The Islamic University of Gaza
Deanery of Graduate Studies
Faculty of Engineering
Electrical Engineering Department

الجامعة الاسلامية – غزة
عمادة الدراسات العليا
كلية الهندسة
قسم الهندسة الكهربائية

Modeling and Control of 5DOF Robot Arm Using Supervisory Control

Ahmed Zakari Alassar

Advisors
Dr. Hatem Elaydi
Dr. Iyad Abuhadrous

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical Engineering

March 2010
1431 ربيع أول

ABSTRACT

Modeling and control of 5 degree of freedom (DOF) robot arm is the subject of this thesis. The modeling problem is necessary before applying control techniques to guarantee the execution of any task according to a desired input with minimum error. Deriving both forward and inverse kinematics is an important step in robot modeling based on Denavit Hartenberg (DH) representation.

The main objective of this thesis is to control a robot arm using three controllers to acquire the desired position. Proportional integral derivative (PID) controller is used as a reference benchmark to compare its results with fuzzy logic controller (FLC) and fuzzy supervisory controller (FSC) results. FLC is applied as a second controller because of the nonlinearity in the robot manipulators. We compare the result of the PID controller and FLC results in terms of time response specifications. FSC is a hybrid between the previous two controllers. The FSC is used for tuning PID gains since PID alone performs not satisfactory in nonlinear systems. Hence, comparison of tuning of PID parameters is utilized using classical method and FSC method. Based on simulation results, FLC gives better results than classical PID controller in terms of time response and FSC is better than classical methods such as Ziegler-Nichols (ZN) in tuning PID parameters in terms of time response.

ملخص

التحكم المراقب لجهازِ روبوت من ذوي الخمسةِ مفاصل

تُعنى هذه الرسالةُ بمشاكلِ النمذجةِ والتحكمِ لذراعٍ آلي من ذوي الخمسةِ مفاصلٍ. ونمذجةُ الروبوتِ ضروريةٌ قبلَ تطبيقِ انظمةِ التحكمِ وذلك لضمانِ تنفيذِ المهمةِ المطلوبةِ وفقاً للمدخلاتِ بأقلِ نسبةِ خطأٍ ممكنةٍ. وعندَ نمذجةِ الروبوتِ فانّ اشتقاقَ كُلٍ من الحركةِ الأماميةِ والحركةِ الخلفيةِ لمحاورِ الروبوتِ هو خطوةٌ اساسيةٌ استناداً الى طريقة (Denavit Hartenberg).

إن الهدفَ الرئيسَ لهذه الرسالةِ هو التحكمُ بذراعِ الربوتِ باستخدامِ ثلاثةِ متحكماتٍ للوصولِ الى الهدفِ المطلوبِ، حيثُ تمَّ استخدامُ المتحكمِ التناسبي التكاملي التفاضلي (PID) كأساسٍ مرجعيٍّ لمقارنةِ نتائجهِ بنتائجِ المتحكمِ المنطقي الغامض (FLC) والمتحكمِ الإشرافي الغامضِ (FSC). وقد تمَّ تطبيقُ المتحكمِ FLC كمتحكمٍ ثانٍ نظرَاً لعدم خطيةِ ذراعِ الروبوتِ، ومن ثمَّ تمّتْ مقارنةُ نتائجَ هذا المتحكمِ بنتائجِ المتحكمِ PID بدلالةِ مقياسِ استجابةِ الزمنِ. لقد تمَّ تصميمُ متحكمٍ ثالثٍ FSC يجمعُ بينَ كلا المتحكمينِ السابقينِ، ويهدفُ الى ضبطِ معاملاتِ المتحكمِ PID بواسطةِ المتحكمِ FLC حيثُ إن المتحكمَ PID يُعطي نتائجَ غيرَ جيدةٍ في حالةِ الأنظمةِ الغيرِ خطيةٍ. وبالتالي سوفَ نقومُ بمقارنةِ النتائجِ التي تمَّ الحصولُ عليها باستخدامِ المتحكمِ FSC بالنتائج التي تمَّ الحصولُ عليها باستخدامِ اساليبِ الضبطِ التقليديةِ. واستناداً الى نتائجِ المحاكاةِ، فانّ هذه الرسالةَ تبينُ أنَ استخدامَ كلٍ من المتحكمين FLC و FSC يُعطي نتائجَ جيدةً مقارنةً بالنتائجِ التي تمَّ الحصولُ عليها باستخدامِ المتحكمِ PID.

بسم الله الرحمن الرحيم

Dedicated to

My Father
Dr. Zakari Ahmed Alassar

ACKNOWLEDGEMENT

I thank Allah, the lord of the worlds, for His mercy and limitless help and guidance. My peace and blessings be upon Mohammed the last of the messengers.

I would like to express my deep appreciation to my thesis advisor Dr. Hatem Elaydi for providing his advice and encouragement. Also, I am heavily indebted to my supervisor Dr. Iyad Abu Hadrous for his excellent guidance, the warm discussion and regular meeting I had with him during research. I would like to thank the other committee members Dr. Assad Abu-Jasser and Dr. Basel Hamed for providing their valuable suggestions. In addition, my thanks go to my friends and colleagues for their advices and supports.

There are no words that describe how grateful I am to my parents Dr. Zakari Alassar and my mother for raising me and for their support and encouragement through years. My deepest thanks go to my brothers Mohammed and Yahiya and my sister. Lastly, I am grateful to my wife and my kids Yusuf and Zakaria for their patience and understanding during my busy schedule.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

$r, r(t)$	Reference input
$e, e(t)$	Error between the input signal and the output
$u, u(t)$	Input applied by the controller to plant
$y, y(t)$	Output of the closed loop control system
$G_p(s)$	Plant transfer function
$G_c(s)$	Controller transfer function
$G_d(s)$	Disturbance input to output transfer function
$H(s)$	Feedback measurement
$D_t(s)$	Disturbance input
$v_a(t)$	Input source voltage, [Volt]
$v_b(t)$	The Back EMF
$i_a(t)$	Armature current, [Ampere]
R_a	Armature resistance, [Ohm]
L_a	Electric inductance, [H]
τ_m	Motor torque, [Nm]
K_m	Proportional constant
K_b	Back EMF constant, [V/ms^{-1}]
K_t	Motor torque constant, [Nm/A]
J_m	Moment of inertia of the rotor, [Kgm^{-1}]
θ_m	Rotor position, [rad]
ϕ	Magnetic flux, [Weber]
B_m	Coefficient of motor friction
ω_m	Angular speed, [rad/sec]
$\theta_i(t)$	Joint angle for joint *i* of robot manipulator

gr	Gear ratio
q_i	Joint variable
$\theta_i, a_i, d_i, \alpha_i$	Denavit Hartenberg parameters
A_i	Transformation matrix for joint *i*
H_n^0	Total transformation matrix
T_i^{i-1}	Homogeneous transformation matrix of *i* relative to *i*-1
K_P	Proportional gain
K_D	Derivative gain
K_I	Integral gain
T_I	Integral time constant
T_D	derivative time constant
L	Ziegler-Nicholas apparent dead time
a	Ziegler-Nicholas time constant
K_{CR}	Critical gain
P_{CR}	Oscillation period
t_r	Rising time, [sec]
t_s	Settling time, [sec]
ζ	Damping ratio
ω_n	Natural frequency
π	Pi (constant = 3.14)
U	Universe of discourse
A, B, C	fuzzy sets
$\mu(u)$	Membership function
Φ	Null fuzzy set (Phi)
Δe	Change of the error

k_e	Scaling factor for error
k_{de}	Scaling factor for change of error
ku	Scaling factor for output
X	Symbolic name of linguistic variable
T(X)	A set of linguistic variable name
S_{X}	Function give interpretation of linguistic variable
n	Degree of freedom of robot manipulator (n-DOF robot manipulator)
i	Number of links
R	Number of the fuzzy rules

ABBREVIATIONS

BOA	Bisector of Area
COA	Centroid of Area
CL	Closed Loop
DOF	Degree of Freedom
DH	Denavit Hartenberg
DC	Direct Current
DFC	Direct Fuzzy Controller
EMF	Electromagnetic Force
FITA	First Inference Then Aggregation
FK	Forward Kinematic
FGS	Fuzzy Gain Scheduling
FLC	Fuzzy Logic Controller
FPD	Fuzzy Proportional Derivative Controller
FPI	Fuzzy Proportional Integral Controller
FSC	Fuzzy Supervisory Control
GMP	General Modus Ponens
GMT	General Modus Tollens
GUI	Graphical User Interface
IJC	Independent Joint Control
IFC	Indirect Fuzzy Controller
IK	Inverse Kinematic
LOM	Largest of Maximum
Max	Maximum
MM	Maximum Method
MOM	Mean of Maximum

MF	Membership Function
Min	Minimum
MIMO	Multi Input Multi Output
MISO	Multi Input Single Output
NN	Neural Network
OL	Open Loop
P	Prismatic
PID	Proportional Integrated Derivative
R	Revolute
SIMO	Single Input Multi Output
SISO	Single Input Single Output
SOM	Smallest of Maximum
TS	Takagi Sugeno
TF	Transfer Function
Z-N	Ziegler-Nicholas

CHAPTER 1 INTRODUCTION

1.1. Motivation

In recent years, industrial and commercial systems with high efficiency and great performance have taken advantages of robot technology. Large number of control researches and numerous control applications were presented during the last years, concentrated on control of robotic systems. Robot manipulator field is one of the interested fields in industrial, educational and medical applications. It works in unpredictable, hazard and inhospitable circumstances which human cannot reach. For example, working in chemical or nuclear reactors is very dangerous, while when a robot instead human it involves no risk to human life. Therefore, modeling and analysis of the robot manipulators and applying control techniques are very important before using them in these circumstances to work with high accuracy.

In Gaza strip, many industrial applications can utilize robot technology and develop robot manipulators. It is an attractive field to be applied and developed for industrial applications. This thesis is meant to be suitable for these applications. On the other side, some universities and colleges offers, some courses related to robotics. These courses mainly focus on the theoretical concepts without giving much attention for controlling different robot manipulators in the practical side. This thesis may be considered as a valuable educational tool in their laboratories.

The essential problem is to study the robot manipulator problem from two sides: the first one is the mathematical modeling of the manipulator and the actuators, which includes an analysis for the forward kinematic, the inverse kinematic and modeling the direct current (DC) motor because it is an important issue in a robot manipulator. The second problem is the control of the robot manipulator.

The main objective of this thesis is concerned with designing a controller for the motion of the robot manipulator to meet the requirement of the desired trajectory input with suitable error and disturbance values. The motivation of control technique designs the usage of the high precision performance of the robot manipulators in complicated and hazardous environments.

Various controllers have been designed and applied in the robot manipulator. The first question that may arise is the different types of these controllers and the difference between these controllers in terms of best performance will be shown.

Proportional Integral Derivative (PID) controller may be the most widely used controller in the industrial and commercial applications for the early decades, due to its simplicity of designing and implementation, so the first attempt is to apply PID control; however, PID does not give optimal performance due to the nonlinear elements. Robot manipulators are classified as nonlinear systems, so classical controllers are not sufficient to give the best results. Fuzzy logic controller (FLC) was found to be an efficient tool to control nonlinear systems. Designing and testing FLC will be shown as a second option.

In recent years, hybrid between fuzzy and classical controllers has combined to design a controller such as fuzzy plus PID and fuzzy logic supervisory (FLS) creates more appropriate solution to control robot manipulator. Through the thesis, FLC is considered as an important controller for on-line tuning of PID parameters. FLC may design to monitor and enhance the PID parameters online.

The robot movements' analysis is important before the implementation of the actual system in order to prevent possible environmental hazards. Therefore, computer simulations are important to perform any controller, where developing distinct mathematical model for any robot manipulator is an important issue to perform the simulations.

1.2. Background

According to the RIA[1], a robot is a “reprogrammable multifunctional manipulator designed to move material, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks” [1]. Through this thesis, the term “robot manipulator” is used.

Robot manipulator is one of the motivating disciplines in industrial and educational applications, and an essential branch to control sciences because of its intelligent aspects, nonlinear characteristics, and its real time implementation. It was developed to enhance human’s work such as in the manufacturing or manipulation of heavy materials, and unpredictable environments. Whatever the kind of task robot manipulator may be provided with, robot performance measures the high quality and large quantity of work that it can do in the desired time and place. Robot manipulator has immeasurable tasks, so it is designed to be flexible in general motions to move from one position to another with smooth movement to avoid sharp jolt in the robot arm. These jolts may damage the arm.

There are three main subsystems in robot manipulators: mechanical system, electrical system, and control system [2]. Mechanical system comprises of all movable parts. It consists of a group of links (rigid bodies) connected together by joints which allow the motion for the desired link. The mechanical system is used to move the end effector (the top link) to *xyz* position with respect to the base. This movement depends on the electrical system (e.g. motors, power amplifiers, and other electronic circuits) and it is done by some rotations and translations to the other links. Due to the varity of tasks and duties in robot manipulators, its construction is divided into two main classes: serial manipulator and parallel manipulators [2].

Serial manipulators consist of some links connected in series, which form an open loop chain. At the end of the chain, the end effector is connected to the base by single kinematic chain. On the other side, the parallel manipulators form a closed loop chain finished by the end effector and is connected to the base by two, or more kinematic chains (e.g. arm, or legs). The only drawback of the parallel manipulator over the serial manipulator is that the parallel robots manipulators suffer from limited workspace as compared with serial robot manipulators.

Figure 1.1 shows a schematic diagram of a robot manipulator. It consists of three joints and each one of these joints will have a motor to actuate the desired link. There are two widespread types of joints on this manipulator. The first type is represented by a cylinder, and it allows only relative rotation between two links. This type of joint is called ***revolute*** or ***rotary*** joint (e.g. human joints), and it is the most common joint type in robots.

[1] RIA: Robot Institute of America

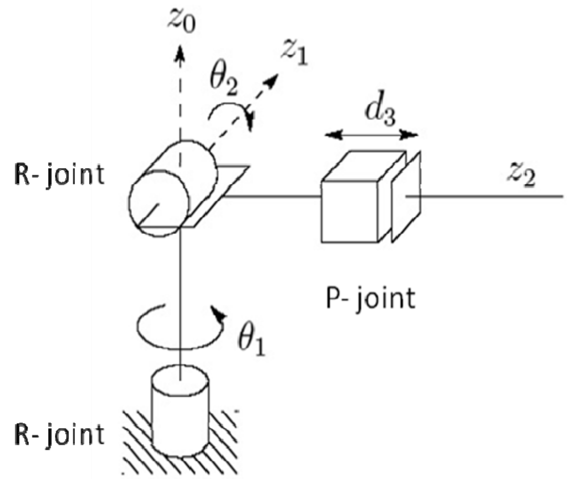


Figure 1.1: Manipulator with Revolute and Prismatic Joints [1]

The second type of joints is called ***prismatic*** or ***sliding*** joint. It is represented by square box. This type of joints allows only linear relative motion between two links along its axis. Both types are denoted as R and P joints. The robot manipulator whose all joint variables are prismatic is known as a cartesian manipulator, while the robot whose all joints are revolute is known as an articulated manipulator. The robot manipulator in Figure1.1 is called revolute revolute prismatic (RRP) manipulator. For more information and classification of robot manipulators readers may be referred to [1], [2] and [3].

The third system in robot manipulators is the control system. It is very important to control and adjust robot manipulator. Generally, two types of control systems are used: the open loop (OL) control system and closed loop (CL) control system. In OL control system, the controller sends a signal to the motor but does not measure the error action. On other hand, in CL control system, the controller sends the signal to the motor, and the output signal will be returned as feedback to describe the current state of the motor. CL controller has some advantages over the OL controller such as: disturbance rejection like friction in motors, improve reference-tracking performance and stabilization of an unstable process.

A control system consists of some devices and tools (e.g. sensors, controllers, and knowledge base) that provide convenient duty to robot manipulators. When the controller is moving, the robot manipulator during the working environment, the sensor or feedback system is gathering the information about the robot manipulator state and the surrounding circumstances, and then exploiting the information to modify and enhance the system behavior. Control system provides some function for the plant (robot arm) such as: a) providing the capability to move the robot manipulator in the surrounding environments. b) Collecting information about the robot manipulator in the working place. c) Using this information to give a methodology to control the robot manipulator. d) Storing the data then providing it to the robot manipulator then updating it at an instant.

One of the most vital and powerful issues in robotic fields is the control motion of the manipulator, because the robot operation must be accurate, without any effect in surrounding circumstances. Controlling manipulators is a major research area to limit the time history of joint inputs that required to move the end-effector to execute the required mission.

Generally, a controller is used to modify the behavior of the physical system according to the input value through computations and actuations. Over the early decade, numerous control techniques and methodologies have been proposed to control the motion of the robot manipulators such as point-to-point, sequencing “continuous path”, speed and incremental motions. As an example, the first control method capable for stopping at several different programmed positions, it can be used to pick and place operations. The required control method is chosen depending on the type of the robot manipulator and its possible applications. Varity of robot manipulators and their architectures influence the control methodology, for example, to control the robot manipulator movement between two points *x* and *y* (point to point) needs a different controller than the continuous path tracking. On the other hand, the mechanical design of the manipulator affects the controller type; for example, if there are two robots: one has RRR or (3R) joints as PUMA 560 and the other has PPP or (3P) joints as cartesian robot, then the control problems encountered with the RRR different from those encountered the PPP.

Although robot manipulators have variety of tasks in all applications, it has limited behavior as compared with human. Therefore, the control technique must apply to achieve the desired behavior. The basic configurations of the closed-loop control system depicted in Figure 1.2 include some main building components. These components will be discussed briefly.

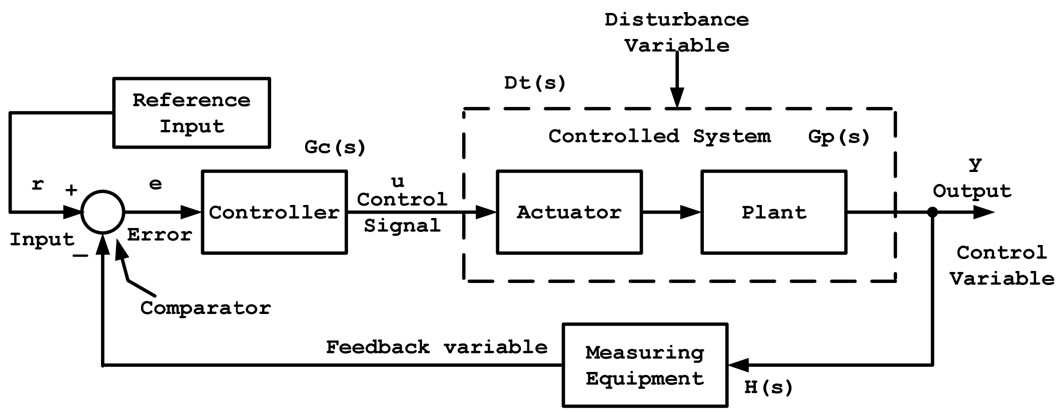


Figure 1.2: Block diagram of a closed-loop control system

The closed loop control system consists of the reference input or the set point of the closed loop, $r(t)$, summer, controller, the controlled plant, the output or the measured value $y(t)$, and the feedback loop. The plant, $G_p(s)$, is the physical system (e.g. a robot manipulator); it includes the actuators, gears, and mechanical design. The controller $G_c(s)$, is a device which is used to correct the error signal $e(t) = r(t) - y(t)$ and supply appropriate input to modify the physical system behavior, and enhance the characteristics of the closed loop system.

The controller attempts to reduce the error between the set point and the feedback signal to zero. However, if the input signal and feedback signal are not equal, the controller will correct the position signal until the difference between both signals is zero. Some disturbance variables $D_t(s)$, may influence the output signal. These disturbances are unpredictable inputs. The sensor $H(s)$, is the device that measures the output signal. Many systems may be unstable at all times due to nonlinearity, so a good control system must produce control output to track the desired response. This confirms that the important issue in designing a control system is to ensure that the dynamic response of the closed loop systems is stable.

1.2.1. Linear and Nonlinear Control

There are two methods used in control theory to control systems, linear method and nonlinear method. Using linear control is applicable only when the controlled system can be modeled mathematically [3]. The facts that the majority of physical systems have nonlinear characteristics; hence, linear controllers fail to meet the requirements due to system nonlinearities. The variations and the nonlinear parameters such as gear backlash, load variations and other parameters have unpredictable effects on the controlled systems (e.g. robot manipulator) diminish the performance. Therefore, the robot manipulator may be considered as a linear model when it works on small space, or it has a large gear ratio between the joints and their links. Nonlinear methods considered as general case when compared to linear methods because it can be applied successfully on the linear methods, but linear method is not sufficient to solve and control nonlinear problems. Common methodologies are used to solve the nonlinearities in control systems such as sliding mode control, and state feedback control are discussed in [4].

1.2.2. Independent Joint Control

Independent joint control (IJC) is considered as the simplest and traditional type to control the motion of the robot manipulator. The basis of IJC that is the robot manipulator is treated as a set of independent actuators works independently. This means that each link of the robot manipulator considered as single input single output (SISO) system it has independent controller. The IJC for 1 DOF is illustrated in Figure 1.3.

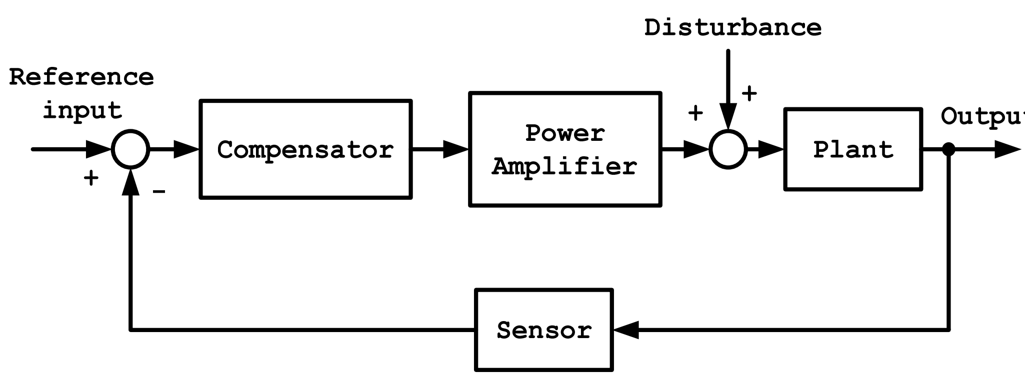


Figure 1.3: The basic structure of (SISO) system

Linear control techniques such as proportional integral derivative controller are suitable to control robot manipulators with a high gear ratio such as industrial robots, because the coupling effects between joints and links considered as disturbance that will be reduced. These effects may be ignored when the joint gear ratio is large (e.g. 200 to 1). Each joint of the robot consists of two subsystems: the first is the drivers (e.g. motors and gear's train), and the second is the link of the robot manipulator. In the case of the DC motor as a joint actuators each motor torque τ_t influence the motor shaft (its own link), but in the later term the nonlinear characteristics of the links (e.g. inertia term, external forces that may acts on the practical link) affects the performance of the robot manipulator.

1.2.3. Control Techniques

Due to uncertainty and instability effects, unknown or unpredictable inputs that manipulate the plant output to the incorrect target. These inputs are called disturbance or noise, so analyzing and designing the mathematical model of the system includes the controller and plants to get the desired behavior is required. Many control techniques have been proposed to control robot manipulator ranging in complexity from linear to the advanced control system, which compute the robot dynamic and save it from damage in real environments. Three different control schemes namely PID controller, FLC, and the fuzzy supervisory controller (FSC) will be implemented through this thesis. The performance of these controllers will be based on the high precision in reducing the overshoot, minimizing steady state error, damping unwanted vibration of robot manipulator, and handling the unpredictable disturbances.

PID controller is one of the earliest controllers in the industrial robot manipulators, so the first attempt to control the plant is use the PID controller. PID controller is still considered the most widely used in industry [5] and [6]. The popularity of using the PID or the PID-types controllers is that they have a simple structure, and they give satisfactory results when the requirements are reasonable and the process parameters variations are limited. In addition, the majority of applications are familiar with the PID controller based on the knowledge of the system characteristics. Several techniques used for tuning PID parameters that have been developed over the past decade such as Ziegler-Nichols (Z-N) tuning methods [7]. One of the drawbacks for using the PID control techniques is that, they are not sufficient to obtain the desired tracking control performance because of the nonlinearity of the robot manipulator. Hence, a lot of time is required to tune the PID parameters. On the other hand, other techniques are used to overcome the previous problem, such as fuzzy controller that emulates human operation.

FLC is an emerging technique in control systems. It is considered as intelligent controller. Many studies show that the fuzzy controller (FC) performs superior to conventional controller algorithms will be discussed in the next section. Zadeh [8] did the main idea of FLC and fuzzy set theory. Mamdani and his colleagues [9] have done a pioneering research work on FLC in the mid-’70 for engine steam boiler. The benefit of FLC is obvious when the controlled process is too complicated to be analyzed using PID controller or when the information about the controlled system does not exist.

FLC is classified into two categories: the first, involves the fuzzy logic system based on a rule based on expert system, to determine the control action. The second used FL to provide online adjustment for the parameters of the conventional controller such as the PID control [10]. This method attempts to combine the merits of FL with those control techniques to expand the capability of linear control technique to handle the nonlinearity in the physical system. Fuzzy supervisory is used to reduce the amount of tuning the PID controller with a fuzzy system [11]. It is considered as an attractive method to solve the nonlinear control problems, one of the advantages of fuzzy supervisory that the control parameters changed rapidly with respect to the variation of the system response. The fuzzy supervisor operates in a manner similar to that of the FLC and adds a higher level of control to the existing system. Fuzzy supervisory is hybrid between the PID controller and FLC that designed to overcome the problem of tuning PID in nonlinear systems using FLC as an adaptive controller [12]. The basic structure of FSC resembles the structure of PID controller, but the controlled parameter of PID controller depends on the output of the fuzzy controller.

1.3. Literature Review

Generally, robotic system is designed and developed to assist or replace a human in doing and accomplishing tasks that are boring, complex, too dangerous, and impossible for human. As an advanced investigation in this emerging field, a vast number of researches have been proposed over the last two decades for robot manipulator fields, some of these literatures discussed the kinematics analysis of industrial and educational robots such as PUMA 560, SCARA, and SG5-UT robot manipulators [11], [13] [14] and [15]. Other papers discussed control technique problems, such as PID, FLC and other techniques [16], [17], [18], [19] and [20]. A good review of some literatures is listed next:

Kinematics of robot arm was mathematically modeled using a Denavit Hartenberg (DH) method [1], [2] and [3]. Forward and Inverse equation analysis, were generated and implemented using a simulation program [21] and [22]. In [14] Annand derived the kinematics analysis of PUMA 560, and calculated the equation of motion of the robot by deriving the so-called Euler-Lagrange equations. In [23] achieving a high level of complexity for robotic system design was straightforward and highly intuitive when using the PTOLEMY II software environment. After deriving the inverse kinematics equations, Antonia used PTOLEMY II to design, and simulate robot arm. The benefit of this software is that designing complicated system requires simple building blocks.

Elgazzar presented in her paper [15] efficient solutions for the kinematics positions, velocities, and accelerations for the 6DOF PUMA 560 robots. The solution method was based on a method that fully exploits the special geometry of the robot in the derivation of the solution. Elgazzar showed that for the accurate control of the arm motions all these solutions were needed, resulting in a substantial saving in computation time, and a critical consideration for real time control.

Position control performed using independent joint control in [1]. This method was using PID controller, and it worked by controlling each joint independently. The coupling effect between the joints and links could be ignored if the gear ratio was large.

The work presented by Delibasi [24] illustrated the position control of DC motor using FLC and PID control algorithms. Both two controllers were designed based on LABVIEW program. After applying controllers, results showed that the desired position was achieved with 0.4% overshoot, and 80*msec* settling time for fuzzy controller, but when using PID controller with the overshoot is 4%, and the settling time is 120*msec*. Delibasi featured the influence of FLC upon the performance of robot movement simulation, which was controlled by a digital controller. Khoury [18] presented the design of a fuzzy logic controller of 5DOF robot arm. Through his paper, he introduced two structures of FLC: first three inputs with coupled rule base and the second structure was two inputs with coupled rules. Khoury confirmed the success of the proposed fuzzy controller. In addition, when a fuzzy controller in comparison to other nonlinear controls, it confirmed again the success in tracking control system.

In [25] the performance between proportional derivative fuzzy logic controllers (PDFLC) with 5 membership functions (MF) controller and PDFLC with 7 MFs controller was analyzed in terms of time response specifications and integral square error. Based on the simulation result Samin proved that PDFLC with 5 MFs was better than PDFLC with 7 MFs. For the time response performance, the PDFLC with 5 MFs produced settling time and rise time of 0.247*sec* and 0.156*sec* respectively whereas the

PDFLC with 7 MFs produced settling time and rise time of 0.298*sec* and 0.184*sec* respectively. It showed that the PDFLC with 35 rules and 7 MFs resulted in a slower response as in comparison to PDFLC with 11 rules and 5 MFs.

Wei_Li [26] presented an approach to combining a fuzzy logic controller with PID controller. In his paper, he replaced the proportional term in the conventional controller with fuzzy P controller for implementing fuzzy P+ID controller. The proposed controller (fuzzy P+ID) combined the advantages of the fuzzy controller and conventional controller. Comparison between the existing controller and PID controller Wei_Li shows that the fuzzy P term improved the overshoot and rising time while as the conventional terms; I and D controllers reduced the steady state error and enhanced the system stability. Wei_Li presented two features of his controller: first fuzzy P+ID controller kept the simple structure of PID controller, second the stability condition unchanged if the fuzzy P + ID controller replaced the PID controller.

In [19], the PDFLC fuzzy controller was combined with PID controller to enhance the performance and robustness of the controller. Simulation results showed that the combined structure did a very good controller. The work presented by Yang and Chen in [27] compared two fuzzy controllers: PDFLC and PIFLC. They proved by simulation that the PDFLC better than PIFLC. In PDFLC, response had a larger steady-state error than PIFLC response and the PIFLC response was less damped with an overshoot. Therefore, he combined two controllers to improve the step response.

In [16], two controllers were used for robot movement, with and without FLC. Soh and Alwi proved that the system with FC was found to be more efficient, where it increased the system stability. Mathematical model was developed for 3 DOF robot arm, FLC and neural network (NN) was developed to trace desired trajectory for 3 DOF robot arm. A fuzzy controller was applied to an inverted pendulum and was presented in [28]. Chopra and colleagues proved that increasing the rules beyond that limit was ineffective. Secondly, the rules can be reduced using the fuzzy subtractive clustering (FSC) approach, and it gave similar performance as by the larger rule set. The rules were reduced to 8 from 81, 49 and 25. Simulation and comparison of results had shown effectiveness of a proposed fuzzy controller using FSC approach. MATLAB simulation was presented in [17] proved that the fuzzy PID controller achieved better performance as in comparison to traditional controllers.

Khong [29] was applied FLC to control the position of DC servomotor. He explained that the result of the experiment with fuzzy controller reached the reference position and speed without any overshoot. The results of an experiment showed that the position control of DC servomotor was investigated with optimal performance and the proposed controller achieved and overcame the disadvantage of the conventional PID control sensitivity to inertia variation and sensitivity to variation of the position with the drive system of DC servomotor.

Sreentha and Pradhanb in [30] designed FLC for the position control of a revolute, single flexible link. The FLC was based on 49 IF-THEN rules and used the error in the angular displacement at the joint and its time rate of change as input variables. Both theoretical and experimental results showed that the angular displacement at the base joint was highly oscillatory and suffered from an excessively large settling time.

Surdhar and White [31] focused on the control of a nonlinear two-axis manipulator with a single flexible link. They used a PD controller with the gain of the derivative term being adjusted by a FLC whose measured inputs are the error and error change. The results showed that the fuzzy PD controller exhibited shorter settling time, smaller steady state error and handled some nonlinearity than the conventional PD controller. In [32] Nil and Yuzgec proved that the proposed FLC and NN control had reached the desired performance, and NN control traced the desired trajectory closer and smoother than the FLC.

The study, fuzzy supervisory have attracted attention in many papers through the history of FLC. Good presentation on this subject presented by Zhen-Yu [11] when he developed a fuzzy gain scheduling (FGS) of PID controller, the main idea presented how the parameters of the PID controller adapted on-line. The results showed that the variety of the process can be satisfactory controlled by the FGS and these results better as in comparison to the PID results. Due to the characteristic variations in the physical system, PID controller may not be sufficient. Therefore, [33] and [34] presented solutions for adapting PID parameters on-line. The results verified that the efficiency of the fuzzy supervisory control in improving the system response by making online modification to the original parameters.

In [12] and [20], the authors presented the fuzzy supervisory method for tuning PID parameters. This method was used to improve the performance given by Z-N parameters. The simulation results showed the superiority of the FLC, and it proved that it guaranteed very good performance in the set point and the load disturbance, and promising in an industrial environment. Limei presented a fuzzy self-adjusting PID controller [35]. This controller had advantages over PID control and fuzzy control. This controller was used to adjust the PID parameters according to the two inputs of the fuzzy controller. Limei showed that the fuzzy self-adjusting PID controller had shorter time and more robustness as in comparison to traditional PI controller.

In [36] Kyoung and Bao applied adaptive self tuning fuzzy PID control to real time position for Shape Memory Alloy. The results proved that the self-tuning fuzzy PID control achieved better performance as compared with PID controller without fuzzy tuning. Tzafestas [37] and Zhen [38] proposed approach for self-tuning PID control based on fuzzy logic. This approach assumed that through a classical tuning technique such as Z-N method, there were one available controller parameters. Then, using fuzzy logic as self-tuning for PID, these parameters were varied during system operation.

1.4. Problem Statement

Generally, any robot consists of motors and arms. DC motor modeling and robot arm kinematic analysis are rich research areas in literatures. The control task is to move the robot arm from an initial position to a final position. To achieve that we require prior knowledge of either desired position or angle of each joint, where using the angels is called forward kinematic while using the position is called inverse kinematic. This is done using many types of controllers. The controller is used to minimize the error between the desired and the actual positions. In doing so, the controller must meet certain specifications. These specifications such as reducing overshoot, minimizing rising time and eliminating steady state error. In addition reducing the load disturbances, which modeled on each motor.

1.5. Thesis Objectives and Methodologies

This work investigates modeling and control of the robot manipulator by analyzing the kinematics of the robot and applying control techniques. Thesis work is undertaken in the following developmental stages; first, we derive the forward and inverse kinematics equations of the robot. Then the complete mathematical model for a 5DOF robot arm including the dynamics of the motor actuators in both time and frequency domains is to be developed. The next stage we apply the PID controller with a feedforward compensator to reject the load of the motor, which is modeled as disturbance. The fuzzy logic controller is the second controller to be implemented. To form the third controller, the FSC, we combine both the PID controller and FLC in order to improve the tuning of the PID parameters.

The performance of the PID controller is to be compared with FLC in terms of time response, and then we compare the results of the PID classical tuning methods with the FSC. MATLAB will be the platform to simulate the 1 DOF and the 5DOF robot arm as a case study in this thesis.

1.6. Thesis Contribution

In this thesis, a mathematical model for the educational Lynx6 robot arm was used to implement three different controllers' techniques; a classical PID controller was tuned and used as a reference benchmark for the two other controllers, which are FLC and FSC. The feedforward compensator is added to the PID controller for disturbance rejection. Remodeling of the DC motor was required to achieve this goal. The FLC controller was used and a special rule base designed, the number, shape and range of the memberships were chosen to achieve the best performance. The third applied controller is the FSC. 49-rule base was designed for the PID parameters, Numbers and types of membership function for the FSC controller are chosen to give the desired performance. This comparative study can be used as a document of reference for other researches that are interested in this area of research.

1.7. Thesis Structure

This section outlines the overall structure of the thesis, and provides a brief description for each chapter.

Chapter 2 provides some basic knowledge about robot manipulators and presents two common problems in a robot manipulator: the first one is the kinematics analysis of the robot manipulator; the kinematics problem separated into two parts: the forward kinematics and the inverse kinematics. The second problem that will be discussed through this chapter is the modeling of the DC motor.

Chapter 3 presents one of the most commonly controllers used in control theory; it is the PID control. Review of the structure and concepts of the PID control and several tuning technique for PID parameters is presented. The characteristics of PID parameters and the effects of each parameter on the system response are illustrated. Following these sections, a good method for disturbance rejection namely feedforward method for disturbance rejection is presented. Finally independent joint control technique is presented for designing N DOF controllers for robot manipulator.

Chapter 4 presents the idea of the fuzzy logic control. Preliminaries of some basic concepts for fuzzy theory are discussed. First definition of the fuzzy set, the fuzzy subset and some operation of fuzzy logic are explained. Followed by the main block diagram of fuzzy controllers. Detailed description is discussed for each block of the

FLC such as fuzzifier, inference mechanism and defuzzifier. Finally, designing the fuzzy PID and its several types is presented.

Chapter 5 presents FSC for auto tuning PID control. FSC is designed to adjust the PID parameters on-line to acquire the best results. Chapter 6 shows the simulation results of the three different controllers. The three different controllers PID, FLC and FSC are implemented using SIMULINK and MATLAB. Comparison between the different controllers is presented. Chapter 7 summarizes the work presented in this thesis and indicates some recommendation and suggestion for future works.

CHAPTER 2 KINEMATICS AND MATHEMATICAL MODELING

2.1. Introduction

There are two main classes in a robot manipulator: serial manipulators designed using an open loop kinematic chain and parallel manipulator designed using closed loop kinematic chains.

This thesis handles serial manipulators. Robot manipulator consists of a collection of ***n-links*** that connected together by joints. Each one of these joints has a motor allowing the motion to the commanded link. The motors have feedback sensors to measure the output (e.g. position, velocity, and torque) at each instant. Links and joints form a kinematic chain connected to ground from one side, and the other is free. At the end of the open side, the end-effector (e.g. gripper, welding tool, or another tool) is used to do some tasks as welding, or handle materials [2]. Robot manipulator is named according to number of DOF, which refers to the number of joints. As an example, robot manipulator has 5 joints, which mean the robot has 5DOF, and so on.

In physical applications, it is important to describe the position of the end effector of the robot manipulator in one global coordinates. In transforming, the coordinates of the end effector from the local position to the global position, the robot movements are represented by a series of movements of rigid links. Each link defines a proper transformation matrix relating the position of the current link to the previous one.

As mentioned previously, robot manipulator whose all joints are prismatic is known as a cartesian manipulator while the robot whose joint variables are revolute is called an articulated manipulator. Figure 2.1 shows a cartesian manipulator with 3 rigid bodies and three joint variables represents the cartesian coordinates of the end effector with respect to the body 0, which is fixed.

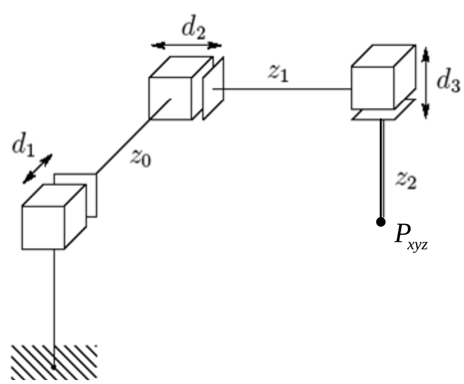


Figure 2.1: Robot manipulator with PPP joints [1]

Body 2 is fixed to body 1 and body 3 is fixed to body 2. The end effector P_{xyz}, body 3 and its movement relative to 2. The coordinate of the wrist point P_{xyz} with respect to the fixed body is:

$$\begin{bmatrix} x_p & y_p & z_p \end{bmatrix}' = \begin{bmatrix} d_1 & d_2 & d_3 \end{bmatrix}' \tag{2.1}$$

where d_1, d_2, and d_3 are the given range of motion

Kinematics is the motion geometry of the robot manipulator from the reference position to the desired position with no regard to forces or other factors that influence robot motion [3]. In other words, the kinematics deals with the movement of the robot manipulator with respect to fixed frame as a function of time. The fixed frame in robot represents the base and all other movements measured from the base as reference. It is one of the most fundamental disciplines in robots, providing tools for describing the structure and behavior of robot manipulator mechanisms, and it is important in practical applications such as trajectory planning and control purposes.

Generally, to control any robot manipulator the core of the controller is a description of kinematic analysis, this is done by using a common method in industrial and academic research, namely ***Denavit-Hartenberg*** method [1], [2] and [3]. The distinct of this method gives a mathematical description for all serial manipulators depending on the robot geometry, and it defines the position and orientation of the current link with respect to previous one. In addition, it allows the desired frame to create a set of steps to bring the other links coordinate into corresponding with another one. For more information, readers may return to the previous references. The kinematic solution in this chapter will focus on two important problem arises in robot manipulator. Section (2.2) discusses methodologies to solve the forward and inverse kinematic respectively.

The first problem is determining the forward kinematic (FK) where the robot manipulator end-effector will be if all joints are known. This means what rigid motion each joint effect on its link to obtain the desired configuration. The configuration space of the end-effector contains the transformation matrix T that relates the position and orientation of the end-effector. The following equation explains the forward kinematic problem.

$$F(\theta_1, \theta_2, \ldots, \theta_n) = [x, y, z, R_d] \tag{2.2}$$

where θ_1, θ_2 and θ_n are the input variables, $[x, y, z]$ are the desired position and R_d the desired rotation.

The second problem is determining the inverse kinematic (IK), which calculates the value of each joint variable if the desired position and orientation of end-effector are known. That means if the final link configuration is known, what is the possible configuration (e.g. solutions) of the robot manipulator to move the end effector of the robot arm to desired position and orientation in space. Inverse kinematic problem may express mathematically as follows:

$$F(x, y, z, R) = \left[\theta_1, \theta_2, \ldots, \theta_n\right] \tag{2.3}$$

For serial manipulators with revolute or prismatic joints the FK is derived using procedures such as the DH convention matrix [3], but in the parallel manipulator, the forward kinematic be not easy to be solved due to the complexity of the robot manipulator. Therefore, it may solved by using a set of nonlinear equations. On the other hand, solving the IK for parallel manipulator is easier than FK solution, and there are many solutions to achieve the desired task.

The second issue that will be discussed through Chapter 2 is the DC motor modeling. DC motor modeling is an important issue before designing a controller to know the system characteristics and its mathematical model.

To have a good model, it is important to understand the system behavior and to solve the associated problems. In general, an accurate model means that a designer would be able to predict the action of the system, diagnoses failure and simulates it in a precise manner or controls it to go to the desired position. Section (2.3) presents modeling and analysis of DC motor to derive the motor speed and position transfer functions.

2.2. Kinematic Analysis

This section will discuss the main two problems in the robot manipulator kinematic: the forward and inverse kinematic respectively and develop general steps to obtain the kinematic equations for the configuration of any serial robot arm to determine the position and orientation of the end effector relative to the base.

2.2.1. Forward Kinematic

The forward kinematic equations, describe the functional relationship between the joint variables and the position and orientation of the end-effector. Suppose the robot has *i*-links, the joints and links numbered from 1 to *i* and 0 to *i* respectively. The joint variables are denoted by q_i. In the case of prismatic joint, q_i represents the displacement, similarly q_i represent the angle of rotation for the revolute joint.

Figure 1.1 illustrates the kinematic diagram and the frame assignment of a robot manipulator with n-DOF. We will derive the forward kinematics for *i-links* robot manipulator according to the DH convention.

Consider a fixed frame $o_0x_0y_0z_0$ and the rotation frame $o_1x_1y_1z_1$. The orientation is represented as a series of three revolute about a combination of the principle axes of the link frame. The rotation of the rotated frame about the fixed frame represented by the three angles α, β and γ. The first rotation about z axis by angle α, and the next rotation about current y by angle β and the third rotation about the current z axis by the angle γ.

According to [1] the rotational transformation matrix that represents the position of the frame *i* with respect to frame 0 is expressed in equation (2.4). This equation represents the rotation matrix of the ZYZ Euler angles:

$$\begin{aligned} R_{ZYZ} &= R_{Z,\alpha}R_{Y,\beta}R_{Z,\gamma} \\ &= \begin{bmatrix} c_\alpha c_\beta c_\varphi - s_\alpha s_\gamma & -c_\alpha c_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta \\ s_\alpha c_\beta c_\gamma + c_\alpha c_\gamma & -s_\alpha c_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta \\ -s_\beta c_\gamma & s_\beta s_\gamma & c_\beta \end{bmatrix} \end{aligned} \tag{2.4}$$

To obtain the forward kinematic equations the following steps should be done:

a) Obtain the DH parameters.

To describe the kinematics of any robot, four parameters are given for each link $\theta_i, a_i, d_i, \alpha_i$ where two of them described the link, and the others describe the connection with other links. In the case of revolute and prismatic robots, the variable θ_i and d_i are denoted as joint variable. DH parameter is computed manually or using computer programs such as MATHEMATICA or MATLAB programs. Table (2.1) shows the DH parameters for *i*-link robot manipulator.

Table 2.1: DH parameter representation

Link	Joint	a_i	α_i	d_i	θ_i
1	0-1	a_1	α_1	d_1	θ_1
2	1-2	a_2	α_2	d_2	θ_2
3	--	--	--	--	--
4	--	--	--	--	--
i	$(i-1)\rightarrow i$	a_i	α_i	d_i	θ_i

b) Obtain link transformation matrices A_i (A matrices).

After obtaining the table of DH convention, a series of homogeneous matrices can be derived depending on the number of the DOF. The transformation matrix for each joint from joint 1 to the joint *i* can be calculated as:

$$A_i = Rot(z,\theta_i)Trans(z,d_i)Trans(x,a_i)Rot(x,\alpha_i) \tag{2.5}$$

or in terms of the full matrices

$$A_i = \begin{bmatrix} c_{\theta i} & -s_{\theta i} & 0 & 0 \\ s_{\theta i} & c_{\theta i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha i} & -s_{\alpha i} & 0 \\ 0 & s_{\alpha i} & c_{\alpha i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.6}$$

By multiplication, we obtain:

$$A_i = \begin{bmatrix} c_{\theta i} & -s_{\theta i}c_{\alpha i} & s_{\theta i}s_{\alpha i} & a_i c_{\theta i} \\ s_{\theta i} & c_{\theta i}c_{\alpha i} & -c_{\theta i}s_{\alpha i} & a_i s_{\theta i} \\ 0 & s_{\alpha i} & c_{\alpha i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.7}$$

where a_i is the distance along x_i from o_i to the intersection of x_i and z_{i-1} axes, d_i is the distance along z_{i-1} from o_{i-1} to the intersection of x_i and z_{i-1} axes, α_i is the angle between z_{i-1} and z_i measured about x_i , and θ_i is the angle between x_{i-1} and x_i measured about z_{i-1}. Equation (2.6) shows the symbolic of the i^{th} 4×4 homogenous transformation matrix. The homogeneous matrix houses the position and orientation information of a link frame with respect to adjacent link frame. If we employ equation (2.6) and Table (2.1), we can determine the *A* matrices for each link as presented in Appendix A.

c) Obtain the manipulator transformation matrix H_i^0 (H matrix).

After the homogeneous matrix has been defined for each link of the robot manipulator, simple solution to find the total homogeneous matrix for robot manipulator with *i-links* is accomplished by multiplying all the transformation matrices from A_1 to A_i as follows:

$$H_i^0 = A_1 A_2 .... A_i \tag{2.8}$$

The matrices from A_1 to A_i are the transformation matrices from joint 1 to joint *i* and H_i^0 is the location of the i^{th} coordinate frame with respect to the base coordinate.

d) Calculate the position and orientation of the end-effector.

The general homogeneous matrix for the desired position and orientation of the end-effector that obtained from Table (2.1) as follows:

$$H_i^0 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{32} & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_1 A_2 ....A_i \tag{2.9}$$

Equation (2.9) consists of two main components: the rotation matrix and the position vector of the end-effector as follows:

$$R_d = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{2.10}$$

$$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.11}$$

The orientation and the position of the end-effector solved directly once the homogeneous matrices for manipulator with *i*-links are multiplied.

The 3×3 rotation matrix provides the orientation of frame i with respect to the base frame. The position vector $d = (x, y, z)^T$ represents the desired position from the origin o_0 to the origin o_1 expressed in the frame $o_0 x_0 y_0 z_0$. In the previous equations $c_1 = (r_{11}, r_{21}, r_{31})^T$ is a vector represents the direction of x_i in the $o_0 x_0 y_0 z_0$ system, $c_2 = (r_{12}, r_{22}, r_{32})^T$ is a vector represents the direction of y_i , and $c_3 = (r_{13}, r_{23}, r_{33})^T$ represents the direction of z_i .

Solutions of the Euler angles are given as:

$$\alpha = A\tan 2(r_{33}, \sqrt{1 - r_{33}^2}) \tag{2.12}$$

or

$$\alpha = A\tan 2(r_{33}, -\sqrt{1 - r_{33}^2}) \tag{2.13}$$

If $s_\alpha > 0$ then:

$$\beta = A\tan 2(r_{13}, r_{23}) \tag{2.14}$$

$$\gamma = A\tan 2(-r_{13}, r_{32}) \tag{2.15}$$

If the value of $s_\alpha < 0$ is chosen, then

$$\beta = A\tan 2(-r_{13}, -r_{23}) \tag{2.16}$$

$$\gamma = A\tan 2(r_{31}, -r_{32}) \tag{2.17}$$

2.2.2. Inverse Kinematic

This section is concerned with the IK problem to find the joint variables of the robot manipulator for a given position and orientation of the end effector [1]. The problem of the inverse kinematics (IK) is more difficult than the forward kinematics problem. It can be mathematically expressed as:

$$\theta_k = f_k(x, y, z, \alpha, \gamma, \phi) \tag{2.18}$$

where $k = 1, \ldots, i$, θ_k joint angles and $(x, y, z, \alpha, \gamma, \phi)$ represents the position and orientation.

There are steps used to solve the inverse kinematics for robot manipulator as follows:

1. Equate the general transformation matrix to the final transformation matrix of the robot manipulator.

$$H_i^0 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_1 A_2 \ldots. A_i \tag{2.19}$$

2. For the both matrices define:
   a) The elements that contain one joint variable.
   b) Pairs of elements, which contain only one joint variable.
   c) Elements, or combinations of elements, contain more than one joint variable.
3. After defining these elements, equate it to the corresponding elements in the other matrix to form equations, and then solve these equations to find the values of joint variables.
4. Repeat step (3) to identify all elements in the two matrices.
5. In the case of inaccuracy, solutions look for another one.
6. If there is more joint variable to be found, multiply equation (2.19) by the inverse of A matrix for the specified links.
7. Repeat steps (2) through (6) until solution to all joint variables have been found.
8. If there is no solution to the joint variable in terms of an element transformation matrix, it means that the arm cannot achieve the specified position and orientation; the position is outside the robot manipulator workspace.

Calculation of the inverse of A_i matrices and solution of the inverse kinematics for joint variables of the experimental robot are derived in Appendix A.

2.3. DC Motor Modeling

Generally, modeling refers to system (e.g. plant) description in mathematical terms, which characterizes the input-output relationship [39]. Direct current (DC) motor is a common actuator found in many mechanical systems and industrial applications such as industrial and educational robots [3]. DC motor converts the electrical energy to mechanical energy. The motor directly has a rotary motion, and when combined with mechanical part it can provide translation motion for the desired link.

Equation (2.20) states the relation between the current and developed torque in the motor shaft.

$$\tau_m(t) = K_m \phi i_a(t) \tag{2.20}$$

where $\tau_m(t)$, is the motor torque produced by the motor shaft, ϕ, the magnetic flux, $i_a(t)$, the armature current, and K_m, is a proportional constant.

Equation (2.21) illustrates the relation between the produced EMF and the shaft velocity:

$$v_b(t) = K_m \phi \omega_m \tag{2.21}$$

where $v_b(t)$, denotes the back EMF, and ω_m, is the shaft velocity of the motor.

DC motors are important in control systems, so it is necessary to establish and analyze the mathematical model of the DC motors [39]. Figure 2.2 shows the schematic of the armature controlled DC motor with a fixed field circuit.

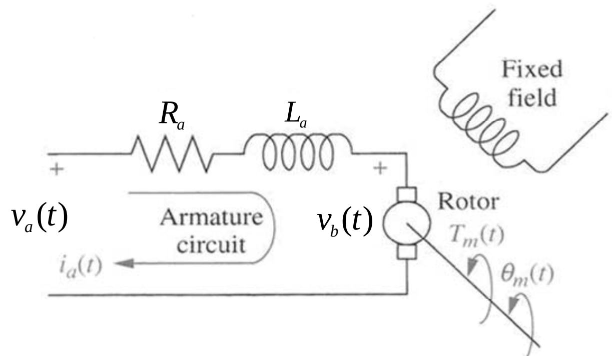


Figure 2.2: Schematic of DC motor system

It is modeled as circuit with resistance and inductance connected in series. The input voltage $v_a(t)$, is the voltage supplied by amplifier to move the motor. The back EMF voltage $v_b(t)$, is induced by the rotation of the armature windings in the fixed magnetic field. To derive the transfer function of the DC motor, the system is divided into three major components of equation: electrical equation, mechanical equation, and electro-mechanical equation [28]. Equation (2.22) and (2.23) are derived in Appendix A. The transfer function of the motor speed is:

$$G_{speed}(s) = \frac{\dot{\theta}(s)}{V(s)} = \frac{K_t}{J_m L_a s^2 + (L_a B_m + R_a J_a)s + K_t K_b} \tag{2.22}$$

In addition, the transfer function of the motor position is determined by multiplying the transfer function of the motor speed by the term $\frac{1}{s}$:

$$G_{position}(s) = \frac{\theta(s)}{V(s)} = \frac{K_t}{s[J_m L_a s^2 + (L_a B_m + J_m R_a)s + K_t K_b]} \tag{2.23}$$

where, J_m, and B_m, are denoted as the moment of inertia and motor friction coefficient. Table (2.2) shows the DC motor.

According to the previous discussion, the schematic diagram in Figure 2.2 is modeled as a block diagram in Figure 2.3. This block diagram represents an open loop system, and the motor has built-in feedback EMF, which tends to reduce the current flow.

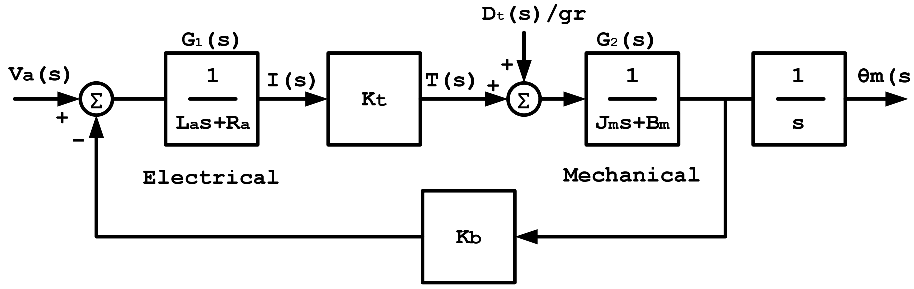


Figure 2.3: Block diagram for DC motor system

The advantage of using the block diagram gives a clear picture of the transfer function relation between each block of the system. Therefore, based on the block diagram in Figure 2.3, the transfer function from $V_a(s)$ to $\theta_m(s)$ with $D_t(\mathrm{s}) = 0$ was illustrated in equation (2.23).

Transfer function from the load torque, $D_t(\mathrm{s})$ to $\theta_m(s)$ is given with $V_a(s) = 0$:

$$\frac{\theta_m(s)}{D_t(\mathrm{s})} = \frac{(L_a s + R_a)/gr}{s\left[(J_m s + B_m L_a)(L_a s + R_a) + K_t K_b\right]} \tag{2.24}$$

where, *gr*, is the gear ratio. Deriving equation (2.24) is obtained in Appendix A.

Using SIMULINK, the model of the motor may be created. This model includes all the parameters derived previously. Figure 2.4 shows the SIMULINK model of DC motor.

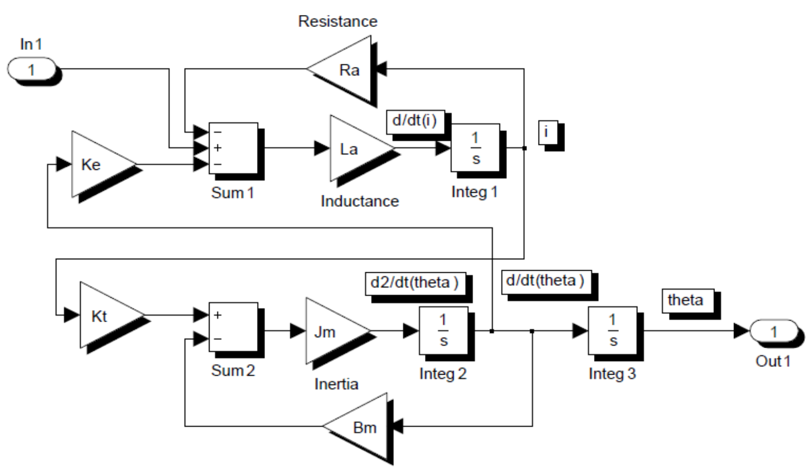


Figure 2.4: DC motor subsystem using SIMULINK

To obtain the state-space representation of DC motor in the space matrix, state space model takes the form:

$$\dot{X}(t) = Ax(t) + Bu(t)$$

and

$$Y(t) = Cx(t) + Du(t) \tag{2.25}$$

To solve DC motor transfer function using state space: first, assign the variables. Let $x_1(t) = i_a(t)$, $x_2(t) = \theta(t)$ and $x_3(t) = \omega(t)$. Second, take the first derivative of the previous system equations as $\dot{x}_1(t) = di_a(t)/dt$, $\dot{x}_2(t) = d\theta(t)/dt$ and $\dot{x}_3 = d\omega(t)/dt$.

The state-space representation of DC motor in space matrix could be expressed in this form:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} \frac{-R_a}{L_a} & 0 & \frac{-K_b}{L_a} \\ 0 & 0 & 1 \\ \frac{K_t}{J_m} & 0 & \frac{-B_m}{J_m} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} 1/L_a \\ 0 \\ 0 \end{bmatrix} v_a(t) \tag{2.26}$$

The output equation is:

$$y(t) = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} \tag{2.27}$$

where A is the system dynamic matrix, U is the input matrix, Y is the output matrix, B , C and D are coefficient matrices.

Table (2.2) shows DC motor parameters and values chosen for motor simulation.

Table 2.2: DC motor parameter and values

Parameter	Value
Moment of inertia	J_m = 0.000052 Kg.m^2
Friction coefficient	B_m = 0.01 N.ms
Back EMF constant	K_b = 0.235 V/ms^{-1}
Torque constant	K_t = 0.235 Nm/A
Electric resistance	R_a = 2 ohm
Electric inductance	L_a = 0.23 H
Gear ratio	gr
Load torque	$\tau_l(t)$
Angular speed	ω_m rad/sec

To study the behavior of the DC motor, consider the system without disturbance, then; substitute the parameter values of DC motor from Table (2.2) into equation (2.23). The open loop transfer function of the motor is:

$$G(s) = \frac{19649}{s^3 + 201s^2 + 6290s} \tag{2.28}$$

Using the MATLAB function ZPK (G(s)), equation (2.28) can be written in the zero/pole/gain form as:

$$G(s) = \frac{19649}{s(s+162.2)(s+38.7)} \tag{2.29}$$

The open loop poles are $s_1 = 0$, $s_2 = -162.2$ and $s_3 = -38.7$. The open loop step response of equation (2.29) is shown in Figure 2.5. As shown, the system does not go to steady state value but to an increasing value. This means the armature rotates at a constant speed, which is achieved by its built-in velocity feedback factor.

Simulation results using SIMULINK are shown in Figure 2.5 and Figure 2.6 for DC motor model with and without load disturbance.

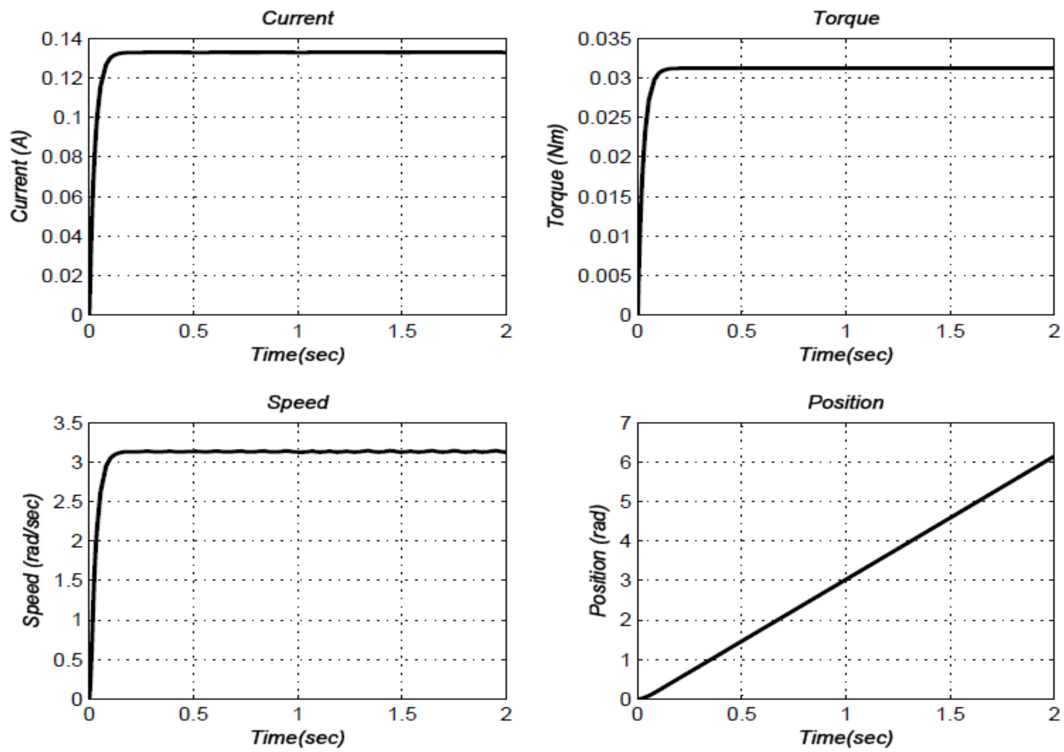


Figure 2.5: DC motor open loop step response without load

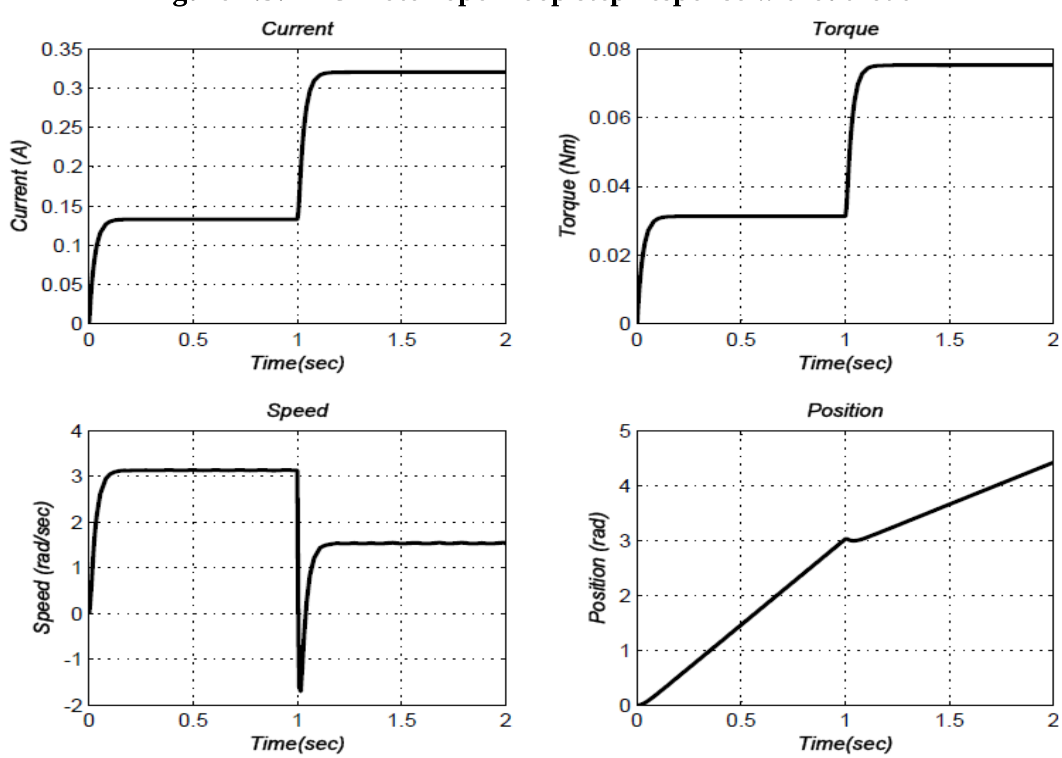


Figure 2.6: DC motor model simulation with load disturbance

Simulation results demonstrated that, the motor running at no-load conditions at startup, and still running to reach the steady state value as shown in Figure 2.5. When a mechanical load is applied suddenly to the shaft as shown in Figure 2.6, a small no-load current did not produce enough torque to carry the load; thus, the motor starts to slow down. This cause counters EMF to diminish resulting in a higher current and a corresponding higher torque. When the torque developed by the motor is exactly equal to the torque imposed by the mechanical load, then the speed will remain constant.

However, we want the motor to move the robot arm to a proper angular position corresponding to the input. This can be achieved by using control technique as PID controller, which will be discussed in the next chapter.

2.4. Summary

The primary purpose of Chapter 2 was to develop a systematic procedure for obtaining the forward and inverse kinematic equations for any robot manipulator combined both revolute and prismatic joints during section (2.2). Section (2.3) presented important issues in the robotic system; it is the modeling of the DC motors. DC motor modeling is important to find the transfer function of any motor.

CHAPTER 3 PID CONTROLLER DESIGN

3.1. Introduction

PID controller is considered the most control technique that is widely used in control applications. A huge number of applications and control engineers had used the PID controller in daily life. On the other hand, many research papers, number of master and doctoral theses and books have been written on this subject. PID control offers an easy method of controlling a process by varying its parameters. PID works well in industrial applications such as slow industrial manipulators were large components of joint inertia added by actuators. Since the invention of PID control in 1910, and Ziegler-Nichols' (ZN) tuning method in 1942 [7] and [38], PID controllers became dominant and popular issues in control theory due to simplicity of implementation, simplicity of design, and the ability to be used in a wide range of applications [40]. Moreover, they are available at low cost. Finally, it provides robust and reliable performance for most systems if the parameters are tuned properly. According to different sources like JEMIMA[2], PID controllers or PID variations (P, PD, and PI) are widely used in more than 90% to 95% of control applications. However, the PID controller has its own limitation; the PID performances can give only satisfactory performance if the requirement is reasonable and the process parameters variation are limited.

Setting the PID parameters is called ***tuning***. During the last six decades, several methods have been proposed for determining the PID controller gains. Some of those methods use the information concerning the open loop characteristics such as Cohen-Coon method. Other methods use the Nyquist curve plotting of the plant such as Ziegler-Nichols tuning method [41]. All of these tuning formulas need to know a prior knowledge about the system. PID control technique is used to control and enhance the system characteristics such as reducing the overshoot, speed up rising time, and eliminating the steady state error. Each one of the PID parameters has specific criteria to enhance the characteristics of the controlled system. These specifications will be discussed through this chapter, and the result will be presented and included in Chapter 6.

3.2. PID Structure

PID algorithm, assumes the controlled plant is known, such as a robot manipulator [17]. As shown in Figure 1.2, the error signal, $e(t)$, is the difference between the set point, $r(t)$, and the process output, $y(t)$. If the error between the output and the input values is large, then large input signal is applied to the physical system. If the error is small, a small input signal is used. As its name suggested, any change in the control signal, $u(t)$ is directly proportional to change in the error signal for a given proportional gain K_P. Mathematically the output of the proportional controller is given as follows:

$$u(t) = K_P e(t) \tag{3.1}$$

where $e(t)$, the error signal and $K_P = u(t)/e(t)$ indicates the change of the output signal to the change of the error signal.

[2] Japan Electric Measuring Instrument Manufacture's Association

Proportional term is not sufficient to be a controller in practical cases to meet a specified requirement (e.g. small overshoot, good transient response) because the large proportional gain gives fast rising time with large overshoot and oscillatory response. Therefore, a derivative term is added to form PD controller, which tends to adjust the response as the process approaches the set point. The output of PD controller is calculated based on the sum of both current error and change of error with respect to time. The effects of PD give a slower response with less overshoot than a proportional controller only. Mathematically, PD controller is represented as:

$$u(t) = K_P e(t) + K_D \frac{de(t)}{dt} \tag{3.2}$$

The main function of the third term; integral control tends to reduce the effect of steady state error that may be caused by the proportional gain, where a smaller integration time result is the faster change in the controlled signal output. The general form of the PID controller in continuous time formula given as:

$$u(t) = K_P e(t) + K_I \int_0^t e(t)dt + K_D \frac{de(t)}{dt} \tag{3.3}$$

Each term of the three components of PID controller, is amplified by an individual gain, the sum of the three terms is applied as an input to the plant to adjust the process. It will be noted that the purely derivative or integral plus derivative variations never used. In all cases except proportional control, the PID compensator gives at least one pole and one zero.

In real application, PID algorithm can be implemented in different forms depending on the process and control requirements. The easiest form introduced is the ***parallel form***, as shown in Figure 3.1, where the P, I and D elements has the same input signal $e(t)$.

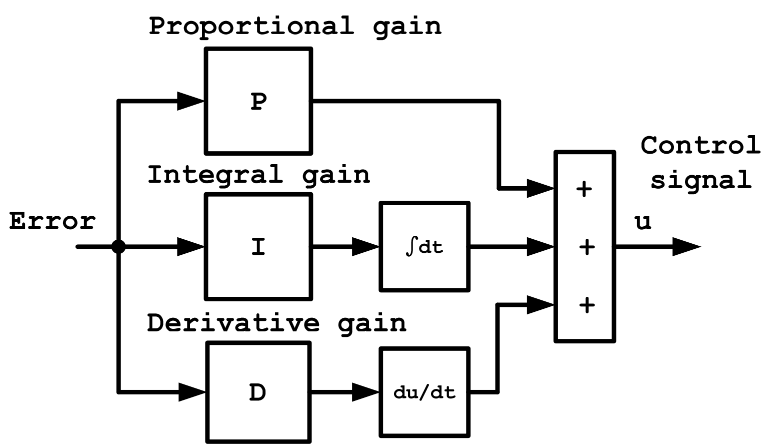


Figure 3.1: PID controller structure

The terms K_P, K_I and K_D stand for the proportional, integral, and derivative gains. The terms $e(t)$ and $u(t)$ represent the error and the control signal respectively.

The transfer function of the PID controller in parallel is:

$$G_{PID\_Paralle}(s) = K_P + \frac{K_I}{s} + K_D s = \frac{K_D s^2 + K_P s + K_I}{s} \tag{3.4}$$

Equation (3.4) may be rearranged to give the ideal form as follows:

$$G_{PID}(s) = K_P(1 + \frac{1}{T_I s} + T_D s) \tag{3.5}$$

where $T_I = K_P / K_I$, and $T_D = K_D / K_P$ are the integral and derivative time constant respectively.

Another type of the PID controller is known as the ***serial form***, and it has the mathematical form:

$$G_{PID\_series}(s) = (K_p + \frac{K_I}{s})(K_D s + 1) \tag{3.6}$$

The construction of both forms indicates that the setting of the PID controller depends on the used algorithm.

Figure 3.2 shows the effect of the three parameters on a unit step input. The response of the three parameters is modeled using SIMULINK.

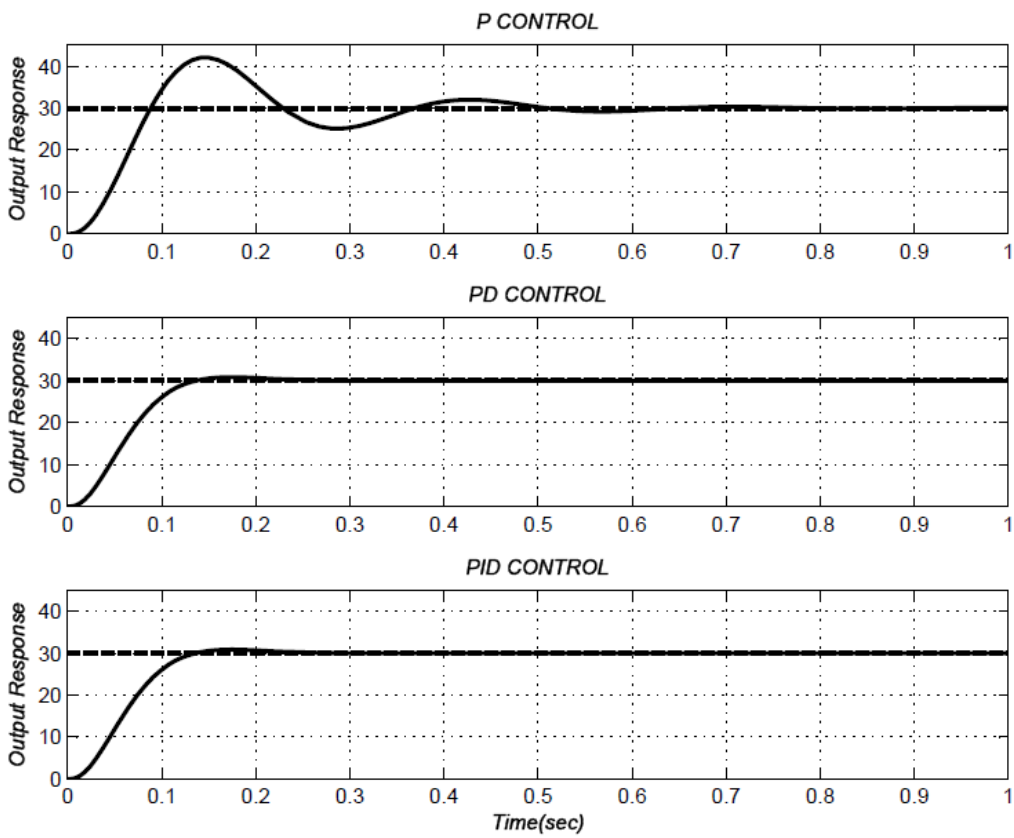


Figure 3.2: Step response of P, PD, and PID controllers

3.2.1. Controller Design Methods

The position control system is widely used in controlling applications such as robot manipulators. After analyzing the DC motor model as discussed in Chapter 2, the subsequent step is to design the controller to achieve the requirements. Control system has two parts: first, the controlled system and the second is the compensator as will be seen later.

According to textbooks and dissertations that concerned in PID controller problem [5] and [42] there are two classes for designing PID parameters: time domain methods and frequency domain methods. Despite the time domain methods are widely used, many of these methods are based on trial and error design strategy. In these methods, the PID parameters are determined using a simulation program or some experiment. Time domain methods are more popular than frequency domain method for tuning PID parameters. However, the frequency domain methods are more convenient and suitable if the exact mathematical model of the physical system is known [6]. One of the advantages of the frequency domain methods is to guarantee the system stability, especially if the system characteristic is unpredictable.

If the PID parameters are chosen incorrectly, the input control process can be unstable, (e.g. output diverges, with or without oscillation). Therefore, adjustment of these parameters is a good solution to acquire the optimum values for the desired control response. As mentioned, the PID tuning is an important issue, and it is concerned with the best selection of the three parameters, so an acceptable performance of the control loop is established. Numerous methods for tuning PID parameters were presented in control textbooks and many papers listed in [38]. Some of the popular PID tuning methods for time and frequency domain methods are discussed below.

Iterative or Manual Tuning Method:

Iterative or manual tuning is considered as an experimental method and it is used to determine the PID controller parameters. An experimental procedure using tuning can be outlined as follows:

1. Integral and derivative gains equal zero.
2. Proportional gain is tuned to give the desired response, neglect the steady state error.
3. Increase K_P gain by small increment and adjust the derivative gain K_D to decrease the damping.
4. Adjust the integral gain K_I to remove the steady state error.
5. Replicate the previous steps until acquiring the desired response.

This method concerned as a time consuming method because it depends on trial and error approach.

Ziegler–Nichols Frequency Domain Method:

This method is based on the closed loop system response. Initially K_I and K_D gains are set to zero. The proportional gain is increased until the process oscillation occurs. It reaches the critical gain value K_{CR} at which the output of the loop starts to oscillate. Using the value of a critical or ultimate gain K_{CR} and the oscillation or ultimate period P_{CR} , the value of PID parameter K_P , K_I and K_D are given in terms of the ultimate gain and ultimate period:

$$
\begin{aligned}
&K_P = 0.6K_{CR} \\
&K_I = \frac{2K_P}{P_{CR}} \qquad \Rightarrow T_I = 0.5P_{CR} \\
&K_D = \frac{K_P P_{CR}}{8} \qquad \Rightarrow T_D = P_{CR}/8
\end{aligned}
\tag{3.7}
$$

Ziegler–Nichols Time Domain Method:

Ziegler and Nicholas introduces this method. It is based on the characteristic of the open loop step response of the system. Two parameters are determined: the maximum point of the slop of the step response, and the intersection between the tangent and the *x-axis*. Four steps to determine a , L , and PID parameters are described as follows:

1. Design the control system at the open loop state.
2. Plot the step response as shown in Figure 3.3.
3. Draw tangent line crossing the middle point of the slop of the step response.
4. Determine the PID parameters according to the following relations:

$$\begin{aligned} K_P &= 1.2/a \\ T_I &= 2L \\ T_D &= 0.5L \end{aligned} \tag{3.8}$$

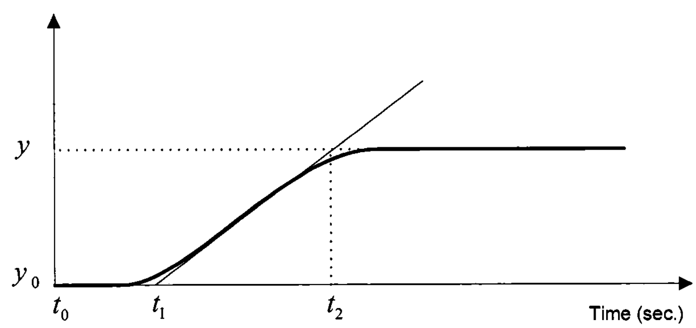


Figure 3.3: Plant step response in Z-N method

where time constant $a = t_1 - t_0$ and the dead time L is defined as $t_2 - t_1$.

Cohen-Coon Method:

The procedure to find the PID parameters in this method is the same as Ziegler-Nichols time domain method. The PID parameters are calculated according to the following formulas:

$$\begin{aligned} K_P &= (0.25 + 1.35L/a)/K \\ T_I &= L\,(2.5 + 0.46L/a)/(1 + 0.61L/a) \\ T_D &= 0.37L/(1 + 0.19L/a) \end{aligned} \tag{3.9}$$

L and a are defined as the same in the Z-N, based on the dead time and the time constant respectively. K is a process gain.

Root Locus Method:

Root locus method is a good technique to design the PID parameters. It’s a graphical technique that gives a description of the control system as various parameters change, such as overshoot and rising time. This method is used to analyze the relationship between the poles, gains, and stability of the system [43].

Root locus means in control theory, the location of the poles and zeros of transfer function. Pole location determines system stability. If the roots of transfer function in the right half plan of the continuous system or inside the circle of discrete systems, it indicates that the system is unstable, where if these roots in the left half plan this means the system is stable. In addition, when root location on jw axis, the system is considered marginal stable.

When designing a PID controller using root locus method, the system must be transformed into the function formula. Root locus method one of the methods is used for designing PID parameters through this thesis. SIMULINK diagram of the overall system with PID controller is shown in Figure 3.4.

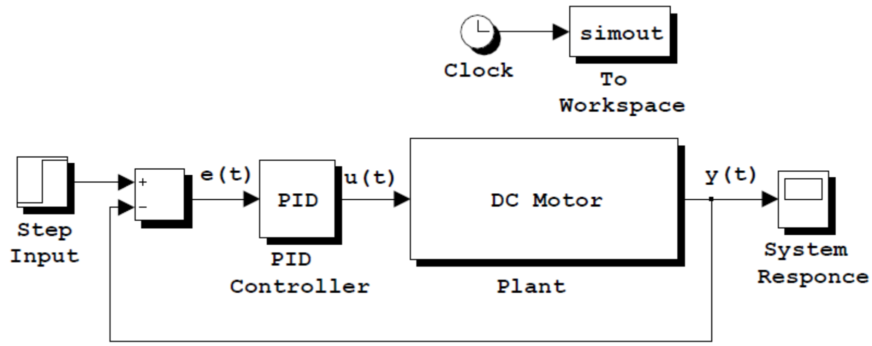


Figure 3.4: SIMULINK model of PID controller

The analytical solution of the root locus method is discussed in Appendix B.

3.2.2. PID Characteristic Parameters

Proportional action K_P improves the system rising time, and reduces the steady state error. This means the larger proportional gain, the larger control signal become to correct the error. However, the higher value of K_P produces large overshoot and the system may be oscillating; therefore, integral action K_I is used to eliminate the steady-state error. Despite the integral control, reducing the steady state error, it may make the transient response worse [38]. Therefore, derivative gain K_D will have the effect of increasing the damping in system, reducing the overshoot, and improving the transient response.

As discussed previously, each one of the three gains of the classical PID control has an effect of the response of the closed loop system. Table (3.1) summarizes the effects of each of PID control parameters. It will be known that any changing of one of the three gains will affect the characteristic of the system response.

Table 3.1: PID characteristic parameters

Closed-Loop Response	Rise Time	Overshoot	Settling Time	Steady State Error
Increasing Kp	Fast	Increase	Small / No effect	Decrease
Increasing Ki	Fast	Increase	Increase	Decrease
Increasing Kd	Small / No effect	Decrease	Decrease	Small / No effect

These characteristics are shown graphically in Figure 3.5 for a unit step response. In addition, it will be defined as:

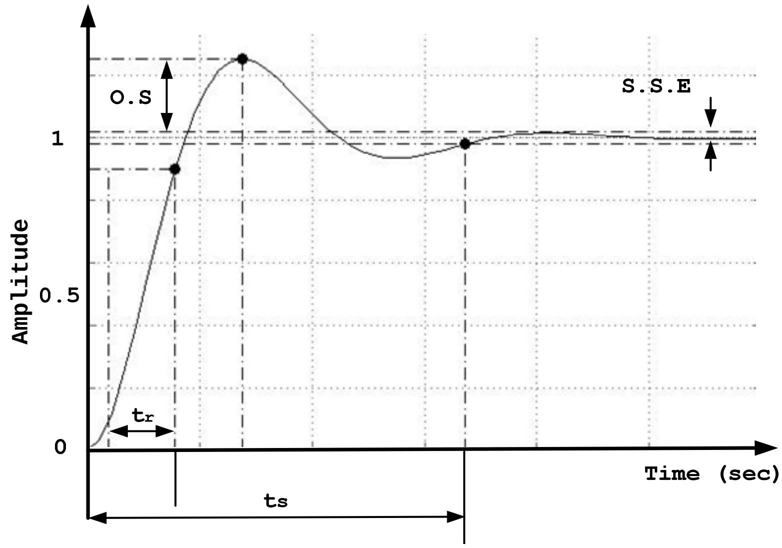


Figure 3.5: Unit step response curve showing $t_r, t_s, O.S$ **and** SSE

1. *Maximum overshoot* (*OS*): it is the maximum peak value of response curve measured from unity.

$$\%OS = e^{-(\zeta\pi/\sqrt{1-\zeta^2})} \times 100 \tag{3.10}$$

2. *Rising time* (t_r): the rise time is the time required for the response to rise from 10% to 90% of its final value.

3. *Settling time* (t_s): the settling time is the time required for the response curve to reach and stay with a range about the final value ($\pm 2\%$).

$$T_s = 4/\zeta\omega_n \tag{3.11}$$

4. *Steady state error* (SSE): it expresses the final difference between the process variable and the set point.

The design parameters in equation (3.10), and (3.11), must be chosen by system designer to acquire the system stability.

To illustrate the performance of the PID controller, substitute the DC motor parameters values from Table (2.2) in equation (2.23) and (2.24) in Chapter 2, and the results are used in MATLAB to compute the response.

The transfer function from reference input, $R(s)$ to output, $Y(s)$ is:

$$\frac{Y(s)}{R(s)} = \frac{0.235}{1.196*10^{-005}s^3 + 0.002404s^2 + 0.07523s} \tag{3.12}$$

The transfer function from disturbance input, $D_t(s)$ to output, $Y(s)$ is:

$$\frac{Y(s)}{D_t(s)} = \frac{(0.000052s + 2)}{1.196*10^{-005}s^3 + 0.002404s^2 + 0.07523s} \tag{3.13}$$

Figure 3.6 illustrates the effect of the proportional, PI, and PID controllers on the response of the system to reference step input. It shows the oscillatory with proportional controller and lower steady state error for PI and PID controllers.

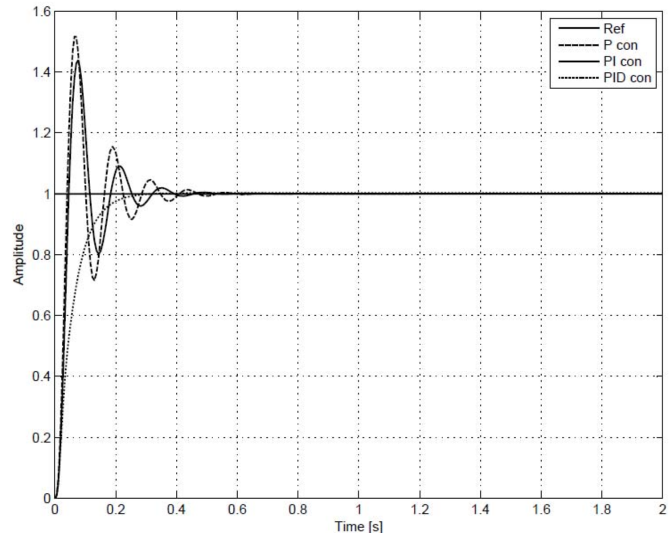


Figure 3.6: Transient response to step reference input

It is cleared when using the proportional term, the system has rising time equal 0.07 *sec*, overshot equal 1.55, and the output response is oscillatory. Improving these variables are accomplished by adding the integral term to reduce the oscillatory. In this case, the oscillatory is starting to fall and the overshoot value is less than the case of the proportional term. Finally, reducing the large overshoot is accomplished using the derivative term. The rising time is equal 0.23 *sec*, but the overshoot is reduced completely.

Effects of the proportional, PI and PID controller on the response of the step disturbance response of the system is shown in Figure 3.7. Adding the integral term increases the oscillatory behavior but lowers in the error, and adding the derivative term reduces the oscillation while maintaining a lowest error.

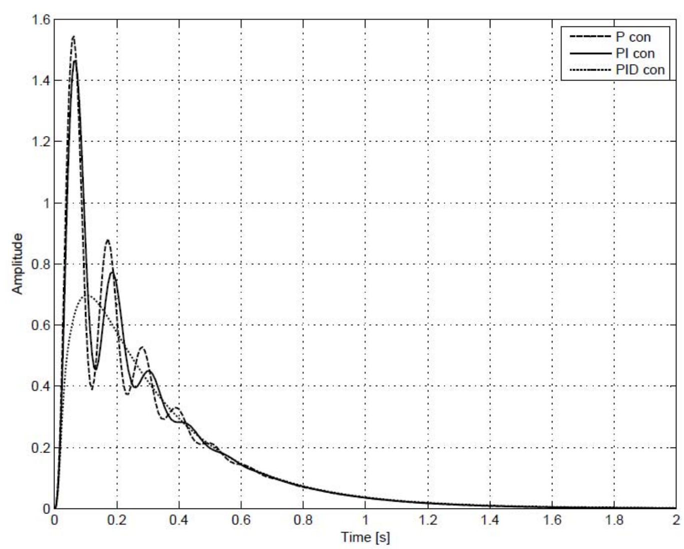


Figure 3.7: Transient response to step disturbance input

If the load torque is applied as disturbance, the output response of the system will change. Therefore, many methods are used to reject disturbance like feedforward.

3.3. Feedforward Disturbance Rejection

In many control systems, attention of disturbance is a necessary concern. Any control system may have unpredictable inputs that affect the plant output. Disturbance is denoted as an external inputs added to the control system, which drive system away from its desired task [44]. In many cases, it is possible to measure disturbances before they influence the processes. Therefore, many control techniques are used to reject the disturbance before they create control error. One of these techniques is the feedforward method.

As mentioned in Chapter 2, DC motor block diagram was shown in Figure 2.3, where $G_1(s)$, is the electrical term, which contains the electrical resistor R_a and the electrical inductance L_a, and, $G_2(s)$, is the mechanical term containing the elements J_m and D_m. If the load torque $\tau_l(t)$ is applied on the motor, it will affect the system response. The load torque on the motor is considered as disturbance input that will be rejected.

The total system output using superposition method as follows:

$$\theta(s) = \frac{K_t G_1(s) G_2(s)}{1 + G_1(s)G_2(s)K_t K_b} V_a(s) + \frac{G_2(s)}{1 + G_1(s)G_2(s)K_t K_b} D_t(s)$$
$$= G_p(s)V_a(s) + G_d(s)D_t(s) \tag{3.14}$$

where, $G_p(s)$, is the transfer function from the input, $V_a(s)$ to the output, and $G_d(s)$ is the transfer function from disturbance input, $D_t(s)$ to the output.

Control systems should eliminate the disturbance input to allow the output respond to the desired input; The block diagram of DC motor with disturbance input was shown in Figure 2.3 is rearranged and drawn in Figure 3.8.

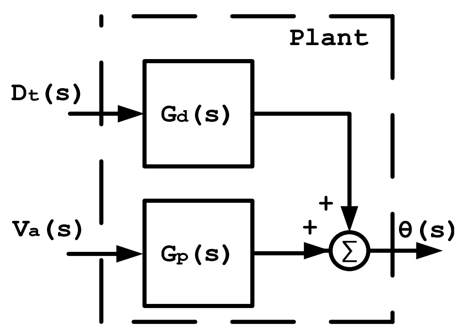


Figure 3.8: Servomotor model

When adding the controller $G_c(s)$ to the plant model as shown in Figure 3.8, we will obtain the closed-loop system as shown in Figure 3.9. The total transfer function of the new system given as:

$$\theta(s) = \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)H(s)} V_{in}(s) + \frac{G_d(s)}{1 + G_c(s)G_p(s)H(s)} D_t(s)$$
$$= T(s)V_{in}(s) + T_d(s)D_t(s) \tag{3.15}$$

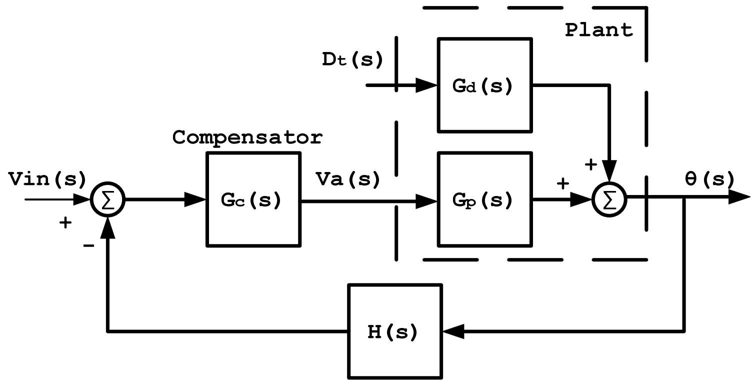


Figure 3.9: Closed loop system with disturbance input

where, $T(s)$, is the transfer function from the reference input $V_{in}(s)$ to the output, and $T_d(s)$ is the transfer function from the disturbance input $D_t(s)$ to the output.

Since the disturbance term was derived in the right side of equation (3.15), we design a controller to reject the disturbance input. This means that the, $T_d(s)$ values must reach zero. In this case, one of the following solutions may be used to reject disturbance.

The first attempt to reject the disturbance is to decrease the gain of the transfer function, $G_d(s)$, as possible between the input and the output, but this attempt is not sufficient to reject the disturbance because of the motor parameters are constant. Second method tends to eliminate, $T_d(s)$, by increasing the loop gain, $G_c(s)G_p(s)H(s)$ without increasing, $G_p(s)$, because increasing the plant gain may increase the disturbance gain. The third method reduces the magnitude of the disturbance input. This method is a good method, but it is not sufficient in practical cases due to electrical and thermal noises produced during the motor rotation. Last method is the feedforward method that is illustrated in Figure 3.10.

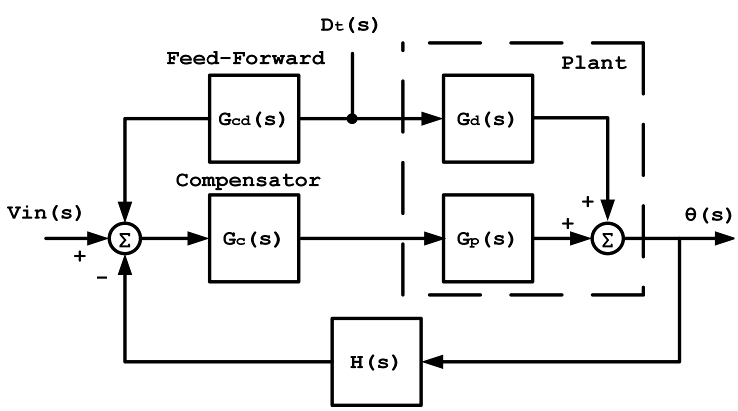


Figure 3.10: Feedforward compensation

This method may be applied when the sensors directly measure the change in the disturbance variable. The feedforward combined with the reference input, output signal and the total error signal is:

$$e(t) = r(t) - y(t) - d(t) \tag{3.16}$$

In this case, the feedforward controller G_{cd}, rejects the disturbance that affect the system without affecting the transfer function $T(s)$, from the input to the output. The transfer function from the disturbance input $D_t(s)$, to the output is now

$$T_d(s) = \frac{G_d(s) - G_{cd}(s)G_c(s)G_p(s)}{1 + G_c(s)G_p(s)H(s)} \tag{3.17}$$

Suppose the disturbance $T_d(s)$, equal zero, then it's good to reject the load torque disturbance . In this case, the numerator will equal zero.

$$G_d(s) - G_{cd}(s)G_c(s)G_p(s)$$

$$G_{cd}(s)G_c = \frac{G_d(s)}{G_p(s)} \tag{3.18}$$

Finally, the efficiency of rejection the disturbance depends on the high precision of the $G_p(s)$ parameters and the type of the $G_{cd}(s)$ controller.

If the PID controller is used as controller plant, the final transfer function of the disturbance compensator, $G_{cd}(s)$, is given by:

$$G_{cd}(s) = \frac{L_a s^2 + R_a s}{K_t(K_D s^2 + K_P s + K_I)} \tag{3.19}$$

Using SIMULINK, the DC motor model with disturbance rejection control using PID controller is created as shown in Figure 3.11. This model includes all transfer functions that are derived previously.

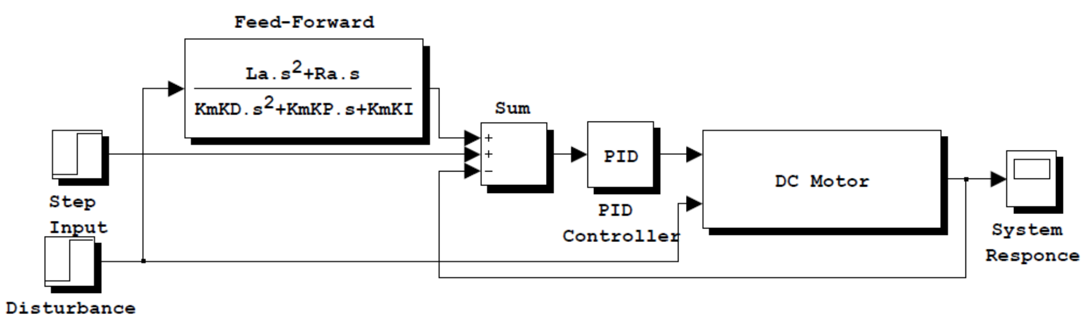


Figure 3.11: PID controller with feedforward compensator

Figure 3.10 and Figure 3.11 represent two degree of freedom controllers. The DOF for a control system is defined as the number of the closed loop transfer function that can be adjusted independently. The controller consists of two compensators: PID controller, $G_c(s)$ is used to control the plant and the feedforward controller, $G_{cd}(s)$ is used to reject the load torque disturbance.

3.4. Implementation of N Independent Joint Control

This section focuses on the design of N independent joint control of the robot manipulator. These controllers will be modeled and tested by using SIMULINK and MATLAB environment. IJC controls the position of each joint independently [1]. This will help to ignore the dynamic coupling between joints.

The simulated model was shown in Figure 3.11 represents only one motor to move one link. IJC for N joint robot manipulator (e.g. revolute or prismatic) joints is accomplished by using N block diagrams with respect to the number of the independent joints. Designing these controllers accomplished using SIMULINK. The main system consists of several levels. As a case study, the independent joint control is implemented to 5DOF robot arm. Figure 3.12 shows the first level of SIMULINK block diagram with three subsystems.

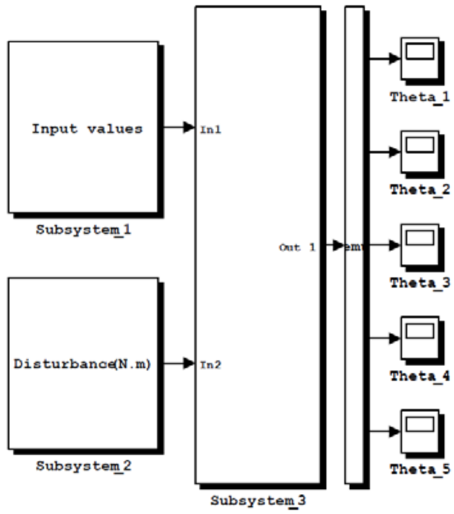


Figure 3.12: IJC for 5DOF using SIMULINK

The main block diagram consist of three subsystems namely as subsystem 1, subsystem 2, and subsystem 3. The first has the input values for each joint variable of the robot manipulator; these values represent the angles required that can be controlled by each one of the controllers independently. When looking under the mask in subsystem 1, Figure 3.13 shows the step input of the five angles from Theta_1 to Theta_5.

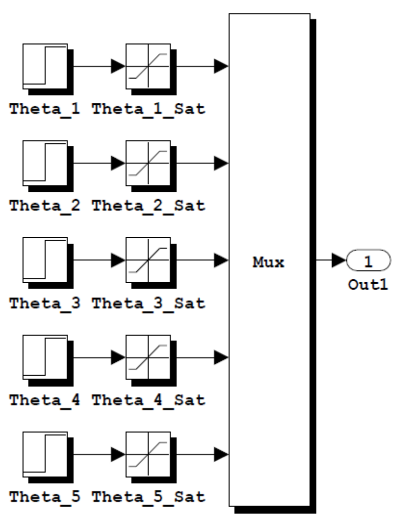


Figure 3.13: Input angles with saturation boundaries

Double-click on subsystem 1. Subsystem 1 provides the input menu as shown in Figure 3.14. This menu allows the user to gives angle for each joint. The saturation block was shown in the figure above is used to limit the values of the input angles for a definite range.

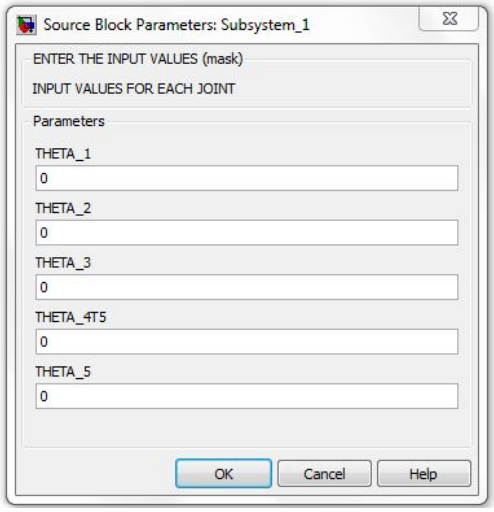


Figure 3.14: Input menu

Similarly, subsystem 2 has the disturbance input values that affect each link, the input menu and the disturbance inputs with saturation boundary under the mask. When looking under the mask in subsystem 3, five blocks is used in subsystem 3. Each one of these blocks has independent controller as seen in Figure 3.15.

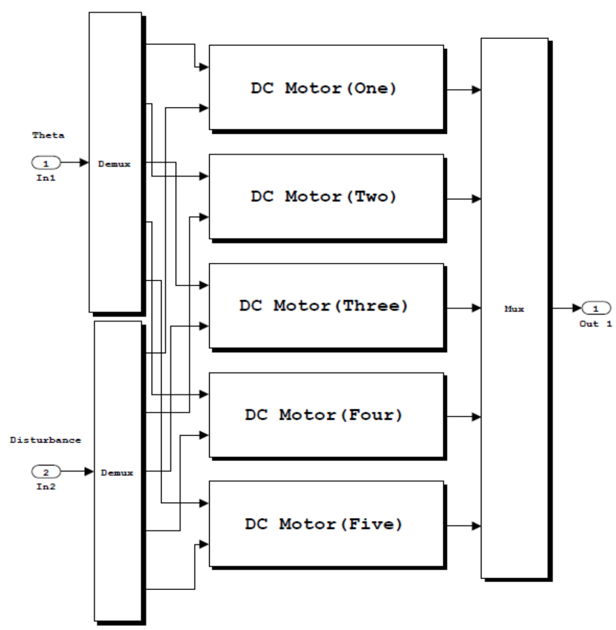


Figure 3.15: Subsystem 3 under the mask

For each IJC, we fill the appropriate angle in the menu. When the signal enters the closed-loop control from input one of Figure 3.16, it passes through many blocks such as gain, sum, and PID control. The gain block is used for multiplication process between the input and a constant value to increase or decrease the signal value. Next, the summation block is used to passes the error signal which is the difference between the desired input and the output. The error enters the PID controller block. Finally, the output of the PID controller is used to move the motor of the desired link. When the disturbance applied from the second input, it passes through the feedforward block. The switch block is used to switch between the zero as constant and the feedforward block to show the effect of disturbance with and without feedforward control.

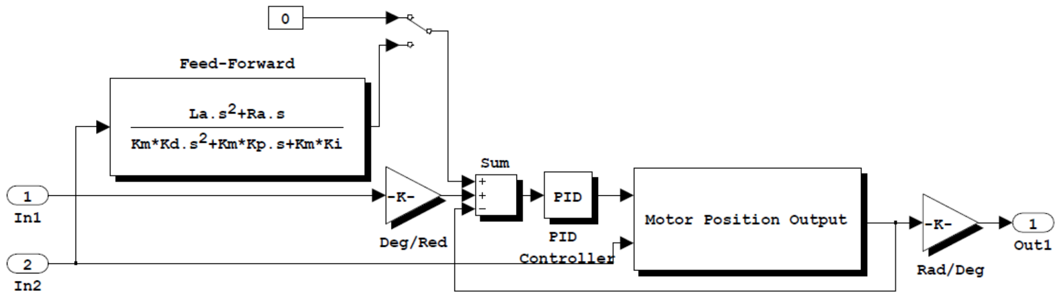


Figure 3.16: One joint variable

3.5. Summary

The purpose of Chapter 3 was to explain the basis of one of the benchmark controllers in control theory. PID control structure, PID controller characteristics, and the PID designing method were introduced in section (3.2). Feedforward method and it is efficacy for disturbance rejection was introduced in section (3.3). Finally, section (3.4) presented the independent joint control method for designing the controllers of 5DOF.

In spite of the diversity in using PID controller, a lengthy time is needed for the trial and error tuning to find suitable PID parameters. Moreover, it is hard to achieve the desired control performance when applying PID controllers to nonlinear and time-varying systems because of the dynamic nature of these systems. Therefore, control techniques designed to endeavors limiting the drawback in the PID controller. One of these techniques is the fuzzy logic controller that will be studied in the next chapter.

CHAPTER 4 FUZZY LOGIC CONTOLLER

4.1. Introduction

Fuzzy logic (FL) is based on fuzzy set theory that was established by Lofti A. Zadeh in 1965, when he presented his milestone paper on fuzzy sets [8], and introduced the concept of linguistic variable in 1973. Zadeh showed that fuzzy logic unlike classical logic could realize values between false and true. In classical set theory, definition of membership function does not matter, but the number belongs to or does not belong to the set, yes or no, and the 0 or 1 takes on the value. This approach is not suitable in many life applications such as the set of age or the set of temperature, but the element has movable values between 0 and 1. This means that the elements of such sets not only represent true or false values but also represent the degree of truth or the degree of falseness for each input. As an example the set of temperature, the temperature of 50 Celsius has the membership function value of 0.7.

Control engineering is one of the major fields where the fuzzy theory has been successfully applied. Many researches and applications have been performed since Mamdani and his colleague [9] presented the first FLC work. Their work mimics the human operator for a steam engine and boiler combination using a set of linguistic variable in the form of IF-THEN rules such as: **IF** (System state) **THEN** (Control action) which referred to “Mamdani controller”. The term of IF-THEN, is obtained experimentally depends on the control engineer, or human expert that produces the appropriate output, depends on the control rules chosen.

Motivated by the success of Mamdani works in applying fuzzy control, the FLC has been one of the most active and fruitful applications of fuzzy set theory during the last two decades [25]. It has been successfully implemented and employed in a wide variety of industrial and commercial applications [45], [46] and [47] including: consumer products such as washing machines, video cameras, and industrial engineering area as controlling cement kilns, automatic container crane, and robot manipulators. In recent years, FLC has appeared as a promising solution when the nonlinear systems are complicated for analysis using classical control. Thus, the FLC may be viewed as a step toward a relationship between control systems and human-like decision-making.

FLC is used in a widespread system nowadays [48]. It is an automatic control, and a self-acting mechanism that controls an object in accordance with a desired behavior. FLC is based on the response, knowledge, and human experience in controlling systems. It can be used for modeling the behavior of linear/nonlinear or static/dynamic systems. Among many control methodologies, fuzzy controller has a greet consequence and a better transparency than other control techniques because of its relative simplicity and agreeable results. The basic idea of FLC is that used to convert the linguistic variable based on the information of the operator into control actions applied to the system under control. A classical controller such as PID controller is efficient and offers powerful method to analysis linear systems. In case of nonlinear systems classical controllers does not produces satisfactory results due to the nonlinearities of these systems [34]. Therefore, FLC may be an efficient tool to control these nonlinear systems [46].

In closed loop control systems, the classical controllers have been replaced by the FLC. This means that the IF-THEN rules and fuzzy membership functions replaces the mathematical models to control the system. Both controllers are designed to enhance the system stability and to meet the requirement of the system behavior. However, the main advantage of fuzzy logic when compared with classical controllers resides in the fact that, the fuzzy controller deals with the all system as a black box [46] because the mathematical model of the system may be too complex to be described; thus, it is difficult to be controlled with classical controllers. The control rules are based especially on knowledge of the system behavior and the experience of the control engineer.

Control engineer may be having idea about the characteristics system and good knowledge about controlling it. In designing a fuzzy control system, we typically express linguistic variable as a process of inputs and outputs in terms of linguistic values such as hot, warm, and cold. As an example, some people may classify 25 degrees Celsius as warm other may be classifying it as hot; hence, this relation between the temperature and its values is fuzzy.

4.2. Preliminaries on FLC

Through this section, as an introduction to fuzzy set theory, some definitions of fuzzy and fuzzy sets are briefly defined and discussed. ***The main idea in fuzzy set theory is that an element has a degree of membership to a fuzzy set.*** For more details, reader may be return to [45], [49] and [50].

Assume U is a collection of all objects, members or elements of U denoted as u, and U is referred to the ***universe of discourse***. The element u represents any element in U.

4.2.1. Fuzzy Sets, Fuzzy Subset, and Null Fuzzy Set

For some objects u in U, a fuzzy set A in a universe of discourse U is defined as a set of ordered pairs u and μ_A, in which each element u of U take a value $\mu_A(u)$ into the interval $[0,1]$, and it is defined as $\{A = (u, \mu_A(u)),\ u \in U\}$, where $\mu_A(u) \in [0,1]$ is the MF for the fuzzy set A. The output of the membership function for a given input u is denoted as the membership degree or degree of the membership.

It is clear that, there is an evident difference between fuzzy set and crisp set in defining the membership function $\mu_A(u)$, as shown in the next two equations for the temperature between 15 and 40 degrees.

In crisp set the membership function, is define as:

$$\mu_A(u) = \begin{cases} 1\ if\ \ temp \in [15,40] \\ 0\ if\ \ temp \notin [15,40] \end{cases} \tag{4.1}$$

On the other hand, the membership function in fuzzy set defined as:

$$\mu_A(u) = fn(temp) \tag{4.2}$$

In classical set theory, a set A is defined to be subset of set B if and only if all element of set A are contained in set B. This definition has extended to fuzzy set theory as follows:

For two fuzzy sets A and B given in the universe of discourse U with two membership functions $\mu_A(u)$ and $\mu_B(u)$ respectively. A is defined as a fuzzy subset in B, defined as $A \subset B$ if:

$$\mu_A(u) \leq \mu_B(u) \tag{4.3}$$

In classical set theory, null set is the set that contains no elements. On the other hand, the null fuzzy set is defined in fuzzy set theory as:

For a given fuzzy set $A \subset U$ with membership function denoted as $\mu_A(u)$, the fuzzy set A is a null fuzzy set Φ if:

$$\mu_A(u) = 0 \qquad \forall\ u \in U \tag{4.4}$$

This means if A is a fuzzy set then no element in U has member in A.

4.2.2. Membership Function Features

A relation gives grades of membership for each element of a fuzzy number; this means that there is no sharp boundary between membership and non-membership. It takes values between 0 and 1. Figure 4.1(a) illustrates the membership function of fuzzy set cold, warm, and hot.

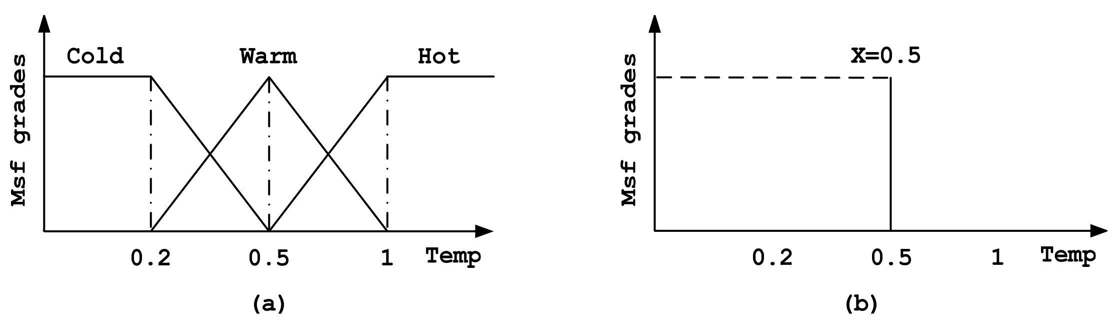


Figure 4.1: (a) Triangular MFs (b) Singleton MFs

The universe of discourse U in the above figure is defined as temperature. There are three fuzzy sets cold, warm and hot with corresponding membership functions $\mu_{\text{cold}}(u)$, $\mu_{\text{warm}}(u)$ and $\mu_{\text{hot}}(u)$. Some features of the MFs are discussed below such as core, boundary, support, singleton, symmetry, normality, center, and height.

The ***core*** of a fuzzy set A is the crisp set of all points u in the universe of discourse U where $\mu_A(u) = 1$. The ***boundary*** of fuzzy set A is the crisp set of all points u in U where $0 < \mu_A(u) < 1$. The ***support*** of fuzzy set A is the crisp set of all points u in U where $\mu_A(u) > 0$. If the fuzzy set, whose support is single point in U it is referred to fuzzy ***singleton***. Figure 4.1(b) shows single point $u = 0.5$ in U. A fuzzy set A is said to be a ***symmetric*** if $\mu_A(u)$ is symmetric around a certain point $x = u$. Figure 4.1(a) shows the fuzzy set warm is symmetric around point $u = 0.5$.

The membership function is said to be ***normal*** if one element at least or more in the universe of discourse U has a value 1 as example $\mu_{warm}(u = 0.5) = 1$. The ***center*** of a fuzzy set A is the mean value of all points u in U that achieves the maximum value of $\mu_A(u)$. The point $u = 0.5$ is the center of fuzzy set warm in Figure 4.1(a). The ***height*** of fuzzy set A is the largest value of $\mu_{warm}(u)$ warm in Figure 4.1(a) is equal 1.

4.2.3. Linguistic Variable

A linguistic variable is a variable, written in a natural language format, which represents imprecise information. For example, if we are studying the case of the room temperature. Using FL, temperature is linguistic variable that takes the fuzzy sets: cold, warm, and hot as shown in Figure 4.2 for the universe of course U defined on [0, 50]. This means that the range of the temperature from 0 to 50 Celsius. Temperature below $10\ \text{C}^{o}$ is interpreted as cold and temperature above $50\ \text{C}^{o}$ is considered as hot.

Now, temperature between 10 and $25\ \text{C}^{o}$ is cold with certain membership value and warm with another membership value. Similarly, the temperature between 25 and $40\ \text{C}^{o}$ may considered as hot with specified membership value and warm with another membership value.

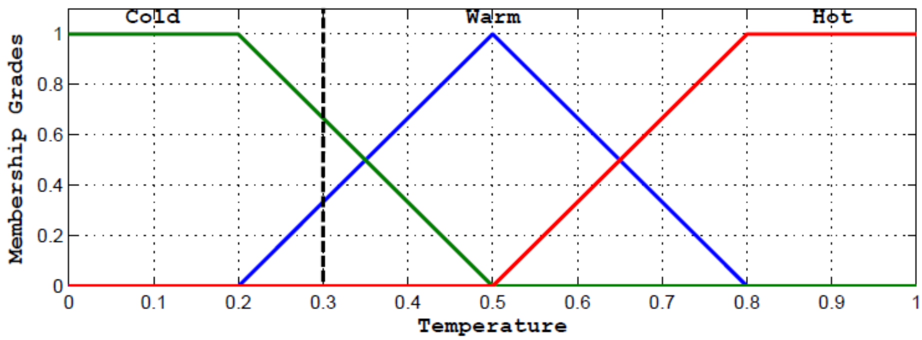


Figure 4.2: Example of fuzzy linguistic variable

Each element of Figure 4.2 is defined as a mathematical function, which normalized to the output range [0, 1] by multiplying scaling factor 1/50 from the discourse U defined on [0, 50]. In addition, this figure illustrates the idea of fuzzy partitioning. In fuzzy partitioning the transition or the moving from one set to another is easy. This means that the degree of a membership function for input u in a set warm increase while its membership function in a set cold decreases as the value of u moves from set cold to set warm.

Many papers used the notation of linguistic variable in the form

$$\{X, T(X), U, S_X\} \tag{4.5}$$

where X denoted as the linguistic variable name such as temperature, error, and error change, $T(X)$ is the set of the names of the linguistic variable. In the case of temperature we have, $T(temp)$ is $\{cold,\ warm,\ hot\}$. S_X , gives the meaning of the linguistic variable/label such as S_X take the label hot, this meaning returns to the linguistic variable temperature. Finally, U is the universe of discourse of the variables where X takes a crisp value.

4.2.4. Linguistic Value

Let U_i^m denotes as a linguistic value of the linguistic variable u defined on the universe of discourse U , then the linguistic variable u is takes on elements from the set of linguistic values are denoted $U_i = \{U_i^m\}$, where m number of the linguistic values. Linguistic values are generally expressed by terms such as Negative Big (NB), Negative Small (NS), Positive Big (PB), and Positive Small (PS). For example, assume u denotes the linguistic variable temperature, then the linguistic values of the temperature variables are $U_i^1 = cold$, $U_i^2 = warm$, and $U_i^3 = hot$, so that $u \in U_i^m$ where $U_1^m = \{U_1^1, U_1^2, U_1^3\}$.

4.2.5. Fuzzy Conditional Statement

For two inputs e and Δe and output u with a universe of discourses $E, \Delta E$ and U respectively. The fuzzy conditional statement consists of two parts: antecedent that represents the condition in the application domain and the consequent represents the control action for the controlled system. The IF-THEN fuzzy control rule has the form:

$$\begin{array}{l} R_1 : \textbf{IF } e \text{ is } A_1 \textbf{ AND } \Delta e \text{ is } B_1 \textbf{ THEN } u \text{ is } C_1 \\ R_2 : \textbf{IF } e \text{ is } A_2 \textbf{ AND } \Delta e \text{ is } B_2 \textbf{ THEN } u \text{ is } C_2 \\ \qquad\quad \cdots \quad \cdots \quad \cdots \quad \cdots \\ R_j : \textbf{IF } e \text{ is } A_j \textbf{ AND } \Delta e \text{ is } B_j \textbf{ THEN } u \text{ is } C_j \end{array}.$$

where e, Δe and u are linguistic variable representing two process state and one control variable. A, B and C are linguistic values of the linguistic variables e, Δe and u in the universe of discourse $E, \Delta E$ and U respectively and $j = 1, 2, \cdots, n$.

4.2.6. Operations on Fuzzy Sets

The following operations are suggested by Zadeh to establish the concept of the fuzzy set theory. Assuming A and B are two fuzzy sets in U with MFs μ_A and μ_B respectively. Fuzzy mathematics denoted as relation between the elements of A and B described using $\mu_{A\times B}(u_1, u_2)$, where $u_1, u_2 \in A, B$. The commonly used set operations are union, intersection, and complement. The union of A and B means that all elements belong to A or B. The intersection of A and B is mean that, the collection of all elements belong to A and B. The complement of A is the elements, which do not belong to A. Figure 4.3, Figure 4.4, and Figure 4.5 shows the basic operation of the fuzzy set respectively.

1. Union

The union of two fuzzy sets A and B on the universe of discourse U is a fuzzy set C is described in the relation $C = A \cup B$, and the membership function of the fuzzy set C is described as following:

$$\mu_C(u) = \mu_{A\cup B}(u) = \max\{\mu_A(u), \mu_B(u)\} \tag{4.6}$$

Moreover, it shown as:

$$\mu_C(u) = \mu_A(u) \vee \mu_B(u) \tag{4.7}$$

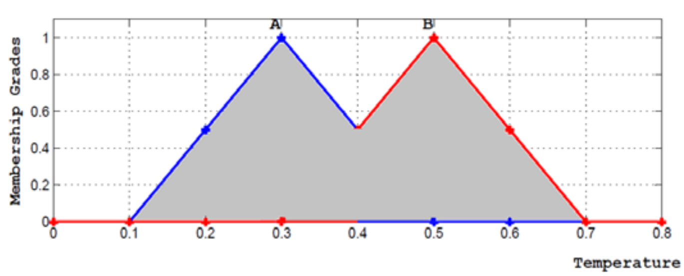


Figure 4.3: Union of fuzzy set *A* and *B*

2. Intersection

The intersection of two fuzzy set A and B on the universe of discourse U is a fuzzy set C is described in the relation $C = A \cap B$, and the membership function of the fuzzy set C is described as following:

$$\mu_C(u) = \mu_{A\cap B}(u) = \min\{\mu_A(u), \mu_B(u)\} \tag{4.8}$$

Moreover, it shown as:

$$\mu_C(u) = \mu_A(u) \wedge \mu_B(u) \tag{4.9}$$

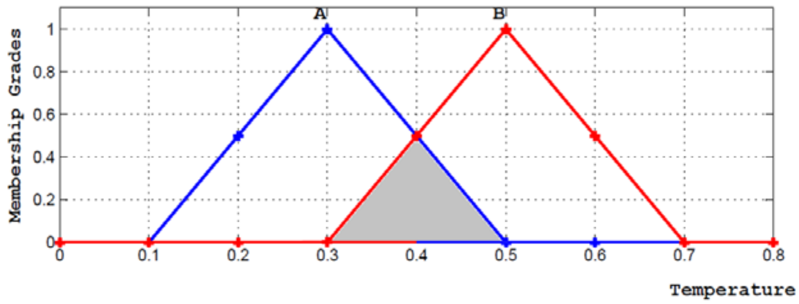


Figure 4.4: Intersection of fuzzy set *A* and *B*

3. Complement

The complement of fuzzy set A is denoted by $\overline{A}$ and its membership function is defined as:

$$\mu_{\overline{A}} = 1 - \mu_A(u) \tag{4.10}$$

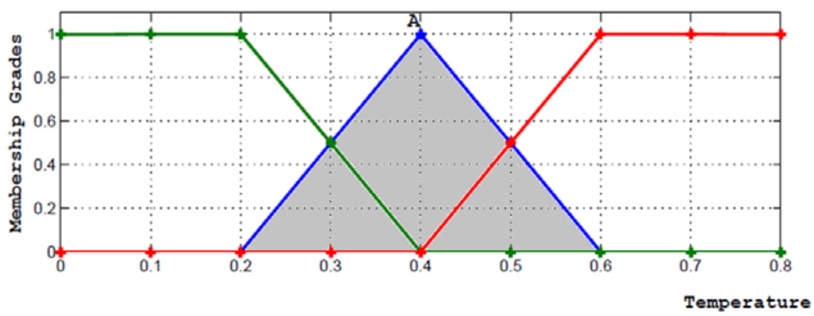


Figure 4.5: Complement of fuzzy set *A*

4. Cartesian product

If $A_1, A_2, \ldots, A_n$ are fuzzy set in the universe of discourses $U_1, U_2, \ldots, U_n$ respectively, then the cartesian product is a fuzzy set denoted by $U_1 \times U_2 \ldots \times U_n$ with membership function expressed as:

$$\begin{aligned}
&\mu_{A_1,A_2,\ldots,A_n}(u_1,u_2,\ldots,u_n) = \min\{\mu_{A_1}(u_1), \mu_{A_2}(u_2), \ldots, \mu_{A_n}(u_n)\} && \text{Minimum} \\
&\mu_{A_1,A_2,\ldots,A_n}(u_1,u_2,\ldots,u_n) = \{\mu_{A_1}(u_1) \times \mu_{A_2}(u_2) \times \ldots \times \mu_{A_n}(u_n)\} && \text{Product}
\end{aligned} \tag{4.11}$$

5. Algebraic product

The algebraic product of two fuzzy sets A and B is defined as:

$$A.B = \{(u, \mu_A(u).\mu_B(u)) \mid u \in U\} \tag{4.12}$$

6. Compositional Rule of Inference

If R is a fuzzy set relation $U \times V$ and A is a fuzzy set in U, then the fuzzy set B in V includes A is given by:

$$B = A * R \qquad \{A\ compsition\ R\} \tag{4.13}$$

There are two cases in the compositional rule. The first case is maximum-minimum (max-min) operation $\mu_A(u) = \max_{A \in U}\{\min(\mu_A(u), \mu_B(v))\}$, and the second case is maximum-product (max-product) operation $\mu_A(u) = \max_{A \in U}\{\mu_A(u) \bullet \mu_B(v)\}$.

7. Fuzzy implication inference

There are two important fuzzy implication inference rules: the general modus ponens (GMP) and general modus tollens (GMT). Structure of the two methods as follows:

GMP	Premise:	x is $\overline{A}$
	Rule:	$if\ (x \text{ is } A)\ then\ (y\ is\ B)$
	Consequence:	$y\ is\ \overline{B}$
GMT	Premise:	y is $\overline{B}$
	Rule:	$if\ (x \text{ is } A)\ then\ (y\ is\ B)$
	Consequence:	$y\ is\ \overline{A}$

The GMP method is closely to forward (data-driven) inference, which is used in fuzzy logic controller construction, while the GMT method is related to backward (goal-driven), and it is commonly used in expert systems. For two fuzzy sets *A* and *B*, the fuzzy inference using GMP state: "*If A is true and A implies B, then B is true*". In this statement, *A* implies *B* is the rule, where *A* is the antecedent and *B* is the consequent. The rule base may be containing several implications as:

Premise:	x is $\overline{A}$ and y is $\overline{B}$
Rule:	$if\ (x \text{ is } A)$ and $(y \text{ is } B)\ then\ (z\ is\ C)$
Consequence:	$z\ is\ \overline{C}$

4.3. FLC Structure

Figure 4.6 shows the basic configuration of MISO fuzzy system, which comprises four main building components: fuzzification method, rule base, inference mechanism, and defuzzification method. As seen in the figure, the input and output data of FLC are crisp (non-fuzzy) values. FLC components are:

1. The fuzzifier: measure the values of input variable and convert the input crisp values into suitable linguistic variables.
2. An expert and skilled operator define the knowledge base. The rule-base holds the knowledge, in the form of a set of rules, of how best to control the system.
3. The inference mechanism evaluates which control rules are relevant for the current time and then decides what the input to the plant should be.

4. The defuzzifier is the opposite operator of fuzzifier interface; it converts the conclusions reached by an inference mechanism into a real value as inputs to plant.

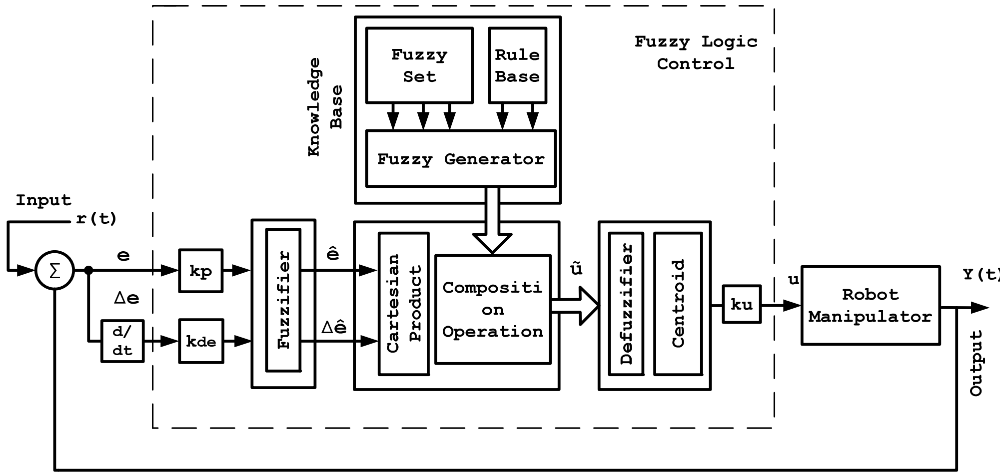


Figure 4.6: Fuzzy control system structure

Before illustrating FLC components, it is important to define the FLC inputs and output variables. As mentioned in Chapter 1, the controller is used to correct the error signal then supply appropriate input to the plant. Two inputs are used for FLC: the error that generated from the feedback loop and derivative of the error, or it may also have an integral input for fuzzy like PI controller. In addition, when designing a fuzzy like PID controller the three inputs are used, and the output is a control signal feeds the plant.

The three variables e, Δe and u of the FLC are the error, error change, and the output action, and the variables $\tilde{e}$, $\Delta\tilde{e}$, $\tilde{u}$ are their fuzzy counterparts respectively, y is the output, and r is the set point, k_e is the scale factor of the error input, k_{de} is the scale factor of the error derivative, and the k_u is the output gain.

4.3.1. Fuzzification

Fuzzification is the first block inside the controller, which scale the input crisp value into a normalized universe of discourse U , then converts each crisp input data e and Δe to a degree of membership function $\mu_A(u)$. In other word, the purpose of fuzzifier is to transform the crisp input to fuzzy set defined in U and characterized by membership function $\mu_A(u):U \to [0,1]$. The input and output memberships overlaps, and they are scaled with respect to input-output of the fuzzy control. It is labeled by linguistic terms such as short, medium, and tall. For example, if the integer value 40 is the input, the fuzzification process converts the integer value into a linguistic variable.

The number of the linguistic variables for the input output domains specified by the designer. In addition, they should be as small as possible because the larger number of linguistic variables, the more complicated inference mechanism. The domain of the input variables e and Δe are chosen from the specification of the controlled system, similarly the domain of the output variable (control signal) u is chosen according to the desired output. Each of the two inputs and output are covered by seven fuzzy set variables: NB, NM, NS, ZE, PS, PM, and PB. Figure 4.7 shows several types of membership functions [51], such as a triangular, trapezoidal, Gaussian, singleton, and the sigmoid membership functions. The most commonly used shapes are triangular, trapezoidal, and singleton. As example, the triangular membership function can be characterized by a triple points (a,b,c) where the point $(a=0)$, $(b=1)$ and $(c=0)$ are vertices of triangle, while trapezoidal membership characterized by quadruple (a,b,c,d) points where the points $((a=0),(d=0))$ and $((b=1),(c=1))$. The shape of several membership functions and its mathematical equations are shown in Appendix C.

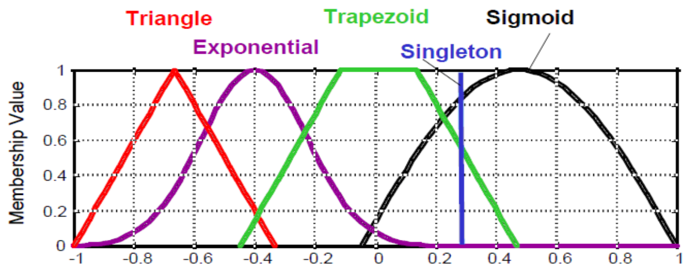


Figure 4.7: Membership functions shapes

In general, the width of each fuzzy set extends to the peak point of the adjacent fuzzy sets, and the peak value of the fuzzy set has a unity value. As an example in Figure 4.2 if u is a value between cold and warm fuzzy sets. The two fuzzy sets are active, and the membership functions of the two fuzzy sets are $\mu_{Cold}(u), \mu_{Warm}(u)$ depends on the intersection between the line of the input point and the two slops of the two fuzzy sets. As example, the crisp input $u=0.3$ has two membership functions $\mu_{Cold}(u)=0.7$ and $\mu_{Warm}(u)=0.3$. Summation of both membership functions is equal one. In this thesis, for simplicity we have used triangular fuzzy set with seven linguistic terms for both inputs, and output variables of FLC. The universe of discourse for the inputs e and Δe, and output u are partitioned into seven fuzzy sets as shown in Figure 4.8 and Figure 4.9 respectively. The universe of discourse U of the error, change of error and the output are defined as $e=[-160,160]$, $\Delta e=[-100,100]$ and $u=[-160,160]$ respectively.

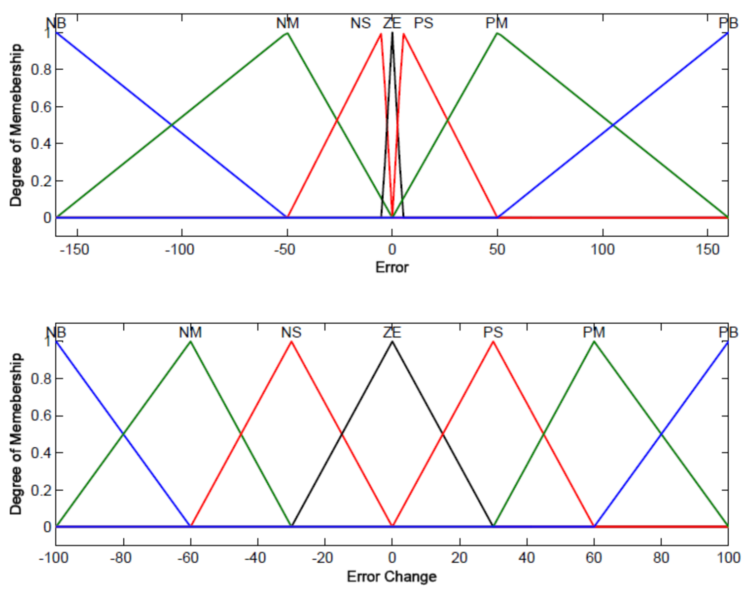


Figure 4.8: Membership function for inputs e and Δe

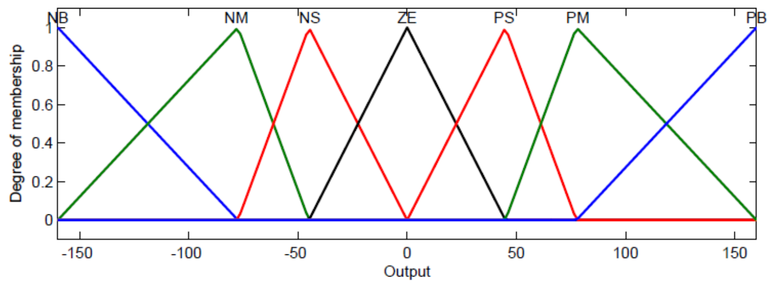


Figure 4.9: Membership function for output u

From the definition of membership function in section (4.3), it is clear that the membership functions of the error e and the output u are normal and non-symmetric, but in case of the change of error Δe, the membership function of the fuzzy sets is normal and symmetric with an exception of the end membership function for the NB and PB fuzzy sets.

It is suggested to design membership function that the peak around zero should be fixed and the width around the zero fuzzy set should be small to minimize the steady state error. Finally, fuzzy set width should be fixed, equal the difference between the corresponding fuzzy set, and neighboring fuzzy set.

4.3.2. Knowledge Base

In fuzzy control, it is important to define the state variable and the control variable of the fuzzy control. The good choice of these variables is important to have the desired behavior of the system. The function of the knowledge base is to model a human expert qualitative knowledge about the desired relationship between the input and the output of the fuzzy system [52].

Knowledge base includes three main parts as mentioned in Figure 4.6: rule base, fuzzy set and the fuzzy generator. Rule base is the core of the FLC [53]; it is a set of rules in the form of IF-THEN statement that describe the state and the behavior of the system. It consists of all fuzzy implications that used to describe the relation between the input and the output.

Selection of the rule base affects the performance of the FLC. In addition, it is used to provide and characterize the control goal and control policy of the controller into a set of linguistic rules. Rule base express the relation between inputs; it may use several variables usually in the form of conditional statements that have the following form:

$$R_j : \textbf{IF } e \textit{ is } E_j, \textbf{ AND } \Delta e \textit{ is } \Delta E_j \textbf{ THEN } u \textit{ is } U_j, \quad j = 1, 2, .., n \qquad (4.14)$$

where $e, \Delta e,$ and u are the input and output of fuzzy system, E_j, ΔE_j and U_j are fuzzy variables with fuzzy membership function $\mu_E(e)$, $\mu_{\Delta E}(\Delta e)$ and $\mu_U(u)$ respectively.

The second term is the fuzzy set data. It contains of the information about the universe of discourse, number of the input and output and the number of the membership functions. It contains the fuzzy set, which quantify the statement of the input and the output such as: u *is* U and y *is* Y. Finally, fuzzy generating tends to access the fuzzy set and fuzzy implication to generate a fuzzy relation for each implication was defined in the rule base.

Many categories was proposed to form the IF-THEN statement. Two prevalent methods are Mamdani and Takagi-Sugeno [54]. The first method, is considered as the most popular method is used to design FLC because it is simple to be implemented. The second is Takage-Sugeno (TS) method, which is called TS fuzzy rule. The two methods have a fuzzy antecedent part. The antecedent evaluation of the IF-THEN statement is the same for both methods, but the essential difference between them lies in the consequent part structure, where the consequent part of TS rule is a function of real value. The next two rules structure describes Mamdani and TS fuzzy methods.

Mamdani style

$$\textbf{IF } e_i \text{ is } E_i \textbf{ AND / OR } \Delta e_j \text{ is } \Delta E_j \textbf{ THEN } u_{ij} \text{ is } U_{ij} \tag{4.15}$$

Sugeno style

$$\textbf{IF } e_i \text{ is } E_i \textbf{ AND / OR } \Delta e_j \text{ is } \Delta E_j \textbf{ THEN } u_{ij} = k_i u_i + k_j u_j \tag{4.16}$$

In Mamdani fuzzy rule method, the knowledge of the system (antecedent) and the set of the action (consequent) depend on the human operator. Through the thesis, Mamdani fuzzy rule is adopted to construct the fuzzy model of the system, which is a robot manipulator with SISO. This method was structured with two inputs as antecedent and one output as the consequence with variables (e, Δe, u and R) as error, error change, output, and rules respectively.

Generally, for nonlinear systems the linguistic rule is one of four forms. Each one of these forms expressed linguistically as:

Single Input single output (SISO)

$$\textbf{IF } e \textit{ is } E \textbf{ THEN } u \textit{ is } U \tag{4.17}$$

Multi input single output (MISO)

$$\textbf{IF } e_1 \textit{ is } E_1 \textbf{ AND } e_2 \textit{ is } E_2 \textbf{ THEN } u \textit{ is } U \tag{4.18}$$

Single Input multi output (SIMO)

$$\textbf{IF } e_1 \textit{ is } E_1 \textbf{ THEN } u_1 \textit{ is } U_1 \textbf{ AND } u_2 \textit{ is } U_2 \tag{4.19}$$

Multi input multi output (MIMO)

$$\textbf{IF } e_1 \textit{ is } E_1 \textbf{ AND ... AND } e_r \textit{ is } E_r \textbf{ THEN } u_1 \textit{ is } U_1 \textbf{ AND ... AND } u_r \textit{ is } U_r \tag{4.20}$$

In this thesis, the rule base is constructed as referred in the second and fourth cases. These cases will use in Chapter 4 and Chapter 5 respectively.

There are some techniques are used for generating rule base. These techniques may combine to construct an effective method to derive rule base [45] and [47]. Derivation rule base depends on using the antecedent (error and error change) and the consequent (control signal) to describe the relation between the inputs and outputs of the fuzzy controller. First technique is based on the knowledge of the controlled system by analyzing the behavior of the system. We can use the input output relation that obtained from system identification to generate a set of fuzzy rules to achieve the optimal behavior of the system. Although this method does not effective for all systems due to complexity and nonlinearity, it gives good results.

The second method is based on experienced human operator [52]. In many industrial control systems, the relation between the inputs and the output are not known with high precision to employ the control algorithms, so, human operator can generate a set of fuzzy IF-THEN rules to control the process. This method gives popularity for FLC in industrial fields because it does not require a mathematical model of the complex systems, but the main drawback of this method is it dependency on the knowledge of the human [47]. Another technique depends on the learning algorithm of the fuzzy controller. It based on the ability to create fuzzy control rules and modifying them based on experience and system behavior.

The required rules used to control the plant using Mamdani method are derived in appendix C.

4.3.3. Inference Mechanism

Inference mechanism process is obtaining the relevant control rule at the current time then decides what the output of the controller should be. The membership function value for each rule for controller input is calculated using fuzzy inference mechanism (implication). The following rule base R_1 is expressed as:

$$R_1 : \textbf{IF } e \textit{ is } E_1 \textbf{ THEN } u \textit{ is } U_1 \tag{4.21}$$

The fuzzy implication is expressed as a cartesian product of the antecedent and the consequent as $R_1 = E_1 \times U_1$. In addition, this is similarly done for all rules. Several ways are used to implement fuzzy inference methods, but the most widely used in control applications are Mamdani method (minimum-maximum) and Larsen method (product-maximum) [55].

$$\begin{aligned} &\mu_{R_1}(e,u) = \min\{\mu_{E_1}(e), \mu_{U_1}(u)\} \qquad \textit{Minimum} \\ &\mu_{R_1}(e,u) = \{\mu_{E_1}(e) \cdot \mu_{U_1}(u)\} \qquad \mathrm{Pr}\textit{oduct} \end{aligned} \tag{4.22}$$

In Mamdani method, the output membership function for the clipped output at the minimum height of the two input memberships, and in the second method the output membership function is the product of the two input membership functions. To illustrate the operation of inference mechanism, this is given by the following example. Assume the FLC has two inputs E_1 and E_2 and output U, all of them have universe of discourse [-1, 1], with peak points (-0.5, 0, 0.5), (-0.3, 0, 0.3), and (-0.5, 0, 0.5) respectively.

Before solving this example, the inference mechanism is divided into three steps. First, specifying the inputs and determine the degree of its membership function. The second, applying the desired operation such as minimum or product operations. The final step aggregates, all the outputs, then combine them using maximum or sum aggregation method.

For the two inputs E_1 and E_2, where each one has three membership functions, it is clear that there are nine rule base. If two inputs $x_1 = 0.1$ and $x_2 = 0.2$ are chosen, then only four rules will be active. These rules are:

$$\begin{aligned} &R_5 : \textbf{IF } x_1 \textit{ is } A_2 \textbf{ AND } x_2 \textit{ is } B_2 \textbf{ THEN } u \textit{ is } C_2 \\ &R_6 : \textbf{IF } x_1 \textit{ is } A_2 \textbf{ AND } x_2 \textit{ is } B_3 \textbf{ THEN } u \textit{ is } C_3 \\ &R_8 : \textbf{IF } x_1 \textit{ is } A_3 \textbf{ AND } x_2 \textit{ is } B_2 \textbf{ THEN } u \textit{ is } C_3 \\ &R_9 : \textbf{IF } x_1 \textit{ is } A_3 \textbf{ AND } x_2 \textit{ is } B_3 \textbf{ THEN } u \textit{ is } C_3 \end{aligned} \tag{4.23}$$

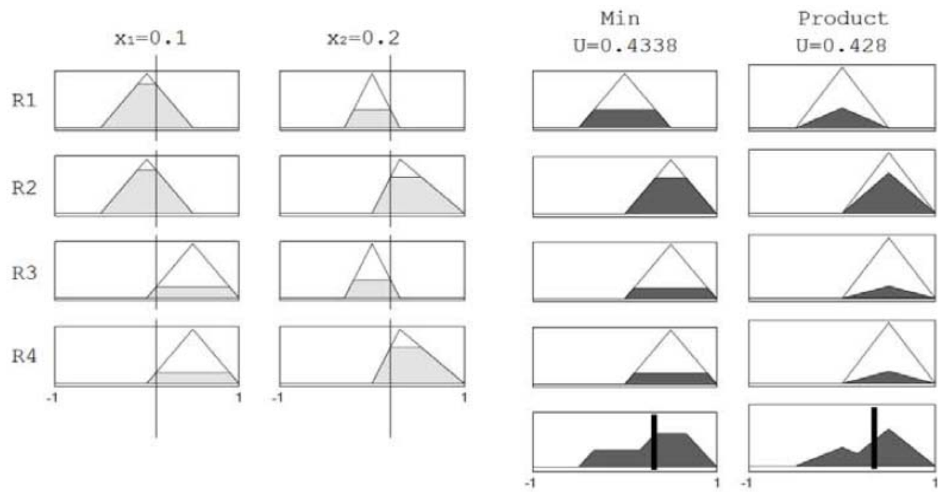


Figure 4.10: Inference process using Mamdani and Larson methods

From Figure 4.10, the values of the membership functions of the proposed fuzzy sets are: $\mu_{A2}(x_1) = 0.8$, $\mu_{A3}(x_1) = 0.2$, $\mu_{B2}(x_2) = 0.18$ and $\mu_{B3}(x_2) = 0.82$. Two inference methods are used to calculate the weight of the four rules: first, Mamdani method. The weight of the rules is $w_1 = \min(\mu_{A2}(x_1), \mu_{B2}(x_2)) = \min(0.8, 0.18) = 0.18$. Similarly, $w_3 = 0.18$ and $w_4 = 0.2$. Weight of the rules solved using Larson inference mechanism is $w_1 = \mu_{A2}(x_1).\mu_{B2}(x_2) = 0.144$, similarly $w_2 = 0.656$, $w_3 = 0.036$ and $w_4 = 0.164$. The control action of both rules will be determined using a centroid of area defuzzification method in the next section.

In this example, the inference mechanism is called *First Inference Then Aggregation* (FITA), since the output of each rule is inferred first using composition rule of inference in each rule of the rule set, and the overall output is obtained using the max operation. Simulation test is carried out with the *Fuzzy Toolbox for MATLAB and SIMULINK.*

4.3.4 Defuzzification

Defuzzification method is the final stage of the FLC. After the inference mechanism is finished. The defuzzification method tends converts the resulting fuzzy set into a crisp values that can be sent to the plant as a control signal.

In general, there are several methods used for defuzzification such as centroid of area (COA), maximum method (MM), mean of maximum (MOM), and bisector of area (BOA) [56]. The most frequently used are COA, MM, and MOM methods. First, MOM method produces the control action that represents the mean value of the all control actions whose membership function has the max value. Second is MM method that produces control action, which the fuzzy set reaches the maximum point. This method is divided into two parts: first, the smallest of max (SOM), which has the minimum value of the support of fuzzy set. The second method is the largest of max (LOM), which has the maximum value of the support of the fuzzy set.

The last is COA method that is used through this thesis. This method produces a control action that represents the center of the output of the fuzzy set. The weighted average of the membership function or the COA bounded by the membership function curve and it is converted to a typical crisp value. This method yield:

$$u = \frac{\sum_{i=1}^{m} \mu(x_i).x_i}{\sum_{i=1}^{m} \mu(x_i)} \tag{4.24}$$

A graphical interpretation of defuzzification methods is presented in Appendix C. Figure 4.11 shows the various defuzzification methods.

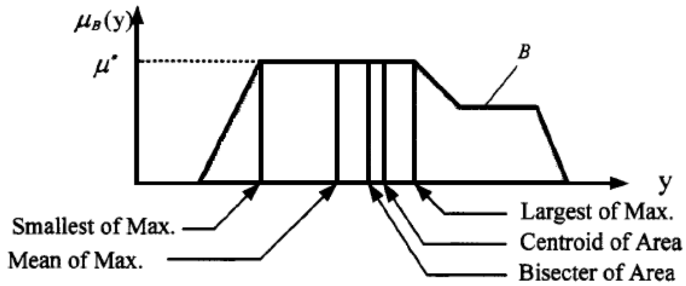


Figure 4.11: Defuzzification methods

The control action for the values, which obtained using Mamdani and Larson methods through section (4.4.3) using COA method, is determined as follows.

Mamdani Method

$$\text{Control action} = \frac{\sum_{i=1}^{m} \mu(x_i).x_i}{\sum_{i=1}^{m} \mu(x_i)} = \frac{0.18*0+0.8*0.5+0.18*0.5+0.2*0.5}{0.18+0.8+0.18+0.2} = 0.4338$$

Larson Method

$$\text{Control action} = \frac{0.144*0+0.656*0.5+0.036*0.5+0.164*0.5}{0.144+0.656+0.036+0.164} = 0.428$$

4.4. Fuzzy Control

Fuzzy PID controller has been applied successfully in many applications of control fields. Different design methods of fuzzy PID controller are used nowadays [10], [26] and [57] that present significant performance comparing to classical PID controller; moreover, designing and tuning fuzzy PID controller with simple structure gives satisfactory results for the system. Fuzzy PID controller is good controller because its components help to provide convenient closed loop response characteristics. It combines the advantages of each term of PID controller individually as mentioned in Chapter 3, taking into account that the fuzzy PID terms are better than the classical PID terms [46]. Fuzzy proportional term reduces the rising time, while the fuzzy integral term eliminates the steady state error, and the third fuzzy derivative term enhances the system stability by reducing the overshoot and improving the transient response.

Generally, the number of rules, which cover all possible inputs equal number of fuzzy sets of first input multiplied by the number of the fuzzy sets of the second input. As an example, a fuzzy controller with two inputs (error and change of error) each one has 7 fuzzy sets. The possible rules are 49 (7×7) rules that cover all possible input variation. Similarly if number of the inputs are (increased / decreased) the number of the rules will (increased/decreased) [28]. As mentioned above, there are several types of fuzzy PID controller discussed. Through this thesis, two categories of fuzzy PID controller will be discussed as follows:

1. Three inputs structure with and without coupled rules.
2. Two inputs structure with and without coupled rules.

In the first category, the similarities between the three input structure of fuzzy PID controller with and without coupled rules each have three inputs and one output. In the first structure, the three inputs fed the same fuzzy PID controller, but in the second. Each input feeds individual fuzzy controller.

Type I of the first category is three input structure with coupled rules. This type is difficult because the number of the rules increases as the number of the input variables increases. For example, if there are three inputs each one has N fuzzy sets, this means that the total number of the required rules for this structure is $N_1 \times N_2 \times N_3$. So this operation becomes very tedious and complicated. This requires the computer to process a huge database, mathematical operation and longer computational time. Type I of fuzzy PID controller expressed as:

$$\textbf{IF } e \text{ is } E \textbf{ AND } \Delta e \text{ is } \Delta \text{E} \textbf{ AND } \sum e \text{ is } \sum E \textbf{ THEN } u_{PID} \text{ is } U_{PID} \tag{4.25}$$

The final PID control output is produced after taking the cumulative sum of the FLC output as shown in Figure 4.12.

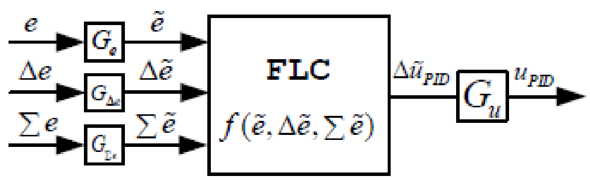


Figure 4.12: Three input fuzzy PID (Type I)

Type II is the three input structure without coupled (with decoupled) rules. Figure 4.13 shows structure of fuzzy PID controller of the three inputs with decoupled rules. In this structure, each one of the controller inputs has independent controller unlike type I. In addition, the control output is represented by a separated set of rules.

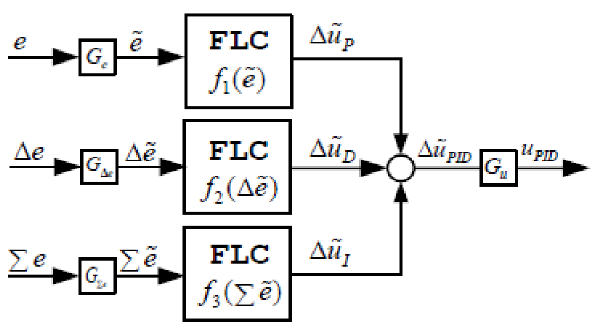


Figure 4.13: Three input fuzzy PID (Type II)

The rule base in this structure is represented as follows:

$$\begin{aligned} &\textbf{IF } e \text{ is } E \textbf{ THEN } u_P \text{ is } U_P \\ &\textbf{IF } \Delta e \text{ is } \Delta \text{E} \textbf{ THEN } u_D \text{ is } U_D \\ &\textbf{IF } \sum e \text{ is } \sum E \textbf{ THEN } u_I \text{ is } U_I \end{aligned} \tag{4.26}$$

The inference of each rule is independent and the output constitutes three separated nonlinear functions. As mentioned in type I if there are N fuzzy sets for each of the three inputs, then the total number of the required rules is $N_1 + N_2 + N_3$.

The second category of fuzzy PID controller is the structure of the two inputs with and without coupled rules. This category has two approaches to reduce fuzzy PID controller rules. Both of them take the advantages of combination between the fuzzy PD and fuzzy PI controllers. Type I is represented by two inputs with coupled rules when combining both PI and PD actions as shown in Figure 4.14. In this case, as an example if each one of the two inputs has N fuzzy sets. The total number of the required rules equal to $(N_1 \times N_2 + N_1 \times N_3)$.

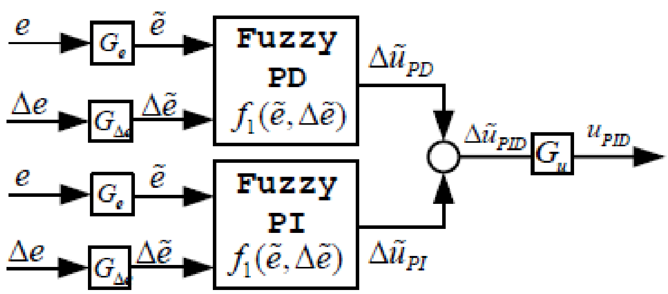


Figure 4.14: Two input fuzzy PID (Type I)

Structure of the rule base of fuzzy PID with coupled rules is:

$$\begin{aligned}&\textbf{IF } e \text{ is } E \textbf{ AND } \Delta e \text{ is } \Delta \mathrm{E} \textbf{ THEN } u_{PD} \text{ is } U_{PD}\\&\textbf{IF } e \text{ is } E \textbf{ AND } \sum e \text{ is } \sum E \textbf{ THEN } u_{PI} \text{ is } U_{PI}\end{aligned} \tag{4.27}$$

The second type has two inputs without coupled rules. In this type, the individual proportional and derivative actions are generated by the input e and Δe respectively. Integral action inferred from the proportional term by taking the sum of the proportional action. The total number of rules is equal $(N_1 + N_2)$, and the rule base corresponding to this action as follows:

$$\begin{aligned}&\textbf{IF } e \text{ is } E \textbf{ THEN } u_P \text{ is } U_P\\&\textbf{IF } \Delta e \text{ is } \Delta \mathrm{E} \textbf{ THEN } u_D \text{ is } U_D\end{aligned} \tag{4.28}$$

Figure 4.15 shows the general structure for Type II of fuzzy PID controller with two inputs.

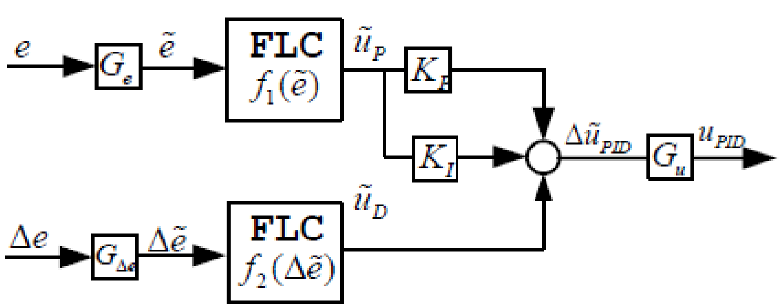


Figure 4.15: Two input fuzzy PID (Type II)

Figure 4.16 shows the effect of fuzzy PD and fuzzy PI controller. Assume the reference input r = 60 implemented for the DC motor that in Chapter 2. The response shows that the fuzzy PD has a faster response $t_r = 0.2$ sec than fuzzy PI $t_r = 0.3$ sec. This means that the fuzzy PD controller rising time is less 33% than fuzzy PI rising time. However, the fuzzy PD controller has large steady sate error $SSE = 0.04$ than the fuzzy PI controller where $SSE = 0.002$.

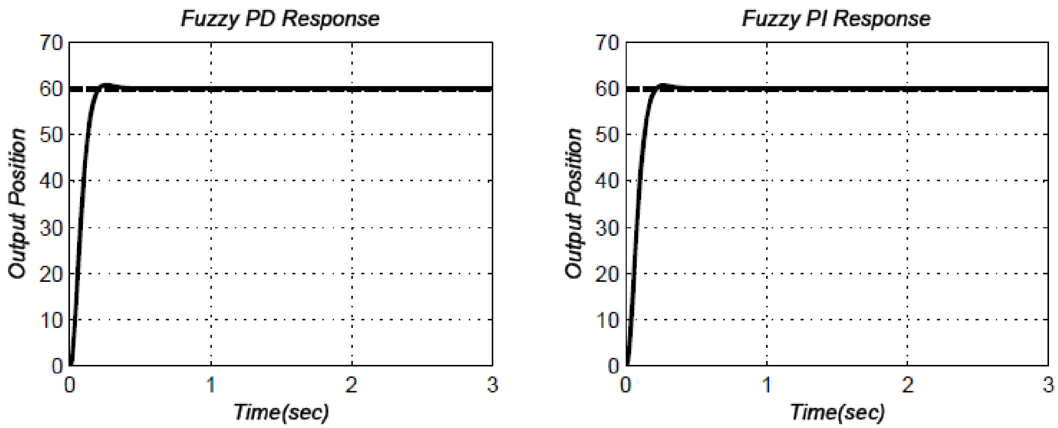


Figure 4.16: Output of fuzzy PD and fuzzy PI for *r* = 60.

There are two types, as illustrated in Figure 4.17(a) and Figure 4.17(b), that combines two inputs and one input, which are used to reduce the PID rules. The first type denoted as fuzzy PD + fuzzy I. Three inputs had been used to fuzzy PD and fuzzy I: the error and change of the error that feeds the fuzzy PD controller and the last input feeds the fuzzy I controller. As mentioned previously the first term of this controller fuzzy PD enhances the system stability and the second is used to reduce the steady state error. In this case, the fuzzy rules will equal $(N_1 \times N_2 + N_1)$ rules.

If the fuzzy I control is replaced by classical integral action as illustrated in Figure 4.17(b), the control output u is the sum of the fuzzy PD controller and the integral action of the error. The number of the fuzzy rules will equal $(N_1 \times N_2)$ rules. First, the term fuzzy PD gives the system stability, then the pure integral term reduces the steady state error.

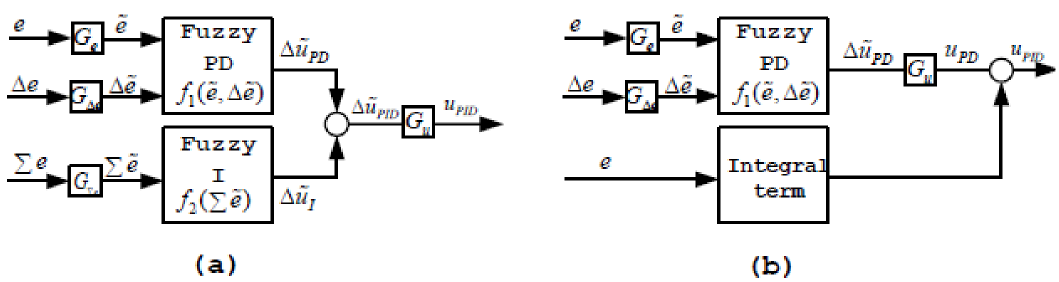


Figure 4.17: a) fuzzy PD + fuzzy I, b) fuzzy PD +I

4.5. Summary

Chapter 4 was described and covered FLC. In section (4.3), basic notation and definitions of a fuzzy logic and fuzzy set were presented and defined. Following this some definitions for membership functions and operations, which are used in fuzzy theory were introduced. In the following section (4.4) and its subsections, the structure of the FLC was presented. It consists of four components: fuzzifier, rule base, inference mechanism, and defuzzifier. Finally, section (4.5) was presented method for designing fuzzy controller and the effect of the number of the input and output variables on the derivation of the rule base.

This chapter is concluded by the following characteristics of the FLC in comparison to PID control techniques, which is used as reference benchmark for other techniques:

- Fuzzy logic control deals with the system without requiring a lot of information about the dynamics of the system.
- Fuzzy logic controller is nonlinear control and the design of FLC parameters depends on the characteristics of the system.
- Fuzzy logic controller is more robust than a classical controller is, and it uses the linguistic data to find the suitable step response.

CHAPTER 5 FUZZY SUPERVISORY CONTROLLER

5.1. Introduction

Although a large amount of literature discussed the FLC and PID control problem, the research in tuning PID parameters using supervisory control is still at an early stage.

In fuzzy theory, FLC grouped into two classes [57]: direct fuzzy controller (DFC) and indirect fuzzy controller (IFC). Through this thesis, the first type of fuzzy controller is used. DFC computes the controller action through the fuzzy mechanism, then the output control signal of the fuzzy controller feeds the plant. DFC is classified into three categories: First, the direct (pure) fuzzy controller, second FSC, and the third is a combination between both two categories [10]. In the first and the second type of direct fuzzy controller, the fuzzy reasoning attempts to provide a nonlinear action to control the output. Therefore, the fuzzy tuning PID controller is considered as nonlinear controller in contrast to linear PID controller in the conventional theory.

An adaptive controller is "***a controller with adjustable parameters and a mechanism for adjusting parameters***" [51]. A supervisory controller is a type of adaptive controllers since it searches to observe the current behavior of the plant and modifies the controlling unit "PID" to improve the performance. In other words, the design task of FSC has two main issues: first determine the knowledge parameter to generate the necessary crisp control action, then tune these parameters to obtain the desired performance for the plant. Fuzzy adaptive controller has two types: in first type, the rule base of the FLC is updated via a fuzzy system [58]. In the second type, the supervisory controller tends to monitors the controller when it does not properly tuned, then seeks to adjust the controller to obtain the desired response. The appreciation of fuzzy controller is easier to implement because it needs a little knowledge about the process, it reduces the difficulty of fine-tuning of the PID parameters and improves the response online during operation.

FLC is better than the PID controller when dealing with nonlinear systems. On the other hand, PID controller is used in more than 90% of control applications. Therefore, it is costly to replace the PID controller with FLC in all applications. Thus, one of the motivating and attractive solutions is to combine the PID controller and FLC to form the FSC. In real environments, unexpected changes in system characteristics may occurs, which causes the system to produce undesirable output. Therefore, the adaptive "supervisor" controller has the capabilities to change the PID parameters under varying process conditions. In other words, it should change the parameters of the control system to prevent the undesired behavior from occurring because the adaptive action is based upon the performance indices of the system.

5.2. Fuzzy Supervisory Controller Structure

This section is divided into three sections: first PID controller, second FLC and the third is FSC. As mentioned previously PID controller is considered as one of the most popular control technique used to control systems. Designing of the PID controller tends to tune K_P , K_I and K_D parameters to observe how the system effect by these parameters.

Tuning PID parameters is classified into two methods: first, classical tuning and the second is tuning PID using supervisory controller [59]. PID parameters are fixed during process control after they have been tuned for the first time. Since the fixed PID, parameters do not produce satisfactory results for systems. Therefore, this method is not a good choice in nonlinear circumstances. Therefore, the second method is used to overcome the limitation of tuning PID gains. This method depends on tuning these parameters on-line during process operation. This is accomplished using FLC, as supervisory for tuning PID parameters.

Some rules for tuning PID parameters are discussed as follows:

- If the error $e(t) = r(t) - y(t)$ is positive large, then the proportional gain K_P must be large, integral term K_I small and the derivative term K_D is small. Therefore, this will speed the system output.
- If the current error is very small, the PID parameters will have to be a smaller value for proportional gain, larger value of integral time constant and larger value of derivative gain. Therefore, the speed of the system response will be small to reduce the overshoot of the output.
- If the current error and its first difference are approaching zero (i.e. the system output will approach the steady state), then these parameters of PID controller must keep the values of the last state (i.e. the changes of these parameters must be very small). Therefore, this work will maintain the system output at the set point, and let the system output approach the steady state, etc....

Classical tuning methods are boring because a lot of time has to be spent to obtain the values of the three parameters. Therefore, FLC is a good choice for tuning PID gains online. Research in fuzzy controller design method is divided into two strategies: fuzzy control structure design and fuzzy control parameters design methods. First strategy intends to achieve the optimal control solution based on choice of the fuzzy controller structure such as number of inputs and outputs, fuzzification method, number and shape of membership function, defuzzification method, and rule base structure. Disadvantages of the first that the complexity of the fuzzy controller is increased as the number of the inputs increased. On the other side, the purpose of fuzzy controller parameter design is to find the optimal control solution by adjusting the parameters of the fuzzy controller with fixed structure. The parameter adjusting depends on the shape and the position of the membership function, and the scaling factor of the fuzzy controller.

The third controller is the FSC. Many papers [34], [35], [60], [61] and [62] discussed the FSC problem. FSC means that the fuzzy controller tunes PID parameters K_P, K_I and K_D. The constant PID parameters are not properly for nonlinear plants with unpredictable variables. Hence, it is necessary to be tuned automatically. The basic structure of fuzzy supervisory PID controller is shown in Figure 5.1. This structure has the form of PID controller but its parameters are adopted using the FLC, which provides a rule base on the fuzzy inference. Fuzzy inference provides a rule base to change these parameters within initial parameters [62]. Regarding the supervisory structure, it consists of two levels. FLC that is represent the upper level and the PID controller as a lower level, which is tuned using FLC.

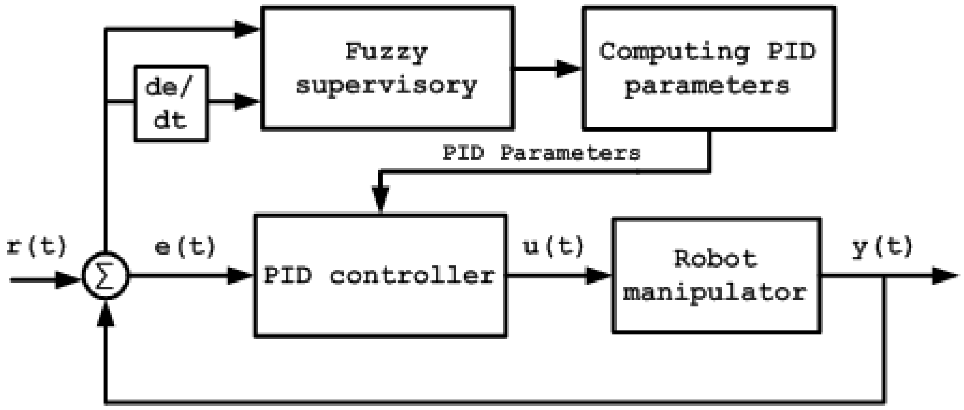


Figure 5.1: Fuzzy supervisory control structure

Fuzzy supervisory tuner has two inputs the error $e(t)$ and change of error $\Delta e(t)$, and three outputs $\tilde{K}_P, \tilde{K}_I$, and $\tilde{K}_D$. The proposed algorithm evaluates the process output at a specified time then generates the proposed values to tune PID parameters instantaneously. This process accomplished using the rule base that derived from studying the rules of tuning PID parameters [60].

The fuzzy logic tuning mechanism use the current error $e(t)$ between the actual system output $y(t)$ and the set point $r(t)$, and the first derivative $\Delta e(t)$ as inputs, which are obtained according to the following formulas:

$$e(t) = r(t) - y(t) \tag{5.1}$$

$$\Delta e(t) = e(t) - e(t-1) \tag{5.2}$$

The output of FLC generates $\tilde{K}_P, \tilde{K}_I$, and $\tilde{K}_D$ vales for tuning PID gains. The purpose of the fuzzy supervisory is to generate a desired value for each one of the three parameters such as when the process output is approaching the setpoint; the PID parameters should speed up the convergence. In addition, if the output starts to deviate from the setpoint, the PID parameters should slow that deviation down.

To implement the fuzzy tuner, the inputs and outputs may be needed to normalize in the range [-1, 1] using scaling factors. Choosing a scaling factor depends on the engineering experience. Once the two inputs fuzzified into two linguistic variables $e(t)$ and $\Delta e(t)$, these fuzzy variables feed to an inference engine, which use the fuzzy rule and fuzzy compositional rule [11]. Using the center of area (COA) defuzzification method for each one of the three tuners, we get crisp values, which are used to update K_P, K_I, and K_D parameters according to the following equations:

$$K_p = (K_{P_{\max}} - K_{P_{\min}})\tilde{K}_p + K_{P_{\min}} \tag{5.3}$$

$$K_D = (K_{D_{\max}} - K_{D_{\min}})\tilde{K}_D + K_{D_{\min}} \tag{5.4}$$

$$K_I = (K_{I_{\max}} - K_{I_{\min}})\tilde{K}_I + K_{I_{\min}} \tag{5.5}$$

where $\tilde{K}_P, \tilde{K}_I$, and $\tilde{K}_D$ are the output of FLC.

Using Mamdani method as discussed in Chapter 4, the fuzzy inference block of the controller design is shown in Figure 5.2. This figure shows the number and shape of the used input and output membership functions for the fuzzy supervisory controller.

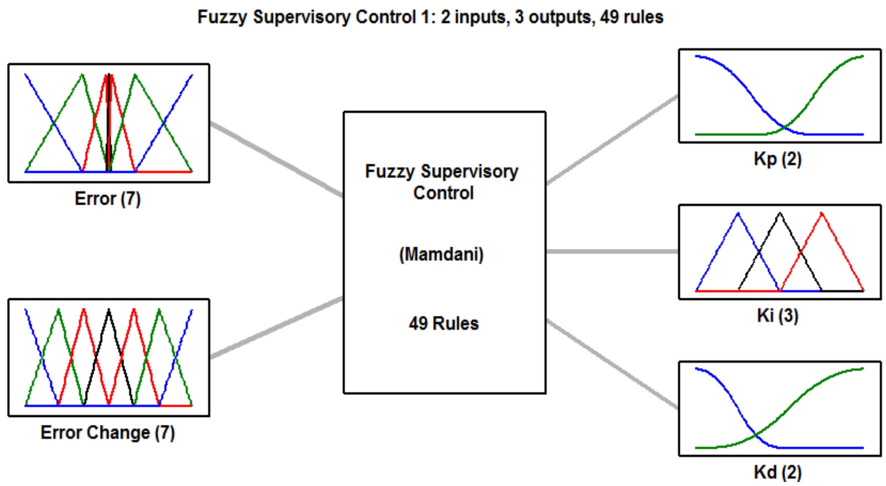


Figure 5.2: Fuzzy inference block

$\tilde{K}_P, \tilde{K}_I,$ and $\tilde{K}_D$ variations are used for tuning the corresponding values of PID controller on-line. This process decided by the inference mechanism according to the current error, e and its first difference Δe [61]. The designed processes about this controller will be introduced as following: first, decide what is the proper membership function that will be use from the two inputs of the fuzzy controller using fuzzy inference mechanism. The second step tends to fuzzify the proposed membership functions variables that defined. Figure 5.3 shows the used membership functions of the input variables, e and Δe. The linguistic variables for these membership functions are assigned from negative big (NB) to positive big (PB).

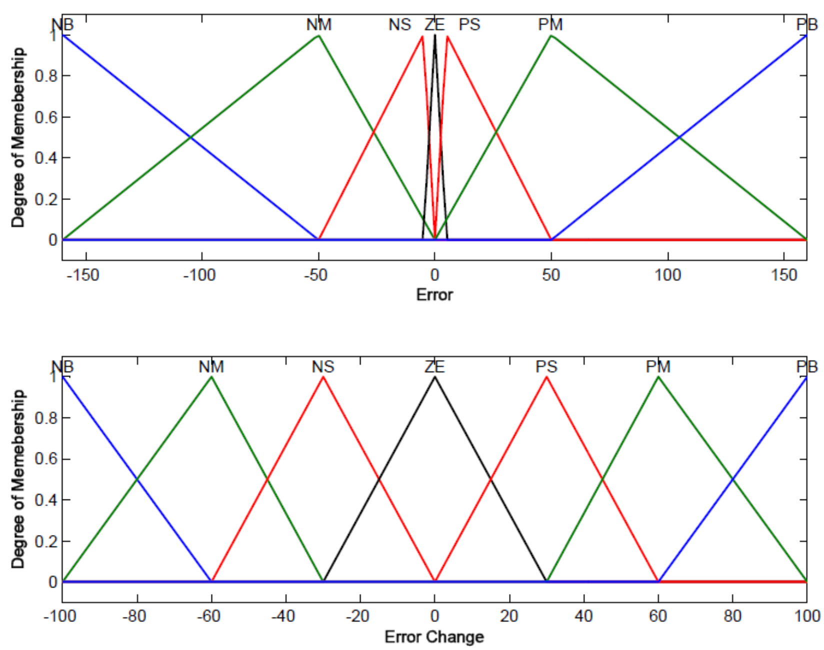


Figure 5.3: Input variables of fuzzy controller

Figure 5.4 shows the membership functions of output variables $\tilde{K}_P, \tilde{K}_I$ and $\tilde{K}_D$. The membership functions that are used for the proposed FLC tuner are triangular, Gaussian, and sigmoid membership functions.

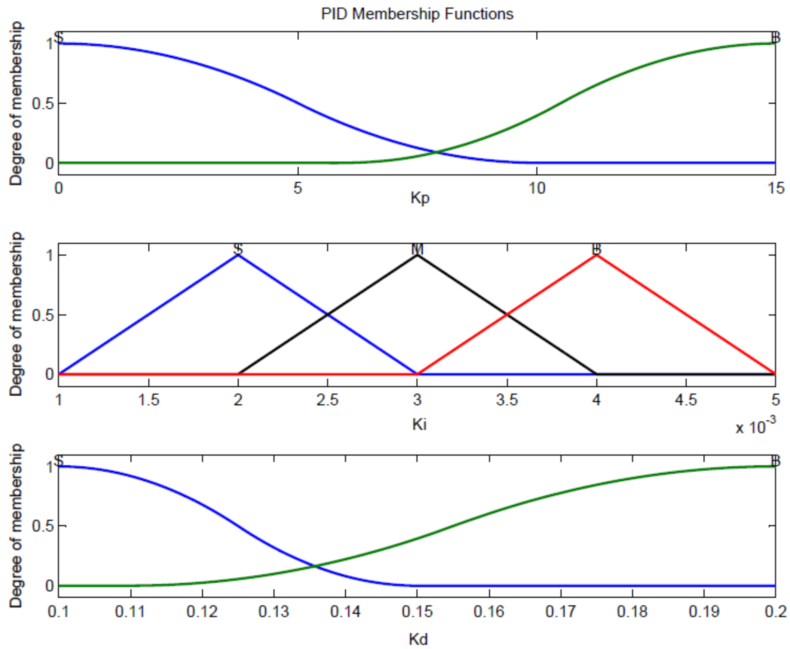


Figure 5.4: PID parameters membership functions

$\tilde{K}_P$ and $\tilde{K}_D$ output has two membership functions in sigmoid shape chosen for tuning the proportional and derivative gains of PID controller. The fuzzy set variables of the $\tilde{K}_P$ and $\tilde{K}_D$ are small (S) and big (B). The term $\tilde{K}_I$ is used for tuning the integral term. It has three membership functions in triangular and it covers three fuzzy set variables: S, Medium (M), and B.

After the required membership function of each input is decided, then the fuzzy inference mechanism will be established. Based on the result values which obtained using classical tuning methods, the fuzzy inference rules are established in Table (D.1) in Appendix D. For example, if the error is very large and positive and Δe approaches zero, then the PID parameters must be as follows: larger proportional gain, larger integral gain, and smaller derivative gain. Finally defuzzify the output variables of fuzzy mechanism. In building the supervisory rule base, the knowledge on tuning PID parameters gained from the experience of the desired controller. As classical PID controller, the $\tilde{K}_P$, $\tilde{K}_D$ and $\tilde{K}_I$ parameters play an important role in achieving a desired performance.

5.3. Fuzzy Controller Design

Generally, fuzzy controller design is important because the good structure of the FLC leads to improve the system performance.

Determination and evaluation of the inputs and outputs, number and shape of the membership functions, scaling factors, rule base, inference method, and the system stability are characterize the FLC design [60]. The previous factors will be discussed below. To design a fuzzy logic controller determine:

- The number of the inputs and outputs
- The range of the input and output values

- Membership functions and rule base.
- Fuzzification and defuzzification strategy
- The inference mechanism
- Control and stability

The first step for designing the controller is determining number of the inputs and outputs variables. Number of the input and output variables are better to be limited because the larger input output variables the larger system complexity. The number of the inputs and outputs for the proposed controller are fixed. There are two inputs (error and change of error) and three outputs (PID parameters). Two methods used for designing FSC are discussed below. Figure 5.5 shows the first method.

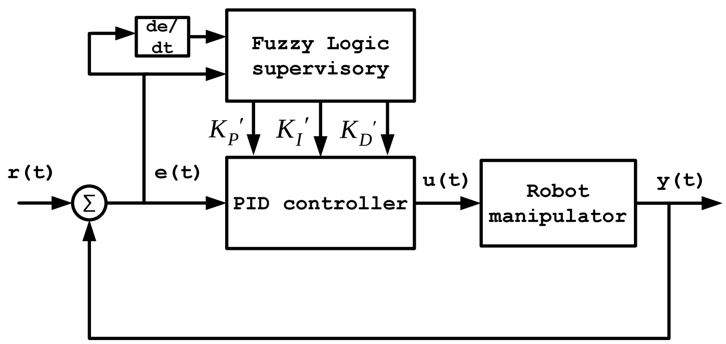


Figure 5.5: Fuzzy supervisory: first method

In the first method, the error and the changes of error are considered as the inputs of the FLC, then the output of FLC feed the PID controller. This method is defined as multi-input multi-output (MIMO) because the output of the fuzzy controller (PID parameters) feeds the PID controller as a package.

The second method as shown in Figure 5.6. Each one of the three fuzzy tuners has two inputs error and change of error and one output K_P', K_I', or K_D' feed the three parameters of the PID controller K_P, K_I and K_D respectively.

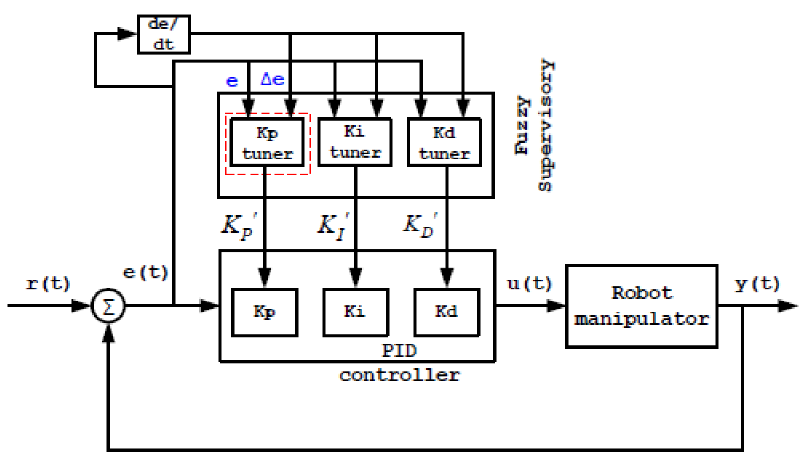


Figure 5.6: Fuzzy supervisory: second method

This method is defined as multi-input single-output (MISO) because each one of PID parameters has an independent tuner. This means that the output of each fuzzy tuner feeds the PID parameters independently. The dashed block represents a fuzzy tuner for K_P. Fuzzy tuners for K_I and K_D have the same structure.

The second and the third step for designing fuzzy controller guarantee that there is a good input signal applied to the controller if they are properly chosen. The range of the PID parameters is chosen into two ways: First, choose the range of the three parameters as $K_{x_{\min}} < K_x < K_{x_{\max}}$, where K_x represents K_P, K_I, or K_D. Second, determine the fixed PID parameters, which is obtained using Z-N tuning method, then choose values around it.

Defining the membership function is essential part of the expert system because the proper choice of the membership function guarantees a good design of the rule base. As mentioned, the rule numbers increase with the number of linguistic terms per input variable. Therefore, the number of terms per parameters should be kept low. Rule base may choose depends on the control engineer. The good choice of the rule provides a better starting point than the week choice. As discussed in the first step, two methods for design the FSC. Each one of both structures has rule base may be written in the following two forms. The first structure has the rule base in MIMO form as:

$$\textbf{IF } e \textit{ is } E \textbf{ AND } \Delta e \textit{ is } \Delta E \textbf{ THEN } k_p \textit{ is } K_P \textbf{ AND } k_i \textit{ is } K_I \textbf{ AND } k_d \textit{ is } K_D \tag{5.6}$$

Whereas the second structure has the MISO form as follows:

$$\begin{aligned} &\textbf{IF } e \textit{ is } E \textbf{ AND } \Delta e \textit{ is } \Delta E \textbf{ THEN } k_p \textit{ is } K_P \\ &\textbf{IF } e \textit{ is } E \textbf{ AND } \Delta e \textit{ is } \Delta E \textbf{ THEN } k_i \textit{ is } K_I \\ &\textbf{IF } e \textit{ is } E \textbf{ AND } \Delta e \textit{ is } \Delta E \textbf{ THEN } k_d \textit{ is } K_D \end{aligned} \tag{5.7}$$

The detail's description of the different defuzzification methods was discussed in the previous chapter. The general defuzzification method used is the center of area (COA) method. Inference mechanism as mentioned had several types used in fuzzy control. In this chapter Mamdani, inference method (max-minimum) method is used.

In design a control system, control performance depends on the system response. It is important to consider the system as a linear system to restrict the choice of the referred parameters. Number and type of membership functions, fuzzification and defuzzification methods, are used to determine the degree of implication. All these parameters influence the system output.

5.4. Rule Base Derivation

As mentioned previously, there are four methods to derive rule base [46]. Deriving fuzzy rule base is an important part in FLC design and implementation [63]. Two approaches for deriving rule base: heuristic and deterministic approaches. The first, heuristic approach it is based on the qualitative knowledge of the system behavior. The rule is formed by analyzing the behavior of the process such as the convergence from the proposed output may be correct [64]. The second approach can systematically determine the parameters and the linguistic structure of the rules, which satisfy the desired control.

Consider a unit-step response for control system is shown in Figure 5.7(a). The error signal, which is the difference between the unit step input and the output response, is shown in Figure 5.7(b). Assume the system consists of a motor with torque $\tau(t)$ that proportional to $e(t)$.

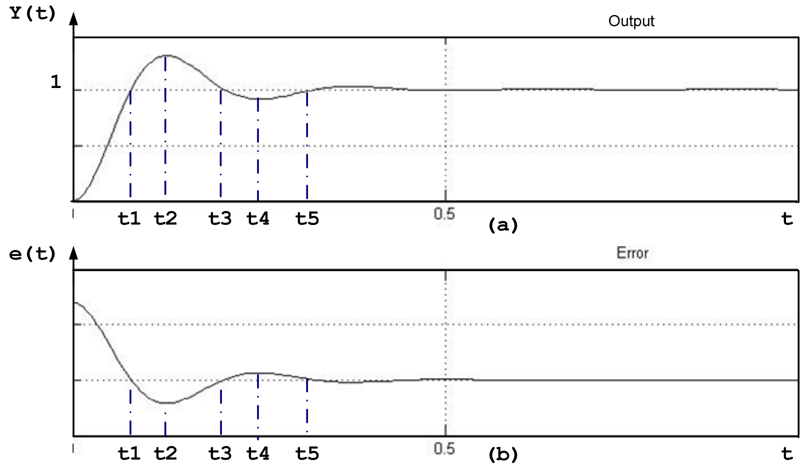


Figure 5.7: Waveform showing (a) output, (b) error

The performance of the system is analyzed as follows: During the time interval $0 < t < t_1$, the error signal is positive, and corresponding motor torque is positive and it is rising rapidly. In the time interval, $t_1 < t < t_3$, the error signal is negative and the motor torque is negative. The negative torque tends to slow down the output acceleration and cause the direction of the output $y(t)$ to opposite direction. During the time interval $t_3 < t < t_5$, the motor torque is a gain positive; thus, tends to reduce the response caused negative torque in previous time interval. Since the system will be considered stable along t.

To illustrate the above mechanism for rule derivation, Figure 5.8 shows the response of a process to be controlled, where the input of the fuzzy controller are the error and change of error.

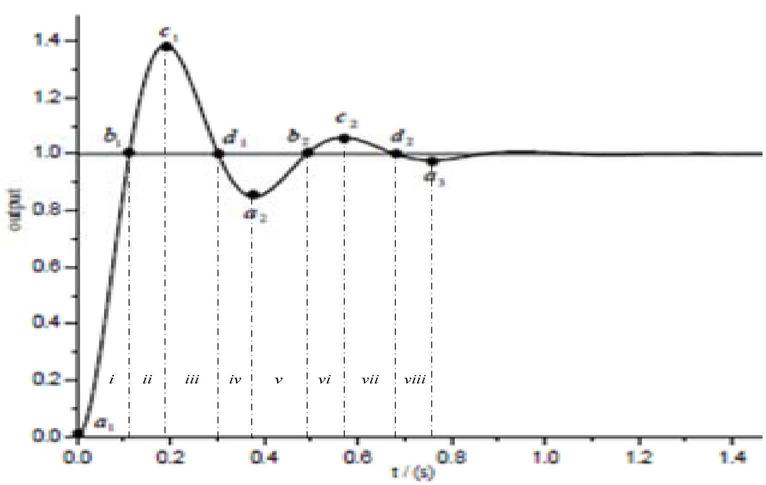


Figure 5.8: Step response

In the first region ($e > 0$ and $\Delta e < 0$) i.e. between point a_1 and point b_1 the control signal u should increased to speed up the rising time, then decreased gradually as e will approaches zero to reduces the overshoot. Similarly, (e and Δe smaller than zero) so the control signal should decrease to slow divergence of the response and eliminating the overshoot. The other two cases similarly as the above two cases.

Let studying first case around point a_1. If the error is positive big and its derivative approaches zero, then the PID parameters should have large proportional gain, the integral time constant must be smaller, and the derivative time constant must be smaller. In this case, the proportional gain K_P may represented by a fuzzy set (Big) and the derivative gain is K_D represented by a fuzzy set (Small), and the integral gain K_I is represented by a fuzzy set (Small). Therefore, the rule base a round a_1 reads as:

$$\textbf{IF } e \textit{ is } PB \textbf{ AND } \Delta e \textit{ is } Z \textbf{ THEN } K_p^{'} \textit{ is } B \textbf{ AND } K_I^{'} \textit{ is } S \textbf{ AND } K_D^{'} \textit{ is } S \tag{5.8}$$

Around point b_1 small control signal to avoid large overshoot. If the error is zero and its derivative is negative big, then the PID controller have to be smaller value of the proportional gain, larger value for the integral time constant, larger value for derivative time constant. Thus, the following fuzzy rule is written as:

$$\textbf{IF } e \textit{ is } Z \textbf{ AND } \Delta e \textit{ is } NB \textbf{ THEN } K_p^{'} \textit{ is } S \textbf{ AND } K_I^{'} \textit{ is } B \textbf{ AND } K_D^{'} \textit{ is } B \tag{5.9}$$

If the error and its difference are approaching zero (i.e. the system output will approach the steady state, at point d_2), then parameters of PID controller must keep the values of the last state (i.e. the changes of these parameters must be very small). Therefore, this step will maintain the system output at set point, and let the system output approach the steady state. The desired rule is written as follows:

$$\textbf{IF } e \textit{ is } Z \textbf{ AND } \Delta e \textit{ is } Z \textbf{ THEN } K_p^{'} \textit{ is } B \textbf{ AND } K_I^{'} \textit{ is } M \textbf{ AND } K_D^{'} \textit{ is } S \tag{5.10}$$

The fuzzy control rules of the proposed algorithm have been derived experimentally from studying the step response of the process to be controlled. All rule bases derived in the same way. The rule base table of K_P', K_D' and K_I' are shown in Table (5.1), Table (5.2) and Table (5.3) respectively.

Table 5.1: Fuzzy control rule of K_P'

K_P'		**ERROR**						
		NB	NM	NS	Z	PS	PM	PB
CANGE OF ERROR	NB	B	S	S	S	S	S	B
	NM	B	B	S	S	S	B	B
	NS	B	B	B	S	B	B	B
	Z	B	B	B	B	B	B	B
	PS	B	B	B	S	B	B	B
	PM	B	B	S	S	S	B	B
	PB	B	S	S	S	S	S	B

Table 5.2: Fuzzy control rule of K_D'

K_D'		ERROR						
		NB	NM	NS	Z	PS	PM	PB
CANGE OF ERROR	NB	S	B	B	B	B	B	S
	NM	S	B	B	B	B	B	S
	NS	S	S	B	B	B	S	S
	Z	S	S	S	B	S	S	S
	PS	S	S	B	B	B	S	S
	PM	S	B	B	B	B	B	S
	PB	S	B	B	B	B	B	S

Table 5.3: Fuzzy control rule of K_I'

K_I'		ERROR						
		NB	NM	NS	Z	PS	PM	PB
CANGE OF ERROR	NB	S	M	B	B	B	M	S
	NM	S	M	M	B	M	M	S
	NS	S	S	M	M	M	S	S
	Z	S	S	S	M	S	S	S
	PS	S	S	M	M	M	S	S
	PM	S	M	M	B	M	M	S
	PB	S	M	B	B	B	M	S

5.5. Summary

The purpose of this chapter was to combine between the conventional PID control and FLC to obtain a new control technique denoted as FSC. The idea of FSC is to employ the FLS to tune the parameters of PID controller online during process operation. Section (5.2) discussed the main structure of FSC, while section (5.3) and section (5.4) discussed the problem of designing the FLC and deriving the rule base respectively.

CHAPTER 6 RESULTS

6.1. Introduction

Implementation of the proposed controllers for the robot arm has been presented in the previous three chapters. In this chapter, the results will be discussed. MATLAB and SIMULINK are used to simulate and evaluate the performance of the proposed controllers that applied on the robot. The three types of control algorithms are PID controller, FLC, and FSC are implemented to control the 5DOF robot arm using an independent joint control mechanism. The purpose of the three controllers is to improve the performance of the robot arm to acquire the desired tasks.

6.2. Lynx6 as Case Study

The proposed controllers will be used to control the Lynx 6 robot arm as a case study. Lynx6 is an articulated manipulator RRR, with 5DOF, 5 rotational joints. The robot mounted with moving gripper at the end of the chain. Figure 6.1 shows the Lynx6 robot arm.

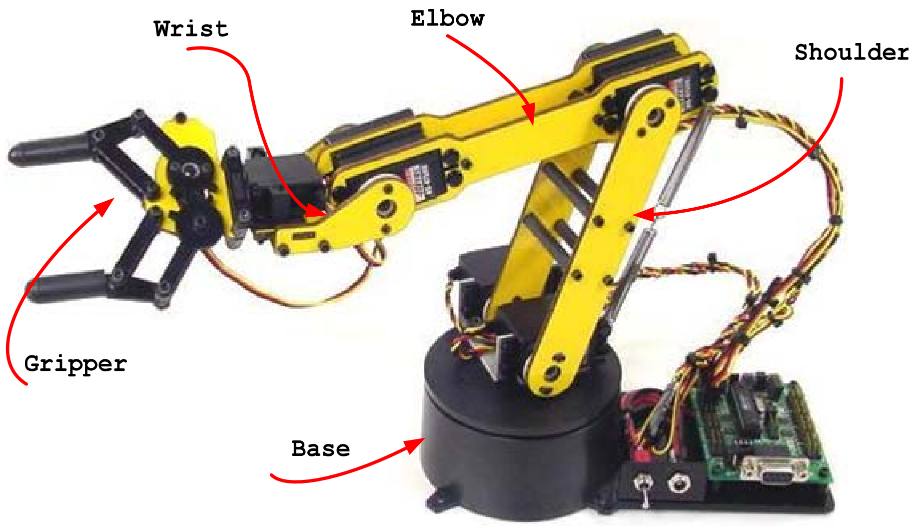


Figure 6.1: Lynx6 robot arm

(Source: http: //www.lynxmotion.com)

The 5 joints are namely as the base, shoulder, elbow, wrist, and gripper designed to catch and hold work pieces respectively. A dedicated servomotor controls each of these joints; these motors are connected to a serial servo controller card (SSC32) to control the Lynx 6 from a computer through the serial port.

As mentioned previously, this thesis was discussed the method to model and controls the different kinds of robot manipulator without regarding to the number of joint variable and its types. Lynx6 robot arm was chosen as a case study due to its small size, lightweight, and it is inexpensive unlike industrial robots such as PUMA 560. In addition, if any kind of robot manipulator available the modeling procedure will be the same. Figure 6.2 depicts a geometric model for the Lynx 6 robot arm, which will be used for its kinematics derivation. The joint angles of Lynx6 are $\theta_1, \theta_2, \theta_3, \theta_4$ and θ_5 .

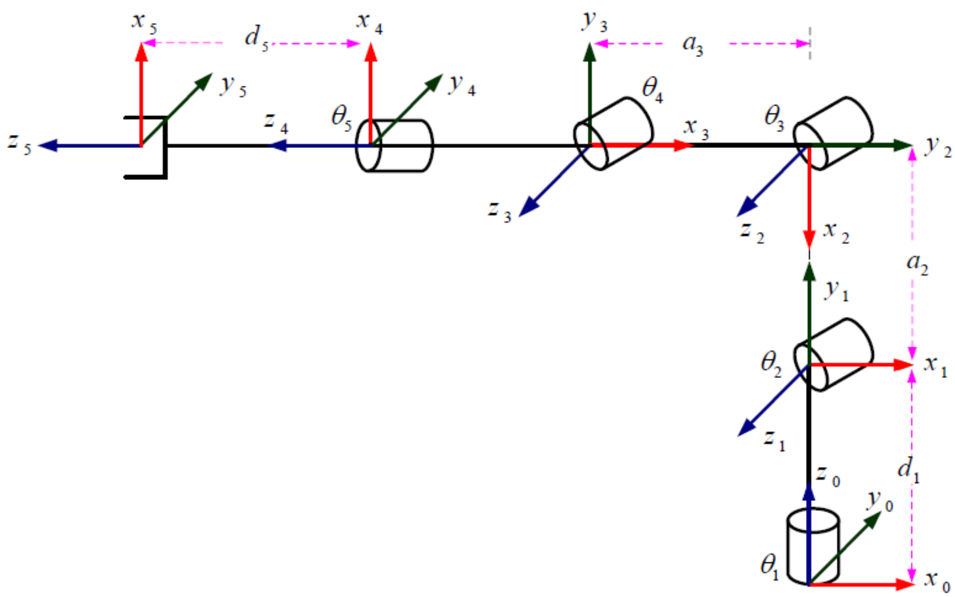


Figure 6.2: Frame assignment for the Lynx 6 robot arm.

Lynx6 robot arm contains six servomotors as seen in Figure 6.1. Table (6.1) gives joint names and the combination of actuators that control the movements of these joints.

Table 6.1: Arm joints and actuators

Joint name	Actuators
Shoulder rotation	HS-442
Shoulder	2x HS-442
Elbow	HS-442
Wrist	HS-442
Wrist rotation	HS-81
Gripper	HS-81

Table (6.2) shows the DH parameters of Lynx6 robot arm. By using the equations derived in the Chapter 2, mathematical derivation of the forward kinematic and inverse kinematic of the Lynx6 robot arm is obtained in Appendix A.

Table 6.2: DH parameter of Lynx6 robot arm

Link	Joint	a_i	α_i	d_i	θ_i
1	0-1	0	90^o	8 *cm*	θ_1^*
2	1-2	12 *cm*	0	0	θ_2^*
3	2-3	12 *cm*	0	0	θ_3^*
4	3-4	6	-90^o	0	θ_4^*
5	4-5	0	0	6 *cm*	θ_5^*

For testing the 5DOF Lynx6 robot arm, the joint desired input angles are $\theta = \{120^\circ, 66^\circ, 100^\circ, 45^\circ, 15^\circ\}$, with initial position of the robot arm is the home position $\theta_{\text{int}} = \{0^\circ, 0^\circ, 0^\circ, 0^\circ, 0^\circ\}$. Figure 6.3 shows the home position for the Lynx6 and Figure 6.4 shows the final configuration for the input joint variables.

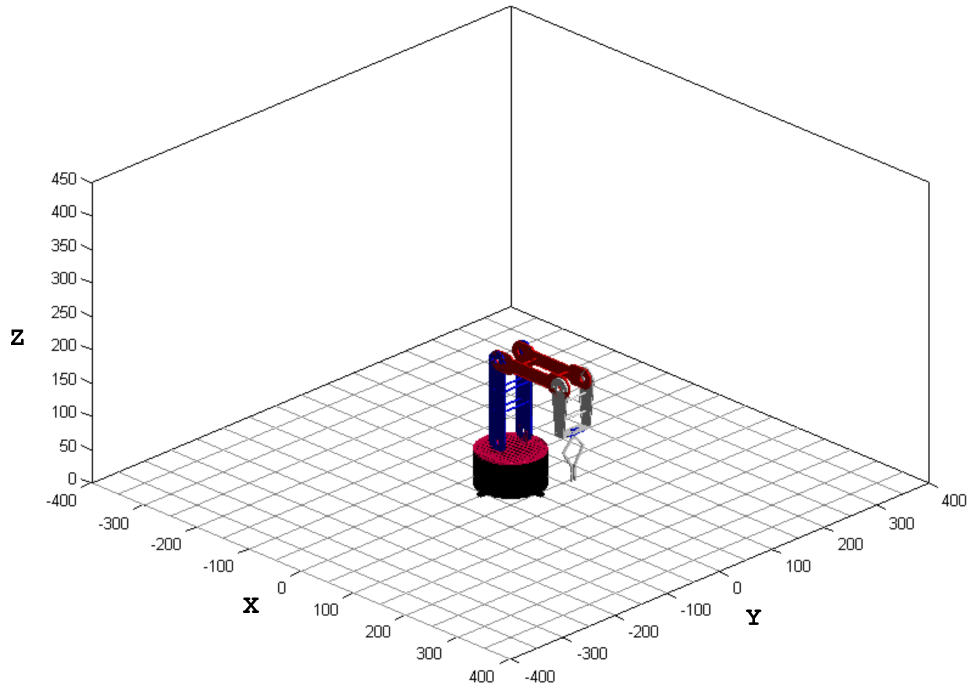


Figure 6.3: Lynx6 home position

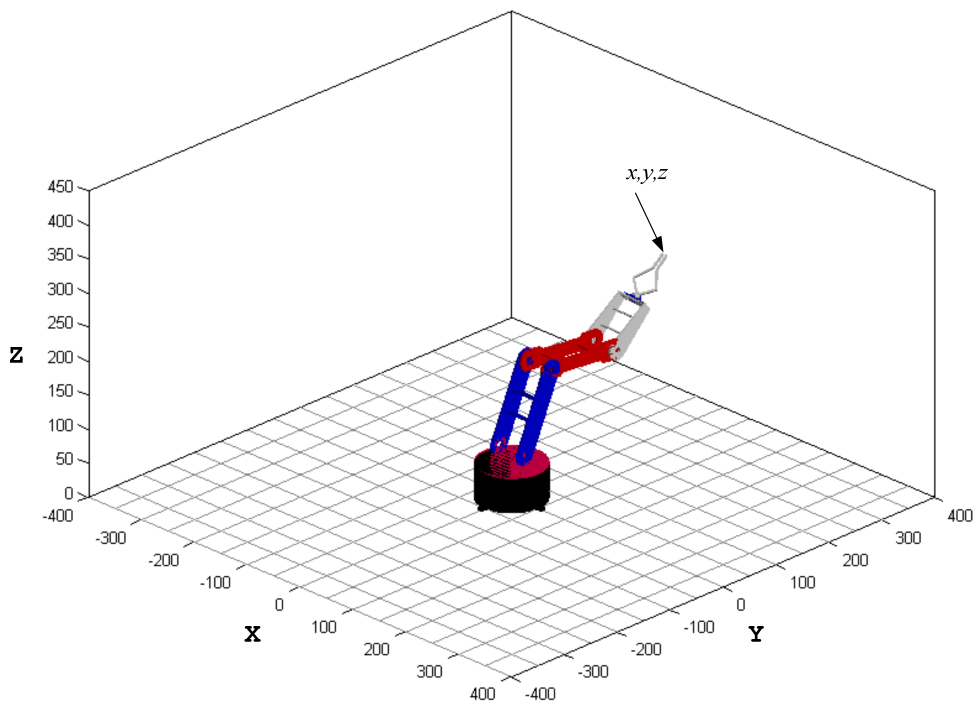


Figure 6.4: Lynx6 desired position

In order to assess the efficacy of the proposed controller, simulation studies have been conducted to check the efficiency of the system. PID controller is tested as the first attempt to control the Lynx6 robot arm.

Figure 6.5, Figure 6.6 and Figure 6.7 show the output response of the motors of the 5DOF Lynx6 using PID controllers.

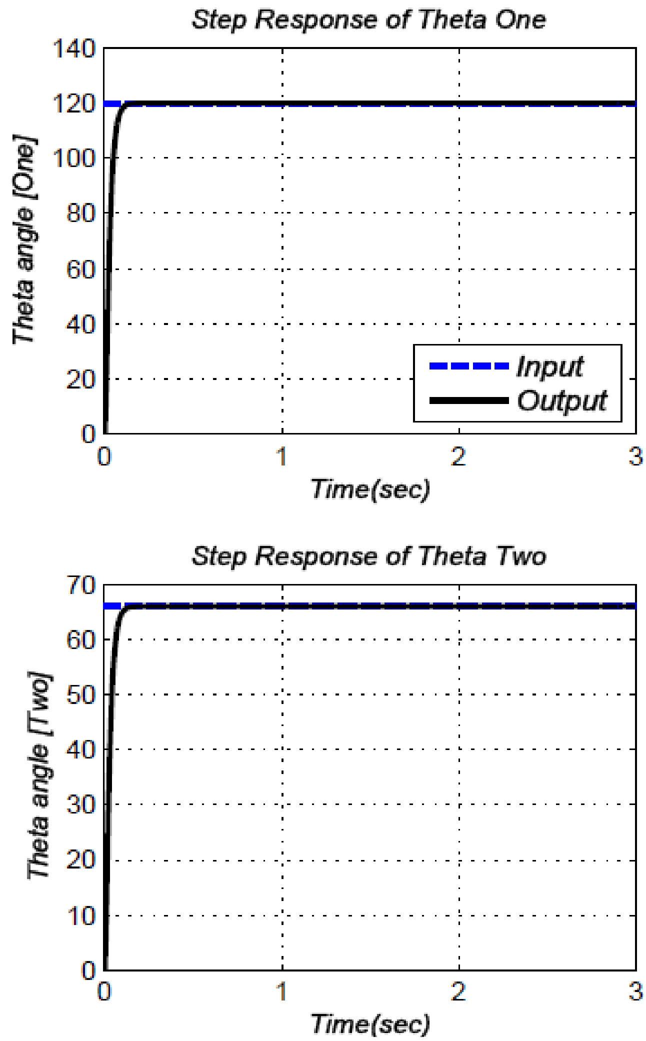


Figure 6.5: PID control step response for θ_1, θ_2

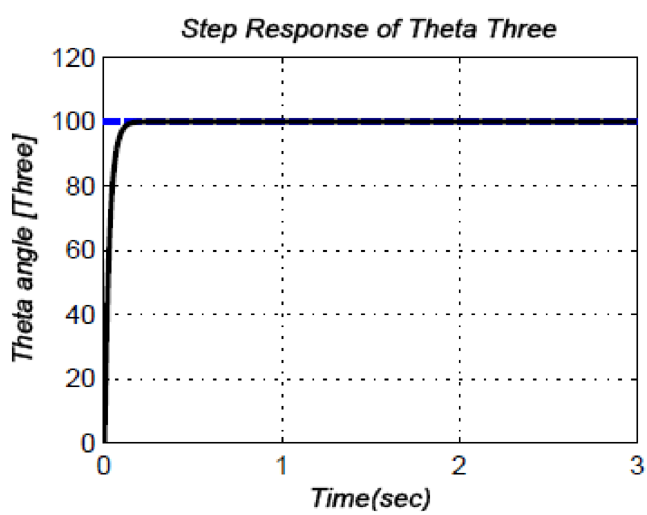


Figure 6.6: PID control step response for θ_3

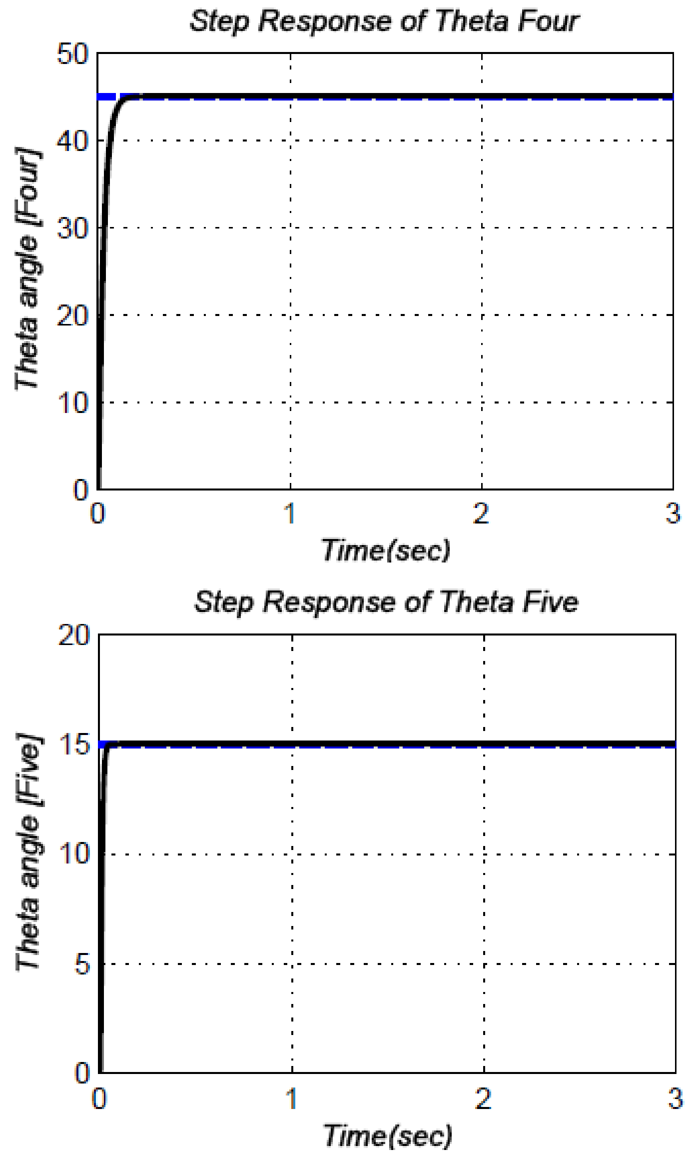


Figure 6.7: PID control step response for θ_4 and θ_5

Table (6.3) tabulated the performance of the five motors using PID controller.

Table 6.3: Performance of the PID controller

Motor number	**System characteristics**		
	Overshoot($O.S$)	Rising Time(t_r) sec	Steady state error ($S.S.E$)
Motor one	0.03	0.246	0.03
Motor two	0.008	0.247	0.001
Motor three	0.06	0.274	0.005
Motor four	0.02	0.277	0.01
Motor five	0.01	0.21	0.002

Effect of disturbance is studied by performing simulation of the control system in the presence of the disturbance. The disturbance is considered as the load torque that applied to the motor for each joint. The type of disturbance used in the simulation is step input disturbance. Figure 6.8 and Figure 6.9 shows the effect of disturbance on the output response of θ_1, θ_2 and θ_5. In the presence of disturbance after one and a half second, the output of the angular position will deviate. In this case, the feedforward method, which discussed in Chapter 3, will be applied.

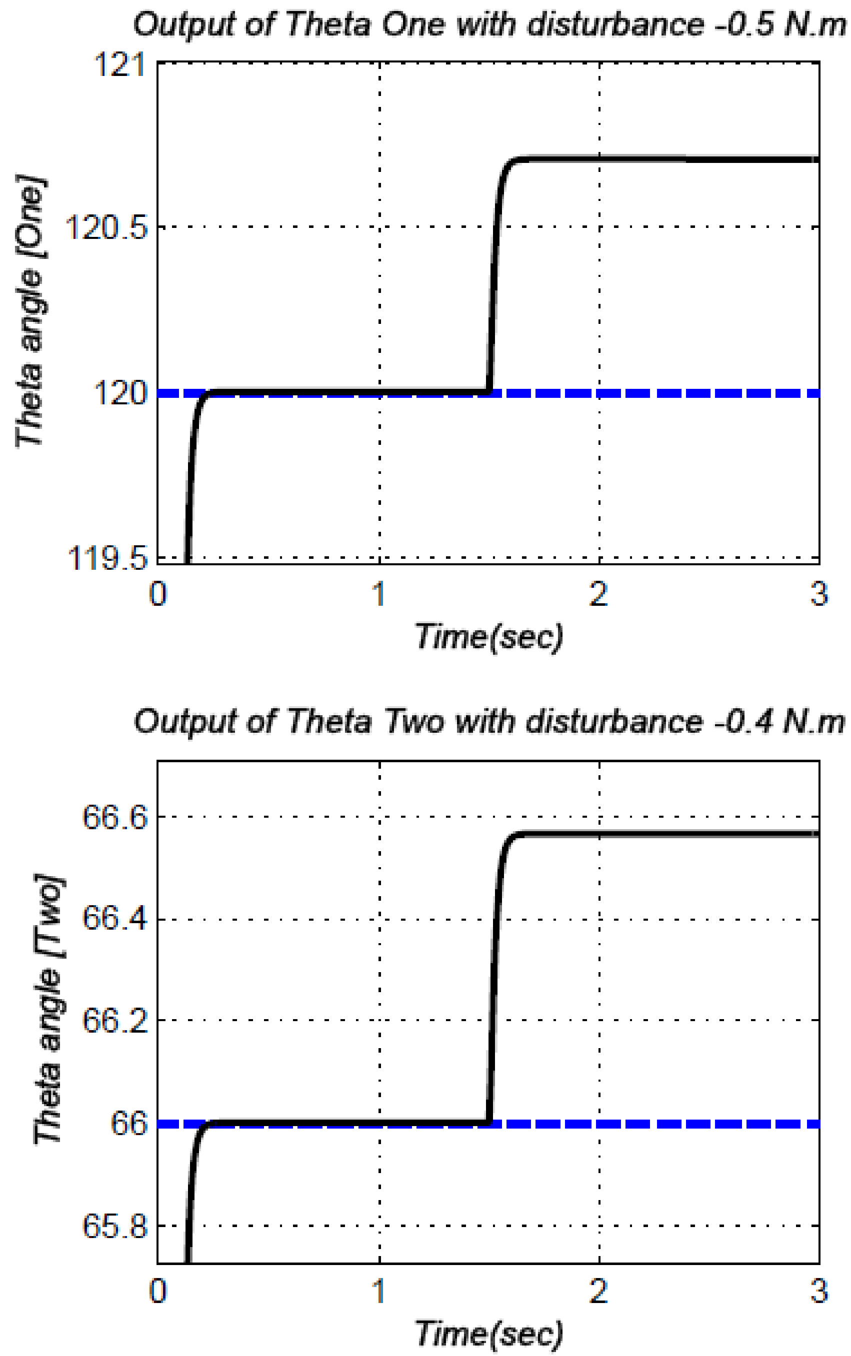


Figure 6.8: Output with disturbance for θ_1, θ_2

For θ_1 the output response shows that the output deviates from 120 degrees to 120.7 degree, similarly θ_2 deviate from 66 to 66.5 degree. In this case, the feedforward method will be applied to reject and eliminate these deviations.

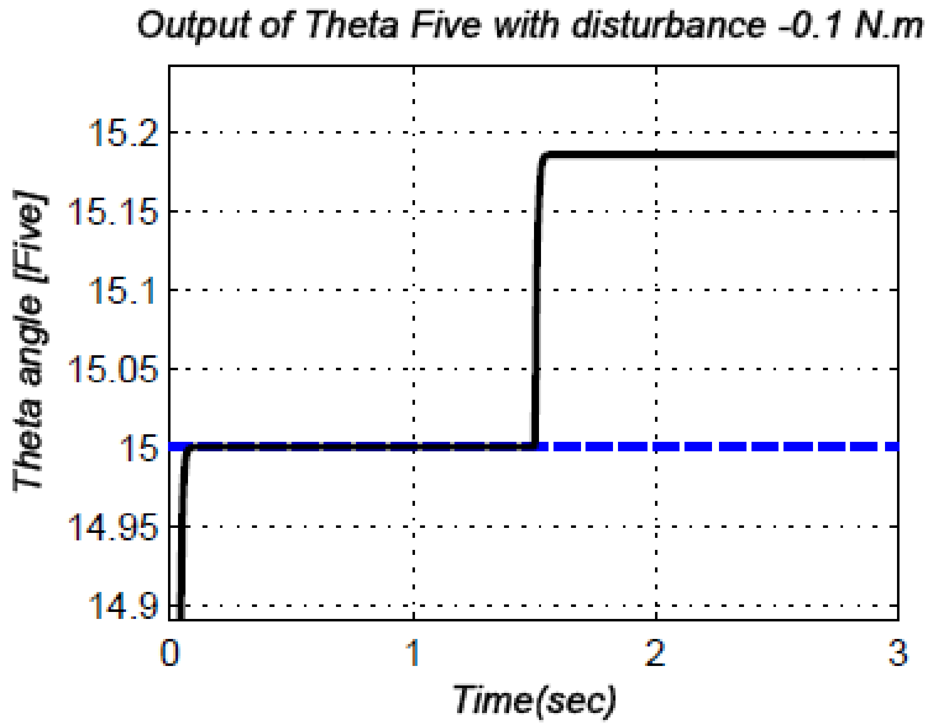


Figure 6.9: Output with disturbance for θ_5

Figure 6.10 and Figure 6.11 show the output response when the feedforward method is applied for θ_1, θ_2 and θ_5.

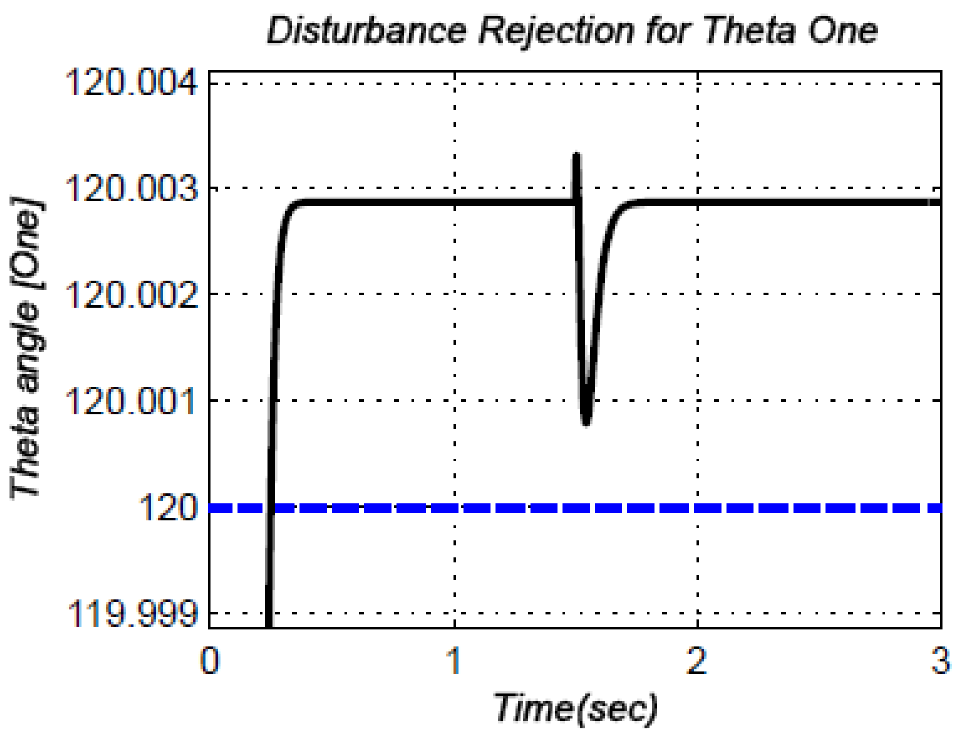
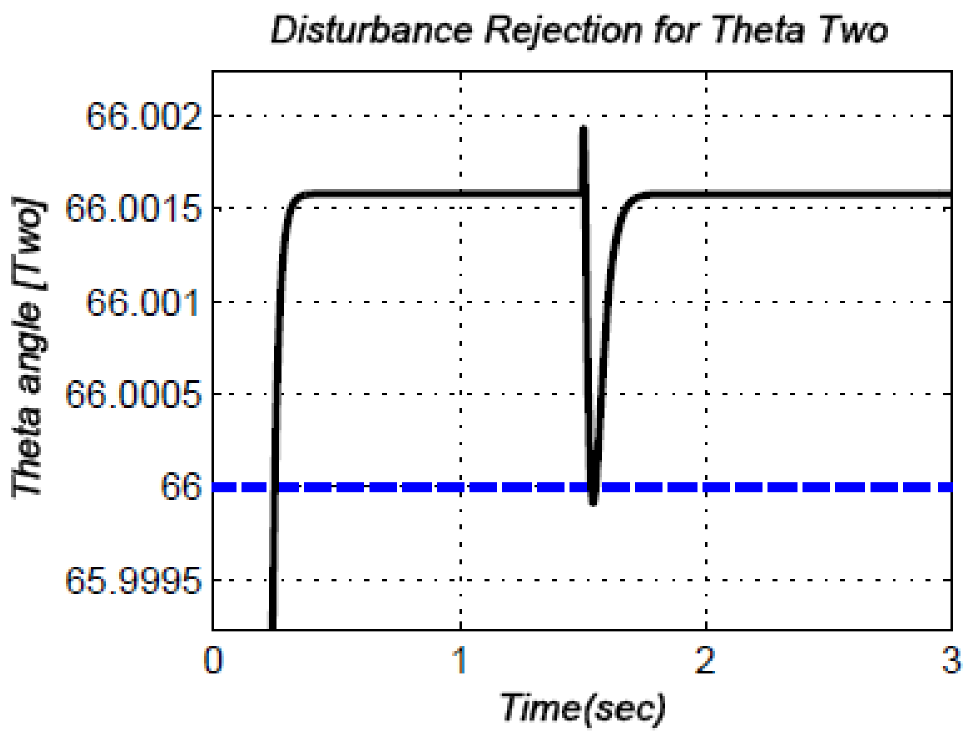


Figure 6.10: Output with disturbance rejection for θ_1



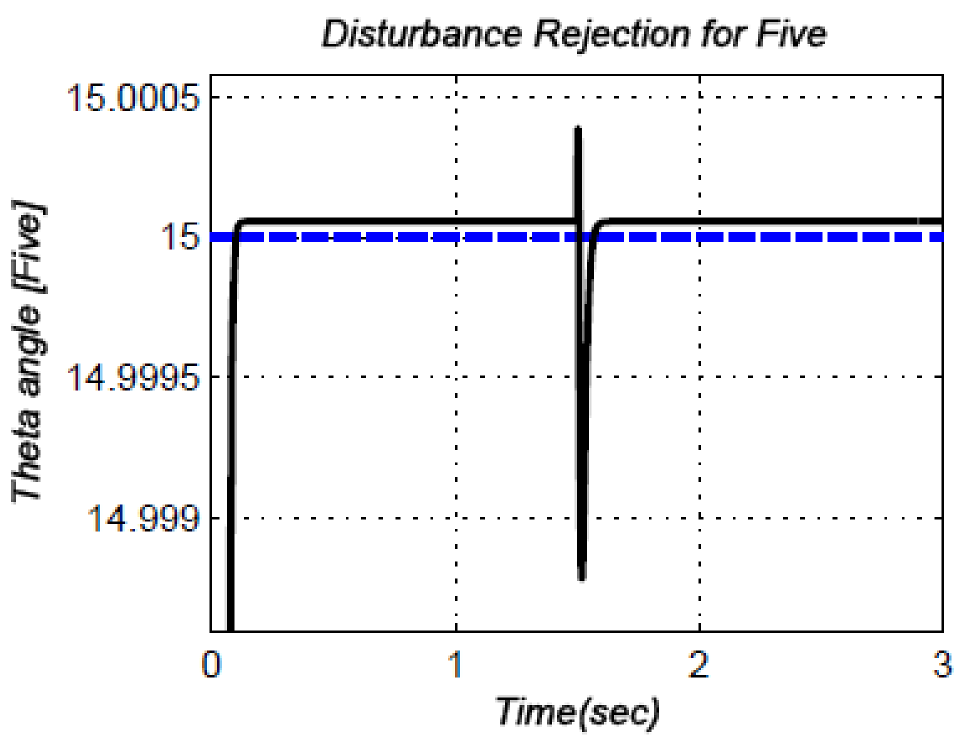


Figure 6.11: Output with disturbance rejection for θ_2 and θ_5

After the feedforward method is applied for theta one and theta two respectively the output response is reduced from 0.65 degree to 0.0045 degree in the case of theta one, and from 0.7 degree to 0.0032 in case of theta two respectively. This illustration proves that the feedforward method is very efficient for disturbance rejection.

When applying the step input to the fuzzy logic controller, the output response of the five motors is shown in Figure 6.12, Figure 6.13, Figure 6.14 and Figure 6.15 respectively.

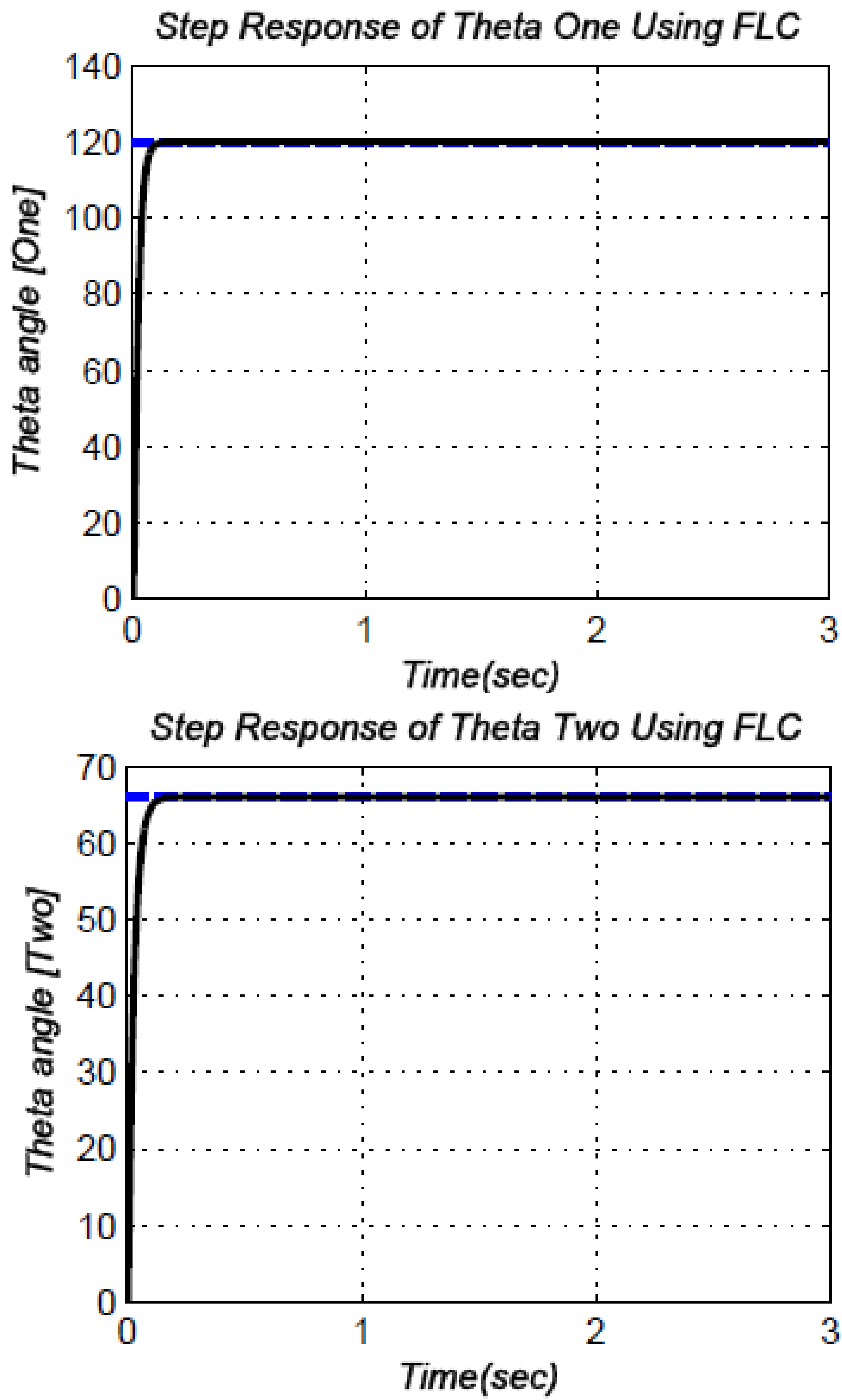


Figure 6.12: FLC step response for θ_1 and θ_2

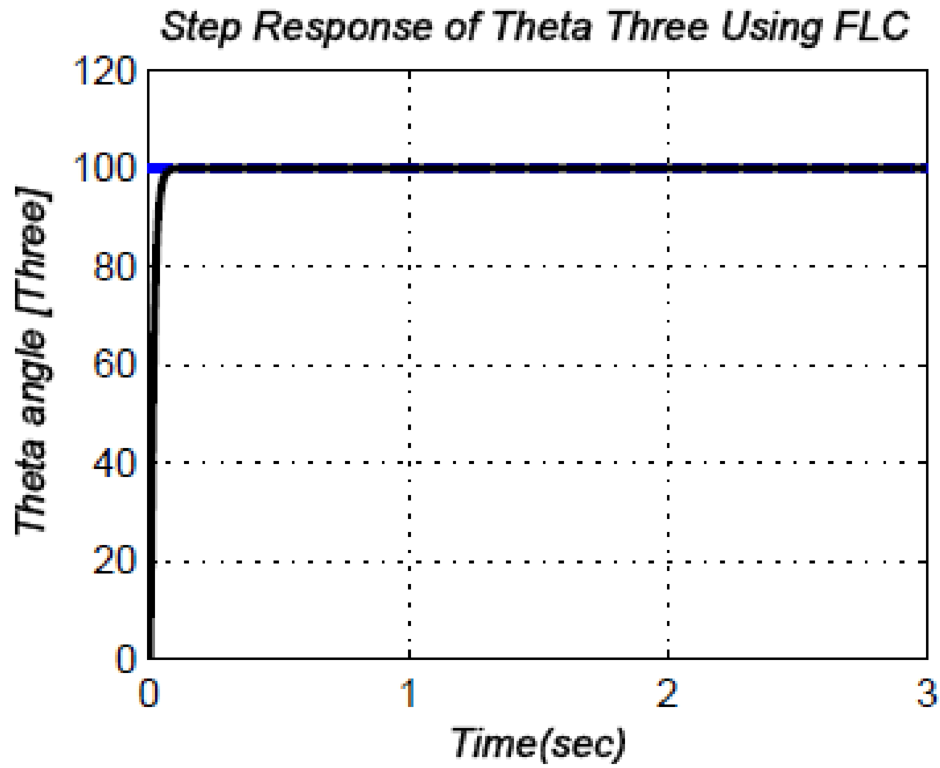


Figure 6.13: FLC step response for θ_3

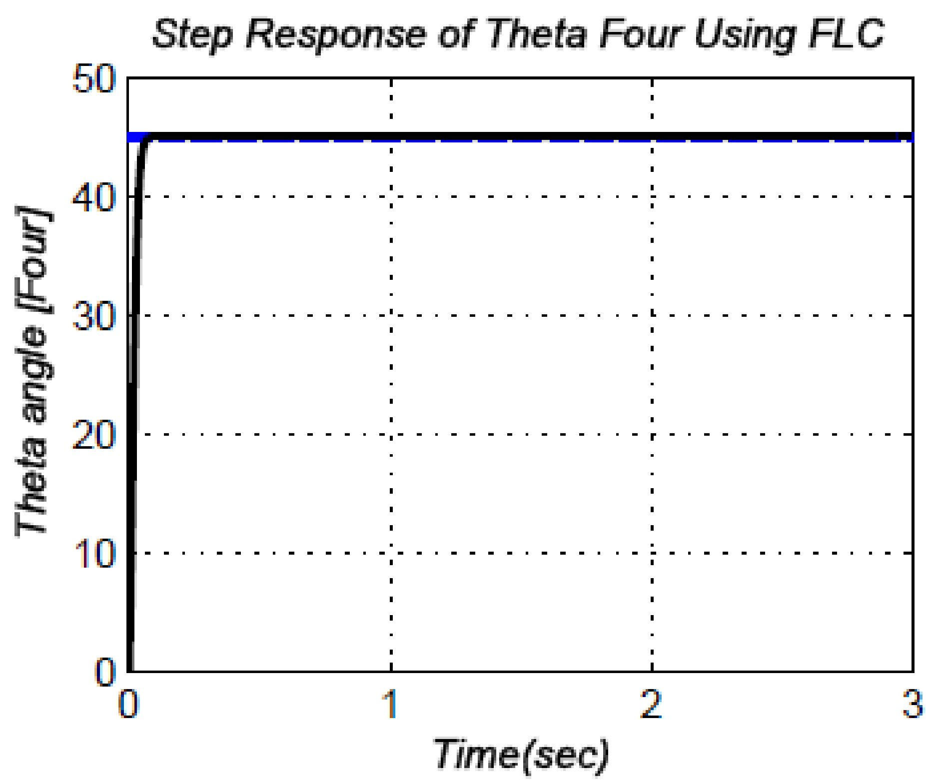


Figure 6.14: FLC step response for θ_4

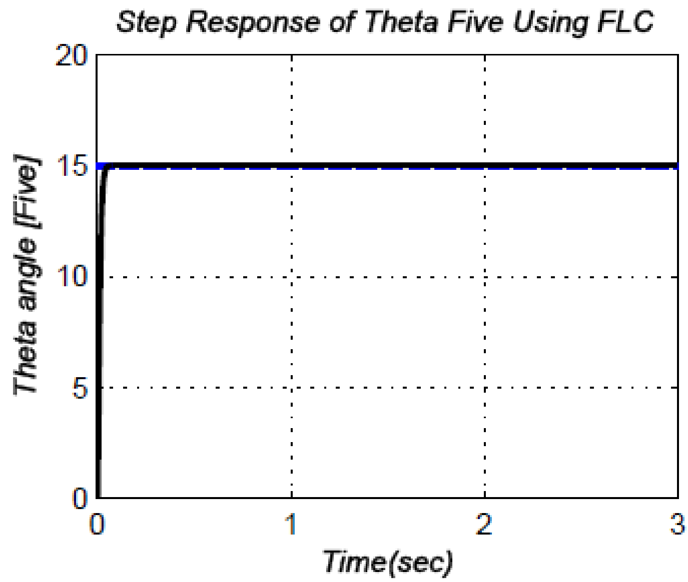


Figure 6.15: FLC step response for θ_5

It is clear from the above figures that the response obtained using the fuzzy logic controller is better than the response of PID controller. Table (6.4) shows the performance results for the motors of the robot manipulator using FLC.

Table 6.4: Performance of the FLC

Motor number	System characteristics		
	Overshoot($O.S$)	Rising Time(t_r) sec	Steady state error ($S.S.E$)
Motor one	0.016	0.142	0.006
Motor two	0.002	0.225	0.0008
Motor three	0.0015	0.109	0.001
Motor four	0.01	0.09	0.001
Motor five	0.001	0.066	0.002

Simulations and numerical results, which compare between PID controller and fuzzy logic controller, prove that the performance of fuzzy logic controller is better than the PID performance for controlling robot manipulator in terms of reducing overshoot size, enhancing rising time and minimizing steady state error. As example, the steady state error of the motor one for the FLC is 0.006, while it is 0.03 for the PID controller. In other words, this means that the FLC steady state error is 80% less than the PID controller. The rising time for the FLC is 45% less than PID controller. Finally, the overshoot of FLC is 46% less than the PID controller.

The fuzzy supervisory PID controller is applied to 5DOF robot arm [65]. The robot has 5DOF each of them has a motor with specific transfer function. As a case study, we will present the output response of the fifth DOF of the robot arm, and we will show the variation of the PID gains through process control. The output response of the other motors can be obtained in the same way.

The transfer function of the motor of the fifth DOF considered is:

$$G(s) = \frac{1.5}{8.75*10^{-7}s^3 + 0.002s^2 + 0.006817s} \tag{6.1}$$

The results show the output response of the theta five of robot arm using the proposed controllers. Simulation result in Figure 6.16 and Figure 6.17 show the output response of the proposed controllers using the step as input signal. The two figures below show the performance of the PID using the classical tuning (without fuzzy tuning) and using the supervisory tuning respectively. In addition, they show the effectiveness of the two controllers for rejection disturbance.

If the load torque with -0.5 *N.m* is applied on the desired angle. The obtained result shows the effect of the disturbance on the output response after one second and the efficacy of the FSC controller for tuning PID parameters and eliminating the disturbance.

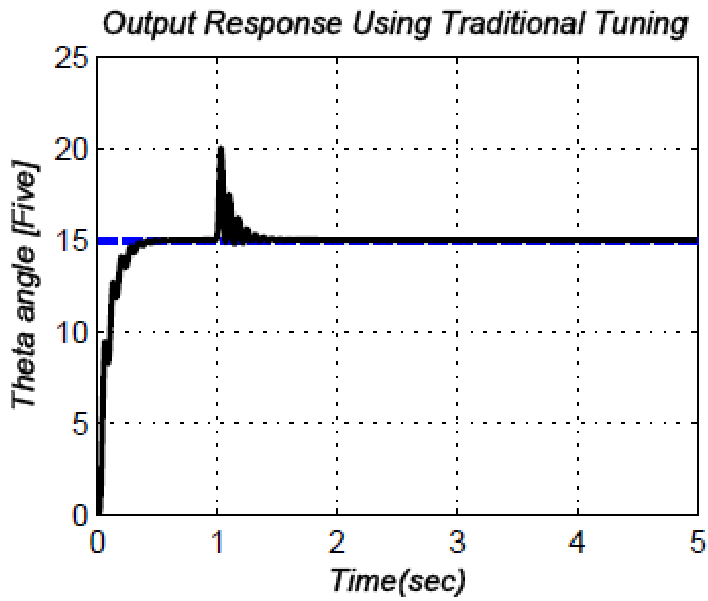


Figure 6.16: Output response using classical tuning methods

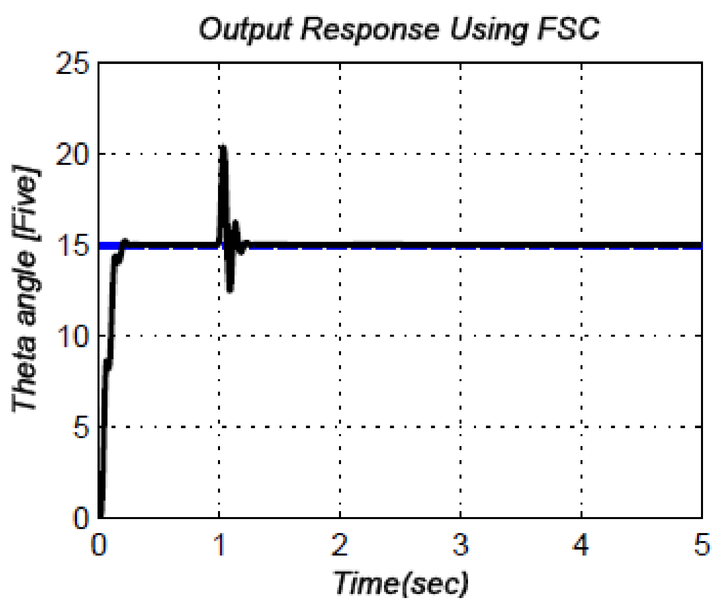


Figure 6.17: Output response using fuzzy supervisory control

Clearly, the supervisory fuzzy control achieved better performance than classical tuning methods in terms of time response. The above figures show the effect of small disturbance after one second and effectiveness of the fuzzy supervisory control in eliminating the presence disturbances.

Fuzzy supervisory controller attempts to vary the PID parameters during process operation to enhance the system response and eliminating the disturbances. Figure 6.18 shows the variation of the PID gains during the operation using fuzzy control as a supervisory controller.

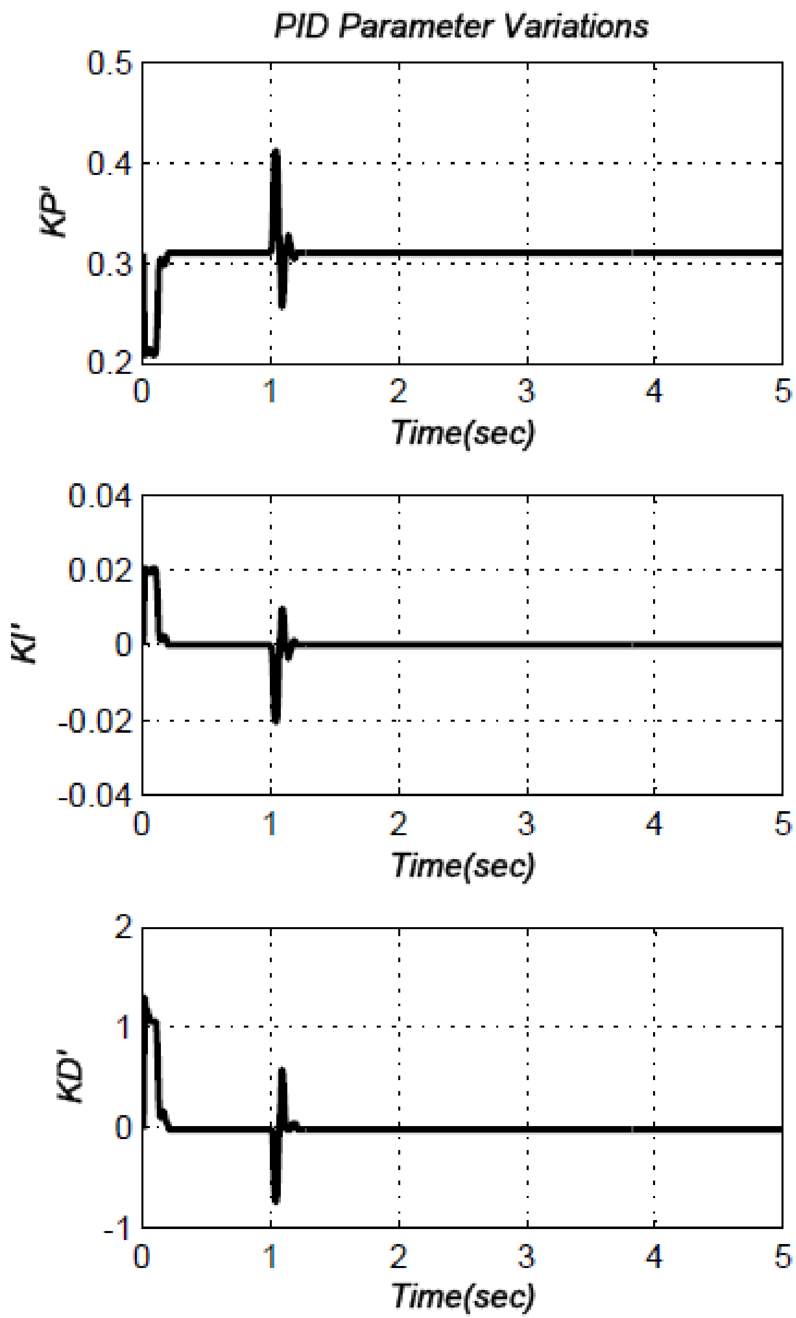


Figure 6.18: PID parameters variations

Performance of the proposed controllers is summarized in Table (6.5).

Table 6.5: Output with and without fuzzy tuning

Tuning method	System characteristics		
	Overshoot($O.S$)	Rising Time(t_r) sec	Steady state error ($S.S.E$)
Classical tuning method	0.08	0.3	0.03
Tuning using FLC	0.001	0.15	0.001

6.3. Summary

This chapter was presented the implementations of the proposed controller to control the Lynx6 robot arm with 5DOF. Section (6.2) described in briefly the Lynx6 robot arm and its frame assignment. Next, the computations, simulation and results for the proposed controllers were presented.

In this thesis, some specifications are used to control the lynx6 robot arm such as overshoot less than 5%, rising time less than 2 second and steady state error approach 0.002. The obtained results achieved the desired specifications in terms of these specifications as expected. In this thesis, the obtained results using PID control with feedforward compensator are good.

In case of replacing PID with FLC, taking into account the previous conditions, the obtained results are satisfactory and are better in comparison to PID control results. For the comparison between the tuning methods, the FSC is tends to enhance the tuning method for PID parameters. The obtained results show that the FSC is better than classical methods in tuning PID parameters.

CHAPTER 7 CONCLUSION AND FUTURE WORK

7.1. Conclusion

Robotics has become recently an interesting area of research. In this thesis, we study the robot manipulator from two sides: modeling and control. Modeling process includes kinematic analysis and DC motor modeling. This process is important before controlling the robot to save the robot from being damaged. Appling a control technique is important to guarantee high efficiency and lower error for the motion of the robot.

The desired tasks were accomplished using three stages: the first stage was to provide systematic rules for analyzing forward and inverse kinematics solutions for the robot manipulator with revolute or prismatic joints using DH parameters, then analyzing the mathematical model of the DC motor in both frequency and time domains. In the second stage, we discussed the problem of control techniques. PID controller was applied, to control the robot manipulator, then FLC was implemented and considered as a second choice to control the robot. The third controller was a hybrid one between the previous controllers, denoted as FSC. In the third stage, we compared the results of using the three controllers for controlling the robot manipulator. First, we compared the results of the PID and the FLC techniques in terms of overshoot, transient response and steady state error. Second, we compared the results of PID classical tuning methods and fuzzy supervisory tuning method. All simulations were presented using MATLAB and SIMULINK, which are used widely in control applications.

The objective of this thesis was to control Lynx6 robot arm to reach the specified location with minimum error while meeting certain specification. The tracking path from the initial position to the final position was not considered in this thesis; we set the final position for each motor used independent joint control method. This thesis used the PID controller to compare its results with FLC and FSC. Feedforward method was used to overcome the disturbances, which loaded on each motor. The system was model as 5DOF, which means it has five motors to control their positions independently. In this thesis, we applied Mamdani method in FLC. This was applied using 30 rules to control robot arm. Tuning PID parameters was accomplished using FSC using 49 rules in order to fine tune KP, KI, and KD of the PID controller. By doing so, we overcame the tuning limitation of PID parameters using classical tuning methods. Both controllers used the center of area defuzzification method, and min max inference mechanism.

The simulations and numerical results of the previous controllers were presented in this thesis. We proved that the FLC is more efficient in the time response behavior than the PID controller. The average steady state error for the five motors for the FLC was 0.0105 while it was 0.042 for PID controller, in other words, the FLC steady state error is 0.75% less than the PID steady state error. We also showed that the rising time for the FLS was 49.6% less than the PID controller. The overshoot size for FLC was 73% less than the PID controller. On the other side, we proved that the FSC is more efficient to control the robot arm to reach the desired output compared to classical tuning methods. Whereas the steady state error of motor five was minimized from 0.03 to 0.001, overshoot size reduced from 0.08 to 0.001and the rising time for the FSC was 50% less than PID controller. These numerical results showed that the FSC gives satisfactory results in tuning PID parameters compared to traditional methods. To summarize, the obtained results achieved the desired performance in terms of reducing time response as expected.

7.2. Future work

- A good research problem in mind is to study different fuzzy approaches. Mamdani approach may be replaced by Takagi-Sugeno model and compared with Mamdani.
- This thesis recommends applying other type of inputs like cubic trajectory.
- In future, we will search for methods to extract the optimum rule base to control DC motor automatically. We also apply different method of inference mechanism and defuzzification methods.
- In the proposed design, we completely used off-line simulation. It can be extended to an on-line controlling for any serial robot manipulator.
- The work done in this thesis was based on having PID control structure. Since the system is nonlinear, it would be interesting to apply nonlinear controllers such sliding mode control algorithms.

REFERENCES

[1] M.W. Spong, S. Hutchinson and M. Vidyasagar, *Robot Modeling and Control*, 1st Edition, Jon Wiley & Sons, Inc, 2005.

[2] J. Angeles, *Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms*, 2nd Edition, Springer, 2003.

[3] J.J. Crage, *Introduction to Robotics Mechanics and Control*, 3rd Edition, Prentice Hall, 2005.

[4] K. H. Kalil, *Nonlinear Systems*, 3rd Edition, Prentice-Hall, 2002.

[5] R. Gorez, "A Survey of PID Auto-Tuning Methods," Journal A, Vol. 38, No. 1, pp. 3–10, 1997.

[6] K.J. Astrom and T. Hagglund, *PID Controllers: Theory, Design and Tuning*, 2nd Edition, Research Triangle Park, NC: Instrument Society of America, 1995.

[7] J.G. Ziegler and N.B. Nichols, "Optimum Settings for Automatic Controllers," *Transaction American Society of Mechanical Engineering.*, Vol. 64, pp. 759–768, 1942.

[8] L.A. Zadeh, "Fuzzy Sets," *Information and Control.*, Vol. 8, pp. 335–353, 1965.

[9] E.H. Mamdani, "Applications of fuzzy logic to approximate reasoning using linguistic synthesis," *IEEE Transactions on Computers*, Vol. 26, No. 12, pp. 1182–1191, Dec. 1977.

[10] G.K.I. Mann, B.-G. Hu, and R.G. Gosine, "Analysis of Direct Action Fuzzy PID Controller Structures," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, Vol. 29, No. 3, pp. 371–388, Jun. 1999.

[11] Z.-Y. Zhao, M. Tomizuka and S. Isaka, "Fuzzy Gain Scheduling PID Controller," *IEEE Transactions on Systems, Man., Cybernetics Society*, Vol. 23, No. 5, pp. 1392–1398, Sep/Oct. 1993.

[12] A. Visioli, "Tuning of PID Controllers with Fuzzy Logic," *IEEE Proceedings Control Theory and Applications.*, Vol. 148, No. 1, pp. 1-8, 2001.

[13] X. Wang, X. Chen, W. Jia, Y. Sun and H. Pu, "Forward Kinematics Analysis and 3-Dimmision Gait Simulation of a MiniQuad Walking Robot," *IEEE International Conference on Mechatronics and Automation*, pp. 1932–1937, Aug. 2007.

[14] S. Annand, *Software for Control and Dynamic Simulation of Unimate Puma 560 Robot*, *MS Thesis, The Faculty of the College of Engineering and Technology, Ohio University*, June 1993.

[15] S. Elgazzar, "Efficient Kinematic Transformations for the Puma 560 Robot," *IEEE Journal of Robotics and Automation.*, Vol. 1, No. 3, pp. 142–151, 1985.

[16] A.C. Soh, E.A. Alwi, R.Z. Abdul Rahman and L.H. Fey, "Effect of Fuzzy Logic Controller Implementation on a Digitally Controlled Robot Movement," *Kathmandu University Journal of Science, Engineering and Technology.*, Vol. I, No. V, pp 28–39, Sep. 2008.

[17] D.P. Kwok, Z.Q. Sun and P. Wang, "Linguistic PID Controller for Robot Arms," *IEEE International Conference*, Vol. 1, pp. 341–346, Mar. 1991.

[18] 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, No. 3, pp.299–320, 2004.

[19] S.G. Anavatti, S.A. Salman and J.Y. Choi, "Fuzzy + PID Controller for Robot Manipulator," *International Conference on Computational Intelligence for Modelling Control and Automation.*, pp. 75, Dec. 2006.

References

[20] B.W. Bekit, L.D. Seneviratne, J.F. Whidborne and K. Althoefer, "Fuzzy PID Tuning for Robot Manipulators," *IEEE Industrial Electronics Society, in 24th Annual Conference.*, Vol. 4, pp. 2452–2457, Sep. 1998.

[21] B. Koyuncu and M. Güzel, "Software Development for the Kinematic Analysis of a Lynx 6 Robot Arm," *Proceedings of World Academy of Science, Engineering and Technology.*, Oct. 2007.

[22] O. Gürdal, M. Albayrak and T. Aydogan, "Computer Aided Control and Simulation of Robot Arm Moving in Three Dimension," *Gazi University and Suleiman Demirel University., Turkey*.

[23] A.Y. -Nones, "Heterogeneous Modeling & Design of Robot Arm Control System," *University of Puerto Rico*, 2004.

[24] A. Delibasi, T. Türker and G. Cansever, "Real Time Dc Motor Position Control by Fuzzy Logic and PID Controllers Using Labview,"

[25] R.E. Samin, M.S. Ramli and M.A. Ahmad, "Performance Comparison of PD-type Fuzzy Logic Controller of USBM Simplified Model," *Proceedings of The 2nd International Power Engineering and Optimization Conference.*, pp. 217–221, 2008.

[26] W. Li, "Design of a Hybrid Fuzzy Logic Proportional plus Conventional Integral-Derivative Controller", *IEEE Transaction on Fuzzy Systems.*, Vo1. 6, No. 4, pp.449–463, Nov 1998.

[27] Y.C. Yung and C.Y. Chen, "PD type vs. PID type Fuzzy Controller," *IEEE 10th Region Conference Melbourne, Australia.*, Vol.1, pp. 341-345, Nov. 1992.

[28] S. Chopra, R. Mitra and V. Kumar, "Fuzzy Controller: Choosing an Appropriate and Smallest Rule Set," *International Journal of Computational Cognition*, Vol. 3, No. 4, Dec.2005.

[29] N. Khong, A. Kanch, S. Nopna, S. Tanut, S. Tuny and R. Kagawa, "Control of the Position Dc Servo Motor by Fuzzy Logic," *IEEE Proceedings of TENCON.*, Vol. 3, pp.354–357, Sep.2000.

[30] A.G. Sreenatha and P. Makarand, "Fuzzy Logic Controller for Position Control of Flexible Structures," *Acta Astronaut journal.*, Vol. 50, No. 11, pp. 665–671, 2002.

[31] J.S. Surdhar and A.S. White, "A Parallel Fuzzy-Controlled Flexible Manipulator Using Optical Tip Feedback," *Journal of Robotics and Computer Integrated Manufacturing,* Vol. 19, No. 3, pp. 273–282, 2002.

[32] M. Nil, U. Yuzgec, M. Sonmez and B. Cakir, "Fuzzy Neural Network Based Intelligent Controller For 3-DOF Robot Manipulators," *the 5th International Symposium on Intelligent Manufacturing Systems.*, pp. 884–895, May.2006.

[33] R.P. Copeland and K.S. Rattan, "A Fuzzy Logic Supervisor for PID Control of Unknown Systems," *IEEE International Symposium on Intelligent Control.*, pp. 22–26, Aug.1994.

[34] M. Dotoli, B. Maione and B. Turchiano, "Fuzzy-Supervised PID Control: Experimental Results," *EUNITE the 1st European Symposium on Intelligent Technologies, Hybrid Systems and their Implementation on Smart Adaptive Systems*, pp. 31–35, 2001

[35] L. Wang, M. Tian and Y. Gao, "Fuzzy Self-adapting PID Control of PMSM Servo System," *IEEE International Electric Machines & Drives Conference.*, Vol. 1, pp. 860–863, May.2007.

[36] K.K. Ahn and N.B. Kha, "Position Control of Shape Memory Alloy Actuators Using Self Tuning Fuzzy PID Controller," *IEEE Conference on Industrial Electronics and Applications*., pp. 1–5, May.2006.

[37] S.G. Tzafestas and N.P. Papanikolopoulos, "Incremental fuzzy expert PID Control," *IEEE Transactions on Industrial Electronics.*, Vol. 37, No. 5, pp. 365–371, Oct.1990.
[38] K.H. Ang, G.C.Y. Chong, and Y. Li, "PID Control System Analysis, Design, and Technology," *IEEE Transactions on Control Systems Technology.*, Vol. 13, No. 4, pp. 559–576, July.2005.
[39] C.K. Benjamin, *Automatic Control System*, 7th Edition, Englewood Cliffs, Prentice Hall, pp. 175–180, 1995.
[40] Z.-Y. Zhao, M. Tomizuka and S. Isaka, "Fuzzy Gain Scheduling of PID Controllers," *IEEE Conference on Control Applications.*, Vol. 2, pp. 698–703, Sep.1992.
[41] C.C. Hang, K.J. Astrom, and W.K. Ho, "Refinements of the Ziegler–Nichols Tuning Formula," *IEE Proceedings D Control Theory and Applications.*, Vol. 138, No. 2, pp. 111–118, Mar. 1991.
[42] T.C. Chi, *Analog and Digital Control System Design: Transfer-Function, State-Space, and Algebraic Methods*, Saunders College, pp. 551–563, 1993.
[43] N.S. Nise, *Control System Engineering*, 4th Edition, Wiley, 2004.
[44] S.W. Zhao and N.C. Cheung, "Disturbances Rejection for Precise Position Control of Linear Switched Reluctance Motors," *Proceedings of the 17th World Congress the International Federation of Automatic Control*, July. 2008.
[45] G. Feng, "A Survey on Analysis and Design of Model-Based Fuzzy Control Systems," *IEEE Transactions on Fuzzy Systems.*, Vol. 14, No. 5, pp. 676–697, Oct. 2006.
[46] C.C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part I", *IEEE Transactions on Systems, Man and Cybernetics.*, Vol. 20, No.2, pp. 404–418, March/April 1990.
[47] M. Sugeno, "An Introductory Survey of Fuzzy Control," *Information Sciences.*, 36, pp. 59–83, 1995.
[48] V. Kumar and A.P. Mittal, "Parallel Fuzzy P + Fuzzy I + Fuzzy D controller: Design and Performance Evaluation," *International Journal of Automation and Computing.*, 2009.
[49] C.C. Lee, "Fuzzy logic in control systems: Fuzzy Logic Controller-part II," *IEEE Transactions on Systems, Man and Cybernetics.*, Vol. 20, No.2, pp. 419–435, March/April 1990.
[50] T. Terano, K. Asia, and M. Sugeno, *Fuzzy Systems Theory and its Applications*, Academic Press, pp. 34–59, 1992.
[51] J. Jantzen, Foundations *of Fuzzy Control*, John Wiley & Sons, pp. 17-19, 2007.
[52] E.H. Mamdani, H.J. Efstathiou and K. Sugiyama., "Development of Fuzzy Logic Control," The 23rd *IEEE Conference on Decision and Control.*, Vol. 23, pp. 888–893, 1984.
[53] V.G. Moudgal, S. Yurkovich, and K.M. Passino, "Rule-Based Control for A Flexible-Link Robot," *IEEE Transactions on Control Systems Technology.*, Vol .2, No. 4, pp. 392–405, Dec.1994.
[54] H. Ying, "The Takagi-Sugeno Fuzzy Controllers Using The Simplified Linear Rules Are Nonlinear Variable Gain Controllers", Automatica, Vol. 34, No. 2, pp. 157–167, 1998.
[55] O.A.S. Youssef, "Applications of Fuzzy Inference Mechanisms to Power System Relaying," *IEEE Power Systems Conference and Exposition.*, Vol. 1, pp. 560–567, Oct. 2004.

References

[56] S.N. Sivanandam, S. Sumathi and S.N. Deepa, *Introduction to Fuzzy Logic Using MATLAB*, Springer, pp. 97–107, 2007.

[57] G.K.I. Mann, B.G. Hu and R.G. Gosine, “A systematic study of fuzzy PID controllers - function-based evaluation approach,” *IEEE Transactions on Fuzzy System*., Vol. 9, No. 5, pp. 699–712, Oct. 2001.

[58] R. Ketata, D. De Geest, and A. Titli, “Fuzzy controllers: Design, evaluation, parallel, and hierarchical combination with a PID controllers,” *Fuzzy Sets System*., Vol. 71, No. 1, pp. 113–129, Apr. 1995.

[59] A.Z. Alassar, I.M. Abuhadrous and H.A. Elaydi “Control of 5DOF Robot Arm Using PID Controller with Feedforward Compensation”, *Accepted in the 2nd Conference on Computer and Automation Engineering*, Singapore, 5. Dec. 2009.

[60] K.-N. Luo, C.-Y. Kuo, and L.-T. Sheu, “A Novel Method for Fuzzy Self-Tuning PID Controllers,” *IEEE Proceedings of the Asian Fuzzy Systems Symposium*, pp. 194–199, Dec. 1996.

[61] L. Yao and C.-C. Lin, “Design of Gain Scheduled Fuzzy PID Controller,” *International Conference on Computational Intelligence.*, pp. 432-436, Nov. 2004.

[62] Y. Yang, W. Wang, D.-J. Yu and G. Ding, “A Fuzzy Parameters Adaptive PID Controller Design of Digital Positional Servo System,” *IEEE Proceeding of the First International Conference on Machine Learning and Cybernetics*., Vol. 1, pp. 310–314, Nov. 2002.

[63] A.Z. Alassar, I.M. Abuhadrous and H.A. Elaydi “Comparison between FLC and PID Controller for 5DOF Robot Arm”, *The 2nd IEEE International Conference on Advanced Computer Control*, Shenyang, Dec. 2009.

[64] Y.-Y. Chen, “Rules Extraction for Fuzzy Control System,” *IEEE international Conference on System Man and Cybernetics.*, Vol. 2, pp. 526–527, Nov 1989.

[65] A.Z. Alassar, I.M. Abuhadrous and H.A. Elaydi “Modeling and Control of 5DOF Robot Arm Using Supervisory Control”, *The 2nd International Conference on Computer and Automation Engineering*, Singapore, Jan. 2010.

PUBLICATIONS

[1] A.Z. Alassar, I.M. Abuhadrous and H.A. Elaydi **"Control of 5DOF Robot Arm Using PID Controller with Feedforward Compensation"**, *Accepted in the 2nd Conference on Computer and Automation Engineering*, Singapore, 5. Dec. 2009.

[2] A.Z. Alassar, I.M. Abuhadrous and H.A. Elaydi **"Comparison between FLC and PID Controller for 5DOF Robot Arm"**, *The 2nd IEEE International Conference on Advanced Computer Control*, Shenyang, Dec. 2009.

[3] A.Z. Alassar, I.M. Abuhadrous and H.A. Elaydi **"Modeling and Control of 5DOF Robot Arm Using Supervisory Control"**, *The 2nd Conference on Computer and Automation Engineering*, Singapore, Jan. 2010.

APPENDIX A: FORWARD AND INVERSE KINEMATICS ANALYSIS

A.1 Solving *A* Matrices Using MATHEMATICA Program

1. Open a text editor and create a text file with the following contents

```
DOF = 5
The Denavit-Hartenberg table:
joint1 = revolute
a1     = 0
alpha1 = 90*Degree
d1     = d1
theta1 = q1
joint2 = revolute
a2     = a2
alpha2 = 0*Degree
d2     = 0
theta2 = q1
joint3 = revolute
a3     = a3
alpha3 = 0
d3     = 0
theta3 = q3
joint4 = revolute
a4     = a4
alpha4 = -90*Degree
d4     = 0
theta4 = q4
joint5 = revolute
a5     = 0
alpha5 = 0*Degree
d5     = d5
theta5 = q5
```

2. Save the file as Robot.txt then after loading Mathematica, enter the following commands.

```
In[1]= << robotica.m
In[2]= DataFile ["C:\\Robot.txt"]
```

The DH parameter table display in the form:

Joint	Type	a	alpha	d	theta
1	revolute	0	90 * Degree	d1	q1
2	revolute	a2	0	0	q1
3	revolute	a3	0	0	q3
4	revolute	a4	-90 * Degree	0	q4
5	revolute	0	0	d5	q5

Figure A.1: Obtain DH parameters using MATHEMATICA

3. Solve the forward kinematic using the following command

```
In[3]= FKin[]
```

4. MATHEMATICA program generates each A and H matrices. To print the matrices from A1 to A5 write the following commands:

```
In[4]= MPrint[A[1],"A1="]
       MPrint[A[2],"A2="]
       MPrint[A[3],"A3="]
       MPrint[A[4],"A4="]
       MPrint[A[5],"A5="]
```

5. The output is:

```
     | Cos[q1]   0    Sin[q1]    0          |
     | Sin[q1]   0    -Cos[q1]   0          |
A1=  | 0         1    0          d1         |
     | 0         0    0          1          |

     | Cos[q1]   -Sin[q1]   0   a2 Cos[q1]   |
     | Sin[q1]   Cos[q1]    0   a2 Sin[q1]   |
A2=  | 0         0          1   0            |
     | 0         0          0   1            |

     | Cos[q3]   -Sin[q3]   0   a3 Cos[q3]   |
     | Sin[q3]   Cos[q3]    0   a3 Sin[q3]   |
A3=  | 0         0          1   0            |
     | 0         0          0   1            |

     | Cos[q4]   0     -Sin[q4]   a4 Cos[q4]   |
     | Sin[q4]   0     Cos[q4]    a4 Sin[q4]   |
A4=  | 0         -1    0          0            |
     | 0         0     0          1            |

     | Cos[q5]   -Sin[q5]   0   0     |
     | Sin[q5]   Cos[q5]    0   0     |
A5=  | 0         0          1   d5    |
     | 0         0          0   1     |
```

Figure A.2: A matrices for Lynx6 robot arm

6. Finally calculate the total transformation matrix H5. entering the command:

```
In[5]:MatrixForm[T[0,5]]
```

A.2 Link Transformations for Lynx6 Robot Arm.

$$A_1^0 = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.1}$$

$$A_2^1 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2c_2 \\ s_2 & c_2 & 0 & a_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.2}$$

$$A_3^2 = \begin{bmatrix} c_3 & -s_3 & 0 & a_3c_3 \\ s_3 & c_3 & 0 & a_3s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.3}$$

$$A_4^3 = \begin{bmatrix} c_4 & 0 & -s_4 & a_4c_4 \\ s_4 & 0 & c_4 & a_4s_4 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.4}$$

$$A_5^4 = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.5}$$

The matrices are calculated as following:

$$\begin{aligned} H_2^0 &= A_1^0 A_2^1 \\ &= \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_2 & -s_2 & 0 & a_2c_2 \\ s_2 & c_2 & 0 & a_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_1c_2 & -c_1s_2 & s_1 & a_2c_1c_2 \\ c_2s_1 & -s_1s_2 & -c_1 & a_2c_1s_1 \\ s_2 & c_2 & 0 & d_1 + a_2s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{A.6}$$

$$\begin{aligned} H_3^0 &= H_2^0 A_3^2 \\ &= \begin{bmatrix} c_1c_2 & -c_1s_2 & s_1 & a_2c_1c_2 \\ c_2s_1 & -s_1s_2 & -c_1 & a_2c_1s_1 \\ s_2 & c_2 & 0 & d_1 + a_2s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_3 & -s_3 & 0 & a_3c_3 \\ s_3 & c_3 & 0 & a_3s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} c_1c_2c_3 - c_1s_2s_3 & -c_1c_3s_2 - c_1s_2s_3 & s_1 & a_2c_1c_2 + a_3c_1c_2c_3 - a_3c_1s_2s_3 \\ c_2c_3s_1 - s_1s_2s_3 & -c_3s_1s_2 - c_2s_1s_3 & -c_1 & a_2c_2s_1 + a_3c_3c_2s_1 - a_3s_1s_2s_3 \\ c_3s_2 + c_2s_1 & c_2c_3 + s_2s_3 & 0 & d_1 + a_2s_2 + a_3c_3s_2 + a_3c_2s_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{A.7}$$

$$\begin{aligned} H_4^0 &= H_3^0 A_4^3 \\ &= \begin{bmatrix} c_1c_2c_3 - c_1s_2s_3 & -c_1c_3s_2 - c_1s_2s_3 & s_1 & a_2c_1c_2 + a_3c_1c_2c_3 - a_3c_1s_2s_3 \\ c_2c_3s_1 - s_1s_2s_3 & -c_3s_1s_2 - c_2s_1s_3 & -c_1 & a_2c_2s_1 + a_3c_3c_2s_1 - a_3s_1s_2s_3 \\ c_3s_2 + c_2s_1 & c_2c_3 + s_2s_3 & 0 & d_1 + a_2s_2 + a_3c_3s_2 + a_3c_2s_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_4 & 0 & -s_4 & a_4c_4 \\ s_4 & 0 & c_4 & a_4s_4 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \begin{matrix} c_1c_4(c_2c_3 - s_2s_3) - \\ c_1s_4(c_3s_2 + c_2s_3) \end{matrix} & -s_1 & \begin{matrix} -c_1c_4(c_3s_2 + c_2s_3) - \\ c_1s_4(c_2c_3 - s_2s_3) \end{matrix} & \begin{matrix} a_2c_1c_2 + a_3(c_2c_3 - s_2s_3) + a_4c_1c_4 \\ (c_2c_3 - s_2s_3) - a_4s_4c_1(c_3s_2 + c_2s_3) \end{matrix} \\ \begin{matrix} c_4s_1(c_2c_3 - s_2s_3) - \\ s_1s_4(c_3s_2 + c_2s_3) \end{matrix} & c_1 & \begin{matrix} -c_4s_1(c_3s_2 + c_2s_3) - \\ s_1s_4(c_2c_3 - s_2s_3) \end{matrix} & \begin{matrix} a_2c_2s_1 + a_3s_1(c_2c_3 - s_2s_3) + a_4c_4s_1 \\ (c_2c_3 - s_2s_3) - a_4s_4s_1(c_3s_2 + c_2s_3) \end{matrix} \\ \begin{matrix} c_4(c_3s_2 + c_2s_3) + \\ s_4(c_2c_3 - s_2s_3) \end{matrix} & 0 & \begin{matrix} c_4(c_2c_3 - s_2s_3) - \\ s_4(c_3s_2 + c_2s_3) \end{matrix} & \begin{matrix} d_1 + a_2s_2 + a_3(c_3s_2 - c_2s_3) + a_4c_4 \\ (c_3s_2 + c_2s_3) + a_4s_4(c_2c_3 - s_2s_3) \end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

(A.8)

$$H_5^0 = H_4^0 A_5^4$$

$$= \begin{bmatrix} c_1c_4(c_2c_3 - s_2s_3) - c_1s_4(c_3s_2 + c_2s_3) & -s_1 & -c_1c_4(c_3s_2 + c_2s_3) - c_1s_4(c_2c_3 - s_2s_3) & a_2c_1c_2 + a_3(c_2c_3 - s_2s_3) + a_4c_1c_4(c_2c_3 - s_2s_3) - a_4s_4c_1(c_3s_2 + c_2s_3) \\ c_4s_1(c_2c_3 - s_2s_3) - s_1s_4(c_3s_2 + c_2s_3) & c_1 & -c_4s_1(c_3s_2 + c_2s_3) - s_1s_4(c_2c_3 - s_2s_3) & a_2c_2s_1 + a_3s_1(c_2c_3 - s_2s_3) + a_4c_4s_1(c_2c_3 - s_2s_3) - a_4s_4s_1(c_3s_2 + c_2s_3) \\ c_4(c_3s_2 + c_2s_3) + s_4(c_2c_3 - s_2s_3) & 0 & c_4(c_2c_3 - s_2s_3) - s_4(c_3s_2 + c_2s_3) & d_1 + a_2s_2 + a_3(c_3s_2 + c_2s_3) + a_4c_4(c_3s_2 + c_2s_3) + a_4s_4(c_2c_3 - s_2s_3) \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c_1c_4c_5(c_2c_3 - s_2s_3) - c_1c_5s_4(c_3s_2 + c_2s_3) - s_1s_5 & -c_1c_5 - c_1c_4s_5(c_2c_3 - s_2s_3) - c_1s_4s_5(c_3s_2 + c_2s_3) & -c_1c_4(c_3s_2 + c_2s_3) - s_4c_1(c_2c_3 - s_2s_3) & a_2c_1c_2 + a_3c_1(c_2c_3 - s_2s_3) + a_4c_1c_4(c_2c_3 - s_2s_3) - a_4c_1s_4(c_3s_2 + c_2s_3) - d_5c_1c_4(c_3s_2 + c_2s_3) - d_5c_1s_4(c_2c_3 - s_2s_3) \\ c_4c_5s_1(c_2c_3 - s_2s_3) - s_1s_4(c_3s_2 + c_2s_3) + c_1s_5 & c_1c_5 - c_4s_1s_4(c_2c_3 - s_2s_3) - s_1s_4s_5(c_3s_2 + c_2s_3) & -c_4s_1(c_3s_2 + c_2s_3) - s_4s_1(c_2c_3 - s_2s_3) & a_2c_2s_1 + a_3s_1(c_2c_3 - s_2s_3) + a_4c_4s_1(c_2c_3 - s_2s_3) - a_4s_1s_4(c_3s_2 + c_2s_3) - d_5c_4s_1(c_3s_2 + c_2s_3) - d_5s_1s_4(c_2c_3 - s_2s_3) \\ c_4c_5(c_3s_2 + c_2s_3) + c_5s_4(c_2c_3 - s_2s_3) & -c_4s_5(c_3s_2 + c_2s_3) + s_4s_5(c_2c_3 - s_2s_3) & c_4(c_2c_3 - s_2s_3) - s_4(c_3s_2 + c_2s_3) & d_1 + a_2s_2 + a_3(c_3s_2 + c_2s_3) + a_4c_4(c_3s_2 + c_2s_3) + a_4s_4(c_2c_3 - s_2s_3) + d_5c_4(c_2c_3 - s_2s_3) - d_5s_4(c_3s_2 + c_2s_3) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(A.9)

The final transformation matrix H_5^0 for Lynx6 is:

$$H_5^0 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_1^0 A_2^1 A_3^2 A_4^3 A_5^4 \qquad \text{(A.10)}$$

where

$$r_{11} = c_1c_4c_5(c_2c_3 - s_2s_3) - c_1c_5s_4(c_3s_2 + c_2s_3) - s_1s_5 \qquad = c_1c_5c_{234} - s_1s_5$$
$$r_{21} = c_4c_5s_1(c_2c_3 - s_2s_3) - s_1s_4(c_3s_2 + c_2s_3) + c_1s_5 \qquad = c_1c_{234}s_1 + c_1c_5$$
$$r_{31} = c_4c_5(c_3s_2 + c_2s_3) + c_5s_4(c_2c_3 - s_2s_3) \qquad = c_5s_{234}$$

$$r_{12} = -c_1c_5 - c_1c_4s_5(c_2c_3 - s_2s_3) - c_1s_4s_5(c_3s_2 + c_2s_3) \qquad = -c_5s_1 - c_1c_{234}s_5$$
$$r_{22} = c_1c_5 - c_4s_1s_4(c_2c_3 - s_2s_3) - s_1s_4s_5(c_3s_2 + c_2s_3) \qquad = c_1c_5 - c_{234}s_1s_5$$
$$r_{32} = -c_4s_5(c_3s_2 + c_2s_3) + s_4s_5(c_2c_3 - s_2s_3) \qquad = -s_5s_{234}$$

$$r_{13} = -c_1c_4(c_3s_2 + c_2s_3) - s_4c_1(c_2c_3 - s_2s_3) \qquad = -c_1s_{234}$$
$$r_{23} = -c_4s_1(c_3s_2 + c_2s_3) - s_4s_1(c_2c_3 - s_2s_3) \qquad = -s_1s_{234}$$
$$r_{33} = c_4(c_2c_3 - s_2s_3) - s_4(c_3s_2 + c_2s_3) \qquad = c_{234}$$

$$x = a_2c_1c_2 + a_3c_1(c_2c_3 - s_2s_3) + a_4c_1c_4(c_2c_3 - s_2s_3) - a_4c_1s_4(c_3s_2 + c_2s_3) - d_5c_1c_4(c_3s_2 + c_2s_3) - d_5c_1s_4(c_2c_3 - s_2s_3)$$
$$= a_2c_1c_2 + a_3c_1c_{23} + a_4c_1c_{234} - d_5c_1s_{234}$$
$$y = a_2c_2s_1 + a_3s_1(c_2c_3 - s_2s_3) + a_4c_4s_1(c_2c_3 - s_2s_3) - a_4s_1s_4(c_3s_2 + c_2s_3) - d_5c_4s_1(c_3s_2 + c_2s_3) - d_5s_1s_4(c_2c_3 - s_2s_3)$$
$$= a_2c_2s_1 + a_3c_{23}s_1 + a_4c_{234}s_1 - d_5s_1s_{234}$$
$$z = d_1 + a_2s_2 + a_3(c_3s_2 + c_2s_3) + a_4c_4(c_3s_2 + c_2s_3) + a_4s_4(c_2c_3 - s_2s_3) + d_5c_4(c_2c_3 - s_2s_3) - d_5s_4(c_3s_2 + c_2s_3)$$
$$= a_2s_2 + a_3s_{23} + a_4s_{234} + d_5c_{234}$$

A.3 Inverse of Link Transformations Matrices.

$$\left[A_1^0\right]^{-1} = \begin{bmatrix} c_1 & s_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ s_1 & -c_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.11}$$

$$\left[A_2^1\right]^{-1} = \begin{bmatrix} c_2 & s_2 & 0 & -a_2 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.12}$$

$$\left[A_3^2\right]^{-1} = \begin{bmatrix} c_3 & s_3 & 0 & -a_3 \\ -s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.13}$$

$$\left[A_4^3\right]^{-1} = \begin{bmatrix} c_4 & s_4 & 0 & -a_4 \\ 0 & 0 & 1 & 0 \\ -s_4 & c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.14}$$

$$\left[A_5^4\right]^{-1} = \begin{bmatrix} c_5 & s_5 & 0 & 0 \\ -s_5 & c_5 & 0 & 0 \\ 0 & 0 & 1 & -d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.15}$$

The analytical solution for deriving the joint variables is given as.

Multiply both sides of equation (1.6) by $\left[A_1^0\right]^{-1}$.

$$\left[A_1^0\right]^{-1} H_5^0 = A_2^1 A_3^2 A_4^3 A_5^4 \tag{A.16}$$

Since

$$\begin{bmatrix} c_1 r_{11} + s_1 r_{21} & c_1 r_{12} + s_1 r_{22} & c_1 r_{13} + s_1 r_{23} & c_1 x + s_1 y \\ r_{31} & r_{32} & r_{31} & z - d_1 \\ s_1 r_{11} - c_1 r_{21} & s_1 r_{12} - c_1 r_{22} & s_1 r_{13} - c_1 r_{23} & s_1 x - c_1 y \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_5 c_{234} & -s_5 c_{234} & -s_{234} & a_2 c_2 + a_3 c_{23} + a_4 c_{234} - d_5 s_{234} \\ c_5 s_{234} & -s_5 s_{234} & c_{234} & a_2 s_2 + a_3 s_{23} + a_4 s_{234} + d_5 c_{234} \\ -s_5 & -c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.17}$$

Equating the elements from the two sides of the above equation, we have:

$$s_1 r_{11} - c_1 r_{21} = -s_5 \tag{A.18}$$
$$s_1 r_{12} - c_1 r_{22} = -c_5 \tag{A.19}$$
$$s_1 r_{13} - c_1 r_{23} = 0 \tag{A.20}$$
$$s_1 x - c_1 y = 0 \tag{A.21}$$

From the above equations the result is:

$$\theta_1 = A\tan 2(y,x) \tag{A.22}$$

$$\theta_5 = A\tan 2(s_5,c_5) \tag{A.23}$$

The elements (1, 2) and (2, 2) can solve as:

$$\begin{gathered} -c_1r_{12} + s_1r_{22} = -s_5c_4c_{23} - s_5s_4s_{23} \\ r_{32} = -s_5c_4s_{23} - s_5s_4c_{23} \end{gathered} \tag{A.24}$$

$$\theta_4 = A\tan 2(s_4,c_4) \tag{A.25}$$

where

$$\begin{gathered} s_4 = \frac{c_1r_{12} + s_1r_{22} + s_5c_4c_{23}}{s_5s_{23}} \\ c_4 = \frac{r_{32} + s_5s_4c_{23}}{s_5s_{23}} \end{gathered} \tag{A.26}$$

Similarly, we can solve the other angles as the same way

$$\theta_2 = A\tan 2(s_2,c_2) \tag{A.27}$$

and

$$\theta_3 = A\tan 2(s_3,c_3) \tag{A.28}$$

Low of inverse matrices: For 3×3 inverse

$$A = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{A.29}$$

The matrix inverse is:

$$A^{-1} = \frac{1}{|A|}\begin{bmatrix} \begin{vmatrix} r_{22} & r_{23} \\ r_{32} & r_{33} \end{vmatrix} & \begin{vmatrix} r_{13} & r_{12} \\ r_{33} & r_{32} \end{vmatrix} & \begin{vmatrix} r_{12} & r_{13} \\ r_{22} & r_{23} \end{vmatrix} \\ \begin{vmatrix} r_{23} & r_{21} \\ r_{32} & r_{31} \end{vmatrix} & \begin{vmatrix} r_{11} & r_{13} \\ r_{31} & r_{33} \end{vmatrix} & \begin{vmatrix} r_{13} & r_{11} \\ r_{23} & r_{21} \end{vmatrix} \\ \begin{vmatrix} r_{21} & r_{22} \\ r_{31} & r_{32} \end{vmatrix} & \begin{vmatrix} r_{12} & r_{11} \\ r_{32} & r_{31} \end{vmatrix} & \begin{vmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{vmatrix} \end{bmatrix} \tag{A.30}$$

A.4 Deriving Transfer Function for Motor Position

The electrical equation for DC motor system is obtained based on Kirchhoff's Voltage Law as follows:

$$R_a i_a(t) + L_a \frac{di_a(t)}{dt} = v_a(t) - v_b(t) \tag{A.31}$$

On the other side, the mechanical equation (Newton low of motion) is obtained as follows:

$$\tau_m(t) = J_m \frac{d^2\theta(t)}{dt^2} + B_m \frac{d\theta(t)}{dt} = J_m \ddot{\theta}(t) + B_m \dot{\theta}(t) \tag{A.32}$$

Based on the previous two equations (A.31) and (A.32), when the input voltage $v_a(t)$, is applied, the armature current $i_a(t)$ goes through resistance R_a and inductance L_a

producing magnetic flux and causing the motion of the rotor according to the motor torque as illustrated in equation (A.33)

$$\tau_m(t) = K_t i_a(t) \tag{A.33}$$

This motion induces the back EMF by the angular speed of the motor shaft as follows:

$$V_b(t) = K_b \omega_m(t) = K_b \frac{d\theta(t)}{dt} \tag{A.34}$$

The equations (A.31), (A.32), (A.33) and (A.34) are combined as follow:

$$L_a \frac{di_a(t)}{dt} + R_a i_a(t) = v_a(t) - K_b \frac{d\theta(t)}{dt} \tag{A.35}$$

$$J_m \frac{d^2\theta(t)}{dt^2} + B_m \frac{d\theta(t)}{dt} = K_t i_a(t) \tag{A.36}$$

Transforming the above two equations using Laplace transformation, we obtain the two equations as follows:

$$L_a s I(s) + R_a I(s) = V_a(s) - K_b s\theta(s) \tag{A.37}$$

and

$$J_m s^2\theta(s) + B_m s\theta(s) = K_t I(s) \tag{A.38}$$

Substituting (A.37) in (A.38) gives the motor speed the multiply by 1/s as follows:

$$G_{position}(s) = \frac{\theta(s)}{V(s)} = \frac{K_t}{s[J_m L_a s^2 + (L_a B_m + J_m R_a)s + K_t K_b]} \tag{A.39}$$

A.5 Deriving Equation (2.35) the Transfer Function From the Load Torque to Input.

Let:

$$\frac{C(s)}{X(s)} = \frac{1}{Js + D} \Rightarrow C(s) = \frac{X(s)}{Js + D} \tag{A.40}$$

$$X(s) = \frac{E(s)}{L_a s + R_a} - \frac{D(t)}{gr} \tag{A.41}$$

The error will be –C(s) for reference input zero, so

$$X(s) = \frac{C(s) K_b K_t}{L_a s + R_a} - \frac{D(t)}{gr} \tag{A.42}$$

$$= \frac{-X(s) K_b K_t}{(L_a s + R_a)(J_m s + D_m)} - \frac{D(t)}{gr} \tag{A.43}$$

$$X(s) + \frac{-X(s) K_b K_t}{(L_a s + R_a)(J_m s + D_m)} = \frac{D(t)}{gr} \tag{A.44}$$

$$X(s) = \frac{-D(t)/gr}{1 + \frac{K_b K_t}{(L_a s + R_a)(J_m s + D_m)}} \tag{A.45}$$

$$\frac{-grC(s)\left[1 + \frac{K_b K_t}{(L_a s + R_a)(J_m s + D_m)}\right]}{D(t)} = \frac{1}{(J_m s + D_m)} \tag{A.46}$$

$$\frac{-grC(s)}{D(t)} = \frac{1}{(J_m s + D_m)\left(1 + \frac{K_b K_t}{(L_a s + R_a)(J_m s + D_m)}\right)} \tag{A.47}$$

$$\frac{C(s)}{D(t)} = \frac{(L_a s + R_a)/gr}{(L_a s + R_a)(J_m s + D_m) + K_b K_t} \tag{A.48}$$

APPENDIX B: ROOT LOCUS ANALYSES METHOD

As mentioned in Chapter 3, root locus method is a technique that can be used as a tool for designing PID control parameters. There are some steps used in designing PID parameters:

1. Evaluate the uncompensated system at the required system characteristics. The required response specifications are rising time (Ts) less than 0.2 sec, overshoot less than 10% and steady state error around 0.2.

Uncompensated open loop transfer function is:

$$G(s)K = \frac{0.235K}{s(s+162.2305)(s+38.77)} \tag{B.1}$$

The dominant pole ate $s_1 = -16.4 + j22.1$ with $K_1 = 6.51$.

2. Design PD control to meet the transient response specification.

The design point $s_2 = -24.4 + j33.6$

The PD controller is:

$$G_{PD}(s) = (s+90.45) \tag{B.2}$$

3. After designing PD controller, it will be design integral compensator to reduce steady state error to zero for step unit.

Choose ideal integral compensator:

$$G_{PI}(s) = \frac{s+0.1}{s} \tag{B.3}$$

The dominant pole at $s_3 = -25.8 + j35.1$

4. The overall transfer function is:

$$G_{PID}(s)G(s) = \frac{K_3(s+90.45)(s+0.1)}{s^2(s+162.2305)(s+38.77)} \tag{B.4}$$

From equation (B.4) determine the K_P, K_P, and K_D gains. The PID equation is:

$$G_{PID}(s) = \frac{K_3(s+90.45)(s+0.1)}{s} = \frac{0.16(s^2+90.45+9.045)}{s} \tag{B.5}$$

Matching $K_D s^2 + K_P s + K_I \big/ s$ and equation (B.5) then the gains of PID controller are $K_P = 14$, $K_I = 1.4$ and $K_D = 0.16$ respectively.

5. Table (B.1) summarizes the value of the uncompensated, PD and PID compensated system.

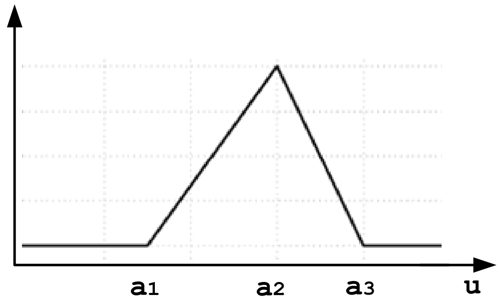
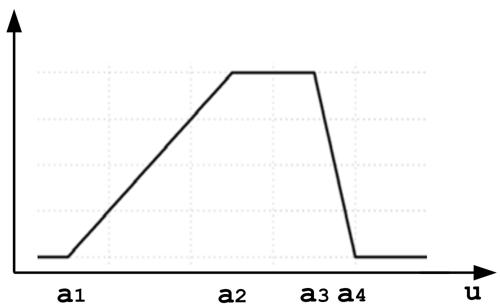
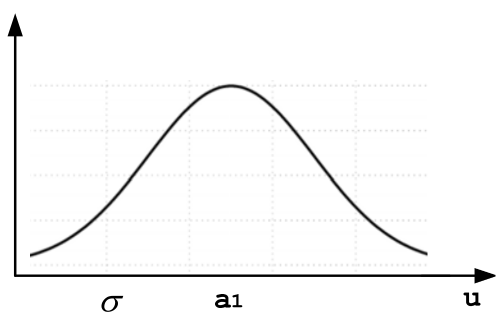
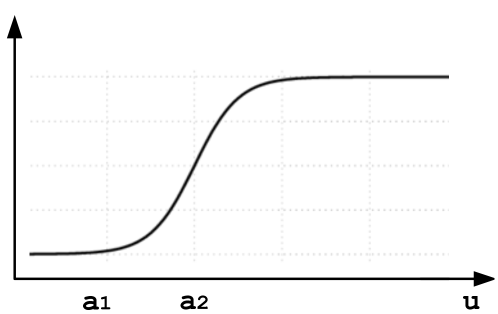
Table B.1: Table characteristics

	Uncompensated	**PD- compensated**	**PID compensated**
Plant &compensator	$\frac{0.235K}{s(s+162.23)(s+38.77)}$	$\frac{0.235K\,(s+90.45)}{s\,(s+162.23)(s+38.77)}$	$\frac{0.235K\,(s+90.45)(s+0.1)}{s^2\,(s+162.23)(s+38.77)}$
Dominant pole	$S_1 = -16.4 + j22.1$	$S_2 = -24.6 + j33.6$	$S_3 = -25.8 + j35.1$
K	6.51	0.154	0.16
ζ	0.591	0.591	0.591
w_n	27.75	41.62	43.65
$\%OS$	10	10	10
T_s	0.244	0.162	0.155
T_p	0.142	0.094	0.089
$e(\infty)$	0	0	0

APPENDIX C: FUZZY MEMBERSHIP FUNCTION AND DEFUZZIFICATION

C.1 Membership Function Types

Table C.1: Membership functions

Function Name	Example	Mathematical Description
Triangle		$\mu(u)=\begin{cases}0 & if\ \ u\le a_1\\ \dfrac{u-a_1}{a_2-a_1} & if\ \ a_1\le u\le a_2\\ \dfrac{a_3-u}{a_3-a_2} & if\ \ a_2\le u\le a_3\\ 0 & if\ \ u\ge a_3\end{cases}$
Trapezoidal		$\mu(u)=\begin{cases}0 & if\ \ u\le a_1\\ \dfrac{u-a_1}{a_2-a_1} & if\ \ a_1\le u\le a_2\\ 1 & if\ \ a_2\le u\le a_3\\ \dfrac{a_3-u}{a_3-a_2} & if\ \ a_3\le u\le a_4\\ 0 & if\ \ u\ge a_4\end{cases}$
Gaussian		$\mu(u)=e^{\frac{-(u-a)}{2\sigma^2}}$ where *: the mean* a_1*, and variance* σ
Sigmoid		$\mu(u)=\dfrac{1}{1+e^{-a_1(u-a_2)}}$

C.2 Defuzzification Methods

1. Centroid Method

Centroid method is the center of area or center of gravity method. It the most prevalent and physically method of all defuzzification methods.

The following expression represents the centroid method.

$$x = \int \mu_c(x)dx \Big/ \mu_c(x)dx \tag{C.1}$$

where x is the defuzzification value. This method illustrated in Figure C.1. Here the corresponding element for the centroid method equals to 0.4724

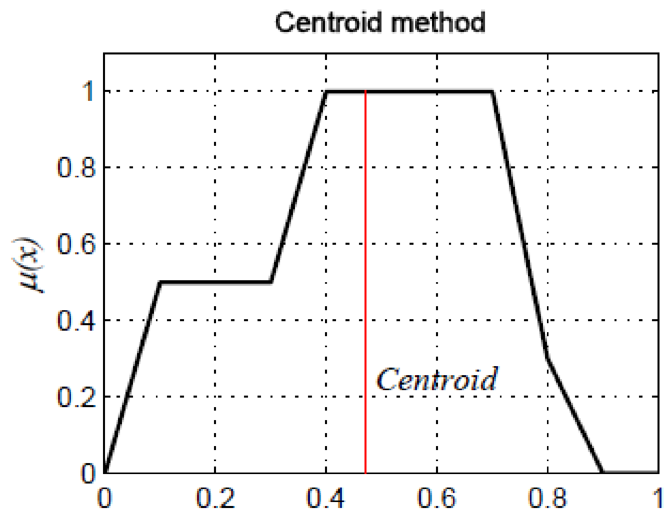


Figure C.1: Centroid defuzzification method

2. Bisector Method

This method illustrated in Figure C.2, and the corresponding element for the bisector defuzzification method equals to 0.5

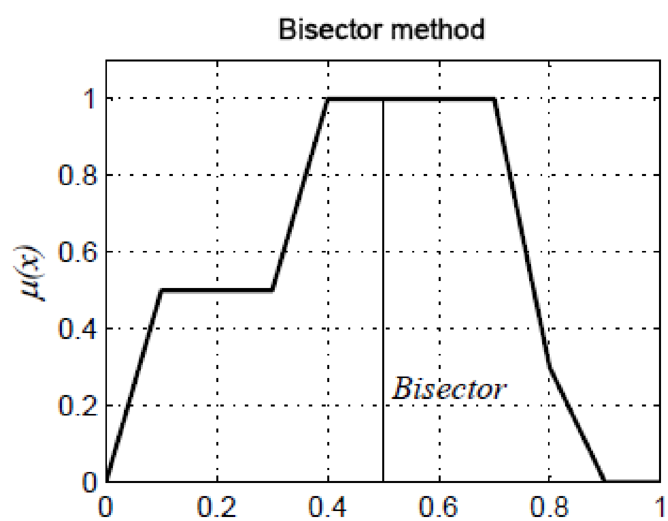


Figure C.2: Bisector defuzzification method

3. Mean of Maximum Method (MOM)

This method illustrated in Figure C.3, and the corresponding element for the MOM defuzzification method equals to 0.55. Mean of Max method represented by the following equation:

$$x = (a+b)/2 \tag{C.2}$$

where a=0.4 and b=0.7 are the first and the last maximum values in Figure C.3

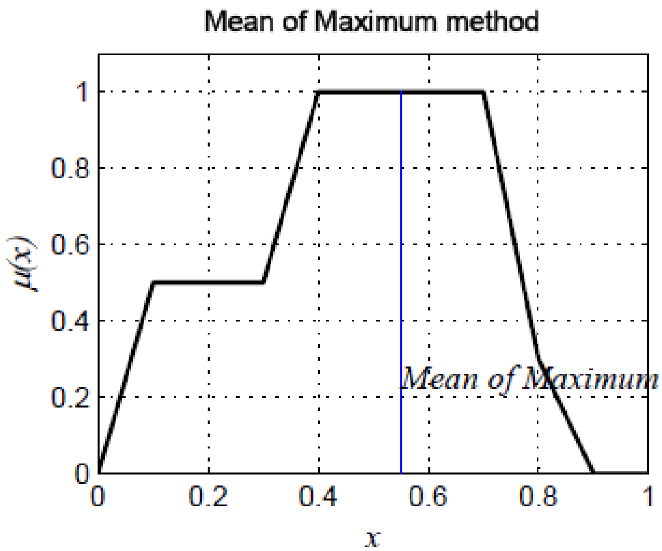


Figure C.3: Mean of Max defuzzification method

4. Largest of Maximum Method (LOM)

This method use the union of the fuzzy set and takes the largest value of the domain with maximal membership degree. Figure C.4 shows this method. The corresponding element for the MOM defuzzification method equals to 0.7.

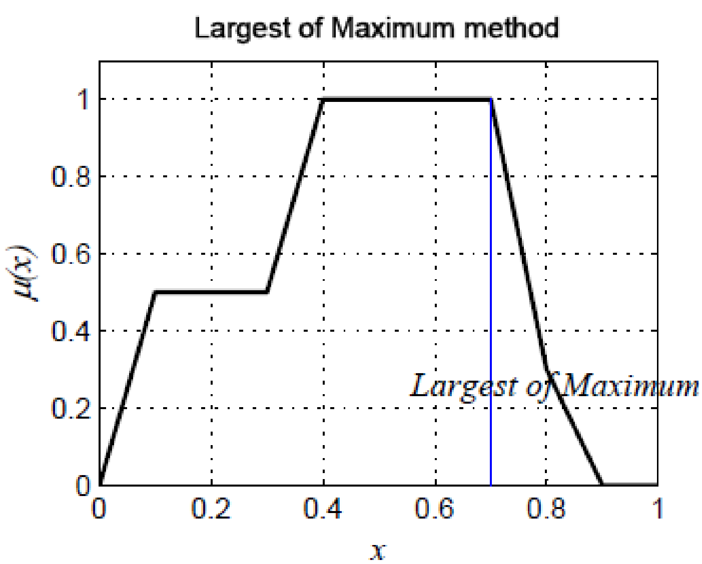


Figure C.4: Largest of Maximum defuzzification method

5. Smallest of Maximum Method (SOM)

This method the same as the largest of maximum method but it takes the smallest value of the domain with maximal membership degree. The corresponding output of this method is equal 0.4. Figure C.5 shows the SOM method.

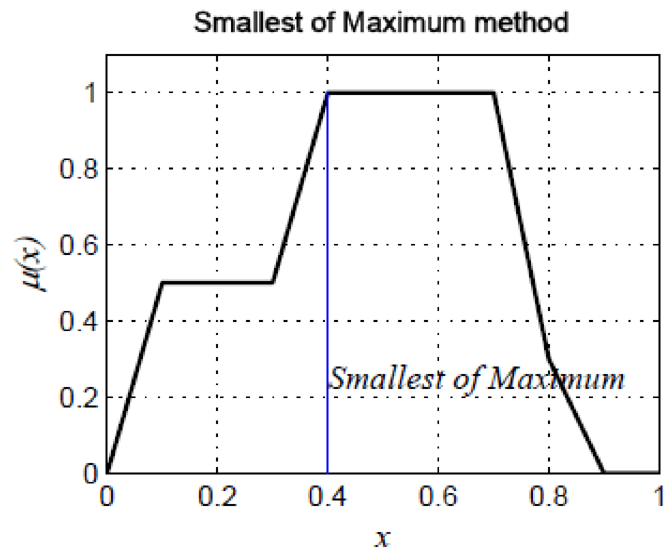


Figure C.5: Smallest of Maximum defuzzification method

C.3 Rule Base
C.3.1 Control rule base for fuzzy controller

Table C.2: Rule base for fuzzy controller

RULES FOR TUNING PID GAINS												
1	IF	e	Is	NB	And	Δe	Is	NB	Then	Output	Is	NB
2	IF	e	Is	NM	And	Δe	Is	NB	Then	Output	Is	NM
3	IF	e	Is	NS	And	Δe	Is	NB	Then	Output	Is	NS
4	IF	e	Is	ZE	And	Δe	Is	NB	Then	Output	Is	NS
5	IF	e	Is	PS	And	Δe	Is	NB	Then	Output	Is	PS
6	IF	e	Is	PM	And	Δe	Is	NB	Then	Output	Is	PM
7	IF	e	Is	PB	And	Δe	Is	NB	Then	Output	Is	PB
8	IF	e	Is	NM	And	Δe	Is	NM	Then	Output	Is	NM
9	IF	e	Is	NS	And	Δe	Is	NM	Then	Output	Is	NS
10	IF	e	Is	ZE	And	Δe	Is	NM	Then	Output	Is	NS
11	IF	e	Is	PS	And	Δe	Is	NM	Then	Output	Is	PS
12	IF	e	Is	PM	And	Δe	Is	NM	Then	Output	Is	PM
13	IF	e	Is	NS	And	Δe	Is	NS	Then	Output	Is	NS
14	IF	e	Is	ZE	And	Δe	Is	NS	Then	Output	Is	NS
15	IF	e	Is	PS	And	Δe	Is	NS	Then	Output	Is	PS
16	IF	e	Is	NB	And	Δe	Is	ZE	Then	Output	Is	NB
17	IF	e	Is	NM	And	Δe	Is	ZE	Then	Output	Is	NM
18	IF	e	Is	NS	And	Δe	Is	ZE	Then	Output	Is	NS
19	IF	e	Is	ZE	And	Δe	Is	ZE	Then	Output	Is	ZE
20	IF	e	Is	PS	And	Δe	Is	ZE	Then	Output	Is	PS

21	IF	e	Is	PM	And	Δe	Is	ZE	Then	Output	Is	PM
22	IF	e	Is	PB	And	Δe	Is	ZE	Then	Output	Is	PB
23	IF	e	Is	NS	And	Δe	Is	PS	Then	Output	Is	NS
24	IF	e	Is	ZE	And	Δe	Is	PS	Then	Output	Is	PS
25	IF	e	Is	PS	And	Δe	Is	PS	Then	Output	Is	PS
26	IF	e	Is	NM	And	Δe	Is	PM	Then	Output	Is	NM
27	IF	e	Is	NS	And	Δe	Is	PM	Then	Output	Is	NS
28	IF	e	Is	ZE	And	Δe	Is	PM	Then	Output	Is	PS
29	IF	e	Is	PS	And	Δe	Is	PM	Then	Output	Is	PS
30	IF	e	Is	NB	And	Δe	Is	PB	Then	Output	Is	NB
31	IF	e	Is	NM	And	Δe	Is	PB	Then	Output	Is	NM
32	IF	e	Is	NS	And	Δe	Is	PB	Then	Output	Is	NS
33	IF	e	Is	ZE	And	Δe	Is	PB	Then	Output	Is	PS
34	IF	e	Is	PS	And	Δe	Is	PB	Then	Output	Is	PS
35	IF	e	Is	PM	And	Δe	Is	PB	Then	Output	Is	PM
36	IF	e	Is	PB	And	Δe	Is	PB	Then	Output	Is	PB

C.3.2 Control rule base for tuning PID parameters

Table C.3: Rule base for tuning PID controller

RULES FOR TUNING PID GAINS																		
1	IF	e	Is	NB	And	Δe	Is	NB	Then	K'_P	Is	B	K'_D	Is	S	K'_I	Is	S
2	IF	e	Is	NB	And	Δe	Is	NM	Then	K'_P	Is	B	K'_D	Is	S	K'_I	Is	S
3	IF	e	Is	NB	And	Δe	Is	NS	Then	K'_P	Is	B	K'_D	Is	S	K'_I	Is	S
4	IF	e	Is	NB	And	Δe	Is	Z	Then	K'_P	Is	B	K'_D	Is	S	K'_I	Is	S
5	IF	e	Is	NB	And	Δe	Is	PS	Then	K'_P	Is	B	K'_D	Is	S	K'_I	Is	S
6	IF	e	Is	NB	And	Δe	Is	PM	Then	K'_P	Is	B	K'_D	Is	S	K'_I	Is	S
7	IF	e	Is	NB	And	Δe	Is	PB	Then	K'_P	Is	B	K'_D	Is	S	K'_I	Is	S
8	IF	e	Is	NM	And	Δe	Is	NB	Then	K'_P	Is	S	K'_D	Is	B	K'_I	Is	M
9	IF	e	Is	NM	And	Δe	Is	NM	Then	K'_P	Is	B	K'_D	Is	B	K'_I	Is	M
10	IF	e	Is	NM	And	Δe	Is	NS	Then	K'_P	Is	B	K'_D	Is	S	K'_I	Is	S
11	IF	e	Is	NM	And	Δe	Is	Z	Then	K'_P	Is	B	K'_D	Is	S	K'_I	Is	S
12	IF	e	Is	NM	And	Δe	Is	PS	Then	K'_P	Is	B	K'_D	Is	S	K'_I	Is	S
13	IF	e	Is	NM	And	Δe	Is	PM	Then	K'_P	Is	B	K'_D	Is	B	K'_I	Is	M
14	IF	e	Is	NM	And	Δe	Is	PB	Then	K'_P	Is	S	K'_D	Is	B	K'_I	Is	M
15	IF	e	Is	NS	And	Δe	Is	NB	Then	K'_P	Is	S	K'_D	Is	B	K'_I	Is	B
16	IF	e	Is	NS	And	Δe	Is	NM	Then	K'_P	Is	S	K'_D	Is	B	K'_I	Is	M
17	IF	e	Is	NS	And	Δe	Is	NS	Then	K'_P	Is	B	K'_D	Is	B	K'_I	Is	M
18	IF	e	Is	NS	And	Δe	Is	Z	Then	K'_P	Is	B	K'_D	Is	S	K'_I	Is	S
19	IF	e	Is	NS	And	Δe	Is	PS	Then	K'_P	Is	B	K'_D	Is	B	K'_I	Is	M
20	IF	e	Is	NS	And	Δe	Is	PM	Then	K'_P	Is	S	K'_D	Is	B	K'_I	Is	M
21	IF	e	Is	NS	And	Δe	Is	PB	Then	K'_P	Is	S	K'_D	Is	B	K'_I	Is	B
22	IF	e	Is	Z	And	Δe	Is	NB	Then	K'_P	Is	S	K'_D	Is	B	K'_I	Is	B

23	IF	e	Is	Z	And	Δe	Is	NM	Then	K_P'	Is	S	K_D'	Is	B	K_I'	Is	B
24	IF	e	Is	Z	And	Δe	Is	NS	Then	K_P'	Is	S	K_D'	Is	B	K_I'	Is	M
25	IF	e	Is	Z	And	Δe	Is	Z	Then	K_P'	Is	B	K_D'	Is	B	K_I'	Is	M
26	IF	e	Is	Z	And	Δe	Is	PS	Then	K_P'	Is	S	K_D'	Is	B	K_I'	Is	M
27	IF	e	Is	Z	And	Δe	Is	PM	Then	K_P'	Is	S	K_D'	Is	B	K_I'	Is	B
28	IF	e	Is	Z	And	Δe	Is	PB	Then	K_P'	Is	S	K_D'	Is	B	K_I'	Is	B
29	IF	e	Is	PS	And	Δe	Is	NB	Then	K_P'	Is	S	K_D'	Is	B	K_I'	Is	B
30	IF	e	Is	PS	And	Δe	Is	NM	Then	K_P'	Is	S	K_D'	Is	B	K_I'	Is	M
31	IF	e	Is	PS	And	Δe	Is	NS	Then	K_P'	Is	B	K_D'	Is	B	K_I'	Is	M
32	IF	e	Is	PS	And	Δe	Is	Z	Then	K_P'	Is	B	K_D'	Is	S	K_I'	Is	S
33	IF	e	Is	PS	And	Δe	Is	PS	Then	K_P'	Is	B	K_D'	Is	B	K_I'	Is	M
34	IF	e	Is	PS	And	Δe	Is	PM	Then	K_P'	Is	S	K_D'	Is	B	K_I'	Is	M
35	IF	e	Is	PS	And	Δe	Is	PB	Then	K_P'	Is	S	K_D'	Is	B	K_I'	Is	B
36	IF	e	Is	PM	And	Δe	Is	NB	Then	K_P'	Is	S	K_D'	Is	B	K_I'	Is	M
37	IF	e	Is	PM	And	Δe	Is	NM	Then	K_P'	Is	B	K_D'	Is	B	K_I'	Is	M
38	IF	e	Is	PM	And	Δe	Is	NS	Then	K_P'	Is	B	K_D'	Is	S	K_I'	Is	S
39	IF	e	Is	PM	And	Δe	Is	Z	Then	K_P'	Is	B	K_D'	Is	S	K_I'	Is	S
40	IF	e	Is	PM	And	Δe	Is	PS	Then	K_P'	Is	B	K_D'	Is	S	K_I'	Is	S
41	IF	e	Is	PM	And	Δe	Is	PM	Then	K_P'	Is	B	K_D'	Is	B	K_I'	Is	M
42	IF	e	Is	PM	And	Δe	Is	PB	Then	K_P'	Is	S	K_D'	Is	B	K_I'	Is	M
43	IF	e	Is	PB	And	Δe	Is	NB	Then	K_P'	Is	B	K_D'	Is	S	K_I'	Is	S
44	IF	e	Is	PB	And	Δe	Is	NM	Then	K_P'	Is	B	K_D'	Is	S	K_I'	Is	S
45	IF	e	Is	PB	And	Δe	Is	NS	Then	K_P'	Is	B	K_D'	Is	S	K_I'	Is	S
46	IF	e	Is	PB	And	Δe	Is	Z	Then	K_P'	Is	B	K_D'	Is	S	K_I'	Is	S
47	IF	e	Is	PB	And	Δe	Is	PS	Then	K_P'	Is	B	K_D'	Is	S	K_I'	Is	S
48	IF	e	Is	PB	And	Δe	Is	PM	Then	K_P'	Is	B	K_D'	Is	S	K_I'	Is	S
49	IF	e	Is	PB	And	Δe	Is	PB	Then	K_P'	Is	B	K_D'	Is	S	K_I'	Is	S

APPENDIX D: MATLAB AND SIMULINK

MATLAB and SIMULINK are both programs that were created by Mathwork Inc. MATLAB is powerful mathematical tool allowing most mathematical operations. A lot of function used in MTLAB also used in SIMULINK program, which is considered as a graphical interface where block diagrams is drawn which represents the program. Programming using SIMULINK is easier than MATLAB because the code to perform each task has been written for MATLAB and included in SIMULINK. When block in SIMULINK is run, the code is called automatically to generate the output. This appendix presents the SIMULINK block diagrams and the some MATLAB codes for control methods.

The first SIMULINK Diagram, Figure D.1 shows the independent joint control of the PID controller for five DOF robot arm. The input of the controller as seen is the five angles and the disturbance for each one of them. The output of the controller is the position of the robot arm and it represents b five thetas.

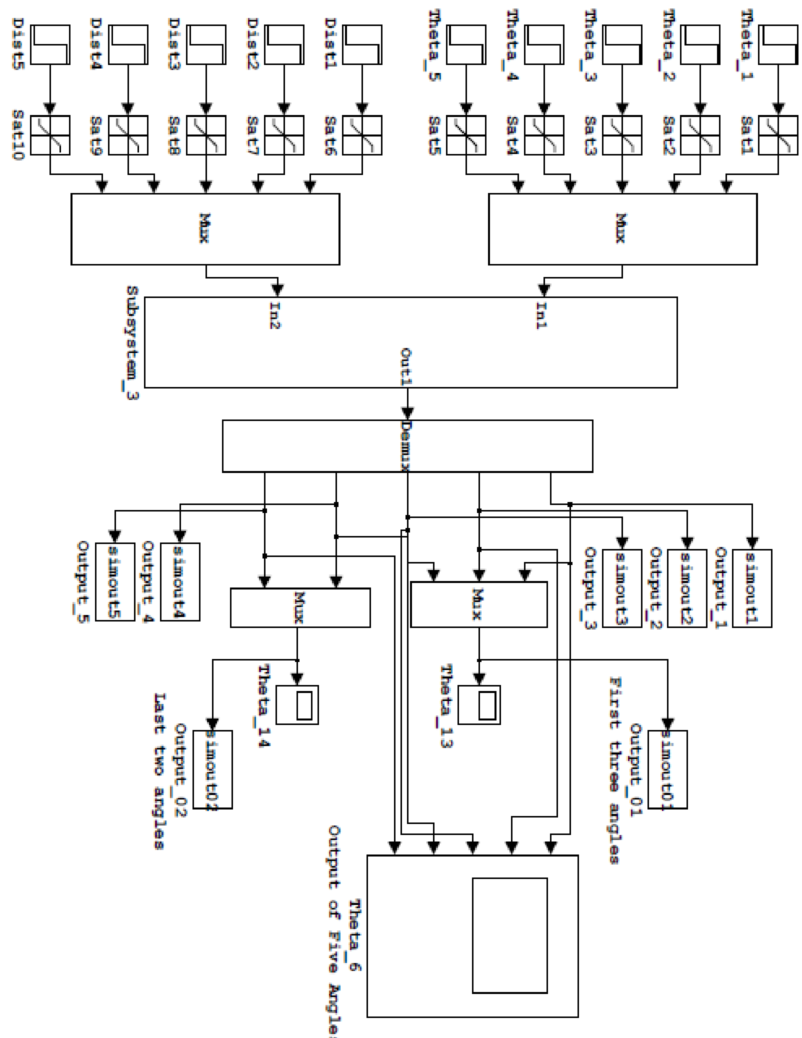


Figure D.1: Independent joint control of five DOF robot arm

Figure D.2 represents the fuzzy PID controller. As shown in the figure the fuzzy PID control combine between fuzzy PD and fuzzy PD controller. G2, G4, G1, G5, and are the gain of the error, error change and error rate respectively. G3 and G6 the output gains of the fuzzy PD and fuzzy PI controllers.

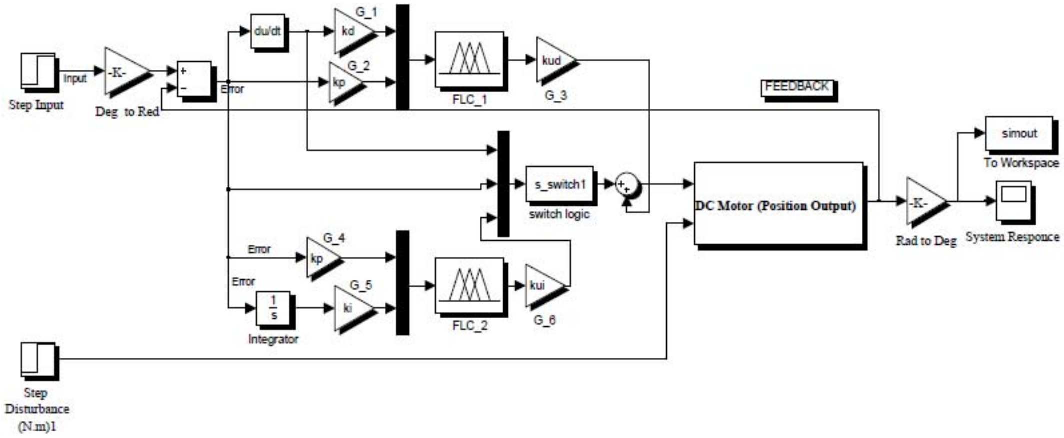


Figure D.2: Fuzzy PID controller

Figure D.3 shows the fuzzy supervisory control and subsystems. By locking under the mask of the motor position system, it shows the DC motor system. The components of the DC motor are shown as discussed in Chapter 2. the first block in DC model represents the electrical term and the second block represents the mechanical term while Km and Kb are the torque constant and back EMF constant respectively. Double click on the PID subsystem, the PID controller is shown in Figure D.4.

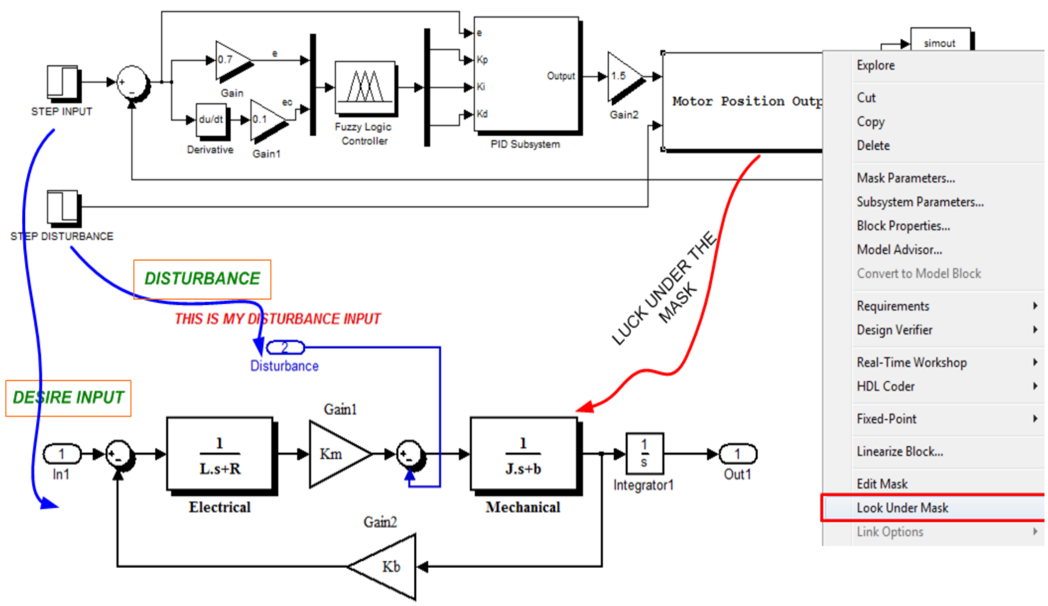


Figure D.3: Fuzzy supervisory controller

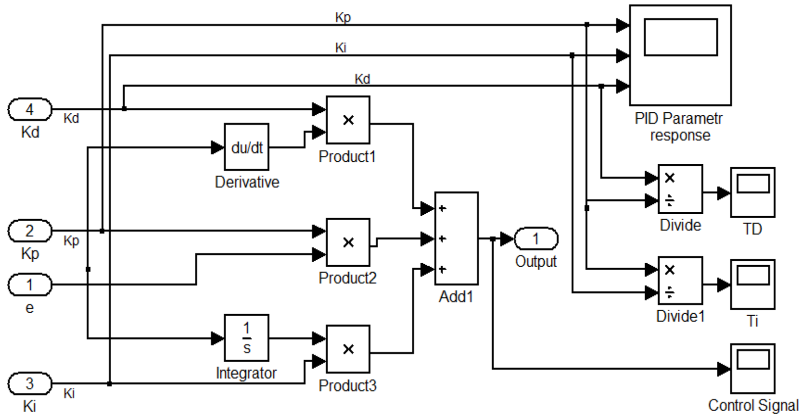


Figure D.4: PID controller

Figure D.5 shows the output response of 1 DOF of the robot arm using fuzzy supervisory control and the variation of the PID parameters during operation.

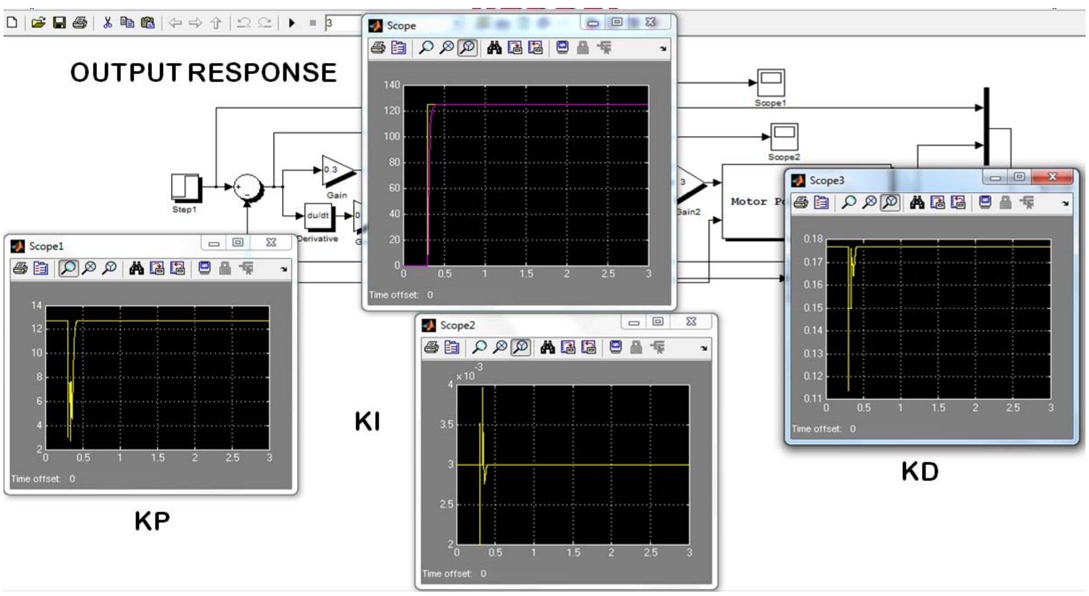


Figure D.5: Output response using FSC with PID parameter variations

```
% ************************************************** *
%           MATLAB Program plotting the response     *
%              of the SIMULINK Model                 *
%           Represented in chapter two               *
% ************************************************** *

sim('DCMotor');
  y1=simout1.signals.values;
  y2=simout2.signals.values;
  y3=simout3.signals.values;
  y4=simout4.signals.values;
  t1 = simout1.time;
  t2 = simout2.time;
  t3 = simout3.time;
```

```
  t4 = simout4.time;
figure(1)
subplot(411), plot(t1,y1,'-k','Linewidth',1.3),
title('Output Step Response','fontsize',12,'fontweight','b')
subplot(412), plot(t2,y2,'-B','Linewidth',1.3),
ylabel('Torque','fontsize',10),grid on
subplot(413), plot(t3,y3,'-k','Linewidth',1.3),
ylabel('Speed','fontsize',10),grid on
subplot(414), plot(t4,y4,'-B','Linewidth',1.3),
ylabel('Position');xlabel('Time(sec)','fontsize',10), grid

        % ************************************************* *
        %          MATLAB Program for plotting              *
        %          SIMULINK Model response                  *
        %        Output Response PID controller             *
        %       IJC = Independent Joint Control             *
        % ************************************************* *

% Inter the five angle of the robot manipulator
T1=121;     T2=66;      T3=100;     T4=45;      T5=15;
% Inter the Disturbance for each angle of the robot manipulator
D1=0; D2=0; D3=0; D4=0; D5=0;

sim ('IJC_PID');
  y1=simout1.signals.values;
  y2=simout2.signals.values;
  y3=simout3.signals.values;
  y4=simout4.signals.values;
  y5=simout5.signals.values;
  t1 = simout1.time;
  t2 = simout2.time;
  t3 = simout3.time;
  t4 = simout4.time;
  t5 = simout5.time;
figure(1)
subplot(511),plot(t1,y1,'-k','Linewidth',1.3),xlabel('Time(sec)'),
ylabel('\theta_1','fontsize',11),title('PID Out_1'),grid on
subplot(512),plot(t2,y2,'k','Linewidth',1.3),xlabel('Time(sec)'),
ylabel('\theta_2','fontsize',11),title('PID Out_2'),grid on
subplot(513),plot(t3,y3,'-k','Linewidth',1.3),xlabel('Time(sec)'),
ylabel('\theta_3','fontsize',11),title('PID Out_3'),grid on
subplot(514),plot(t4,y4,'-k','Linewidth',1.3),xlabel('Time(sec)'),
ylabel('\theta_4','fontsize',11), title('PID Out_4'),grid on
subplot(515),plot(t5,y5,'-k','Linewidth',1.3),xlabel('Time(sec)'),
ylabel('\theta_5','fontsize',11), title('PID Out_5'),grid on
```