Review of ANFIS-based control of induction motors

M.K. Masood*, Wooi Ping Hew and Nasrudin Abd. Rahim
Department of Electrical Engineering, University of Malaya, Kuala Lumpur, Malaysia

Abstract: This paper reviews the use of Adaptive Neuro-Fuzzy Inference Systems (ANFIS) for vector-controlled induction motor drives. While conventional schemes do not deal well with the highly nonlinear nature of motor control, fuzzy logic with its adjustability and neural networks with their adaptability have been shown to be excellent alternatives. ANFIS combines the advantages of fuzzy logic and neural networks and yields excellent results when used at various stages of the motor control process. The most prominent use of ANFIS with motor drives has been for parameter estimation, speed control and torque and flux control. The merits and demerits of these methods are examined. This paper is intended to serve as a reference for researchers considering the use of ANFIS for the control of motor drives.

Keywords: ANFIS, induction motor, flux, parameter, torque

1. Introduction

Motor drives, which consist essentially of an electric motor driving a mechanical load, find widespread application in industry. Traditionally, DC motors have been used for variable speed drives due to their simple design, easy operation and excellent performance. However, the need for a cheaper, low maintenance alternative lead to the use of AC motors. Unlike DC motors, the stationary and rotating parts of AC motors are not connected. They are therefore brushless and practically maintenance-free. Also, they can work in volatile environments without producing sparks or corroding. This, combined with their low cost, is why AC motors constitute 90% of all industrial motors [1]. They are used in industrial machinery, fans, blowers, vacuum cleaners, air conditioners and a plethora of other applications.

Vector control came to the fore as a method promising the excellent performance of a DC motor while using the relatively inexpensive AC motors. Field Oriented Control (FOC) used stator currents to control the torque of the motors. With the introduction of Direct Torque Control (DTC), however, improved response with reduced complexity was made possible. Combination of FOC and DTC with artificial intelligence (AI) opened an immensely promising avenue for motor control. Fuzzy logic, with its adjustable membership functions, provided a way to incorporate human expert knowledge in the control process. Neural networks offered the advantage of a training mechanism, a trait that would prove very useful in confronting the nonlinearity that besets modern motor control methods. The culmination of this trend was the combination of fuzzy logic and neural networks into neuro-fuzzy controllers, the most popular configuration of which uses the Adaptive Neuro-Fuzzy Inference System (ANFIS). This paper, besides a brief review of the use of Fuzzy Logic and Neural Network, examines the use of ANFIS for different aspects of vector controlled induction motor drives. Specifically, the use of ANFIS...
for parameter estimation, speed control and torque and flux control is reviewed. The advantages of ANFIS use for different stages of the motor control process are laid out.

2. Principles of vector control

Induction motor drives require frequency variation to vary the rotor speed. Also, at low frequencies, the motor impedance drops and the current shoots up. To reduce the complexity of the control algorithms. However, due to the fact that the control is nonlinear and multivariable, significant effort had to be directed to reducing the complexity of the control algorithms.

The complex space vector description is a commonly adopted method of depicting the variables of the induction motor in per-unit form [3–8]:

$$V_{sf} = R_s I_{sf} + T_N I_{sf} + j\omega_L I_{sf}$$

$$0 = R_s I_{sf} + T_N I_{sf} + j(\sigma f L_{Ls} I_{sf} - \omega_m)$$

$$\psi_{sf} = L_s I_{sf} + L_m I_{rf}$$

$$\frac{d\alpha_m}{dt} = \frac{1}{L_m} [\text{Im}(\psi_{sf}^* I_{sf}) - m_L]$$

Where $V_s$ is stator voltage, $I_s$ is stator current, $\psi_{sf}$ is stator flux linkage, $I_{sf}$ is rotor current, $\alpha_m$ is the mechanical angular speed, $m_L$ is the load torque, $L_m$ is the magnetizing inductance, $T_N$ is $1/2\pi f$ where $f$ is nominal frequency and $T_m$ is the mechanical time constant.

There are two fundamental approaches to controlling an induction motor: scalar and vector control. In scalar control, only the magnitude and frequency of voltages, currents and flux linkages is controlled. While it gives satisfactory performance with low-cost and low-performance drives [9], the speed and torque response for higher speed drives is poor because the stator flux and torque are not directly controlled. In vector control, both the magnitude and phase of the motor variables is controlled. In space vector notation, this translates to the control of both the magnitude and position of the space vectors of voltage, current and flux linkage. While scalar control is restricted to the steady state, vector control provides the correct orientation of space vectors both in the steady and transient state [10].

Vector control became prominent with the proposition of Field-oriented control (FOC) by Hasse [11] and Blaschke [12]. FOC was an attempt to mimic the decoupled control of the field and torque that was possible in separately-excited DC machines. The rotor winding in induction motors is not physically connected to the stator and the rotor current is induced, not directly supplied by an external source. Due to this, independent control of torque and the magnetic field (and thus the flux) is not straightforward in induction motors [13,15]. FOC involves the transformation of the three phase stator currents into an asynchronously rotating two axis d-q-reference frame. The stator current is the torque control quantity. If the rotor flux amplitude is kept constant, the following equation depicts how torque is controlled by the stator current, $I_{sq}$:

$$T = \frac{3}{2} \frac{L_m}{L_s} |\psi_r||\psi_s| \sin \delta_{sr}$$

The current-controlled inverter is well suited to implementing FOC [16]. In 1985, Takahashi and Noguchi [17] presented the strategy of Direct Torque Control. They proposed the simultaneous, not merely independent, control of torque and flux. The need for current loops was eliminated and torque and flux were controlled in the manner of a closed loop system [18–23]. DTC requires the knowledge of stator resistance only and thus greatly diminishes sensitivity to parameter variations. The need for a speed sensor does not arise. Also, the need for coordinate transformation between the stationary frame and synchronous frame is removed [24]. The expression for torque that forms the basis of DTC is [25]:

$$\sigma = 1 - \frac{L_s^2}{L_{Ls} L_{Lr}} \psi_r$$

Where $N_p$ is the pole-pair number, $\delta_{sr}$ is the spatial angle between stator and rotor fluxes and $\sigma$ is as in (8).

By keeping the stator flux constant, a fast torque response can be obtained by changing the angle $\delta_{sr}$ quickly.
A voltage source inverter is well suited to the implementation of DTC-based drives [26, 89]. A six pulse voltage source inverter consists of six non-zero active voltage switching space vectors and two zero vectors, giving a six sector formation as shown in Fig. 1. Consider the relationship between the voltage vector and stator flux:

\[ V_T = \frac{d\psi_s}{dt} \]  

(9)

Depending on the magnitude and position of the voltage vector applied, the stator flux vector moves with a particular speed in the direction of the voltage vector. A zero vector makes the flux vector remain stationary. Thus, the stator flux can be controlled with the application of suitable voltage vectors in each sampling period.

Conventionally, DTC has used hysteresis comparators to select voltage vectors from a switching table. In every sampling period, the actual and reference torque and flux values are compared. The errors are fed to a two-level hysteresis comparator. Depending on whether the errors are positive, negative or zero, the hysteresis comparators output 1, 0 or \(-1\). This information, along with the position of the stator flux, is used to select the appropriate voltage vector from the switching table. The block diagram of this scheme is shown in Fig. 2.

The conventional scheme, while providing fast torque response, has a few drawbacks. With only six active voltage vectors, the torque and flux errors are non-zero for a large portion of the control process, leading to torque and flux ripples [16]. The problem is compounded by the variation of switching frequency. To remedy these issues, Space Vector Modulation was proposed to synthesize the voltage vectors, as illustrated in [27, 28]. This method allowed for a constant switching frequency and a significant reduction in torque ripple.

### 3. Intelligent control

Fuzzy logic provided a viable alternative to model nonlinear relationships that posed difficulties in vector control. In [29], a fuzzy logic controller (FLC) replaced the hysteresis comparators for a direct self-controlled induction motor drive. Using errors between actual and estimated torque and flux values, the FLC determined the switching states of the inverter for each sampling period. Not only was the response faster than that of conventional DTC, the flux regulation was vastly improved as well. The FLC has been extensively used with switching tables. In [30], a FLC determined the

![Fig. 2. Conventional DTC scheme block diagram.](image-url)
duration within one sampling period for which a torque increasing vector selected from the switching table was applied. For the remaining duration, a torque decreasing vector was applied. As a result, the average value of the voltage vector could be from a wider range than the conventional seven voltage vectors. In [31–33], two FLCs were employed to calculate the optimum duty cycle per sampling period within which to apply the voltage vector. If in a particular sampling period, the inverter state was to be changed, one fuzzy logic controller selected the optimum duty cycle. If, in subsequent sampling periods, the inverter state was to remain the same, a second FLC selected an increment to the duty cycle. Only one FLC worked per sampling period and thus, the reduction in torque ripple came without a significant increase in computational burden compared to conventional DTC.

The use of FLC to regulate torque in [34] resulted in excellent torque tracking. To reduce the steady state error observed with FLC use, a PI-Fuzzy scheme was proposed in [35]. In addition to the reduction of the steady state error, this method provided a fast response and low overshoot. Another variant of a PI-fuzzy scheme was attempted in [36]. The torque component of the voltage vector was provided by an FLC, while a PI controller gave the flux component. With the flux estimated using PI flux magnitude control, the induction motor drive was made insensitive to DC drift. The issue of DC drift was addressed in [37] as well, where the error between the stator flux estimated by integrating back emf and low-pass filtered flux was input to an FLC. The absence of a complex observer or estimator reduced the computational burden. Also, the estimation was more accurate than methods used in works such as [38, 39], where the FLC used stator current error as the input. A noteworthy application was the placing of a PID-like fuzzy controller with another fuzzy controller for speed control in a vector controlled drive. In [40], a fuzzy adaptive gains for speed control of a FOC drive. While fuzzy logic’s ability to mimic human experts is tremendously useful, it lacks the adaptive capability that would eliminate the need for extensive trial and error. Neural networks, with their learning capability, have thus gained widespread use in the control of plants that vary with time or where the model of the system is partially known. Extensive research has therefore been conducted with the objective of finding the most suitable ANN training algorithms for vector control. The use of neural networks has been examined for two level inverters [45], three level inverters [46–48] and even a five level inverter [49]. ANNS have proven useful at various stages of control. In [50], an ANN was used to estimate the feedback signals in an induction motor drive. In [51], using parallel recursive prediction error and backpropagation training algorithms, ANN was used to tune the stator resistance for DTC. Besides demonstrating that the ANN could deal with large resistance variations and still estimate accurately, it was also proved that more neurons in the hidden layer provided a better approximation.

ANN use in speed control has been especially varied, as is evident in [52–56]. In [57], an ANN was used for modeling. Using the Maximum Likelihood (ML) estimation method and free acceleration response data, nonlinear models of the induction machine were created. In [58], a neural network was used as a state selector for a direct torque controlled induction motor drive. The errors between actual and estimated torque and flux were fed to a comparator. The comparator outputs, along with the stator flux angle, were made the inputs of a neural network, which output the switching states of the inverter. Using a reference switching state vector, an error vector was generated which was used to tune the weights of the ANN. Four training algorithms were compared: backpropagation, extended Kalman filter [90], adaptive neuron model and parallel recursive prediction error. While this work did present a comparative analysis of the algorithms, it did not show neural networks to offer better results than conventional DTC. This issue occurred in [59]. An ANN trained with the Levenberg Marquardt method and used along with a modified switching table produced torque
responses from the motor not significantly different from conventional DTC. However, the ANN controller was more stable and had the advantage of adaptability. In [60], the pattern recognition capability of a General Mapping Regressor, a type of neural network, was used to replace the hysteresis comparators in conventional DTC scheme. In addition to drastic reduction in torque ripple, response faster than even fuzzy logic based DTC was achieved. A recent development was the use of wavelet networks [61]. In field oriented control, it is often difficult to continuously modify the neural network offline to improve performance. To overcome this, a wavelet function was introduced in neural networks. A wavelet network allows for online parameter tuning. This network was used for stator estimation identification and shown to be more accurate than conventional neural networks.

With all their advantages, neural networks are not a preferred solution if the training data is insufficient to cover all operating modes [62]. To overcome their respective disadvantages, fuzzy logic and neural networks have been combined into neuro-fuzzy controllers. In so doing, the fuzzy logic ability to take account of human expert knowledge is paired with the learning ability of neural networks. The Adaptive Neuro-Fuzzy Inference System (ANFIS) is the most popular neuro-fuzzy configuration. Its structural flexibility, adaptability and simple mathematical representations have made it an excellent choice of controller in motor control.

4. Function of the ANFIS

An ANFIS makes an intelligent choice of the parameters of the membership functions (the antecedent parameters) and the coefficients of the output equation (the consequent parameters), thus eliminating a great deal of trial and error that normally plagues fuzzy control. The basic ANFIS architecture is shown in Fig. 3. A square node represents an adaptive node while a circle node has no parameters. The functions of each layer are described below:

Layer 1: This layer contains square nodes which receive the input and produce the degree of membership to each linguistic variable, which are represented by membership functions that are usually triangular or bell-shaped. For example, the membership value of an input \( x \) to a linguistic variable \( A_i \) (represented by a bell-shaped function) of node \( i \), is given by:

\[
O_{1i} = \mu_{A_i}(x) = \frac{1}{1 + \left( \frac{x - c_i}{a_i} \right)^2 b_i}
\]  

(10)

The parameters \( A_i, b_i \) and \( c_i \) are the premise parameters.

Layer 2: The nodes in this layer provide the firing strength of each rule. A preferred choice of doing so is by multiplication [63]. For example,

\[
O_{2i}^2 = w_i = \mu_{A_i}(x) \mu_{B_i}(y)
\]  

(11)

Layer 3: The firing strengths are normalized in this layer. The output of each node in this layer is

\[
O_{3i}^3 = \frac{w_i}{w_i + w_2}
\]  

(12)

Layer 4: The output of each node in this layer is:

\[
O_{4i}^4 = \pi_i f_i = \pi_i(p_i x + q_i y + r_i)
\]  

(13)

Where \( p_i, q_i \) and \( r_i \) are called consequent parameters.

Fig. 3. Basic ANFIS architecture.
Layer 5: All incoming signals are summed in this layer to obtain the crisp output.

\[ O^i = \sum \omega_i f_i = \sum \omega_i w_i f_i \]  \hspace{1cm} (14)

Training the ANFIS controller essentially involves tuning the premise and consequent parameters. The hybrid learning algorithm [64], which combines gradient descent and the least squares estimate (LSE), is a popular method due to its rapid convergence.

In the forward pass of the hybrid learning algorithm, premise parameters remain fixed and the consequent parameters are upgraded by the least squares estimate. In the backward pass, consequent parameters remain fixed and the premise parameters are updated by gradient descent.

5. ANFIS estimators

A good model of the induction motor is one that incorporates the variation of parameters over a wide range of operating conditions. This section examines the use of ANFIS to accomplish this.

To derive a dynamic model of the induction machine, the three-phase voltages and currents of the stator and rotor (denoted by \( s \) and \( r \) subscripts, respectively) are transformed into the more workable two-axis, \( d-q \) coordinate system. The voltage equations are:

\[ V_{ds} = R_r I_{ds} + \frac{d}{dt} \lambda_{ds} \]  \hspace{1cm} (15)

\[ V_{qs} = R_r I_{qs} + \frac{d}{dt} \lambda_{qs} \]  \hspace{1cm} (16)

\[ 0 = R_i I_{dq} + \frac{d}{dt} (\omega_s - \omega_r) \lambda_{dq} \]  \hspace{1cm} (17)

\[ 0 = R_i I_{dq} + \frac{d}{dt} (\omega_s - \omega_r) \lambda_{dq} \]  \hspace{1cm} (18)

Where \( \lambda \) is the flux linkage and \( \omega \) is the angular velocity.

The flux linkages depend on the currents and inductances. When the machine becomes saturated, the mutual and leakage inductances become nonlinear and depend on the currents that flow through the inductances. The stator and rotor flux linkages also depend nonlinearly on all currents. To calculate the flux derivatives in the voltage equations, we need to apply the chain rule. The resulting derivatives are called incremental inductances [65]. For example, the \( q \)-axis stator incremental inductance is given in (19).

\[ \frac{\partial \lambda_{qs}}{\partial t} = \frac{\partial \lambda_{qs}}{\partial I_{qs}} \frac{d I_{qs}}{dt} + \frac{\partial \lambda_{qs}}{\partial I_{qr}} \frac{d I_{qr}}{dt} + \frac{\partial \lambda_{qs}}{\partial I_{ds}} \frac{d I_{ds}}{dt} + \frac{\partial \lambda_{qs}}{\partial I_{dq}} \frac{d I_{dq}}{dt} \]  \hspace{1cm} (19)

In a similar manner, the \( d \)-axis stator incremental inductance and the rotor incremental inductances can be calculated. The dynamic machine model is obtained by substituting these equations in the voltage equations. The resulting model would contain 21 parameters, each varying with the operating conditions. In [66], a simplified model which ignores the cross-coupling effects between the \( d \) and \( q \) axis, and thus assumes the incremental inductances to be influenced only by their own axis currents, was proposed. The simplified model is given in (20). The model reduces the number of parameters to be estimated to 11. These are:

\[ R_s, R_r, L_{ds}, L_{qs}, T_{dqs}, T_{dqr}, T_{dld}, T_{qlq} \]

In [66], parameter estimation through neural networks was tested. The main shortcoming of this approach was the absence of intuitive modeling in neural networks. As a remedy, an ANFIS estimator was used in [67, 68]. Each parameter to be estimated requires its own ANFIS model, since the ANFIS can have only one crisp output.

\[ \begin{bmatrix} \frac{d V_{ds}}{dt} \\ \frac{d V_{qs}}{dt} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} R_s & \omega_s (L_{dq} + L_{ds}) & 0 & \omega_s L_{ds} \\ -(\omega_s - \omega_r) L_{ds} & -\omega_r L_{ds} & 0 & 0 \\ -(\omega_s - \omega_r) L_{ds} & 0 & -\omega_s (L_{dq} + L_{ds}) & 0 \\ -(\omega_s - \omega_r) L_{ds} & 0 & 0 & -\omega_s (L_{dq} + L_{ds}) \end{bmatrix} \begin{bmatrix} I_{ds} \\ I_{qs} \\ I_{dqr} \\ I_{dld} \end{bmatrix} + \begin{bmatrix} T_{dqs} + T_{dss} \\ T_{dqs} \\ T_{qlq} + T_{qss} \end{bmatrix} \begin{bmatrix} \omega_s \omega_r \omega_s \omega_r \end{bmatrix} \]  \hspace{1cm} (20)
Data for modeling must contain transient responses from various operating conditions because the parameters depend nonlinearly on operating conditions. That is why, free acceleration data is used to create the models [66]. The measurements are obtained by applying three phase power to the motor while it is at standstill without load. When the motor starts, the stator and rotor are saturated with current. As the rotor accelerates, the stator voltages, stator currents and rotor angular velocities are recorded. Hall effect sensors can be used for the voltage and current measurements. An encoder can get the rotor angular position. Angular velocity can be calculated using a DSP.

Now, the slip of the motor is given by:

\[ s = \frac{\omega_{st}}{\omega_{syn}} - 1 \]  

(21)

Where \( \omega_{st} \) is the velocity of the stator field, also called the synchronous velocity. As the rotor speed varies from zero to just below the synchronous speed, the slip varies from one to zero. Therefore, a thorough representation of parameter variation with rotor speed can be obtained from this test procedure.

The training data set is prepared by selecting a short time duration (In [66], this duration is 0.05 seconds) within which the parameter variation is infinitesimal. Average values of the stator voltages and rotor angular velocity (i.e. the ANFIS inputs) and the eleven parameters to be estimated (the ANFIS outputs) are measured for each duration. The resulting list of parameters for successive durations constitutes the training data to be used to create the ANFIS model.

To test the model, varying values of stator current and rotor angular velocity are provided to the model and the output of the ANFIS is examined. This is compared to actual measured values of the parameters. In [67], errors
as low as 0.000341% were obtained with the ANFIS estimator.

6. ANFIS use in speed control

FOC methods have two primary disadvantages: sensitivity to motor parametric variations and flux errors at low speeds [68]. These are problems PID controllers do not deal with well, leading to deterioration in performance. This is where artificial intelligent controllers have proven to be excellent alternatives to speed control. The ANFIS controller can be trained to accommodate a wide range of operating conditions.

In [69, 70], the conventional PI speed controller was replaced by an ANFIS controller. The controller used speed error and the rate of change of speed error as inputs. The inputs were normalized before being fed to the ANFIS controller, according to equations (22) and (23).

\[
\varepsilon_\omega = \frac{\omega^* - \omega}{\omega^*} \times 100\% \quad (22)
\]

\[
\Delta \varepsilon_\omega = \frac{\varepsilon_\omega(n) - \varepsilon_\omega(n-1)}{T} \times 100\% \quad (23)
\]

Where \( \varepsilon_\omega \) is the normalized speed error and \( \Delta \varepsilon_\omega \) is the rate of change of speed error. Figure 4 shows the block diagram of the speed controller.

To train the controller, the hybrid learning algorithm was used. Excellent current and speed response was achieved. However, while works such as [69, 70] used the torque producing current component \( i_{qs} \) as the speed controller output, it is very difficult to practically generate current values to be used as offline training data for the ANFIS controller. To address this problem, an online tuning method was used in [71] to update the ANFIS weights based on the error between a reference model of motor speed and actual motor speed accelerations. The motor speed acceleration is calculated as the slope of motor speed.

Taking the specific requirements of the IM drive into consideration, the motor speed slope was expressed as:

\[
y = (1 - \exp\left(\frac{-(\omega)^2}{2 \times 0.01}\right)) \times 1000 \times \text{sign}(\omega) \quad (24)
\]

Where \( y \) is the reference motor speed slope, \( \left( \frac{d\omega}{dt} \right)^2 \)

The ANFIS weights were tuned so as to make the actual speed slope follow the reference. The objective function to be minimized was:

\[
E = \frac{1}{2} \left( y - \frac{d\omega}{dt} \right)^2 \quad (25)
\]

This model was implemented using a DSP. The transient response showed no overshoot and a small settling time. Also, limiting the input membership functions to three made for low computational burden. The controller was tested with sudden load disturbance, to which it showed a negligible amount of speed deviation, as is shown in Fig. 5. Additionally, the ANFIS controller forced the speed to follow the reference far more closely than a PI controller that was tuned to have minimum overshoot and settling time. This is shown in Fig. 6.

The use of two controller inputs, as has been done in [72–76], leads to a large number of membership functions and rules. An improved scheme, proposed in [77] and experimentally verified in [78], utilized only the speed error as input to the ANFIS controller. The ANFIS used one input with three membership functions. The normalized speed error was input to the
Fig. 7. Torque and flux control using ANFIS.

Fig. 8. Architecture of ANFIS controller.
The command current was calculated using (40).

\[ i^\ast_q(n) = \frac{2}{3} \frac{L_r}{P} \omega_r \lambda_{dr} \]

Where \( T^\ast_e \) is the command torque, \( P \) is the number of poles of the motor, \( L_r \) is the rotor inductance, \( L_m \) is the magnetizing inductance and \( \lambda^\ast_{dr} \) is the rotor’s d-axis flux linkage.

Training of the controller was done using an unsupervised on-line self tuning method. The objective function that the algorithm sought to minimize is given in (27).

\[ E = \frac{1}{2} (\omega^* - \omega)^2 \]

Comparison of the speed response of this one input scheme with the conventional two-input one showed no significant decrease in performance. The increased simplicity of the design was thus shown to be a satisfactory tradeoff.

On-line training of the ANFIS controller was also examined in [79]. Here again, the objective function was as in (41). The weights of the controller were adjusted using (42).

\[ w_r(k+1) = w_r(k) - \gamma \frac{\partial E(k)}{\partial w_r(k)} \]

Where \( \gamma \) is the learning rate, \( E \) is the objective function and \( w_r \) is the weight of the \( r \)-th rule. However, this algorithm converges slowly, leading to an unsatisfactory learning rate. Therefore, a modified algorithm suggested in [80] was used. This is based on local gradient PD control:

\[ w_r(k+1) = w_r(k) + O_{gb}^r(k) \Delta x_r(k) \]

Where \( O_{gb}^r \) is the firing strength of \( r \)-th rule and \( \Delta x_r(k) \) is the tracking error. The parameters \( k_p \) and \( k_d \) were determined using a genetic algorithm to rapidly eliminate the tracking error. The response of the drive was tested by a slow reverse of the reference speed. Barr- ring small speed oscillations, the controller eliminated the speed tracking error and the drive remained stable under full load torque.

7. ANFIS use in torque and flux control

The use of ANFIS to directly generate voltage vectors according to toque and flux values was first proposed in [81]. The block diagram of this scheme is shown in Fig. 7.

Reference torque and flux values are compared with actual estimated values. The errors are fed to the ANFIS controller, which outputs the reference voltage vector amplitude and phase.

The architecture of the ANFIS controller is shown in Fig. 8. Layer 5 outputs the voltage vector amplitude and uses an increment angle (described later in this section) to calculate the phase of the voltage vector.

To train the controller automatically, the hybrid learning algorithm was used. The consequent parameters that define the ANFIS output were tuned using least-square estimation and the antecedent parameters were trained using the backpropagation method. Reference values of voltage vectors, obtained from a PI controller, combined with torque and flux error values
were used to allow the controller to learn the pattern. The Kalman filter is also an appropriate option of learning algorithm, as has been demonstrated in [82].

For low-speed motor operation, a manual tuning procedure was suggested in [83]. The antecedent parameters defining the membership functions are chosen on the basis of calculation time. While an extension of membership functions only marginally improves the performance, the calculation time is considerably increased. The width of the membership functions is determined by the product of the weights and the error inputs. By locating the minimum value of this product, effective tuning can be accomplished. The tuning surface, shown in Fig. 9, is constructed by comparing the effects of various weight values with the torque and flux errors, separately. In [83], it is suggested to first identify the optimal weights for least flux error and then for the torque error. The vector adder in Fig. 8 outputs the amplitude and angle of the voltage vector. For each sampling period, the average value of all the voltage vectors obtained from the ANFIS output is calculated.

To calculate the increment to the angle of the voltage vector, a selection table is constructed. Consider the equations for stator flux and torque for a stator flux coordinate system [84], as shown in (44) and (45):

\[ \lambda_s = \frac{1}{T_s} \int_{0}^{t} v_s \cos \phi \, dt + \lambda_{s0} \]  

(30)

were used to allow the controller to learn the pattern. The Kalman filter is also an appropriate option of learning algorithm, as has been demonstrated in [82].

For low-speed motor operation, a manual tuning procedure was suggested in [83]. The antecedent parameters defining the membership functions are chosen on the basis of calculation time. While an extension of membership functions only marginally improves the performance, the calculation time is considerably increased. The width of the membership functions is determined by the product of the weights and the error inputs. By locating the minimum value of this product, effective tuning can be accomplished. The tuning surface, shown in Fig. 9, is constructed by comparing the effects of various weight values with the torque and flux errors, separately. In [83], it is suggested to first identify the optimal weights for least flux error and then for the torque error. The vector adder in Fig. 8 outputs the amplitude and angle of the voltage vector. For each sampling period, the average value of all the voltage vectors obtained from the ANFIS output is calculated.

To calculate the increment to the angle of the voltage vector, a selection table is constructed. Consider the equations for stator flux and torque for a stator flux coordinate system [84], as shown in (44) and (45):

\[ \lambda_s = \frac{1}{T_s} \int_{0}^{t} v_s \cos \phi \, dt + \lambda_{s0} \]  

(30)
\[ T = \lambda_s \left( \frac{v_s \sin \phi - \omega_s \lambda_s}{r_s} \right) \]  

(31)

Where \( v_s \) is the stator voltage and \( \phi \) is the angle increment \( \Delta \gamma \).

Consider the case when the flux error is zero and the torque error is either positive or negative. No change in flux is required and therefore the \( \cos \phi \) term in (44) should be made zero. Thus, \( \phi = \pm \pi/2 \) would be a prudent choice for the angle increment. This will make the torque value to become:

\[ T = \lambda_s \left( \frac{v_s}{r_s} \right) \]  

(32)

When both torque and flux errors are either positive or negative, an intermediate value is chosen. For example, when both torque and flux errors are positive, the angle increment is chosen to be \( \pi/4 \).
Table 1

<table>
<thead>
<tr>
<th>eλ</th>
<th>P</th>
<th>Z</th>
<th>N</th>
<th>P</th>
<th>Z</th>
<th>N</th>
<th>P</th>
<th>Z</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>π</td>
<td>0</td>
<td>π</td>
<td>2</td>
<td>π</td>
<td>2</td>
<td>π</td>
<td>2</td>
<td>π</td>
<td>2</td>
</tr>
</tbody>
</table>

In this manner, a selection table is constructed. Table 1 shows the table used in [84].

A number of advantages have been achieved using this control method. Firstly, while conventional DTC is limited to eight switching vectors, the ANFIS control combined with space vector modulation enables any number of voltages to be synthesized. This shows in the reduction of torque ripple, as shown in Fig. 10. This also allows for a smooth stator flux trajectory, as shown in Fig. 11. Besides the scheme described above, two variants have been documented:

1. Phase correction as ANFIS output:
   In [85], the ANFIS controller was trained to output only the angle increment. Figure 12 shows the block diagram of this control scheme.

   The ANFIS controller’s output is given by:
   \[ A_\theta = \sum_{i=1}^{n} \frac{W_i C_i}{\sum_{i=1}^{n} W_i} \]  (33)

   The amplitude calculator of Fig. 13 utilizes Equations (34) and (35).
   \[ V_{a0} = \frac{2}{3} \cos \left( \frac{A_\theta + \frac{\pi}{3}}{3} \right) \]  (34)
   \[ V_{a0} = \frac{2}{3} \sin \left( \frac{A_\theta + \frac{\pi}{3}}{3} \right) \]  (35)

2. Two ANFIS controllers for three-level inverters:
   Three-level voltage source inverters consist of four switches in each inverter leg. They allow for higher voltage ratings compared to two-level inverters and produce higher quality currents [1]. Three switching states are used per inverter leg and therefore a total of 27 states are available [86]. Each state corresponds to a switching vector, of which three are zero-state vectors. The voltage vectors are classified into four groups [87]: Zero Voltage Vectors (V0, V1, V2), Low Voltage Vectors (V3, V4, V5, V6, V7, V10, V11, V12, V13), Intermediate Voltage Vectors (V14, V15, V16, V17, V18, V19, V20, V21), and High Voltage Vectors (V22, V23, V24, V25, V26).

   In [88], the use of two separate ANFIS controllers was proposed. Both controllers use the torque and flux errors, but one controller outputs the low voltage vectors, while the other covers the intermediate and high voltage vectors. This scheme is depicted in Fig. 13.

   This scheme reduced harmonic distortion in currents, in comparison to conventional DTC using a two-level inverter. This is depicted in Fig. 14.

8. Conclusion

This paper has reviewed the use of ANFIS to control induction motor drives. The main features of ANFIS are structural flexibility, adaptability and simple mathematical representation. It has been used with both Field Oriented Controlled and Direct Torque Controlled drives at various stages of the control process. The most recurring use, though, has been for modeling, parameter estimation, and speed, torque and flux control. The control schemes used by researchers are...
examined, the methods explained and the advantages offered by ANFIS over conventional control methods are presented.

ANFIS has proven to be tremendously accurate in the modeling of motor parameters, enabling the incorporation of a wide range of operating conditions in the model. In speed control, ANFIS controllers have been shown to give excellent transient response with precise speed tracking. ANFIS-based torque and flux controllers significantly reduced torque ripple, generated smooth flux trajectories and reduced stator current distortion. This paper is intended to serve as a reference to the most recurring use of ANFIS in induction motor control. The application of ANFIS to remedy common control problems may give an insight as to how it may address problems of a similar nature. With the constant advancement of simulation software and hardware such as microprocessors and DSPs, intelligent control schemes such as ANFIS are expected to become the most effective control choices, not least in the control of variable frequency drives.

References


[34] H. Rehman, Fuzzy logic enhanced robust torque controlled induction motor drive system, IEE Proceedings on Control Theory and Applications 151(6) (2004), 754-762.


