ISSN-L: 2223-9553, ISSN: 2223-9944
Vol. 2, No. 1, January 2012

APPLICATION OF VECTOR CONTROL METHOD FOR DEVELOPING ANFIS OBSERVER AS SPEED SENSOR FOR INDUCTION MOTOR SPEED CONTROL IN ELECTRIC VEHICLE

Era Purwanto
Electronic Engineering Polytechnic,
Institute of Surabaya – ITS
INDONESIA
era@eepis-its.edu

Prof. Dr. Ir Soebagio
Sepuluh November Institute
of Tecnology, Surabaya
INDONESIA
soebagio@ee.its.ac.id

Prof. Dr. Ir. Mauridhi Herry
Sepuluh November Institute
of Tecnology, Surabaya
INDONESIA
hery@ee.its.ac.id

ABSTRACT

There is a growing interest in electric vehicles due to environmental concerns. Recent efforts are directed toward developing an improved propulsion system for electric vehicles applications. This paper is aimed at developing the system design philosophies of ANFIS observer as speed sensor .Now the induction motor is growing up to use as propulsion in the electric vehicles. The simple method to develop observer for detects the speed of Induction motor as propulsion in the electric vehicle is proposed in this paper, vector control method is one's of the method which developing in this paper. Direct field-oriented induction motor drive system need rotor flux observer and rotor angular speed identifier. ANFIS is used for identifying parameter dynamics and system variable estimation, linear either non-linear. ANFIS with back propagation learning algorithm has applied to estimate flux rotor and identify rotor angular speed of three-phase induction motor. The simulation result is found well for the speed up to 200 rpm, and no good result for the speed less than 200 rpm. There is a growing interest in electric vehicles due to environmental concerns. Recent efforts are directed toward developing an improved propulsion system for electric vehicles applications. This paper is aimed at developing the system design philosophies of ANFIS observer as speed sensor. Now the induction motor is growing up to use as propulsion in the electric vehicles. The simple method to develop observer for detects the speed of Induction motor as propulsion in the electric vehicles are proposed in this paper, direct field-oriented is one's of the method which developing in this paper. Direct field-oriented induction motor drive system need rotor flux observer and rotor angular speed identifier. ANFIS is used for identifying parameter dynamics and system variable estimation, linear either non-linear. ANFIS with back propagation learning algorithm has applied to estimate flux rotor and identify rotor angular speed of three-phase induction motor. The simulation result is found well for the speed up to 200 rpm and no good result for the speed less than 200 rpm.

Keywords: Observer, Vector control, speed sensor observer, ANFIS

INTRODUCTION

Automobile manufacturers are actively studying possibilities of electric vehicles for widespread practical use. However, there are a variety of problems that must be solved for electric vehicles , one's of the problem is how to measure the speed of induction motor as propulsion. The control system for induction motor has the problem for uncertainly parameters, for this case considered the controller is known structure and uses the values of the state variable. However, this causes a new problem, cores easily saturate when the primary currents flowing in the winding are increased, because the primary and secondary iron core sizes for the motors must be made smaller. This leads to difficulties in accurate and stable torque control of the motor in regions where the iron cores of the motor is saturated. But some state variables cannot be measured, such as rotor flux, to solving this problem, the observer to estimate the state variable that cannot be measured is needed. A flux observer based on ANFIS will be used to estimate the flux, for identifying motor induction angular speed. Automobile manufacturers are actively studying possibilities of electric vehicles for widespread practical use .

However, there is a variety of problems that must be solved for electric vehicles, one's of the problem is how to measure the speed of induction motor as propulsion. The control system for induction motor has the problem for uncertainly parameters, for this case considered the controller is known structure and uses the values of the state variable. However, this causes a new problem; cores easily saturate when the primary currents flowing in the winding are increased, because the primary and secondary iron core sizes for the motors must be made smaller. This leads to difficulties in accurate and stable torque control of the motor in regions where the iron cores of the motor is saturated. But some state variables cannot be measured, such as rotor flux, to solving this problem, the observer to estimate the state variable that cannot be measured is needed. A flux observer based on ANFIS will be used to estimate the flux, for identifying motor induction angular speed.

The state equations for induction motor are developed by using dq model, and it can be expressed as following equation in the stationary reference frame.

$$\frac{d}{dt}\begin{bmatrix} i_s \\ \lambda_r \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}\begin{bmatrix} i_s \\ \lambda_r \end{bmatrix} + \begin{bmatrix} B_1 \\ 0 \end{bmatrix} v_s \tag{1}$$

$$= \mathbf{Ax} + \mathbf{Bv}_s$$

$$\mathbf{i}_s = \mathbf{Cx} \tag{2}$$

Where

$$\mathbf{X}=\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}, \mathbf{B}=\begin{bmatrix} \mathbf{B}_1 \\ 0 \end{bmatrix}$$

$\mathbf{i}_s = [\mathbf{i}_{sd} \ \ \mathbf{i}_{sq}]^T$: Stator current

$\lambda_r = [\lambda_{rd} \ \ \lambda_{rq}]^T$: Rotor flux

$\mathbf{v}_s = [v_{sd} \ \ v_{sq}]^T$: Stator voltage

$\mathbf{A}_{11} = -\{R_s/(\sigma L_s) + (1-\sigma)/(\sigma\tau_r)\}\mathbf{I} = a_{r11}\mathbf{I}$

$\mathbf{A}_{12} = M/(\sigma L_s L_r)\{(1/\tau_r)\mathbf{I} - \omega_r \mathbf{I}\} = a_{r12}\mathbf{I} + a_{i12}\mathbf{J}$

$\mathbf{A}_{21} = (M/\tau_r)\mathbf{I} = a_{r21}\mathbf{I}$ $\quad$ $\mathbf{A}_{22} = -(1/\tau_r)\mathbf{I} + \omega_r\mathbf{J} = a_{r22}\mathbf{I} + a_{i22}\mathbf{J}$

$\mathbf{B}_1 = 1/(\tau L_s)\mathbf{I} = b_1\mathbf{I}$

$\mathbf{C} = [\mathbf{I} \ \ 0]$

$\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $\quad$ $\mathbf{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$

R_s, R_r : Stator and rotor resistance

L_s, L_r : Stator and rotor self inductance

M : Mutual inductance

σ : Leakage coefficient, $\sigma = 1 - M^2/L_s L_r$

τ_r : Rotor time constant, $\tau_r = L_r / R_r$

ω_r : Motor angular speed

THEORY AND SYSTEM DESIGN

A. Adaptive Networks of ANFIS System

An adaptive network is a multilayer feed forward network in which each node performs a particular function (node function) on incoming signals using a set of parameters specific to this node [3]. The form of node functions may vary from node to node, and the choice of each node function depends on the overall function, which the adaptive network is designed to implement. To reflect different capabilities, both circle and square nodes in an adaptive network is used. A square node (adaptive node) has modifiable parameters while a circle node (fixed node) has none. The parameter set of an adaptive network is the union of the parameter sets of each adaptive node. In order to achieve a desired input-output mapping, these parameters are updated according to given training data and a gradient-based update procedure described below. Suppose that a given adaptive network has L layers and layer l (l = 0,1,...L; l = 0 represents the input layer) has $N(l)$ node. Then the output and function of node i (i = 1,...,$N(l)$) in layer l can be represented as $x_{l,i}$ and $f_{l,i}$, as shown in the figure 1. Since the output of a node depends on the incoming signals and the parameter set of the node, we have the following general expression for the node function:

$$x_{l,i} = f_{l,i}\,(x_{l-1,1},\ldots,x_{l-i,N(l)},\ \alpha, \beta, \gamma,\ldots) \qquad (4)$$

α,β,γ are the parameters of this node.

Assuming that the given training data set has P entries, we can define an error measure for the pth (1< = p < = P) entry of the training data as the *sum of squared error* :

$$Ep = \sum_{k=1}^{N(l)} (d_k - x_{L,k})^2 \qquad (5)$$

where d_k : the kth component of the pth desired output vector, $x_{L,k}$: the kth component of the actual output vector produced by presenting the pth input vector to the network. For notational simplicity, we omit the subscript p for both d_k and $x_{L,k}$) Thus our task here is to minimize an overall error measure, which defined as

$$E = \sum_{p=1}^{P} E_p \qquad (6)$$

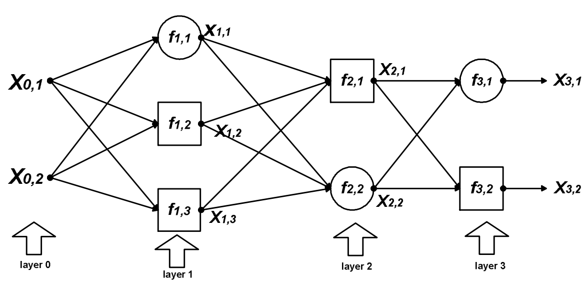

Figure 1. Feed forward adaptive network

The error signal $\varepsilon_{l,i}$ is defined as the derivative of the error measure E_p with respect to the output of node i in layer l,

$$\varepsilon_{l,i} = \frac{\partial^+ E_p}{\partial x_{l,i}} \qquad (7)$$

This expression was called the ordered derivative by Werbos. The error signal for the ith output node at layer L could be calculated directly

$$\varepsilon_{l,i} = \frac{\partial^+ E_p}{\partial x_{L,i}} = \frac{\partial E_p}{\partial x_{L,i}} \qquad (8)$$

For the internal node at the ith position of layer l, the error signal can be derived by the chain rule

$$\varepsilon_{l,i} = \frac{\partial^+ E_p}{\partial x_{l,i}} = \sum_{m=1}^{N(l+1)} \frac{\partial^+ E_p}{\partial x_{l+1,m}} \frac{\partial f_{l+1,m}}{\partial x_{l,i}} = \sum_{m=1}^{N(l+1)} \varepsilon_{l+1,m} \frac{\partial f_{l+1,m}}{\partial x_{l,i}} \qquad (9)$$

Where 0 <= l <= L-1

For any l and i (0 < = l < = L and 1 < = i < = $N(l)$) the error signal can be found by first applying Equation (8) once to get error signals at the output layer, and then applying Equation (9) iteratively until we reach the desired layer l. The underlying procedure is called *back propagation* since the error signals are obtained sequentially from the output layer back to the input layer.

The gradient vector is defined as the derivative of the error measure with respect of each parameter. If α is a parameter of the ith node at layer l,

$$\frac{\partial^+ E_p}{\partial \alpha} = \frac{\partial^+ E_p}{\partial x_{l,i}} \frac{\partial f_{l,i}}{\partial \alpha} = \varepsilon_{l,i} \frac{\partial f_{l,i}}{\partial \alpha} \qquad (10)$$

The derivative of the overall error measure E with respect to α

$$\frac{\partial^+ \mathrm{E}}{\partial \alpha} = \sum_{p=1}^{P} \frac{\partial^+ \mathrm{E}_p}{\partial \alpha} \qquad (11)$$

The update formula for the generic parameter α

$$\Delta\alpha = -\eta \frac{\partial^+ \mathrm{E}}{\partial \alpha} \qquad (12)$$

Where η is the learning rate, which can be further expressed as

$$\eta = \frac{\kappa}{\sqrt{\sum_{\alpha} \left(\frac{\partial E}{\partial \alpha} \right)^2}} \qquad (13)$$

where κ is the *step size,* the length of each transition along the gradient direction in the parameter space. Usually we can change the step size to vary the speed of convergence.

B. Adaptive Neuro-Fuzzy Inference Systems

Adaptive Neuro-Fuzzy Inference Systems (ANFIS) is a class of adaptive networks that are functionally equivalent to fuzzy inference systems. Structurally, the only limitation on the network configuration is that it should be feed forward type if we do not want to use the more complex asynchronously operated model. The fuzzy inference system under consideration has two inputs x and y, and one output z is assumed. For a first order Sugeno fuzzy model, a common rule set with two fuzzy if-then rules is expressed such as the following model:

1 : If x is A_1 and y is B_1, then $f_1 = p_1x+q_1y+r_1$
2 : If x is A_2 and y is B_2, then $f_2 = p_2x+q_2y+r_2$

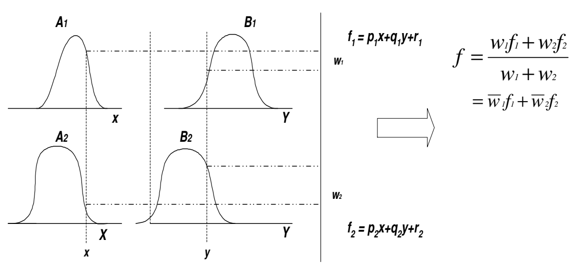

Figure 2. A two input first order Sugeno fuzzy model with two rules

Figure 2 illustrates the reasoning mechanism for this Sugeno model, the corresponding equivalent ANFIS architecture is as shown in figure 3 and the output of the ith node in layer l as $O_{l,i}$

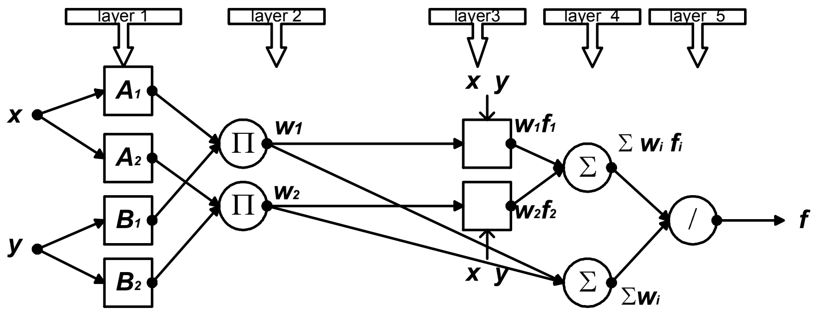

Figure 3. Equivalent ANFIS architecture

Layer 1: Every node *i* in this layer is an adaptive node with a node function

$O_{1,\iota} = \bigcirc_{A\iota}(\xi)$ $\phi_{o\rho}$ $\iota = 1, 2, o\rho$

$\boldsymbol{O_{1,\iota} = \bigcirc_{B\iota-2}(\psi)}$, **for $i = 3, 4$**

where x (or y) is the input to the node i and A_i (or B_{i-2}) is a linguistic variable. In other words, $O_{1,i}$ is the membership grade of a fuzzy set A ($=A_1$, A_2, B_1, B_2) and it specifies the degree to which the given input x (or y) satisfies the quantifier A. Here the membership function for A can be any appropriate parameterized membership function, such as Gaussian function:

$$\mu_A(x) = e^{-\frac{1}{2}\left(\frac{x-b_i}{a_i}\right)^2} \tag{14}$$

where $\{a_i, b_i\}$ is the parameter set. As the values of there parameters change, the Gaussian-shaped function varies accordingly. Parameters in this layer are preferred to as premise parameters.

Layer 2 : Every node on this layer is a fixed node ,whose output is the product of all the incoming signals

$$O_{2,i} = w_i = \bigcirc_{Ai}(x)\ \bigcirc_{Bi}(y) \quad i = 1, 2 \tag{15}$$

Each node output represents the firing strength of a rule.

Layer 3 : Every node in this layer is a fixed node labeled N. The *i*th node calculates the ratio of the *i*th rule's firing strength to the sum of all rule's firing strengths

$$O_{3,i} = w_i f_i = w_i (p_i x + q_i y + r_i) \tag{16}$$

$\{p_i, q_i, r_i\}$ is the parameter set of this node, known as consequent parameters.

Layer 4 : The first node in this layer computes the output of layer 3, the second node below computes the normalized firing strength of layer 2.

Layer 5: The single node in this layer is a fixed node, which computes the overall output as the division of the incoming signals:

$$overall\ output = O_{5,1} = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{17}$$

DESIGN AND EXPERIMENT

The observer was designed to estimate some induction motor outputs that can not be directly measured. Observer performance was measured by the difference between induction motor output and observer. More less the difference then can be said that observer already has high performance.

Training data taken while the overall system simulation running without the observer. The induction motor driven by PWM Inverter and proportional speed controller with gain=1, at the set point 250 rad/s. Simulation run for 5000 iteration, so we can get 5000 pairs data. This data used for find the fuzzy inference system and the training.

A. Fuzzy Inference System Design

The Gaussian function is used in this simulation for its smoothness, instead of triangular shape. We do not have domain knowledge from a human operator's point a view about the observer, then the number of fuzzy if-then rules has to be decided by trial and error. The flux observer and angular speed identifier is implemented as ANFIS with four inputs on each. With subtractive clustering method, we have eight membership function for each input, and eight rules for each observer and identifier. Though the number of fuzzy rules can be more than eight, the simulation indicates eight rules are enough for identification.

B. Training

To speed up the convergence, we follow a strict gradient descent in the sense that each transition of the parameters will lead to a smaller error measure. If the error measure increases after parameter update, we back up to the original point in the parameter space and decrease the current step size by half. This process is repeated until the weight update leads to a smaller error measure. However, this step size update rule tends to use a small step size if the error measure surface encountered in the first few updates is not smooth. Therefore we multiply the step size by 4 after observing three consecutive transitions without any backup actions. The initial step size in the simulation is 2 and the learning process stops whenever the number of epoch has given reached. As shown in the figure 4 is off-line configuration of the rotor flux observer, figure 5 for rotor angular speed identifier. On-line configuration at figure 6 shows us that estimated flux from flux observer make the input for rotor angular speed identifier together with stator current.

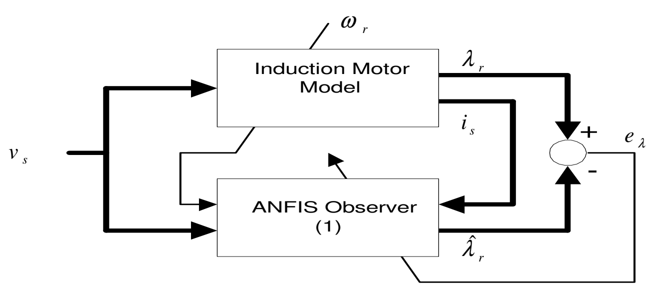

Figure 4. Training scheme of observer

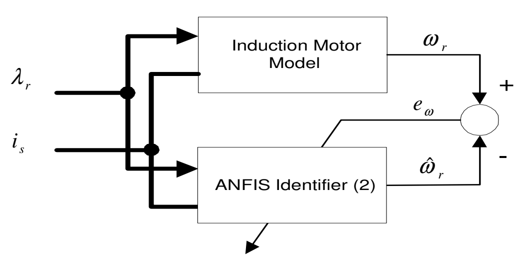

Figure 5. Training scheme of identifier

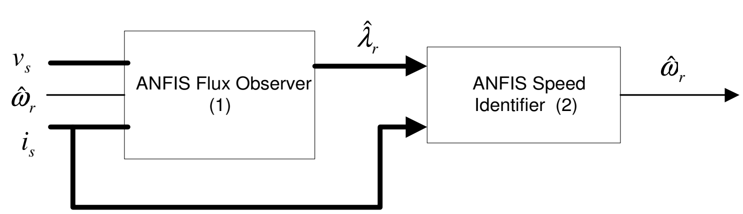

Figure 6. Configuration of flux observer and speed identifier at the simulation

RESULT AND DISCUSSION

The validation stage, observer and identifier is applied to connecting with induction motor, as shown the following figure 7. Time sampling 0.00005 seconds, motor were running for 0.5 seconds.

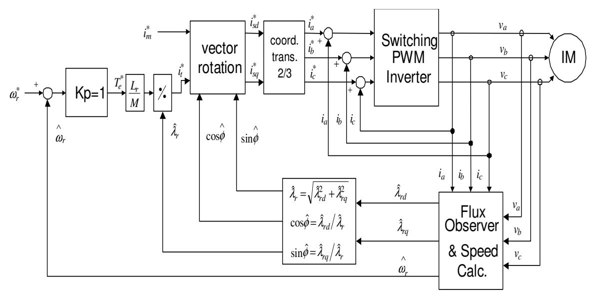

Figure 7. overall system used for simulation

Parameters used for observer and identifier:

Table 1. Parameters of observer and identifier

Parameter	I	II	III
Total Nodes of the input layer	32	32	32
Total Nodes of the 1st hidden layer	8	8	8
Total Nodes of the 2nd hidden layer	8	8	8
Total Nodes of the output layer	1	1	1
Total Parameters	104	104	104
Total Training data pairs	5.000	5.000	5.000
Epoch	200	500	1000

The investigated simulation is done by using the ANFIS composition which determined in the table 1, the standard Error Estimation (SEE) is found in the first training 200 epoch results SEE direct rotor flux 0.0437 *Wb, SEE* quadratur rotor flux 0.0412 *Wb,* and *SEE* rotor angular speed identifier 19.2394 *rad/s,* the graphic of simulation result is shown in the fig 8 for direct rotor flux estimation , fig.9 for the quadrature rotor flux estimation and speed angular is shown in the Fig.10.

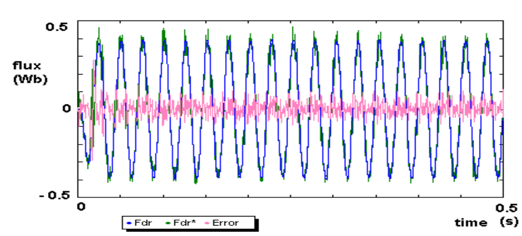

Figure 8. Simulation results for direct rotor flux estimation with training 200 epochs

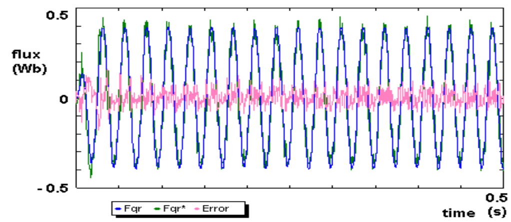

Figure 9. Simulation results for quadrature rotor flux estimation with training 200 epochs

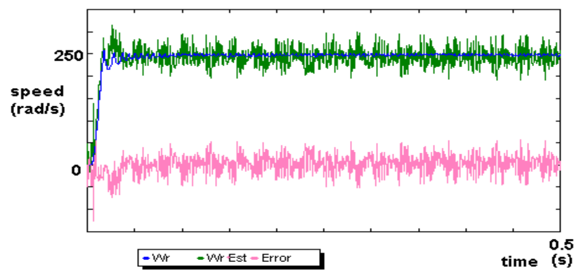

Figure 10 Simulation results for rotor angular speed estimation with training 200 epochs

CONCLUSION

The ANFIS observer is completely simulated for identify the speed induction motor and the results show that rotor flux observer and rotor angular speed identifier can estimate flux rotor and rotor angular speed. Thus rotor flux observer and rotor angular speed identifier shown a good performance by using the back propagation learning method.. The problem is for the speed under 200 rpm, the observer cannot estimate clearly.

Rating and Parameters of Induction Motor Used for Simulation

10 Watt , 115 Volt, 2 poles, 60 Hz

R_s = 176	[Ohm]
R_r = 190	[Ohm]
L_s = 3.079	[H]
L_r = 3.31	[H]
M= 3.21	[H]
J = 0.0000105	[kg.m^2]
K_d = 1.49e-5	[kg.m^2 / s]

REFERENCES

Boldea, I. (1992), “Vector Control of AC Drives,” CRC Press, Inc.

Baburaj Karanayil et al (2007). Online Stator and Rotor Resistance Estimation Scheme Using Artificial Neural Networks for Vector Controlled Speed Sensorless Induction Motor Drive. *IEEE Transactions On Industrial Electronics*, VOL. 54, NO. 1.

George Ellis*(nd). Observer In Control System A Practical Guide “,* Academic Press.

Hisao K. and Kouki M. (1994). Speed Sensorless Field Oriented Control of Induction Motor with Rotor Resistance Adaptation. *IEEE Trans. On Industrial Applications,* Vol. 30, No. 5.

Liu Xu1, Ruan Yi1, et al(2008). *On Speed Sensorless Vector Control System for Induction Motor Based on Estimating Speed by Torque Current Differential.* Proceedings of the 27th Chinese Control Conference , Kunming,Yunnan, China

Murat Barut, *Student Member, IEEE*, et al(2007). Speed-Sensorless Estimation for Induction Motors Using Extended Kalman Filters. *IEEE Transactions On Industrial Electronics*, VOL. 54, NO. 1.

M.N. Cirstea, A. Dinu, J.G. Khor,M. McCormick (nd). *Neural and Fuzzy Logic Control of Drives and Power Systems* ", **ewnes**

N.Ravi Sankar Reddy et al(2010), "*Adaptive Neural Fuzzy Inference Systems Controller for Hybrid PWM based vector controlled Induction motor drives",* 16th National Power Systems Conference.

Jang, J. S. R(1992)., "Self Learning Fuzzy Controllers Based on Temporal Back Propagation", IEEE *Trans. On Neural Networks*, Vol.3, No. 5.

Jang, J. S. R., C. T. Sun and E. Mizutani (1997). *Neuro-Fuzzy and Soft Computing*. Prentice-Hall International, Inc.

S.L. Chiu(1994). Fuzzy Model Identification Based On Cluster Estimation. *Journal of Intelligent and Fuzzy Systems.*

Zhang Yongchang and Zhao Zhengming (2008). Speed Sensorless Control for Three-Level Inverter-Fed Induction Motors Using an Extended Luenberger Observer. *IEEE Vehicle Power and Propulsion Conference (VPPC).*