Performance Study of Kinematic Control of 6 Dof Robot Arm Using Fuzzy Logic Control

JIMMA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

JIMMA INSTITUTE OF TECHNOLOGY

FACULTY OF MECHANICAL ENGINEERING

MANUFACTURING SYSTEMS ENGINEERING STREAM

A Thesis submitted to Graduate Studies of Jimma University in Partial Fulfillment of the Requirements for the Degree of Master Science in Mechanical Engineering (Manufacturing System Engineering)

By: Roman Girma Muluneh

March 2021

Jimma, Ethiopia

JIMMA UNIVERSITY

SCHOOL GRADUATE STUDIES

JIMMA INSTITUTE OF TECHNOLOGY

FACULTY OF MECHANICAL ENGINEERING

MANUFACTURING SYSTEMS ENGINEERING STREAM

Performance study of Kinematic Control of 6 DOF Robot Arm Using Fuzzy Logic Control

A Thesis submitted to Graduate Studies of Jimma University in Partial Fulfillment of the Requirements for the Degree of Master Science in Mechanical Engineering (Manufacturing System Engineering)

By: Roman Girma Muluneh

Advisor: Dr. Mesay Alemu (Ass. Prof)

Co advisor: Mr. Abu Feyo (MSc.)

March 2021

Jimma, Ethiopia

Declaration

I hereby declare that by my signature below this thesis “**Performance study of Kinematic Control of 6 DOF Robot Arm Using Fuzzy Logic Control’’** is my own work and this has not been submitted elsewhere for the award of any other degree or diploma. It is being submitted for the degree of Master Science in Mechanical Engineering (specialization in Manufacturing System Engineering), and all source of material used for this have been dully acknowledged.

Mrs. Roman Girma	-----------------------------	--------------------------
Name of candidate	Signature	Date

Jimma University
Jimma Institute of Technology
Faculty of Mechanical Engineering

Approval sheet

As a member of the examination committee of the final Master of Science open defense, we certify that we have read and evaluated the thesis prepared by Roman Girma Muluneh entitled "**Performance study of Kinematic Control of 6 DOF Robot Arm Using Fuzzy Logic Control**''. We advocated that it could be accepted as a fulfilling the thesis requirement for degree of Master of Science in Mechanical Engineering (Manufacturing System Engineering)

Dr. Mesay Alemu (Ph.D) ----------------------------- --------------------------
Name of Advisor Signature Date

Abu Feyo (MSc.) ----------------------------- --------------------------
Name of Co-Advisor Signature Date

Hana Beyene (MSc.) ----------------------------- --------------------------
Name of Chairperson Signature Date

D.K.Rao (Ph.D) ----------------------------- --------------------------
Name of Internal Examiner Signature Date

Dr. Dessalegn Wagesso (Ph.D) ---------------------------- --------------------------
Name of External Examiner Signature Date

Acknowledgment

I would like to thank first the almighty God for giving me this opportunity. Next, it is my pleasure to express the deepest appreciation his excellency to Dr. Mesay Alemu (PhD) for helping me to finalize my thesis and instructing me how to write a paper. I have learned many things since I became his student. Along with Dr. Mesay I would like to thank Mr. Abu Feyo (MSc) my Co-Advisor for his careful review, comments and suggestions with regards to complete this thesis.

I am grateful to the Faculty of Mechanical Engineering at the Jimma Institute of Technology for providing me the excellent work environment during my study. Let my acknowledgment extend to my classmates, friends, instructors and colleagues in and outside the university. I would like to appreciate all for their friendship and support during my stay at Jimma University.

Finally, I would like to thank all the instructors who increased my knowledge in many areas, and who always gave insightful responses to any questions and concerns. Last but not least, I would like to thank my husband Reta Degefa who stand beside me to complete my work.

Abstract

There are many fields for the applications of robot manipulators to execute the given tasks such as material handling, painting, welding, pick-and-place operations, undersea explorations, space manipulation, hazardous field, etc. The important role of industrial robot manipulators is to substitute labor power by doing the repetitive, dangerous and mundane tasks in different type of application area. The robotic systems design and their control methods are difficult to make zero error for complex activity like, welding electronics, inspection and medical tasks. Those tasks need high accuracy and precisions especially on medical fields like surgical operations. To achieve the desired task the position and orientation (pose) of end effector or tool must be controlled to be complete the execution of the specific task. The above stated goal to have the sound knowledge of modeling of the robot arm and to solve the control problems. The kinematic modeling is the first task to study the position and orientation of the end effector numericaly performed and see the kinematic analysis using Matlab toolbox. Then the independent joint controlling will apply for each joint actuators. The main objective of thesis concerned about Performance study of Kinematic Control of 6 DOF Robot Arm Using Fuzzy Logic Control. From existing robots type I select KUKA KR 16-2 manipulator robot as case study. The study performed using theoretical analysis including forward and inverse kinematic mathematical modeling approach of the robot based on transformation matrix , the kinematic modeling implemented by robot toolbox that work under MATLAB program, the forward and inverse kinematic simulation done by robot toolbox with MATLAB/SIMULINK environment. Then the simulation show as the pose of end effector of the robot. Design of the robot controller using fuzzy logic controller (FLC) method and the result were compared with the proportional integral derivative controller (PID) use as benchmark then the discussion will done based on the result output in terms time of response. FLC gives as better results than PID controller in terms of time response of the settling time minimum and no overshot, but PID is better result gives fast rising time than FLC.

Contents

List of Figure

List of Table

Abbreviation

3D	Three Dimension
CAD	Computer Aided Design
DC	Direct Current
DH	Denavit- Hartenberg
DOF	Degree of Freedom
EMF	Electro-Motive Force
FLC	Fuzzy Logic Control
Kd	Derivative gain
Ki	Integral gain
Kp	Proportional gain
PID	Proportional, Integral and Derivative gain control
Pose	position and orientation
PPP	Prismatic, prismatic and prismatic (Cartesian configuration)
RIA	Robot Institute of America
RVC	Robot Vision Control
RRR	Revolute, revolute and revolute (Articulated Configuration)
SISO	Single Input Single Output
XML	Extensible Markup Language

CHAPTER ONE

Introduction

1.1. Motivation

The advancements in various technological domains during the last decades have transformed dramatically. The 'fiction' robots become in reality in today's technology and the robots are considered as an integral part of industries and other areas. Robotics is a relatively young field of modern technology that crosses traditional engineering boundaries. Studying on this field is a multidisciplinary engineering and other fields that dedicated to the development of autonomous. To understand the complexity of their application requires the knowledge of computer science, electrical engineering, industrial engineering, mechanical engineering, economics, and mathematics. New disciplines of engineering, such as applications engineering, knowledge engineering, and manufacturing engineering has emerged to study the complexity of the field of robotics and factory automation. Its application is one of the interested fields in industrial, educational and medical applications, they are extensively used in the industrial manufacturing sector to perform several tasks like materials handling, welding, assembly, spray painting, and other applications [1]. Because of it works in unpredictable, hazardous place and inhospitable circumstances which human cannot reach. For example, working in chemical, basic metal foundry and nuclear reactors is very dangerous for human being, while when a robot instead of human it involves no risk to human life. Robots can be classified as industrial robot manipulators, mobile robots, and other autonomous mechanical systems [2].

There are many kind of automation applications that industrial robots are equipped to handle tasks like, material transfer, machine loading, spot welding, continuous arc welding, spray coating, material removal, cutting operations, assembling operations, part inspection, part sorting, part cleaning, part polishing, and a done more specialized tasks. Industrial robots have transformed the manufacturing industry for a reason they come with many bottom line benefits. Their first and most important benefit of robots are efficiency. Robot complete tasks more quickly than manual labors, and their uptime is significantly higher. The combination of uptime and speed leads to higher turnout at lower operating costs.

Figure 1.1. Different application of industrial robot [3]

Therefore, modeling and analysis of the robot manipulators and dealing with a control techniques are very important before using them in these circumstances to work with high precisions and accuracy. Many researchers' studies in controlling the motion and movement of industrial robot were the most interesting field in recent year. Due to advance of computer, visualization technology and the number of sensing devices the robotic manipulator controlling methods are various types. Basically, robotic manipulator modeling is divided in two categories, mathematical modeling and computer modeling of the manipulator and the actuators, which includes an analysis for the forward kinematic and the inverse kinematics. The kinematics modeling is a prerequisite for the dynamic modeling and fundamental for practical aspects like motion planning, singularity and workspace analysis of robot manipulators.

This study is focusing on Kinematic Control of 6DOF of manipulator and designing a Fuzzy logic controller (FLC) for the position controlling of the industrial robot manipulator to meet the requirement of the desired task and compare with PID. Robot manipulator is classified as a complex system due to nonlinear systems. FLC was found to be an efficient tool to control nonlinear systems.

1.2. Background

The early work leading up to today's robots began after World War II in the development of remotely controlled mechanical manipulators, developed at Argonne and Oak Ridge National Laboratories for handling radioactive material. These early mechanisms were of the master slave type, consisting of a master manipulator guided by the user through a series of moves which were then duplicated by the slave unit. Referring to Robot Institute of America (RIA) a robot is a multifunctional, reprogrammable manipulator designed to move tools, parts, material, or specialized devices through various programmed motions for the performance of a variety of tasks. According to Webster a robot is" An automatic device that performs functions normally ascribed to humans or a machine in the form of a human. Generally, robots are automatically controlled, reprogrammable and multipurpose uses in which the mechanism usually consists of a series of segments, jointed or sliding relative to one another, for the purpose of grasping and/or moving objects usually in several degrees of freedom [2, 4]. Based on this definition, the mechanical system of robot manipulators is characterized by its arm and wrist. Each of these mechanical components are essential in accomplishing the manipulation tasks, which include the movement of a work piece or tool from one point to another with required orientation, trajectory, position and velocity of end effector.

A robot manipulator consists of a kinematic chain of n+ 1 links connected by means of n joints. Joints can essentially be of two types, revolute and prismatic. Complex joints can be decomposed into these simple joints. Revolute joints are usually preferred to prismatic joints in view of their compactness and reliability. One end of the chain is connected to the base link, whereas an end effector is connected to the other end. The basic structure of a manipulator is the open kinematic chain which occurs when there is only one sequence of links connecting the two ends of the chain. Alternatively, a manipulator contains a closed kinematic chain when a sequence of links forms a loop an open-chain robot manipulator is illustrated with conventional representation of revolute and prismatic joints. Direct kinematics of a manipulator consists of determining the mapping between the joint variables and the end-effector pose with respect to some reference frame. From classical rigid body mechanics, the direct kinematics equation can be expressed in terms of the (4 x 4) homogeneous transformation matrix [5].

1.3. Problem statement

Position of end-effector of manipulator control is the fundamental task to many robotics applications in industry. A practical and mathematical model of industrial robot required many equations and consumed much time when it comes to design and experiment a real model. Today's industries in developed countries the advance controlling method they use according to the complexity of the robotic tasks is getting more and more accurate. But developing country's industries like Ethiopian industries still they didn't adopt and use the robotic technology. To be competent with worldwide industries we should adopt robotic technology and need to use an intelligent, robust and computationally easy to implement. We must be designed a controller and analyzed to optimize and maximize the performance of industrial robot. Generally, different types of robot controller give different output performance of robot movement. The performance of a robot arm can be analyzed by evaluating the movement from an initial position to a final position. Nonlinearities exist in robot controllers. Therefore, it is difficult to eliminate these undesired effects in order to achieve good stability and precise tracking control performance. Output response of linear controller would sometimes deviate from the desired inputs. Thus, a robot controller must be designed to handle both linear and nonlinear systems. The controller is used to minimize the error between the intended and the actual positions, so the controller must meet certain specifications. The purpose of this research is Performance study of Kinematic Control of 6 DOF Robot Arm Using Fuzzy Logic Control to acquire the desired end effector position by minimizing the position error in terms of time response, settling time and present of overshoot.

1.4. Objective

1.4.1 General Objective

The main objective of this thesis is to design and analysis Kinematic Control of 6 DOF (case study on KUKA KR16_2) of Robot arm using FLC.

1.4.2 Specific objectives

To achieve the above general objective, the following specific objectives will be formulating as follows:-

1. To perform numerical modeling of inverse and forward kinematics 6 Dof of Robot arm and make case study on KUKA KR16_2.
2. To design the controller using PID (proportional integral derivative) and FLC approach for KUKA KR16_2 of Robot arm.
3. To analysis on the control performance of FLC compare with PID and Make discussion on the result.

1.5. Significance of the Research

The contribution of this thesis is to use Fuzzy controller approach to control the manipulator end effector pose. The FLC controller use a special rule base designed, the number, shape and range of the memberships were chosen to achieve for the best performance. This study can be used as a document of reference for other researches that are interested in this area of robotics using FLC.

1.6. Scope of the Study

At the end of this thesis it expected that appropriate control strategy in order to eliminate the position error that expected output will be minimum settling time, less overshooting and rising time after using of controlling technique. To get the 3D CAD model of KUKA KR16_2 part were downloaded from kuka robot manufacturing website and make assemble CAD mode on Solid work software, then convert to XML file and export to MATLAB/SIMULINK. After that we make analysis on the performance of FLC and PID controller in terms of time response.

1.7. Limitation of this thesis

Based on the objectives of this thesis concern about the kinematic analysis and FLC control conduct for the 6 DOF robot arm. But this work not include the external disturbance factors when controlling the robot and the dynamic modeling do not consider in this work.

1.8. Structure of the thesis

All the thesis work process involved in the mathematical modeling, simulation on MATLAB and designing the PID an FLC controller and finally the result and discussion will be done. The work flow of document is as follows. Chapter 1 introduction and background, Chapter 2 Literature review on robotic kinematic modeling, PID and FCL control methods, Chapter 3 Fundamentals of Robots Manipulator Modeling deals with Denavit-Hartenberg(DH) method that uses four parameters, forward and inverse kinematics case study on KUKA KR16_2 Robot, then make MATLAB Simulation forward and inverse kinematic using RVC toolbox that compatible with MATLAB software, Chapter 4Robot controller design and Chapter 5 Result and Discussion, Comparison of PID and FLC method in terms of overshooting and settling time. Finally, conclusions and future work in Chapter 6.

CHAPTER TWO

Literature Review

2.1. History of robotics

The history of industrial robotics is conventionally set in the 1950's, the first generation of industrial robot spans from 1950 to 1967. The robots of this generation were basically programmable machines that did not have the ability to really control the modality of task execution; moreover, they had no communication with the external environment. With respect to the hardware, the first generation robots were provided with low-tech equipment, and servo-controllers were not present. However, the turning point for industrial robotics was due to the genius of George Devol, who designed in 1954 a "Programmable Article Transfer" Such a device was the base for the development of unmated, that is considered the first "true" industrial robot in history. The origin of the term "robot" is placed in more recent times: namely, it comes from the Czech word "robota", meaning "heavy work" or "forced labour". The introduction of this term is due to the Czech writer Karel Čapek (1890-1938), who used it for the first time in 1920 in his novel "R.U.R.: Rossum's Universal Robots". Robot in Czech is a word for worker or servant. The early work leading up to today's robots began after World War II in the development of remotely controlled mechanical manipulators, developed at Argonne and Oak Ridge National Laboratories for handling radioactive material. These early mechanisms were of the master slave type, consisting of a master manipulator guided by the user through a series of moves which were then duplicated by the slave unit. [2, 4].

Referring to Robot Institute of America (RIA) a robot is" A reprogrammable, multifunctional manipulator designed to move material, parts, tools, or specialized devices through various programmed motions for the performance of a variety of tasks. Since then the term has been applied to a great variety of mechanical devices, such as teleoperators, underwater vehicles, autonomous land rovers, etc. Virtually anything that operates with some degree of autonomy, usually under computer control, has at some point been called a robot [4, 6].

2.2. Robots classification and their mechanisms

Robot manipulators can be classified by several criteria, such as their power source (hydraulic, pneumatic or electric powered), their geometry (serial or parallel manipulator), in the application area, or in their control method. Robots have three subsystems such as the motion, recognition and control subsystem. The motion subsystem basically includes mechanical part of the robot and the electrical actuation system and electronic parts that give the motions for the robot. The recognition subsystem work as the sense organ for the robot to communicate from the environment. The last one control subsystems work as the brain of the robot to control each activity and tasks that ordered by the operator. Those criteria and subsystem that we discussed above can be determined the type and classification of industrial robot. Some of them will discuss in the next chapters and used in this study.

2.2.1. Robot Components

A robot is a machine capable of physical motion of interacting with the environment. Physical interactions include manipulation, locomotion and any other tasks changing the state of the environment or the state of the robot relative to the environment. A robot has some form of mechanisms for performing a class of tasks. A robot mechanism is a multi-body system with the multiple bodies connected together. Each body treating as rigid body, by ignoring elasticity and any deformations caused by large load conditions. Each rigid body involved in a robot mechanism is called a **link**, and a links is connected to the other link by joints these joints can be revolute that gives the rotational motion or prismatic that gives the translational motion to the manipulator. A particular kind of robot, whether Humanoid or industrial manipulator, consists of the following fundamental components. [4]

- Body: the main part of the robot that helps the transformation of motion, torque or forces. It refers to the linkages and their mechanisms.
- Effectors: This are the last components to receive motion and enables a robot to perform the desired task by interacting with the external environments.
- Actuators: An actuator or drive is a source of motion for a robot to perform its task. It can be of linear type (Cylinders) or rotary type (e.g. AC or DC servo Motors). Commonly they are classified based on the kind of medium used for energy transmission. Such as Electrical, Hydraulic, Pneumatic and Internal Combustion (IC) hybrids.
- Sensors: sensors are those components that helps to acquire a feedback action by interaction with the surrounding of the robot.
- Controller: The controllers are used to maintain the robot precision, stability, linearity and manageability within a desired appropriate limits.

2.3. Robot control method

Robots are classified by control method into servo and non-servo robots. The earliest robots were non-servo robots. These robots are essentially **open-loop devices** whose movement is limited to predetermined mechanical stops, and they are useful primarily for materials transfer. Servo robots use **closed-loop** computer control to determine their motion and are thus capable of being truly multifunctional, reprogrammable devices. Servo controlled robots are further classified according to the method that the controller uses to guide the end-effector such as point-to-point robot control and continuous path robots control. A point-to-point robot control can be taught a discrete set of points but there is no control on the path of the end-effector in between taught points. Such robots are usually taught a series of points with a teach pendant points are stored and played back. This method capable for stopping at several different programmed positions, it can be used to pick and place operations. The other one is continuous path robots control, the entire path of the end-effector can be controlled. For example, the robot end-effector can be taught to follow a straight line between two points or even to follow a contour such as a welding seam. In addition, the velocity and/or acceleration of the end-effector can often be controlled. These are the most advanced robots and require the most sophisticated computer controllers and software development [7].

2.4. Fuzzy Controller Design

Fuzzy Logic was initiated in 1965 by Lotfi, professor of computer science at the University of California

in Berkele [16]. A fuzzy system is an alternative to traditional notions of set membership and logic that has its origins in ancient Greek philosophy. 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 represents the degree of truth or the degree of falseness for each input. As an example the set of speed, the speed of 50 m/s 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 [17] 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. It has been successfully implemented and employed in a wide variety of industrial and commercial applications 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 [18]. 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. Among many control methodologies, fuzzy controller has a great consequence and a better transparency than other control techniques because of its relative simplicity and agreeable results. The basic idea of the 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 a powerful method to analysis linear systems. In case of nonlinear systems classical controllers have not produced satisfactory results due to the nonlinearities of these systems. Therefore, FLC may be an efficient tool to control these nonlinear systems. 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 [18]because the mathematical model of the system may be too complex to be described; thus, it is difficult to be controlled by 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 an idea about the characteristic system and good knowledge about controlling it. In designing a fuzzy control system, we typically express a linguistic variable as a process of inputs and outputs in terms of linguistic values such as running, jogging, and walking. As an example, some people may classify 1.5 meters as short other may be classifying it as long; hence, this relation between the speed and its values is fuzzy.

2.4.1. Fuzzy Set Theory

Fuzzy logic is a logic. Logic refers to the study of methods and principles of human reasoning. Classical logic, as common practice, deals with propositions (e.g., conclusions or decisions) that are either true or false. Each proposition has an opposite classical logic, therefore, deals with combinations of variables that represent propositions. As each variable stands for a hypothetical proposition, any combination of them eventually assumes a truth value (either true or false), but never is in between or both (i.e., is not true and false at the same time). Through this section 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 in a fuzzy set [19]. Assume ψ is a collection of all objects, members or elements of ψ denoted as u, and ψ is referred to the universe of discourse. The element you represent any element in theψ. For some objects u in ψ, a fuzzy set A in a universe of discourse ψ is defined as a set of ordered pairs u and μ_A, in which each element u of ψ take a value $\mu_A(u)$ into the interval [0,1], and it is defined as $\{ (u , (\mu_A (u)), u \; \varepsilon \; \psi \}$, where $\mu_A(u) \; \varepsilon \; [0,1]$ 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 speed between 1.0 and 2.0 meters. In crisp set the membership function, is defined as:

$$\mu_A(u) = \begin{cases} 1, \; if \; speed \in [1.0, 2.0] \\ 0, \; if \; speed \notin [1.0, 2.0] \end{cases} \tag{2.1}$$

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

$$\mu_A(u) = fn(speed) \tag{2.2}$$

In classical set theory, a set A is defined to be a subset of set B if and only if all elements of sets are contained in set B. This definition has extended to fuzzy set theory as follows: For two fuzzy sets A and B is given in the universe of discourse ψ 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:

$$\mu_A(u) \leq \mu_B(u) \tag{2.3}$$

In classical set theory, the 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 $A \subset \psi$ with membership function denoted as$\mu_A(u)$, the fuzzy set A is a null fuzzy set ϕ if:

$$\mu_A(u) = 0 \;\; \forall u \in \psi \tag{2.4}$$

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

2.4.1.1.Membership Function Features

Let see this topic by using the example about man height. Someone sees a person and say it is tall and other say it is short. There is no bounder between short and tall. 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 2 illustrates the membership function of fuzzy set short and tall.

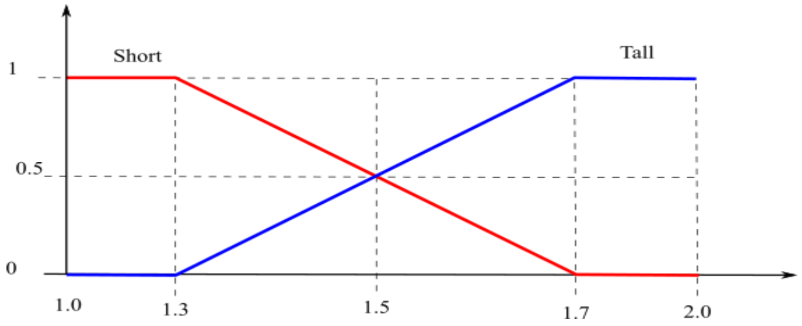


Figure 2.1. Membership function, linguistic variable

The universe of discourse ψ in the above figure 2.1 is defined as height. There are two fuzzy sets short and tall with corresponding membership functions $\mu_{\text{short t}}$ (u) and μ_{tall} (u).

2.4.1.2.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 length of a man using FL, height is the linguistic variable that takes the fuzzy sets: short and tall as shown in Figure 2.1 for the universe of course ψ defined in {1.0, 2.0}. This means that the range of the length from 1 to 2 meters. Length below 1.30 meters is interpreted as short and length above 2.0 meters is considered as tall. Each element of Figure 2.1 is defined as a mathematical function, which normalized to the output range {0, 1} by multiplying scaling factor 1/5 from the discourse ψ defined on {1.0 , 2.0}. 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 of input u in a set short increase while its membership function in a set tall decreases as the value of u moves from set short to set tall. Many papers used the notation of linguistic variable in the form: {X, T(X), U, and SX}

Where X denoted as the linguistic variable name such as speed, error, and error change, T (X) is the set of the names of the linguistic variable. In the case of speed we have, T (length is {short, tall}. S_X, gives the meaning of the linguistic variable/label such as S_X take the label tall, this meaning returns to the linguistic variable length. Finally, ψ is the universe discourse of the variable, when X takes as a crisp value [18].

2.4.1.3.Linguistic Value

Letψ_i^m denotes as a linguistic value of the linguistic variable u defined on the universe of discourseψ, then the linguistic variable u is takes on elements from the set of linguistic values are denoted $\psi_i=\{\psi_i^m\}$, where m number of the linguistic values. Linguistic values are generally expressed by terms such as Big Negative (BN), Negative Small (SN), Big Positive (BP), and Small Positive (SP). For example, assume u denotes the linguistic variable length, then the linguistic values of the length variables are ψ_i^1=short, $\psi_i^2 = tall$so that u=$\psi_i^m$where$\psi_i^m=\{\psi_i^1, \psi_i^2\}$.

2.4.1.4.Fuzzy Conditional Statement

For two inputs e and Δe and output u with a universe of discourses E, ΔE and ψ 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:

Rule1: if e is small, short and Δe is big tall, then the person is tall

Rule2: if e is big short and Δe is small tall, then the person is short

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, ΔE and ψ respectively and j =1, 2, n.

2.4.1.5.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 ψ with MFs μ_A and μ_B respectively. Fuzzy mathematics denoted as relations between the elements of A and B described using μ_{AxB} (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 belongs to A and B. The complement of A is the elements, which do not belong to A.

2.4.1.5.1. Union

The union of two fuzzy sets A and Bon the universe of discourse ψ is a fuzzy set C is described in the relation C=A∪B, and the membership function of the fuzzy set C is described as follows:

$$\mu_c(u) = \mu_{AB}(u) = \max\{\mu_A(u), \mu_B(u)\} \qquad (2.5)$$

Moreover, it’s shown as:

$$\mu_c(u) = \mu_A(u)\mu_B(u) \qquad (2.6)$$

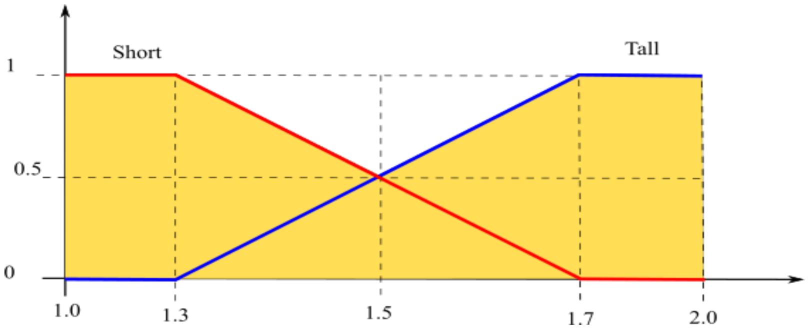


Figure 2.2. Fuzzy union

2.4.1.5.2. Intersection

The intersection of two fuzzy sets A and Bon the universe of discourse ψ is a fuzzy set C is described in the relation C=A∩B, and the membership function of the fuzzy set C is described as follows:

$$\mu_c(u) = \mu_A(u)\mu_B(u) \qquad (2.7)$$

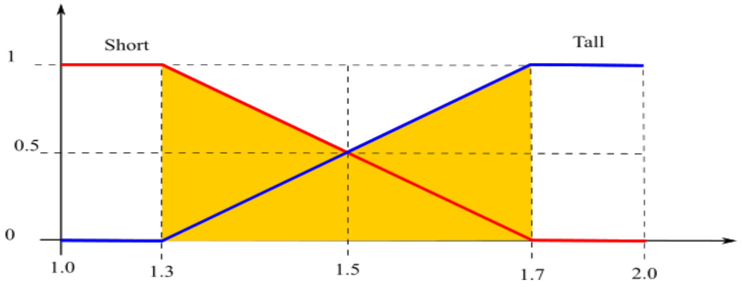


Figure 2.3. Fuzzy intersection

2.4.1.5.3. Complement

The complement of fuzzy set A is denoted by A and its membership function is defined as:

$$\mu_{\square}(u)=1-\mu_{A}(u) \tag{2.8}$$

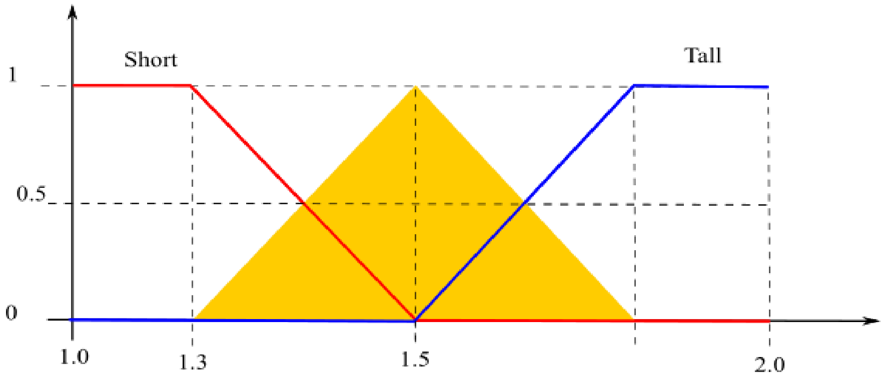


Figure 2.4. Complement

2.5. Literatures

A controller is used to modify the behavior of the physical system according to the input value through computations and actuations. Variety 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(3R) joints such as PUMA 560 and the other has PPP(3P) joints as Cartesian robot, then the control problems encountered with the RRR different from those encountered the PPP. In the field of robot manipulators, many researchers have proposed literatures and discussed the kinematics analysis of industrial robots such as SCARA, PUMA 560 and SG5-UT robot manipulator Position control performed using independent joint control in. 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 [4]. Different researchers study various type of robot control method, some of them reviewed below regarding to robot control.

Azita Azarfar [8]in his paper, a fuzzy controller with modifiable scaling factors is proposed to control the robot end-effector in task space. The controller is a fuzzy system with a mechanism to change the scaling factors when the error is bounded under a predetermined value. The controller is designed in joint space and is developed to work space by using inverse Jacobian strategy. The simulations results on Puma 560 robot manipulator illustrate the high performance of the presented control method and he get 0.04m position error and the settling time is 0.7s as his work.

Andualem Welabo [9] though in his study a Fuzzy Gain Scheduling Terminal Sliding Mode (FGSTSM) controller with a hyperbolic tangent function instead of signum function, is proposed and applied for tracking control of the UR5 robot manipulator. Hence the trajectory tracking control of the UR5 robot manipulator simulation was conducted on MATLAB/SIMULINK and simulation purposes, a curve (arc) was taken as the desired trajectory for the robot manipulator to track and finally his result is compared with the conventional sliding mode controller (SMC) controller. On the other hand, the proposed controller successfully tune the switching gain K adaptively and will help the controller handle undesired uncertainties as well as helpful for alleviating chattering.

Alim [10] present his paper on modeling a 2-R robot using mathematical modeling and Lagrange dynamic equations, then use the PID controller to validate the models and to notice the difference in accuracy achieved by each technique. But the result he gets were found using feedback linearization.

Khong [11]was applied FLC to control the position of DC servo-motor. 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 servo motor.

Song, Park in their paper presented [12] the rule-based fuzzy PID controller and the learning fuzzy PID controller are both applied to a laparoscopic surgery robot, currently developing. The learning fuzzy PID controller is realized through the design of fuzzy rules and membership functions. Through the experiments applying to the translation axis of a laparoscopic surgery robot prototype, currently developing, the learning fuzzy PID controller was compared with conventional PID controller and typical rule-based fuzzy PID controller and was proved to have better performance through the comparative analysis. Also, through all types of experiments under load and dynamic condition similar to surgery, a tracking performance of the system confirmed that is a suitable level. Accordingly, we will be going to process not only the experiments about the translation axis but also the experiments of the other axis.

Ahmed in his work presented [13] the design and performance comparison of PID and FLC strategies. Given step input signal to characterize the response, simulation results demonstrate that PID has superior performance in terms of transient parameters. In Steady state response, both PID and FLC manage to converge to the desired output but in terms of overshot FLC outperformed with zero value. Different defuzzification strategies were employed.

Alexander in his study was presented [14] on integrating the autonomous robotic arm with fuzzy logic-based joint controller (FLJC) with a machine vision system capable of accurate color discrimination into a color-based sorter system. An improved robotic arm simulator made it possible to tune the membership functions and see the actual effect on the robotic arm's response. Additionally, the end-effector is well accurate enough to have less than 2 cm absolute error. The overall accuracy of the machine vision system shows that it has the same precision as the end-effector and is at least 95% accurate in properly discriminating colored objects.

Manafaddin in his paper presented [15] Dynamic model of a 2-link robot manipulator is created and a FLC is designed and implemented in order to adjust the PD-controller parameters for improved reference-tracking performance. The system is then simulated in MATLAB/Simulink and the output curves are presented to emphasize the results. The FLC is found to be efficient in terms of adjusting Kp and Kd parameters, which in turn decreased the tracking error, and improved the reference tracking performance of the robot manipulator.

Generally, different researcher use different types of robot controller and they get different output performance of robot movement to the desire trajectory to desire position. The performance of a robot arm can be analyzed by evaluating the target value from the initial position. Therefore, on this paper we will make the controller become to reduce the error between the target and the initial positions. The controller must meet certain specifications such as reducing overshoot, minimizing rising time and reducing steady state error.

CHAPTER THREE
Numerical Modeling and Kinematics Analysis of Robot Arm

3.1. Introduction

The robot arm or manipulator is a mechanical arm constructed from rigid body links connected by means of joints. Joints are generally two types: the revolute type which gives rotational motion and the other one is prismatic type which gives translational movements. So robot arms are usually programmable and used in different application. 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 concerned with the serial manipulators. Robot manipulator consists of a collection of ***n-links*** that connected together by joints. Each one of these joints has a driving 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 movement. Links and joints form a kinematic chain connected to ground from one side, and the other is free. The mechanical structure of a manipulator is characterized by a number of degrees of mobility which uniquely determine its configuration. Each degree of mobility is typically associated with a joint articulation and constitutes a joint variable [4, 20].

At the end of the manipulator is called end-effector use for operating the required task like welding handling materials or painting and other (e.g. gripper, welding tool, or another tool). Robot manipulator is named according to number of degree of freedom, which refers to the number of joints. As an example, robot manipulator has 6 joints, which mean the robot has 6 DOF, 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 link.

3.2. Kinematic Modeling of Robot Arm

Kinematics is the branch of mechanics that deals with motion of system without considering forces and inertia. It defines the position, velocity, acceleration and higher derivatives of the variables. The kinematic studies of robot manipulator are divided into two types: the first one is called forward kinematics or direct kinematic and the second one is known as inverse kinematics. Forward kinematics determines the pose of end-effector when all the joint angles are provided. On the other hand, inverse kinematics calculates the solutions of each joint variable corresponding to a specified end-effector pose in Cartesian space. Hence, forward kinematics is defined as the transformation from joint space to Cartesian space where as inverse deals with transformation from Cartesian space to the joint space. The kinematic control problem focuses upon the computation of the joint positions required to locate the end-effector at a desired Cartesian pose. Since feedback, that requires the measurement of the Cartesian pose of the end-effector. In industrial serial robots, inverse kinematics gives a multi-solution problem.

For the representation of space movements there are several methods such as rotation matrix, vectors, quaternions, roll pitch and yaw, Euler angles, homogenous matrix. The selected method used for the developing of the direct kinematic model in this work is the homogeneous matrix Homogeneous matrices are 4×4 matrixes, which can represent by rotations matrix and translations matrix that shown blow in equation 3.1 and equation 3.2 respectively. In general, the homogeneous matrices represent linear transformations.

Rigid body motion in the three-dimensional Cartesian space comprises of translation and rotation. The pose of a rigid body with respect to the reference coordinate system is known from the six independent parameters X-Y-Z Coordinate system be the 'fixed reference frame' and U-V-W moving frame with respect to the fixed frame. Position of any point respect to fixed frame.

$$[P_F] = \begin{bmatrix} P_X \\ P_Y \\ P_Z \end{bmatrix} \tag{3.1}$$

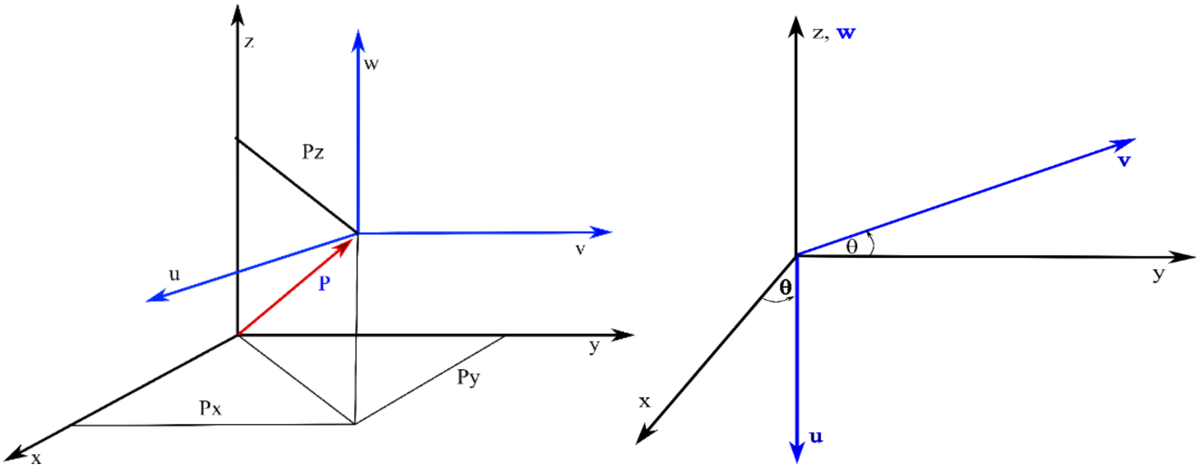


Figure 3.1. Pure translation Px, Py, Pz. And pure rotation around axis Z

$$T = \begin{bmatrix} 1 & 0 & 0 & P_X \\ 0 & 1 & 0 & P_Y \\ 0 & 0 & 1 & P_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

Rotate the fixed frame, *F*, by angle, θ, about axis Z, as indicated in Fig (3.1). This rotation is described by the rotation matrix R_Z, as derived

$$R_Z = \begin{bmatrix} C\theta_2 & -S\theta_2 & 0 & 0 \\ S\theta_2 & C\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

3.2.1. Forward kinematic

Forward kinematic to the use of the kinematic equations of a robot to compute the position and location of the end-effector in terms of the joint angles. A robot manipulators forward kinematics problem is solved by attaching a single frame to each joint along with the robot's base. Each frame describes the pose of each joint of the robot relative to the base or any other global coordinate. Attaching these frames to the joints reduces the calculation of the robot's end effector's position and orientation to a coordinate translation problem which is solved by transformation matrices. Therefore, every joint has a pose relative to its previous joint. This relationship can be derived using the Denavit Hartenberg (DH) convention, an algorithm achieving the kinematic of a chain of rigid bodies. Since each joint of robot arm has a single degree of freedom, the motion of each joint can be described by a single number, i.e. θ_1 θ_2 θ_3 the angle of rotation in the case of a revolute joint. In this model, the movements of the robot (coordinates of degrees of freedom) are given and the final positions are found. To find the direct kinematic model, using the homogeneous matrix method, is necessary to make the moves of a coordinated system from the fixed base until the last link. For each movement, the homogeneous matrices are obtained and the final result is the product of these matrices is transformation matrix. In order to have forward kinematics for a robot mechanism in a systematic manner, one should use a suitable kinematics model.

3.2.1.1. Denavit Hartenberg Representation (DH parameters)

Jacques Denavit and Richard Hartenberg introduced many of the key concepts of kinematics for serial-link manipulators method also well known as D-H convention is explained in detail is a systematic way of describing the geometry of a serial chain of links and joints. The DH parameters were originally proposed by Denavit and Hartenberg in 1955 and they put forwards to a matrix method to build the attached coordinate system on each link in the joint chains of the robot to describe the relationship of translation or rotation between the contiguous links. This method is used to define links' configuration of a robotic manipulator consisting of one degree-of-freedom joints [4]. A robot manipulator consists of several links connected by revolute or a prismatic joints.

In order to control the end-effector with respect to the base, it is necessary to find the relation between the coordinate frames attached to the end -effector and the base. In this convention the forward kinematics, defined as the relation between the individual joints that connects rigid body's (arms) of the robot manipulator and the last arm namely end effector of the robot. Or in other words, to determine of the end effector pose in terms of the joint variables such as angle for a rotational joints and link distance for prismatic joints of the robot. In this convention each homogenous transformation A_i contains four simple transformations.

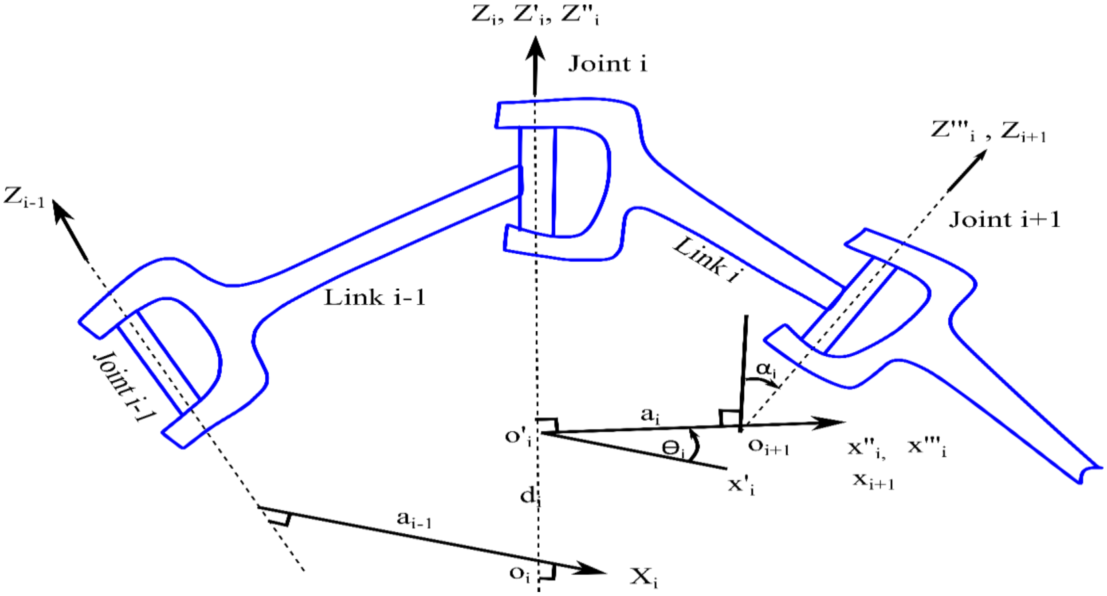


Figure 3.2.Frame convention and DH parameters for the ith revolute joint

To describe the kinematics of any robot, four parameters are given for each link twist angle (α_i), link length(a_i), joint offset (d_i)and joint angle (θ_i), where the two of them described link, and the others describe connection with other links variables as shown on table 3.1.

Table 3.1: DH parameters of six degree of freedom (6DOF) robot manipulator

Link	Joints	α_i	a_i	d_i	θ_i
1	**0-1**	α_1	a_1	d_1	θ_1
2	**1-2**	α_2	a_2	d_2	θ_2
3	**2-3**	α_3	a_3	d_3	θ_3
4	**3-4**	α_4	a_4	d_4	θ_4
5	**4-5**	α_5	a_5	d_5	θ_5
6	**5-6**	α_6	a_6	d_6	θ_6

3.2.1.2. Transformation Matrices

After obtaining the table of DH convention, a series of homogeneous matrices can be derived depending on the number of the degree of freedom. The transformation matrix for each joint from joint 1 to the joint i can be calculated as show on equation (3.3):

$$T_i = Rot(Z,\theta_i)\mathrm{Trans}(Z,d_i)Trans(X,a_i)Rot(X,\alpha_i) \tag{3.4}$$

Or in terms of the full matrices illustrate on equation 3.5 and the resultant matrix show on equation (3.6).

$$T_i = \begin{bmatrix} C\theta_i & S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_i & -S\alpha_i & 0 \\ 0 & S\alpha_i & C\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

$$T_i = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.6}$$

3.2.1.3. Homogenous Matrix Representation

Homogeneous matrices are 4×4 matrixes, which can represent rotations, translations, scales and perspectives. In general, the homogeneous matrices represent linear transformations. 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 T_1 to T_i.The general form is presented in equation:

$$\boldsymbol{H_i^0} = T_1 * T_2 * T_3 * \ldots T_i \tag{3.7}$$

The matrices from $T_1$to T_i are the transformation matrices from joint 1 to joint i and $\boldsymbol{H_i^0}$ is thelocation of the i^{th} coordinate frame with respect to the base coordinate.

3.2.1.4. The position and orientation of the end-effector.

The general homogeneous matrix for the desired pose of the end effector that DH obtained from Table (3.1) and overall homogeneous metrics can get as shown equation (3.8).

$$H_i^0 = \begin{bmatrix} r11 & r12 & r13 & Px \\ r21 & r22 & r23 & Py \\ r31 & r32 & r33 & Pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

Equation (3.8) consists of two main components: the rotation matrix and the position vector of the end-effector as follows:

$$R_d = \begin{bmatrix} r11 & r12 & r13 \\ r21 & r22 & r23 \\ r31 & r32 & r33 \end{bmatrix} \tag{3.9}$$

$$P = \begin{bmatrix} Px \\ Py \\ Pz \end{bmatrix} \tag{3.10}$$

The orientation and position of the end-effector solved directly once the homogeneous matrices for manipulator with i-links are multiplied. The rotation matrix provides orientation of frame i with respect to base frame. Solutions of the Euler angles (roll-yaw-pith) are given as:

$$\theta = \text{Atan2}\left(r33, \pm\sqrt{1-(r33)^2}\right) \tag{3.11}$$

The function θ = Atan(x, y) computes the arc tangent function, where x and y are the cosine and sine. The angle θ, if value of Sin (θ)> 0and Sin (θ) < 0 is chosen, respectively, then:

$$\beta = \text{Atan2}(r13, r23) \tag{3.12}$$

$$\gamma = \text{Atan2}(-r13, r32) \tag{3.13}$$

$$\beta = \mathrm{Atan2}\left(-r13, -r23\right) \tag{3.14}$$

$$\gamma = \mathrm{Atan2}\left(r31, -r32\right) \tag{3.15}$$

3.2.2. Inverse kinematic

This section is concerned with the invers kinematic problem to find the joint variables of the robot manipulator for a given pose of the end effector. The problem of the inverse kinematics is more difficult than the forward kinematics problem. Inverse kinematics refers to the use of the kinematics equations of a robot to determine the joint parameters that provide a desired position of the end-effector. Specification of the movement of a robot so that its end-effector achieves a desired task is known as motion planning. Inverse kinematics transforms the motion plan into joint actuator trajectories for the robot. The inverse kinematics seeks the coordinates of each degree of freedom based on the final position of the robot. There are two approaches to solve the inverse kinematics problem of a robot manipulator; mathematical or algebraic and geometrical. The higher degrees of freedom require the more complicated algebraic solution.

$$\theta_i = f_i\left(X,\ Y,\ Z,\ \alpha, \beta, \gamma\right) \tag{3.16}$$

Where $(X, Y, Z, \alpha, \beta, \gamma)$ represents the pose, i=1, 2 3 . . . i.e. and θ_i joint angles. To solve the inverse kinematics for robot manipulator the follows steps are used:

1) Equate the general transformation matrix to the final transformation matrix of the robot manipulator show on equation (3.6).
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 (3.19) 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 pose; the position is outside the robot manipulator workspace.

3.2.3. Trajectory generation

Trajectory planning is a basic thing for robotics applications and automation in different area. With high operating velocities, required in many tasks the possibility of generating trajectories which is satisfy the specific targets and requirements is a basic step to ensure optimal results. Trajectories are the time based functions defined in geometrical spaces, such as Cartesian space and joint space. A Path denotes the locus of points in the joint space, or in the Cartesian space, which is the manipulator has to follow in the execution of assigned motion. This path is a pure geometric description of motion. So, the goal of trajectory generation is to generate the reference inputs to the motion control system which ensures that the manipulator executes the planned trajectory.

Trajectory refers to a time history of position, velocity, and acceleration for each degree of freedom of the robot arm. An important characteristic of a trajectory is that it is smooth pose vary smoothly with time. These vibrations induced by non-smooth trajectories can cause damage to the actuators, as well as reduce the tracking performance of the trajectory and it make compromise the quality of the workmanship [21].

3.2.3.1. Joint-space motion

Trajectory in Joint-space motion is when the trajectory planning done in term of joint positions, velocities and accelerations which the path shapes (in space and time) in terms of function of joint angles. Typically, robot manipulators are studied from the point of view of their displacements on joint space, in other words, robot's displacements inside of its workspace are usually considered as joint displacements and for this reason the robot is analyzed in a joint space reference. Using Joint Space Trajectories has many advantages, i.e. less computation, easier to plan trajectories in real-time or no problem with singularities.

What is required is a function for each joint whose value at time zero (t_0) is the initial position of the joint and whose value at final time (t_f) is the target position of that joint. There are many smooth functions that might be used to interpolate the joint value. It is assumed that the initial and final joint positions corresponding to the initial and final configurations of the end-effector are known. Thus the motion from the initial position $q_0$to the final position q_f in $s_e$cond and $t_0$is zero, starting and ending with zero velocity and acceleration is considered.

$$q(0) = q_0, \quad q(t_f) = q_f \tag{3.17}$$

$$\dot{q}(0) = \dot{q}(t_f) = 0 \tag{3.18}$$

$$\ddot{q}(0) = \ddot{q}(t_f) = 0 \tag{3.19}$$

3.2.3.2. Cubic polynomials

Cubic polynomial function is since one has a polynomial with four independent coefficients. In the trajectory planning can be use the Point-to-point motion, that give us the path of manipulator has to move from an initial to a final joint configuration in a given time. When we need to plan the trajectory of end effector of robot arm we will consider the problem of moving the end effector of robot arm from its initial position to a target position in a certain amount of time. The inverse kinematics allow the set of joint angles that correspond to the target pose to be calculated. A cubic trajectory can be done as the following form in equation (3.20):

$$\begin{cases} q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \\ \dot{q}(t) = a_1 + 2a_2 t + 3a_3 t^2 \\ \ddot{q}(t) = 2a_2 + 6a_3 t \end{cases} \tag{3.20}$$

$$\begin{bmatrix} q_0 \\ \dot{q}_0 \\ q_f \\ \dot{q}_f \end{bmatrix} = \begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 1 & t_f & 2t_f^2 & 3t_f^3 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix} * \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{3.21}$$

$$\begin{cases} a_0 = q_0 \\ a_1 = 0 \\ a_2 = \dfrac{3}{t_f^2}(q_f - q_0) \\ a_3 = -\dfrac{2}{t_f^3}(q_f - q_0) \end{cases} \tag{3.22}$$

when the position time diagram is described by the third order polynomial function, then the velocity time diagram can be described by the second order polynomial function (at max $t = t_f$ 2) and the acceleration time diagram will be a straight line (at max $t = 0$).

3.3. Numerical Modeling of robot manipulator (for KUKA KR16_2)

KUKA KR16 _2 is a six degree of freedom an industrial robot fast and accurate robot suitable for general handling short cycle time operations, arc welding and machining. It designed by links which are connected to each other by six revolute joints. All the joints of this robot are the same revolute and there is no prismatic, cylindrical, and planar or any other type of joint in the structure of the robot, the table shows the robot specifications.

Figure 3.3. KUKA KR16_2 industrial robot [3]

Table 3.2: KUKA KR16 robot specification

KUKA KR16 robot specification	
Payload	16kg
Maximum total load	46kg
Maximum reach	1.610mm
Number of controlled axis	6
Position repeatability	±0.1000 mm
weight (excluding controller), approx.	235kg
Mounting positions	Floor, inverted, angle
Temperature during operation	+5 °C to +55 °C
Controller	KR C2 Controller &KR C4Controller
Structure	Articulated

3.3.1. Forward kinematic of KUKA KR16_2 Robot

The kinematics model is based on the use of the homogeneous matrix (translation and rotational matrix) for this purpose coordinated systems are located in a convention proposed by the authors. Supported by recommendations of the Denavit-Hartenberg algorithm method.

Table 3.3: DH parameters of KUKA KR16_2 Robot

Link	Joint angle (θ_i) [°]	Joint offset (d_i) m	Link length (a_i) m	Twist angle (α_i) [°]	θ_{min} [°]	θ_{max} [°]
1	θ_1	0.675	0.260	-90	-185	185
2	θ_2	0	0.680	0	-155	35
3	θ_3	0	-0.035	-90	-130	154
4	θ_4	0.670	0	90	-350	350
5	θ_5	0	0	-90	-130	130
6	θ_6	0.115	0	0	-350	350

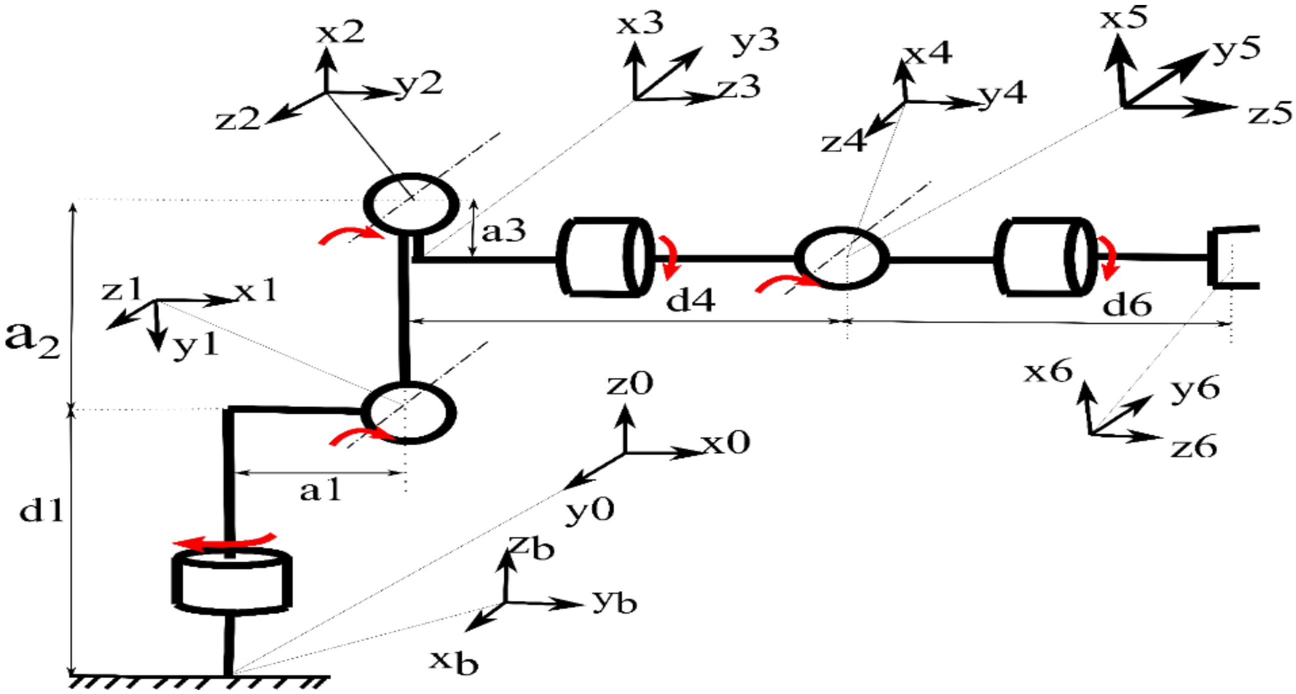


Figure 3.4.Schematic diagram of KUKA KR16_2

The transformation matrices of KUKA KR16_2 are computed in the following:

$$T_6^0 = \begin{bmatrix} C\theta_1 & 0 & -S\theta_1 & a_1C\theta_1 \\ S\theta_1 & 0 & C\theta_1 & a_1S\theta_1 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} C\theta_2 & -S\theta_2 & 0 & a_2C\theta_2 \\ S\theta_2 & C\theta_2 & 0 & a_2S\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} C\theta_3 & 0 & S\theta_3 & a_3C\theta_3 \\ S\theta_3 & 0 & -C\theta_3 & a_3S\theta_3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} *$$

$$\begin{bmatrix} C\theta_4 & 0 & -S\theta_4 & 0 \\ S\theta_4 & 0 & C\theta_4 & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} C\theta_5 & 0 & S\theta_5 & 0 \\ S\theta_5 & 0 & -C\theta_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\theta_6 & -S\theta_6 & 0 & 0 \\ S\theta_6 & C\theta_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.20}$$

The total transformation between the base of the robot and the hand is:

$$H_6^0 = \mathrm{T}_1 * \mathrm{T}_2 * \mathrm{T}_3 * \mathrm{T}_4 * \mathrm{T}_5 * \mathrm{T}_6 = T_6^0 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \mathrm{P}_X \\ r_{21} & r_{22} & r_{23} & \mathrm{P}_Y \\ r_{31} & r_{32} & r_{33} & \mathrm{P}_Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.21}$$

3.3.2. Inverse kinematic of KUKA KR16_2 Robot

The inverse kinematic analysis KUKA KR16_2 robot manipulators which have the opposite of the forward kinematic analysis. The corresponding variables of each joint could found with the given location requirement of the end effector of the manipulator in the given references coordinates system. Inverse kinematic analysis is done by multiplying each inverse matrix of T matrices on the left side of above equation and then equalizing the corresponding elements of the equal matrices of both ends. With inverse kinematic solutions, the value of each joint can be determined in order to place the arm at a desired pose. To decouple the angles, the R_{TH} matrix is routinely pre multiplied with the individual T_{n-1} matrices.

To be more precise, in case of a six-DOF manipulator a spherical wrist. First step is to find the position of the intersection of the wrist axes. The next is to find the orientation the wrist. The wrist position can be found by solving equations (3.20 - 3.21), where P_x, P_y, P_z, n_x, n_y, n_z, are elements of a translational matrix, d_6 is the length of the length of the wrist, which for Kuka KR 16-2 is equivalent to 100mm and l is the length of the tool.

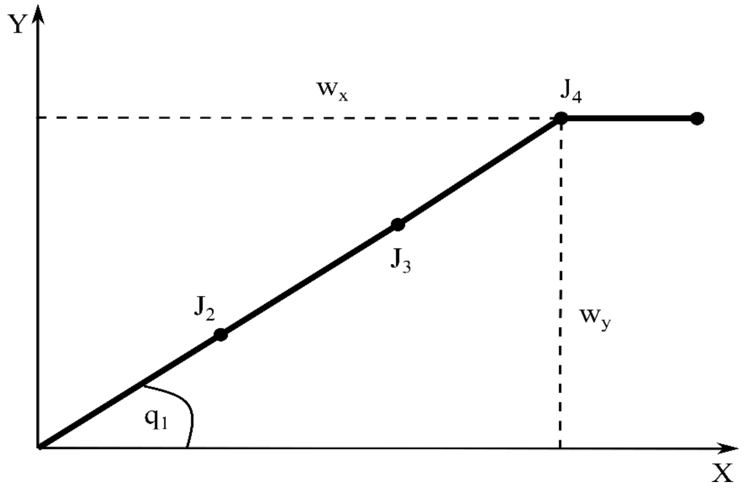


Figure 3.5.Showhow to get angle (q1) of joint one

$$w_x = P_x - (d_6 + l) * n_x \tag{3.28}$$

$$w_y = P_y - (d_6 + l) * n_y \tag{3.29}$$

$$w_z = P_z - (d_6 + l) * n_z \tag{3.30}$$

$$q_{11} = tan^{-1}\left(\frac{w_y}{w_x}\right) \tag{3.31}$$

$$q_{12} = tan^{-1}\left(\frac{w_y}{w_x}\right) + \pi \tag{3.31}$$

This gives two possible solutions for the angle of the first joint. These refer to the front or the rear arm configuration. For both solutions of angle q_1 two solutions for angles q_2 and q_3 can be obtained, when the elbow is above or below it. The values of angles q_2 and $q_3$can be found by solving equations for four configurations:

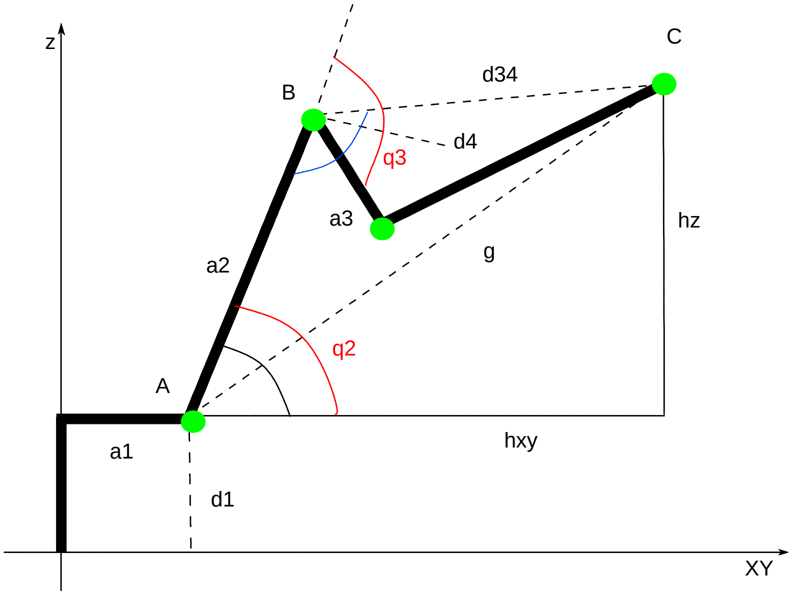


Figure 3.6.Showa front-above configuration

$$q_{21} = -(\alpha + \beta) \tag{3.32}$$

$$q_{31} = \frac{\pi}{2} - \delta + \gamma \tag{3.33}$$

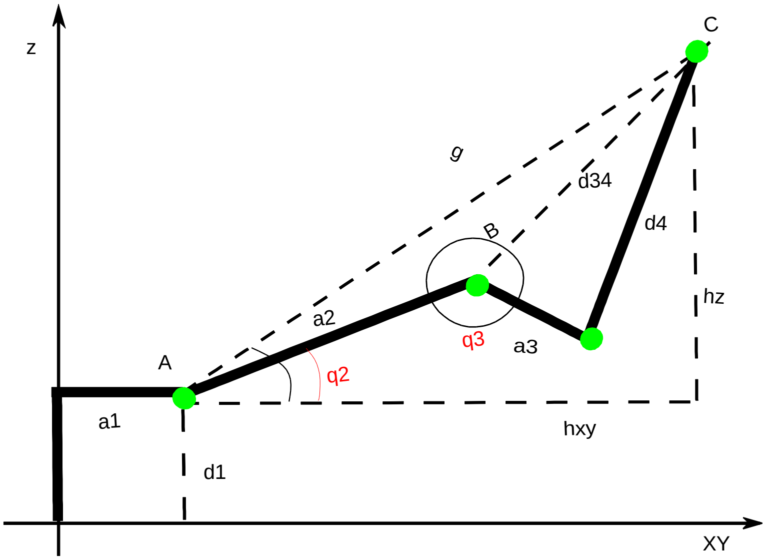


Figure 3.7. Show a front-below configuration

$$q_{22} = -(\alpha - \beta) \tag{3.34}$$

$$q_{32} = \left(\frac{3\pi}{2} - \delta - \gamma\right) \tag{3.35}$$

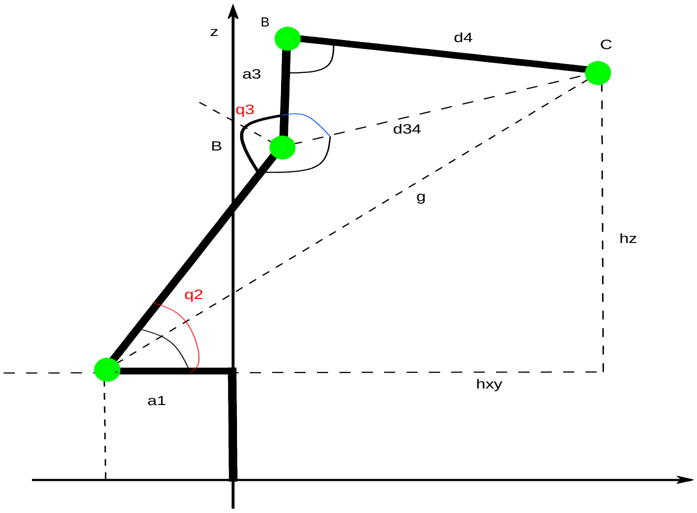


Figure 3.8. Show a rear-above configuration

$$q_{23} = -(-\pi - \alpha - \beta) \tag{3.36}$$

$$q_{33} = -\left(\frac{3\pi}{2} - \delta - \gamma\right) \tag{3.37}$$

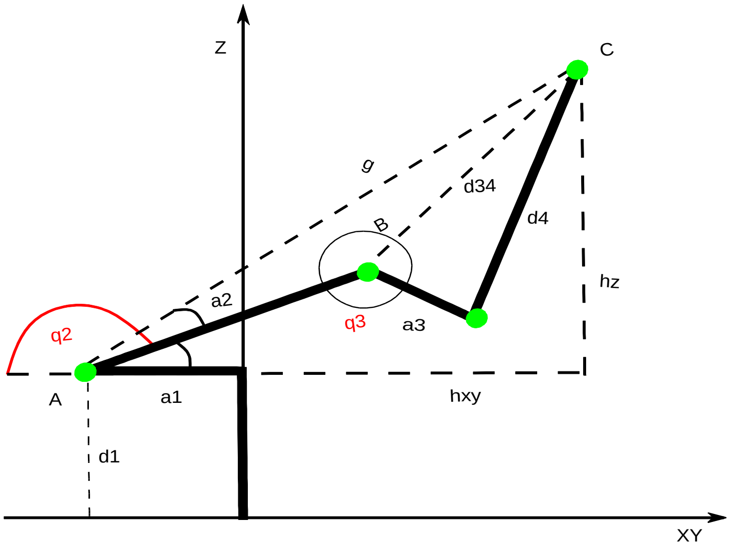


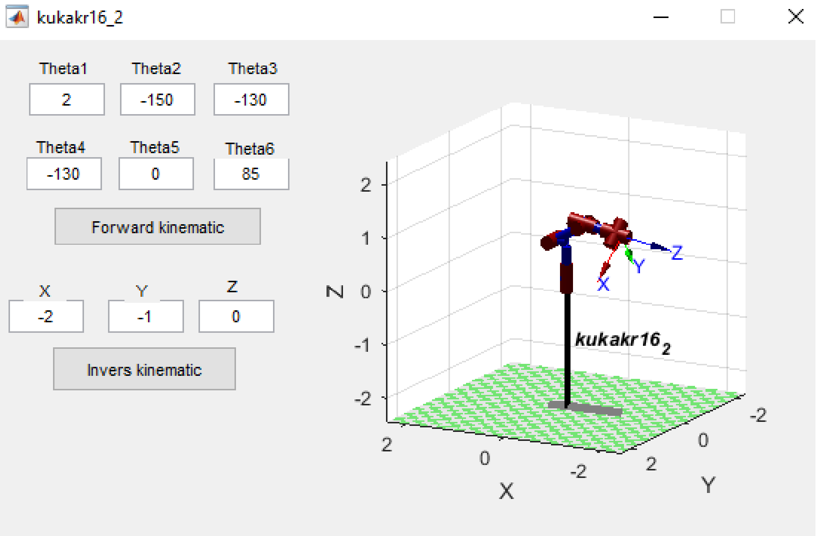
Figure 3.9. Show a rear-below configuration

$$q_{24} = -(-\pi - \alpha + \beta) \tag{3.38}$$

$$q_{34} = \frac{\pi}{2} - \delta + \gamma \tag{3.39}$$

3.3.3. MATLAB Simulation for KUKA KR16_2 Robot

Several tools are now available in robotics area, among them RVC (Peter Corke toolbox) is a MATLAB based computational toolboxes dedicated to robotics applications. This toolbox provides collection of functions (tools) that support representation and presentation of fundamental concepts in robotics such as robot configuration based on standard notations, robot kinematics, dynamics and trajectory generation, etc. in this study Peter Corke robotic toolbox will used in this study and this MATLAB Toolbox has a rich collection of functions that are useful for the study and simulation of robots.



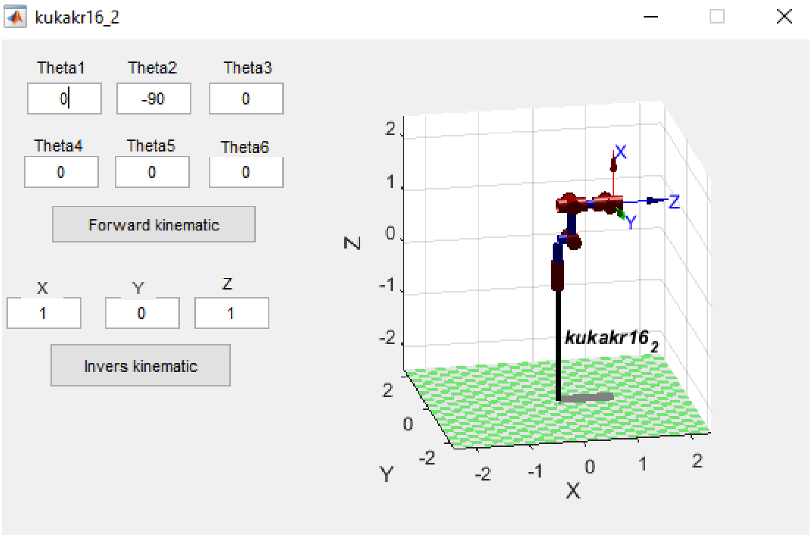
kukakr16_2

Theta1	Theta2	Theta3
2	-150	-130

Theta4	Theta5	Theta6
-130	0	85

X	Y	Z
-2	-1	0

Figure 3.10. Show robot position a



kukakr16_2

Theta1	Theta2	Theta3
0	-90	0

Theta4	Theta5	Theta6
0	0	0

X	Y	Z
1	0	1

Figure 3.11. Show robot position b

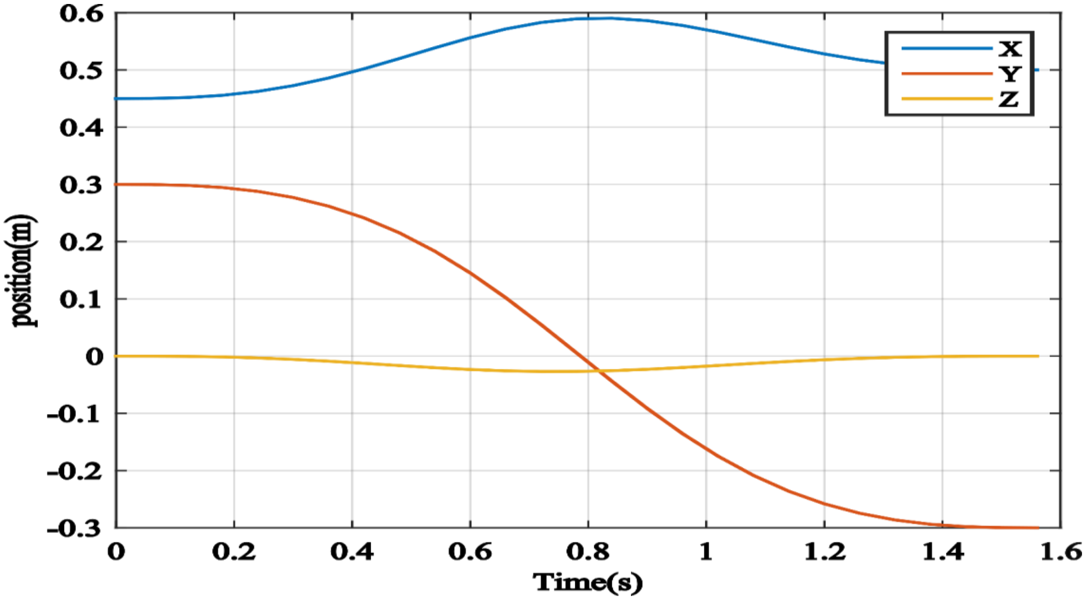


Figure 3.12. End-effector position

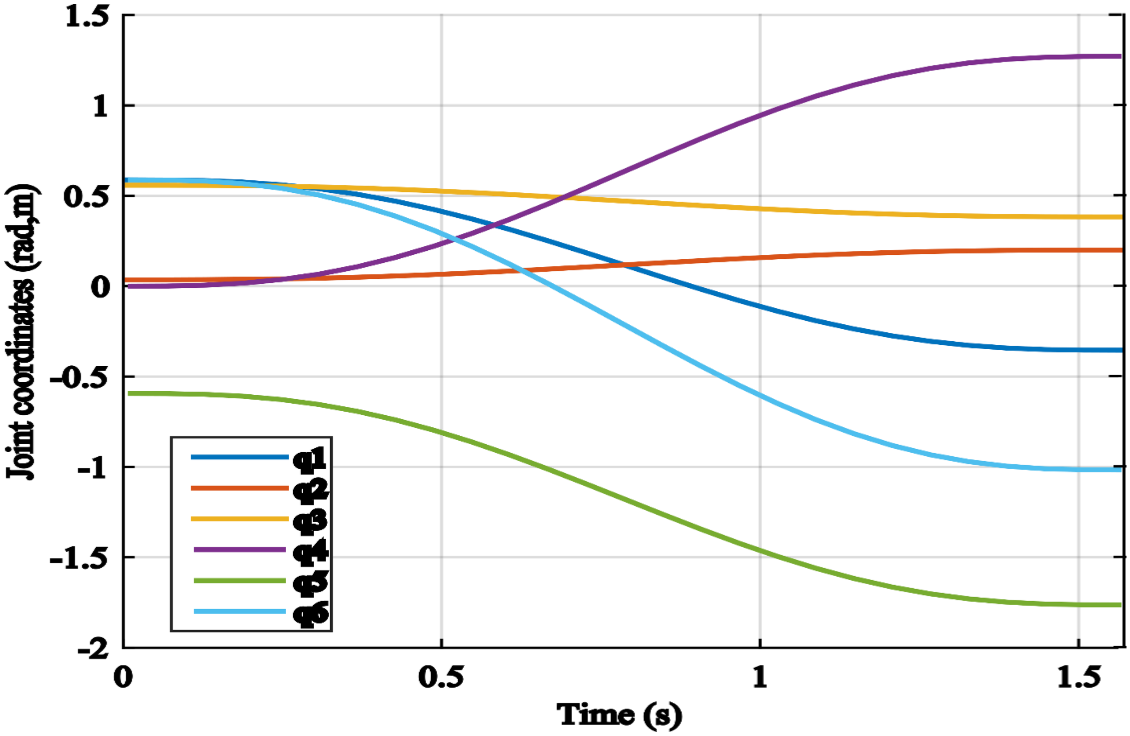


Figure 3.13. Joint position angle

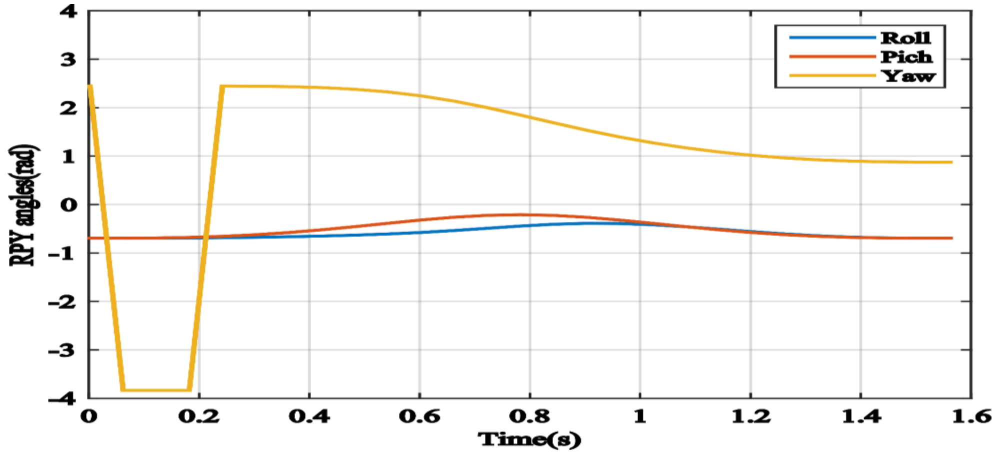


Figure 3.14. Roll, Yaw and Pitch with respect to time

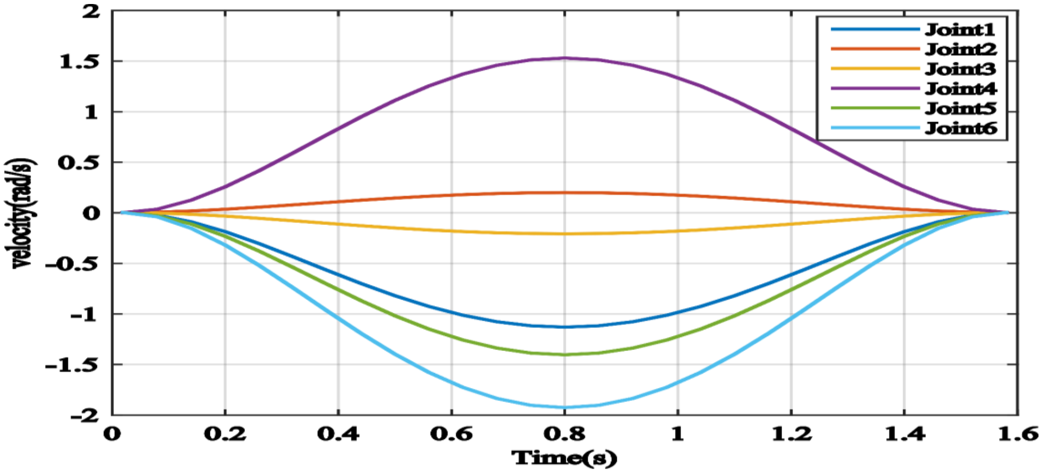


Figure 3.15.Joint angle velocity

CHAPTER FOUR

Robot Controller Design

4.1. Introduction

Mechanical system, electrical system, and control system are three main subsystems in robot manipulators [4]. The mechanical system comprises of all movable parts. It consists of a group of links connected together by joints which allow the motion for the desired link used to move the end effector to XYZ position with the reference of the base. This movement depends on the Mechanical system and electrical system which is done by some rotations and translations to the other links. We need to control and adjust robot manipulator, to control it two types of control systems are used: the open loop control system and closed loop control system. In the first control system, the controller sends a signal to the motor but does not measure the error action or feedback to the controller. On the other hand, in later one control system is the controller sends the signal to the motor, and the output signal will be sensed and returned as feedback to describe the current state of the joint. The closed loop controller has some advantages over the open loop controller such as: disturbance rejection like friction in motors, improve reference-tracking performance and stability. Control system provides some function for the plant (robot arm) such as: Providing the capability to move the robot manipulator in the surrounding environments. Secondly, it Collect information about the robot manipulator in the working place and Using this information to give a methodology to control the robot manipulator. Finally, storing the data to providing it to the robot manipulator then updating it at an instant.

4.2. Mechanical 3D model to Matlab Executed XML file

Simscape Multibody (SimMechanics) provides a multibody simulation environment for 3D mechanical systems, such as robots, vehicle suspensions, construction equipment, and aircraft landing gear. One can model multibody systems using blocks representing bodies, joints, constraints, force elements, and sensors. Simscape Multibody formulates and solves the equations of motion for the complete mechanical system. One can import complete CAD assemblies, including all masses, inertias, joints, constraints, and 3D geometry, into her model. An automatically generated 3D animation lets you visualize the system dynamics. Simscape Multibody helps you develop control systems and test system-level performance. One can parameterize his/her models using MATLAB variables and expressions, and design control systems for your multibody system in Simulink. One can integrate hydraulic, electrical, pneumatic, and other physical systems into his/her model using components from the Simscape family of products.

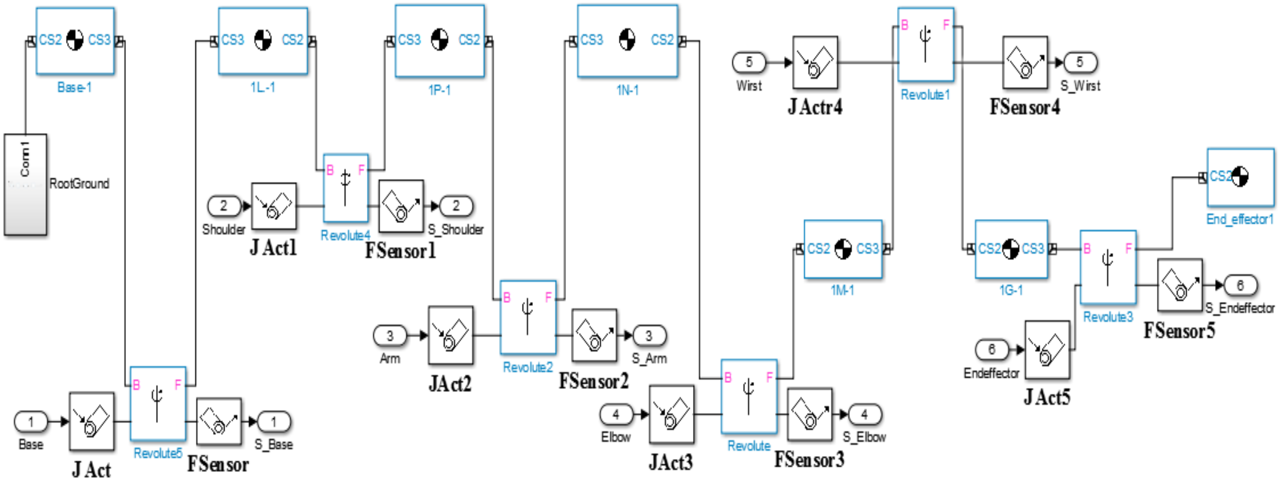


Figure 4.1. Solid work 3D-model of KUKA Kr16-2 to 3D-XML file to Simulink

4.3. Joint Actuator Modeling

Modeling refers to the plant system in mathematical terms, which characterizes the input and output relationship [22]. Independent joint control is considered as the simplest type to control the motion of the robot manipulator. In the independent joint control a 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), so each system it has independent controller. Linear control techniques are suitable to control robot manipulators with a high gear ratio. Because a high gear ratio can reduce the disturbance generated by coupling effects between joints and links. Each joint of the robot consists of two subsystems: the first is the drivers (e.g. motors and gear), and the second is the link of the robot manipulator. In the case of the DC motor as a joint actuator each motor torque τ_m influence the motor shaft or its own link. Direct current (DC) motor is a common actuator found in many mechanical systems and industrial applications such as industrial and educational robots [23]. The function of DC motor is to convert the electrical energy to mechanical energy. The motor as rotary movement, and when combined with mechanical part it can provide translation movement or the desired link. The equation (4.1) states the relation between the current and the developed torque in the motor shaft.

$$\tau_m(t) = K_m \Phi i_a(t) \tag{4.1}$$

Where, $\tau_m(t)$is motor torque produced by the motor shaft,

Φ The magnetic flux

$i_a(t)$ The armature current

K_mIs a proportional constant

$$V_b(t) = K_m \Phi \omega_m \tag{4.2}$$

Where, $V_b(t)$ is denotes the back EMF

ω_m 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 [23]. Figure 4.1 shows the schematic diagram of the armature controlled DC motor with a fixed field circuit

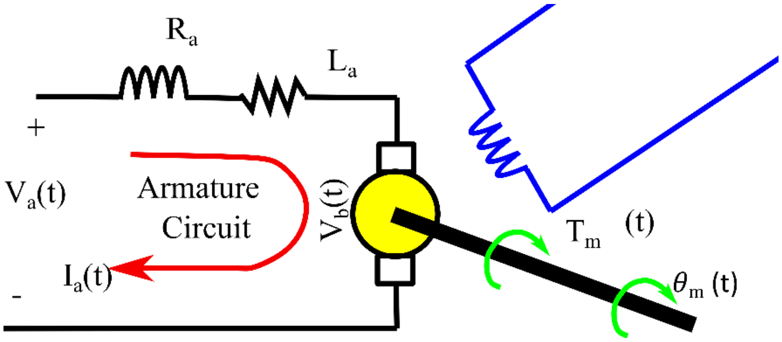


Figure 4.2. Schematic diagram of DC motor system

It is modeled as a 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 motors, the system is divided into three major components of equation: electrical equation, mechanical equation, and electro-mechanical equation [14].

The transfer function of the motor speed is

$$G_{speed}(S)=\frac{\dot{\theta}(s)}{V(s)}=\frac{K_t}{J_mL_aS^2+\left(L_aB_m+R_aJ_a\right)s+K_bK_t} \tag{4.3}$$

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

$$G_{position}(S)=\frac{\theta(s)}{V(s)}=\frac{K_t}{[J_mL_aS^2+\left(L_aB_m+R_aJ_a\right)s+K_tK_b]s} \tag{4.4}$$

Where, J_m andB_m, are denoted as the moment of inertia and motor friction Coefficient.

According to the previous discussion, the schematic diagram of DC motor system on figure 4.2 is modeled as a block diagram on Figure 4.3, this block diagram represents an open loop system, and the motor has built-in feedback EMF, which tends to reduce the current flow.

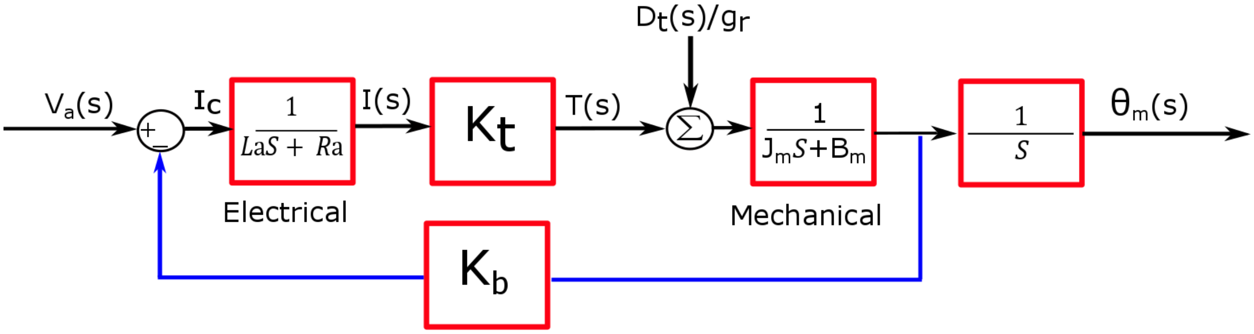


Figure 4.3. Block diagram for DC motor system

The advantage of using the block diagram is that it gives a clear picture of the transfer function relation between each block of the system. Therefore, based on the block diagram in Figure 4.3, the transfer functions from $v_a(s)$ to $\theta_m(s)$ with $D_t(s)$ =0 was illustrated in equation 4.4. As gear ratio increase the disturbance go to zero. For industrial robot this value can be ignored.

Transfer function from the load torque $v_a(s)$ to $\theta_m(s)$ could be

$$\frac{\theta(s)}{V(s)}=\frac{K_t}{[(J_ms+L_aB_m)\left(L_as+R_a\right)+K_tK_b]s} \tag{4.5}$$

Where, gr, is the gear ratio

Table 4.1: DC motor parameter and values

Parameters	Values	Units
Moment of Inertia(Jm)	0.00003	Kg.m2
Friction coefficient (Bm)	0.196	N.ms
Back EMF constant(Kb)	0.04	V/ms-1
Torque constant(Kt)	0.7	Nm/A
Electrical Resistance(Ra)	2	Ohm
Electrical Inductance(La)	0.015	H

close Loop System of the DC servo Motor

$$G(S)_{position} = \frac{\theta(s)}{V(s)} = \frac{19649}{S^2 + 201S + 4617} * \frac{1}{S} \qquad 4.6)$$

4.4. Controller Design

To be able to control a motion process, the precise position of object needs to be measurable and maintained. The designed system should respond to the applied input with a suitable overshoot a settling time and a zero steady state error as possible. The position control system is widely used in controlling applications such as robot manipulators. After analyzing the DC motor model as in previous section, the subsequent step is to design the controller to achieve the requirements.

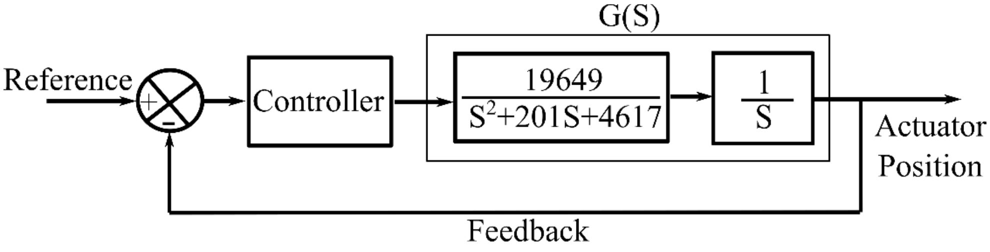


Figure 4.4. Closed loop system of Joint Controller

4.5. PID Controller

According to textbooks and dissertations that concerned with PID controller problem there are two classes for designing PID parameters [24]: 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 a 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. 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. 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 [24]. Some of the popular PID tuning methods for time and frequency domain methods are discussed below

4.5.1. 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:

a) Integral and derivative gains equal zero.
b) Proportional gain is tuned to give the desired response, neglect the steady state error.
c) Increase *Kp* gain by small increment and adjust the derivative gain *Kd* to decrease the damping.
d) Adjust the integral gain *Ki* to remove the steady state error.
e) 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.

4.5.2. Ziegler–Nichols Frequency Domain Method:

This method is based on the closed loop system response. Initially *Ki* and *Kd* gains are set to zero. The proportional gain is increased until the process oscillation occurs. It reaches the critical gain value *Kcr* 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 *Pcr* , the value of PID parameter *Kp* , *Ki* and *Kd* are given in terms of the ultimate gain and ultimate period:

$$K_p = 0.6 * k_{cr} \tag{4.7}$$

$$K_i = \frac{2 * k_p}{P_{cr}} \tag{4.8}$$

$$K_d = \frac{P_{cr} * k_p}{8} \tag{4.9}$$

4.5.3. 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. Root locus means in control theory, the location of the poles and zeros of the transfer function. Pole location determines system stability. If the roots of the 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.

4.5.4. PID Characteristic Parameters

Proportional action Kp improves the system rising time, and reduces the steady state error. This means the larger proportional gain, the larger the control signal become to correct the error as shows on appendix B table B1.1. However, the higher value of Kp produces large overshot and the system may be oscillating; therefore, integral action Ki, shown in appendix B. table B1.3, is used to eliminate the steady state error. Despite the integral control, reducing the steady state error, it may make the transient response worse. Therefore, derivative gain Kd will have the effect of increasing the damping in system, reducing the overshoot, and improving the transient response as show on appendix B. table B1.2.

Table 4.2: Kp, Ki, Kd change on system response

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

PID controller [14] uses the error between the reference input and the output as input, and then generates a control signal for the controlled system. The transfer function of the PID controller has the following form:

$$G_{pid}(s) = K_p + \frac{K_i}{s} + K_d s \tag{4.12}$$

Where K_p, K_i and K_d are the proportional, integral, and derivative gains respectively.

Using SIMULINK, the model of the motor can be created. Figure 4.5 shows the SIMULINK model of DC motor of joint actuator and the PID value is Kp = 25, Ki = 10, Kd = 1.1 for the step response of the robot joint actuator as shown on the figure 4.6 .

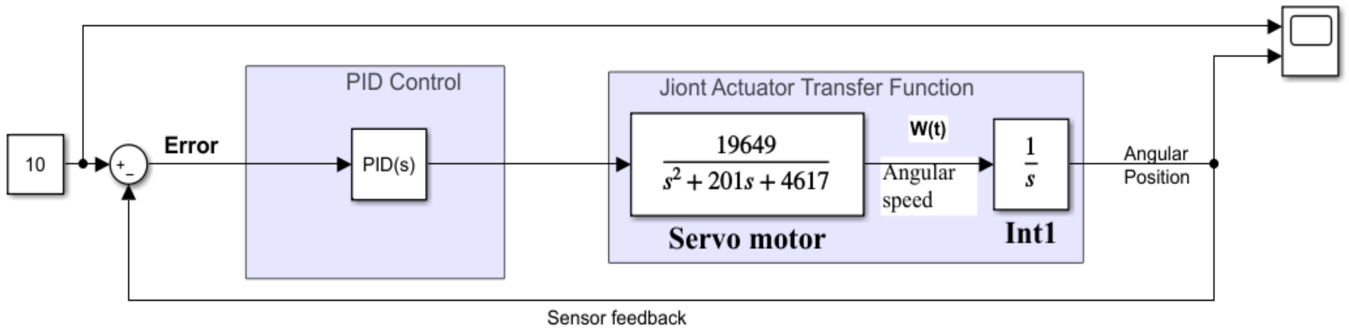


Figure 4.5. DC motor subsystem using SIMULINK.

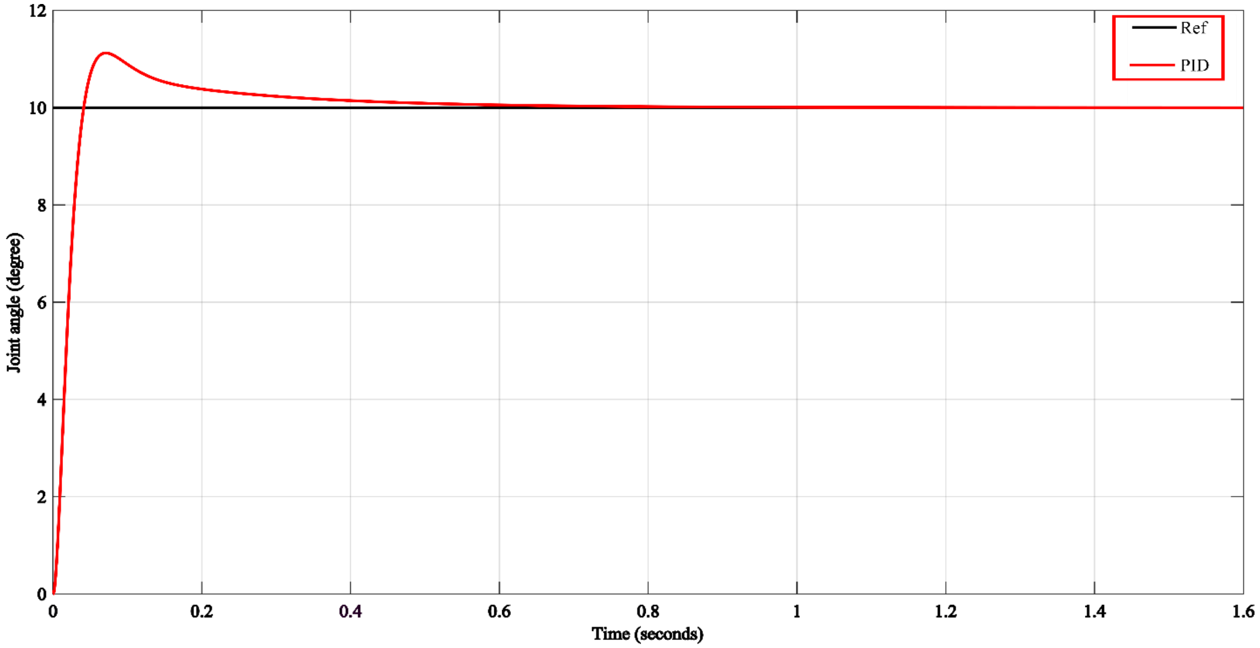


Figure 4.6. PID step response for KUKA Kr16-2 Joints

4.6. Fuzzy Logic Control Design

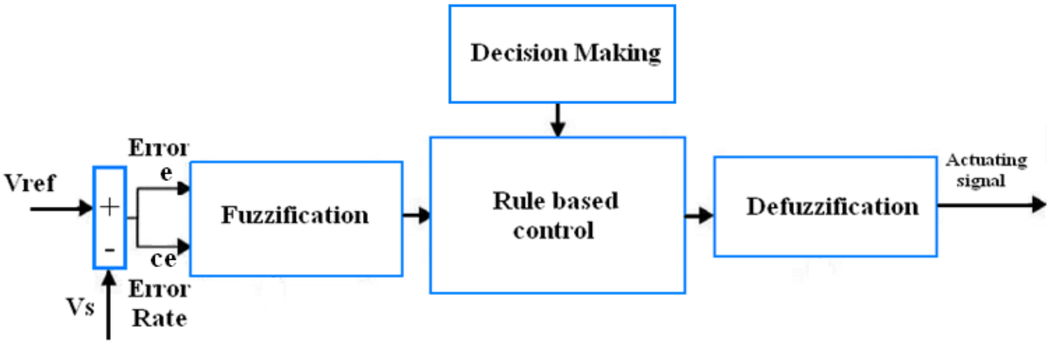


Figure 4.7.Fuzzy logic controller details

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

a. **The fuzzifier**: measure the values of input variable and convert the input crisp values into suitable linguistic variables.
b. 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.
c. **The inference mechanism** evaluates which control rules are relevant for the current time and then decides what the input to the plant should be.
d. **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.

4.6.1. Fuzzification

Fuzzification is the first block inside the controller, and process of decomposing a system input and/or output into one or more fuzzy sets. Fuzzification scale the input crisp value into a normalized universe of discourse ψ, then converts each crisp input data e and Δe to a degree of membership function μ_A (u). In other word, the purpose of fuzzifier is to transform the crisp input to fuzzy set defined in ψ and characterized by membership function $\mu_A(u)$: ψ [0,1]. The membership functions should overlap to allow smooth happing of the system. The process of fuzzification allows the system inputs and outputs to be expressed in linguistic terms so that rules can be applied in a simple manner to express a complex system.

It is labeled by linguistic terms such as short, medium, and tall. For example, if the integer value 4 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 u is chosen according to the desired output. Figure 4.8 shows several types of membership functions [25], 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).

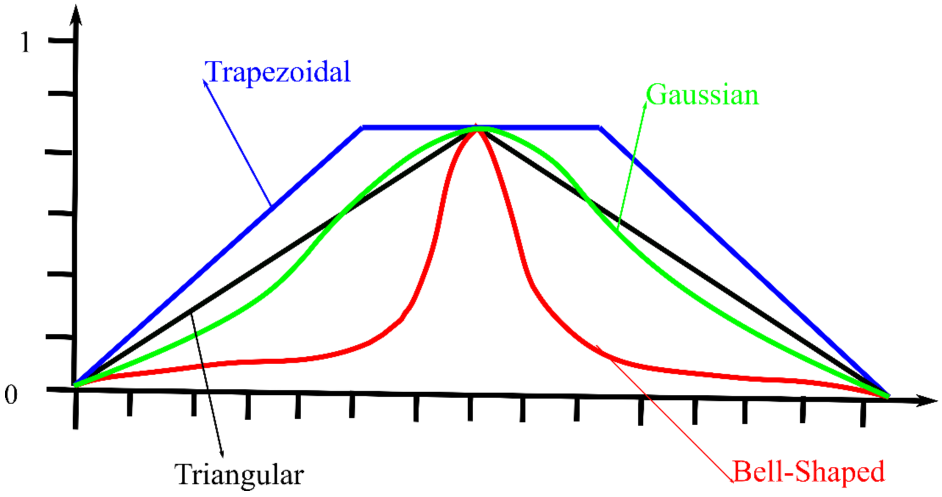


Figure 4.8. Membership functions in fuzzy set

To design FLC we should now our system, the robot in this case Kuka KR16_2 robot. This robot joints are rotational joint type, which each joint rotate on the given axis. So the maximum error (e) range is 360 degree in between [-180 180] and the maximum error rate (Δe) is also 360 degree in the range of [-180 180]. This error and rate of error input is crisp value, then converts each crisp input data e and Δe to a degree of membership function.

$$e= [-180\ 180] \quad \Delta e= [-180\ 180]$$

In this thesis, for robot arm control, we used triangular fuzzy set with seven linguistic terms for error inputs, bell-shaped fuzzy set with seven linguistic terms for error rate, and thirteen linguistic terms for output variables of FLC. The universe of discourse for the inputs e and Δe are partitioned into seven fuzzy sets and output upartitioned into thirteen fuzzy sets as shown in Figure 4.8, 4.9 and Figure 4.10 respectively. The universe of discourse ψof the error, change of error and the output are defined as e= [-180 180], Δe= [-180 180] and u= [-1800, 1800] respectively.

4.6.1.1. **Membership Functions**

For the FLC the input variables are Angle error (e) and rate of error (Δe), and the output Angle out (u). The membership functions include the linguistic terms in Table 4.6. Triangular membership functions used for input variables and the output variable. Error has seven membership functions as shown in Figure 4.9, change of error has seven membership functions as shown in Figure 4.10, and output variable has thirteen membership functions as shown in Figure 4.11.

Table 4.6: Linguistic variables and their abbreviation

Angle error/Rated error		Angle out	
Abbreviation	Linguistic Variable	Abbreviation	Linguistic Variable
-	-	VVBN	Very very big negative
-	-	VBN	Very big negative
-	-	BN	Big negative
-	-	VLN	Very large negative
LN	Large negative	LN	Large negative
MN	Medium negative	MN	Medium negative
N	Negative	N	Negative
SN	Small negative	SN	Small negative
Z	Zero	Z	Zero
SP	Small positive	SP	Small positive
P	Positive	P	Positive
MP	Medium positive	MP	Medium positive
LP	Large positive	LP	Large positive
-	-	VLN	Very large negative
-	-	BN	Big negative
-	-	VBN	Very big negative
-	-	VVBN	Very very big negative

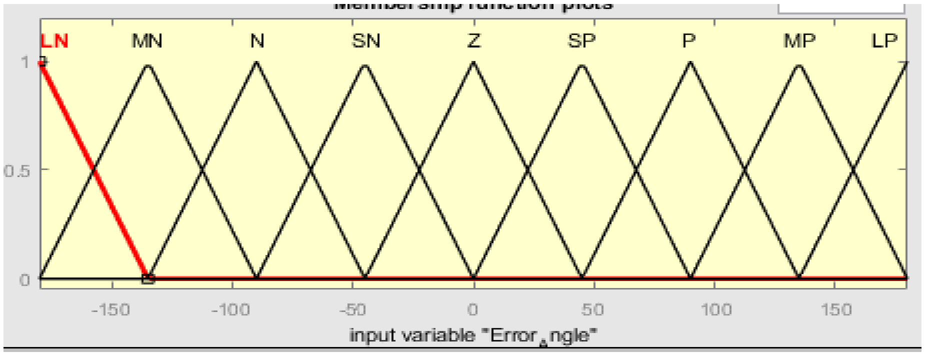


Figure 4.9. Error input Linguistic Variables

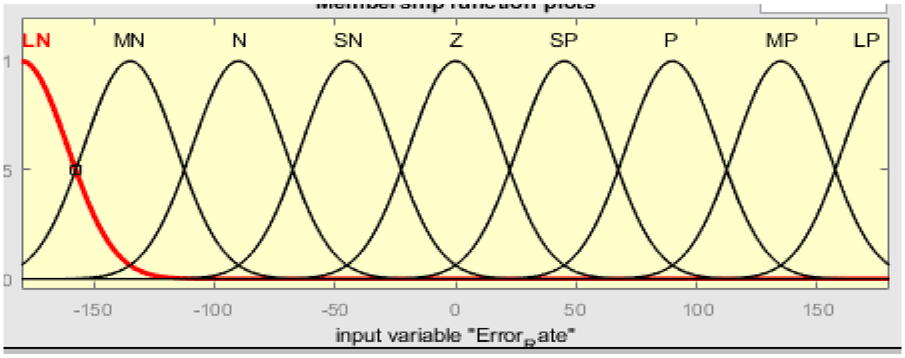


Figure 4.10.Error rate input Linguistic Variables

The output value is range between [-1800 1800] as shown below with 13 membership sets.

u= 10* [-180 180]

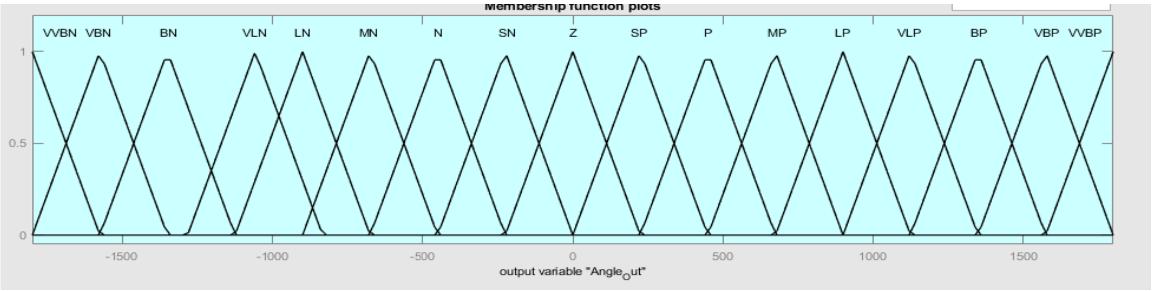
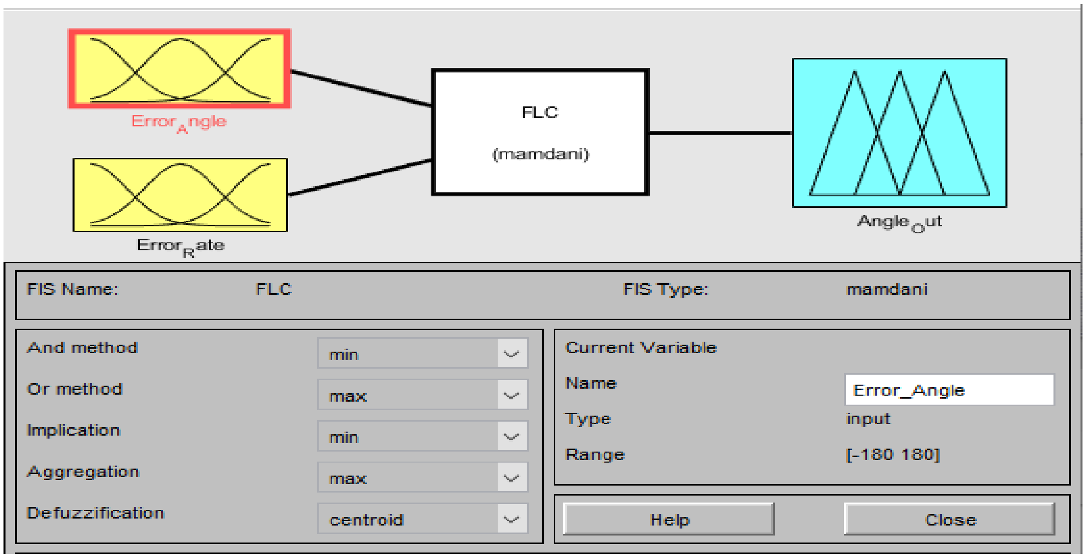


Figure 4.11.Output Linguistic Variables

Matlab software has special toolbox for manipulating FLC. The mamdani base fuzzy logic controller is selected for this thesis as shown in Figure 4.12.



FIS Name: FLC
FIS Type: mamdani

And method: min
Or method: max
Implication: min
Aggregation: max
Defuzzification: centroid

Current Variable
Name: Error_Angle
Type: input
Range: [-180 180]

Figure 4.12. Mamdani based Fuzzy logic

4.6.2. Fuzzy inference system and Fuzzy Rule Base

The fuzzy inference system has been considered the min-max method (Mamdani) and, where the implication has been assumed to min and the aggregation has been considered to max. The fuzzy input variable e has nine membership functions and fuzzy input variable Δe has nine membership functions, and the output variable has seventeen membership functions. Thus, there were total 81 rules generated as shown in table 4.7. The rule editor in Matlab used for formatting the rules using different arrangements of input variables.

Table 4.7: Fuzzy logic, if and then rules

	LN	***MN***	***N***	***SN***	***Z***	***SP***	***P***	***MP***	***LP***
LN	VVBN	VBN	*BN*	*VLN*	*LN*	*MN*	*N*	*SN*	*Z*
MN	*VBN*	*BN*	*VLN*	*LN*	*MN*	*N*	*SN*	*Z*	*SP*
N	*BN*	*VLN*	*LN*	*MN*	*N*	*SN*	*Z*	*SP*	*P*
SN	*VLN*	*LN*	*MN*	*N*	*SN*	*Z*	*SP*	*P*	*MP*
Z	*LN*	*MN*	*N*	*SN*	*Z*	*SP*	*P*	*MP*	*LP*
SP	*MN*	*N*	*SN*	*Z*	*SP*	*P*	*MP*	*LP*	*VLP*
P	*N*	*SN*	*Z*	*SP*	*P*	*MP*	*LP*	*VLP*	*BP*
MP	*SN*	*Z*	*SP*	*P*	*MP*	*LP*	*VLP*	*BP*	*VBP*
LN	*Z*	*SP*	*P*	*MP*	*LP*	*VLP*	*BP*	*VBP*	*VVBP*

Analysis of the two inputs (error and change in error) and output in Matlab rule viewer shown in (Appendix C).

4.6.3. Defuzzification

After fuzzy reasoning, we have a linguistic output variable which needs to be translated into a crisp value. The objective is to derive a single crisp numeric value that best represents the inferred fuzzy values of the linguistic output variable. Defuzzification is such inverse transformation which maps the output from the fuzzy domain back into the crisp domain. Some defuzzification methods tend to produce an integral output considering all the elements of the resulting fuzzy set with the corresponding weights. Other methods consider just the elements corresponding to the maximum points of the resulting membership functions. Defuzzification method is the final stage of the FLC. 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) [18]. 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 in this thesis. This method provides a crisp value based on the center of gravity of the fuzzy set. The total area of the membership function distribution used to represent the combined control action is divided into a number of sub-areas. The area and the center of gravity or centroid of each sub-area is calculated and then the summation of all these sub-areas are taken to find the defuzzifier value for a discrete fuzzy set. 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. The equation 4.15 is the formula for Center of Area method.

$$u = \frac{\sum_{i=1}^{m} \mu(x_i)\, x_i}{\sum_{i=1}^{m} \mu(x_i)} \tag{4.15}$$

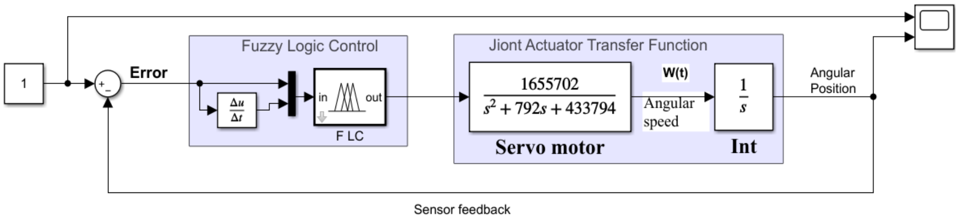


Figure 4.13. Fuzzy logic controller Simulink model

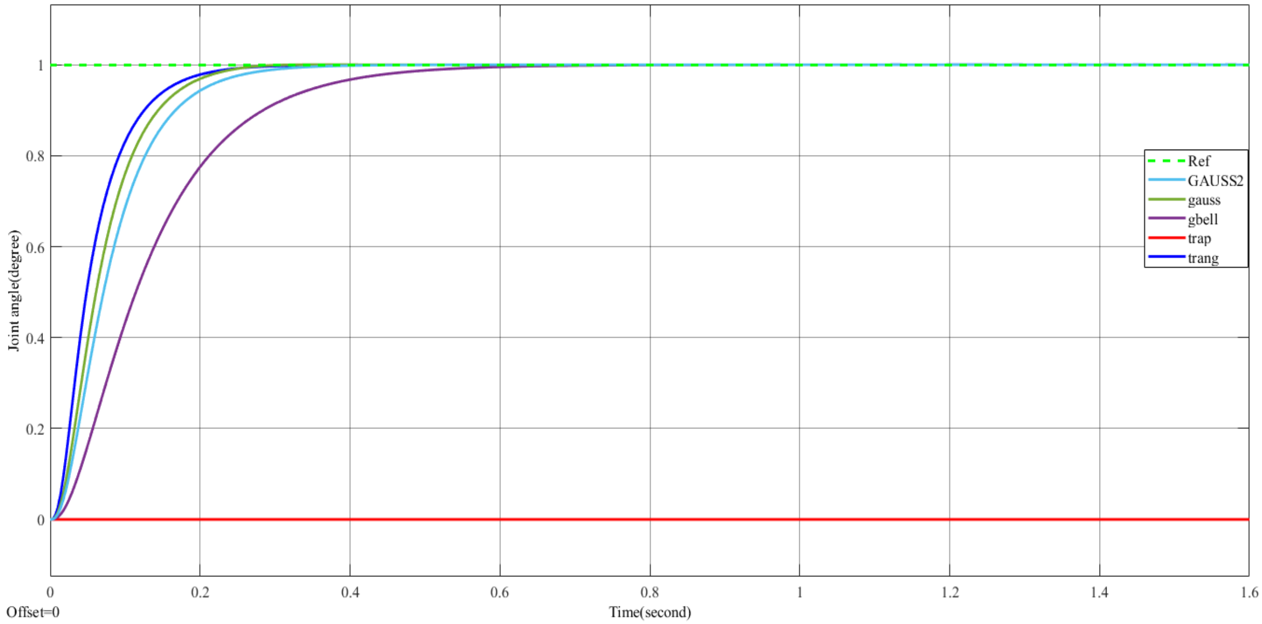


Figure 4.14. FLC step response for Kuka Kr16-2 robot joints by different membership functions

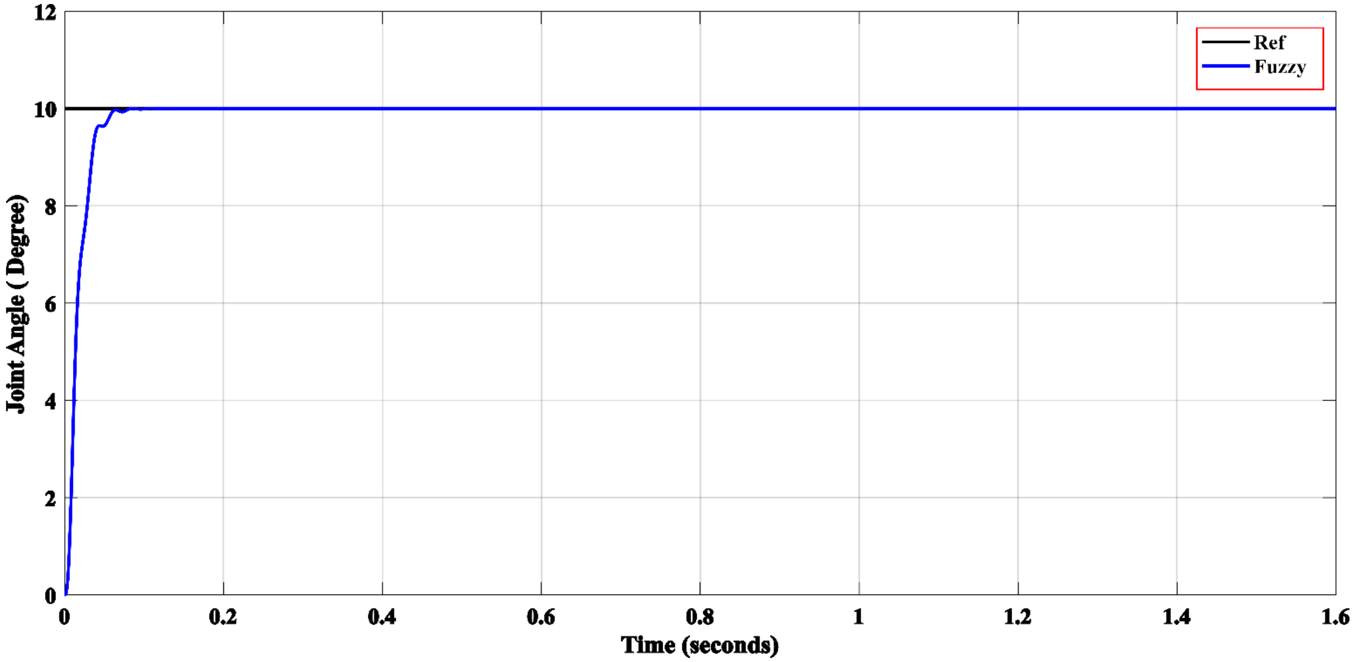


Figure 4.15. FLC step response for Kuka Kr16-2 robot joints

4.7. Matlab Simulation

After import the XML file to Matlab, Simulink environment as discussed in section 4.2. And design the actuator is as discussed in section 4.3. The next step is connecting the controller, actuator and the robot arm. To connect the external actuator for robot arm joint and to connect sensor to sense actual angular displacement we should add to the robot joint on Simulink to take the actuator and sensor by entering the value to 2 (two), (i.e. one for sensor and the other for actuator terminal) then we connect our actuator to control the angular movement of the joint of robot arm and sensor to other terminal to sense the actual value of the joint variable and compare with reference value to control. The reference is joint angle in degree which is given for each robot joints, in our case 6 reference angle each reference angle is compared with own joint sensor value. Each joint angle error is connected for controller and the controller control the joint. The figure below shows the Simulink simulation model.

The FLC is better than a PID controller by settling time and with no overshoot value. In the manufacturing process the main concern is production quality. Assume this KUKA Kr16-2 is working on cutting machine when the controller have over shoot the value you can think that the cutting is not correct or the machine could cut itself. So to avoid this overshoot we can use FLC. In the next chapter we will see the compression of FLC with PID control for the use of robot kinematics.

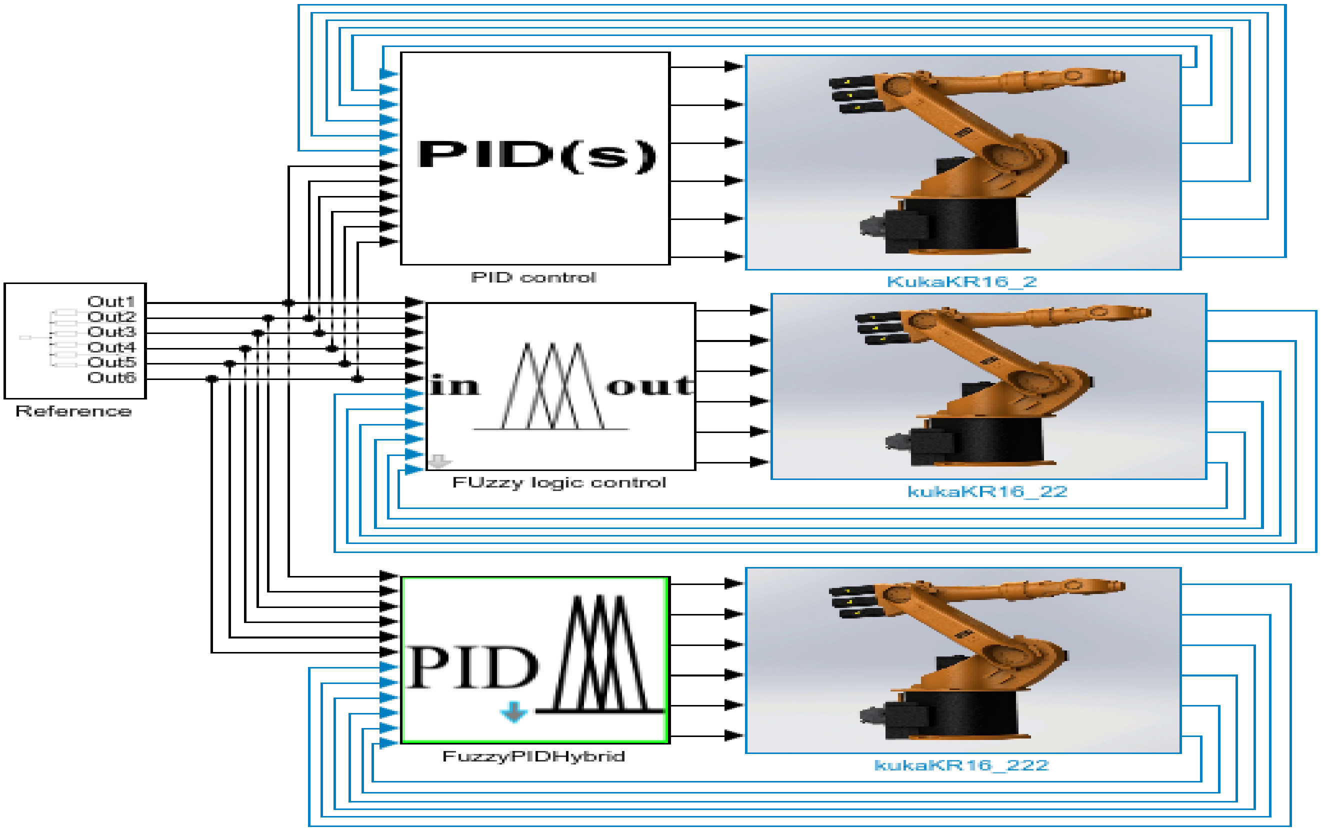


Figure 4.16. Matlab Simulink model of Kuka Kr16-2 Kinematic Control

CHAPTER FIVE

RESULT AND DISCUSSION

The robot controller was connected to robot actuators, which attached to robot joint. The PID controller have good rising time than FLC, but the drawback of PID is the overshoot. When we come to settling time the FLC is better than PID with no overshoot value.

The Figure 5.1 and 5.2 below shows the step, and sine response of PID and FLC for robot arm of KUKAKR16-2 from the Figure 5.1. We can observe that FLC have zero (0 %) overshoot value, but PID and hybrid have 11.1% and 0.4 % overshoot respectively.

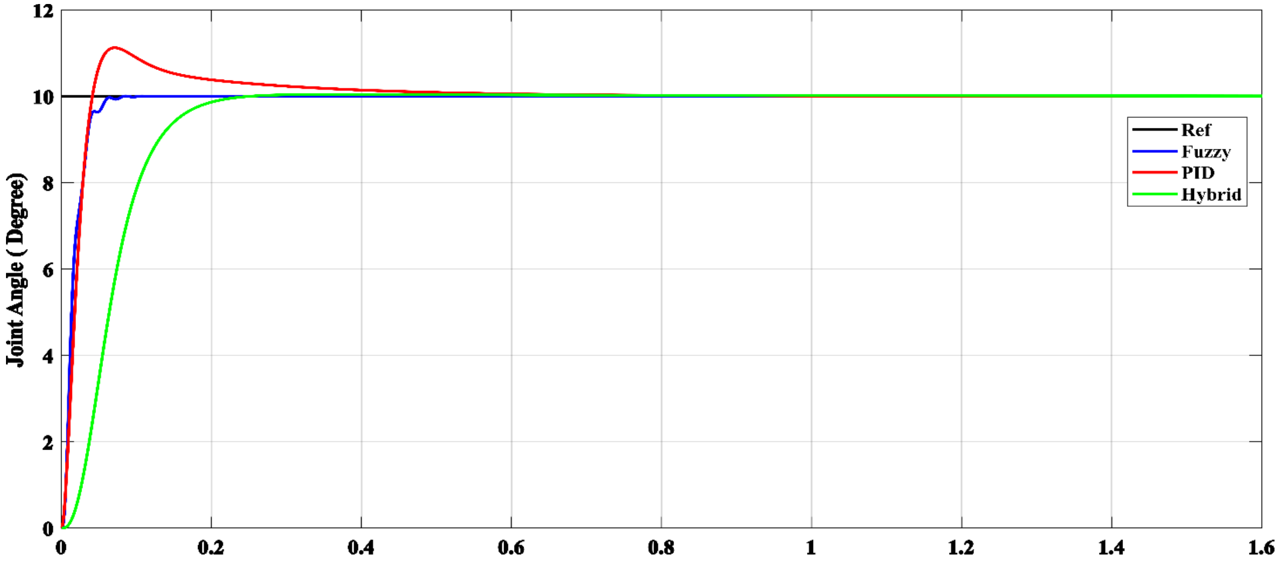


Figure 5.1. Step response of PID, FLC and Hybrid control

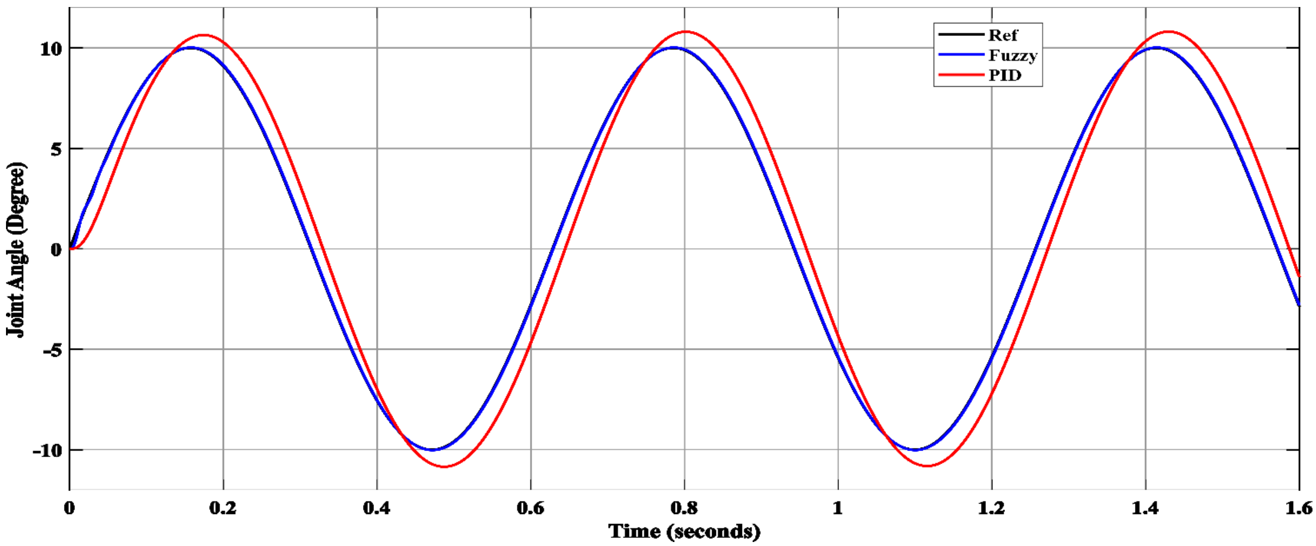


Figure 5.2.Sine response of PID and FLC

5.1. Comparison of FLC and PID Controller of robot

Table 5.1: Comparison of FLC and PID controller of robot

Controller type	System Characteristics		
	Rise time(t_r)	Settling time(t_s)	Overshot (M_p%)
PID	0.0278	0.332	11.1
Fuzzy	0.0284	0.0587	0
Hybrid	0.1035	0.187	0.4

The above comparison result can be observed on the 3 D Matlab animation. Figure 5.3. Shows that when the controller start at time t=0, PID, FLC and Fuzzy- PID hybrid controller value is equal and the robot is at home position.

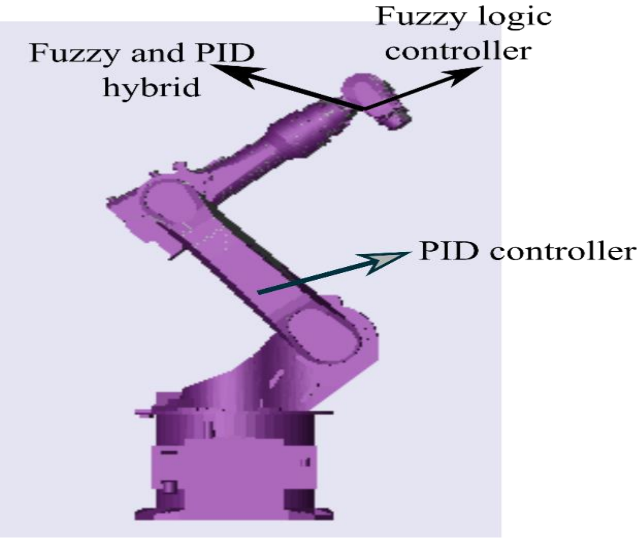


Figure 5.3. When t=0, the KR16-2 is at home position.

When the step signal command given to the robot actuator the fuzzy controller trigger faster than both PID and hybrid controllers, then after PID pass all controllers to reach the target. On the figure 5.4 and figure 5.6 the figure shows the rising time. In this case PID run faster than Fuzzy to reach the target position and FLC robot arm follow the PID controlled robot with small gaps. But PID passes the target position and move beyond the target. Figure 5.4. t=0.08 the PID controlled robot pass the target On the figure 5.6 the robot arm animation shows the settling step, fuzzy logic controlled robot arm reach the target, but PID controlled robot try to reach back to the target from overshot position and the hybrid controller is follow the fuzzy controller to reach the target.

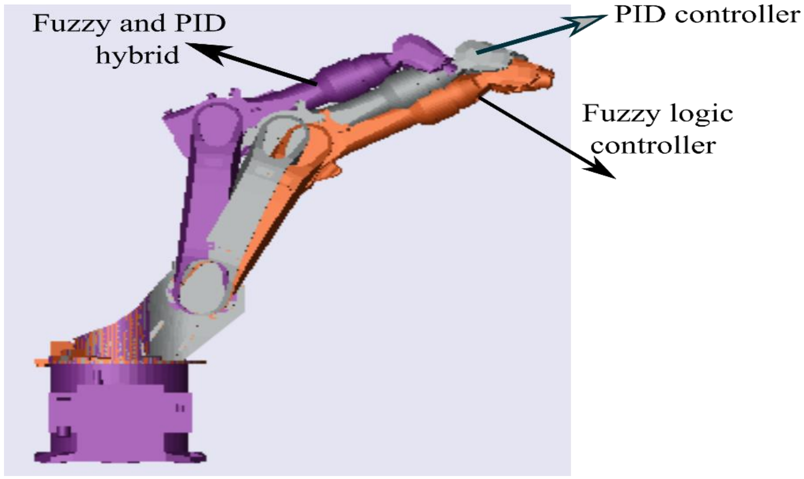


Figure 5.4. Rising of the controller response for step input.

In this figure 5.5 it can observed that fuzzy is faster than PID to reach a target position and settle.

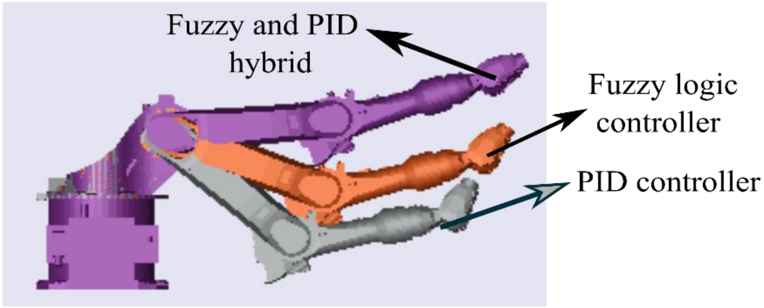


Figure 5.5. t=0.0587 second the fuzzy logic controlled robot arm reach the target

Figure 5.6 show the fuzzy-PID hybrid controller were reach the target faster than PID controller and at this point all controllers reach the target

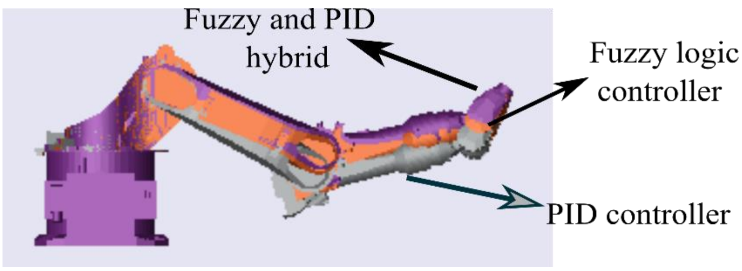


Figure 5.6. t=2 the PID controller robot arm arrive the target

CHAPTER FIVE

Conclusion and Future Work

5.1. Conclusion

Robotics has become recently an interesting area of research. The objective of this paper is to control the 6DOF kuka KR16_2 robot arm to reach the specified location with minimum error while meeting certain specification and this paper presented the kinematic modeling of the robot arm manipulator, the forward and inverse kinematic analysis of 6 DOF robot arm (Kuka KR16_2 as case study). The parts downloaded from kuka website and assembling in SolidWorks 2018 was done then export to Matlab Simulink environment for the purpose of controlling the joint angle. The mathematical modeling approach only gives the forward kinematics values and inverse kinematics values from the motion analysis result in this thesis, the exported xml file from solid Work data is controlled and analyzed in MATLAB 2018a, then fuzzy controller is applied to robot manipulator .

The model allows studying the performance of robot arm system from home position to target position. It also presented a comparative study between three different control algorithms, fuzzy logic, PID and hybrid controller on the given step input signal to characterize the response time to control the robot arm the best performance and to get smooth motion. The proposed controller designed for the robotic model is done through a computer simulation for the arm movement. While designing FLC, the choice of the membership function, their spacing, and rule base play an important role to control the system effectively.

The simulation results demonstrate that PID has better performance in terms of rising time but in terms of overshot percentage FLC and hybrid controller outperformed with zero overshot value. That the fuzzy logic control is better than the two controls where it has a no overshot and good stealing time, generally the fuzzy logic control is better than PID and hybrid controller by removing overshoot and good settling time which is necessary for industrial robot.

5.2. Recommedation and future work

In this thesis we focus on kinematic modeling of robot manipulator and designing a controller by using fuzzy logic controller (FLC) for the position controlling and compared with the proportional integral derivative (PID) controlling method. But still remains unaddressed issues are there that will conduct for in further work. To attain a complete solution for the kinematic modeling of a robotic manipulator using MATLAB or other software that help to simulate and making analysis for robot manipulator, and also it is possibility to change robot controlling method to get better result.

The following areas can be carried out for its future work:

- The future and extension of this work can be model own robot model design for n degree of freedom and then make experimental validation of the controller.
- Dynamic modelling, workspace analysis and apply it on specific industrial operation like, spray painting, welding, material handling and other task.
- Improve the robot controlling mechanism by using Newro-Fuzzy controller.

References

[1] J. Lavín-Delgado, E. Solís-Pérez, "Trajectory tracking control based on non-singular," *Multibody System Dynamics,* vol. 50, pp. 259-303, 16 June 2020.

[2] A. Gasparetto, L. Scalera, "A Brief History of Industrial Robotics in the 20th Century," *Advances in Historical Studies,* vol. 8, pp. 24-35, 15 February 2019.

[3] K. ROBORTER, "Kuka," KUKA, 26 May 2015. [Online]. Available: www.kuka-robotics.com. [Accessed 15 December 2020].

[4] M. Spong, S. Hutchinson , Robot Dynamics and Control 2nd Edition, Jon Wiley & Sons, 2005.

[5] M.Murray, Z. Li, A Mathematical Introduction to Robotic Manipulation, CRC Press, 1994.

[6] D. Todd, Fundamentals of robot technology, London: Kogan Page Ltd, 1986.

[7] F. L.Lewis, Robot Manipulator Control Theory and Practice, NEW YORK: CRC Press, 2003.

[8] A. Azarfar, "Self-Tuning Fuzzy Task Space Controller for Puma 560 Robot," *Journal of Electrical Engineering & Technology,* 5 October 2020.

[9] A. Welabo, "Trajectory Tracking Control of UR5 Robot Manipulator Using Fuzzy Gain Scheduling Terminal Sliding Mode Controller," *Journal of Mechatronics and Robotics,* 22 July 2020.

[10] M. Mustafa , A AL-SAIF, Modeling, Simulation and Control of 2-R Robot, vol. 14, Global Journals of Research In Engineering , 2014.

[11] N. Khong, A. Kanch , Control of the Position Dc Servo Motor by Fuzzy Logic, vol. 3, IEEE Proceedings of TENCON, sep 2000.

[12] K. Sun, S. Song, Comparative Study of Fuzzy PID Control Algorithm for Position Control, Seoul, 2006.

[13] Kharidege A. , Jianbiao D., "Performance Study of PID and Fuzzy Controllers for Position Control of 6 DOF arm Manipulator with Various Deffuzzification Strategies," in *MATEC Web of Conferences*, Beijing, 2016.

[14] C. Abad, D. Ligutan, Fuzzy Logic-Controlled 6-DOF Robotic Arm Color-based Sorter with Machine Vision Feedback, vol. 9, Manila: International Journal of Advanced Computer Science and Applications, Nov. 5, 2018.

[15] N. Manafaddin, "Fuzzy Logic Control Design for 2-Link Robot Manipulator in MATLAB/Simulink via Robotics Toolbox," *IEEE,* 2018.

[16] L. Zadeh, "Fuzzy Sets," Vols. Vol. 8, , pp. 335–353, 1965.

[17] E. Mamdani, "Applications of fuzzy logic to approximate reasoning using linguistic synthesis," Vols. 26, pp.1182–1191, no. 12, Dec. 1977.

[18] M.Kevin, Y. Stephen, Fuzzy Control, California: Addison Wesley Longman, Inc., 1998.

[19] A.Ebrahimnejad J. Luis, Fuzzy Sets Theory Preliminary, J. Kacprzyk, Ed., Cham, Switzerland: springer, 2018.

[20] N. Hazim , K. Jármai, Optimization for Robot Modelling with MATLAB, Cham: Springer, 2020.

[21] J.John, in *Introduction to Robotics,* Third ed., A. Dworkin, Ed., Pearson Education, 2005.

[22] X. Wang, X. Chen, "Forward Kinematics Analysis and 3-Dimmision Gait Simulation of a MiniQuad Walking Robot," *IEEE International Conference on Mechatronics and Automation,* 2007.

[23] Z. Zhao, M. Tomizuka, "Fuzzy Gain Scheduling PID Controller," *IEEE Transactions on Systems,Man., Cybernetics Society,* vol. 23, october 1993.

[24] D. Xue, Y. Chen, Linear Feedback Control Analysis and Design with MATLAB, Philadelphia: Society for Industrial and Applied Mathematics(SIAM), 2007.

[25] C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part I," vol. 20, p. 404–418, March/April 1990.

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

[27] A. Zadeh, "Fuzzy Sets," *Information and Control,* vol. 8, p. pp. 335–353, 1965.

[28] R. Samin, M. Ramli, "Performance Comparison of PD-type Fuzzy Logic Controller of USBM Simplified Model,” pp. 217–221," 2008.

[29] V. Kumar, A. Mittal,, "Parallel Fuzzy P + Fuzzy I + Fuzzy D controller: Design and Performance Evaluation," 2009.

Appendix A

Appendix A1: Transformation matrices numerical result

$$ {}^{0}_{1}T = \begin{bmatrix} -0.9962 & 0 & 0.0872 & -0.2591 \\ -0.0872 & 0 & 0.9962 & -0.0227 \\ 0 & -1 & 0 & 0.675 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^{1}_{2}T = \begin{bmatrix} 0.8192 & -0.5736 & 0 & 0.5570 \\ 0.5736 & 0.8192 & 0 & 0.3900 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{2}_{3}T = \begin{bmatrix} -0.8988 & 0 & -0.4384 & 0.0314 \\ 0.4384 & 0 & -0.8988 & -0.0153 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^{3}_{4}T = \begin{bmatrix} 0.9848 & 0 & -0.1736 & 0 \\ -0.1736 & 0 & -0.9848 & 0 \\ 0 & 1 & 0 & 0.67 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{4}_{5}T = \begin{bmatrix} -0.6428 & 0 & -0.7660 & 0 \\ 0.7660 & 0 & -0.6428 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^{5}_{6}T = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.115 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{0}_{6}T = \begin{bmatrix} -0.0850 & 0.7519 & -0.6537 & -1.0279 \\ 0.9960 & -0.0463 & 0.0763 & -0.0746 \\ 0.0272 & -0.6576 & -0.7529 & 0.8547 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Appendix A2: Forward and inverse kinematics of KUKA KR16-2 with trajectory.

```
% create links using D-H parameters of KUKA KR16_2 L = Link ([theta, d, a, alpha]
L(1) = Link([0  0.675  0.265  -pi/2]); L(2) = Link([0  0  0.680  0]); L(3) = Link([0  0  -0.035  -pi/2]);
L(4) = Link([0  0.670  0  pi/2]);L(5) = Link([0  0  0  -pi/2]);L(6) = Link([0  0.115  0  0]);
% Joint limits
L(1).qlim = pi/180*[-185 185];L(2).qlim = pi/180*[-155 35];L(3).qlim = pi/180*[130 154];
L(4).qlim = pi/180*[-350 350];L(5).qlim = pi/180*[-130 130];L(6).qlim = pi/180*[-350 350];
KR16_2 = SerialLink(L);KR16_2.name = 'KR16_2';KR16_2;
% when the end-effector moving between two cartesian poses(pose A to pose B)
T1 = SE3(0.45, 0.3, 0) * SE3.Rx(pi);T2 = SE3(0.5, -0.3, 0) * SE3.Rx(pi/2);
q1 = KR16_2.ikine(T1);q2 = KR16_2.ikine(T2);
t = [0:0.06:2];
q = mtraj(@tpoly, q1, q2, t);
q = mtraj(@lspb, q1, q2, t);
q = jtraj(q1, q2, t);
[q,qd,qdd] = jtraj(q1, q2, t);
KR16_2.fkine(q).print('xyz')
KR16_2.plot(q)
figure,qplot(t, q);% Plot the all joint angle versus time
% Determine how the robot’s end-effector will move in joint space by applying forward kinematics to the joint
coordinate trajectory
T = KR16_2.fkine(q);
p = T.transl;about(p) % The translational part of this trajectory
% The translation and orientation of end-effector, in XYZ roll-pitch-yaw angle form,can be plotted versus time
figure,plot(t, T.transl),xlabel('Time(s)'), ylabel('position(m)'),grid, legend('X','Y','Z'), title('End-effector
Position');figure,plot(t, T.torpy('xyz')),xlabel('Time(s)'), ylabel('RPY angles(rad)'),grid,
legend('Roll','Pich','Yaw'), title('Roll-Pich-Yaw angles versus time');figure,plot(t, qd),xlabel('Time(s)'),
ylabel('velocity(rad/s)'),grid,legend('Joint1','Joint2','Joint3','Joint4','Joint5','Joint6'),
title('Joint velocity');figure,plot(t, qdd),xlabel('Time(s)'), ylabel('accelratio(nrad/s2)'),grid,
legend('Joint1','Joint2','Joint3','Joint4','Joint5','Joint6'), title('Joint acceleration');
```

Appendix B

Appendix B1: PID data

\Closed loop Model with PID by considering the Kp=25, Ki=10 and Kd =1.1 can be obtaind as below

$$\frac{2.161e04s^{2}+491225s+196490}{s^{4}+201s^{3}+2.623e04s^{2}+491225s+196490} \tag{4.14}$$

Table B1.1: Kp change on system response

Closed-Loop Response(Kp)	Rise Time	Overshoot	Settling Time
0	∞		∞
5	0.0618	18.1	0.708
10	0.0357	9.75	0.378
15	0.0276	11.8	0.324
20	0.0236	18.1	0.22
25	0.021	24.2	0.0947
30	0.0192	30	0.174

Table B1.2: Kd change on system response

Closed-Loop Response(Kd)	Rise Time	Overshoot	Settling Time
0	0.0226	77	0.88
0.4	0.021	24.1	0.0946
0.8	0.0159	9.39	0.13
1	0.0138	7.74	0.0437
1.05	0.0133	7.65	0.0408
1.1	0.0129	7.65	0.0385
2	0.00828	13.3	0.0711
3	0.00613	20.7	0.0885
4	0.00508	26.6	0.0967
5	0.00439	31.3	0.096

Table B1.3: Ki change on system response

Closed-Loop Response(Ki)25/1.1	Rise Time	Overshoot	Settling Time
0	0.013	7.16	0.0522
10	0.0129	7.25	0.0372
20	0.0129	7.33	0.0372
30	0.0129	7.42	0.0377
40	0.0129	7.51	0.038
50	0.0129	7.59	0.0383

Appendix B2: Fuzzy Rule results Data

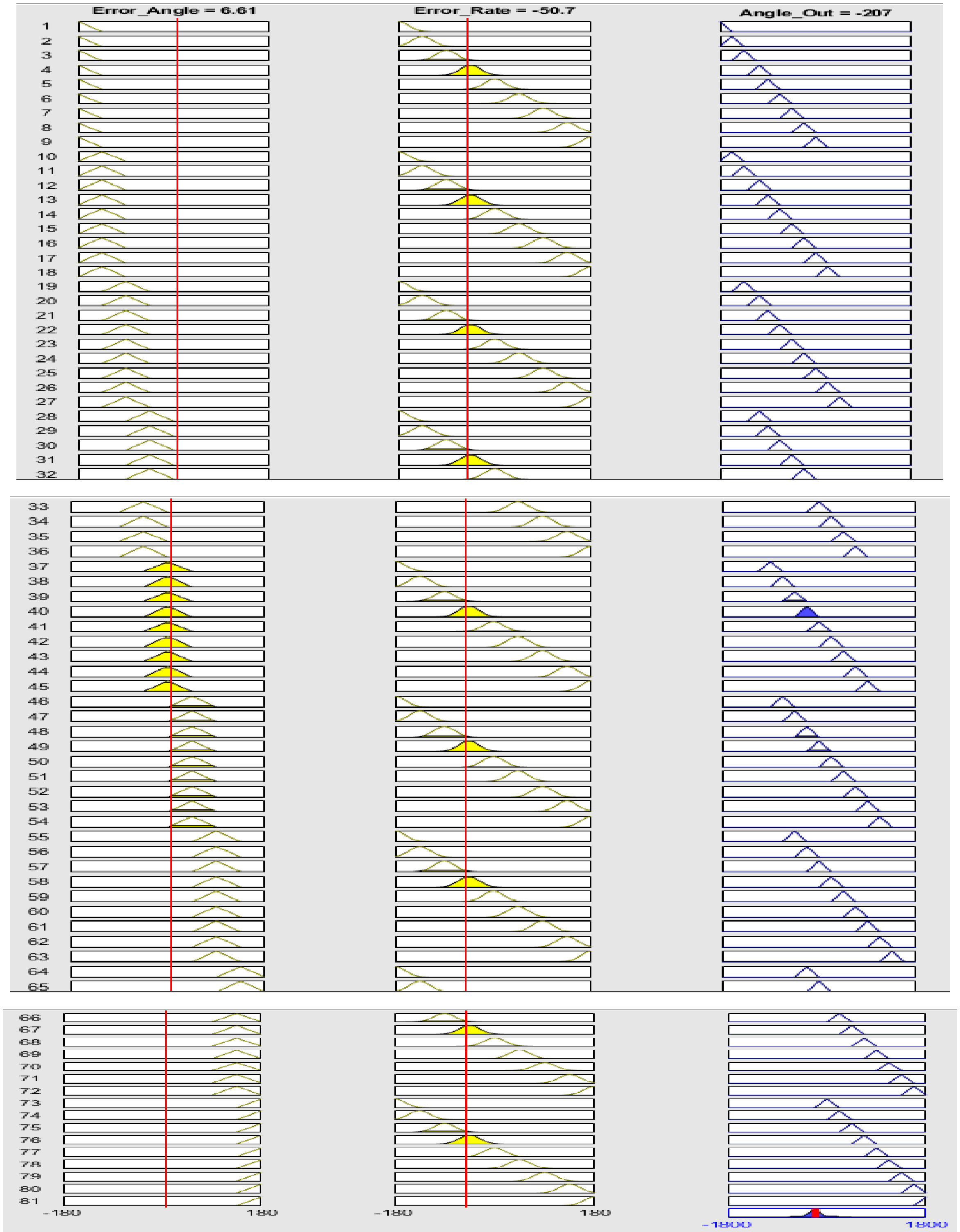
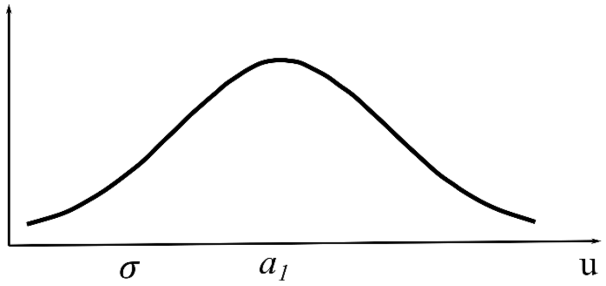


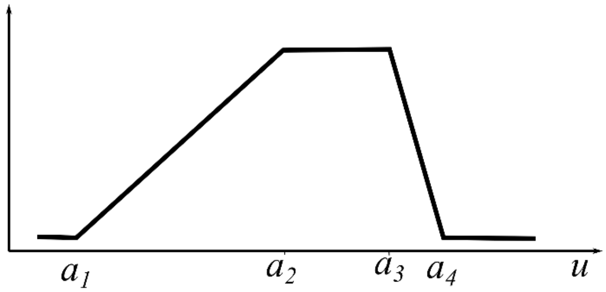
Figure B2.1. FLC step response for Kuka Kr16-2 robot joints

Appendix B3: Membership Function Types

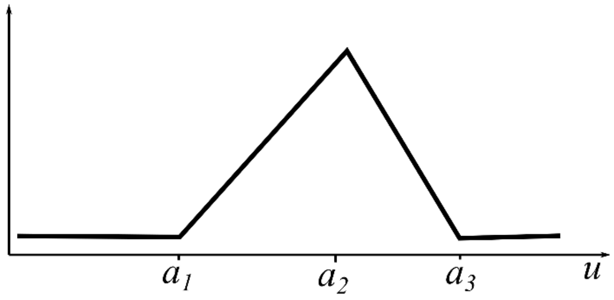


$$\mu\ (\mathrm{u}) = \mathrm{e}^{\frac{-(\mathrm{u}-a)}{2\sigma^2}}$$

where the mean a_1 and variance σ



$$\mu\ (\mathrm{u}) = \begin{cases} 0 & if\ u \le a_1 \\ \frac{\mathrm{u-a1}}{a_2 - a_1} & if\ \ a_1 \le u \le a_2 \\ 1 & \mathrm{if}\ \ \mathrm{a}_2 \le \mathrm{u} \le \mathrm{a}_3 \\ \frac{\mathrm{a}_3\mathrm{-u}}{a_3 - a_2} & \mathrm{if}\ \ \mathrm{a}_3 \le \mathrm{u} \le \mathrm{a}_4 \\ 0 & if\ u \ge a_4 \end{cases}$$



$$\mu\ (\mathrm{u}) = \begin{cases} 0 & if\ u \le a_1 \\ \frac{\mathrm{u-a1}}{a_2 - a_1} & if\ \ a_1 \le u \le a_2 \\ \frac{\mathrm{a}_3\mathrm{-u}}{a_3 - a_2} & \mathrm{if}\ \ \mathrm{a}_2 \le \mathrm{u} \le \mathrm{a}_3 \\ 0 & if\ u \ge a_1 \end{cases}$$