In this paper a hybrid intelligent control based torque tracking approach for Doubly Fed Asynchronous Motor (DFAM) drive is proposed. First, a mathematical model of DFAM written in an appropriate d-q reference frame is established to investigate simulations. In order to control the rotor currents of DFAM, a torque tracking control law is synthesized using PI controllers, under conditions of the stator side power factor is controlled at unity level. A four layer Neural Network (NN) is used to adjust input and output parameters of membership functions in a fuzzy logic controller (FLC). The back propagation learning algorithm is used for training this network. The performances of neuro-fuzzy controller (NFC) which is based on the torque tracking control algorithm are investigated and compared to those obtained from the PI controller. Results obtained in Matlab/Simulink environment show that the NFC is more robust, superior dynamic performance and hence found to be a suitable replacement of the conventional PI controller for the high performance drive applications.

Keywords: Doubly-Fed Asynchronous Motor, Torque Tracking Control, Hybrid control, power factor.

1. INTRODUCTION

Doubly-Fed Asynchronous Motor (DFAM) is an electrical asynchronous three-phases machine with open rotor windings which can be fed by external voltages. The typical connection scheme of this machine is is reported in fig. 1. The stator windings are directly connected to the line grid, while the rotor windings are controlled by means of an inverter. This solution is very attractive for all the applications where limited speed variations around the synchronous velocity are present, since the power handled by the converter at rotor side will be a small fraction (depending on the slip) of the overall system power,[1]÷[4].

The DFAM control issues are traditionally handled by fixed gain proportional integral (PI) controllers. However, the fixed gain controllers are very sensitive to parameter variations, cannot usually provide good dynamic performance, etc. So, the controller parameters have to be continually adapted. The problem can be solved by several adaptive control techniques such as model reference adaptive control (MRAC), sliding mode control (SMC),[5] etc. The design of all of the above controllers depends on the exact system mathematical model. During the past decade, various adaptive and robust controllers based on variable structure controller [6-7]. And fuzzy-neural techniques [8][9][10] are proposed for electrical drives. Neuro-fuzzy systems combine the advantageous of neural networks and fuzzy logic systems. Neural networks provide connectionist structure and learning abilities to the fuzzy logic systems, and the fuzzy logic systems provide the neural networks with a structural framework with high-level fuzzy IF-THEN rule thinking and reasoning. Neural network-based fuzzy systems, which have the learning ability of neural networks to realize the fuzzy logic inference system, are gained popularity in the control of nonlinear systems,[11]. Neuro-fuzzy control technique does not need accurate system modeling and gives superior performance than the conventional proportional-integral (PI) control.
In this paper torque tracking control strategy is achieved by adjusting rotor currents and using stator voltage vector oriented reference frame. The performances of neuro-fuzzy controller (NFC) which is based on the torque tracking control approach are investigated and compared to those obtained from the PI controller. Results obtained show that the NFC is more robust and superior dynamic performance.

2. MATHEMATICAL MODEL OF THE DFAM

The equivalent two-phase model of the symmetrical DFAM is represented in stator voltage-vector oriented frame (d-q) is:

\[ J \frac{d\omega}{dt} = [p\mu(\Psi_{qs} \cdot i_{dr} - \Psi_{qs} \cdot i_{qr}) - T_L - f\omega] \]  

(1)

\[ \frac{d\Psi_{ds}}{dt} = -\alpha_s \Psi_{ds} + \omega_s \Psi_{qs} + \alpha_s M i_{dr} + V_{ds} \]  

(2)

\[ \frac{d\Psi_{qs}}{dt} = -\alpha_s \Psi_{qs} - \omega_s \Psi_{ds} + \alpha_s M_i_{qr} + V_{qs} \]  

(3)

\[ \frac{di_{dr}}{dt} = -\gamma_i i_{dr} + \omega_i i_{qr} + \alpha_s \beta \Psi_{ds} - \beta \omega \Psi_{qs} - \beta V_{ds} + \frac{1}{\sigma_r} V_{dr} \]  

(4)

\[ \frac{di_{qr}}{dt} = -\gamma_i i_{qr} - \omega_i i_{dr} + \alpha_s \beta \Psi_{qs} + \beta \omega \Psi_{ds} - \beta V_{qs} + \frac{1}{\sigma_r} V_{qr} \]  

(5)

\[ \hat{\Theta} = \omega \]  

(6)

Positive constants related to DFAM electrical parameters are defined as:

\[ \alpha_s = R_s/L_s; \sigma_t = L_t(1-M^2/L_s L_r); \beta = M/(L_s \sigma_t); \mu = 3M/2L_s; \gamma_t = R_t/\sigma_t + (R_s M^2)/L_s 2\sigma_t \]

3. TORQUE TRACKING CONTROL ALGORITHM FOR DFAM

In this paper we present a stator voltage vector oriented reference frame instead of the stator flux oriented. Specifically, the torque tracking, stator-side unity power factor control problem are considered, with the requirement to achieve stable rotor current and stator flux error dynamics, independently of speed behavior. Conditions of stator flux field orientation and line voltage orientation are equivalent if the stator side power factor is controlled at unity level, [12]. Under such condition the stator flux modulus is not a free output variable.
but it is a function of the produced electromagnetic torque. This last is a positive trapezoidal torque reference.

The reactive component of the stator current is practically equal to zero as it is the given working condition of DFAM control algorithm, [13]. Under this reference frame, the flux errors are defined as:

\[ \dot{\Psi}_d = \Psi_d^*, \quad \dot{\Psi}_q = \Psi_q - \Psi^* \]

\[ V_{ds} = U_m = V_s, \quad V_{qs} = 0, \quad i_{qs} = 0. \]

where \( \Psi^* \) is the flux level reference trajectory. The setting of different vectors and transformation angle is represented on fig. 2. Fig. 3. shows the control diagram of power factor. The complete equations of the vector control of the doubly fed asynchronous motor are given by:

Stator flux vector controller:

\[ \dot{i}_{qr} = \frac{1}{\alpha_i M} \left( \alpha_s \Psi^* + \dot{\Psi}^* \right) \quad (7) \]

\[ \Psi^* = \frac{-U_m - \left(U_m^2 - 4(2/3)\omega_s R_s T_e^*\right)}{2\omega_s} \quad (8) \]

Torque controller:

\[ \dot{i}_{dr}^* = \frac{T_e^*}{\mu \Psi_{qs}^*} \quad (9) \]

Rotor current controller:

\[ U_{dr} = \sigma_i [\gamma \ddot{i}_{dr} - \omega \dot{i}_{dr}^* + \beta \alpha \Psi^* + \beta U_m + \frac{d\ddot{i}_{dr}}{dt} - k_p \ddot{i}_{dr} + x_d] \]

\[ U_{qr} = \sigma_i [\gamma \ddot{i}_{qr} + \omega \dot{i}_{qr}^* - \beta \alpha \Psi^* + \frac{d\ddot{i}_{qr}}{dt} - k_p \ddot{i}_{qr} + x_q] \quad (10) \]

where:

\[ \ddot{s}_d = -k_i \ddot{i}_d; \quad \ddot{i}_d = i_{dr} - \dot{i}_{dr}^* \]

\[ \ddot{s}_q = -k_i \ddot{i}_q; \quad \ddot{i}_q = i_{qr} - \dot{i}_{qr}^* \]

\( \ddot{i}_{dr}, \ddot{i}_{qr} \) are rotor current errors. \( k_p \) and \( k_i \) are proportional and integral gains of the rotor current controllers. where \( \dot{i}_{dr}^*, \dot{i}_{qr}^* \) are rotor current references in (d-q) reference frame;

\( \Psi^* \) is stator flux reference; \( x_d, x_q \) are integral components of current controllers. After computation the values of \( k_p = 500 \) and \( k_i = 62040. \)
The stator side active and reactive powers are given by:

\[ P_s = -\frac{3}{2} U_m i_{ds} \]  \hspace{1cm} (11) \\
\[ Q_s = \frac{3}{2} U_m i_{qs} \]  \hspace{1cm} (12)

The study, that we present, consists in using a motor where the rotor is supplied through a converter. This latter is based on PWM control algorithm, [14], operating at 2 KHz switching frequency.

4. DESIGN OF THE NEURO-FUZZY CONTROLLER

Fig. 4 shows the block diagram of the neuro-fuzzy controller (NFC) system. The NFC controller is composed of an on-line learning algorithm with a neuro-fuzzy network. The neuro-fuzzy network is trained using an on-line learning algorithm.

The NFC has two inputs, the rotor current error \( e_{idr} \) and the derivative of rotor current error \( \dot{e}_{idr} \). The output is rotor voltage \( V_{dr} \).

For the NFC of rotor current \( I_{rq} \) is similar with \( I_{rd} \) controller.
4.1 Description of NFC:

For the NFC, a four layer NN as shown in fig. 5 is used. Layers I–IV represents the inputs of the network, the membership functions, the fuzzy rule base and the outputs of the network, respectively.

4.1.1 Layer I: input layer

Inputs and outputs of nodes in this layer are represented as:

\[
\text{net}_1^1 = e_{idr}(t), \quad y_1^1 = f_1^1(\text{net}_1^1) = e_{idr}(t)
\]

\[
\text{net}_2^1 = \text{\hat{e}}_{idr}(t), \quad y_2^1 = f_1^1(\text{net}_1^1) = \text{\hat{e}}_{idr}(t)
\]

where \(e_{idr}\) and \(\text{\hat{e}}_{idr}\) are inputs \(y_1^1\) and \(y_2^1\) are outputs of the input layer. In this layer, the weights are unity and fixed.

4.1.2 Layer II: Membership layer

In this layer, each node performs a fuzzy set and the Gaussian function is adopted as a membership function

\[
\text{net}_{1,j}^II = \frac{(x_{1,j}^II - m_{1,j}^II)^2}{\sigma_{1,j}^II}, \quad y_{1,j}^II = f_{1,j}^II(\text{net}_{1,j}^II) = \exp(\text{net}_{1,j}^II)
\]

\[
\text{net}_{2,k}^II = \frac{(x_{2,k}^II - m_{2,k}^II)^2}{\sigma_{2,k}^II}, \quad y_{2,k}^II = f_{2,k}^II(\text{net}_{2,k}^II) = \exp(\text{net}_{2,k}^II)
\]

Where: \(m_{1,j}^II, m_{2,k}^II\) and \(\sigma_{1,j}^II, \sigma_{2,k}^II\) are, respectively, the mean and the standard deviation of the Gaussian function. There are \(j + k\) nodes in this layer.

4.1.3 Layer III: Rule layer

This layer includes the rule base used in the fuzzy logic control (FLC). Each node in this layer multiplies the input signals and outputs the result of product

\[
\text{net}_{jk}^III = (x_{1,j}^III \times x_{2,k}^III), \quad y_{jk}^III = f_{jk}^III(\text{net}_{jk}^III) = \text{net}_{jk}^III
\]

where the values of link weights between the membership layer and rule base layer are unity.

4.1.4 Layer IV: Output layer

This layer represents the inference and defuzzification used in the FLC. For defuzzification, the center of area method is used, therefore the following form can be obtain:
where $y_{jk}^\text{III}$ is the output of the rule layer; $a$ and $b$ are the numerator and the denominator of the function used in the center of area method, $w_{jk}^{IV}$ is the center of the output membership functions used in the FLC, respectively.

The aim of the learning algorithm is to adjust the weights of $w_{jk}^{IV}, m_{1,j}^{II}, m_{2,k}^{II}, \sigma_{1,j}^{II}, \sigma_{2,k}^{II}$. The on-line learning algorithm is a gradient descent search algorithm in the space of network parameters.

### 4.2 On-Line Learning Algorithm:

The error expression for the input of Layer IV

$$
\delta_0^{IV} = -\frac{\partial e_{idr}(t)}{\partial i} \frac{\partial i}{\partial y_0^{IV}} = \mu_5 e_{idr}(t)
$$

where $\mu_5$ is the learning-rate for $w_{jk}^{IV}$ and it can be shown in following equation.

Therefore, the changing of $w_{jk}^{IV}$ is written as [15]:

$$
\Delta w_{jk}^{IV} = -\frac{\partial e_{idr}(t)}{\partial i} \frac{\partial i}{\partial y_0^{IV}} \frac{\partial y_0^{IV}}{\partial w_{jk}^{IV}} = \frac{1}{b} \delta_0^{IV} y_{jk}^{III}
$$

Since the weights in the rule layer are unified, only the approximated error term needs to be calculated and propagated by the following equation:

$$
\delta_{jk}^{III} = -\frac{\partial e_{idr}(t)}{\partial i} \frac{\partial i}{\partial y_0^{IV}} \frac{\partial y_0^{IV}}{\partial y_{jk}^{III}} = \frac{1}{b} \delta_0^{IV} (w_{jk}^{IV} - y_0^{IV})
$$

The error received from Layer III is computed as:

$$
\delta_{jk}^{III} = \sum_k \left[ -\frac{\partial e_{idr}(t)}{\partial i} \frac{\partial i}{\partial y_0^{IV}} \frac{\partial y_0^{IV}}{\partial y_{jk}^{III}} \frac{\partial y_{jk}^{III}}{\partial y_{jk}} \right] = \sum_k \delta_{jk}^{III} y_{jk}^{III}
$$

$$
\delta_{jk}^{II} = \sum_j \left[ -\frac{\partial e_{idr}(t)}{\partial i} \frac{\partial i}{\partial y_0^{IV}} \frac{\partial y_0^{IV}}{\partial y_{jk}^{III}} \frac{\partial y_{jk}^{III}}{\partial y_{jk}} \right] = \sum_j \delta_{jk}^{III} y_{jk}^{III}
$$

The updated laws of $m_{1,j}^{II}, m_{2,k}^{II}$ and $\sigma_{1,j}^{II}, \sigma_{2,k}^{II}$ also can be obtained by the gradient decent search algorithm
where $\mu_4$, $\mu_3$, $\mu_2$ and $\mu_1$ are the learning-rate parameters of the mean and the standard deviation of the Gaussian functions.

### 5. Simulation Results

Fig. 6 shows the block diagram of the DFAM neuro-fuzzy control. The results, reported in figs. 7 to 16 were performed to investigate system behavior during torque tracking. The sequence of operation during this test is shown in fig. 7. The DFAM, already connected to the line grid, is required to track a trapezoidal torque reference, which starts at $t = 0.2\, \text{s}$ from zero initial value and reaches the rated value of $10\, \text{Nm}$ at $t = 0.3\, \text{s}$. Note that flux value, required to track torque trajectory with unity power factor at stator side is not a constant.

Waveforms of the rotor reference currents $i_{d^*r}$ and $i_{q^*r}$ are shown in Fig. 9. The rotor speed is shown in fig.10.

Fig.11 and Fig.12 shows the results of rotor currents responses using PI and neuro-fuzzy controller, respectively. The stator power $P_s$ follows the current $i_{q^*r}$ as shown in fig. 13.

This results in unity power factor on the grid as the stator reactive power $Q_s$ is zero. Rotor current errors are controlled at zero level. The reactive component of the stator current $i_{qs}$ is almost equal to zero during all the time. As result, the stator phase current, reported in fig. 14, has an opposite phase angle to the line voltage indicating that the stator power is
injected to the line grid.

In order to test the robustness of the two controllers, the value of the rotor resistance $R_r$ is augmented by 50%, $L_s$, $L_r$ and $M$ is decreased by 25% at $t = 0.5s$. Figs.17 to 20 shows the effect of parameters variations on the DFAM response for the two controllers respectively. This robustness test shows that in the case of a PI regulator, the time response is strongly altered whereas it remains unmodified when the NFC is used.

Figure 6: Block diagram of the DFAM neuro-fuzzy control

Figure 7: Torque reference

Figure 8: Flux reference

Figure 9: Rotor reference currents $I_{dr}^*$ and $I_{qr}^*$

Figure 10: Speed
Figure 11: Rotor currents responses with PI controllers

Figure 12: Rotor currents responses with NFC controllers

Figure 13: Active and reactive powers

Figure 14: Stator voltage and current

Figure 15: Stator currents responses

Figure 16: Stator flux $\psi_{ds}$ and $\psi_{qs}$

Figure 17: Rotor currents responses to parameters variations with PI controllers

Figure 18: Zoom of fig 17.
6. Conclusion

In this paper a hybrid intelligent control based torque tacking approach for doubly fed asynchronous motor (DFAM) drive have been proposed. Torque tracking control strategy has been achieved by adjusting rotor currents and using stator voltage vector oriented reference frame. The performances of neuro-fuzzy controller which is based on the torque tracking control algorithm has been investigated and compared to those obtained from the PI controller. Simulations results have shown that the NFC is more robust, efficient and more robust under parameters variations of the DFAM.

Appendix

Parameters of the DFAM:

\begin{align*}
&\text{Stator resistance } R_s = 4.7 \, \Omega, \quad \text{Rotor resistance } R_r = 5.3 \, \Omega, \\
&\text{Stator inductance } L_s = 0.161 \, \text{H}, \quad \text{Rotor inductance } L_r = 0.161 \, \text{H}, \\
&\text{Mutual inductance } M = 0.138 \, \text{H}, \quad \text{Rated current } I_r = 5.2 \, \text{A}, \\
&\text{Rated voltage} 220/380 \, \text{V}, \quad \text{Rated torque} 15 \, \text{Nm}, \quad p = 3, \\
&\text{Rated speed} 880 \, \text{rev/min}, \quad \text{Friction coefficient} f = 0.45.
\end{align*}

References


