CHAPTER I

1. INTRODUCTION

1.1 INTRODUCTION TO ROBOT CONTROL

Robotic manipulators have become increasingly important in the field of flexible automation. Robotic manipulators are very complicated nonlinear systems. A robot is typically modeled as a chain of rigid bodies. In general one end of the chain is fixed to some reference surface while other end is free, thus forming an open kinematics chain of moving rigid bodies. Dynamics of a manipulator involve nonlinear mapping between applied joint torques and joint positions, velocities and accelerations. These relationships can be described by a set of second-order nonlinear and highly coupled differential equations with uncertainty as a robot work under unknown and changing environments in executing different tasks.

There are many control strategy that can be applied for control of robotic manipulator are PI, PID And Fuzzy Logic controllers .The Fuzzy Logic approach is particularly important for control of robots and can be used to compensate for highly coupled and nonlinear arm dynamics. Many strategies have been developed for controlling the motion of a robot. Existing robotic manipulators use simple proportional-(integral)-differential controllers with the gains tuned for critical damping. The advantages of a PID controller include its simple structure along with roust performance in a wide range of operating conditions. A lot of research has been done on PID control scheme and available methods for tuning PID gains are advanced and accurate. This makes the PID as one of the most favored control strategy. However, the design of a PID controller is generally based on the assumption of exact knowledge about the system. This assumption is often not valid since the development of any practical system may not include precise information of factors such as friction, backlash, unmodeled dynamics and uncertainty arising from any of the sources. Advanced modern approaches to the design of controllers for robots includes computed torque control robust control model. However, most of these are too complicated and expensive for industrial use. A heavy computational burden prevents them being employed for real-time control applications. Also some of them need an accurate dynamic model which is not always available especially when robot is performing under different operating conditions. In order to overcome above problems, intelligent controlling techniques are used.

Intelligent control is a control technology that replaces the human mind in making decisions, planning control strategies, and learning new functions whenever the environment does not allow or does not justify the presence of human operator. Artificial fuzzy logic are potential tools for intelligent control engineering. Intelligent controllers are Fuzzy logic, PID control and PI control. Fuzzy logic are best known for their learning capabilities. Fuzzy logic is a method of using human skills and thinking processes in a machine.

The underlying idea of fuzzy control is to build a model of a human expert who is capable of controlling the plant without thinking in terms of a mathematical model. The control expert specifies the control actions in the form of linguistic rules. The specification of good linguistic rules depends on the knowledge of the control expert, but the translation of these rules into fuzzy set theory framework is not formalized and arbitrary choices concerning, for example the shape of membership functions have to be made. The quality of a fuzzy logic controller can be drastically affected by the choice of membership functions. Thus, methods for tuning fuzzy logic controllers are necessary. Neural networks offer the possibility of solving the problem of tuning. A combination of neural networks and fuzzy logic offers the possibility of solving tuning problems and design difficulties of fuzzy logic. The resulting network can be easily recognized in the form of fuzzy logic control rules. This new approach combines the well-established advantages of both the methods and avoids the drawbacks of both. The computation of control value from the given measured input value is seen as a feed forward procedure as in layered networks, where the inputs are forwarded through the network resulting in some output value(s). If the actual output value differs from the desired output value, the resulting error is propagated back through the architecture, which in turn results in modification of certain parameters and reduction in error during the next cycle. Interpreting the fuzzy controller as a neural network helps in training the fuzzy controller with learning procedures and the modified structure can still be interpreted as fuzzy logic controller.

1.2 LITERATURE REVIEW ON PID CONTROLLER

C.G .Atkeson, J.D.Griffiths, J.M.Hollerbach, C.H.An, proposed the controllers range from PID control applied independently at each joint to feed forward and computed torque methods incorporating full dynamics. Study shows that dynamic compensation by model based controller can improve trajectory accuracy significantly [4].

Sudeept Mohan, Surekha Bhanot presented a comparative study of simulated performance of some conventional algorithms, like simple PID, Feed forward inverse dynamics, computed torque control, and critically damped inverse dynamics. Study shows that the critically damped inverse dynamics controller in generally performs better then the rest of algorithms particularly when the uncertainty of the system increases.

D.P.kwok, T.P.Leung, Fang Sheng described the use of Genetic Algorithms (GAS) for optimizing the parameters of PID controllers for a robot arm. The simulation results obtained are compared with that obtained by traditional optimization techniques, wherever applicable and showed that the GA-based optimal-tuning technique can work effectively and efficiently and has great potential to become a common optimal-tuning approach for the robot arm [5].

1.3 LITERATURE REVIEW ON FUZZY CONTROL

Han-Xiong Li, H.B.Gatland explain systematic analysis and design of the conventional fuzzy control. A general robust rule base is proposed for fuzzy two-term control, and leave the optimum tuning to the scaling gains, which greatly reduces the difficulties of design and tuning. The digital implementation of fuzzy control is also presented for avoiding the influence of the sampling time [6].

T.Brehm, K.S.Rattan proposed a hybrid fuzzy PID controller which takes advantage of the properties of the fuzzy PI and PD controllers and compared Fuzzy PID and Hybrid Fuzzy PID in terms of rule base, design and implementation problems [7].

G.M.Khoury, M.Saad, H.Y.Kanaan, and C.Asmar presented elaboration of fuzzy control laws based on two structures of coupled rules fuzzy PID controllers and compared the

Two-input FLC with coupled rule, Three-input FLC with coupled rule, computed torque control, and direct adaptive control method on a five-DOF robot arm in terms position tracking errors [8].

Abdollah Homaifar, Ed McCormick examines the applicability of genetic algorithms (GA's) in the simultaneous design of membership functions and rule sets for fuzzy logic controllers. This new method has been applied to two problems, a cart controller and a truck controller. Beyond the development of these controllers, they also examine the design of a robust controller for the cart problem and its ability to overcome faulty rules [9].

1.4 ORGANIZATION OF THE DISSERTATION

This Dissertation is organized as follows.

Chapter 2 Details the Dynamics Modelling Of A Single Flexible Link Robot Arm.

Chapter 3 Explains Conventional Controller.

Chapter 4 Explains the Fuzzy Logic Control.

Chapter 5 Design Of PI, PID And Fuzzy Logic Controller.

Chapter 6 Deals the Design and implementation of Robot Arm, PI, PID and Fuzzy Logic controller in Simulink/Matlab 7.01..

Chapter 7 Simulation Remarks.

Chapter 8 Conclusions.

CHAPTER 2

2. DYNAMIC MODELLING OF A SINGLE FLEXIBLE LINK ROBOT ARM

2.1 INTRODUCTION

J_P is the end mass inertia, J_0 is hub inertia and V_l is the velocity of the end mass M_e at $x=l$.

It is well known that the demand for increased productivity by robots can be partly met by the use of lighter robots operating at high speed and consuming less energy, which may lead to a reduction in the stiffness of the manipulator structure. This would result in an increase in robot deflection and poor performance due to the effect of mechanical vibration in the links and bring difficulties for control [1]. Thus, vibration control of a robotic manipulator system has been an important research area in the last decade [1]-[5].

On the other hand, a robot system is a highly nonlinear and heavily coupled mechanical system, the mathematical model of such system usually consist of a set of linear or nonlinear differential/difference equations derived by using some form of approximation and simulation [1]. Therefore, the traditional model-based control techniques will break down when a complete robot representative model is difficult to obtain due to uncertainly and complexity [6]-[7].

In this paper, the main object of this study is concentrated on the mathematical model for simulation and control of nonlinear vibration of a single flexible link. This establishes a fuzzy logic controller to control nonlinear vibration of a single flexible link. Therefore, a fuzzy logic controller design is adopted and will be applied for such system. In feedback loop of our control system, a fuzzy logic controller (FLC) is used to provide control signals for the manipulator system and used to generate the joint torques and to enhance the performance of the system in vibration process. Namely, we guarantee the nonlinear vibration control and the stability of a single flexible link robot arm by using fuzzy logic controller in the control of this system. Earlier studies discussed about two time scale fuzzy logic and fuzzy neuro control of flexible and rigid links , modelling and control of a single flexible beam such as our study [1]-[5]. And this study brings together and employs the concepts of several of the papers mentioned above and showed references.

2.2 DYNAMIC MODEL

The system under consideration is a single flexible link robotic manipulator. The physical configuration of the robot arm considered in this work is given in Fig.1. The elastic arm is assumed as an Euler-Bernoulli beam. The mass and elastic properties are assumed to be distributed uniformly along the elastic arm. Torque *T* rotates the elastic arm about *Z* axis.

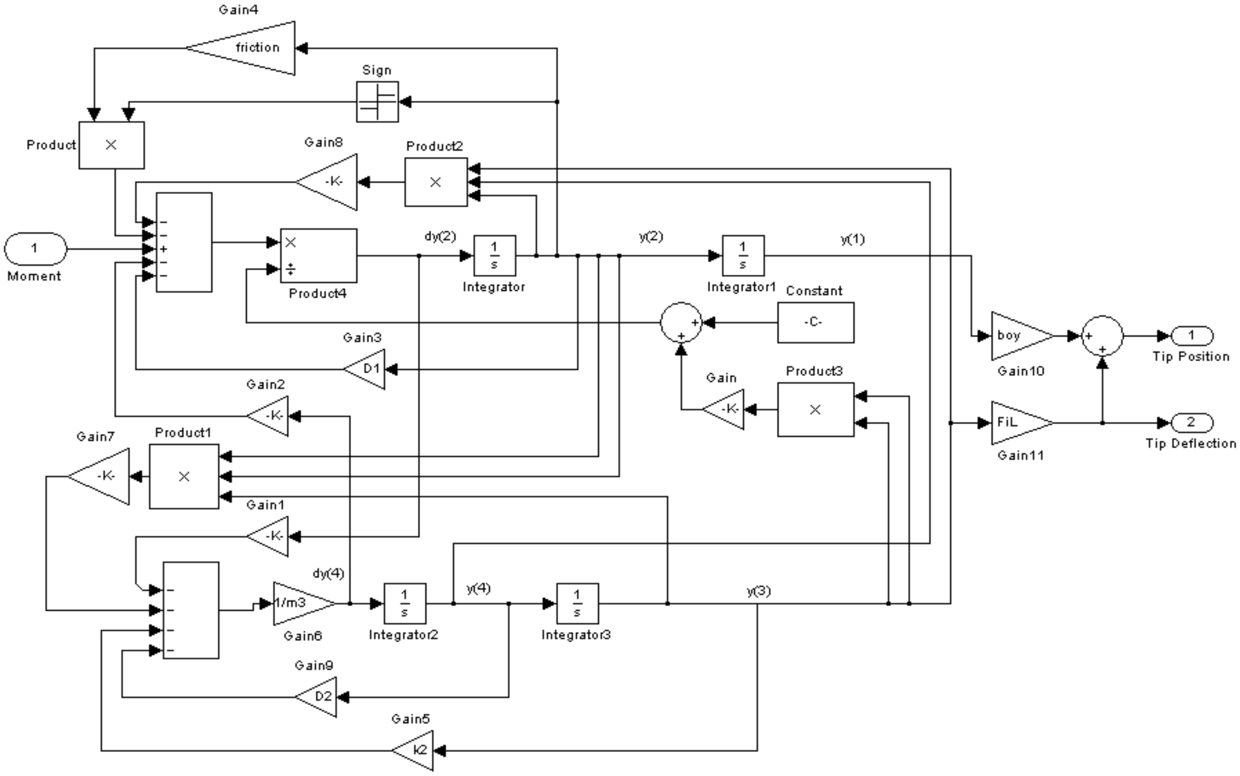


Fig1.1 Model of a single flexible link robot arm manipulator

2.3 KINEMATIC ANALYSIS AND ENERGY TERMS

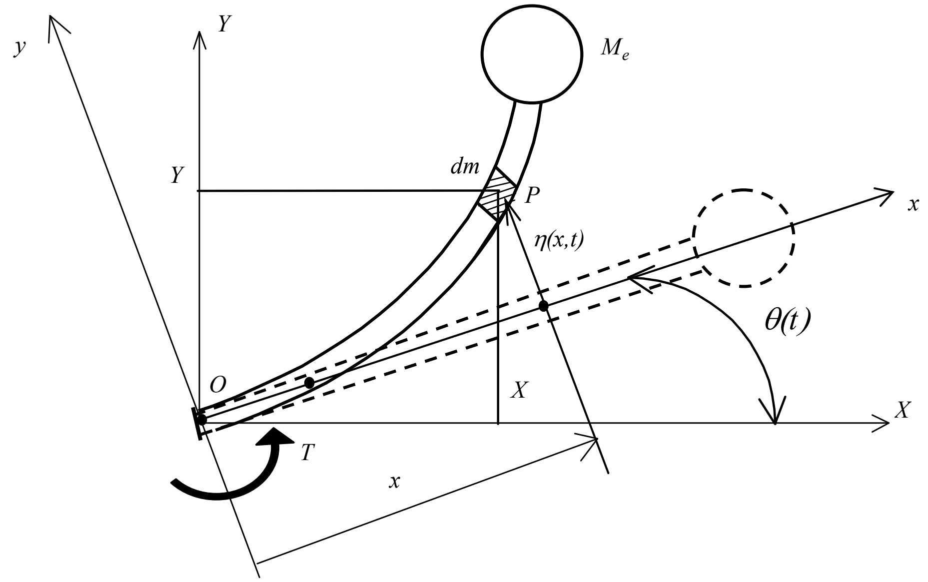


Figure 1.2 Robot manipulator with elastic arm sliding in a prismatic joint

In order to obtain the equations of motion by using Lagrange's equation of motion; the energy terms have to be evaluated. Transformation matrices are used in kinematic analysis of the robot arm. *XYZ* is the global reference frame, while *xyz* is rotating reference frame. The angle between the rotating reference frame *xyz* and the global reference frame *XYZ* is θ. In order to obtain the velocity terms; an infinitesimal mass *dm* on the elastic arm is considered as seen in Figure 1.2 Distance of *dm* to the origin in *x* direction is *x* and the displacement from the undeformed position in *y* direction of the elastic arm is η. The coordinates of *dm* with respect to the global reference frame *XYZ* are found as:

$$\begin{Bmatrix} X \\ Y \end{Bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{Bmatrix} x \\ \eta \end{Bmatrix} \tag{1}$$

Differentiating the expression in Equation (1) we obtain

$$\dot{X} = -\eta\dot{\theta}\cos\theta - \left[x\dot{\theta} + \dot{\eta}\right]\sin\theta$$

$$\dot{Y} = -\eta\dot{\theta}\sin\theta + \left[x\dot{\theta} + \dot{\eta}\right]\cos\theta$$

By using these Equations, velocity of *dm* can be written as:

$$V^2 = \dot{X}^2 + \dot{Y}^2 = \left[x^2 + \eta^2\right]\dot{\theta}^2 + \dot{\eta}^2 + 2\dot{\theta}\,x\,\dot{\eta}$$

Kinetic energy of the elastic arm is written as:

$$KE = 0.5J_0\dot{\theta}^2 + 0.5\int_0^l V^2 dm + 0.5M_e V_l^2 + 0.5J_P\left[\dot{\theta} + \eta'\right]^2$$

where Assuming that the length of the link is an order of magnitude larger than its cross-sectional dimensions, shear and rotary inertia of the cross section can be neglected. In this case the only source of potential energy due to elastic deformations is written as:

$$PE = 0.5\int_0^l EI\left(\frac{\partial^2\eta}{\partial x^2}\right)^2 dx$$

2.4 EQUATIONS OF MOTION

An infinite series solution in the form of

$$\eta(x,t) = \sum_{i=1}^{\infty}\phi_i(x,t)q_i(t) = [\phi]\{q\}$$

is assumed for the elastic displacements of the elastic arm where $\{q\}$ is the vector of time dependent generalized coordinates and $[\phi]$ is the matrix of time and space dependent eigenfunctions. The elastic robot arm is assumed as a cantilever beam carrying an end mass. Eigenfunctions of a cantilever beam with constant length is:

$$\phi = \cosh\lambda\frac{x}{l} - \cos\lambda\frac{x}{l} - \frac{\cosh\lambda + \cos\lambda}{\sinh\lambda + \sin\lambda}\left(\sinh\lambda\frac{x}{l} - \sin\lambda\frac{x}{l}\right) \tag{2}$$

Each individual mode shape function ϕ may be found by substituting the value λ determined from the following transcendental equation into Equation 2

$$\cosh\lambda\cos\lambda + 1 = 0$$

The Lagrangian can be found as:

$$L = KE - PE$$

Therefore, the Euler-Lagrange equations can be applied to L to find the equations of motion given by

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right)-\frac{\partial L}{\partial \theta}=T-\mu_1\dot{\theta}-T_{\mu}\left(\dot{\theta}\right)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \ddot{q}}\right)-\frac{\partial L}{\partial q}=-\mu_2\dot{q}$$

Where μ_1 and μ_2 the viscous and structural damping coefficients and $T_{\mu}\ (\dot{\theta})$ is the Coulomb friction torque. Now the equations of motion are obtained as:

$$[M]\begin{Bmatrix}\ddot{\theta}\\ \ddot{q}\end{Bmatrix}+\begin{bmatrix}H_1+T_{\mu}\left(\dot{\theta}\right)\\ H_2\end{bmatrix}+\begin{bmatrix}0 & 0\\ 0 & EI\int_0^l \phi''\phi''dx\end{bmatrix}\begin{Bmatrix}\theta\\ q\end{Bmatrix}+\begin{bmatrix}\mu_1 & 0\\ 0 & \mu_2\end{bmatrix}\begin{Bmatrix}\dot{\theta}\\ \dot{q}\end{Bmatrix}=\begin{Bmatrix}T\\ 0\end{Bmatrix}$$

Where [M] represents the inertia matrix, H_1 and H_2 represents Coriolis and centrifugal forces.

CHAPTER 3

3. CONVENTIONAL CONTROLLERS

3.1 INTRODUCTION

There are many control strategies that can be applied for control of robot arm. These strategies are conventional, adaptive and intelligent control strategies. The general structure of a robot manipulator with controller is shown in figure 3.1 below. The trajectory generator provides the controller with information about the desired position, velocity and acceleration ($\theta_d, \dot{\theta}_d, \ddot{\theta}_d$) for each joint and keeps updating this information at the path update rate. The controller takes this information and compares it with the present (actual) position and velocity (sometimes acceleration also) of joints ($\theta, \dot{\theta}, \ddot{\theta}$), which are provided as feedback through the sensors.

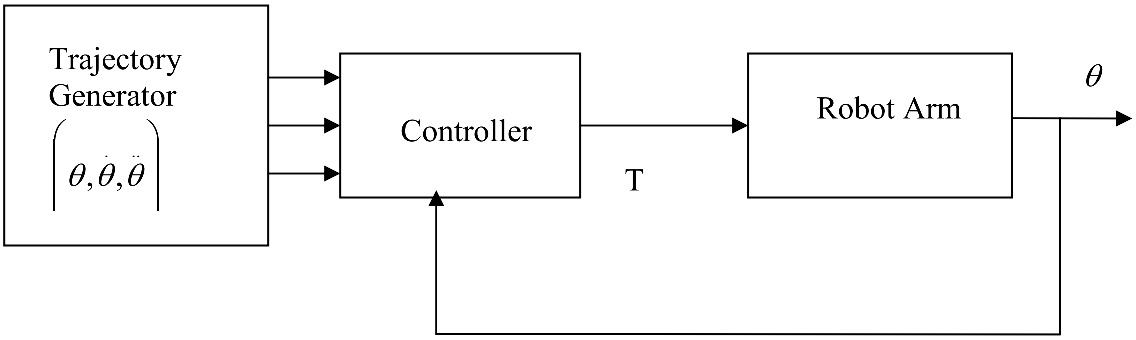


Fig .3.1 General structure of robot control system

Based upon the error between the desired and actual values, the controller calculates a vector of torques (τ) which should be applied at respective joints by the actuators to minimize these errors. The torques is calculated using control law. The goal of the controller is thus, minimization of error, e and its first derivative $\dot{e}$. The dynamic model of Robotic manipulator can be described in the form of equation as below

$$\tau(t) = M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) \qquad \ldots 3.1$$

Where, $M(\theta)$ is the inertia matrix, $C(\theta,\dot{\theta})$ is the centripetal-coriolis matrix, and $G(\theta)$ is the Gravity vector, θ is Joint angles, τ is the joint actuator torque. The use of linear control techniques for any system is valid only when the system to be controlled can be modeled by linear differential equations. Thus the linear control of robot manipulators is essentially an approximation, as the manipulator dynamics is described by highly non-linear equations. The linear control strategies for robots give excellent performance for manipulators having highly geared joints. This is the case with most of the industrial robots in use today.

3.2 PID Control

One common linear control strategy is PID (proportional-derivative and integral) control. The control law used for this strategy is given by

$$\tau_{PID} = K_D \dot{e} + K_P e + K_I \int e dt \qquad \ldots 3.2$$

K_D, K_I and K_P are the controller gain matrices. τ_{PID} is the vector of joint torques. It is possible to get the desired performance from the system by choosing the appropriate values of parameters of PID controller. Hand tuning method is used for selection of PID control gains. A robotic control system cannot be allowed to have an oscillatory response for obvious reasons. For instance, in a pick-*n*-place operation, an oscillating end-effecter may strike against the object before picking it to manipulate. Hence, highest possible speed of response and yet non-oscillatory response, dictates that the controller design parameters shell be chosen to have the damping ratio equal to unity or least close to it but less than unity.

3.3 Feed Forward inverse dynamics control

Feed forward inverse dynamics control is a model based non-linear technique. Scheme for feed forward inverse dynamics control is shown below Fig 3.2. This scheme uses the inverse dynamics equations of robot manipulator in feed forward mode. As can be seen from this figure, the sum of the outputs of the inverse model and feedback controller (i.e. PID Controller) will be the actual input torque to robot.

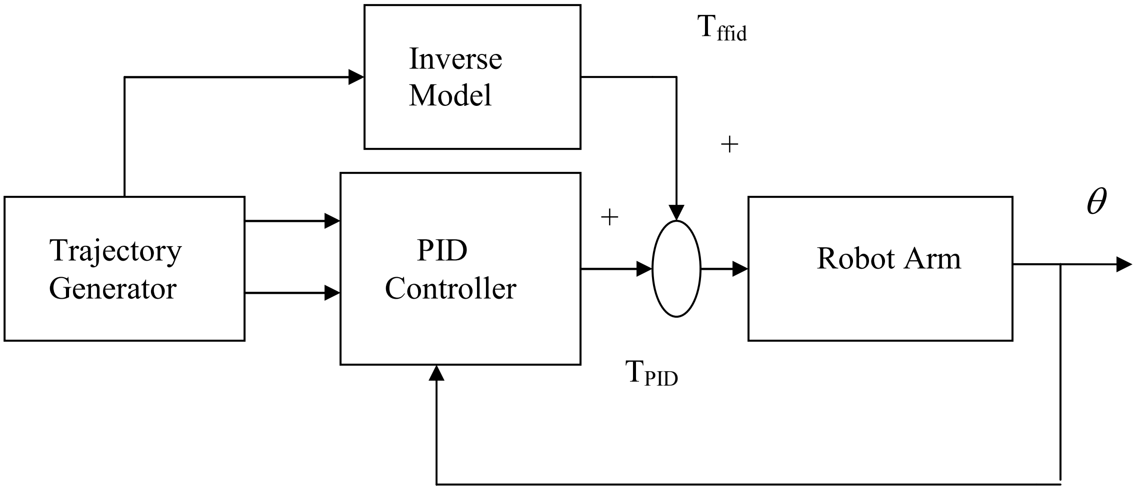


Fig.3.2 Feed forward inverse dynamics controller

In this strategy the torque is calculated as

$$\tau_{ffid} = M(\theta)\ddot{\theta} + C\left(\theta, \dot{\theta}\right)\dot{\theta} + G(\theta) \qquad \ldots 3.3$$

$$\tau_{PID} = K_D \dot{e} + K_P e + K_I \int e dt \qquad \ldots 3.4$$

Total control torque is $\tau = \tau_{PID} + \tau_{ffid}$. The feedback controller plays a role in making the whole system stable.

3.4 Computed Torque Control

The most common non-linear control technique for manipulator control is the Computed torque control. Scheme is similar to feed forward inverse dynamic control. Here the computed torque is given by

$$\tau_{CTC} = \tau_{PID} + M(\theta)\left[\ddot{\theta} + K_D \dot{e} + K_P e\right] + C\left(\theta, \dot{\theta}\right)\dot{\theta} + G(\theta) \qquad \ldots 3.5$$

If the manipulator model is known exactly then this scheme results in asymptotically stable and provides asymptotically exact tracking.

3.5 Critically damped inverse dynamics control

This control strategy is almost same as inverse dynamics except that the feed forward torque is calculated using reference velocity and reference acceleration instead of the desired values. These reference values are defined as

$$\begin{aligned} \dot{\theta}_R &= \dot{\theta}_d + K_P\left(\theta_d - \theta\right) \\ \ddot{\theta}_R &= \ddot{\theta}_d + K_D\left(\dot{\theta}_d - \dot{\theta}\right) \end{aligned} \qquad \ldots 3.6$$

In this strategy the torque is calculated as

$$\tau_{CDID} = \tau_{PID} + M(\theta_R)\ddot{\theta}_R + C\left(\theta_R, \dot{\theta}_R\right)\dot{\theta}_R + G(\theta_R) \qquad \ldots 3.7$$

CHAPTER 4

4. FUZZY LOGIC PRINCIPLES AND FUZZY CONTROL

4.1 GENERAL

Conventional controllers are derived from control theory techniques based on mathematical models of the open-loop process. These processes are called system to be controlled. The purpose of the feedback controller is to guarantee a desired response of the output say y. The process of keeping the output say ‘y’ close to the setpoint (reference input) ‘y*’, despite the presence of disturbances and noise in the system parameters, is called regulation .The output of the controller (which is the input of the system) is the control action u. The general form of the discrete-time control law is

$$u(k) = f(e(k), e(k-1), \ldots, e(k-\tau), u(k-1), \ldots, u(k-\tau))$$

providing a control action that describes the relationship between the input and the output of the controller.

- ‘e’ represents the error between the desired setpoint y* and the output of the system y,
- parameter ‘τ’ defines the order of the controller,
- ‘f’ is in general a nonlinear function.

4.2 Basic Principle of Fuzzy System

In a fuzzy logic controller (FLC), the dynamic behavior of a fuzzy system is characterized by a set of linguistic description rules based on expert knowledge. The expert knowledge is usually of the form IF (a set of conditions are satisfied) THEN (a set of consequences can be inferred). Since the antecedents and the consequents of these IF-THEN rules are associated with fuzzy concepts (linguistic terms), they are often called fuzzy conditional statements. In our terminology, a fuzzy control rule is a fuzzy conditional statement in which the antecedent is a condition in its application domain and the consequent is a control action for the system under control.

Basically, fuzzy control rules provide a convenient way for expressing control policy and domain knowledge. Furthermore, several linguistic variables might be involved in the antecedents and the conclusions of these rules. When this is the case, the system will be referred to as a multi-input-multi-output (MIMO) fuzzy system.

For example, in the case of two-input-single-output (MISO) fuzzy systems, fuzzy control rules have the form

1 if x is A_1 and y is B_1 then z is C_1
also
2 if x is A_2 and y is B_2 then z is C_2
n if x is A_n and y is B_n then z is C_n

where x and y are the process state variables, z is the control variable, A, B, and C are linguistic values of the linguistic variables x, y and z in the universes of discourse U, V , and W, respectively, and an implicit sentence connective also links the rules into a rule set or, equivalently, a rule-base. Which can represent the FLC in a form, similar to the conventional control law

$$u(k) = F(e(k), e(k-1), \ldots, e(k-\tau), u(k-1), \ldots, u(k-\tau)) \quad (4.1)$$

where the function F is described by a fuzzy rule base. However it does not mean that the FLC is a kind of transfer function or difference equation. The knowledge-based nature of FLC dictates a limited usage of the past values of the error ‘e’ and control ‘u’ because it is rather unreasonable to expect meaningful linguistic statements for

$e(k-3), e(k-4), \ldots, e(k-\tau)$.

A typical FLC describes the relationship between the change of the control

$$\Delta u(k) = u(k) - u(k-1) \quad (4.2)$$

on the one hand, and the error e(k) and its change

$$\Delta e(k) = e(k) - e(k-1). \quad (4.3)$$

on the other hand. Such control law can be formalized as

$$\Delta u(k) = F(e(k), \Delta(e(k))) \quad (4.4)$$

and is a manifestation of the general FLC expression
with $\tau = 1$.
N error ZE P
The actual output of the controller u(k) is obtained from the previous value of control u(k − 1) that is updated by $\Delta u(k)$

$$u(k) = u(k-1) + \Delta u(k). \tag{4.5}$$

This type of controller was suggested originally by Mamdani and Assilian in 1975 and is called the Mamdani type FLC. A prototypical rule-base of a simple FLC realising the control law above is listed in the following
1: If e is ”positive” and Δe is ”near zero” then Δu is ”positive”
2: If e is ”negative” and Δe is ”near zero” then Δu is ”negative”
3: If e is ”near zero” and Δe is ”near zero” then Δu is ”near zero”
4: If e is ”near zero” and Δe is ”positive” then Δu is ”positive”
5: If e is ”near zero” and Δe is ”negative” then Δu is ”negative”
So, our task is to find a crisp control action z_0 from the fuzzy rule-base and from the actual crisp inputs x_0 and y_0:
1: if x is A_1 and y is B_1 then z is C_1
2: if x is A_2 and y is B_2 then z is C_2
n: if x is A_n and y is B_n then z is C_n input x is x_0 and y is y_0 output z_0

Of course, the inputs of fuzzy rule-based systems should be given by fuzzy sets, and therefore, we have to fuzzify the crisp inputs. Furthermore, the output of a fuzzy system is always a fuzzy set, and therefore to get crisp value we have to defuzzify it. Different defuzzification methods are available for the purpose. Fuzzy logic control systems usually consist of four major parts: Fuzzification interface, Fuzzy rulebase, Fuzzy inference machine and Defuzzification interface. The (fig4.1) shows the functional block diagram of fuzzy logic control system .A fuzzification operator has the effect of transforming crisp data into fuzzy sets. In most of the cases we use fuzzy singletons as fuzzifiers fuzzifier $(x_0) := \bar{x}_0$ where x_0 is a crisp input value from a process. The (fig3.2) shows the fuzzy singleton as fuzzyfier

.

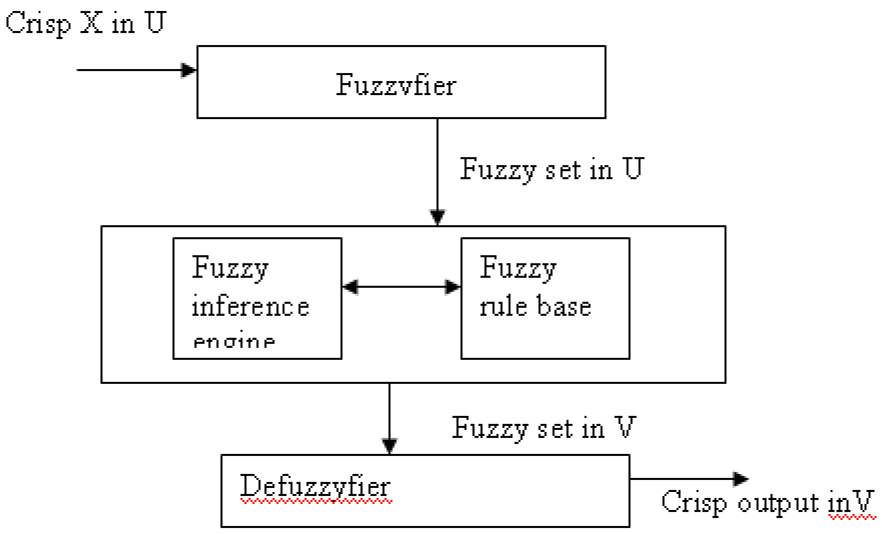


Fig 4.1 Block Diagram Of Fuzzy Controller

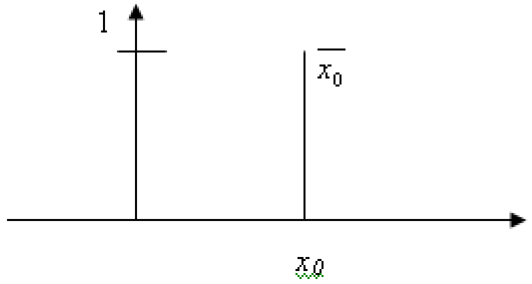


Fig 4.2 Fuzzy Singleton

4.3 Preliminary Mathematics

Suppose that there are two input variables x and y. A fuzzy control rule i : if (x is Ai and y is Bi) then (z is Ci) is implemented by a fuzzy implication Ri and is defined as

$$Ri(u, v,w) = Ai(u) \text{ and } Bi(v) \rightarrow Ci(w) \qquad (4.6)$$

where the logical connective and is implemented by the minimum operator, i.e.

$Ai(u) \text{ and } Bi(v) \rightarrow Ci(w) =$

$$Ai(u) \times Bi(v) \rightarrow Ci(w) = \min\{Ai(u),Bi(v)\} \rightarrow Ci(w) \qquad (4.7)$$

Of course, one can use any t-norm to model the logical connective. Fuzzy control rules are combined by using the sentence connective also. Since each fuzzy control rule is represented by a fuzzy relation, the overall behavior of a fuzzy system is characterized by these fuzzy relations. In other words, a fuzzy system can be characterized by a single fuzzy relation which is the combination of question involves the sentence connective also. Symbolically, if we have the collection of rules

1 : if x is A_1 and y is B_1 then z is C_1
2 : if x is A_2 and y is B_2 then z is C_2
n : if x is A_n and y is B_n then z is Cn

The procedure for obtaining the fuzzy output of such a knowledge base consists of following three steps:

1 Find the firing level of each of the rules.
2 Find the output of each of the rules.
3 Aggregate the individual rule outputs to obtain the overall system output.

To infer the output z from the given process states x, y and fuzzy relations Ri, we apply the compositional rule of inference:

1 : if x is A_1 and y is B_2 then z is C_1
2 : if x is A_2 and y is B_2 then z is C_2
n : if x is A_n and y is B_n then z is C_n
fact : x is $\bar{x}_0$ and y is $\bar{y}_0$
consequence : z is C where the consequence is computed by

consequence = Agg (fact ∘ _1, . . . , fact ∘ _n).

That is,

$$C = \mathrm{Agg}(\bar{x}_0 \times \bar{y}_0 \circ R1, \ldots, \bar{x}_0 \times \bar{y}_0 \circ Rn) \tag{4.8}$$

taking into consideration that

$\bar{x}_0(u) = 0$, u _= x_0

and

$\bar{y}_0(v) = 0$, v _= y_0,

the computation of the membership function of C is very simple:

$C(w) = Agg\{A_1(x_0) \times B_1(y_0) \rightarrow C1(w), . . . , A_n(x0) \times B_n(y0) \rightarrow C_n(w)\}$ (4.9)

for all $w \in W$. The procedure for obtaining the fuzzy output of such a knowledge base can be formulated as

- The firing level of the i-th rule is determined by $Ai(x_0) \times Bi(y_0)$.
- The output of of the i-th rule is calculated by $Ci(w) := Ai(x_0) \times Bi(y_0) \rightarrow Ci(w)$ for all $w \in W$.
- The overall system output, C, is obtained from the individual rule outputs Ci by

$C(w) = Agg\{C_1, . . . , C_n\}$ for all $w \in W$.

4.4 Defuzzification

The output of the inference process so far is a fuzzy set, specifying a possibility distribution of control action. In the on-line control, a nonfuzzy (crisp) control action is usually required. Consequently, one must defuzzify the fuzzy control action (output) inferred from the fuzzy control algorithm, namely: z_0 = defuzzifier(C), where z_0 is the nonfuzzy control output and defuzzifier is the defuzzification operator. Defuzzification is a process to select a representative element from the fuzzy output C inferred from the fuzzy control algorithm.

4.5 Fuzzification

The first block inside the controller is fuzzification, which converts each piece of input data to degrees of membership by a lookup in one or several membership functions. The fuzzification block thus matches the input data with the conditions of the rules to determine how well the condition of each rule matches that particular input instance. There is a degree of membership for each linguistic term that applies to that input variable.

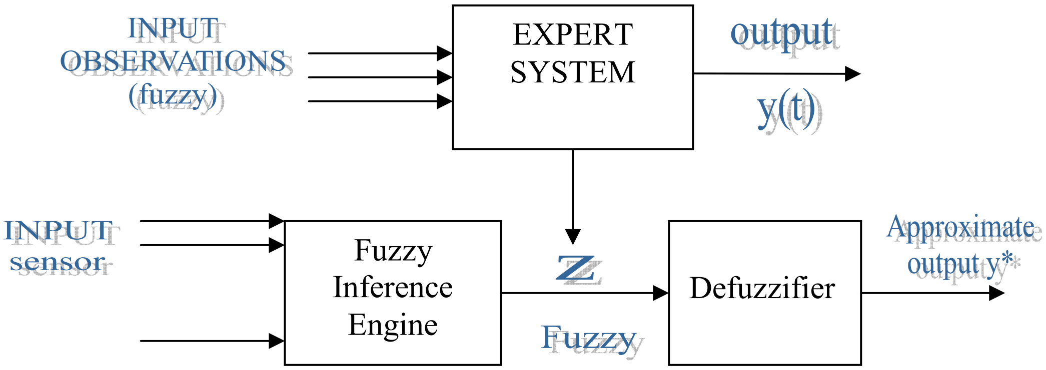


Fig 4.3 The basic configuration of the fuzzy system.

4.6 Rule Base

The rules may use several variables both in the condition and the conclusion of the rules. The controllers can therefore be applied to both multi-input-multi-output (MIMO) problems and single-input-single-output (SISO) problems. The typical SISO problem is to regulate a control signal based on an error signal. The controller may actually need both the error, the change in error, and the accumulated error as inputs, but we will call it single-loop control, because in principle all three are formed from the error measurement. To simplify, this section assumes that the control objective is to regulate some process output around a prescribed set point or reference. The presentation is thus limited to single-loop control.

Rule Format- Basically a linguistic controller contains rules in the if -then format, but they can be presented in different formats. In many systems, the rules are presented to the end-user in a format similar to the one below,

1. If error is Neg and change in error is Neg then output is NB
2. If error is Neg and change in error is Zero then output is NM
3. If error is Neg and change in error is Pos then output is Zero
4. If error is Zero and change in error is Neg then output is NM
5. If error is Zero and change in error is Zero then output is Zero

6. If error is Zero and change in error is Pos then output is PM
7. If error is Pos and change in error is Neg then output is Zero
8. If error is Pos and change in error is Zero then output is PM
9. If error is Pos and change in error is Pos then output is PB

It should be emphasized though, that the relational format implicitly assumes that the connective between the inputs is always logical AND or logical OR for that matter as long as it is the same operation for all rules and not a mixture of connectives. Incidentally a fuzzy rule with an OR combination of terms can be converted into an equivalent AND combination of terms using laws of logic (DeMorgan's) laws among others). The input variables are laid out along the axes, and the output variable is inside the table. In case the table has an empty cell, it is an indication of a missing rule, and this format is useful for checking completeness. When the input variables are error and change in error, as they are here, that format is also called a linguistic rules. In case there are n>2 input variables involved, the table grows to an n-dimensional array, rather user-ufriendly.

Connectives- In mathematics, sentences are connected with the words AND,OR,IF-THEN and IF AND ONLY IF, or modifications with the word NOT. These five are called five connectives. It also makes a difference how the connectives are implemented. The most prominent is probably multiplication for fuzzy AND instead of minimum.

Universe-The universe contains all elements that can come into consideration. Before designing the membership functions it is necessary to consider the universes for the inputs and outputs. Take for example the rule If error is Neg and change in error is Pos then output is zero. Naturally, the membership functions for Neg and Pos must be defined for all possible values of error and change in error and a standard universe may be convenient .Another consideration is whether the input membership functions should be continuous or discrete. A continuous membership function is defined on a continuous universe by means of parameters. A discrete membership function is defined in terms of a vector with a finite number of elements. In the latter case it is necessary to specify the range of the universe and the value at each point. The choice between fine and coarse resolution is a trade off between accuracy, speed and space demands. The quantiser takes time to execute, and if this time is too precious, continuous membership functions will make the quantiser obsolete. A way to exploit the range of the universes better is scaling. If a controller input mostly uses just one term, the scaling factor can be turned up such that the whole range is used. An advantage is that this allows a standard universe and it eliminates the need for adding more terms.

Membership Functions- Every element in the universe of discourse is a member of a fuzzy set to some grade, maybe even zero. The grade of membership for all its members describes a fuzzy set, such as Neg. In fuzzy sets elements are assigned a grade of membership function such that the transition from membership to non-membership is gradual rather than abrupt. The set of elements that have a non-zero membership is called the support of the fuzzy set. The function that ties a number to each element of the universe is called the membership function,.(fig6) shows different types of membership function.

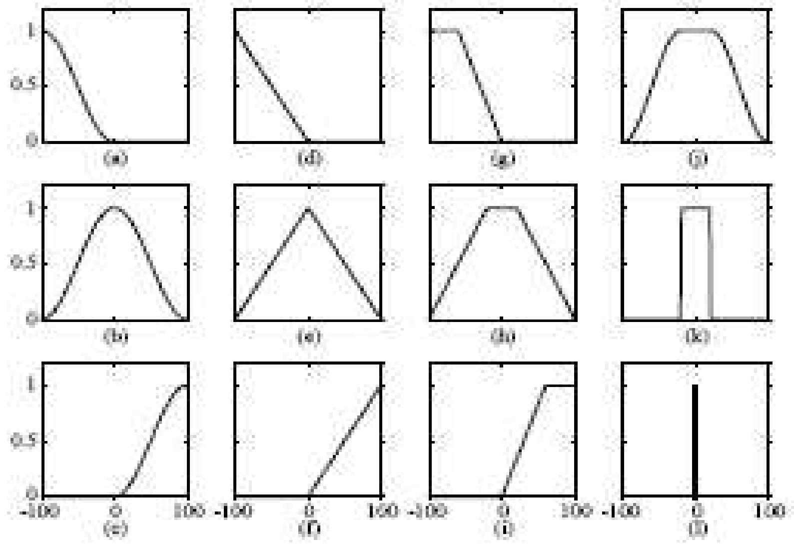


Figure 6: Examples of membership functions. Read from top to bottom, left to right: (a) s−function, (b) π− function, (c) z−function, (d-f) triangular versions, (g-i) trapezoidal versions, (j) flat π− function, (k) rectangle, (l) singleton.

.

A certain amount of overlap is desirable in selection of the range of the membership functions; otherwise the controller may run in poorly defined states, where it does not return a well defined output.

The preliminary answer to question is that the necessary and sufficient number of sets in a family depends on the width of the sets, and vice versa. A solution could be to ask the process operators to enter their personal preferences for the membership curves, but operators also find it difficult to settle on particular curves. The manual for the TIL Shell product recommends the membership function to be selected.

4.7 Inference Engine

For each rule, the inference engine looks up the membership values in the condition of the rule.

Aggregation -The operation is used when calculating the degree of fulfillment or firing strength. A_k of the condition of a rule k. A rule, say rule 1, will generate a fuzzy membership value from the error and a membership value from the change in error measurement. The aggregation is their combination, is their AND. Similarly for the other Rules. Aggregation is equivalent to fuzzification when there is only one input to the controller. Aggregation is sometimes also called fulfillment of the rule or firing strength.

Activation – The Activation of a rule is the deduction of the conclusion, possibly reduced by its firing strength. Thickened lines in the column indicate the firing strength of each rule. Only the thickened part of the singletons are activated, and MIN or product (*) is used as the operator of activation

Accumalated- All activated conclusions are accumulated, using the MAX operation. Alternatively, SUM accumulation counts overlapping are as more than once. Singleton output and sum accumulation results in the simple output.

4.8 DEFUZZICATION

The resulting fuzzy set must be converted to a number that can be sent to the process as a control signal. This operation is called defuzzication. The resulting fuzzy set is thus defuzzified into a crisp control signal. There are several defuzzication methods.

Centre of gravity -The crisp output value x is the abscissa under the centre of gravity of the fuzzy set

$$U = \frac{\sum \mu\ (xi)xi}{\mu(xi)}$$

Here xi is a running point in a discrete universe, and μ(xi), is its membership value in the membership function. The expression can be interpreted as the weighted average of the elements in the support set. For the continuous case, replace the summations by integrals. It is a much used method although its computational complexity is relatively high. This method is also called centroid of area.

Centre of gravity and methods of singletons- If the membership functions of the conclusions are singletons, the output value is as per the mathematical expression

Here S_i is the position of singleton i in the universe, and U (s_i) is equal to the firing strength. This method has a relatively good computational complexity, and U is differentiable with respect to the singletons S_i, which is useful in fuzzy systems.

Bisection of area - This method picks the abscissa of the vertical line that divides the area under the curve in two equal halves. In the continuous case

$$U = \int_{\min}^{x} \mu(xi)dx = \int_{x}^{\max} \mu(xi)dx$$

Here X is the running point in the universe, U(x) is its membership, MAX is the leftmost value of the universe, and MIN is the rightmost value. Its computational complexity is relatively high, and it can be ambiguous. For example, if the fuzzy set consists of two singletons any point between the two would divide the area in two halves; consequently it is safer to say that in the discrete case, BOA is not defined.

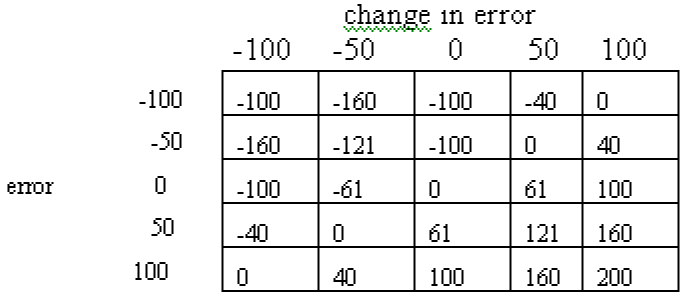
Mean of maxima- An intuitive approach is to choose the point with the strongest possibility, i.e. maximal membership. It may happen, though, that several such points exist and a common practice is to take the Mean of Maxima (MOM). This method disregards the shape of the fuzzy set, but the computational complexity is relatively good Left Most Maximum(LM) Right Most Maxima(RM)- Another possibility is to choose the leftmost maximum (LM), or the rightmost maximum (RM). In the case of a robot for instance, it must choose between left or right to avoid an obstacle in front of it. The defuzzifier must then choose one or the other, not something in between. These methods are indifferent to the shape of the fuzzy set, but the computational complexity is relatively small.

4.9 Post-Processing

Output scaling is also relevant. In case the output is defined on a standard universe this must be scaled to engineering units_ for instance, volts, meters, or tons per hour. An example is the scaling from the standard universe [-1,1] to the physical units [-10,10] volts. The post processing block often contains an output gain that can be tuned, and sometimes also an integrator.

4.10 Table Based Controller

If the universes are discrete, it is always possible to calculate all thinkable combinations of inputs before putting the controller into operation. In a Table Based Controller the relation between all input combinations and their corresponding outputs are arranged in a table. With two inputs and one output, the table is a two-dimensional look-up table. With three inputs the table becomes a three-dimensional array. The array implementation improves execution speed, as the run-time inference is reduced to a table look-up which is a lot faster, at least when the correct entry can be found without too much searching.



		change in error				
		-100	-50	0	50	100
	-100	-100	-160	-100	-40	0
	-50	-160	-121	-100	0	40
error	0	-100	-61	0	61	100
	50	-40	0	61	121	160
	100	0	40	100	160	200

A typical application area for the table based controller is where the inputs to the controller are the Error and the Change of Error. The controller can be embedded in a larger system, a car for instance, where the table is downloaded to a table look-up mechanism(above table).

4.11 Procedure

When fuzzy set theory is used to solve the real problems, the following steps are generally followed

Step1 Description of original problem. The to be solved first stated mathematically and linguistically

Step2 Defining the thresholds of the variables. The values corresponding to the greatest and least degree of satisfaction are termed thresholds.

Step3 Fuzzy quantization .Base on the threshold values the membership functions are selected

Step4 Selections of the fuzzy operations, in terms of decision making process by human experts. The most commonly used operations are Mamdani's and Zadeh's.

4.12 Structure of Fuzzy Controller

A Fuzzy controller is similar to fuzzy controller comprising of fuzzification interface, a knowledge base, an inference engine and defuzzification engine

1) The fuzzification engine performs following functions

- Measures the values of input variables
- Performs a scale mapping that transfers the range of values of input variables into corresponding universes of discourse
- Performs the functions of fzzification that converts input data into suitable linguistic values

2) Knowledge Base it comprises of date base and rule base

3)Decision making logic operations based on fuzzy concepts and of inferring fuzzy control actions employing Fuzzy implications and rules of inference in Fuzzy logic

4)The defuzzification interface yields a non Fuzzy control action from an inferred fuzzy control action . It converts the range of values of output variables into corresponding universes of discourse.

4.13 Implementation of Fuzzy Controller using Fuzzy Logic Tool Box in Simulink.

Using GUI tools of Fuzzy Logic Tool Box of Simulink , fuzzy inference system of this Pd controller has been constructed. To start this system , we type fuzzy at the MATLAB promt. The generic untitled FIS Editor opens, with one input, labeled input1 , and one output, labled output1. Here we select two input and one output .The two input are error and derror and the output was named Ref. Torque

The steps to Implement the Controller

1) Assign the names to inputs and output
2) Select the membership functions and assign their ranges using Membership functions in the edit window of the editor.
3) Give rule base to the controller by selecting Rules in the Edit window
4) Now the design of the controller is complete. Export the file either to Disk or Workspace.

There are two controller designed in order to Control the system are

1) pendulum1

Inputs: error and d error

Output: output1

2) displacement2

Inputs: Trial and Error

Output : Ref. Torqu

4.14 Details Of The (Proportional And Derivative) Controller For Control Of Angular Position Of Robot Arm ’Pendulum1

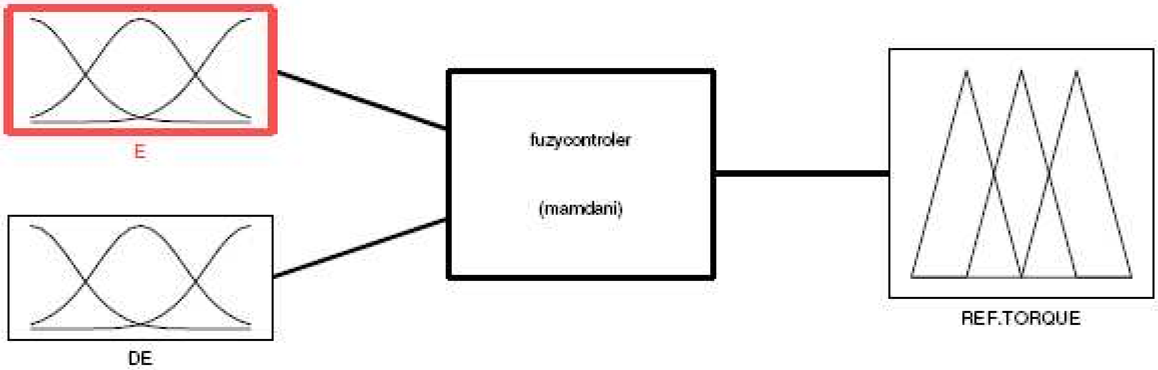


Fig 4.4 Block Diagram Of Controller To Control The Angular Position Of Robot Arm

Fig 4.4 Shows the block diagram of the controller to control the angular position of Robot Arm which has two input the error and the change of derror and the output1 is the Ref. Torque generated to control the angular position of the system of the system.

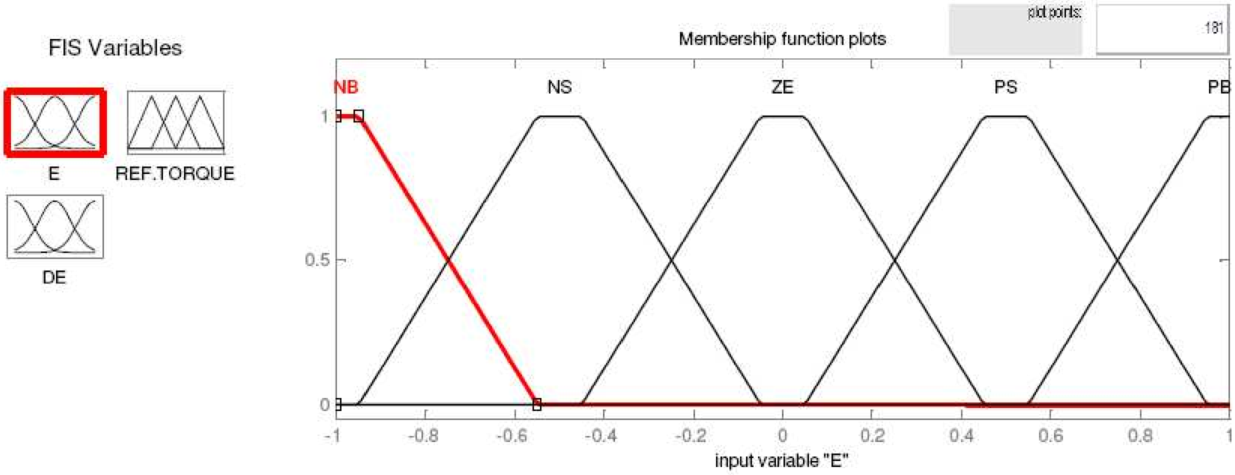


Fig4.5 Membership Function for Input1 (Error)

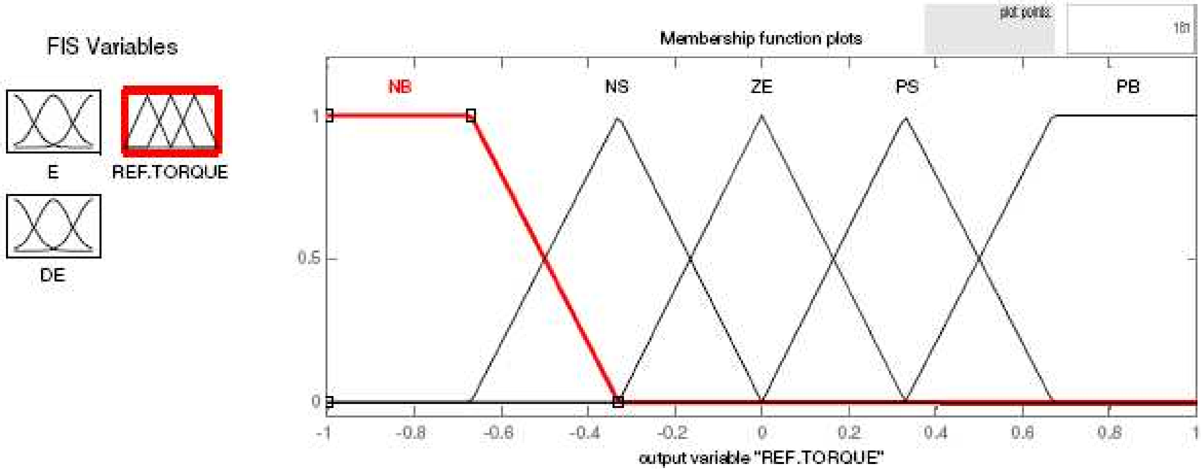


Fig 4.6 Membership Function of output

Fig 4.3, Fig 4.5, Fig 4.6 Gives the type and the range of the membership function applied as two inputs and the output, to design the Fuzzy Logic controller to control the angular position of Robot Arm

4.15 Rule Base for the Controller

The rule base for implementing fuzzy PD controller for angular position control of guided Robot Arm was developed using IF THEN relationship and rule base edit window of Simulink. The sample of rule base for fuzzy PD controller to control the angle Robot Arm is given as under:

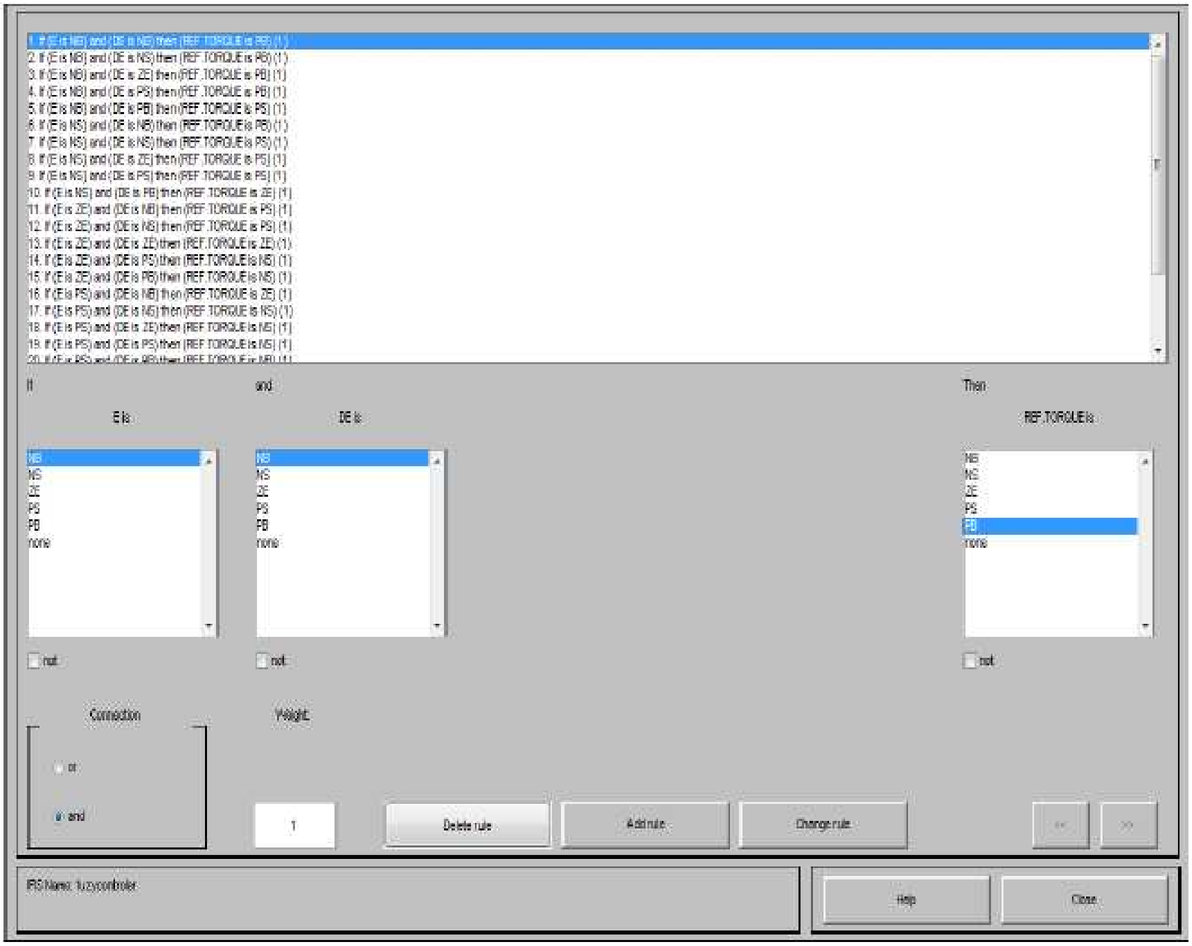
The obtained IF-THEN rule base is shown in Table 1. Let's look at examples as follows:

1. If e is NB and $\dot{e}$ is PB then T is PS.
2. If e is NB and $\dot{e}$ is PS then T is PB.
3. If e is NB and $\dot{e}$ is Z then is PB.

Table 1. Fuzzy IF-THEN rule base

	Derivative of Position Error				
Position Error	**NB**	**NS**	**Z**	**PS**	**PB**
NB	PB	PB	PB	PB	PS
NS	PB	PS	PS	PS	Z
Z	PS	PS	Z	NS	NS
PS	Z	NS	NS	NS	NB
PB	NS	NB	NB	NB	NB

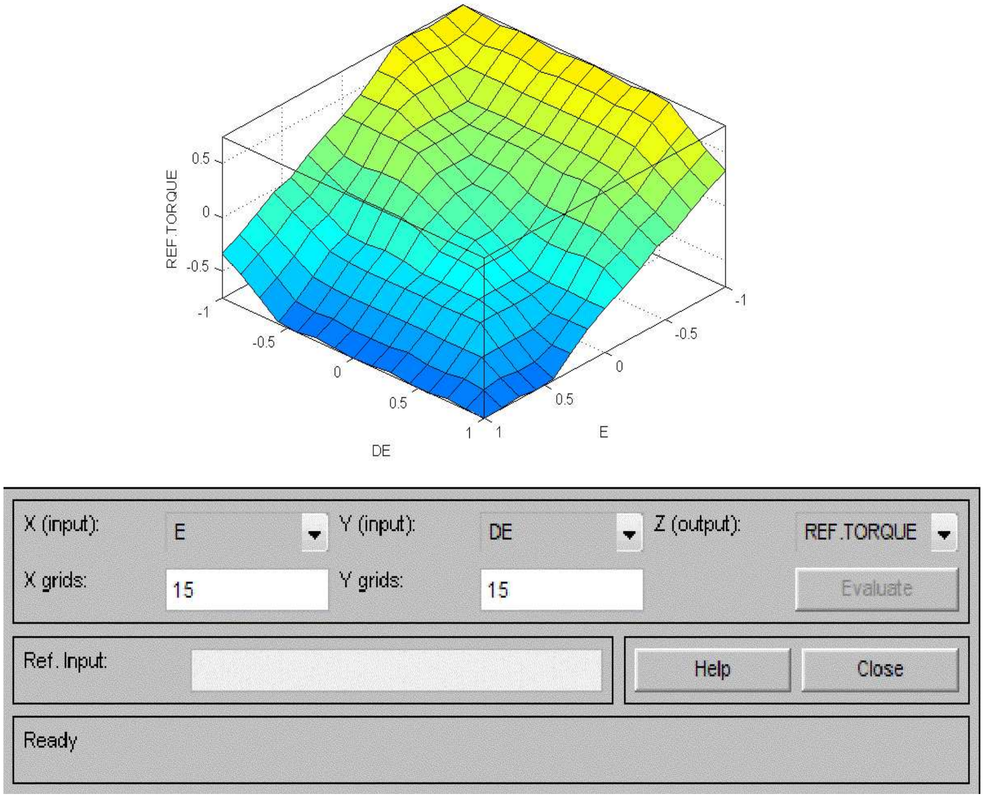
Rule Base:-



1. If (E is NB) and (DE is NB) then (REF.TORQUE is PB) (1)
2. If (E is NB) and (DE is NS) then (REF.TORQUE is PB) (1)
3. If (E is NB) and (DE is ZE) then (REF.TORQUE is PB) (1)
4. If (E is NB) and (DE is PS) then (REF.TORQUE is PB) (1)
5. If (E is NB) and (DE is PB) then (REF.TORQUE is PS) (1)
6. If (E is NS) and (DE is NB) then (REF.TORQUE is PB) (1)
7. If (E is NS) and (DE is NS) then (REF.TORQUE is PS) (1)
8. If (E is NS) and (DE is ZE) then (REF.TORQUE is PS) (1)
9. If (E is NS) and (DE is PS) then (REF.TORQUE is PS) (1)
10. If (E is NS) and (DE is PB) then (REF.TORQUE is ZE) (1)
11. If (E is ZE) and (DE is NB) then (REF.TORQUE is PS) (1)
12. If (E is ZE) and (DE is NS) then (REF.TORQUE is PS) (1)
13. If (E is ZE) and (DE is ZE) then (REF.TORQUE is ZE) (1)
14. If (E is ZE) and (DE is PS) then (REF.TORQUE is NS) (1)
15. If (E is ZE) and (DE is PB) then (REF.TORQUE is NS) (1)
16. If (E is PS) and (DE is NB) then (REF.TORQUE is ZE) (1)
17. If (E is PS) and (DE is NS) then (REF.TORQUE is NS) (1)
18. If (E is PS) and (DE is ZE) then (REF.TORQUE is NS) (1)
19. If (E is PS) and (DE is PS) then (REF.TORQUE is NS) (1)
20. If (E is PS) and (DE is PB) then (REF.TORQUE is NB) (1)

FIS Name: fuzzycontroler

Fig 4.7 Rule Base Of Robot Arm



X (input): E
Y (input): DE
Z (output): REF.TORQUE
X grids: 15
Y grids: 15
Ref. Input:
Ready

Fig 4.8 A control surface of max-min using trapezodial MF.

There are total 25 rules in Fuzzy Proportional and Derivative Controller to control the angular position of Robot Arm in FIS file **‘pendulum1’**

Complete Description of the FIS **‘pendulum1’**

```
[System]
Name=’pendulum1’
Type=’mamdani’
Version=2.0
NumInputs=1
NumOutputs=-1
NumRules=25
AndMethod=’min’
OrMethod=’max’
```

```
ImpMethod=’min’
AggMethod=’max’
DefuzzMethod=’som’

[Input1]
Name=’error’
Range= [-1.57 1.57]
NumMFs=5
MF1=’nb’:’trapmf’ [-Inf –Inf -1.57 -0.785]
MF2=’ns’:’trimf’ [-1.57 -0.785 0]
MF3=’zero’:’trimf’ [-0.785 0 0.785]
MF4=’ps’:’trimf’ [0 0.785 1.57]
MF5=’pb’:’trapmf’ [0.785 1.57 Inf Inf]

[Input2]
Name=’derror’
Range= [-0.785 0.785]
NumMFs=5
MF1=’nb’:’trapmf’ [-Inf –Inf -0.785 -0.3925]
MF2=’ns’:’trimf’ [-0.785 -0.3925 0]
MF3=’zero’:’trimf’ [-0.3925 0 0.3925]
MF4=’ps’:’trimf’ [0 0.3925 0.785]
MF5=’pb’:’trapmf’ [0.3925 0.785 Inf Inf]

[Ref.Torque]
Name=’output1’
Range= [-5 5]
NumMFs=5
MF1=’nb’:’trimf’ [-7.5 -5 -2.5]
MF2=’ns’:’trimf’ [-5 -2.5 0]
MF3=’zero’:’trimf’ [-2.5 0 2.5]
```

```
MF4=’ps’:’trimf’ [0 2.5 5]
MF5=’pb’:’trimf’ [2.5 5 7.5]
```

The rule base of FIS ‘displacement2’ contains 9 rules to control the horizontal position of vehicle on which Arm is installed.

Complete Details of FIS **‘displacement2’**

```
[System]
Name='displacement2'
Type='mamdani'
Version=2.0
NumInputs=2
NumOutputs=1
NumRules=9
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='som'

[Input1]
Name='displacement'
Range= [-1 1]
NumMFs=3
MF1='n':'trapmf' [-Inf -Inf -0.7 0]
MF2='z':'trimf' [-0.1 0 0.1]
MF3='p':'trapmf' [0 0.8 Inf Inf]

[Input2]
Name='Trial'
Range= [-1 1]
NumMFs=3
MF1='n':'trapmf' [-Inf -Inf -1 0]
```

```
MF2='z':'trimf' [-0.1 0 0.1]
MF3='p':'trapmf' [0 1 Inf Inf]

[Output1]
Name='force'
Range= [-1 1]
NumMFs=7
MF1='nb':'trimf' [-1.4 -1 -0.6]
MF2='nm':'trimf' [-1 -0.6 -0.2]
MF3='ns':'trimf' [-0.4 -0.2 0]
MF4='z':'trimf' [-0.2 0 0.2]
MF5='pm':'trimf' [0.2 0.6 1]
MF6='ps':'trimf' [0 0.2 0.4]
MF7='pb':'trimf' [0.6 1 1.4]
```

4.11 Conclusion

The chapter presents the stepwise design of fuzzy logic controllers. It also includes the procedure of implementing fuzzy controller in MATLAB simulink. The details of the controllers and their functional analysis of is also done

The chapter presents the stepwise design of fuzzy logic controllers. It also includes the procedure of implementing fuzzy controller in MATLAB simulink. The details of the controllers and their functional analysis of also done.

CHAPTER 5

5. DESIGN AND SIMULATION IN SIMULINK/MATLAB7.01

5.1 INTRODUCTION

The aim of simulation is to develop complete model of the physical system and to analyze the system in different ways before going to implement it practically. In my dissertation control of robot arm is analyzed with different controllers such as Conventional and Intelligent controllers. In this chapter design and development of simulink model for robot Arm, Conventional controllers and Fuzzy Logic controllers are explained.

5.2 ROBOT ARM MODEL

$$[M]\begin{Bmatrix}\ddot{\theta}\\ \ddot{q}\end{Bmatrix}+\begin{bmatrix}H_1+T_\mu(\dot{\theta})\\ H_2\end{bmatrix}+\begin{bmatrix}0 & 0\\ 0 & EI\int_0^l \phi''\phi''dx\end{bmatrix}\begin{Bmatrix}\theta\\ q\end{Bmatrix}+\begin{bmatrix}\mu_1 & 0\\ 0 & \mu_2\end{bmatrix}\begin{Bmatrix}\dot{\theta}\\ \dot{q}\end{Bmatrix}=\begin{Bmatrix}T\\ 0\end{Bmatrix}$$

…7.1

By equation 7.1, we can develop the $T_\mu(\dot{\theta})$ is the Coulomb friction torque. Now the equations of motion are obtained

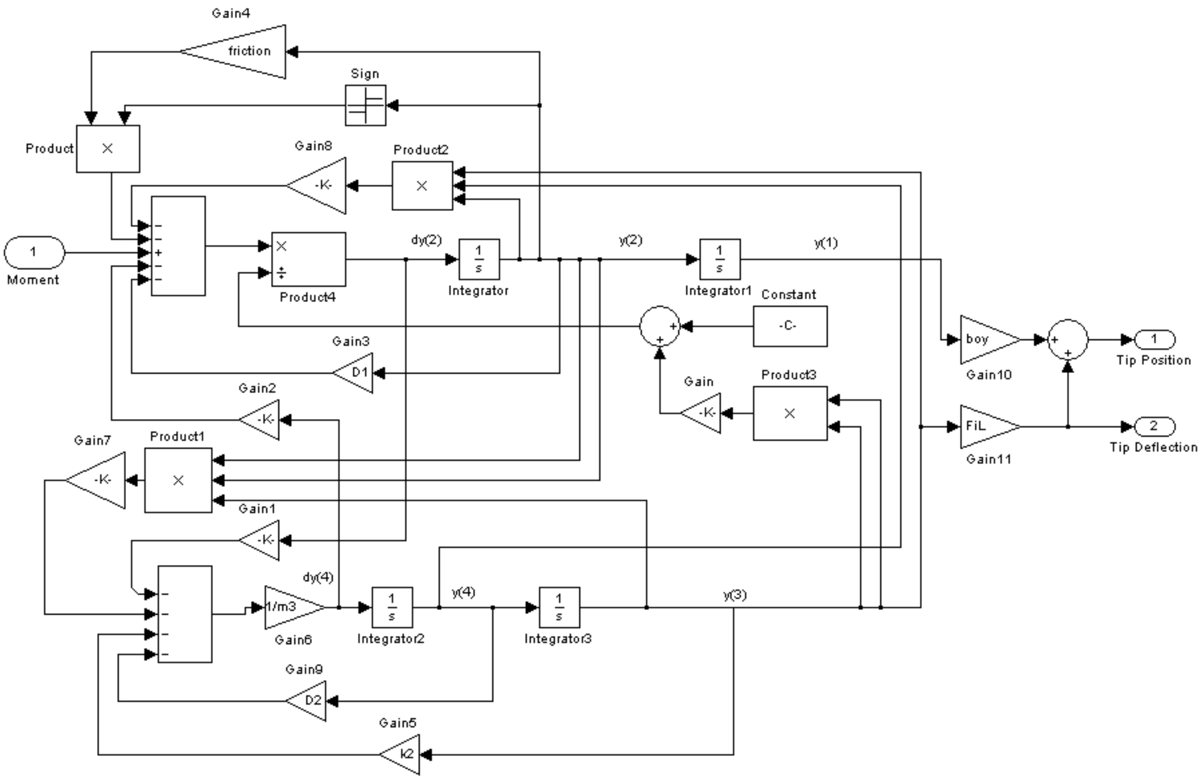


Fig .5.1 Model of a single flexible link robot arm manipulator

5.3DESIGN OF PID CONTROLLER

Equation for the PID controller is

$$U_i = P_i e + D_i \frac{de}{dt} + I_i \int e \qquad (i=1, 2\ldots6) \qquad \ldots 7.2$$

Where e is the error

P_i is the proportional gain

D_i is the differential gain

I_i is the integral gain

U_i is the controller output

The objective of designing PID controller is to find the P_i, D_i, I_i for the optimum response of the system

Hand tuning procedure for the tuning of the PID controller

a) remove all integral and differential action
b) tune the proportional gain or increase the proportional P_i to give the desired response ignoring any offset or peak over shoots
c) then tune the differential gain D_i (increase) until the oscillations are under the allowable range
d) tune the integral gain I_i (increase) until the until offset is in the allowable range
e) repeat this until P_i as large as possible

In designing considered that controller output should not more than 40 volts

Total system with PID controller is shown in Fig 5.2 and Response of system is shown in Fig 5.2

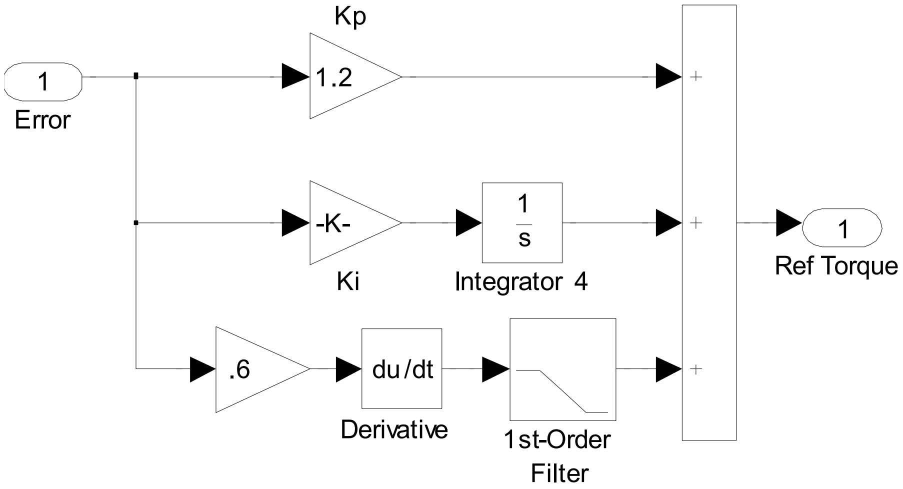


Fig 5.2 PID controller

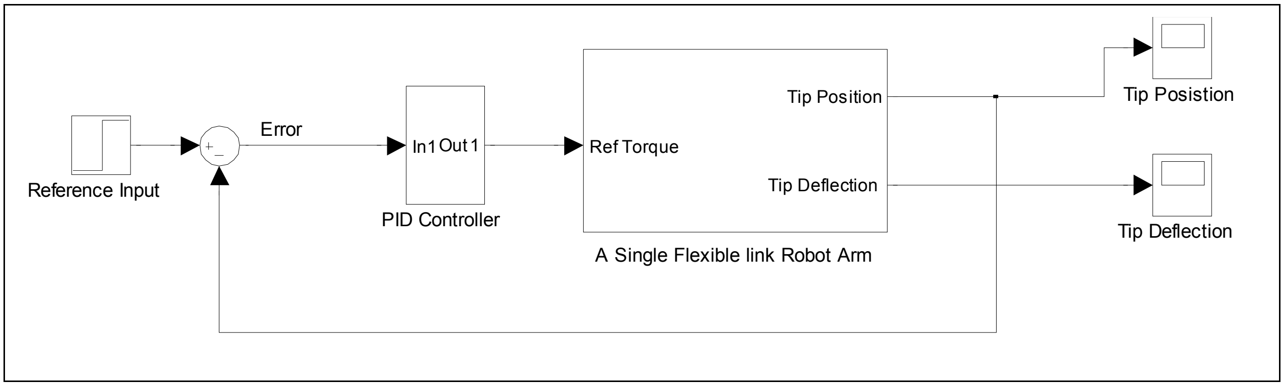


Fig 5.3 Total system with PID controller

5.4DESIGN OF PI CONTROLLER

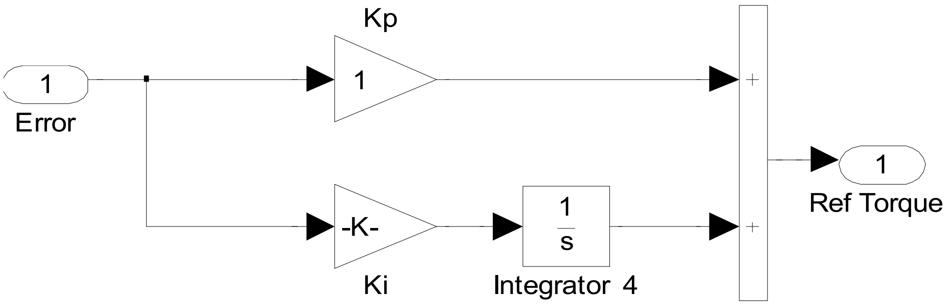


Fig 5.4 PI controller

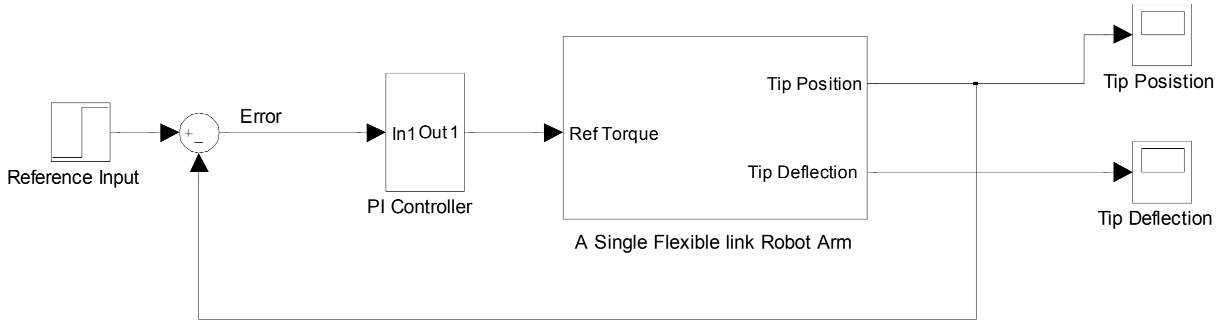


Fig 5.5 Total system with PI controller

5.5DESIGN OF FUZZY CONTROL

Total system with Fuzzy controller is shown in Fig 5.7 and Response of system is shown in Fig 5.7.Designing the Fuzzy controller in simulink consists of two steps

1. Designing the rule base
2. gain scheduling

Table1. Fuzzy IF-THEN rule base

Derivative of Position Error					
Position Error	**NB**	**NS**	**Z**	**PS**	**PB**
NB	**PB**	**PB**	**PB**	**PB**	**PS**
NS	**PB**	**PS**	**PS**	**PS**	**Z**
Z	**PS**	**PS**	**Z**	**NS**	**NS**
PS	**Z**	**NS**	**NS**	**NS**	**NB**
PB	**NS**	**NB**	**NB**	**NB**	**NB**

Design of rule base

Table 1 shows the rule base for the Fuzzy PD controller the rule base is to design as explained in the second chapter, but complex systems such as robot understanding the system behavior is very difficult so set of PD rules were proposed [6]. These rules generally used for the Fuzzy PD controller

5.6 Gain scheduling

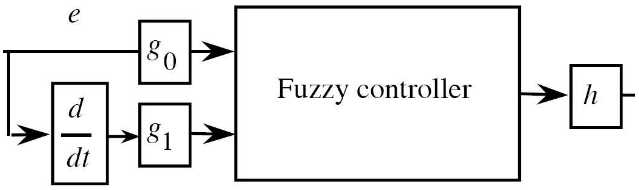


Fig 5.6 Fuzzy control with gains

Gain scheduling means designing of g_0, g_1 and *h for* the optimum response of the system Gain scheduling procedure for the Fuzzy controller

1. Initially put g_0=0 and increase g_1 until the controller gives the output normally, when the signal after the gain g_1 crosses the universe of discourse the there will not be any rule to processes then controller then the output will be zero before this happens previously designed gain will be the optimum gain for the g_1
2. increase h until the controller will gives the maximum output ,that will be the maximum controller output
3. then increase g_0 until overshoots under the allowable range

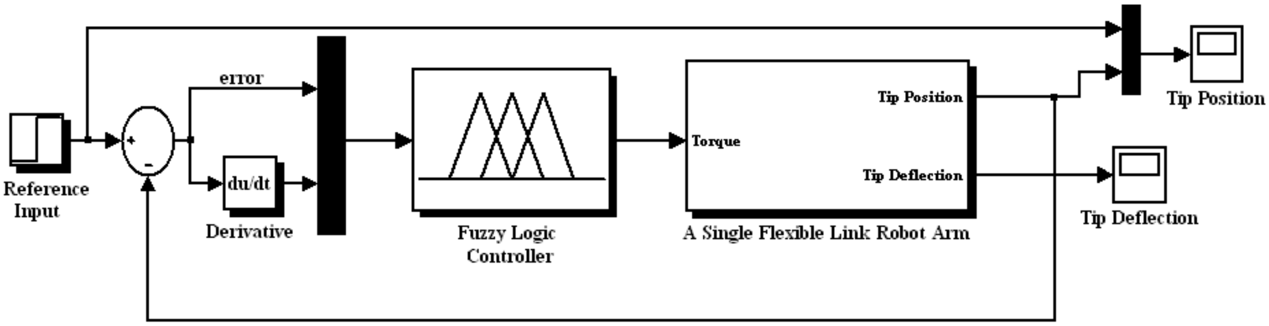


Fig 5.7 System with fuzzy controller

CHAPTER 6

RESULTS

6. 1 Tip position response of a single flexible link robot arm with PI control

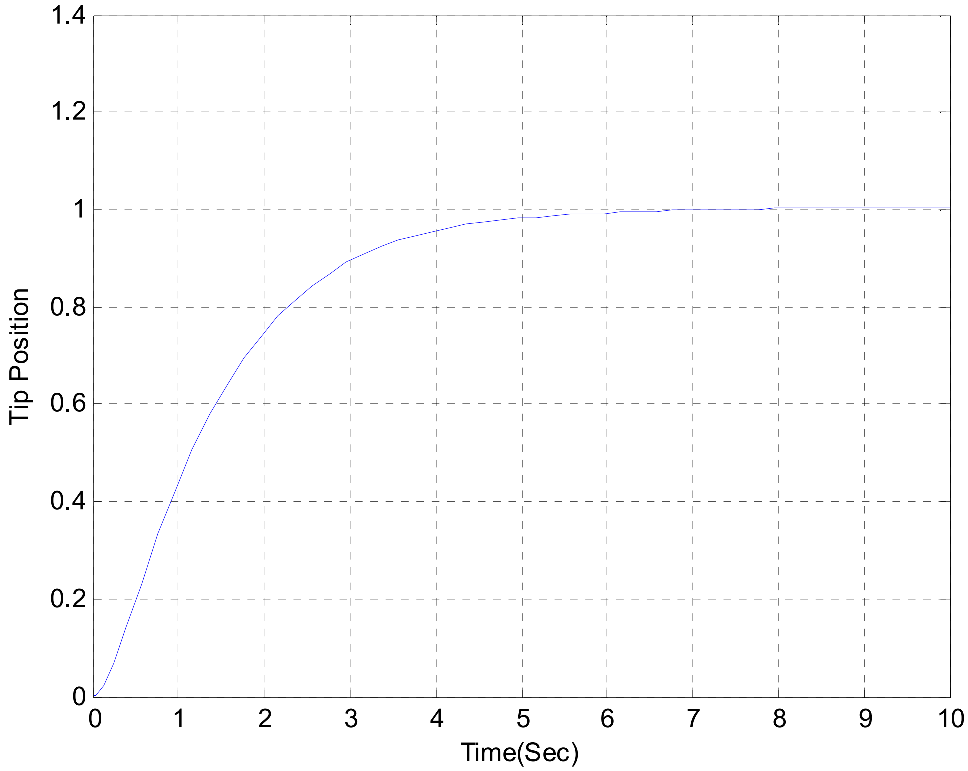


Fig. 6.1 Tip position response of a single flexible link robot arm with PI control

Tip position response of a single flexible link robot arm system with PI control is shown in Figure 6.1 From these figures a step input is applied to the system as a reference input. Desired response of the system with using PI controller is to get the tip position of a single flexible link robot arm to this reference position. PI controller using in such system which has nonlinear vibrations gives a result as tip position control of a single flexible link robot arm. Settling time of system is approximately 5.9 sec. shown in Fig. 6.1 It is observed as acceptable result and there is no maximum overshoot in response of tip position control. According to these results, suitable performance of PI controller is determined for tip position control of a single flexible link robot arm system.

6. 2 Tip deflection response of a single flexible link robot arm with PI control

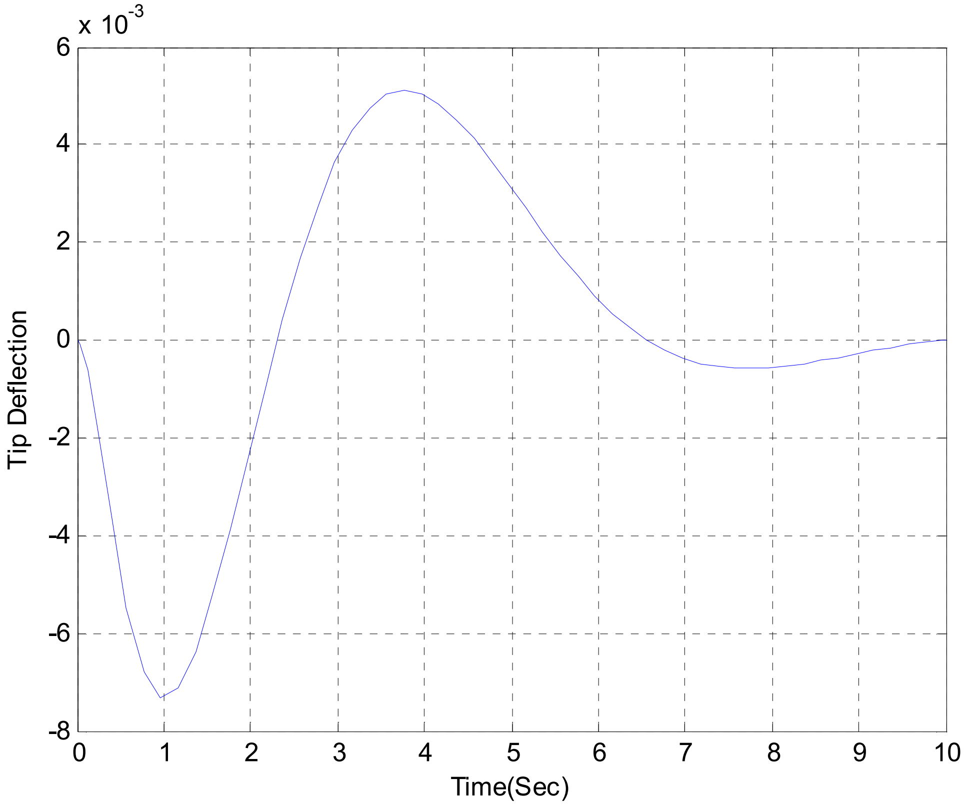


Fig. 6.2 Tip deflection response of a single flexible link robot arm with PI control

Tip deflection response of a single flexible robot arm system is shown in Figure 6.2 In this figure we can see tip deflection response of flexible link is zero approximately .8 seconds after initial position Is 3 vibrations until approximately 2.7 Sec Motion of flexible link has begun with nonlinear vibrations until when response of tip position reached 9.9 second. After that the maximum overshoot appears and then system goes to stable position. Tip deflection response depends on tip position control that can be seen Figure 6.1 and Figure 6.2 Tip deflection control is supplied properly when tip position of flexible link can be controlled with minimum time. We can say from Figure 6.2 that PI controller designed for tip position control of a single flexible link robot arm is accomplished. PI control can be used in as closed-loop controller to control such system and prove that it can be applied to nonlinear systems with suitable results

6.3 Tip position response of a single flexible link robot arm with PDI control

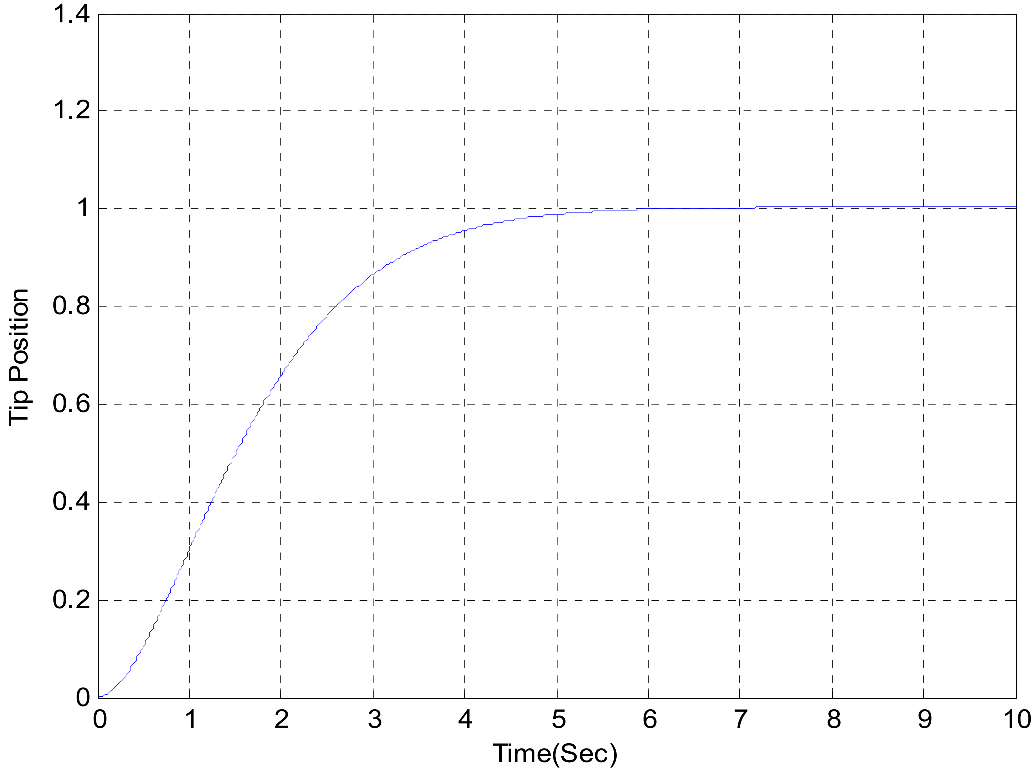


Fig. 6.3 Tip position response of a single flexible link robot arm with PDI control

Tip position response of a single flexible link robot arm system with PDI control is shown in Figure 6.3 From these figures, a step input is applied to the system as a reference input. Desired response of the system with using PDI controller is to get the tip position of a single flexible link robot arm to this reference position in minimum time range. PDI controller using in such system which has nonlinear vibrations gives a result as tip position control of a single flexible link robot arm. Settling time of system is approximately 5.5 sec. shown in Fig. 6.3 It is observed as acceptable result and there is no maximum overshoot in response of tip position control. According to these results, suitable performance of PID is determined for tip position control of a single flexible link robot arm system.

6. 4 Tip deflection response of a single flexible link robot arm with PID contro

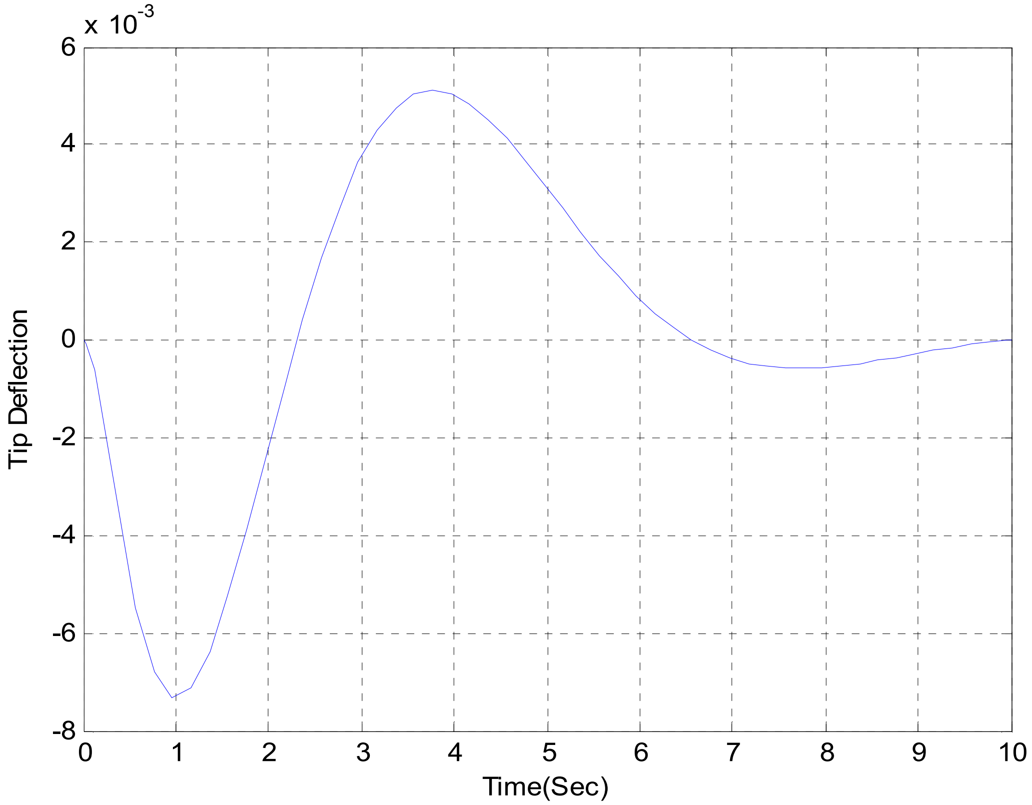


Fig. 6 .4 Tip deflection response of a single flexible link robot arm with PID control

Tip deflection response of a single flexible robot arm system is shown in Figure 6.4 In this figure we can see tip deflection response of flexible link is zero approximately .9 seconds after initial position Is 3 vibrations until approximately 2.7 Sec Motion of flexible link has begun with nonlinear vibrations until when response of tip position reached 9.7 second. After that the maximum overshoot appears and then system goes to stable position. Tip deflection response depends on tip position control that can be seen Figure 6.3 and Figure 6.4 Tip deflection control is supplied properly when tip position of flexible link can be controlled with minimum time. We can say from Figure 6.3 that PID controller designed for tip position control of a single flexible link robot arm is accomplished. PID can be used in as closed-loop controller to control such system and prove that it can be applied to nonlinear systems easily with suitable results

6.5 Tip position response of a single flexible link robot arm with fuzzy logic control

Fuzzy Logic Controllers is designed to control nonlinear vibration of a single flexible link robot arm and the effects of the controllers over the system are examined. The digital simulations and graphics are realized by MATLAB/Simulink software programme. The exact mathematical model of the system and designed fuzzy logic controller are used into simulations

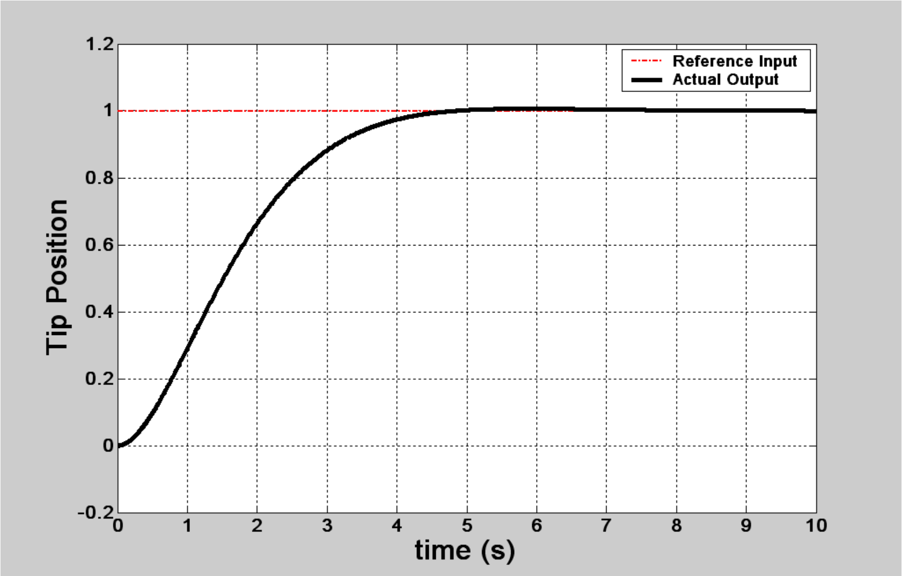


Fig. 6.5 Tip position response of a single flexible link robot arm with fuzzy logic control.

Tip position response of a single flexible link robot arm system with fuzzy logic control is shown in Figure 6.5 From these figures, a step input is applied to the system as a reference input. Desired response of the system with using fuzzy logic controller is to get the tip position of a single flexible link robot arm to this reference position in minimum time range. Fuzzy logic controller using in such system which has nonlinear vibrations gives a good result as tip position control of a single flexible link robot arm. Settling time of system is approximately 5 s. shown in Fig. 6.5 It is observed as acceptable result and there is no maximum overshoot in response of tip position control. According to these results, suitable performance of fuzzy logic controller is determined for tip position control of a single flexible link robot arm system. Finally fuzzy logic controller designed is established properly and this controller can be used for such kind of system.

6.6 Tip deflection response of flexible link robot arm with fuzzy logic control

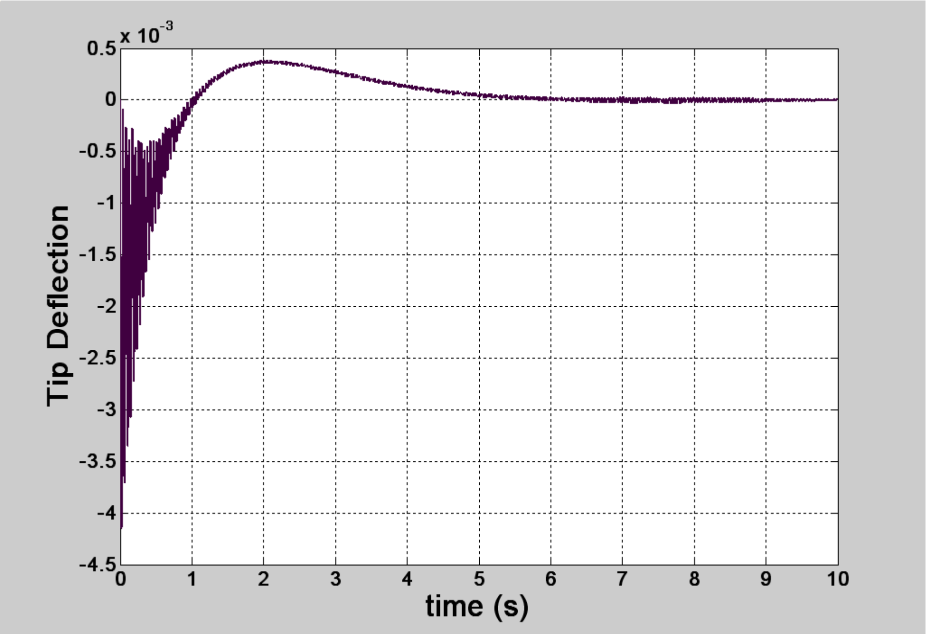


Fig. 6.6 Tip deflection response of flexible link robot arm with fuzzy logic control

Tip deflection response of a single flexible robot arm system is shown in Figure 6.6 In this figure we can see tip deflection response of flexible link is zero approximately 5 seconds after initial position. Motion of flexible link has begun with nonlinear vibrations until when response of tip position reached 1 second. After that the maximum overshoot appears and then system goes to stable position. Tip deflection response depends on tip position control that can be seen Figure 6.5 and Figure 6.6 Tip deflection control is supplied properly when tip position of flexible link can be controlled with minimum time. We can say from Figure 6.6 that fuzzy logic controller designed for tip position control of a single flexible link robot arm is accomplished. Fuzzy logic control can be used in as closed-loop controller to control such system and prove that it can be applied to nonlinear systems easily with suitable results.

CHAPTER 8

CONCLUSIONS

From the simulated results, we conclude the following things.

The main contribution of this paper is concentrated on fuzzy logic control approach for nonlinear vibration control of a single flexible-link robot arm. Fuzzy logic controller is designed to terminate nonlinear vibrations which effect motion of flexible link robot arm. A flexible link robot arm model that accurately predicts the link's motion for desired tip position and small deflections was developed. Based on the model developed FLC was designed to control the position of the tip of a single flexible robot arm and its performance was considered.

In this study, fuzzy controller for a single flexible link robot arm with no payload attached was successfully developed. However, performance of the fuzzy logic controller for a single flexible link was satisfactory. We created a desired input for control system as a reference input. Aim of this, controller needs a reference input to create control force to flexible link. As a result of it this controller did this control action well. In future studies, we can deal with to control the multi flexible link

FUTURE SCOPE OF WORK

Main drawback of hand tuning of PI, PID controller may not give good response In Tip position And Tip deflection response of flexible link robot arm The quality of a fuzzy logic controller can be drastically affected by the choice of membership functions and gains. Thus**,** methods for tuning fuzzy logic controllers are necessary. Here we have used hand tuning to select gains and general triangle membership are used which may good performance. The fuzzy logic controller for a single flexible link was satisfactory.

REFERENCES

1. R.K.Mittel, I.J.Nagrath, "**Robotics and Control**", Tata McGraw Hill publishing Company Limited, NEW DELHI.
2. Brian Armstrong, Oussama Khatib, Joel Burdick, "**The Explicit Dynamic Model and Inertial Parameters of the PUMA 560 Arm**", Stanford Artificial Intelligence Laboratory Stanford University, IEEE Transactions and Systems 1986.
3. P.K.Dash, S.K.Panda, T.H.Lee, J.X.Xu, A.Routray, "**Fuzzy and Neural Controllers for Dynamic Systems: an Overview",** Power Electronics and Drive Systems, 1997. Proceedings, 1997 International Conference on Vol 2, 26-29 May 1997, Page(s):810 - 816.
4. C.H.Atkeson, J.D.Griffiths, J.M.Hollerbach, C.H.An, "**Experimental Evaluation of feed forward and computed torque control",** Robotics and Automation, IEEE Transactions on Volume 5, Issue 3, June 1989 Page(s):368 – 373.
5. D.P.kwok, T.P.Leung, Fang Sheng, "**Genetic Algorithms for Optimal Dynamic Control of Robot Arms**", International Conference on Industrial Electronics, Control, and Instrumentation, 1993. Proceedings of the IECON '93, vol.1, 15-19 Nov. 1993 Page(s):380 – 385.
6. Han-Xiong Li; H.B.Gatland, "**Conventional Fuzzy Control and its Enhancement"**, Systems, Man and Cybernetics, Part B, IEEE_Transactions on Volume 26, Issue 5, Oct. 1996 Page(s):791-797.
7. T.Brehm, K.S.Rattan, **"Hybrid fuzzy logic PID controller",** Fuzzy Systems, 1994. IEEE World Congress on Computational Intelligence, Proceedings of the Third IEEE Conference, vol.3, 26-29 June 1994 Page(s):1682 – 1687.

8. G.M.Khoury, M.Saad, H.Y.Kanaan ,and C.Asmar, “**Fuzzy PID Control of a Five DOF Robot Arm**”, Journal of Intelligent and Robotic systems, Vol. 40 , Issue 3, July 2004, Pages: 299 - 320 .

9. Abdollah Homaifar, Ed McCormick, “**Simultaneous Design of Membership Functions and Rule sets for Fuzzy Controllers Using Genetic Algorithms**”, IEEE Transactions on Fuzzy Systems, Vol. 3, No. 2, May 1995.

10. T.Ozaki, T.Suzuki, T.Furuhashi, S.Okuma, Y.Uchikawa, “**Trajectory Control of Robotic Manipulators Using Neural Networks”**, Industrial Electronics, IEEE Transactions on Volume 38, Issue 3, June 1991 Page(s):195 – 202.

11. P.Gupta and N.K.Sinha, "**Control of Robotic Manipulator - a Neural** Network **Approach**", International Journal of System Science, vol.29, no.7, pp.723-730, 1998.

12. M. F.Moller, “**A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning**”, Neural networks, vol. 6, pp.525-533, 1993.

13. J.-S.R.Jang, “**Fuzzy Controller Design Without Domain Experts**”, IEEE International Conference on Fuzzy Systems, 8-12 March 1992 Page(s):289 – 296.

14. Jyh-Shing, Roger Jang, “**ANFIS: Adaptive-Network-Based Fuzzy Inference System**”, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 23, No.3, May/June 1993.

15. G.S.Sandhu, K.S.Rattan, “**Design of a Neuro-Fuzzy Controller**”, Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation, 1997 IEEE International Conference on Volume 4, 12-15 Oct. 1997 Page(s):3170 – 3175.

16. Manish Kumar, Devendra P.Garg, “**Intelligent Learning of Fuzzy Logic Controllers via Neural Network and Genetic Algorithm”,** Proceeding of 2004 JUSFA, 2004 Japan-USA Symposium on Flexible Automation, Denver, Colorado, July 19-21, 2004.

17. Teo Lian Seng, Marzuki Bin Khalid, "**Tuning of a Neuro-Fuzzy Controller by Genetic Algorithm**", IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 29, No. 2, April 1999.