Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 48 (2015) 753 – 768

Procedia Computer Science

International Conference on Intelligent Computing, Communication & Convergence

(ICCC-2015)

Conference Organized by Interscience Institute of Management and Technology,

Bhubaneswar, Odisha, India

Induction Motor Control Using PSO-ANFIS

Sakuntala Mahapatra*, Raju Daniel**, Deep Narayan Dey*, Santanu Kumar Nayak

*Dept. of Electronics and Telecommunication Engineering, Trident Academy of Technology, Bhubaneswar, Odisha-751024
**Scientist-SF at IPR (Autonomous Inst. of the Department of Atomic Energy, India)
Email: mahapatra.sakuntala@gmail.com

Abstract

The speed of the Induction motor can be adjusted to a great extent so as to provide easy control and high performance. In this paper an effort is made to develop a prototype model to control the speed of an induction motor using PSO-ANFIS hybrid technique. We describe the use of Particle Swarm Optimization (PSO) and ANFIS for designing an optimal fuzzy logic controller of an Induction Motor. We have used two input parameters like speed, torque and output is firing angle. PSO-ANFIS speed controller obtains better dynamic behavior and superior performance of the Induction motor speed control. Similar approach can be correlated to the control of plasma column and which can be implemented in fusion reactor to control the plasma column. Plasma position and shape control is very crucial for the overall performance of the fusion reactor such as tokamak. The quality of the discharge in the Saskatchewan TORus-Modified (STOR-M) tokamak is strongly related to the position of the plasma column within the discharge vessel. If the plasma column approaches too near the wall, then either minor or complete disruption occurs. Consequently it is necessary to be able to control dynamically the position of the plasma column throughout the entire discharge.

A comparison analysis of PSO-ANFIS and Fuzzy Back Propagation algorithm is taken in our work to control the speed of induction motor, where the PSO-ANFIS gives better result in terms of fast computing.

Keywords- Fusion reactors; Plasma confinement; PSO; ANFIS; Fuzzy Logic; Optimization

1.INTRODUCTION

PLASMA column control is very necessary for the successful operation of a thermonuclear fusion reactor such as a tokamak. The tokamak is a magnetic confinement system, and is currently the most promising system for the near term development of a fusion reactor. One of the main challenges associated with plasma confinement is the stable

Peer-review under responsibility of scientific committee of International Conference on Computer, Communication and Convergence (ICCC 2015)
doi:10.1016/j.procs.2015.04.212

maintenance of the plasma column near its equilibrium position. This problem has represented a great challenge to scientists and engineers over the years. PSO-ANFIS hybrid model is implemented for the control and prediction of induction motor and similar approach can be used to control plasma column.

In recent years, high performance and high efficiency induction motor (IM) drives are in great demand to serve the industrial needs for sophisticated products and services, such as steel mills, paper mills, servos, machines tools, robotics, elevators, transportation system etc. This is due to its well-known advantages of simple construction, reliability, ruggedness, and low cost. Although IM has been widely used for constant speed operation earlier, it is more and more accepted in advanced variable speed applications. With the advent of vector or field oriented control whereby IM can be controlled like a separately excited dc motor, the high performance control of IM drives entered into its new era. The main problem with the induction motor is difficulty of control. Because of this, its applications in the area of speed control have traditionally been limited. The advent of Field Oriented Control or Vector Control [2] was a major breakthrough in the area of induction motor control. A major drawback of a lot of vector control schemes is that they require a speed sensor. Speed sensors are usually expensive, bulky and reduce the advantage of an induction motor drive. Also, it is not practical to employ speed sensors in some applications, such as motor drives in hostile environments and high speed motor drives. Because of these drawbacks researchers have put in a lot of effort in developing speed sensor less induction motor drives [4].

In recent years, Artificial Neural Networks (ANN) has found widespread use in function approximation [4]. It has been shown that theoretically a three layer ANN can approximate arbitrarily closely any nonlinear function provided the function is non-singular. The main objective of this work is to develop a general purpose induction motor speed estimator based on the dynamic equations of the induction motor. A technique is presented for estimating the speed of an induction motor using PSO-ANFIS.

Fuzzy theory was first proposed and investigated by Prof. Zadeh in 1965. The Mamdani Fuzzy Inference System (FIS) was presented to control a steam engine and boiler combination by linguistic rules. Fuzzy logic is expressed by means of IF-THEN rules with the human language. In the design of a fuzzy logic controller, the mathematical model is not necessary. Therefore the Fuzzy Logic Controller (FLC) is of good robustness [1]. Owing to its easy application, it has been widely used in industry. However, the rules and the membership functions of a fuzzy logic controller are based on expert experience or knowledge database. Much work has been done on the analysis of fuzzy control rules and membership function parameters [14]. The PSO (particle swarm optimization) algorithms are used to get the optimal values and parameters of our FLC. The PSO is based on a metaphor of social interaction. It searches a space by adjusting the trajectories of individual vectors, called ‘particles’, as they are conceptualized as moving as points in multidimensional space. The individual particles are drawn stochastically towards the positions of their own previous best performances and the best previous performance of their neighbors. Of these is the PSO algorithms are applied to choose membership functions and fuzzy rules [15]. However, the expert experiences or knowledge are still necessary for the ranges of membership functions. In this paper, a novel strategy is proposed for designing the optimal fuzzy controller.

PSO algorithms are applied to search globally optimal parameters of fuzzy logic. The best ranges of membership functions, the best shapes of membership functions and the best fuzzy inference rules are dug out at the same time. Simulation results are given to show the effectiveness of FLC-Swarm controller [11].

An induction or asynchronous motor is an AC motor in which all electromagnetic energy is transferred by inductive coupling from a primary winding to a secondary winding, the two windings being separated by an air gap. In three-phase induction motors, that are inherently self-starting, energy transfer is usually from the stator to either a wound rotor or a short-circuited squirrel cage rotor. Three-phase cage rotor induction motors are widely used in industrial drives because they are rugged, reliable and economical. Single-phase induction motors are also used extensively for smaller loads.

In both induction and synchronous motors, the AC power supplied to the motor's stator creates a magnetic field that rotates in time with the AC oscillations. Whereas a synchronous motor's rotor turns at the same rate as the stator

field, an induction motor's rotor rotates at a slower speed than the stator field. The induction motor stator's magnetic field is therefore changing or rotating relative to the rotor. This induces an opposing current in the induction motor's rotor, in effect the motor's secondary winding, when the later is short-circuited or closed through external impedance. The rotating magnetic flux induces currents in the windings of the rotor, in a manner similar to currents induced in transformer's secondary windings. These currents in turn create magnetic fields in the rotor that react against the stator field. Due to Lenz's Law, the direction of the magnetic field created will be such as to oppose the change in current through the windings. The cause of induced current in the rotor is the rotating stator magnetic field, so to oppose this; the rotor will start to rotate in the direction of the rotating stator magnetic field. The rotor accelerates until the magnitude of induced rotor current and torque balances the applied load. Since rotation at synchronous speed would result in no induced rotor current, an induction motor always operates slower than synchronous speed [14].

For these currents to be induced, the speed of the physical rotor must be lower than that of the stator's rotating magnetic field (n_s), or the magnetic field would not be moving relative to the rotor conductors and no currents would be induced. As the speed of the rotor drops below synchronous speed, the rotation rate of the magnetic field in the rotor increases, inducing more current in the windings and creating more torque. The ratio between the rotation rate of the magnetic field as seen by the rotor (slip speed) and the rotation rate of the stator's rotating field is called slip. Under load, the speed drops and the slip increases enough to create sufficient torque to turn the load. For this reason, induction motors are sometimes referred to as asynchronous motors. An induction motor can be used as an induction, or it can be unrolled to form the linear induction motor which can directly generate linear motion.[15,16]

II. THE PROPOSED EXPERIMENTAL MODEL

A. Neurofuzzy Back Propagation Algorithm

In this paper we have implemented a hybrid fuzzy neural network model in which we can map for both real input real output and fuzzy input to fuzzy output with use of fuzzy weights and fuzzy biases. The back propagation algorithm is implemented for neural networks, to train fuzzy systems to match desired input-output pairs. The key ideas in developing this training algorithm are to view a fuzzy system as a three-layer feed forward network, and to use the chain rule to determine gradients of the output errors of the fuzzy system with respect to its design parameters. It is shown that this training algorithm performs an error back propagation procedure: hence, the fuzzy system equipped with the back propagation training algorithm is called the Fuzzy back propagation [1, 8].

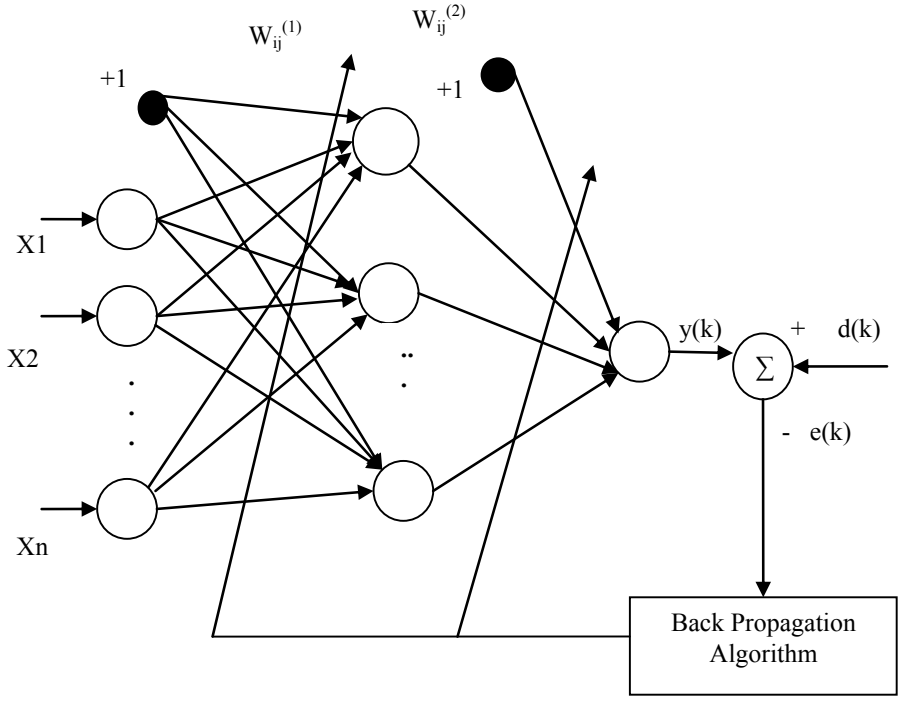

Fig-1: Fuzzy back propagation model

An MLP network with 2-3-1 neurons (2, 3 and 1 denotes the number of neurons in the input layer, the hidden layer, the second hidden layer and the output layer respectively) with the back-propagation (BP) learning algorithm, is depicted in Fig-1. The parameters of the neural network can be updated either in sequential and batch mode of operation.

The output from the k[th] node is

$$\bar{Y}_K^{(M)}(p) = f\left(\sum_{j=0}^{nm} \bar{W}_{kj}^{(M-1)}(p) . \bar{Y}_j^{(M-1)}(p)\right) \quad (1)$$

Where f(.) is a nonlinear logistic function.

Triangular membership function is taken for the weights which are represented by

$$\bar{W}_{kj}^{(m-p)}(p) = \bar{W}_{kjl}^{(M-1)}(p)/\bar{W}_{kjc}^{(M-1)}(p)/\bar{W}_{kju}^{(M-1)}(p) \quad (2)$$

$$l: lower(\alpha = 0), c: center\ (\alpha = 1)\ and\ u: upper(\alpha = 0)$$

Similarly

$$\bar{Y}_k^{(M)}(p) = y_{kl}^M(p)/y_{kc}^M(p)/y_{ku}^M(p) \quad (3)$$

Equation (2) is computed by min-max principle as;

$$y_{kl}^M(p) = f\left(\sum_{j=0}^{nm} \min\ (\bar{W}_{kj[0]}^{(M-1)}(p) . \bar{Y}_{j[0]}^{(M-1)}(p)\right) \quad (4)$$

$$y_{kc}^M(p) = f\left(\sum_{j=0}^{nm} (\bar{W}_{kj[1]}^{(M-1)}(p) . \bar{Y}_{j[1]}^{(M-1)}(p))\right) \quad (5)$$

$$y_{ku}^M(p) = f\left(\sum_{j=0}^{nm} \max\ (\bar{W}_{kj[0]}^{(M-1)}(p) . \bar{Y}_{j[0]}^{(M-1)}(p)\right) \quad (6)$$

The error at (M)[th] layer is,

$$e_{kr}^M(p) = d_{kr}(p) - y_{kr}^M(p) \quad (7)$$

Where , $r = (l\ /\ c\ /\ u)$

Here a generalized cost function is used to minimize and update weights.

$$w_{kjr}^{M-1}(p+1) = w_{kjr}^{M-1}(p) - \eta . \frac{\partial E_{\lambda}(p)}{\partial w_{kjr}^{M-1}(p)} \quad (8)$$

The stepwise algorithm for back propagation algorithm is listed out here. The training phase of back propagation involves four basic stages. They are as follows:

(i) Initialization of Weights
(ii) Feed Forward
(iii) Back Propagation of the errors
(iv) Updating of the weights and biases.

The initial random weights are selected between [-0.5, 0.5]. Each unit receives an input signal and delivers it to all the hidden units with activation function. The signal is then sent to each of the output units. Finally the output unit provides the desired output to form the response of the net for the given input pattern applied to the structure. Each output units compares its computed activation function *y(k)* with its target value *d(k)* to determine the associated error for the input pattern with that unit. The final output *y(k)* is compared with the desired output *d(k)* and the resulting error signal *e(k)* is thus produced using the error equation. This error function is then feed back to the structure.

B. Adaptive Network Based Fuzzy Inference Systems (ANFIS)

Fuzzy systems present particular problems to a developer:

- **Rules:** The IF-THEN rules have to be determined. This is usually done by 'knowledge acquisition' from an expert. It is a time consuming process that is fraught with problems.
- **Membership functions**: A fuzzy set is fully determined by its membership function. Thus proper choice of membership function is challenging one.

The ANFIS approach learns the rules and membership functions from data. ANFIS is an adaptive network. An adaptive network is network of nodes and directional links. Associated with the network is a learning rule - for example back propagation. It's called adaptive because some, or all, of the nodes have parameters which affect the output of the node. These networks are learning a relationship between inputs and outputs [14].
Adaptive networks cover a number of different approaches but for our purposes we will investigate in some detail the method proposed by Jang known as ANFIS [17].
The ANFIS architecture is shown below. The circular nodes represent nodes that are fixed whereas the square nodes are nodes that have parameters to be learnt.

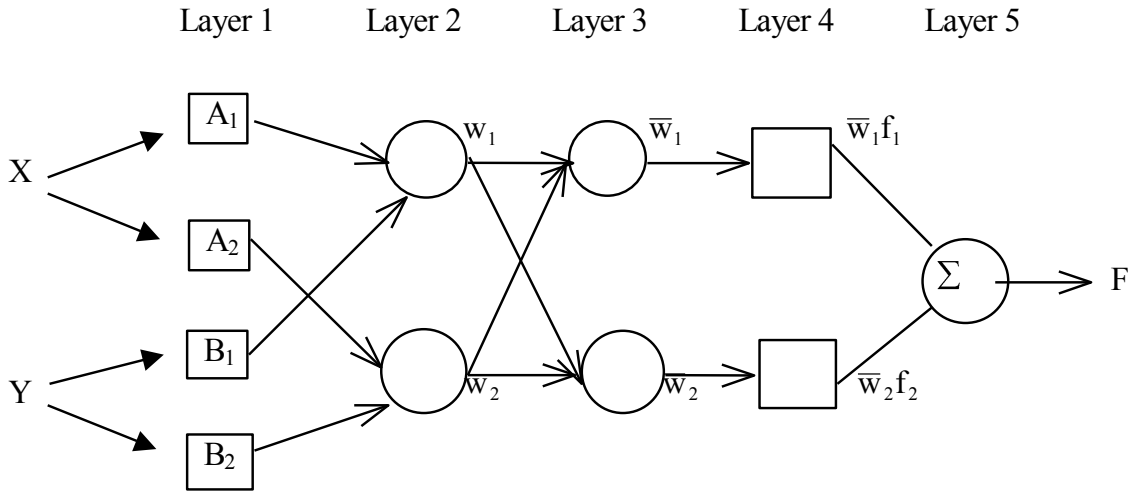

Fig-2: An ANFIS architecture for a two rule Sugeno system

A Two Rule Sugeno ANFIS has rules of the form:

$$\begin{aligned} &If\ x\ is\ A_1\ and\ y\ is\ B_1 \quad THEN\ f_1 = p_1x + q_1y + r_1 \\ &If\ x\ is\ A_2\ and\ y\ is\ B_2 \quad THEN\ f_2 = p_2x + q_2y + r_2 \end{aligned} \tag{9}$$

For the training of the network, there is a forward pass and a backward pass. We now look at each layer in turn for the forward pass. The forward pass propagates the input vector through the network layer by layer. In the backward pass, the error is sent back through the network in a similar manner to back-propagation.

Layer 1

The output of each node is:

$$O_{1,i} = \mu_{A_i}(x) \qquad for\ i = 1,2$$

$$O_{1,i} = \mu_{B_{i-2}}(y) \qquad for\ i = 3,4$$

So, the $O_{1,i}(x)$ is essentially the membership grade for x and y.

The membership functions could be anything but for illustration purposes we will use the bell shaped function given by:

$$\mu_A(x) = \frac{1}{1 + \left| \frac{x - c_i}{a_i} \right|^{2b_i}} \tag{10}$$

Where a_i, b_i, c_i are parameters to be learnt. These are the premise parameters.

Layer 2

Every node in this layer is fixed. This is where the t-norm is used to 'AND' the membership grades - for example the product:

$$O_{2,i} = w_i = \mu_{A_i}(x)\mu_{B_i}(y), \quad i = 1,2 \tag{11}$$

Layer 3

Layer 3 contains fixed nodes which calculate the ratio of the firing strengths of the rules:

$$O_{3,i} = \overline{w}_i = \frac{w_i}{w_1 + w_2} \tag{12}$$

Layer 4

The nodes in this layer are adaptive and perform the consequent of the rules:

$$O_{4,i} = \overline{w_i} f_i = \overline{w_i}(p_i x + q_i y + r_i) \tag{13}$$

The parameters in this layer (p_i, q_i, r_i) are to be determined and are referred to as the consequent parameters.

Layer 5

There is a single node here that computes the overall output:

$$O_{5,i} = \sum_i \overline{w_i} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{14}$$

This output is taken as the actual output of our proposed model to compare with the desired output which is taken from the experimental data to calculate the error.

C. Particle Swarm Optimization (PSO) Algorithm Operation

Particle Swarm Optimization [7-9] optimizes an objective function by undertaking a population – based search. The population consists of potential solutions, named particles, which are metaphor of birds in flocks. These particles are randomly initialized and freely fly across the multi dimensional search space. During flight, each particle updates its own velocity and position based on the best experience of its own and the entire population. The various steps involved in Particle Swarm Optimization Algorithm [8] are as follows:

Step 1: The velocity and position of all particles are randomly set to within pre-defined ranges.

Step 2: Velocity updating – At each iteration, the velocities of all particles are updated according to,

$$v_i = v_i + c_1 R_1 (p_{i,best} - p_i) + c_2 R_2 (g_{i,best} - p_i) \tag{15}$$

where p_i and v_i are the position and velocity of particle i, respectively; $p_{i,best}$ and $g_{i,best}$ is the position with the 'best' objective value found so far by particle i and the entire population respectively; w is a parameter controlling the dynamics of flying; R_1 and R_2 are random variables in the range [0,1]; c_1 and c_2 are factors controlling the related weighting of corresponding terms. The random variables help the PSO with the ability of stochastic searching.

Step 3: Position updating – The positions of all particles are updated according to,

$$p_i = p_i + v_i \quad \textbf{(16)}$$

After updating, p_i should be checked and limited to the allowed range.

Step 4: Memory updating – Update $p_{i,best}$ and $g_{i,best}$ when condition is met,

$$\begin{aligned} p_{i,best} &= p_i \qquad if\ f(p_i) > f(p_{i,best}) \\ g_{i,best} &= g_i \qquad if\ f(g_i) > f(g_{i,best}) \end{aligned} \quad \textbf{(17)}$$

Where f*(x)* is the objective function to be optimized.

Step 5: Stopping Condition – The algorithm repeats steps 2 to 4 until certain stopping conditions are met, such as a pre-defined number of iterations. Once stopped, the algorithm reports the values of g_{best} and $f(g_{best})$ as its solution.

PSO [16] utilizes several searching points and the searching points gradually get close to the global optimal point using its pbest and gbest. Initial positions of pbest and gbest are different. However, using thee different direction of pbest and gbest, all agents gradually get close to the global optimum.

D. Flow Chart of our proposed model

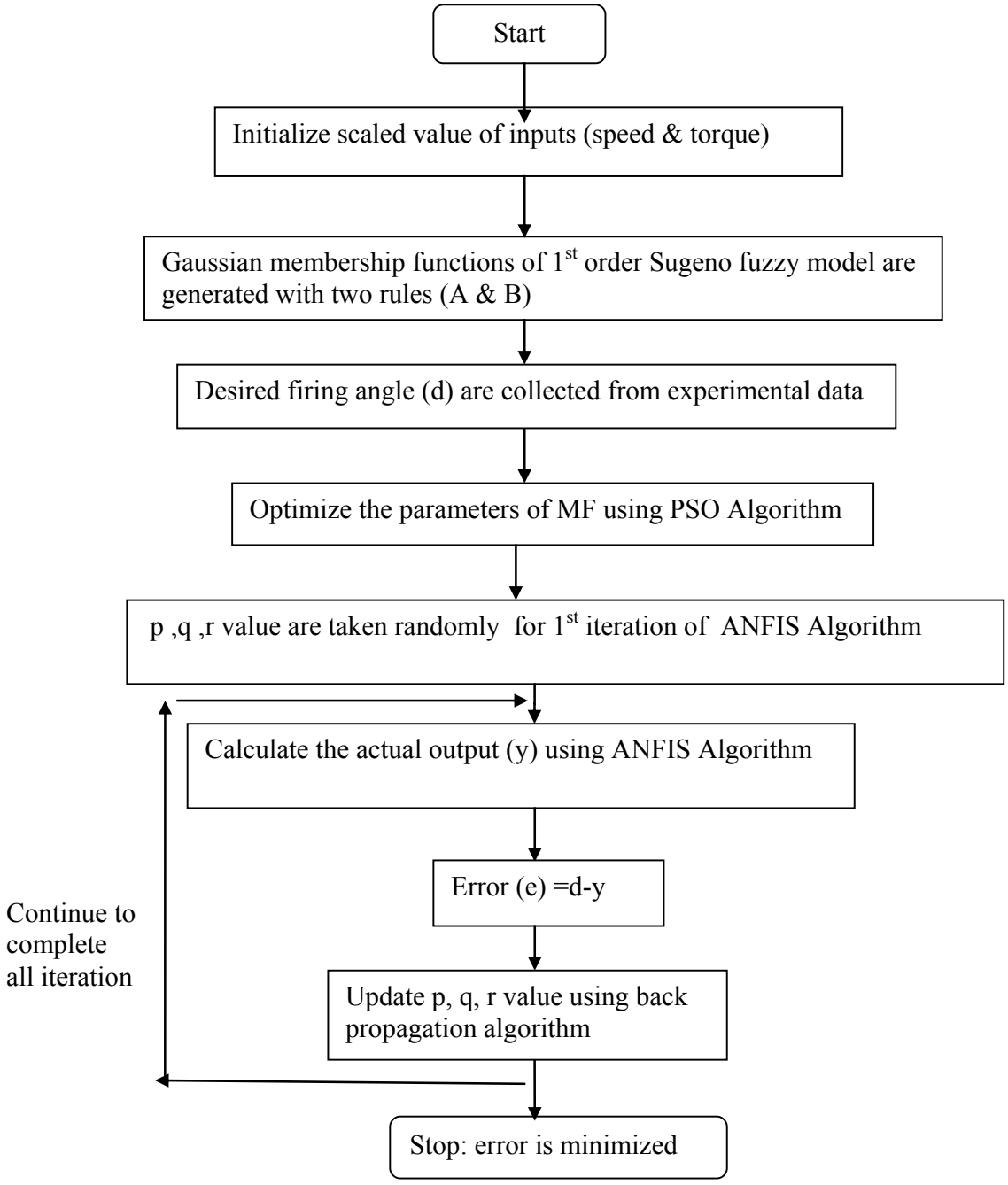

Fig-3: flow chart of our proposed model

E. Algorithm of our proposed model

1. Scaled value of motor speed and torque is taken as inputs (x and y).
2. Using two input first-order Sugeno Fuzzy model with two rules (A and B) are generated.
3. Gaussian Membership functions are plotted.
4. Desired Firing angles are taken from experimental data as'd'.
5. Using ANFIS w1, w2 are found out.

w1=A1(x) *B1(y)
w2=A2(x)*B2(y)

6. Then p1, p2, q1, q2, r1, r2 are randomly taken for 1st iteration.
7. Then the actual output is calculated using ANFIS technique.

IF x is A_1 AND y is B_1 THEN $f_1 = p_1 x + q_1 x + r_1$
IF x is A_2 AND y is B_2 THEN $f_2 = p_2 x + q_2 x + r_2$
The reasoning mechanism for this model is:

$$f = \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2} = \overline{w_1} + \overline{w_2}$$

8. Then the error is calculated from the difference of desired and actual output.

e=d-f

9. Then the p1, p2, q1, q2 are updated using back propagation algorithm for each iteration till the outputs are matched.
10. At last Error converging graph and Desired-actual output matching graphs are plotted.

III. SIMULATION RESULT

Desired Vs Actual Value of Firing Angle

Motor Specification:

- **AC Induction Motor,**
- **220volt,**
- **1600 rpm ,**
- **P=30w**

SL NO	DESIRED F.A	ACTUAL F.A
1	11	13.3357
2	13	15.0416
3	15	16.7474
4	17	18.4533
5	19	20.1592
6	22	21.8650
7	24	23.5709
8	27	25.2767
9	28	26.9826
10	29	28.6885
11	31	30.3943
12	33	32.1002
13	34	33.8060
14	35	35.5119
15	37	37.2178
16	38	38.9236
17	40	40.6295
18	42	42.3353
19	44	44.0412
20	46	45.7471
21	48	47.4529
22	49	49.1588
23	50	50.8647
24	53	52.5705
25	55	54.2764
26	56	55.9822
27	58	57.6881
28	61	59.3940
29	62	61.0998
30	63	62.8057
31	65	64.5115
32	66	66.2174
33	68	67.9233
34	69	69.6291
35	71	71.3350
36	74	73.0408
37	76	74.7467
38	77	76.4526
39	78	78.1584
40	80	79.8643

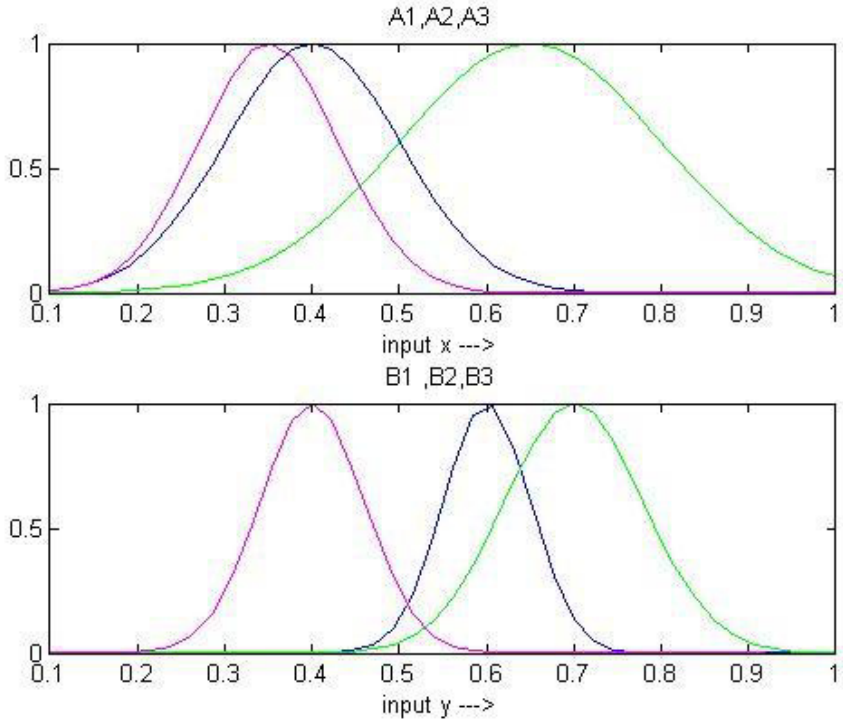

Fig-4: plot of membership function using 2 inputs 3 rules

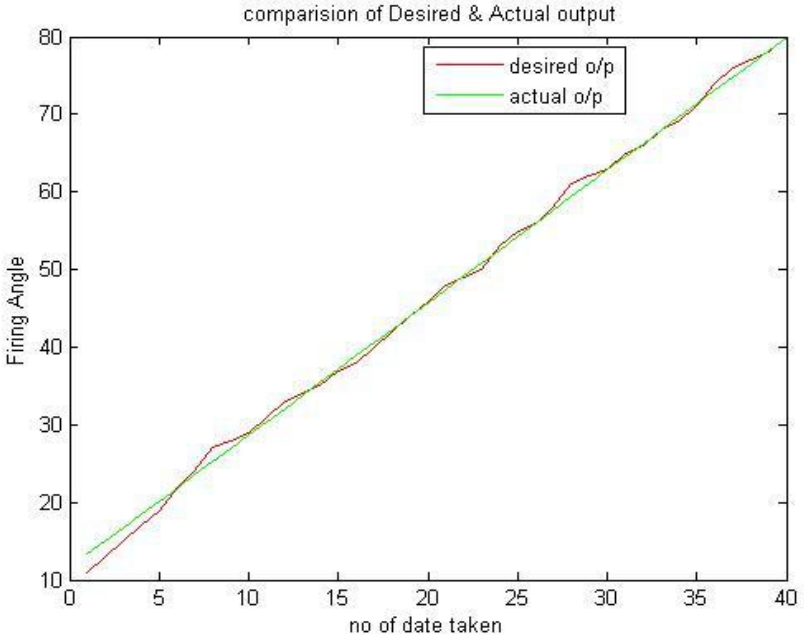

Fig-5: Comparison of Actual and Desired output plot

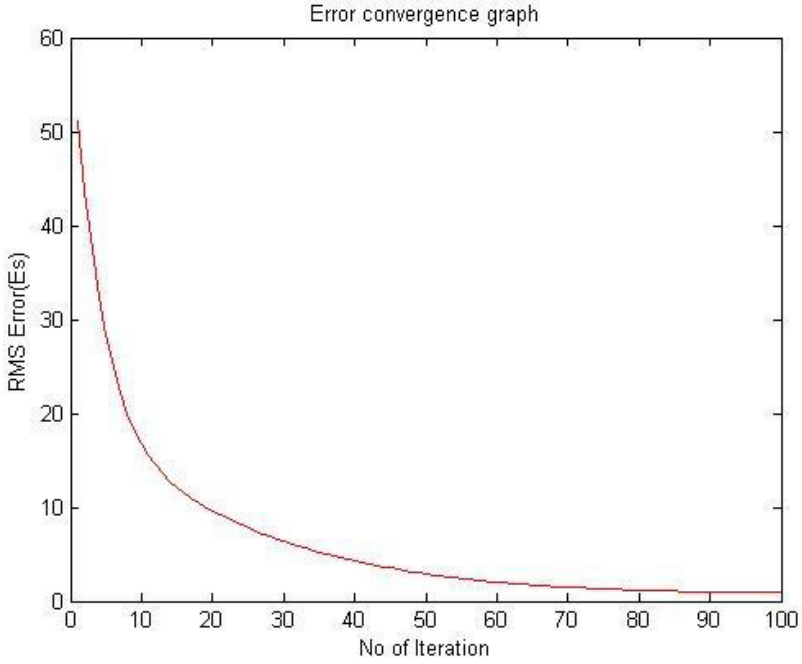

Fig-6: Error convergence plot for ANFIS

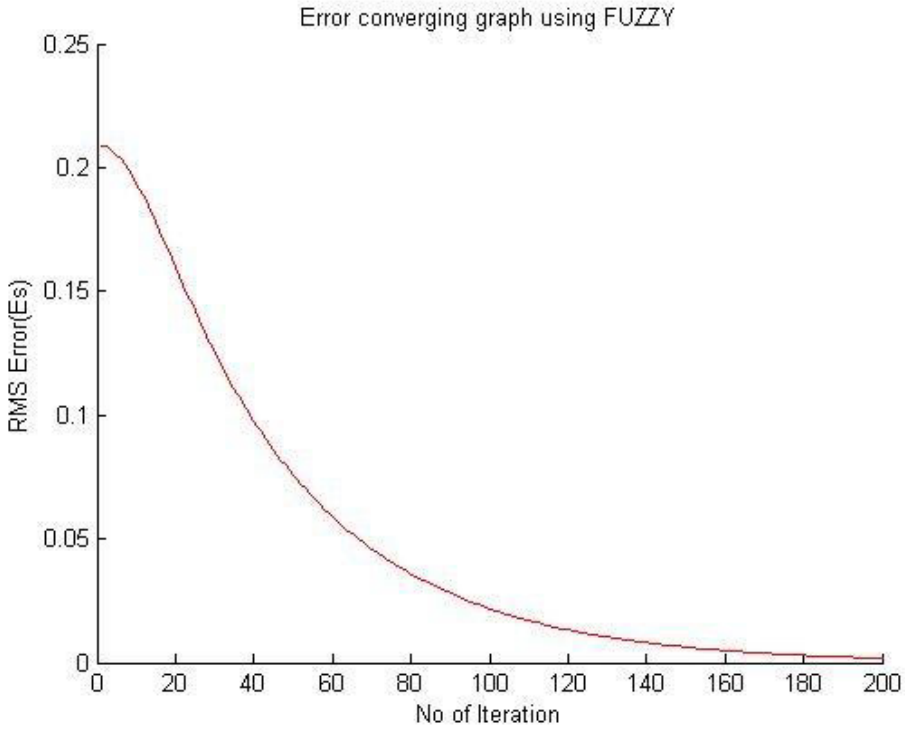

Fig-7: Error convergence plot of Fuzzy back propagation

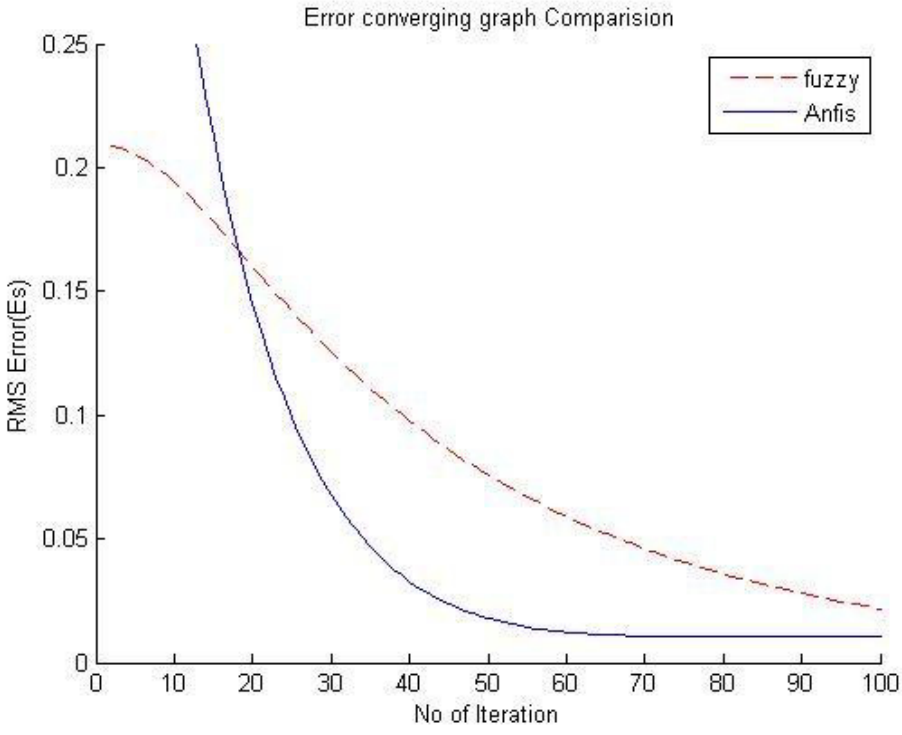

Fig-8: Error convergence comparison between ANFIS and Fuzzy back propagation

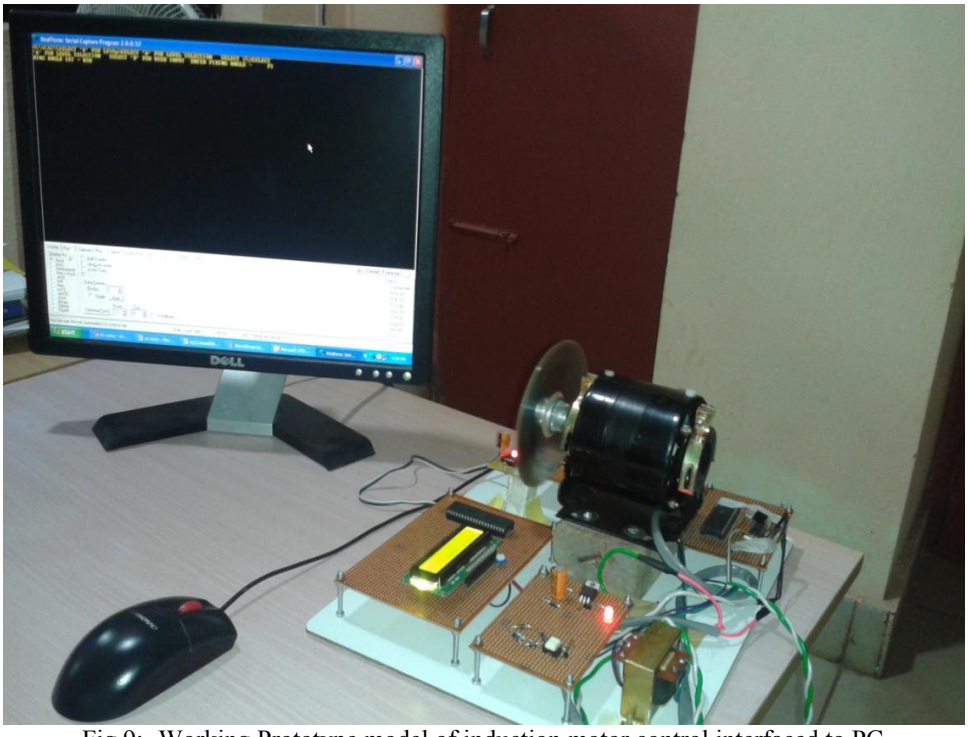

Fig.9:- Working Prototype model of induction motor control interfaced to PC

IV. CONCLUSION

In this paper, the speed of an Induction Motor drive is controlled by the hybrid of PSO- ANFIS algorithms. According to the results of the MATLAB simulation, the Adaptive Neuro Fuzzy (ANFIS) controller efficiently is better than the traditional FLC. The ANFIS-PSO is the best controller which presented satisfactory performances. The major drawback of the fuzzy controller presents an insufficient analytical technique design (choice of the rules, the membership functions and the scaling factors).Thus we chose with the use of the Neural Networks and Particle Swarm Optimization for the optimization of this controller in order to control Induction motor speed. Finally, the proposed controller (PSO-ANFIS) gives a very good result.

The ANFIS-PSO is the best controller which presented satisfactory performances. In the next phase, PSO-ANFIS model would be applied on a nonlinear system followed by development for tokamak plasma position control. By implementing this PSO-ANFIS concept we can predict and control the plasma layer in a fusion reactor which is our future work which is a very challenging but necessary requirement for tokamak.

Acknowledgements

The authors are thankful to Board of Research in Fusion Science & Technology (BRFST), Institute of Plasma Research, Government of India, Ahmedabad for providing the funding to carry out the research work on "Prediction and Control of Plasma layers in a fusion reactor using Hybrid PSO-ANFIS"

REFERENCES

[1] Sakuntala Mahapatra, Santanu K. Nayak, Samrat L. Sabat, "Neuro fuzzy model for adaptive filtering of oscillatory signals", Elsevier Science, Measurement 30,231-239, 2001.

[2] Sakuntala Mahapatra, Samrat L. Sabat, Santanu K. Nayak, " An intelligent instrument for tracking and adaptive filtering of oscillatory signals using Hebbian learning rules", Elsevier Science, Measurement 26, 221-227, July 1999

[3] Samrat L. Sabat, Santanu K. Nayak, Sakuntala Mahapatra,"WNN based intelligent energy meter", Science Direct, Measurement 41, 357-363, 2008

[4] T.S.Radwan,'Perfect Speed Tracking of Direct Torque Controlled-Induction Motor Drive Using Fuzzy Logic', SMIEEE, Riyadh College of Technology, Saudi Arabia, IEEE, 2005.

[5] D. W. Novotny and T. A. Lipo, Vector Control and Dynamics of AC Drives, Oxford, UK, Clarendon, 1996.

[6] P.Tiitinen, "The next generation motor control method, DTC direct torque control", Proceedings of the IEEE Intl. Conf on Power Electronics, Drives, and Energy Systems for lndustrialgrowth, 1996, pp. 37-43.

[7] Ben-Brahim. "Motor Speed Identification via Neural Networks".IEER Ind. Applicat. Magazine, pp. 28-32, Jan/Feb 1995.

[8] R. H. Nielsen. "Theory of Backpropogation Neural Network". In International Joint Conf. on Neural Networks, pp. 1585-Il592, 1989.

[9] C. Schauder. "Adaptive Speed Identification for Vector Control of Induction Motors without Rotational Transducers".IEEE trans. Ind. Applicat., vol. 28, no. 5, pp. 1054-1061,Sep/Oct 1992.

[10] Y. Valle, G. Venayagamoorthy, S. Mohagheghi, J. Hernandez and R.Harley, "Particle swarm optimization: Basic concepts, variants and applications in power systems", IEEE trans. on evolutionary computation, vol. 12, no. 2, April 2008.pp. 171-195.

[11] S. Wahsh and A. Elwer," Improved performance of Permanent Magnet synchronous motor by using Particle swarm optimization techniques," in Proc.of 2007 IEEE International Conference on Robotics, 2008, pp.2095-2100.

[12] Rohit G. Kanojiya, Student, Y.C.C.E, and P. M. Meshram, Associate professor, Y.C.C.E "Optimal Tuning of PI Controller for Speed Control of DC motor drive using Particle Swarm Optimization", IEEE,2012

[13] Y. Tipsuwan, Y. Chow, "Fuzzy Logic Microcontroller Implementation for DC Motor Speed Control", IEEE, 1999.

[14] L. Rajaji,C. Kumar and M. Vasudevan, "Fuzzy and Anfis Based Soft Starter Fed Induction Motor Drive For High Performance Applications." Sathyabama University, India S.K.P. Engineering College, India Vestas RRB India Ltd., India, Vol. 3, No. 4, August 2008

[15] Boumediene Allaoua, Abdellah Laoufi, Brahim Gasbaoui, And Abdessalam Abderrahmani. "Neuro-Fuzzy DC Motor Speed Control Using Particle Swarm Optimization", Department of Electrical Engineering, Bechar University, B.P 417 BECHAR (08000) Algeria

[16] Makarand S. Ballal, Hiralal M. Suryawanshi and Mahesh K. Mishra "Detection of Incipient Faults in Induction Motors using FIS, ANN and ANFIS Techniques", Journal of Power Electronics, Vol. 8, No. 2, April 2008
[17] J.S.R. Jang, "ANFIS: Adaptive-Network-Ba sed Fuzzy Inference System", IEEE Trans. Systems, Man, Cybernetics, 23(5/6):665-685, 1993.
[18] Jordan E. Morelli, Akira Hirose, and Hugh C. Wood, "Fuzzy-Logic-Based Plasma-Position Controller for Stor-M", IEEE Transactions on Control Systems Technology, Vol. 13, No. 2, March 2005