the corners. “Fig.17, 18” show the effect of rotor resistance on the speed when its value has doubled at 6 sec and full load is applied from the beginning. However, the proposed controller is still performing perfectly with only a maximum drop of speed of 0.99 rad/sec whereas the PI controller has a drop of speed of 10 rad/s.
V. CONCLUSION

In this paper, an on-line self-tuning fuzzy logic controller for induction motor drives is presented. The structure of the proposed controller is based on the self-tuning controller that can tune its control output gain to be autonomous relating to system parameter variations.

According to simulation results using Matlab (Simulink software) and in all point of view such as variable speed characteristics, speed recovery time, inertia variation, speed tracking, viscous friction variation and effect on the speed by doubling rotor resistance, the proposed fuzzy logic controller is far better and more robust than the conventional PI controller. Actually, implementation is in progress.

To achieve more improved performances and a reduced amount of computations and complexity, a hybrid controller in which a fuzzy rule based system will modulate the output of a non fuzzy controller such as PID may be designed. The output modulation may be realized using a SF as done in the present case. This is currently under investigation.

REFERENCES