Design and Implementation of Neuro-Fuzzy Algorithm for Fractional Order Processes

Major Project Report

Submitted in partial fulfillment of the requirements
For the Degree of

Master of Technology
In
Instrumentation And Control
(Control And Automation)

Prepared By:

Khushbu Vasoya
(13MICC27)

Instrumentation And Control Engineering Section
Department Of Electrical Engineering
Institute Of Technology
Nirma University
May-2015

Design and Implementation of Neuro-Fuzzy Algorithm for Fractional Order Processes

Project Report

Submitted in partial fulfillment of the requirements
For the Degree of

Master of Technology
In
Instrument And Control
(Control And Automation)

Prepared By:
Khushbu Vasoya

Guided by:
Dr. Dipak Adhyaru
And
Prof. Sandip Mehta

Instrumentation And Control Engineering Section
Department Of Electrical Engineering
Institute Of Technology
Nirma University
Ahmedabad 382 481
May-2015

Declaration

This is to certify that

i) The thesis comprises my original work towards the degree of Master of Technology in Communication Engineering at Nirma University and has not been submitted elsewhere for a degree.

ii) Due acknowledgement has been made in the text to all other material used

Vasoya Khushbu

Annexure

Undertaking for originality of the work

I, **Vasoya khushbu J., Roll No.13MICC27**, give undertaking that the Major Project entitled **Design and implementation of Neuro Fuzzy Algorithm for Fractional order Process** submitted by me, towards the partial fullfilment of the requirements for the degree of **Master of Technology in Control and Automation** of Nirma University, Ahmedabad, is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

____

Signature of Student

Date:______________

Place: Ahmedabad

Endorsed by:

____________________	____________________
Dr.D.M Adhyaru	Prof.Sandip Mehta

Certificate

This is to certify that the Major Project entitled **"Design and Implementation of Neuro-fuzzy Algorithm for Fractional Order Processes"** submitted by **Khushbu Vasoya (13MICC27)** , towards the partial fulfilment of the requirements for the degree of Master of Technology (Electrical Engineering) in Control and Automation Engineering of Nirma University, Ahmedabad is the record of work carried out by her under our supervision and guidance. The work submitted has in our opinion reached a level required for being accepted for examination. The results embodied in this major project work to the best of our knowledge have not been submitted to any other University or Institution for award of any degree or diploma.

Date: Place:Ahmedabad

Co-Guide:

Dr. Dipak Adhyaru

(Section Head,IC)

Guide:

Prof.Sandip Mehta

(Assistant Professor,IC)

Programme Coordinator:

Prof.J.B.patel

(Programme Coordinator,IC)

Head of Department:

Dr.P.N.Tekwani

(Professor,EE)

Director:

Dr.K.Kotecha

(Director,IT,NU)

Acknowledgement

The beatitude, bliss and euphoria that accompany the successful completion of any task would not be complete without the expression of appreciation of simple virtues to the people who made it possible. So with reverence, veneration and honor I acknowledge all those whose guidance and encouragement has made me successful in my Project up to this level.

I take this opportunity to express my sincere gratitude to my project guide **Dr.Dipak Adhyaru and Professor Sandip Mehta**, who has been kind enough to spare his valuable time. His guidance and motivation conceived a direction in me and made this project successful.

I would like to express my gratitude and sincere thanks to Dr.P.N.Tekwani Head of Electrical Engineering Department and Prof. J.B.Patel Coordinator M.Tech Instrumentation And Control Engineering program for allowing me to undertake this thesis work and for his guidelines during the review process.

I am very grateful to Dr.K Kotecha, the Director, Institute of Technology without whose support my research and this dissertation would not have been possible.

I thank one and all who helped me directly or indirectly in my project work.

- Vasoya Khushbu J.

(13MICC27)

Abstract

In this project, design and implementation of Neuro-Fuzzy Algorithm for Fractional Order Controller System has been carried out. First the simulation study for the fractional order transfer function using different control algorithm has been done. Fractional order system was controlled by ANFIS and PID controller. Comparative analysis between ANFIS and PID has been carried out in this project. From this analysis it is observed that ANFIS gives better performance.The fractional order PI algorithm has been tested for the sensorless speed control of 3-phase induction motor.A comparison has been made for the fractional order PI algorithm against the integer order PI algorithm.

Model reference adaptive control of fractional order system using ANFIS and CANFIS has been discussed. ANFIS (Adaptive Neuro Fuzzy Inference System) has been used for the MISO (Multi input single output) system and CANFIS (Coactive Neuro Fuzzy Inference System) is used for MIMO (Multi input Multi output) system. In this paper fractional order MIT rules are used to generate adaption law for MRAC. Tuning for the ANFIS and the CANFIS has been carried out by the fractional order MRAC tuning rule. Fractional order MIT rule has been used to control fractional order system. The comparative analysis between CANFIS and ANFIS has been shown in this paper. It has been shown that the fractional order MRAC can be replaced by an appropriate neural network either in form of ANFIS or CANFIS.

Nomenclature

S^{α} = Fractional order differentiator

yi, ui = Function of time

$H(.)G(.)$ = Combination law of Fractional order

$G(s)$ = Fractional PID transfer function

$C(s)$ = PID transfer function

$\mu A(x)$ = Membership function

O_i^l = Output of layer, l is layer number, i is node

θ = Updating parameter

$J(\theta)$ = Cost function

e = Error

γ = adoption gain

Kp = Proportional gain

Ki = Integral gain

Kd = Derivative gain

KI^{λ} = Fractional integration

KD^{μ} = fractional derivative

λ, μ = Fractional order

d^{α} = Fractional derivation

A = Fuzzy set

$\Gamma(x)$ = Eulers Gamma function

$ANFIS$ = Adaptive Neuro Fuzzy Inference System

$MRAC$ = Model Reference Adaptive Controller

$MRAS$ = Model Reference Adaptive System

$CANFIS$ = Co-active Neuro Fuzzy Inference System

MF = Membership Function

$LTIsystem$ = Linear Time Invariant system

PID = Proportional Derivative Integral

MIT = Massachusetts Institute of Technology

$SISO$ = Single Input Single Output

$MISO$ = Multi Input Single Output

$MIMO$ = Multi Input Multi Output

$NumRule$ = Number of Rule

N_{Inputs} = Number of Inputs

$NumOutVar$ = Number of Output Variables

LE = Learning Enable

Contents

List of Figures

List of Tables

Chapter 1

Literature Survey

1.1 ANFIS

Literature Survey 1:

TITLE: ANFIS: Adaptive Network Based Fuzzy Inference System

AUTHOR: Jyh-Shing, Roger Jang

INFERENCE:

ANFIS (Adaptive Neuro Fuzzy Inference System) is fuzzy inference system which is used for the framework of adaptive network. In adaptive network parameters are updated based on given training data sets. Parameters are updated using hybrid learning algorithm. In this paper ANFIS (adaptive Neuro fuzzy inference system) is engaged to nonlinear function.

A) ADAPTIVE NETWORK: THE ARCHITECTURE AND LEARNING ALGORITHMS

Adaptive network is comprised of node and directional link. Function of Directional link is to connect node. Like every system in which output is depends on parameters of input, here also adaptive network output is depends on node parameters and to minimize the error, parameters are change by learning rules. Parameters are updated to achieve desired input-output mapping. Parameters are updated with use of hybrid learning gradient descent method.

B) HYBRID LEARNING RULE: PATTERN (ON LINE) LEARNING

We have the Hybrid learning (ON Line) for the system where parameters are updated i.e. system in changing characteristics this learning pattern can be useful. In ON line learning gradient descent should be based on Ep i.e. error between targeted component and actual output is squared instead of E i.e. simply difference of targeted component and output one.

1.2 Fractional Order

Literature Survey 1:

TITLE: ANFIS $PI^{\lambda}D^{\mu}$ controller design and comparison for overhead cranes

AUTHOR: Hseyin Arpaca and Faruk zgvenb

INFERENCE: Overhead cranes are used to transport heavy loading from one place to another place. As the cost of transportation is very high and material is to be kept at precise place and with safe landing with heavy load such type of cranes with different capacity is required. To obtain such precise result the non-linear system like crane requires optimum controlling system.

Fuzzy model is comprised of fuzzy sets, membership functions and fuzzy rules. Fuzzy sets, membership function and rules are important to achieve the desired performance. In ANFIS (Adaptive Neuro fuzzy inference system) Neural network is used with adaptive capability to reach at desired output.

In this paper author proposed new systematic method to control over head crane. By considering the non-linear element of crane, design of controller for crane with Adaptive neuro-fuzzy inference system (ANFIS) is designed. Also, Overhead crane system simulation is done with PID and fractional order PID controller. By comparing the performance of ANFIS, PID and fractional order PID it is observed that simulation of ANFIS Controller shows better result in terms of Trolley positions, swing angles and control forces.

Literature Survey 2:

TITLE: Using Fractional Order Adjustment Rules and Fractional Order Reference Models in Model-Reference Adaptive Control

AUTHOR: B. M. VINAGRE, I. PETR and I. PODLUBNY , Y. Q. CHEN

INFERENCE:

The main approach for Adaptive control is Model Reference Adaptive System (MRAS).Working principle of model reference adaptive controller is design adaptive controller that adjust the parameters of controller in such a way that output of actual model tracks the output of reference model which has same reference inputs. MIT rule is used in MRAC. In MIT rule cost function is defined by error between actual model and reference model output. Parameters of controller are changed in such a way, that minimize the error and Actual model can track the output of reference model.

In this paper use of fractional order calculus in MRAC (Model Reference Adaptive Controller) is done. Introduction of Fractional order calculus in MRAC is done in two different ways:

1))Use of fractional order MIT rule

2)employment of fractional order reference model.

In MIT rule the rate of change of parameters are depends on Adaption Gain γ , While in Fractional order MIT Rule the rate of change of parameters are depends on Adaption Gain as well as on Order of derivation. The equation for Fractional order MIT rule is as below:

$$\frac{d\theta}{dt} = -\gamma\frac{dj}{d\theta} = -\gamma e\frac{de}{d\theta} \tag{1.1}$$

Second way of introduction of Fractional order in MRAC is Employment of Fractional order reference model.

Consider a system described by the transfer function

$$G(s) = \frac{1}{S+1} \tag{1.2}$$

The equation for fractional order reference model is as given below:

$$y_m = \frac{1}{S^{0.25} + 1} \mu_c \tag{1.3}$$

1.3 CANFIS: Coactive Neuro Fuzzy Inference System

Coactive Neuro Fuzzy Inference System (CANFIS) is employed by multi input multi output. Where ANFIS is simply constructed as multi input single output. Input space of ANFIS and CANFIS are grouping in three flavors namely: Scatter, Grid, and ART. This input data partitioning is directly affects the approximation capacity and architecture of ANFIS and CANFIS.

In Grid partitioning input data is computationally complex as the input variable increases.

$$N_{Rule} = {N_{Input_Terms}}^{N_{Inputes}} \tag{1.4}$$

This is just another form of the widely known curse of dimensionality. To remove this problem Scatter partitioning is use. In Scatter partitioning number of input is equal to number of fuzzy rule.

$$N_{Rule} = N_{Input_Terms} \tag{1.5}$$

ART partitioning is much smatter way of input space partitioning. ART partitioning is much smarter than scatter partitioning. In ART type partitioning input data is categorized exclusively on area of input space. And each category is allocated with membership function.

Chapter 2

ANFIS: Adaptive Network Based Fuzzy Inference System

2.1 Introduction

Basically system modeling is done by conventional mathematical tools. Adaptive Neuro fuzzy inference system (ANFIS) contains fuzzy If-then rule, which can model the aspect of human knowledge and reasoning process. Initially, numerous applications of fuzzy if-then rule and modeling are analyzed by Takagi and Sugenos. ANFIS is required because we cannot convert human knowledge and its experiences in to fuzzy if-then rules and Standard method is required for tuning of membership function to obtain maximum performance.

In adaptive network there is only constraint is that it should be feed forward type otherwise there is no constraints. In Adaptive network selections of membership functions, numbers and type of parameters is important to acquire desired output or result [1]. Adaptive network is widely used in real world applications of decision making process, signal processing and todays control world.

ANFIS (Adaptive Neuro Fuzzy Inference System) gives the basis for constructing fuzzy if-then rules to obtain the required pairs of input-output with appropriate membership function.

2.2 Fuzzy set and Membership Function

Fuzzy Set is the group of anything i.e. number, color etc. and the classical set has the clear cut or we can say crisp boundary. In contrast to this,Fuzzy set A is the set which doesn't have the crisp boundary and that is in transition from belong to set to not belong to set is characterize by Membership function in certain defined set conditions [2].

Definition: Set of value that define Fuzzy set is called **Membership Function.**

$$A = \{(x, \mu A(x))|x \in X\} \tag{2.1}$$

Where,
X = Collection of object
x = object
A = Fuzzy set
$\mu A(x)$ = Membership function of fuzzy set A

2.2.1 Membership Function Formulation And Parametrization

Membership function of any set characterizes the fuzzy set. Membership function should cover each and every pair but it is not desirable to define the membership for each pair as it would increase the numbers and system will be more complicated and to overcome this more convenient way to define membership functions is to put MF in mathematical presentation. MFs derivatives also can be used for its parameters and inputs which help in fine tuning of Fuzzy inference system which leads to fine mapping of inputs and outputs. Paameterized functions mostly used to define MFs of one (i.e. MFs with single input) and two (i.e. MFs with two inputs) dimensions [2].

MFs of One Dimension

In this section we are going to define the classes of parameterized MFs of one dimension.

1) Triangular MFs

As the name its shelf specifies Triangular MFs are specified by 3 parameters i.e. a, b and c.

$$\mathbf{triangle}(x; a, b, c) = \begin{cases} 0, & x \leq a. \\ \frac{x-a}{b-a}, & a \leq x \leq b. \\ \frac{c-x}{c-b}, & b \leq x \leq c. \\ 0, & c \leq x. \end{cases}$$

For Example: A triangle (x; 20, 60, 80) can be define the triangular MF as below

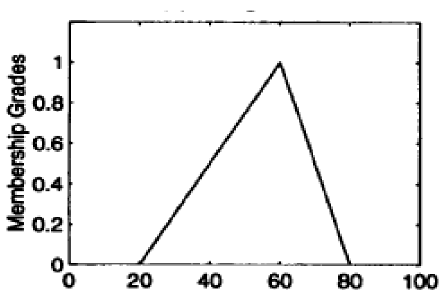

Figure 2.1: Triangle MF

2) Trapezoidal MFs

Trapezoidal MFs are specified by 4 parameters i.e. a, b, c and d.

$$\mathbf{trapezoid}(x; a, b, c, d) = \begin{cases} 0, & x \leq a. \\ \frac{x-a}{b-a}, & a \leq x \leq b. \\ 1, & b \leq x \leq c. \\ \frac{d-x}{d-c}, & c \leq x \leq d. \\ 0, & d \leq x. \end{cases}$$

For Example: A Trapezoidal (x; 10, 20, 60, 95) can be define the Trapezoidal MF as below

As every system has own benefits and limitations, here also both Triangular and Trapezoidal MFs are widely used in real time implementations because of its simple formula and its computing proficiency.

But due to its staight line composition, they are not be able to give smooth

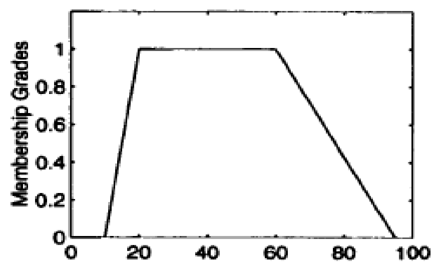

Figure 2.2: Trapezoidal MF

corner at the corner area and introduction of other Mfs is requires which are defined by smooth and non- linear functions.

3) Gaussian MFs

Gaussian MFs are defined and fully determined by two parameters i.e. c and σ.

$$Gaussian(x; c, \sigma) = e^{-\frac{1}{2}(\frac{x-c}{\sigma})^2} \tag{2.2}$$

Where, c : represents MFs Centre and σ determines the MFs Width.

For Example: A Gaussian (x; 50, 20) can be define the Gaussian MF as below

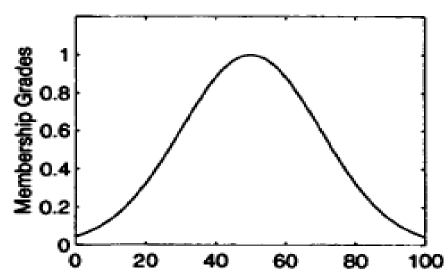

Figure 2.3: Gaussian MF

4) Generalized Bell shaped MFs

Generalized MF is also called as bell MF and this is specified by three parameters i.e. a, b and c.

$$Bell(x; a, b, c) = \frac{1}{1 + |\frac{x - c}{a}|^{2b}} \quad (2.3)$$

For Example: A Generalized (x; 20, 4, 50) can be define the Gaussian MF as below

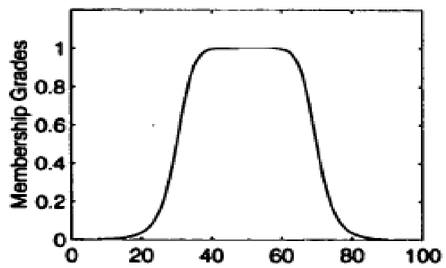

Figure 2.4: Bell MF

2.3 Fuzzy If-Then Rules And Fuzzy Inference System

2.3.1 Fuzzy if-then Rules

If A then B OR if X then Y is the form of conditional statement of Fuzzy if-then rule[2]. Where, A and B are linguistic label. This linguistic label is fuzzy set characterized by appropriate membership function. For example

IF pressure is hieh THEN velocity is less

Here high and small or less is linguistic label OR fuzzy set which is described by the membership function. Pressure is high is known as **premise part** and velocity is less is known as a **consequence part** . Takagi and Sugeons proposed different form of fuzzy if-then rule as stated below.

If velocity is high, then force= $k * (velocity)^2$

Here also high is linguistic label or fuzzy set is characterized by membership function. Here high is premise part and consequence part is given by non-fuzzy equation. Both types of fuzzy if-then rule which are stated above are used in modeling as well as in control. Using this fuzzy if-then rule we can easily capture the rule of thumb used by humans. Fuzzy if-then rule is most important part of fuzzy inference system.

2.3.2 Fuzzy inference systems

Because of multidisciplinary nature of fuzzy inference system it is known as fuzzy associative memories (FAM), fuzzy controller or fuzzy model. Fuzzy inference system comprises five components as shown in below figure 2.5

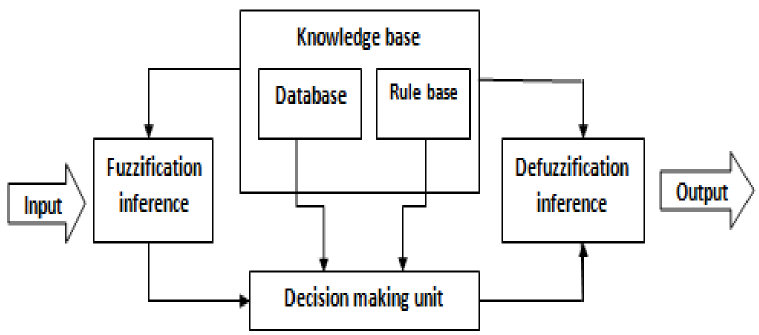

Figure 2.5: Fuzzy Inference System

Component of fuzzy inference system are as below:

1) Rule base: Fuzzy if-then rule is contained by rule base.

2) Data base: it defines membership function for fuzzy used in fuzzy rule.

3) Decision making unit: Fuzzy inference operation on fuzzy rule is performed by this block.

4) Fuzzification: It converts crisp value into fuzzy value.

5) Defuzzification: It converts fuzzy value into crisp value.

The combination of rule base and data base is known as Knowledge base.

2.3.3 Steps of Fuzzy reasoning

Fuzzy inference system performs fuzzy reasoning or inference operation on fuzzy if-then rule is explained below. 1) First of all input variables is compared with membership function to achieve membership value of every linguistic label used in fuzzy rule. This is known as Fuzzification.

2) Now in next step all membership value of linguistic label is combining through T-norm operation and after this we get weight or firing strength of each fuzzy rule.

3) Now take the ratio of every rules firing strength to sum of all rules firing strength.(Rules firing strength / Sum of all rules firing strength)

4) In this step convert fuzzy value into crisp value for this aggregate the qualified consequents. This process is called Defuzzification.

2.4 ANFIS:Adaptive Network Based Fuzzy Inference System

Adaptive network name suggest that network consist of node and directional link network with supervised learning capacity. In adaptive network all or part of node is adaptive means this nodes output depend on parameter of node. Different learning rule is available it specifies how this parameters are update to minimized error. For this network learning rule is based on gradient decent method is given by werbos in 1970s. Adaptive network every node performs node function this node function varies from node to node

2.4.1 ANFIS Architecture

In ANFIS architecture for easy understanding we assume that there is two input and one output. For example there are two rules contain in rule base.

Rule 1: If x is A1 and y is B1,then f1=p1x+q1y+r1

Rule 2: If x is A2 and y is B2,then f2=p2x+q2y+r2

ANFIS architecture is shown in fig. (). Node function of each layer is describe below:

Layer 1: Every node in this layer is give output with node function

$$O_i{}^l = \mu A_i(x) \tag{2.4}$$

Where, A is the linguistic label (small, high etc.) x is the input to the node i, in this node function. In this node function O is output or membership function of A.

Generally bell-shaped Membership Function is use is defined by,

$$\mu A(x) = \frac{1}{1 + |\frac{x-ci}{ai}|^{2b}} \tag{2.5}$$

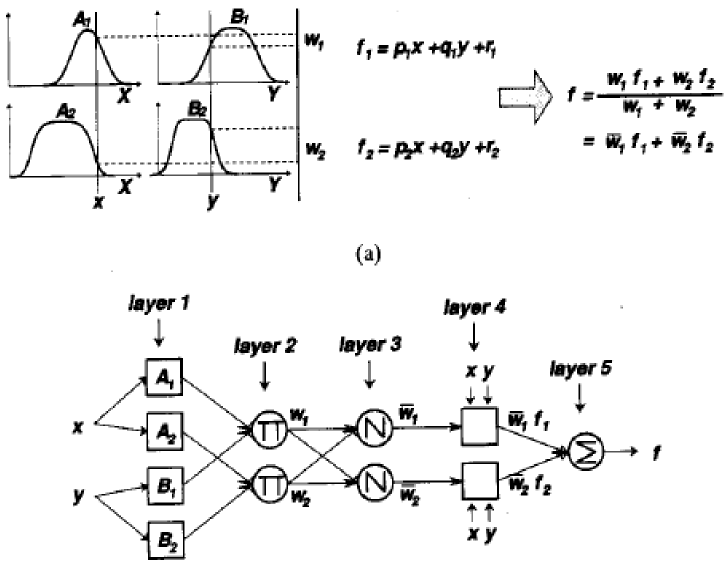

Figure 2.6: ANFIS Architecture

In this membership function (a_i, b, c_i) is the adaptive parameter set. This parameter we can change and we can change value and shape of membership function. Here we also can use trapezoidal and other membership function for this node.

Layer 2: In this layer every node perform fuzzy AND operation on incoming signal and produce product output with circle node

$$Wi = \mu Ai(x) \times \mu Bi(y) \quad where \;\; i = 1, 2.. \tag{2.6}$$

In this layer T-norm AND operation is use and generates Firing strength of each Fuzzy rule.

Layer 3: : In this layer calculates the ratio of i^th rules firing strength to the sum of all rules firing strength with circle node.

$$\overline{w_i} = \frac{w_i}{w_1 + w_2}, i = 1, 2.$$

Output of this layer is also called **normalized firing strength** .

Layer 4: In this layer every node is adaptive node is denoted by square node. Node function of this node is

$$O_i^4 = \overline{w_i} f_i = \overline{w_i}(p_i x + q_i y + r_i)$$

Where pi, qi, ri is parameter set or adaptive parameter. And this parameter is called consequent parameter.

Layer 5: In this layer only single node this is circle node. Node function of this node calculates overall output of network. Is denoted by,

$$O_1^5 = overall\ putput = \sum \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}$$

Adaptive network that constructed here is equivalent to Sugeno type model and network constructed here is not unique. in above network Layer 3 and 4 are combine and obtain new network. in extreme case network can combine in one node.

Chapter 3

Fractional Order System and Controller

3.1 Introduction of Fractional Order Calculus

Fractional calculus is introduced more than 300 year ago. The Euler and Lagrange in eighteenth century give theoretical contributions to fractional order. Systematic studies of fractional order calculus have been made by Liouville, Riemann, and Holmagren in start and middle of 19th century. Liouville expanded functions of exponential series and also defined nth (n is positive integer) order derivatives for this type of series. Also, Riemann given definition which is differ from earlier one which covers definite integral and this definition has wider use as its applicability in power series with non integer exponents.

Krug and Grunwald were the first to combine the result of Riemann and Liouville. By using the original sources of earlier work and taken derivative as a limit of different quotient and generated definite integral formulas for the nth order derivative, this was done by Grunwald. On other hand Krug, used cauchys formula of integration for ordinary derivatives and presented; Riemanns formula had to be interpreted as lower limit and Liouvilles definition had to be interpreted as lower limit as negative infinity.The interested reader can find good surveys of the history of fractional calculus in [3,4].

The application of Fractional calculus was applied by Abel in the year1823.

3.1.1 Definitions of Fractional Order Calculus

There are basic two definition are used in Fractional calculus for integration and differentiation. GL (Grunwald Letnikov) for differentiation and RL (Riemann-Liouville) for integral [5].

1) **Grunwald-Letnikov definition:** The GL definition is as stated below:

$$ {}_a\mathcal{D}_t^\alpha f(t) = \lim_{h\to 0} \frac{1}{h^\alpha} \sum_{j=0}^{\left[\frac{t-a}{h}\right]} (-1)^j \binom{\alpha}{j} f(t-jh) $$

The subscripts on both sides of D represent, respectively, the lower and upper bounds in the integration.

2) **Riemann-Liouville Fractional order differentiation:** The RL definition is as stated below:

$$ D_j^{-\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_a^t (t-\tau)^{\alpha-1} f(\tau) d(\tau) \tag{3.1} $$

Where,$0 < \alpha < 1$ and a is the initial time instance, often assumed to be zero, i.e., a $= 0$. $\Gamma(x)$ is Eulers Gamma function.

In the fractional order calculus, RL definition is generally used.

3)**Cauchys Definition:** Cauchys formula is as stated as below:

$$ D^\gamma = \frac{\Gamma(\gamma+1)}{2\pi j} \int_c \frac{f(\tau)}{(\tau-t)^{\gamma+1}} \tag{3.2} $$

3.2 Fractional Order System

Now a days fractional calculus have wide application in control word where controlled system and controller is fractional order. In this section analysis of fractional order system is carried out [5].

3.2.1 Models and Representations

Continuous time dynamic system is represented by following fractional order differential equation.

$$H(D^{\alpha_0,\alpha_1 \dots \alpha_m})(y_1, y_2, \dots y_l) = G(D^{\beta_0,\beta_1 \dots \beta_n})(u_1, u_2, \dots u_l) \tag{3.3}$$

Where $H(.), G(.)$ are combination law and yi, ui are function of time. For the linear time-invariant signal equation is given by

$$H(D^{\alpha_0,\alpha_1 \dots \alpha_n})y(t) = G(D^{\beta_0,\beta_1 \dots \beta_m})u(t) \tag{3.4}$$

With

$$H(D^{\alpha_0,\alpha_1 \dots \alpha_n}) = \sum_{k=0}^{n} a_k D^{\alpha_k}; G(D^{\beta_0,\beta_1 \dots \beta_m}) = \sum_{k=0}^{m} b_k D^{\beta_k} \tag{3.5}$$

where $ak, bk \in R$. Or, explicitly,

$$a_n D^{\alpha_n} y(t) + a_{n-1} D^{\alpha_{n-1}} y(t) + \dots + a_0 D^{\alpha_0} y(t) = b_m D^{\beta_m} y(t) + b_{m-1} D^{\beta_{m-1}} y(t) + \dots + b_0 D^{\beta_0} y(t) \tag{3.6}$$

In above equation derivation orders are integer multiples base order, α that is$\alpha k, \beta k = k\alpha, \alpha \in R+$,, this is commensurate system, and above equation (3.6) become,

$$\sum_{k=0}^{n} a_k D^{\alpha_k} y(t) = \sum_{k=0}^{m} b_k D^{\beta_k} u(t) \tag{3.7}$$

If in Eq.(3.7) $\alpha = 1/q, q \in Z+$, the system will be of rational-order.

Linear time invariant system can be classified as below:

- LTI Systems
  - Non-integer
    - Commensurate
      - Rational
      - Irrational
    - Non-commensurate
  - Integer

Discrete time transfer function can be written in fractional order is given below:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^{\beta_m} + b_{m-1} s^{\beta_{m-1}} + ... + b_0 s^{\beta_0}}{a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + ... + a_0 s^{\alpha_0}} \tag{3.8}$$

In the case of a commensurate-order system, the continuous-time transfer -function is given by

$$G(s) = \frac{\sum_{k=0}{}^{n} b_k (S^\alpha)^k}{\sum_{k=0}{}^{n} a_k (S^\alpha)^k} \tag{3.9}$$

3.2.2 Introduction Of Fractional Order Controller

Classical PID controller:

Classical PID controller is widely used in control word and industry. The PID (Proportional Derivative Integral) controller measures the error between output value and set point. Controller minimizes the error by adjusting manipulated variable. Equation for the classical PID controller is defined as below:

$$C(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s \tag{3.10}$$

	PROPOTIONAL	**INTEGRAL**	**DERIVATIVE**
1	1. To increase speed of response,	1. To eliminate steady state error 2. decrease relative stability	1.To increase relative stability, 2.sentivity to noise
2	1.There is permanent steady state error	1. Speed of responses decrease 2. As integral order increases oscillation increases.	1.Noisy 2.Introduce overshoot

Table 3.1: Advantage and Disadvantage of PID controller

PID controller control system and their effect on controlled system is given. Derivative action there is positive and negative effect. Positive effect is increased relative stability and in frequency domain is observed by $\pi/2$ phase lead introduce. Negative effect is increased sensitive to high frequency noise. Same for the integral action positive effect is it elimination of steady state error. Negative effect of integral action is decreased relative stability in frequency domain given as $\pi/2$ phase lag.

In high frequency signals we get more noise. If derivative controller is used then it will take action to for input and also considers noise as input, which it should not. This gives disturbed output and makes system unstable for that period of time. The output comes to normal after some time.

Derivative action introduce $\pi/2$ phase lead and integral introduce $\pi/2$ phase lag to overcome this fractional order controller is used.

In fractional order controller there are five parameters so there is more flexibility. The differential equation of fractional order controller is:

$$G(s) = k_p + \frac{k_i}{s^\lambda} + k_d s^\mu \tag{3.11}$$

Chapter 4

Model reference adaptive controller

Model reference adaptive controller

4.1 introductions

In 1950 research on adaptive control were started. Whitaker and his colleagues were able to develop model reference adaptive strategy in 1960 [7]. MIT rule in MRAC is used to update unknown parameters. Adaptive controller is required in control system to overcome the unpredictable parameters, deviations and uncertainties which are observed in the system. To design Advance controller adaptive control (MRAC) adaptive strategy is widely used. Adaptive controller is well suited for change in environment and disturbance. Adaptive control is standard part in textbook [8,9].

Classical controller may not be able to perform fine in online system due to the nonlinear actuator, environment condition and variation in disturbance change because of process dynamic nature. To eliminate above problem design of MRAC (Model Reference Adaptive Controller) for second order system with MIT rule for adaption mechanism is introduced [8]. In MIT rule cost function is defined by error between actual model and reference model. Parameters of controller changes in such a way which leads to minimization of error and actual model can track output of reference model. This controller gives better result but very sensitive to change in amplitude of reference signal. From simulation result it is observed that system become unstable, if adaption gain and amplitude of reference signal is sufficiently large.

4.2 Model reference adaptive controller

4.2.1 Principle of working

Working principle of model reference adaptive controller is to design adaptive controller that adjusts parameter of controller so that the output of actual model tracks, the output of reference model having same reference input.

4.3 Components

Component of MRAC is as given below:

Reference Model: Function of this model is to give ideal response of adaptive control system to the reference input.

Controller: Controller adjusts the adjustable parameter of system. It contains set of adjustable parameter. Here only one parameter can be adjusted i.e. is cost function of MIT rule. Value of depends on adaption gain.

Adjustment Mechanism: Parameter of controller is adjusted by this adjusting mechanism so that actual system gives the same response as of reference model response. In adjustable mechanism mathematical theory like MIT rule, Lyapunov theory and theory of augmented error can be used. MIT rule is used to Normalize algorithm and modified MIT rule.

Block diagram of MRAC is illustrated in fig.(4.1). In the figure reference model output ym(t) and actual model output y(t) and difference between them called error is denoted by e(t).

$$e(t) = y(t) - ym(t) \tag{4.1}$$

Design of a Model Reference Adaptive Controller Using Modified MIT Rule.

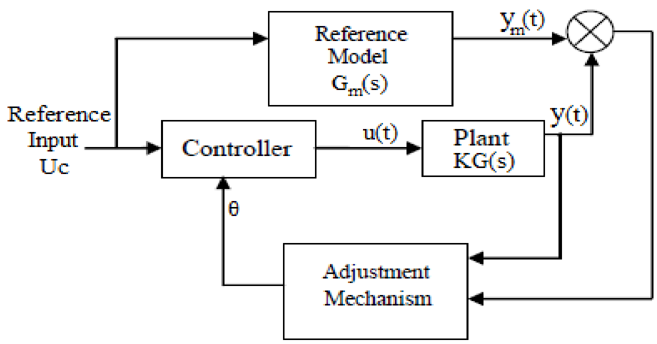

Figure 4.1: Model Reference Adaptive Controller

4.4 MIT Rule

In the year 1960s the researchers of Massachusetts Institute of Technology (MIT) found MIT rule and that was firstly used to design the auto pilot system for aircraft. To design a controller with MRAC, MIT rule can be used. So that design of controller can be possible that adjusts the parameter and system gives response like reference model. In MIT rule cost function is defined as

$$j(\theta) = e^2/2 \tag{4.2}$$

Here e is error between actual system and reference model and
θ is adjustable parameter.

Parameter θ is to be adjusted in the way which leads to minimization of cost function to zero. Because of above reason parameter θ is kept in negative direction. That is given by,

$$\frac{d\theta}{dt} = -\gamma\frac{dj}{d\theta} = -\gamma e\frac{de}{d\theta} \tag{4.3}$$

Here $\frac{de}{d\theta}$ is derivative term of the system is called sensitivity derivative. From this term, change in the error with reference to can be observed. is positive number and is called controllers adaption gain.

Let us assume that KG(s) is the transfer function and k is unknown parameter and G(s) is known transfer function of second order type. Design of controller is to be made such that actual process KG(s) can track the output of reference model Gm(s) = KoG(s) where, Ko is known parameter.

From Eq. (4.1),

$$E(s) = KG(s)U(s) - K_oG(s)Uc(s) \tag{4.4}$$

Defining a control law,

$$u(t) = \theta^* u_c \tag{4.5}$$

From Eqs. (4.5-4.6), and taking partial differentiation,

$$\frac{dE(s)}{d\theta} = KG(s)Uc(s) = \frac{K}{K_o}Ym(s) \tag{4.6}$$

From Eq.(4.4) and Eq.(4.7), we will get,

$$\frac{d\theta}{dt} = -\gamma e\frac{K}{Ko} = -\gamma' eYm \tag{4.7}$$

Eq.(4.8) will give us the law for adjusting the parameter and the Simulink model is shown in fig.(4.2) It has been seen from simulation results that the response of the plant depends upon the adaptation gain γ' . In some industrial plants, larger values of γ's can cause the instability of the system and selection of this parameter is very critical.

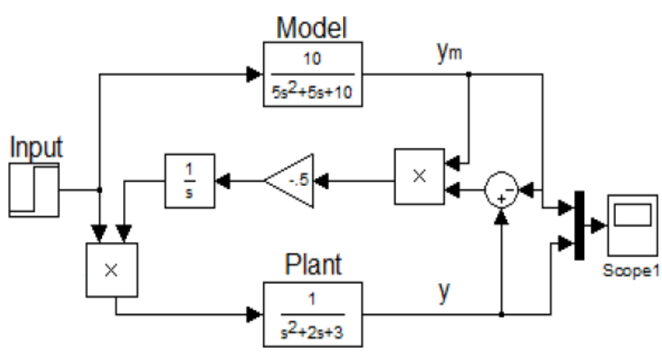

Figure 4.2: Simulink diagram of Model reference Adaptive Controller With MIT rule

There are many different algorithm is used in MIT rule. For example PI adjustment rule, sign-sign algorithm is used in communication system. Fractional order calculus is used in two ways in MRAC. Fractional order reference model is used in One and in other Fractional order MIT Rule.

4.5 Fractional Order Calculus in MRAC

There are two different ways to use Fractional order in MRAC, First one is use of Fractional order reference model with MRAC and other is Fractional order MIT Rule in MRAC.

4.5.1 FRACTIONAL ORDER ADJUSTMENT RULE

In above sectional studied about MIT rule from equation (4.3) can be observed that rate of change of parameter is depend on adaption gain . Now we introduce Fractional order in MRAC taking into consideration Fractional order property, there is probability that the rate of change of parameter is dependent on both adaption gain and derivation order . Fractional order MIT rule is given by,

$$\frac{d^\alpha \theta}{d^\alpha t} = -\gamma \frac{dj}{d\theta} = -\gamma e \frac{de}{d\theta} \tag{4.8}$$

Where α is order of Fractional order derivative. In other word rule can be expressed as follow,

$$\theta = \gamma I^\alpha [\frac{dj}{d\theta}] = \gamma I^\alpha [e \frac{de}{d\theta}]; \qquad I^\alpha \equiv D^{-\alpha} \tag{4.9}$$

For example, here we consider first order SISO system

$$\frac{dy}{dt} + ay = bu \tag{4.10}$$

Above equation is SISO system for controlled. Here y is output, u is input and a and b are unknown constant. Assume that reference model is

$$\frac{dym}{dt} + a_m y_m = b_m u_c \tag{4.11}$$

Here ym is output and Uc is input of reference model. Am and bm is known constants. For the perfect model controller is given by,

$$u(t) = \theta_1 u_c(t) - \theta_2 y(t) \tag{4.12}$$

Where

$$\theta_1 = \frac{b_m}{b}; \qquad \theta_2 = \frac{a_m - a}{b} \tag{4.13}$$

The equations for updating the controller parameters can be designed as

$$\frac{d^\alpha \theta_1}{dt^\alpha} = -\gamma(\frac{1}{p + a_m})u_c e \tag{4.14}$$

$$\frac{d^\alpha \theta_2}{dt^\alpha} = \gamma(\frac{1}{p + a_m})ye \tag{4.15}$$

where p = d/dt, and γ is the adaptation gain, a small positive real number. Equivalently, in frequency domain, can be written as

$$\theta_1 = -\frac{\gamma}{s^\alpha}(\frac{1}{s + a_m})u_c e \tag{4.16}$$

$$\theta_2 = \frac{\gamma}{s^\alpha}(\frac{1}{s + a_m})ye \tag{4.17}$$

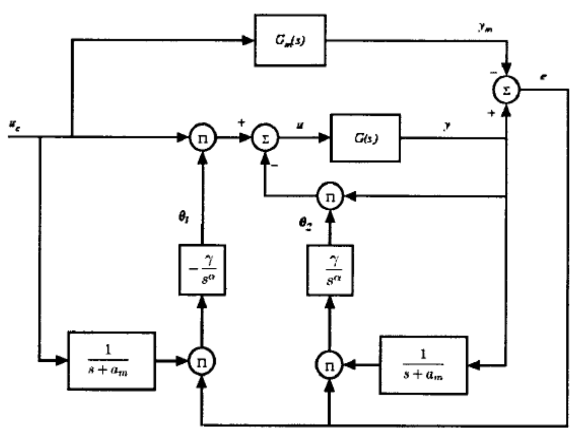

Figure 4.3: Simulink block diagram of simple MRAC

4.5.2 Fractional Order Reference Model

In this section introduce Fractional order system as reference model in MRAC. Fractional order system is introduced as reference model which is another modification to MRAC. FIR or second order dynamic systems are simplest reference model in simple MRAC problem. By using fractional order systems, reference model's set of candidates can be enlarged. Transient response of system made through MRAC can also be improved. Consider an example of system described by the transfer function:

$$G(s) = \frac{1}{s+1} \tag{4.18}$$

The adaptive scheme is used to adjust the feed forward gain in order to track the reference model output

$$y_m = \frac{1}{s^{0.025}+1} u_c \tag{4.19}$$

It would be difficult to track output of reference model even after significant time if $\alpha = 1$ is used as parameter adjusting rule. It would be easier if $\alpha \in (0, 1)$ is used with fractional reference model.

4.6 implementation

Sensor less speed control of induction motor drive is model with braking chopper for AC motor. Speed control of induction motor is carried out by speed estimated using terminal voltage and current based on MRAS (Model Reference Adaptive System). Induction motor is served by PWM voltage source inverter and speed is controlled by PI controller and Fractional PI controller. For both Fractional PI and PI controller obtain same output.

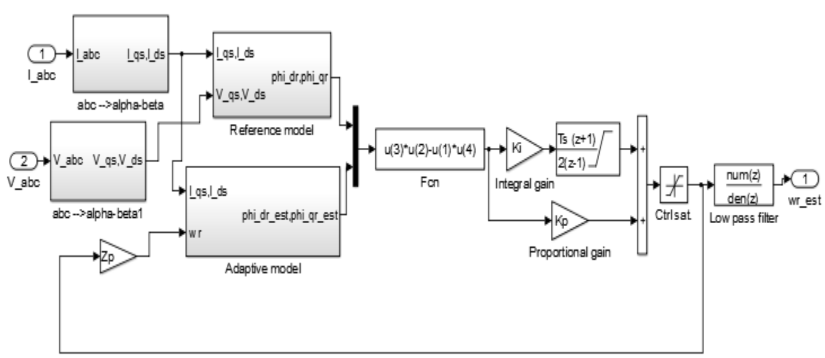

Figure 4.4: Sensor less speed control using PI controller

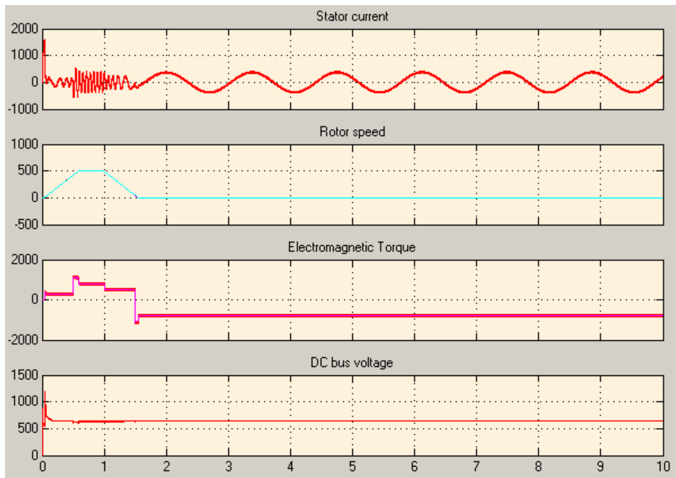

Figure 4.5: Simulation result of PI controller

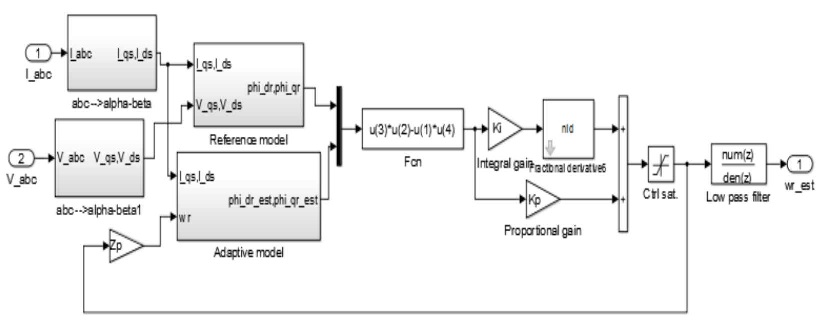

Figure 4.6: Sensor less speed control using Fractional PI controller

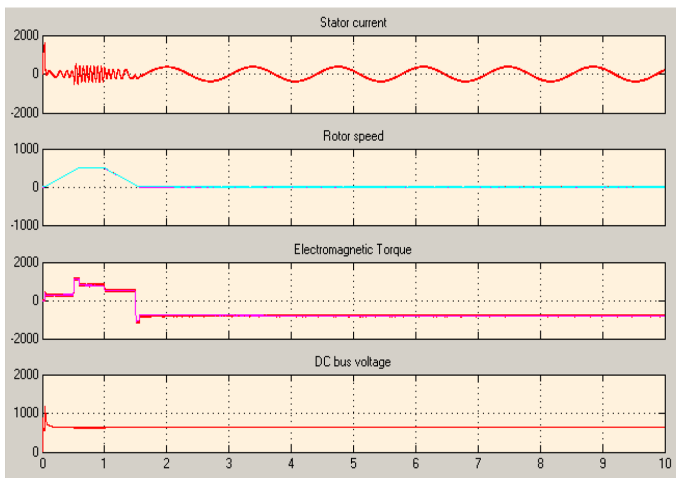

Figure 4.7: Simulation result of Fractional PI controller

Chapter 5

CANFIS: Co-active Neuro Fuzzy Inference System

5.1 Introduction

Coactive Neuro fuzzy inference system (CANFIS) is employed by multi input multi output [15]. Where ANFIS is simply constructed as multi input single output.

Input space of ANFIS and CANFIS are grouping in three flavors namely: Scatter, Grid, and ART. This input data partitioning is directly affects the approximation capacity and architecture of ANFIS and CANFIS.

5.1.1 Scatter, Grid, and ART

In Grid partitioning input data is computationally complex as the input variable increases. In Grid partitioning fuzzy rule increases as number of input increases. This can be indicated by equation (5.1).

$$N_{Rule} = {N_{Input_Terms}}^{N_{Inputes}} \tag{5.1}$$

This is just another form of the widely known curse of dimensionality. To remove this problem Scatter partitioning is use. In Scatter partitioning number of input is equal to number of fuzzy rule.

$$N_{Rule} = N_{Input_Terms} \tag{5.2}$$

ART partitioning is much smatter way of input space partitioning. ART (Adaptive Resonance Theory) algorithm is given by Carpenter et al. in 1991. ART type ANFIS model is similar to scatter type model. But ART partitioning is much smarter than scatter partitioning. In ART type partitioning input data is categories exclusively on area of input space. And each category allocated with membership function.

5.2 Description of ANFIS and CANFIS

Basically ANFIS and CANFIS consist of five layer. Each layre of ANFIS and CANFIS discus brifely up to next. Architecture of ANFIS is shown in figure (5.1).

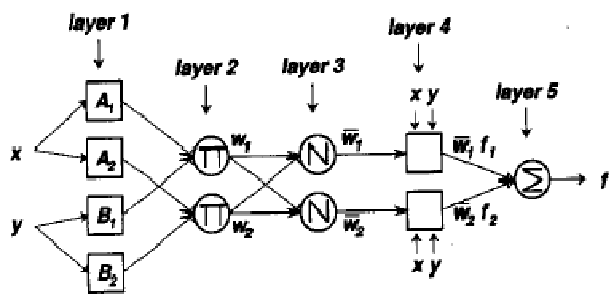

Figure 5.1: ANFIS architecture

Layer 1

In layer 1 ANFIS and CANFIS architecture fuzzyfication was taken place. Fuzzyfication means each input variable is compare to membership value on Premise part. This function is given by,

$$Out_{ij}^{(1)} = \mu_j(In_j^{(1)}) \tag{5.3}$$

Where $Out_{ij}^{(1)}$ is output of layer 1 and j^{th} linguistic variable of i^{th} input variable. In this Generally Gaussian membership function is use is denoted by,

$$\mu_j(x_i) = \frac{1}{1 + |\frac{x_i - C_{ij}}{a_{ij}}|^{b_{ij}}} \qquad i = 1, ..., NumInVars, j = 1, ..., NumInTerms \tag{5.4}$$

Where (a_{ij}, b_{ij}, C_{ij}) are adaptive parameter and by changing this parameter change shape and location of membership function.

Layer 2

In this layer membership value are combine by T-norm operator this is called fuzzy AND operation. So this node output give product of all input of this node.

$$Out_k^{(2)} = w_k = \prod_{i=1}^{N_{Inputs}} Out_{ij}^{(1)}$$

Output of this node give firing strength of fuzzy rule.

Layer 3

In this layer normalized operation is performed in this firing strength of each rule is divided by number of fuzzy rule. This is given by

$$Out_k^{(3)} = \overline{w_k} = \frac{Out_k^{(2)}}{\sum_{m=1}^{N_{Rules}} Out_m^{(2)}}, \quad k = 1, \ldots, \text{NumRules}.$$

Layer 4

In this layer output of previous layer is multiply by $a_{1k}, a_{2k}, ...a_{N_{Input}k}$ parameter. This is ANFIS liner parameter.

$$Out_k^{(4)} = \overline{w_k} f_k = \overline{w_k}(a_{1k} In_1^{(1)} + a_{2k} In_2^{(1)} + \ldots + a_{N_{Inputs}k} In_{N_{Inputs}}^{(1)} + a_{0k}) . \quad k = 1, \ldots, \text{NumRules}$$

Layer 5

This layer creates overall output of the ANFIS (MISO) network. Node function of this layer is given in below equation.

$$Out^{(5)} = \sum_{k=1}^{N_{Rules}} Out_k^{(4)} = \sum_{k=1}^{N_{Rules}} \overline{w_k} f_k = \frac{\sum_{k=1}^{N_{Rules}} w_k f_k}{\sum_{k=1}^{N_{Rules}} w_k} .$$

Now CANFIS architecture is same as ANFIS architecture, up to layer 1 to 3 are same but Layer 4 and layer 5 are different to ANFIS architecture.

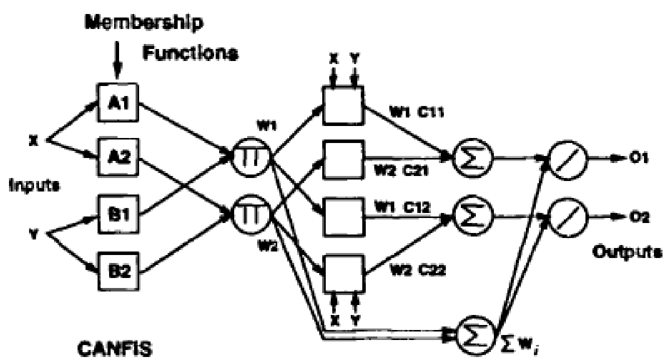

Figure 5.2: CANFIS architecture

Layer 4

In such a system, the output of the K^{th} fuzzy rule that influences the m^{th} network output is written as:

$$Out_{km}^{(4)} = \overline{w_k} f_k^m = \overline{w_k}(a_{1k}^m In_1^{(1)} + a_{2k}^m In_2^{(1)} + \ldots + a_{N_{Inputs}k}^m In_{N_{Inputs}}^{(1)} + a_{0k}^m),$$
$$k = 1, \ldots, \text{NumRules}, \; m = 1, \ldots, \text{NumOutVars}.$$

The parameters are the consequent parameters of the CANFIS system that represent the contribution of the k-th rule to the m-th output.

Layer 5

The m-th output of the network is computed as the algebraic sum of the m^{th} nodes inputs:

$$Out_m^{(5)} = \sum_{k=1}^{N_{Rules}} Out_{km}^{(4)} = \sum_{k=1}^{N_{Rules}} \overline{w_k} f_k^m , \quad m = 1,..., \text{NumOutVars}.$$

Chapter 6

Simulation Results

In this project implementation of Neuro fuzzy algorithm for fractional order processes was done. Fractional order system has wide application in control word because it is a Dynamic system that can be modeled by Fractional order Differential equation. Fractional order systems are useful in irregular behaviour of system. Simulation of Fractional order system using different control algorithm like PID, ANFIS has been carried out. Figure (6.1) shows block diagram of Fractional order system controlled by ANFIS and PID controller. Simulation result shows in figure (6.2). From the figure it is observed that by controlling Fractional order system using ANFIS and PID 0.03 steady state error is obtained in ANFIS simulation.

PID controlller:

$$C(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s \tag{6.1}$$

$K_p = 6$
$K_I = 3$
$K_D = 1$

MODEL 1:

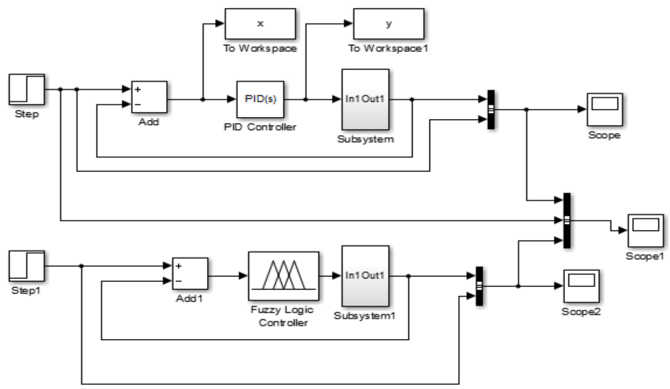

Figure 6.1: Simulation diagram for Fractional order system

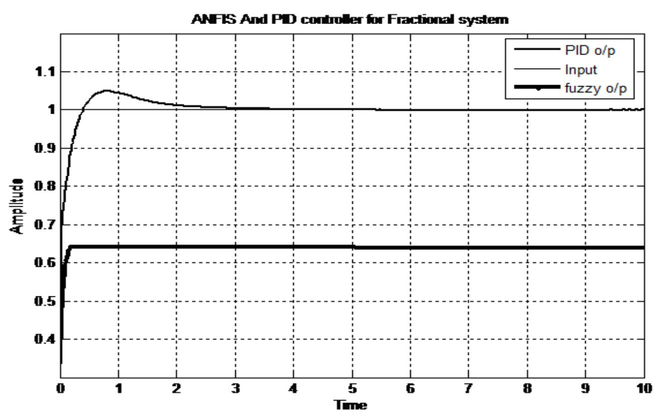

Figure 6.2: Simulation result of Fractional order system using PID and ANFIS

6.1 Membership Function of Fuzzy controller

As per precision and control requirement, usually 5 Membership functions are chosen. Range of input variable E, Ec, and output variable is [0, 1].

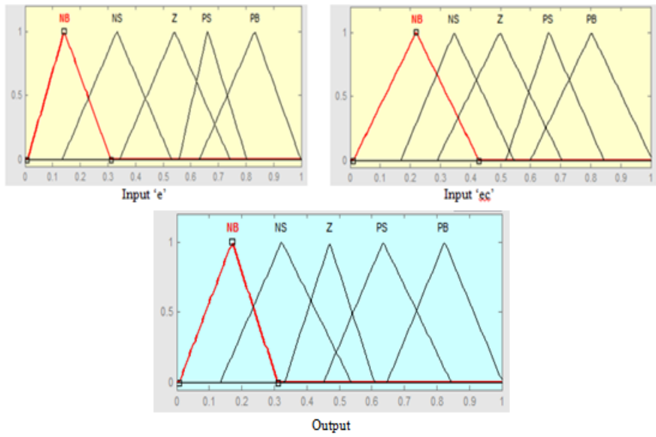

Figure 6.3: Membership Function of Linguistic Variable

6.2 Fuzzy controller Editor

To make Membership Function and fuzzy structure of fuzzy controller a new FIS file is made in MATLAB by FIS editor.

The Input and Output are given by Membership function using FIS editor. In FIS editor different types of Membership Functions are given. Rules are defined for input and output. The file is saved and run from Matlab code when simulation is to be done.

And method	Min
Or method	Max
Implication	Min
Aggregation	Max
Defuzzification	Centroid

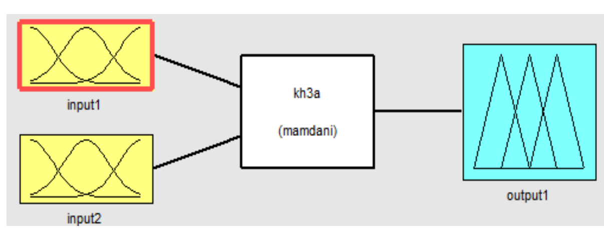

Figure 6.4: Fuzzy editor

6.3 Simulation Result

It is observed from above simulation diagram that there is constant 0.03 steady state error in ANFIS controller. Modification has been done in above diagram; Integration of error and ANFIS controller output was added. ANFIS network was trained by ANFISedit tool; in MATLAB Simulink. Two input error and derivation of error were used to train ANFIS network. After the network is trained, a FIS file is generated. This FIS file has been used to control fractional order system using ANFIS controller.

Figure 6.5 and 6.6 represents Simulation block diagram and simulation results respectively.

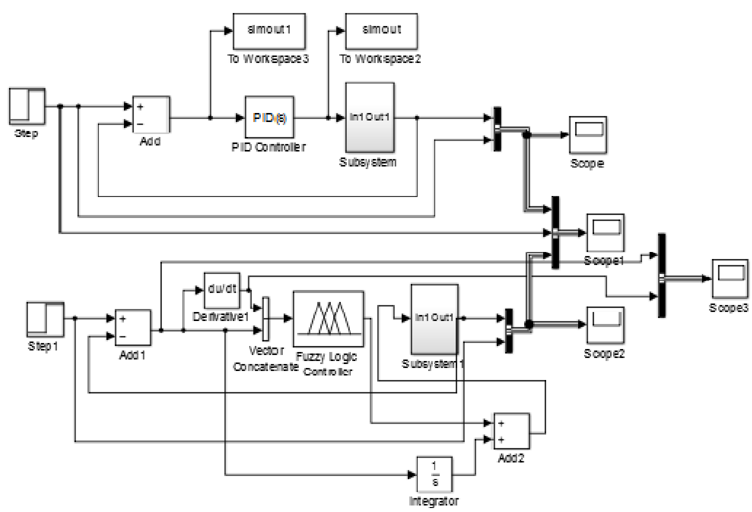

Figure 6.5: Simulation diagram of Fractional ordre system with integration

Simulation with above designed ANFIS controller having step input are shown in figure 6.6. It shows that ANFIS controller gives more overshoot than PID controller.

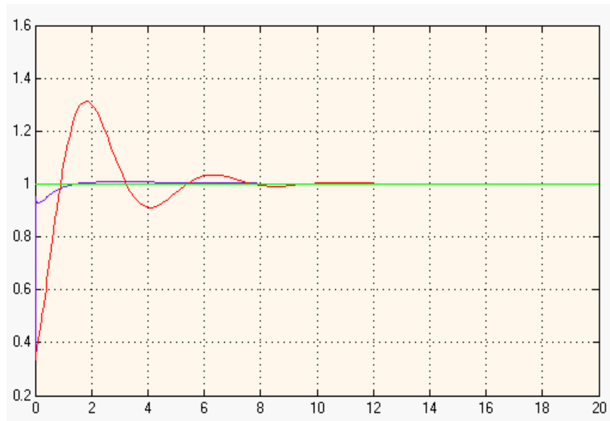

Figure 6.6: Result of figure 6.5

In above Simulation ANFIS network was trained by ANFIS edit tool. Now this ANFIS network has been trained by code and FIS file was generated. Than this FIS file was used to control system. Figure 6.7 shows Block diagram for simulation and figure 6.8 shows result of simulation result. From simulation result it is observed that overshoot was removed but ANFIS controller has more settling time than PID controller.

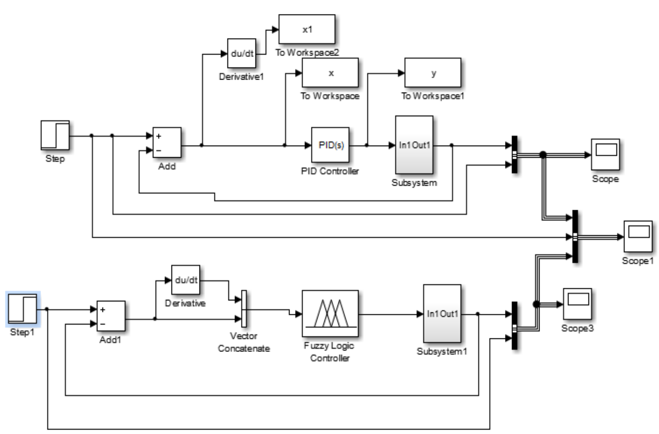

Figure 6.7: Simulation block diagram of Fractional order system

Next the simulation of system was done through ANFIS in which, to train

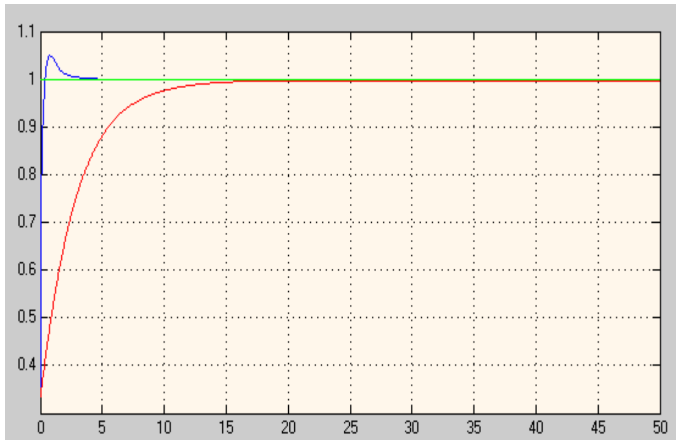

Figure 6.8: Simulation result of Fractional order system

the ANFIS controller using MATLAB commands like GENFIS, ANFIS and EVALFIS are used. Simulation result is shown in figure 6.8. It can be seen from figure 6.8 that settling time for ANFIS controller is very less as compare to PID controller. Settling time for PID is 8.9 and settling time for ANFIS is 6.6 is obtained from simulation. Hence the system can be controlled more efficient and effectively by ANFIS controller. For simulation Fractional order system used is shown in below equation.

$$System1 : \frac{1}{S^{0.5}} \tag{6.2}$$

$$System2 : \frac{S^{1.5} + S^{0.5} + 1}{S^{1.5} + S^{1} + S^{0.5} + 1} \tag{6.3}$$

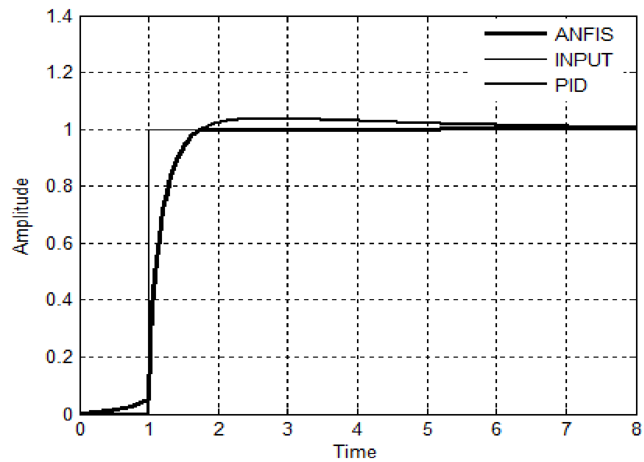

Figure 6.9: step response of Fractional order system 1

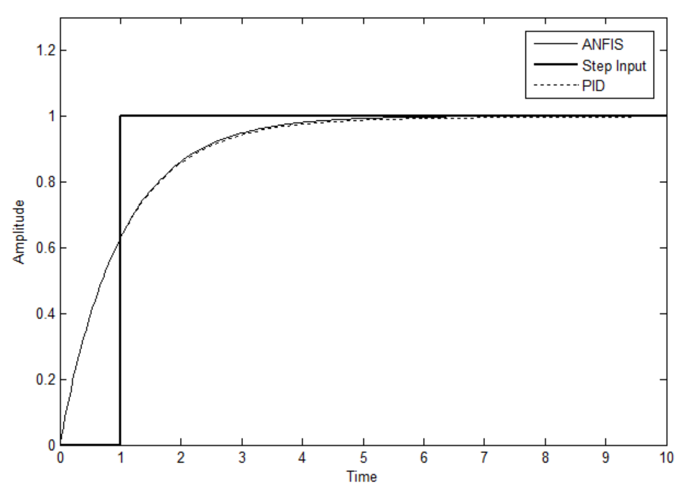

Figure 6.10: step response of Fractional order system 2.

Chapter 7

Fractional order system controlled by ANFIS, CANFIS, and MRAC

Now model reference adaptive control of fractional order system using ANFIS and CANFIS has been carried out in MATLAB Simulink. MIT rules are used to generate adaption law for MRAC. Tuning for the ANFIS and the CANFIS has been carried out by the fractional order MRAC tuning rule. Fractional order MIT rule has been used to control Fractional order system. The comparative analysis between CANFIS and ANFIS has been shown in this Chapter. The fractional order MRAC can be replaced by an appropriate neural network either in form of ANFIS or CANFIS.

Simulation of Fractional order system using Fractional MIT rule has been carried out. Through simulation will verify Theta1 and Theta2 follow each other or not. Simulation was done for First, second, and third order Fractional order system. Figure 7.1 shows Block diagram for simulation.

First order fractional order systems transfer function that was simulated is shown in below equation.

Here we consider reference model is $\frac{0.8S^{1.2}}{3S^{1.2}+1}$

And actual model is $\frac{2S^{1.5}}{2S^{1.5}+1}$

$\frac{\gamma}{S^{\alpha}} = \frac{1}{S^{1.1}}$

$\frac{1}{S+a_m} = \frac{1}{S+2}$

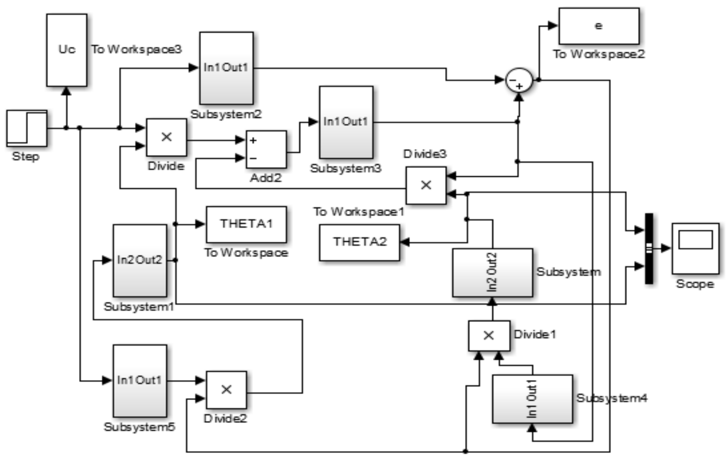

Figure 7.1: Block diagram for fractional order MRAC scheme

Simulation result of Fractional order system using Fractional MIT rule shown in figure(7.2). From figure it was observed that Theta1 and Theta2 follow each other.

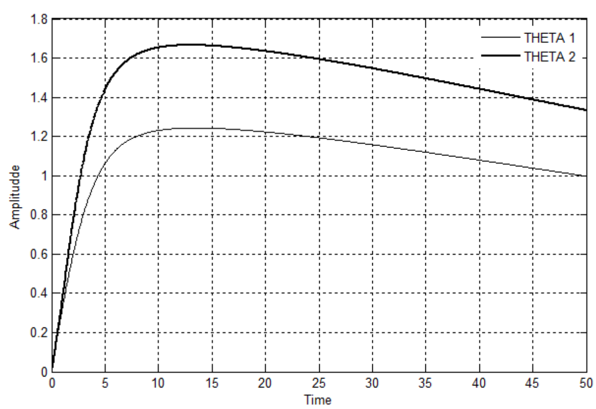

Figure 7.2: Simulation results for Fractional first order system

Second order Fractional order system was controlled by Fractional order MIT rule. Figure (7.3) shows the simulation result of second order Fractional order system. From figure it is observed that second order system gives better result than first order system.

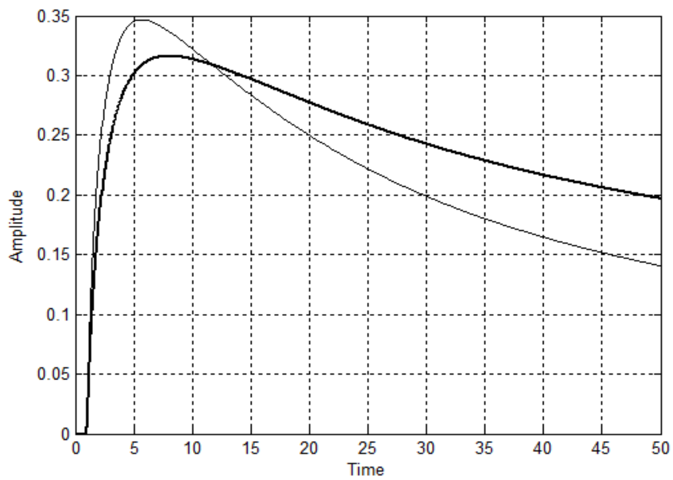

Figure 7.3: Simulation results for Fractional second order system

7.1 MRAC with ANFIS

ANFIS in MATLAB can be used by command line and m-file programs. FIS = GENFIS1 (DATA) generate a single output Sugeno type fuzzy inference system (FIS). FIS is used to provide initial condition to ANFIS training. Here DATA is matrix with N+ 1 column where first column contain each FIS input and last column contain output data.

Simulation studies done using Fractional MRAC to control fractional order system gives output Theta1 and Theta2. It was observed that theta1 and theta2 follow each other as shown in figure (7.2). Figure (7.4) shows the block diagram to control Fractional order system using ANFIS controller which was replaced by fractional MIT rule in the system. ANFIS network was trained by Dataset acquire from simulation model shown in figure (7.1). Dataset contained step input error between Reference model and actual model, Theta1 and Theta2.

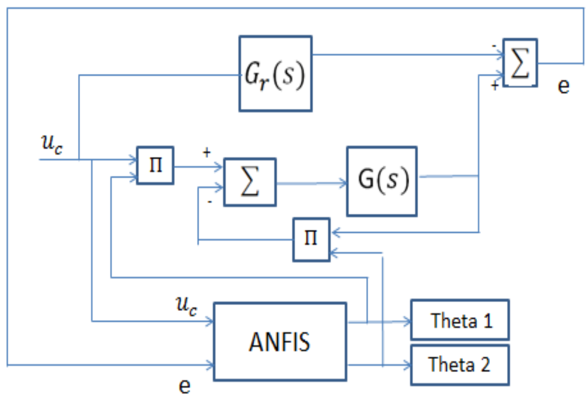

Figure 7.4: block diagram for ANFIS controller

Simulation result of ANFIS controller is shown in figure (7.5). From the figure (7.5) it was observed that Theta 1 and Theta 2 follow each other. Hence, it can be concluded that the ANFIS controller gives better result than MRAC. For the same fractional order system, error analysis was also done and it was observed that the ANFIS has good ability to reduce the amount of feedback error as shown in figure (7.6).

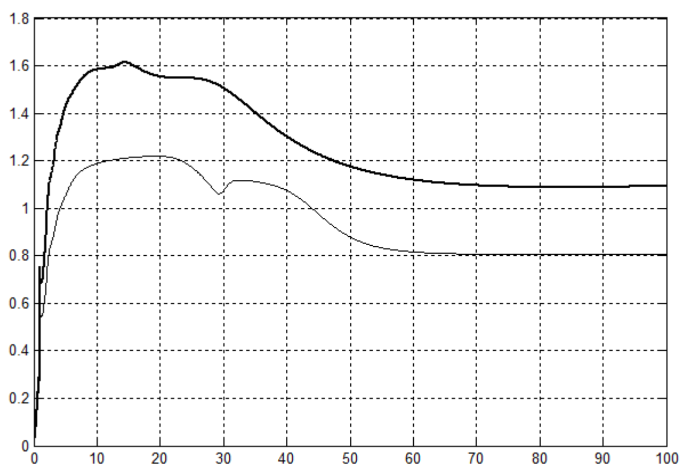

Figure 7.5: Theta 1 and Theta 2 of ANFIS controller

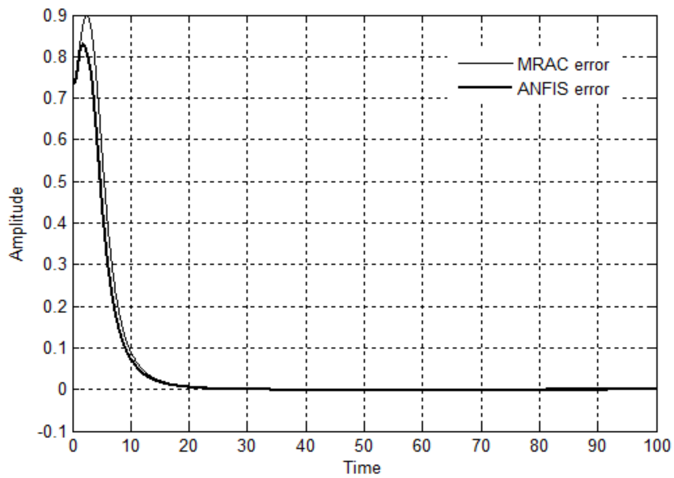

Figure 7.6: comparison of MARC and ANFIS error

7.2 Fractional order system controlled by CANFIS

Simulation was done using ANFIS and Fractional MIT rule. Because ANFIS (Adaptive Neuro Fuzzy Inference System) has been used for the MISO (Multi input single output) system and CANFIS (Coactive Neuro Fuzzy Inference System) is used for MIMO (Multi input Multi output) system. Simulation of Fractional order system using ANFIS controller; there is two ANFIS controller were required because ANFIS is MISO controller. Now ANFIS was replaced by CANFIS. There is three types of CANFIS controller, CANFIS ART, CANFIS GRID, and CANFIS SCATTER. In simulation CANFIS ART controller was used to control Fractional order system. CANFIS ART block explain below.

CANFIS ART block

Input Input x: From this input we give the actual input to CANFIS ART. If give multiple Input by adding multiplexer. When we apply multiple inputs this must be in Vector Formed.

Input e: From this input we give training error signal and dimension of this input is same as output.

Input LE: LE (Learning Enable) as name suggest that it enable (LE=1) or disable (LE=0) the CANFIS. When LE=1 training can be ON and when LE=0 training is running on the basis of previous data.

Output

Output ys: this output of CANFIS ART is main output. Or this is in vector formed.

Output X: This output gives parameter of network during training. This parameter is in vector formed and parameter is saved for future use. In simulation training data set was taken from the figure (7.1). Here we use half data set for training and half for testing.

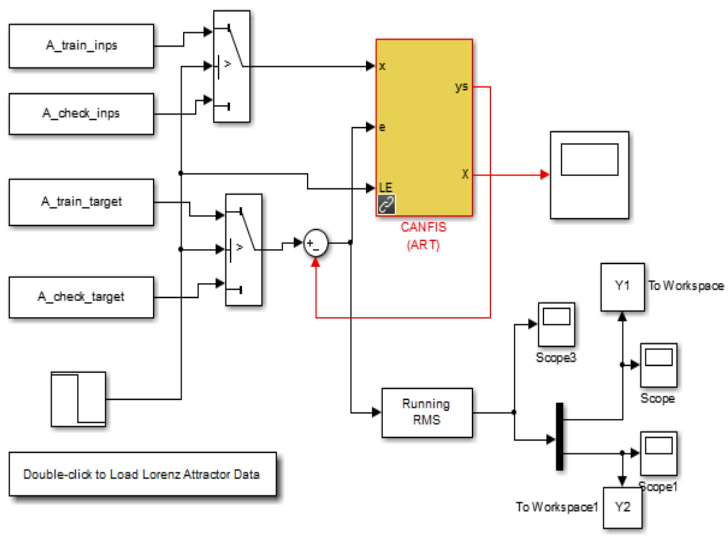

Figure 7.7: block diagram for CANFIS controller

Fractional order system was controlled CANFIS ART controller simulation result of this model in term of error is shown in figure (7.8). Error of ANFIS controller is shown in figure (7.9). From figure (7.8) and figure (7.9) it is observed that CANFIS controller give better result in term of error.

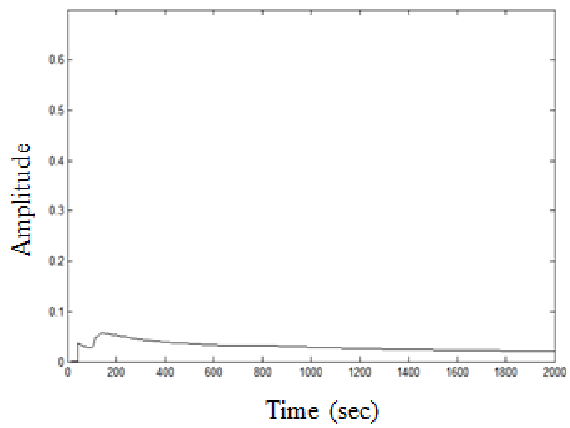

Figure 7.8: simulation Result of CANFIS error

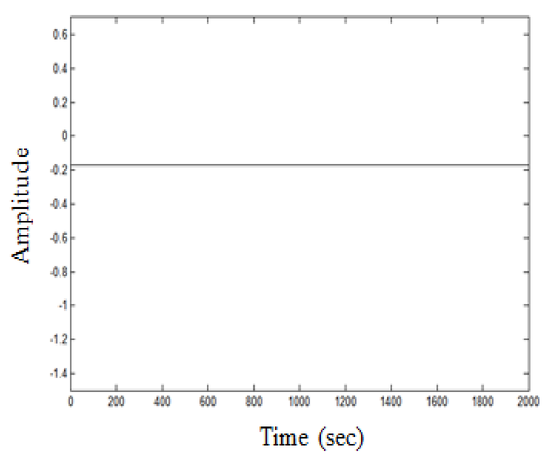

Figure 7.9: Simulation Result of ANFIS error

Chapter 8

Conclusion

First the simulation study for the fractional order transfer function using different control algorithm has been done. Fractional order system was controlled by ANFIS and PID controller. Then comparison of ANFIS and PID was done, from simulation results it was observed that ANFIS gives better result in terms of response time. The fractional order PI algorithm has been tested for the sensor less speed control of 3-phase induction motor. A comparison has been made for the fractional order PI algorithm against the integer order PI algorithm. Fractional order system is controlled by ANFIS and CANFIS. Fractional order MIT rules were used to generate adaption law and control fractional order system. Fractional order MRAC tuning rule was used to tune the ANFIS and CANFIS. Fractional order MRAC was replaced by ANFIS in fractional order system. Then ANFIS was replaced by CANFIS. Comparative analysis between ANFIS and CANFIS has been carried out in terms of error and it has been observed that CANFIS has better ability to reduce the error.

Bibliography

[1] Jyh-Shing Roger Jang, ANFIS : Adap tive-Ne twork-Based Fuzzy Inference System / IEEE Transactions on system , Man, And Cybernetic, Vol. 23, NO. 3, MAY/JUNE 1993

[2] Jyh-Shing Roger Jang , Neuro Fuzzy and Soft Computing , A Computional Approach to Learning and Machine Intelligent.

[3] K.S. Miller, B.Ross. An Introduction to the Fractional Calculus and Fractional Differential Equations. New York: John Wiley and Sons, 1993

[4] K.B.Oldham, J.Spanier. The Fractional Calculus. Theory and Application of Differentiation and Integration of Arbitrary Order. New York: Dover, 2006

[5] Concepcin A. Monje, YangQuan Chen, Blas M. Vinagre, Dingy Xue, Vicente Feliu, Fractional-order Systems and Controls Fundamentals and Applications.

[6] Hseyin Arpaca and Faruk zgvenb, ANFIS and $PI^{\lambda}D^{\mu}$ controller design and comparison for overhead cranes, Indian Journal of Engineering and Materials Sciences Vol. 18, June 2011, pp. 191-203

[7] Osburn, P. V., Whitaker, H. P., and Kezer, A., Comparative studies of model reference adaptive control systems, Institute of Aeronautical Sciences, Paper No. 6139, 1961

[8] Astrom, K. J. and Wittenmark, B., Adaptive Control, Addison-Wesley, Reading, MA, 1995.

[9] Landau, Y. D., Adaptive Control: The Model Reference Approach, Marcel Dekker, New York, 1979.

[10] B. M.Vinagre, I.Petr And I.Podlubny, Y.Q.Chen, Using Fractional Order Adjustment Rules and Fractional Order Reference Models in Model-Reference Adaptive Control Nonlinear Dynamics 29: 269279, 2002. 2002 Kluwer Academic Publishers.

[11] Ankush Kumar , Controller Design for Fractional Order Systems.

[12] Priyank Jain and 2Dr. M.J. Nigam, Design of a Model Reference Adaptive Controller Using Modified MIT Rule for a Second Order System, Advance in Electronic and Electric Engineering. ISSN 2231-1297, Volume 3, Number 4 (2013), pp. 477-484

[13] J.Van.Amerongen , MARS: Model Reference Adaptive System.

[14] D. Matignon. Stability properties for generalized fractional differential systems. D. Matignon, G. Montseny, eds., Proceedings of the Colloquium Fractional Differential Systems: Models, Methods and Applications, 5. Paris, 1998, 145158

[15] Ilias S. Konsoulas , MSc, CEng, MIET, Adaptive Neuro-Fuzzy Inference Systems (ANFIS) Library for Simulink.

[16] Carpenter G.A., Grossberg S., Rosen B.D., Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System, Neural Networks, vol. 4, pp. 759-771, 1991.

[17] C.J. Lin, C.T. Lin, An ART-Based Fuzzy Adaptive Learning Control Network, IEEE Transactions on Fuzzy Systems, vol. 5, No. 4, Nov. 1997.

[18] P. Swarnkar, S. K. Jain and R. K. Nema (2011), Comparative Analysis of MIT Rule and Lyapunov Rule in Model Reference Adaptive Control Scheme, Innovative Systems Design and Engineering,2, 4, pp 154-162.

[19] Podlubny, I.,Fractional-order systems and $PI^{\lambda}D^{\mu}$-controllers, IEEE Transactions on Automatic Control 44(1), 1999, 208214.

[20] http://www.mathwork.com/matlabcentral/example.

[21] D. Valerio. Ninteger Toolbox. [online] http://www.mathworks.com/matlabc

entral /leexchange/ninteger.

[22] A. Garrett, "Fuzzy ART and Fuzzy ARTMAP Neural Networks", 22-12-03, http://www.mathworks.com/matlabcentral/fileexchange/4306-fuzzy-art-and-fuzzy -artmap-neural-networks.

[23] T.Takagi and M.sugeno.Fuzzy identification of system and its applications to modeling and control. IEEE Transactions on system, Man,and Cybernetics,15:116-132, 1985