

Design of Neuro-Fuzzy System Controller for DC Servomotor-Based Satellite Tracking System

Linus A. Alwal^{1*}, Peter K. Kihato² and Stanley I. Kamau³

^{1, 2, 3} Department of Electrical and Electronic Engineering, Jomo Kenyatta University of Agriculture & Technology (JKUAT), P.O.BOX 62000-00200, Nairobi, Kenya.

Abstract: A parabolic dish antenna positioning system is a control unit which directs the antenna to desired target automatically. In all its aspects, automation has become widespread and promises to represent the future for satellite tracking antenna systems. The term tracking is used to mean that the control system should utilize suitable algorithms to automate the process of pointing the antenna to a selected satellite thereby establishing the desired line of sight (LOS).

This paper aims to present and discuss the results obtained from the development of a DC servomotor-based Neuro-Fuzzy System Controller (NFSC) algorithm which can be applied in the positioning of the parabolic reflector antennas. The advantage of using NFSC method is that it has a high degree of nonlinearity tolerance, learning ability and solves problems that are difficult to address with the conventional techniques such as Proportional Integral Derivative (PID) strategy. The architecture of the proposed antenna control system employs Adaptive Neuro-Fuzzy Inference System (ANFIS) design environment of MATLAB/SIMULINK package. The results obtained indicated that the proposed NFSC was able to achieve desired output DC motor position with reduced rise time and overshoot. Future work is to apply the controller in real time to achieve automatic satellite tracking with parabolic antenna using data derived from the signal strength.

Keywords: ANFIS, Antenna-Positioning System, DC servomotor, NFSC

I. Introduction

Parabolic antennas mounted at earth stations which are commonly used in satellite tracking applications, are prone to suffer from environmental disturbances [1]. For several years, DC servomotor-based controllers have been applied in closed-loop control systems to position or stabilize the satellite dishes. However, DC servomotors are known to have nonlinear parameters and dynamic factors, such as saturation, backlash and friction which cannot be overlooked. Several controller models have been developed over time to solve the problem of antenna pointing in satellite and movable targets tracking using servomechanism [2], [3], [4], [5], [6], and [7]. These include conventional controllers such as Proportional-Integral (PI), Linear Quadratic Gaussian (LQG) and Proportional-Integral-Derivative (PID) controller on one hand and new intelligent techniques such as Fuzzy Logic Controller (FLC), Neural Network, Fuzzy-neural networks and Fuzzy-genetic algorithm on the other hand [8], [9] and [10]. Conventional controllers such as PI and PID controllers are sensitive to variation in the motor parameters and load. Also, tuning PI or PID gains to reduce the overshoot due to load disturbance is difficult. Moreover, an accurate non-linear model of actual DC motor is difficult to find and parameter values obtained from system identification may be only approximate values. For this reason, many researchers today are interested in applying intelligent adaptive control techniques to achieve fast speed response and tolerance to parameter variations. Fuzzy Logic Controller can be an alternative control technique to the PID controller [1], [2], [8] and [9]. But the major drawback of the FLC is insufficient analytical design technique with respect to the selection of the rules, the membership functions and the scaling factors. On the other hand, artificial neural network (ANN) has the ability to learn and adapt but cannot explain what it has learnt [10]. As a result, integrating FLC and ANN to generate hybrid model can take advantage of strong points of both [11], a strength which is explored here.

1.1. Problem Formulation

The objective is to design an ANFIS controller for DC servomotor-based antenna pointing system to meet the following time domain step response tracking specifications: rise time (t_r) $\leq 3s$, settling time (t_s) $\leq 3s$ and maximum overshoot ($M_p \leq 10\%$). These limits have been selected based on practical industrial standards.

1.2. System Description

Fig. 1 is the control block diagram of the DC servomotor antenna pointing system. The first input to the summer is set position $r(t)$, the desired position of the azimuth or elevation motor. The second input is the

feedback signal, the current position of the respective motor, captured by some feedback sensor like the potentiometer and changed to a summer readable format. The difference between these two inputs, called position error signal $e(t)$, is given to the controller that reads the signal and produces appropriate output signal, controller output $u(t)$. The controller output reaches the motor driver, which produces a proportional output to rotate the respective motor in either direction according to the sign of the error signal. As the desired position is approached, the error signal reduces to zero and the motor stops [6].

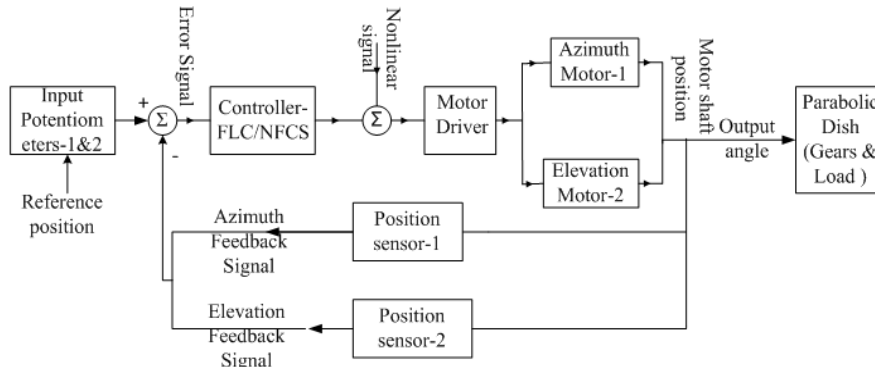


Fig.1. Block diagram of antenna control mechanism

II. Mathematical Modeling of DC (Servo) motor System

Fig.2. represents the DC (servo) motor model with parameters defined in Table 1. For an armature-controlled separately-excited DC motor, the voltage applied to the armature of the motor is varied without changing the voltage applied to the field.

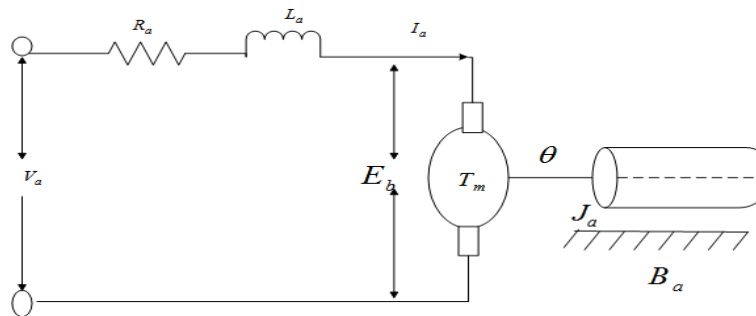


Fig.2. DC motor Circuit Diagram

Using Kirchoff's Voltage Law, the output armature voltage $V_a(t)$ and motor torque is related to (1) while motor torque $T_m(t)$ is related to the armature current $I_a(t)$ by a constant factor K_T given in (2):

$$V_a(t) = R_a I_a(t) + L_a \frac{dI_a(t)}{dt} + E_b(t) \tag{1}$$

$$T_m(t) = K_T I_a(t) \tag{2}$$

The back electromotive force (e.m.f) $E_b(t)$ is related to the angular velocity by (3). Applying Newton's Law, Kirchoff's Law and Laplace transform [7] generates (4) and (5), from which (6) results by eliminating current.

$$E_b(t) = K_T \omega_m(t) = K_T \frac{d\theta}{dt} \tag{3}$$

$$J_a s^2 \theta(s) + B_a s \theta(s) = K_T I_a(s) \tag{4}$$

$$L_a s I_a(s) + R_a I_a(s) = V_a(s) - K_T s \theta(s) \tag{5}$$

$$J_a s^2 \theta(s) + B_a s \theta(s) = K_T \frac{V_a(s) - K_T s \theta(s)}{R_a + L_a s} \tag{6}$$

The transfer function from the input voltage, $V_a(s)$ to output angle θ directly follows (7):

$$G_a(s) = \frac{\theta(s)}{V_a(s)} = \frac{K_T}{s[(R_a + L_a s)(J_a s + B_a) + K_T K_B]} \tag{7}$$

In a fixed field motor it is assumed that $K_T = K_B$ and $R_a \gg L_a$ which simplifies (7) to (8):

$$\frac{\theta_m(s)}{E_a(s)} = \frac{\frac{K_T}{JR_a}}{s(s + \frac{B_m R_a + K_T K_B}{JR_a})} \tag{8}$$

Fig. 3 is a block diagram of a DC (servo) motor system showing elements of the transfer function.

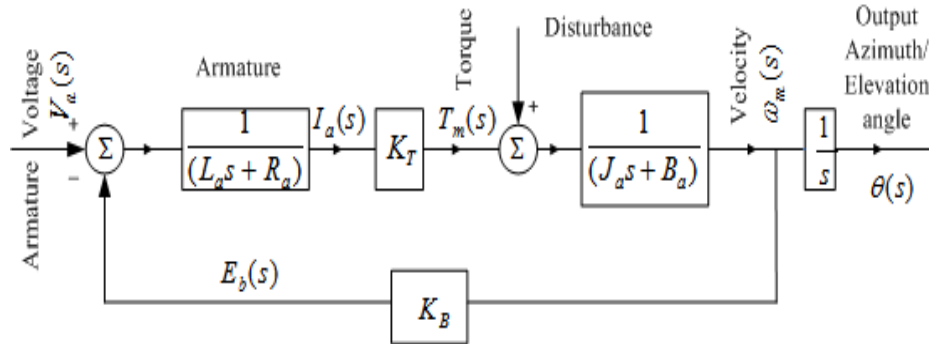


Fig.3. Block-Diagram of Separately Excited DC (servo) motor

III. Antenna Positioning System Transfer Function and PID Controller

The system parameters are given in Table 1 [4, 6]. Detailed derivation of the system transfer functions without any compensator as well as the assumptions made is presented in [6, 7]. The open-loop transfer function for output angular velocity $\omega_o(t)$ without feedback is realized as given by (9):

$$G_a(s) = \frac{\omega_o(s)}{V_p(s)} = \frac{20.83}{s^2 + 101.71s + 171} \tag{9}$$

Table 1 Parameters of Model with DC Servomotors

Parameter	Definition	Azimuth/Elevation
a	Power Amplifier Pole	100
a_m	Motor and Load Pole	1.71
B_a	Motor Dampening Constant[Nms/rad]	0.01
B_L	Load Dampening Constant[Nms/rad]	1
B_m	Equivalent viscous friction coeff. [Nms/rad]	0.02
J_a	Motor Inertial Constant[Kgm ²]	0.02
J_L	Load Inertial Constant[Kgm ²]	1
J_m	Equivalent moment of inertia[Kgm ²]	0.03
K	Preamplifier Gain	-
K₁	Power Amplifier Gain	100
K_B	Back emf Constant[V/s/rad]	0.5
K_g	Gear Ratio	0.1
K_m	Motor and Load Gain	2.083
K_{pot}	Potentiometer Gain	0.318
K_T	Motor Torque Constant[Nm/A]	0.5
L_a	Motor Armature Inductance[H]	0.45
N	Turns on Potentiometer	10
N₁, N₂ N₃	Gear Teeth(Respectively)	25,250,250
R_a	Motor Armature Resistance[Ω]	8
V	Voltage across Potentiometer[V]	10

The closed-loop transfer function without the PID compensator is obtained in (10) by using (9) and the block diagram reduction. According to Routh-Herwitz criterion, the system will give stable response with the value of "K" in the range $0 < K < 2623$, hence 100 is taken [6], [7], giving (11):

$$\frac{\theta_o(s)}{\theta_i(s)} = \frac{6.63K}{s^3 + 101.71s^2 + 171s + 6.63K} \tag{10}$$

$$\frac{\theta_o(s)}{\theta_i(s)} = \frac{663}{s^3 + 101.71s^2 + 171s + 663} \tag{11}$$

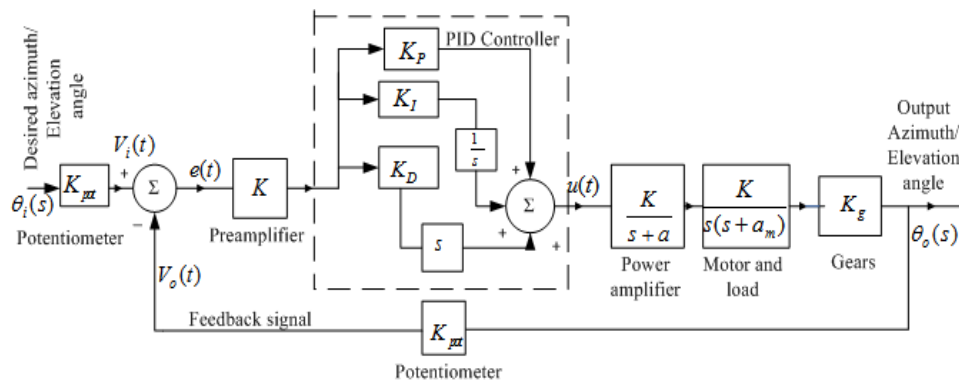


Fig.4. Block diagram of Antenna Azimuth PID controller

The structure of a PID controller applied to antenna positioning system is shown in Fig.4 while its parallel form, [4], is given in (12):

$$U(t) = K_p e(t) + K_I \int e(t) + K_D \frac{de(t)}{dt} \tag{12}$$

where K_p , K_I and K_D are proportional, integral and derivative gains and $e(t)$ = error. It is clear that the control signal is made up of the sum of three stated gain components. The values of K_p , K_I and K_D are obtained using Ziegler- Nichols tuning algorithm in [6, 7]. The three PID controller gain parameters that gave the most stable response were obtained in [7], and quoted as follows: $K_p = 16$, $K_I = 5$ and $K_D = 2$ [7].

IV. Fuzzy Logic Controller (FLC) Design

As shown in Fig.5, the FLC system has four main components: Fuzzification Interface (converts a crisp point into a fuzzy set), Knowledge Base (consists of a database hosting membership functions of the fuzzy sets and a rule-base containing a number of fuzzy IF-THEN rules), Inference Engine (derives a conclusion from the facts and rules contained in the knowledge base), and the Defuzzification Interface (maps a fuzzy set to a crisp set) [8], [11] and [12].

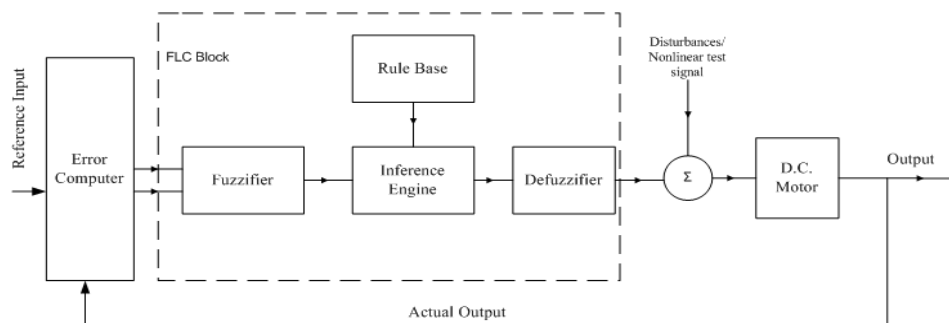


Fig.5. Block Diagram of Fuzzy Logic Control System

The FLC created had 2 inputs: position error $e(t)$ designated as **E** and change in position error $\Delta e(t)$ represented by **DE** and a single output given as control input to the DC servo motor driver denoted as **CI**. The linguistic variables have been defined as: {NB, NM, NS, ZR, PS, PM, PB}, where the initials correspond to negative big, negative medium, negative small, zero, positive small, positive medium and positive big respectively. A total of $7 \times 7 = 49$ Mamdani type rules were used to represent changes in the two inputs. Triangular fuzzy sets, being simple and easy to calculate, have been selected for both input (fuzzification) and output (defuzzification). The span of non-normalized **E** is $[-7 \ 7]$, **DE** is $[-7 \ 7]$ and that of **CI** is $[-7 \ 7]$ as shown in Fig.6 together with the constant Sugeno output fuzzy set. The defuzzification method used is the centroid method because it can be easily implemented in digital control systems and requires less computation time. For given set of inputs to the FLC, appropriate rule(s) must be fired from the rule base and defuzzified to give the control input signal. Table 2 summarizes the control rules which map the fuzzy inputs to fuzzy output.

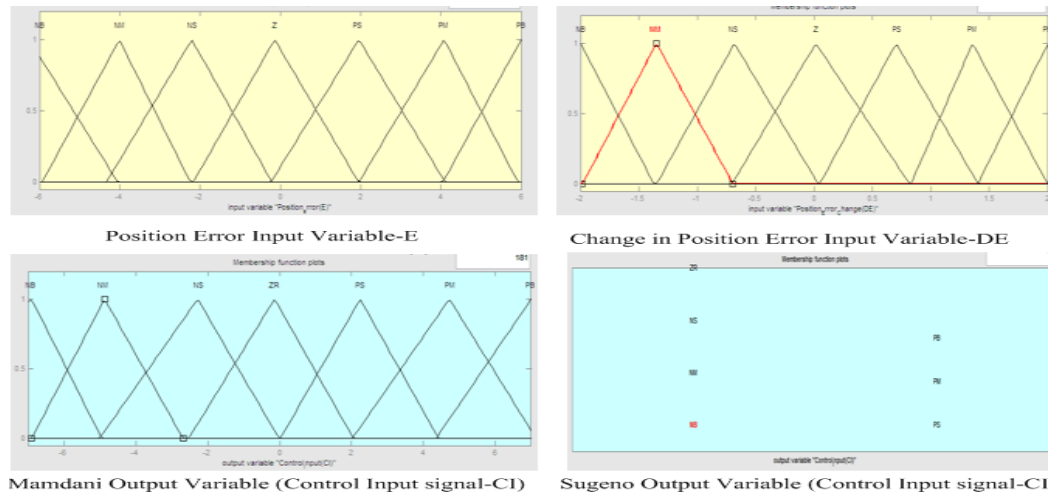


Fig.6.The membership functions of E, CE and CI in FIS File

Table 2 7x7 Rules Fuzzy Associative Memory (FAM)

DE \ E	NB	NM	NS	ZR	PS	PM	PB
NB	NB	NB	NB	NM	NS	NS	ZR
NM	NB	NB	NM	NS	NS	ZR	PS
NS	NB	NM	NS	NS	ZR	PS	PM
ZR	NM	NM	NS	ZR	PS	PM	PM
PS	NM	NS	ZR	PS	PS	PM	PB
PM	NS	ZR	PS	PS	PM	PB	PB
PB	ZR	PS	PS	PM	PB	PB	PB

V. Neuro-Fuzzy System Controller Design

5.1 Adaptive Neuro-Fuzzy Principle

Adaptive Neuro-Fuzzy Inference System (ANFIS) technique is used as a teaching method for Sugeno-type fuzzy systems and was proposed by Jang [12]. In applying ANFIS, the number and type of fuzzy system membership functions (MFs) has to be specified by user. The method is more efficient in the sense that it combines the advantages of FLC and NN approach in order to construct a nonlinear self-tuning controller. In addition, since the rules are in linguistic format, intermediate results can be analyzed and interpreted easily. ANFIS method is also viewed by many researchers, as a hybrid method, which consists of two parts: gradient method that is applied to calculation of input membership function parameters and least square method which is applied to calculation of output function parameters [13]. There are three constraints of using MATLAB ANFIS method as follows: only Sugeno-type decision method is available, there can be only one output and lastly, defuzzification method is weighted mean value. A typical architecture of ANFIS control structure is shown in Fig.7, in which a circle indicates a fixed node, whereas a square indicates an adaptive node [14].

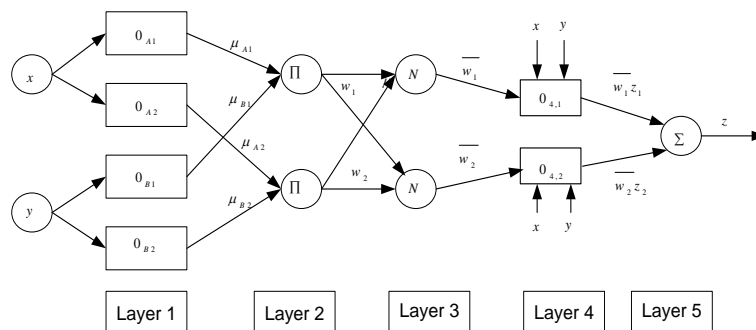


Fig.7. Corresponding ANFIS architecture

For simplicity, two inputs x , y and one output z are considered. For a first order Sugeno fuzzy model with two fuzzy if-then rules, a common rule set can be expressed as in (13) and (14):

Rule 1: if x is A_1 and y is B_1 , then

$$Z_1 = p_1x + q_1y + r_1 \tag{13}$$

Rule 2: if x is A_2 and y is B_2 , then

$$Z_2 = p_2x + q_2y + r_2 \tag{14}$$

where A_i and B_i are the fuzzy sets in the antecedent, and p_i , q_i and r_i are the design parameters that are determined during the training process. The ANFIS network structure, is made up of a set of units (and connections) organized into five connected network layers, 1 to 5 as shown in Fig.7. The detailed functions performed by each layer is explained in [12] and [13] and are summarized in the next section under ANFIS design. The ANFIS structure can be tuned automatically by the hybrid learning algorithm using a least-square estimation (for output membership functions) and a back propagation algorithm (for output and input membership functions) [14], [15] and [16].

5.2 Adaptive Neuro-Fuzzy System Controller Design

The ANFIS controller generates change in reference drive voltage based on position error E and derivative in position error (speed error) DE defined by (15) and (16):

$$\text{Error (E)} = (\text{Desired position} - \text{Actual position}) \tag{15}$$

$$\text{Error Change (DE)} = (\text{Current Error} - \text{Previous Error}) \tag{16}$$

In this study, a first order Sugeno-type fuzzy inference is used for ANFIS and the typical fuzzy rule takes the form of (17):

If E is A_i and DE is B_i , then,

$$z = f(E, DE) \tag{17}$$

where A_i and B_i are fuzzy sets in the antecedent and $z = f(E, DE)$ is a crisp function in the consequent. The significances of each layer and operation of the 2 input -1 output ANFIS structure [12], [13] considered are:

Layer 1: This layer (the fuzzification layer) enables the entry of raw data or crisp inputs from the target system into ANFIS. It is composed of a number of computing nodes whose activation functions are fuzzy logic membership functions, taken here as triangular. Each adaptive node generates the membership grades called fuzzy spaces for the input vectors $A_i, i = 1, \dots, n$ and $B_i, i = 1, \dots, n$ where n is the number of membership functions of the inputs (E and DE) chosen as $n = 7$, defined by (18). The degree to which the inputs lie within the fuzzy space is given a value normalized between -7 and 7.

$$O^1_{A_i} = \mu_{A_i}(E), \quad O^1_{B_i} = \mu_{B_i}(DE), \quad i = 1, \dots, n \tag{18}$$

Layer 2: Is the rule layer where each node is fixed. Once the locations of inputs in the fuzzy spaces are identified, the product of the degrees to which the inputs satisfy the membership functions is found. This product is called the firing strength of a rule whose output is given by (19). In other words, it selects the minimum value of the inputs. In this layer, the total number of Takagi-Sugeno rules used is 49.

$$O^2_i = W_i = \min(\mu_{A_i}(E), \mu_{B_i}(DE)) \tag{19}$$

Layer 3: This is the normalization layer in which the ratio of each rule's firing strength is calculated with respect to the sum of the firing strengths of all the rules. Each node in this layer is fixed. The i^{th} node output is the i^{th} input activation level divided by the sum of all the activation levels of the other inputs, as given in (20):

$$O^3_i = \bar{W}_i = \frac{W_i}{\sum_{i=1}^n W_i} \tag{20}$$

Layer 4: In layer 4, the defuzzification layer, the output of each node is the weighted consequent value. Adaptive node i in this layer calculates the contribution of i^{th} rule towards the overall output, with the following node function in (21):

$$O^4_i = \bar{W}_i Z_i = \bar{W}_i (p_i E + q_i DE + r_i) \tag{21}$$

Layer 5: Layer 5 is the summation layer and its output, which is the sum of all the outputs of layer 4, gives the overall output for the respective inputs within the fuzzy space. The single fixed node in this layer computes the overall output as the sum of each rule's contribution given in (22):

$$O^s_i = \sum_{i=1}^2 \overline{W}_i Z_i = \frac{W_1 Z_1 + W_2 Z_2}{W_1 + W_2} \tag{22}$$

Before the ANFIS system can be used for prediction, the parameters of the rules are determined by first generating an initial FIS where first random values are assigned to the parameters. Next, an optimization scheme is applied to determine the best values of the parameters that would supply rules to idealistically model the target system. After training, the rules remain so that when new input data is presented to the model, the rules provide a corresponding reasonable output [13]. The optimization technique used is a hybrid learning algorithm that minimizes the error between the ANFIS model and the real system using training data from the target system to generate signals that propagate backwards and forwards and update the parameters [14]. The parameters to be trained are A_i , and B_i of the premise parameters and p_i , q_i and r_i of the consequent parameters. The ANFIS Editor GUI window includes four distinct areas to support a typical workflow. It enables realization of the following tasks:

- Loading, Plotting, and Clearing the Data
- Generating or Loading the Initial FIS Structure
- Training the FIS
- Validating the Trained FIS [15]

For generating FIS structure, the triangular MF is used for the two input variables and output type is linear. The number of MFs for the input variables E and DE is 7 each hence the number of rules is $7*7 = 49$. Fig.8 shows the membership functions for E and DE before training. Fig.9 shows the generated ANFIS structure used for the DC servo motor antenna positioning controller design. It has been used 80% of the generated datasets for training the ANFIS system model and 10% each as testing and checking data. Hybrid learning algorithm was used for training the generated FIS with the number of epochs as 100 and tolerance of 0.01. It is clear from [16] that the triangular MF is specified by two parameters. Therefore, the ANFIS used here contains a total of 371 fitting parameters, of which 28 ($2*7 + 2*7 = 28$) are the premise parameters and 343 ($7*49 = 343$) are the consequent parameters.

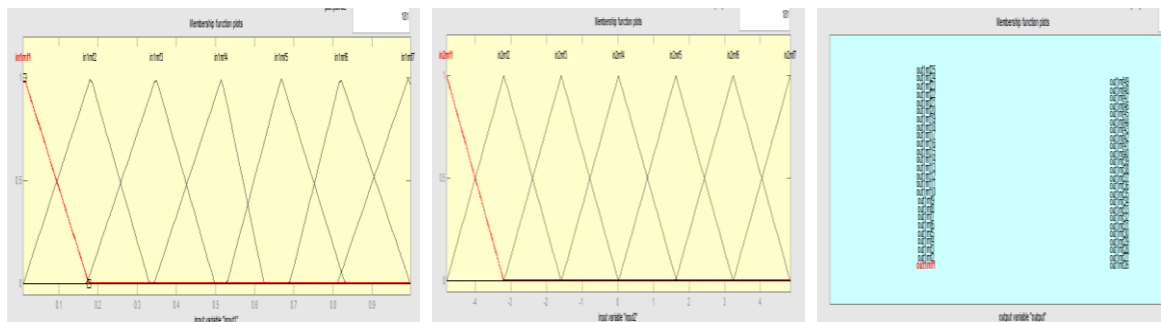


Fig.8. Input and Output Membership functions before training

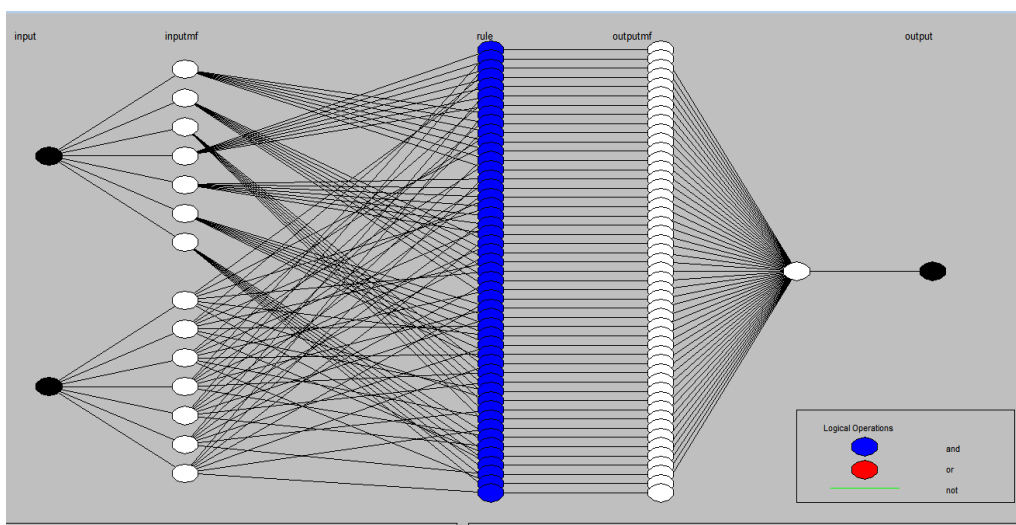


Fig.9. 5-Layer ANFIS model structure with 2 inputs & 1 output

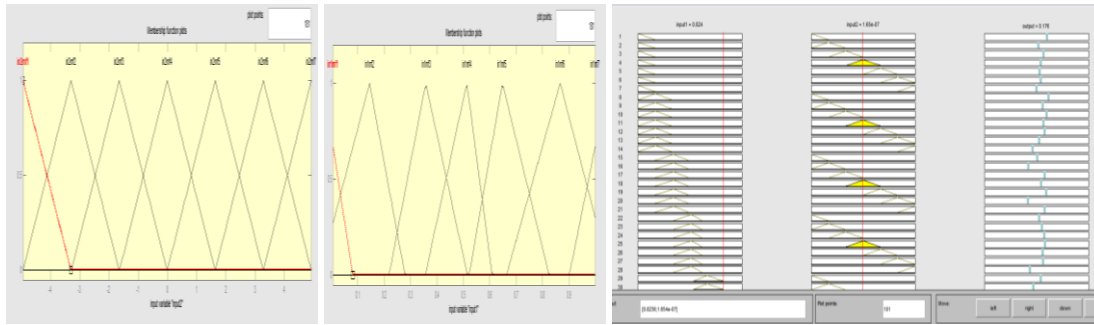


Fig.10. Membership functions for E and DE and Section of ANFIS Rule viewer after training

Fig.10 shows optimized membership function for E and DE and the rule viewer after training. Thus an adaptive network that has exactly the same function as a Sugeno fuzzy model has been constructed. In Fig.11, the trained ANFIS controller with testing and checking data plots are shown. Surface plots showing relationships between input and output parameters before and after training are given in Fig.12 to help view the control surface.

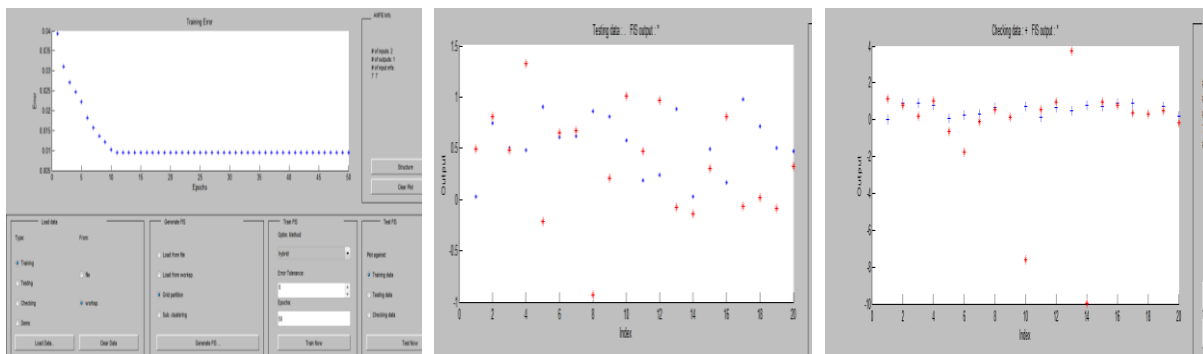


Fig. 11. Training, Testing and Checking ANFIS controller

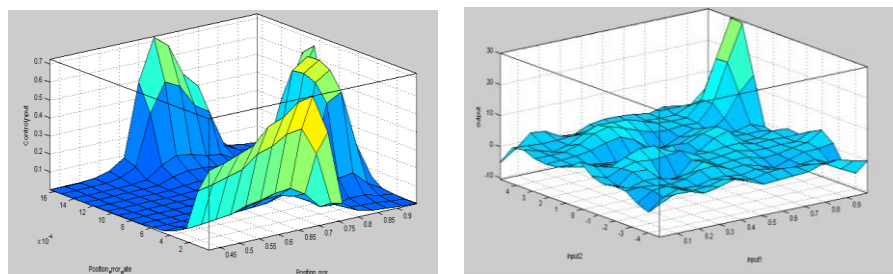


Fig.12 3D-Surface plot before training (left) and after training (right)

The steps for MATLAB simulation are summarized as:

1. The SIMULINK model layout created (Fig.13) is opened and fuzzy editor invoked by typing *fuzzy*.
2. The Mamdani FLC and Takagi-Sugeno (TS) ANFIS based *.fis* files (named *mamdani2*, *anfis3*) are loaded to the fuzzy editor and then exported to workspace.
3. Each of the 49-rule based Mamdani FLC and TS ANFIS models are run and the simulation stopped after 10seconds.
4. The created MATLAB m-file is executed to generate training, testing and checking data used in ratio 8:1:1.
5. The variables x =position error, y =error rate, z =output and three other variables $[a_1, a_2, a_3]$ are selected to hold data from step (4) in excel fields each made up of the corresponding three columns.
6. The *anfis* editor is opened by typing command *anfisedit* and the number and type of Membership functions are specified.
7. Variable $[a_1]$ is loaded from workspace, grid partitioning is selected as FIS generation method, number of epochs is set (between 40 and 100) and tolerance 0.01. Training is done using hybrid algorithm method.
8. The *.fis* ANFIS file is exported to workspace and the SIMULINK control model (ANFIS model) run. This trained the Adaptive Neural-Network.
9. The testing and checking data sets are loaded, in turns, to the ANFIS *.fis* file using $[a_2, a_3]$ from workspace and performance is tested and validated respectively.

- The performance characteristics are observed and analyzed and the design process is stopped once design constraints/objectives have been fulfilled.

VI. Simulink Model

The SIMULINK model for the designed NFSC including PID and FLC controllers created in MATLAB software for conducting the simulations is shown in Fig. 13. The FLC controller and MATLAB code were used to generate the required training, testing and checking data. The inputs of the controller are taken as reference position represented by a step input signal and the actual position obtained from the actual output signal feedback. The output is the driving voltage to the motor driver. The prototype circuit diagram constructed within Proteus 8 Professional Simulator environment for software implementation is as shown in Fig.14.

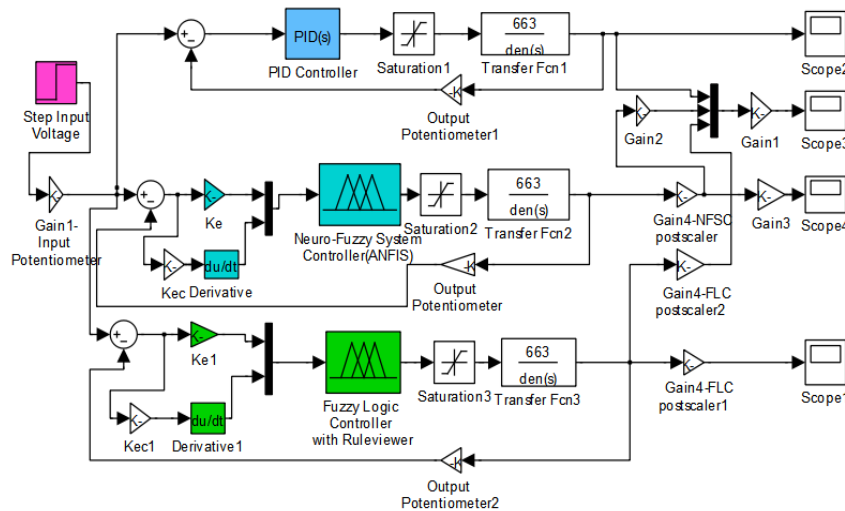


Fig.13. PID, FLC and NFSC (ANFIS) Simulink Model Layout

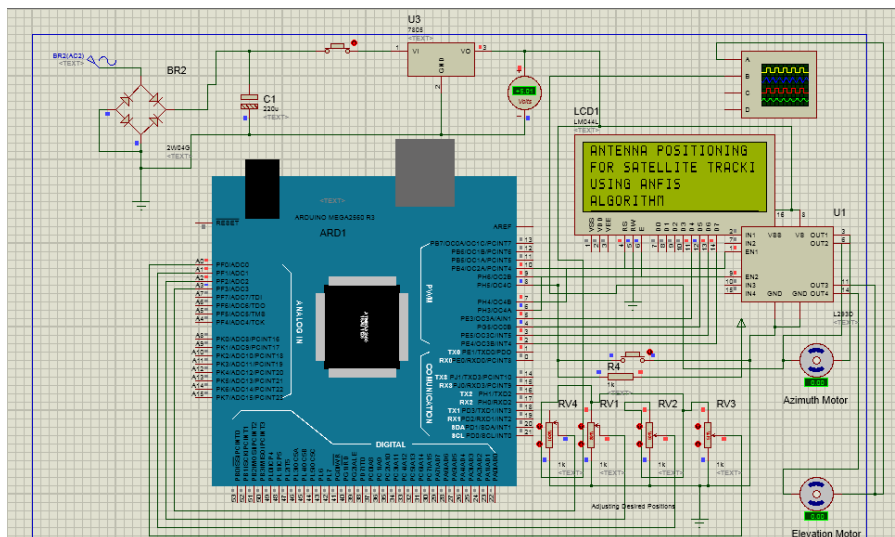


Fig.14. Circuit Diagram for antenna position control with NFSC (ANFIS)

VII. Results and Discussion

The proposed NFSC system has been studied by simulation. Further, it has been compared with the conventional PID controller and FLC in order to evaluate its performance in presence of saturation nonlinearity and to validate the accuracy of the design. In Fig. 15, Fig. 16 and Fig. 17, the outputs of each controller i.e. PID, FLC and ANFIS (before training) are respectively shown while Fig.18 shows the NFSC output after training has been conducted. From this, it was observed that the ANFIS controller output was the best in terms of faster rise time, settling time and amplitude stabilization. Fig.19 shows system response with PID controller in which it is seen that the response is not that good owing to high overshoot and increased settling time.

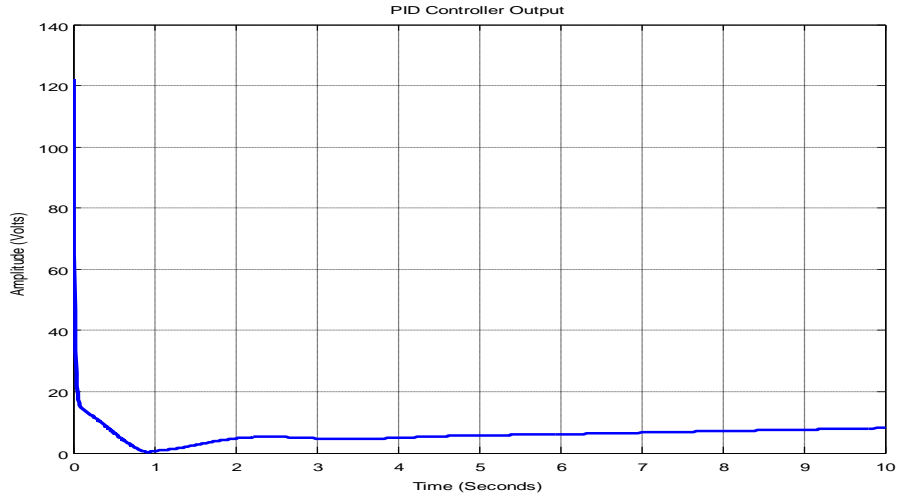


Fig.15. PID Controller Output

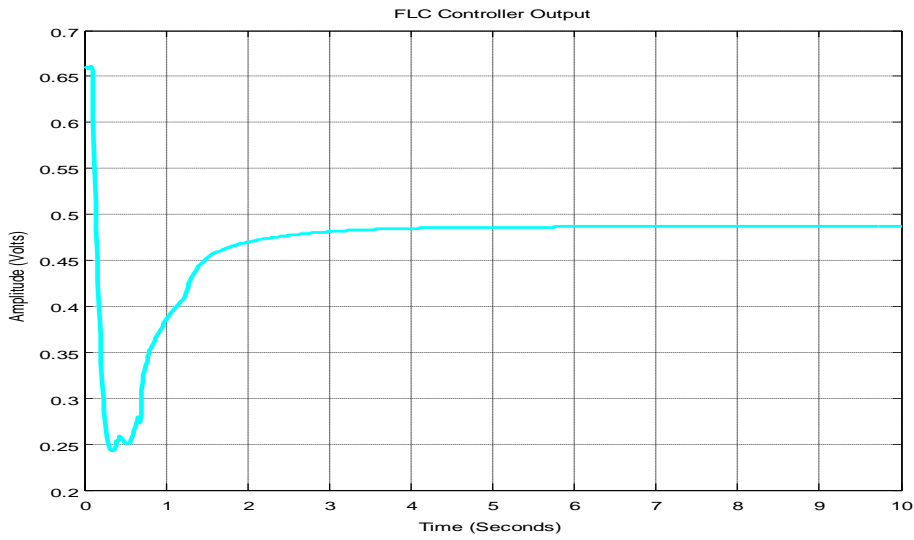


Fig. 16. FLC Controller Output

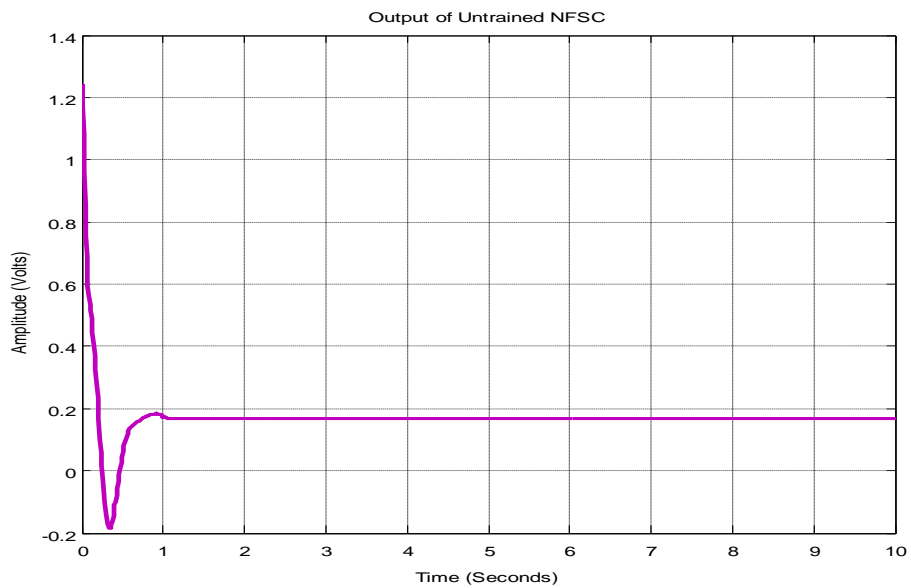


Fig. 17. NFSC (ANFIS) Controller Output before training

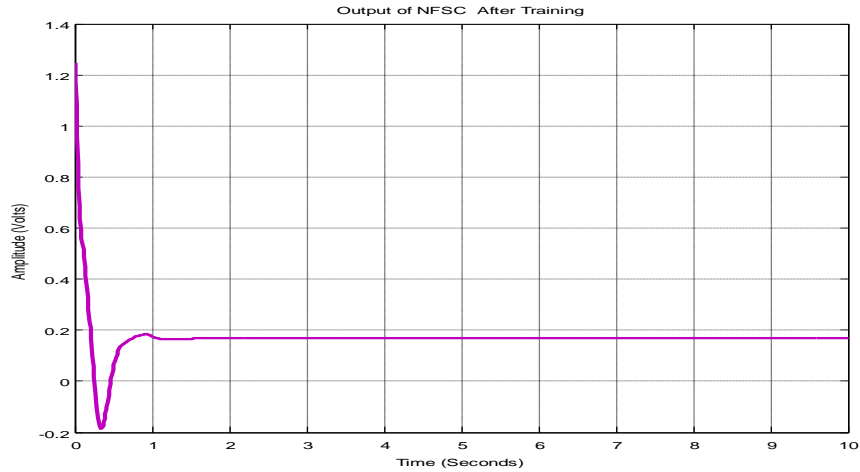


Fig. 18. NFSC (ANFIS) Controller Output after training

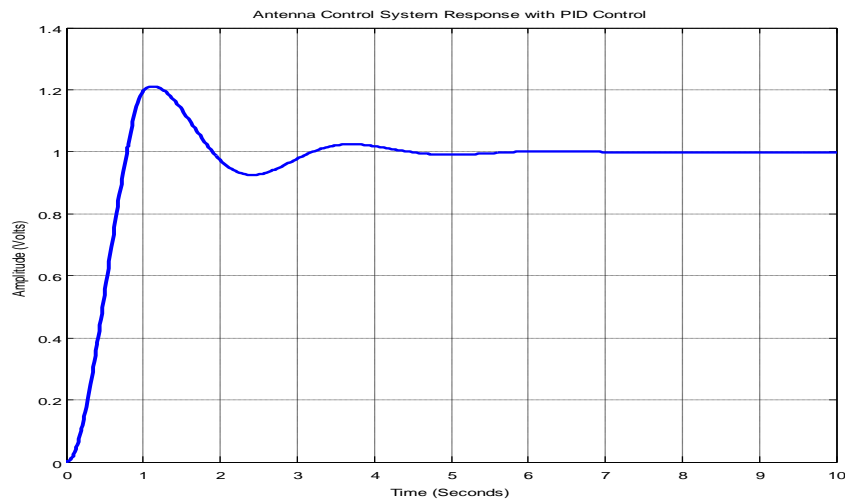


Fig.19 System step response with PID controller

Fig. 20 shows the Mamdani type FLC response to step input which shows no overshoot but the rise time is comparatively longer. Fig.21 gives system response by using the NFSC prior to training while Fig.22 shows NFSC response after training. Fig. 23 shows a combined plot to compare system responses using PID, FLC and NFSC (before training) and in Fig. 24, after training the NFSC.

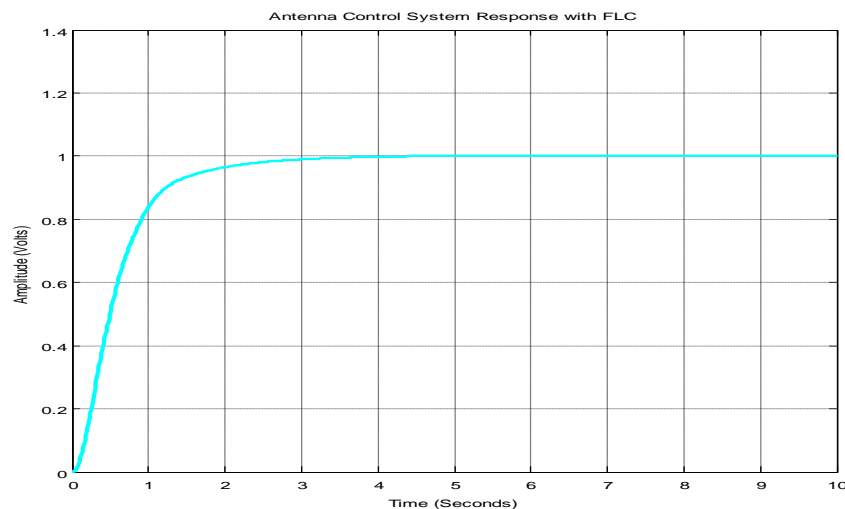


Fig.20. Mamdani Type FLC step input response.

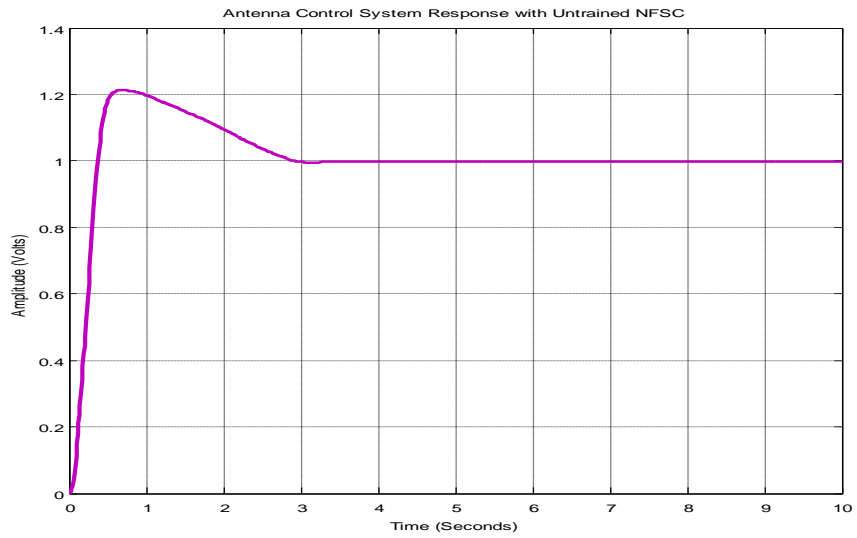


Fig.21 System response using NFSC before training.

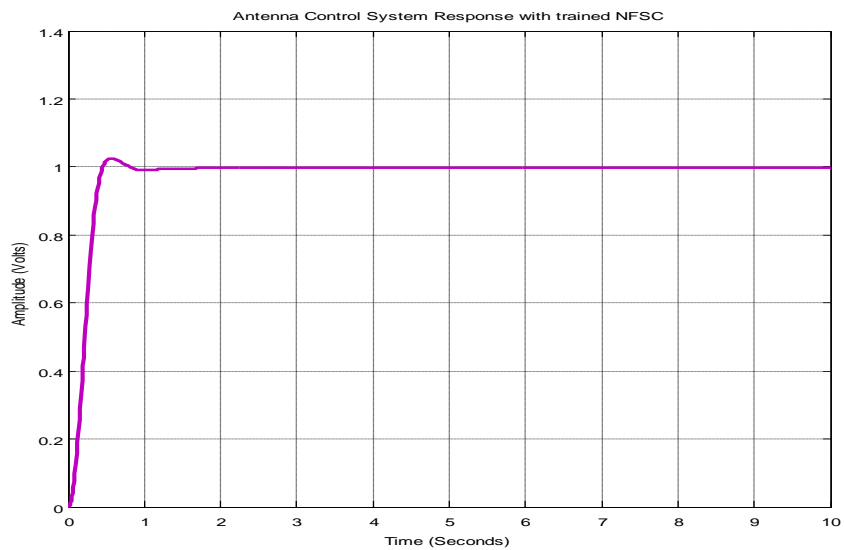


Fig.22 System response using NFSC after training.

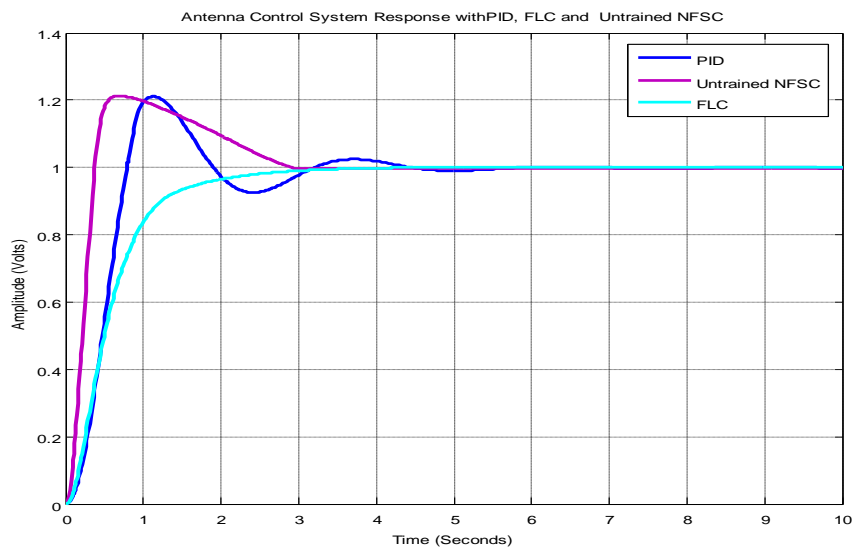


Fig. 23. Response of PID, FLC and NFSC (before training)

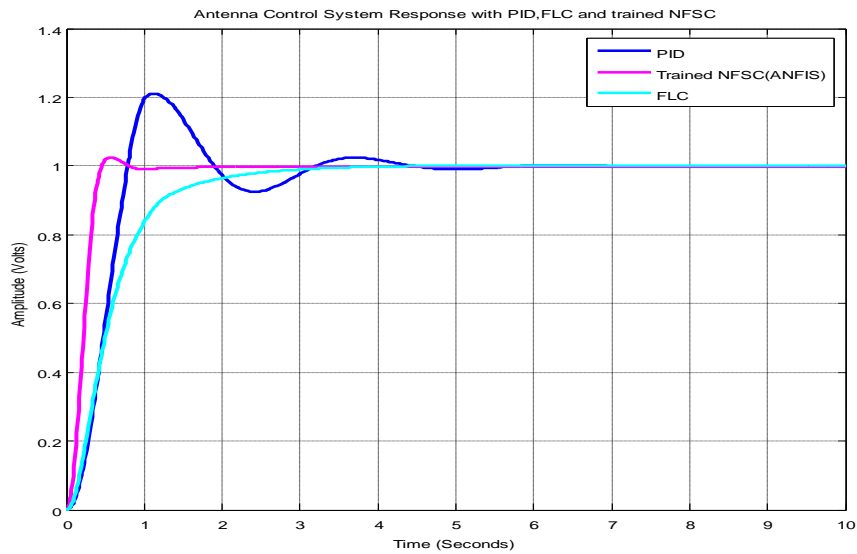


Fig. 24. PID, FLC and NFSC response after training NFSC

Table 3: Simulation Results with Various Step Inputs

Step Input	PID			FLC			NFSC-Before Training		
	t_r	t_s	M_p	t_r	t_s	M_p	t_r	t_s	M_p
(V)	(s)	(s)	(%)	(s)	(s)	(%)	(s)	(s)	(%)
1	0.7	4.5	22.0	2.9	3.1	0.0	0.2	2.6	20.0
2	0.8	4.2	21.0	3.1	3.1	0.0	0.2	2.7	20.0
3	0.8	4.4	23.0	3.2	3.2	0.0	0.3	2.5	19.0
4	0.7	4.8	22.0	3.0	3.3	0.0	0.2	2.7	19.0
5	0.8	4.9	22.0	3.2	3.4	0.0	0.3	2.6	20.0
6	0.8	4.6	23.0	3.2	3.2	0.0	0.2	2.7	20.0
Av	0.8	4.6	22.2	3.1	3.2	0.0	0.2	2.6	20.0
NFSC Response After Training									
Step Input	t_r	t_s	M_p						
(V)	(s)	(s)	(%)						
1	0.2	0.8	4.0						
2	0.2	0.7	4.0						
3	0.3	0.7	3.5						
4	0.2	0.8	4.0						
5	0.3	0.7	4.0						
6	0.2	0.7	4.5						
Av	0.2	0.7	4.0						
Key: t_r -rise time, t_s -settling time and M_p -percentage overshoot, Av=Average									

In addition, the system response was analyzed using the NFSC, FLC and PID controllers when different values of reference step input signal corresponding to the desired positions were applied. The corresponding results were studied and the estimated time response characteristics before training the NFSC and after training have been summarized in Table 3.

By examining the results, it is seen that NFSC provides response with faster settling times and minimized rise time albeit the presence of saturation nonlinearity. From the average values in Table 3, NFSC recorded the best average performance. The rise time is 3.1sec for Mamdani FLC model, 0.2sec for the Takagi-Sugeno NFSC model both prior to and after training and 0.8sec for the PID control. The maximum overshoot is 22.2% with PID, none with the FLC and 20.0% with NFSC (before training) and only 4.0% after training due its learning ability. The settling time is 3.2 sec. for FLC, 2.6 sec and 0.7sec for the NFSC before and after training respectively and 4.6 sec for the PID control. It can be understood from this study that while using NFSC the system tends to approach and settle at the desired position in the fastest times (0.2sec and 0.7sec respectively) as compared to both PID and FLC controller. This means that the rate at which the error between desired antenna position and its actual position can be performed by the NFSC algorithm within the least amount of time and thus allowing for continued establishment of a direct line of sight for quality communication between the parabolic dish and the selected satellite. The PID controller apart from its high overshoot takes longer to reach

steady state. Although no overshoot was registered with FLC, it lacks the learning ability offered by the NFSC which has been acquired at the expense of a 4% overshoot introduced in steady state value but which is still within the desired limit of less than 10%.

VIII. Conclusion

The objective of this paper to design NFSC Controller for DC servomotor has been achieved. This was verified through simulated output responses of the system to step input signal which satisfied the design criteria. The FLC alone increased both the rise and settling times by a few seconds but decreased the overshoot significantly to zero degrees. Using PID, FLC and ANFIS approach it is seen that ANFIS provides the best performance in terms of stabilizing the system response in a much shorter time with minimum rise time and overshoot. Therefore, the objectives were met in both design and software simulations.

Future Work

Further work may focus on the hardware application of the developed NFSC algorithm to achieve automatic and real time satellite tracking with parabolic antenna.

References

- [1] J.K. Kim, K.R. Cho and C. S. Jang, "Fuzzy control of data link antenna control system for moving vehicles," ICCAS, June 2005.
- [2] T.V. Hoi, N. T. Xuan and B. G. Duong, "Satellite tracking control system using Fuzzy PID controller," VNU Journal of Science: Mathematics and Physics, vol. 31, no. 1, pp. 36-46, 2015.
- [3] M. N. Soltani, R. Zamanabadi and R. Wisniewski, "Reliable control of ship-mounted satellite tracking antenna," IEEE Transactions on Control Systems Technology, p. 99, 2010.
- [4] L. Xuan, J. Estrada and J. Di Giacomandrea, "Antenna azimuth position control system analysis and controller implementation," Design Problem, July 2009.
- [5] M. Ahmed, S.B. Mohd Noor, M. K. Hassan and A.B.Che Soh, "A Review of Strategies for Parabolic Antenna Control," Australian Journal of Basic and Applied Sciences, vol. 8(7), pp. 135-148, May 2014.
- [6] N.S. Nise, "Control System Engineering," John Wiley & Sons, 6th Edition, 2011.
- [7] L.A. Aloo, P.K.Kihato and S.I. Kamau, "DC Servomotor-based Antenna Positioning Control System Design using Hybrid PID-LQR Controller", European International Journal of Science and Technology, Vol. 5 No.2, March, 2016.
- [8] A.K. Pandey, "Speed control of DC servomotor by fuzzy controller," International Journal of Scientific and Technology Research, vol.1, 2012.
- [9] A. Mehmet and T. Ismail, "Motion controller design for the speed control of DC servo motor," International Journal of Applied Mathematics and Informatics, vol. 1, no. 4, pp. 131, 2007.
- [10] P.K. Kihato and J.N. Nderu, "Direct Neuro-Fuzzy Approach to Motion Tuning of Servomotors," Kenya Society of Electrical and Electronic Engineers (KSEEE) and Japan Society of Applied Electromagnetic and Mechanics (JSAEM), 2011.
- [11] R.S.Burns, "Advanced Control Engineering," Butterworth-Heinemann, OXFORD, pp.325-373, 2001.
- [12] J.S.R.Jang, "ANFIS: Adaptive Network Based Fuzzy Inference System", IEEE, Transactions on system, Man & cybernetics, vol. 23, 1993.
- [13] B.A.A. Omar, A.Y.M. Haikal and F.F.G. Areed, "Design adaptive neuro-fuzzy speed controller for an electro-mechanical system," Ain Shams Engineering Journal (ASEJ) vol. 2 pp. 99-107, 2011.
- [14] J.P.Lilja, P.M. Kristian and M.H.C Alfonso, "NEURO-FUZZY CONTROL," Soft Computing, University of Iceland, 2005.
- [15] S.Ghadri, S.Javadi and F.Farokhi, "Decreasing Starting Current for Separate Excited DC Motor using ANFIS Controller" International Journal of Smart Electrical Engineering, Vol.2, No.2, Spring 2013.
- [16] H.T. Nguyen, N. R. Prasad, C. L. Walker and E. A. Walker, "A First Course in Fuzzy and Neural control," by Chapman & Hall/CRC, 2003.