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

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

Trajectory Tracking Control of 3-DOF Robot Manipulator Using TSK Fuzzy Controller

By

Mahmoud A. Abualsebah

Advisor

Dr. Hatem Elaydi

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

August 2013

شوال ١٤٣٤

ABSTRACT

Robots are used for various jobs such as dangerous and repetitive jobs that are boring, stressful, or labor-intensive for humans, like cleaning the main circulating pump housing in the nuclear power plant. The subject of this thesis is to presents an implementation of fuzzy modeling methodology for controlling robot manipulator using TSK fuzzy controller. In this thesis, the control method depends mainly on mathematical modeling, analysis and synthesis. The mathematical model of robot based on the Euler-Lagrange formalism represents the main tool for analysis and synthesis of robot control algorithms. Deriving both forward and inverse kinematics is an important step in robot modeling based on Denavit Hartenberg (DH) representation. The control objective is to make the 3-DOF robot manipulator traces desired trajectory using TSK fuzzy model. Computer simulation results shows that the robot tracks the path accurately with very small tracking error when compared to some of previous studies.

ملخص

تستخدم الروبوتات في العديد من الوظائف، مثل المهمات الخطيرة، والمتكررة التي قد تكون مملة ومُجهدة، أو يصعب على الإنسان انجازها، مثل تنظيف المضخات الرئيسية في المحطات النووية.

موضوع هذا العمل هو تقديم تطبيق للتحكم الغامض(FLC)، بهدف التحكم بروبوت باستخدام المتحكم (TSK).

في هذه الرسالة، فإن عملية التحكم تعتمد على اشتقاق وتحليل النموذج الرياضي، وهذا النموذج يعتمد على طريقة (Euler-Lagrange) كطريقة أساسية لتحليل واشتقاق نظام التحكم المناسب لهذا الروبوت.

أيضاً يعتبر اشتقاق الحركة الأمامية والحركة الخلفية للروبوت من الخطوات الأساسية للتحكم في الروبوت.

إن الهدف من التحكم، هو جعل روبوت من ذوات الثلاثة مفاصل يتتبع مساراً معلوماً باستخدام المتحكم الغامض من نوع (TSK).

واستناداً إلى نتائج المحاكاة، فإن الروبوت تتبع المسار بشكل دقيق بنسبة خطأ صغيرة جداًعند المقارنة مع بعض الدراسات السابقة.

DEDICATION

Dedicated to

My parents who have been my constant source of inspiration

My brothers and sisters

My friends

ACKNOWLEDGEMENT

First of all, I would like to thank ALLAH for his unlimited bestowal, my peace and blessings be upon the prophet Mohammed.

I express my thanks to Dr. Hatem Elaydi who gave me the golden opportunity to do this project. Also, I would like to thank the other committee members Dr. Baael Hamed and Dr. Iyad Abu-hadrous for providing their valuable advices and suggestions.

Deepest respects and thanks for the president of the Islamic University of Gaza Dr. Kamalain Shaath and vice president of academic affairs Prof. Dr. Mohammed Shabat for his support. I am very thankful and so greatly appreciate.

Also, I would like to thank the dean of postgraduate studies Prof. Fouad Al-Ajez, the dean of faculty of engineering Prof. Shafik Jendia, the previous vice dean of faculty of engineering Prof. Dr. Mohammed Hussien, and the previous head of electrical engineering department Dr. Ammar Abu-hadrous.

It is my sincere feeling to place on record my best regards, deepest sense of gratitude to my parents Dr. Atallah Abualsebah and my mother for their praying and unlimited support.

Finally, my thanks go to my friends and colleagues، especially Engineer Adel and Engineer Ahmed Alassar.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

$a_{i-1}, \alpha_{i-1}, d_i, \theta_i$	Denavit Hartenberg parameters
T_i^{i-1}	Homogeneous transformation matrix of i relative to *i-1*
θ_k	Invese kinematic
L	Lagrangian
K	Kinetaic Energy
P	Potential Energy
Γ	External Forces and Torques
m	Mass
V_i	Velocity
q_i	Joint Variable
$\dot{q}$	Vector of angular velocity
$\ddot{q}$	Vector of angular acceleration
U_{ij}	The movement effect of joint *j* on the segment
U_{ijk}	The velocity intersection effect
Q_i	Pre-multiplication matrix
v	Linear velocity vectors
w	Angular velocity vectors
I	Inertia Tensor
$J(a)$	The Jacobian matrix
$[D+J](q)$	Inertia Matrix
$h(q,\dot{q})$	Vector of centrifugal and coriolis
$f(\dot{q})$	Vector of friction coefficients
g	Vector of Gravity
r_{ci}	The Coordinate of the center of mass of link i

c_{ijk}	Christofell symbols
$\mu_A(x)$	Membership function for a fuzzy set A
e	Error
Δe	Change of error
G_i	Gain
J_1, J_2, J_3	Inertia for the three motors
f_1, f_2, f_3	Viscous friction for the three motors
f_4, f_5, f_6	Dry friction for the three motors
N_1, N_2, N_3	Reduction ration for the three motors

ABBRIVAITONS

AFNNC	Adaptive Fuzzy-Neural-Network Control
ANNIT2FL	Adaptive Neural Network based Interval Type-2 Fuzzy Logic Controller
BL	Boolean Logic
COA	Center of Area
DOF	Degree of Freedom
DH	Denavit Hartenberg
FK	Forward Kinematic
FLC	Fuzzy Logic Controller
FIS	Fuzzy Inference System
FSC	Fuzzy Supervisory Control
GUI	Graphical User Interface
IK	Inverse Kinematic
NFC	Neuro-Fuzzy Controller
MF	Membership Function
MOM	Mean of Maximum
MRAFC	Model Reference Adaptive Fuzzy Control
P	Prismatic
PID	Proportional Integrated Derivative
R	Revolute
SCARA	Selective Compliance Assembly Robot Arm

TSK	Takagi, Sugeno and Kang
WMR	Wheeled Mobile Robot

Chapter I

Introduction

1.1 Background

A robot is a reprogrammable, multifunctional manipulator designed to move materials, parts, tools or specialized devices through variable programmed motions for the performance of a variety of tasks. Robots are used for various jobs such as dangerous jobs, like cleaning the main circulating pump housing in the nuclear power plant, and repetitive jobs that are boring, stressful, or labor-intensive for humans. The controlled target is a 3-DOF robot manipulator.

The Robot dynamic model is very substantial to generate the control input. It is very complicated operation to obtain its mathematical model, because of many reasons as the coupling between links, the strict nonlinearity and the time varying.

The "n^{th}" degree of freedom rigid Robot Manipulator is characterized by "n" nonlinear dynamic coupled deferential equation.

In robot controlling problems, it is very difficult to traces the desired path, so the problem of controlling robot manipulators still offers many practical and theoretical challenges due to the complexities of the robot dynamics and requirements to achieve high – precision trajectory tracking in the cases of high – velocity movement and highly varying loads.

The control method depends mainly on mathematical modeling, analysis and synthesis. To obtain the dynamic model in the mechatronic system Euler-Lagrange method is used because it is direct method for analysis.

The Denavit–Hartenberg convention is commonly used to select the coordinate frames for formulating the kinematic problem of serial manipulator.

The obtained presentation and the kinematic solution are used in formulating the dynamic equation.

1.2 Motivation

In recent years robotics technology becomes one of the high importance scientific researches; it highlights the growing importance in a wide variety of application and emphasizes its ability to inspire technology education. It is used in many areas and is important to the future of mankind.

Doctors are already using robotics in specialized surgeries. Some kind of robotic instrument that they can control from outside the body can cause the patient less pain and recovery time than having the surgeon completely open them up.

Robots are mostly utilized in the manufacturing industry, where the job is either too heavy or time consuming for a human.

Robotics is positioned to fuel a broad array of next-generation products and applications in fields as diverse as manufacturing, health-care, national defense and security, agriculture and transportation.

The target in robotics is how to control the motion of the robot; the control operation needs to obtain the mathematical model of the robot, which includes the forward and inverse kinematic, and the design the control low.

The main objective in this thesis is to make the robot traces desired trajectory, Infinite number of path to move from one point or position to the next; following a desired path is still a challenging task. Thus the problem of following a desired path will be investigated. Fuzzy logic controller (FLC) was found to be an efficient tool to control nonlinear systems; many applications of fuzzy logic control are reported in the various engineering fields including industrial processes and consumer products. Many model-based fuzzy control approaches are applied in robotics category such as Mamdani models, Takagi-Sugeno models, and Larsen models. The control of robots movements is very important step before implementing the veritable systems; this requires the achievement of the computer simulation to fulfill the control algorithm.

1.3 Literature review:

Robotics is the branch of technology that deals with the design, construction, manufacture and application of robots. A large number of researchers have been proposed a large number of solutions for controlling robot manipulator, some of literatures are listed next.

In 2011, Shahin, et. Al study, designed an adaptive neural network based interval type-2 fuzzy logic controller (ANNIT2FL), circular and handwriting type trajectory planning was proposed to show ability of a 3-DOF SCARA type robot manipulator. The researcher realized that the Cartesian trajectory tracking control of 3-DOF SCARA robot by using (ANNIT2FL) and PID controller. They said that the performances of ANNIT2FL controller has good, such that fast response and small errors for different rise function over circular tool trajectory control, and better than PID controllers performances over 3-DOF SCARA robot [1]. But the trajectory tracking figure shows that the error of tracking needs to be minimized.

In 1999, Young-Wan Cho, et. al, presented in their paper a direct Model Reference Adaptive Fuzzy Control (MRAFC) scheme for the plant model whose structure represented by the Takagi-Sugeno model. The MRAFC scheme proposed to provide asymptotic tracking of a reference signal for the systems with uncertain or slowly time-varying parameters. The proposed adaptive fuzzy control scheme was applied to tracking control of a two-link robot manipulator to verify the validity and effectiveness of the control scheme. From the simulation results, they conclude that the suggested scheme can effectively achieve the trajectory tracking even for the system with relatively large amount of parametric uncertainties [2].

In 2008, Rong-Jong Wai and Zhi-Wei Yang developed an adaptive fuzzy-neural-network control (AFNNC) scheme for an *n*-link robot manipulator to achieve high-precision position tracking. Takagi-Sugeno (T-S) dynamic fuzzy model with on-line learning ability constructed for representing the system dynamics of an *n*-link robot manipulator. Simulations of a two-link robot manipulator via (AFNNC) show the high performance and the high accuracy of the proposed controller [3].

The work presented by St.Joseph's, in 2005 described a fuzzy position control scheme designed for precise tracking of robot manipulator. Simulation results have shown the effectiveness of the proposed scheme [4].

In 2011, Jafar Tavoosi, et. al introduced a Neuro-Fuzzy Controller (NFC) for trajectory tracking control of robot arm. From the simulation results they said that Neuro– Fuzzy controllers provided good performance for control of robot manipulators [5].

The work presented by A. Alassar in 2010 investigated modeling and control of robot manipulator and used PID controller to compare its results with FLC and FSC (which is combining between the PID controller and FLC in order to improve the tuning of the PID parameters). The researcher proved that the FLC is more efficient in the time response behavior than the PID controller and the FSC is more efficient to control the robot arm to reach the desired output compared to classical tuning methods [6].

In 2001, Lam, et. al presented the control of a two-wheeled mobile robot using a fuzzy model approach. A fuzzy controller designed based on a T-S fuzzy plant model of the WMR. The authors said that the proposed fuzzy controller has an ability to drive the system states of the WMR to follow those of a stable reference fuzzy model [7].

In 2011, Wen-Jer Chang, et. al proposed a stability analysis and controller synthesis methodology for an inverted robot arm system. The system modeled by a state space Takagi-Sugeno (T-S) fuzzy model. Simulation results shows that the perturbed inverted robot arm system with disturbance can be controlled by the T-S fuzzy controllers, and the fuzzy controller designed in this paper can stabilized the nonlinear inverted robot arm [8]. But the computational time needs to be minimized.

In 2007, Nour, et. al addressed some of the potential benefits of using fuzzy logic controllers to control an inverted pendulum robot system and presented the stages of the development of a fuzzy logic controller using a four input Takagi-Sugeno fuzzy model. The main idea of their work is to implement and optimize fuzzy logic control algorithms in order to balance the inverted pendulum and at the same time reducing the computational time of the controller. The achieved results showed that proposed fuzzy logic controller is more robust to parameter variations when compared to the PID controller [9]. But the computational time of the controller is not acceptable.

In 2010, M. AbuQassem developed a visual software package (Graphical User Interface), which simulates a 5DOF robot arm; for testing motional characteristics of the AL5B Robot arm. A physical interface between the AL5B robot arm and the GUI was designed and built. Simulation results showed that the developed system was identified as an educational experimental tool. The results were displayed in a graphical format and the motion of all joints and end-effector could be observed [10].

1.4 Problem Statement and Objective

Robot manipulators represent complex dynamic systems with extremely variable inner parameters as well as the large intensive contact with the environment, an accurate control of such a complex system deals with the problem of uncertainty. The direct implementation of control in the control system of a real manipulator is impossible without obtained correct dynamic model. Complex dynamic systems can be modeled using an approach called the Lagrangian formulation. After obtaining the correct model, it is possible to apply the control law to make the robot traces desired trajectory. Infinite number of path to move from one point or position to the next, following a desired path is still a challenging task. Thus the problem of following a desired path will be investigated.

1.5 Methodology

Designing a control system that takes a desired function “sinusoidal for the two revolute joints, and linear function for the prismatic joint” as input and gives the appropriate output is the goal of this thesis. The outputs are the position and the velocity of the robot. Takagi-Sugeno controller will be used to perform control algorithm, the method used to solve this problem is:

1.5.1 Obtain the Dynamic Model

By using Lagrange Euler formulation, which is based on the concepts of generalized coordinates, energy and generalized force are obtained.

1.5.2 Design the Fuzzy Controller

The principal design elements in a general fuzzy logic control are Fuzzification, Control rule base establishment and Defuzzification. In this project, Takagi-Sugeno (also known as the TSK fuzzy model) fuzzy model will be adopted to construct the fuzzy model of the system, due to its capability to approximate any nonlinear behavior.

1.5.3 Design the Simulink Model

MATLAB will be the platform to simulate the 3-DOF robot manipulator as a case study in this thesis.

1.5.4 Simulate and Compare Results

The thesis will compare the results accomplished by the simulation with some previous studies.

1.6 Contribution

This thesis aims to identify the parameters of the robot, derive the mathematical model of the robot, and use algorithm of Takagi-Sugeno fuzzy model to control the robot motion and make it to tracks a desired path accurately. The contribution of this thesis is to use TSK controller to control the manipulator. This study can be used as a document of reference for other researches that are interested in this area of robotics using fuzzy logic control.

1.7 Thesis Structure

A brief description for each chapter and the organization of this thesis is structured as follow.

Chapter 2 presents, the fundamentals of the dynamic model for the manipulators and the common problem in robotics that known as kinematic analysis which separated into two parts, the forward kinematics and inverse kinematics.

Chapter 3 presents the approach of fuzzy logic control. Some basic concepts in fuzzy logic such as fuzzy sets, features of membership functions, linguistic variables, linguistic values, linguistic rules and the operations in fuzzy logic are presented.
Also the four components of fuzzy logic controller, the fuzzifier, rule base, inference mechanism and defuzzifier are discussed. Finally the design of fuzzy logic controller is presented.

Chapter 4 shows the simulation and results of fuzzy position control scheme for precise tracking of robot manipulator.

Chapter 5 presents the conclusion and summarization of this work. Some recommendations and suggestions for future works also presented.

Chapter II

Dynamic Model for Robot Manipulator

2.1 Background:

In Robotics, there are two main types of robots, the first is manipulator robots and the other is mobiles robots, in this work we aim in manipulator robots. A manipulator robot is a very complex, uncertain, nonlinear system with an extremely variable in inner parameters. It is named according to the number of joints or the number of degree of freedom.

Robot manipulator known as a device controlled by a human operator, designed to move materials, parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks. It is created from several segments connected in series by joints which can be moved in a linear or rotate motion. Also it is created from a number of actuators allowing the motion to the link, and a number of sensors to measure the output.

In general practice, the final goal of controlling a manipulator is to put the end-effector, the link furthest from the base, at some specific coordinates. However, in order to put the end-effector at these coordinates, the joints have to be moved to some angles. A direct transformation exists between these angles and the *xyz* coordinates of the end-effector. This transformation is known as the direct kinematics [11].

The Kinematics is the science of motion that treats the subject, without regard to the forces that cause it. Within the science of kinematics, one studies the position, the velocity, the acceleration, and all higher order derivatives of the position variables. Hence, the study of the kinematics of manipulators refers to all the geometrical and time-based properties of the motion. The relationships between these motions and the forces and torques that cause them constitute the problem of dynamic motion geometry of the robot manipulator from the reference position to the desired position [12].

Tha gist of conrtoller designing is to obtain a formulation of kinematic analysis, which done by ***Denivit-Hartenberg*** convention. This Convention is used to select coordinate frames for formulationg the kinematic problem of serial manipulator. The obtained formulation and frame from kinematic solution can be used for, formulating the dynamic model, defining the position and orientation of the current link with respect to previous one. In addition, it allows the desired frame to create a set of steps to bring the other links coordinate into corresponding with another one. The dynamic equations explicitly describe the relationship between force and motion. The equations of motion are important to consider in the design of robots, in simulation and animation of robot motion, and in the design of control algorithms.

2.2 Kinematic Chains

Robot Kinematic refers the analytical study of the motion of a robot manipulator. Formulationg the suitable kinematics models for a robot mechanism is very crucial for analyzing the behaviour of industrial manipulators. There are mainly two types of problems in the kinematic of robot manipulator, the first is the forward and inverse kinematic.

2.2.1 Froward Kinematic

Any manipulator is created from serial of links connected in series by joints, revolute or prismatic, from the base frame through the end-effector. Calculating the position and orientation of the end-effector in terms of the joint variables is known as forward kinematics. To obtain the forward kinematic equations for the manipulator the following steps must be done;

a) Obtain Denavit-Hertenberg convention equations:

Denavit-Hertenberg convention that uses four parameters is the most common method for describing the manipulator kinematic.

These parameters are the link length a_{i-1}, the link twist α_{i-1} , the link offset d_i, and the joint angle θ_i.

A coordinate frame is attached to each joint to determine Denavit-Hertenberg parameters; the coordinate frame for the manipulator is shown in Figure (2.1).

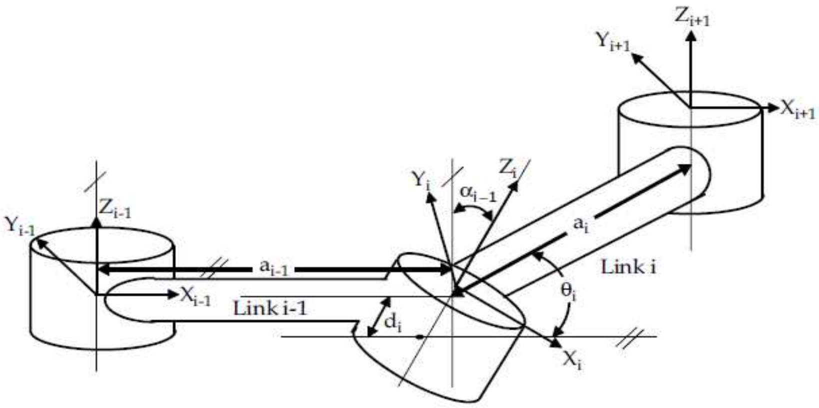


Figure (2.1): *Coordinate Frame for the Manipulator*

The length a_{i-1} is the distance from Z_i and Z_{i-1} measured along X_{i-1}

The twist α_{i-1} is the angle between Z_{i-1} and Z_i measured along X_i.

The offset d_i is the distance between X_{i-1} and X_i measured along Z_i.

The angle θ_i is the angle between X_{i-1} to X_i measured about Z_i [3].

Table (2.1): *Denavit-Hertenberg convention for 3-DOF Manipulator*

$link$	$LinkParameters$			
i	q_i	α_i	a_i	d_i
1	θ_1	0	l_1	0
2	θ_2	0	l_2	0
3	d_3	0	0	d

b) Derivation of link transformations matrices T^{i-1}_{i}:

To construct the transform that defines frame i relative to frame $i-1$. The general transformation matrix T^{i-1}_{i} for a single link from joint 1 to joint i is represented as a product of four basic homogenous transformations,

$$T^{i-1}_{i} = R_x(\alpha_{i-1})\ D_x(a_{i-1})\ R_z(\theta_i)\ Q_i(d_i) \qquad (2.1)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_{i-1} & -S\alpha_{i-1} & 0 \\ 0 & S\alpha_{i-1} & C\alpha_{i-1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T^{i-1}_{i} = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & a_{i-1} \\ S\theta_i C\alpha_{i-1} & C\theta_i C\alpha_{i-1} & -S\alpha_{i-1} & -S\alpha_{i-1} d_i \\ S\theta_i S\alpha_{i-1} & C\theta_i S\alpha_{i-1} & C\alpha_{i-1} & C\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (2.2)$$

Where R_x and R_z present rotation, D_x and Q_i denote translation, and $C\theta_i$ and $S\theta_i$ are the short hands of $Cos\theta_i$ and $Sin\theta_i$ respectively.

The forward kinematics of the end-effector with respect to the base frame is determined by multiplying all of the T^{i-1}_{i} matrices.

Since the matrix T^{i-1}_{i} is a function of single variable, it turns out that three of the above four quantities are constant for a given link “fixed by mechanical design”, while the fourth paramete θ_i for a revoulute joint and d_i for a prismatic joint, is the joint variable [13,14].

c) Concatenating link transformations matrix T_i^0 **:**

It is very important step, to calculate the position and orientation of the end-effector of the manipulator. Once the link frameworks have been calculated and corresponding link parameters defined, developing the kinematic equation is modest and straightforward. From the values of the link parameters, the individual link-transformation matrices can be computed. Then the link transformations can be multiplied together to find the single transformation that relates frame ***i*** to frame **0**, the general homogenous matrix for the desired position and orientation of the end-effector can be written as follows:

$$T_{end-effector}^{base} = T_1^0 \ T_2^1 \ T_3^2 \ \ldots \ T_i^{i-1} \qquad (2.3)$$

It can be written as:

$$T_{end-effector}^{base} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (2.4)$$

Where r_{ij} represent the rotational elements of transformation matrix, P_x, P_y and P_z denote the elements of the position vectors. Equation (2.4) can be divided into two main components, where more information can be found in Appendix A

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

$$(2.5)$$

$$P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}$$

The vector $r_1 = (r_{11}, r_{12}, r_{13})^T$ represents the direction of x_i, the vector $r_2 = (r_{12}, r_{22}, r_{32})^T$ represents the direction of y_i, and the $r_3 = (r_{13}, r_{23}, r_{33})^T$ vector represents the direction of z_i, in the Cartesian Coordinates. The position vector $P = (P_x, P_y, P_z)^T$ represents the vector of translation from the origin 0_i to the origin 0_{i-1} [2,3,4,5].

Inverse Kinematic

Inverse kinematic is concerned with the inverse problem of finding the joint variable in term of the end-effector position and orientation. Solving the inverse kinematics is computationally expensive and generally takes a very long time in real time control of manipulators. Mathemathically it can be expressed as:

$$\theta_k = f(x,y,z,\alpha,\gamma,\phi) \tag{2.6}$$

Where $k = 1,2,\ldots..,i$, θ_k the joint angle and $(x,y,z,\alpha,\gamma,\phi)$ represents the position and orientation.

For solving the inverse kinematic for robot manipulator, the following steps must be followed:

1. Obtain the general transformation matrix for the desired position and orientation of the robot manipulator:

$$T_{end-effector}^{base} = T_1^0 \; T_2^1 \; T_3^2 \; \ldots \; T_i^{i-1} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.7}$$

2. For both matrices, define:
   a) All elements that contain one joint variable.
   b) Pairs of elements, which contain only one joint variable.
   c) Combinations of elements contain more than one joint.

3. Equate it to the corresponding elements in the other matrix to form equation, 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 case of inaccuracy, solutions look for another one.
6. If there is more joint variable to be found, multiply equation (2.7) by the inverse of **T** 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 term of an element transformation matrix, it means that the arm cannot achieve the specified position and orientation; the position is outside the robot manipulator workspace.

The general problem of inverse kinematics can be stated as follows:

1. Given a 4X4 homogeneous transformation:

$$H = \begin{bmatrix} R & o \\ 0 & 1 \end{bmatrix} \tag{2.8}$$

2. Find (one or all) solutions of equation

$$T_i^0(q_1,\ldots.,q_n) = H \tag{2.9}$$

Where

$$T_i^0(q_1,\ldots.,q_n) = T_1^0(q_1) \;\ldots..\; T_i^{i-1}(q_2) \tag{2.10}$$

As shown in Appendix A, H represents the desired position and orientation of the end-effector, and the task is to find the values for the joint variables $(q_1,\ldots.,q_i)$ so that $T_i^0(q_1,\ldots.,q_n) = H$, equation (2.9) results in twelve nonlinear equations in n unknown variables, which can be written as:

$$T_{ij}(q_1,\ldots,q_n) = h_{ij} \qquad i = 1,2,3, \quad j = 1,2,\ldots,4 \qquad (2.11)$$

Where T_{ij}, h_{ij} refer to the twelve nontrivial entries of T_i^0 and H respectively. (Since the bottom row of both T_i^0 and H are (0,0,0,1), four of the sixteen equations represented by (2.9) are trivial [6,14].

2.3 Dynamics

The kinematic equations describe the motion of the robot without the consideration of the forces and torques producing the motion, while the dynamic equations describe the relationship between forces and motion. The dynamic equations of motion are important for, designing the robot, simulation, animation of robot motion and designing control algorithm. Euler-Lagrange equation is a known method to describe the evaluation of a mechanical system. The Lagrangian of the system must be calculated in order to determine the Euler-Lagrange equations.

2.3.1 Lagrange-Euler Equation

The Lagrangian formulation is an "Energy-based" approach to dynamics [12], **L** is the difference between the kinetic energy and the potential energy, it is provides a formulation of the dynamic equations of motion equivalent to those derived using Newton's second law.

$$m\ddot{y} = f - mg \qquad (2.12)$$

The Lagrangian can be written as:

$$\mathcal{L} = \mathcal{K} - \mathcal{P} \qquad (2.13)$$

Note that:

$$\frac{\partial L}{\partial \dot{y}} = \frac{\partial K}{\partial \dot{y}} \quad and \quad \frac{\partial L}{\partial y} = -\frac{\partial P}{\partial y} \qquad (2.14)$$

Equation (2.12) can be written as:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{y}} - \frac{\partial L}{\partial y} = f \qquad (2.15)$$

For any system, an application of the Euler-Lagrange equation leads to a system of ***n*** coupled, second order nonlinear ordinary differential equations of the form:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \Gamma \qquad (2.16)$$

The generalized force Γ represents those external forces and torques that are not derivable from a potential function, it may be motor torque. The order *n* of the system is determined by the number of so-called generalized coordinates that are required to describe the evolution of the system [12,14].

2.3.2 General Expression for Kinetic Energy

First, starting by driving an expression for the kinetic energy of manipulator, and noting that the kinetic energy of any rigid object consists of two terms, the first term is the translational kinetic energy due linear velocity of the center of mass, and the second term is the rotational kinetic energy due to angular velocity of the link.

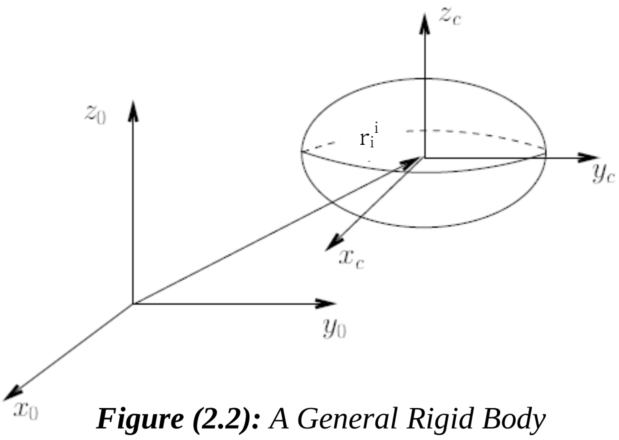


Figure (2.2): *A General Rigid Body*

When a rigid body moves in a pure rotation about a fixed axis, every point of the body moves in a circle. The centers of these circles lie on the axis of rotation. As the body rotates, a perpendicular from any point of the body to the axis sweeps out an angle θ, and this angle is the same for every point of the body.

The segment position vector r_i^i shown in Figure (2.2) can be expressed as:

$$r_i^i = \begin{bmatrix} x_c & y_c & z_c & 1 \end{bmatrix} \tag{2.17}$$

$$r_i^0 = T_i^0 r_i^i$$

In the case of revolute joint the general form of T_i^{i-1} is,

$$T^{i-1}{}_i = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_{i-1} & S\theta_i S\alpha_{i-1} & a_{i-1} \\ S\theta_i C\alpha_{i-1} & C\theta_i C\alpha_{i-1} & -S\alpha_{i-1} & -S\alpha_{i-1} d_i \\ S\theta_i S\alpha_{i-1} & C\theta_i S\alpha_{i-1} & C\alpha_{i-1} & C\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.18}$$

In the case of prismatic joint the general form of T_i^{i-1} is,

$$T^{i-1}{}_i = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_{i-1} & S\theta_i S\alpha_{i-1} & 0 \\ S\theta_i C\alpha_{i-1} & C\theta_i C\alpha_{i-1} & -S\alpha_{i-1} & 0 \\ 0 & C\theta_i S\alpha_{i-1} & C\alpha_{i-1} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.19}$$

The angular velocity of any point on the joint can be expressed as:

$$V_i^0 = \frac{d}{dt}(r_i^0) = \frac{d}{dt}(T_i^{i-1} r_i^i) \tag{2.20}$$

$$= \dot{T}_1^0 T_2^1 \ldots T_i^{i-1} r_i^i + T_1^0 \dot{T}_2^1 \ldots T_i^{i-1} r_i^i + \ldots + T_1^0 T_2^1 \ldots \dot{T}_i^{i-1} r_i^i + T_i^0 \dot{r}_i^i \tag{2.21}$$

$$\dot{r}_i^i = 0 \;, \; \dot{T}_1^0 = \frac{d}{dt}(T_i^0) = \frac{\partial T_i^0}{\partial q_j} \frac{dq_j}{dt}$$

Then, the final expression of the angular velocity is:

$$V_i = \left[\sum_{j=1}^{i} \frac{\partial T_i^0}{\partial q_j} \frac{dq_j}{dt} \right] r_i^i = \left[\sum_{j=1}^{i} \frac{\partial T_i^0}{\partial q_j} \; \dot{q}_j \right] \tag{2.22}$$

The notation (q) refers to two variables; it indicates the variable θ_i in the case of revolute joint, and it indicates the variable d_i in the case of prismatic joint. In the case of revolute joint $(q_i = \theta_i)$

$$\frac{\partial T_i^{i-1}}{\partial \theta_i} = \frac{\partial}{\partial \theta_i}\begin{bmatrix} C\theta_i & -S\theta_i C\alpha_{i-1} & S\theta_i S\alpha_{i-1} & a_{i-1} \\ S\theta_i C\alpha_{i-1} & C\theta_i C\alpha_{i-1} & -S\alpha_{i-1} & -S\alpha_{i-1} d_i \\ S\theta_i S\alpha_{i-1} & C\theta_i S\alpha_{i-1} & C\alpha_{i-1} & C\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.23}$$

$$= \begin{bmatrix} -S\theta_i & -C\theta_i C\alpha_{i-1} & C\theta_i S\alpha_{i-1} & -a_i S\theta_i \\ C\theta_i & -S\theta_i C\alpha_{i-1} & S\theta_i S\alpha_{i-1} & a_i C\theta_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.24}$$

Equation (2.24) can be written as, multiplication between equation (2.23) and pre-multiplication matrix known as Q_i

$$\frac{\partial T_i^{i-1}}{\partial \theta_i} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} C\theta_i & -S\theta_i C\alpha_{i-1} & S\theta_i S\alpha_{i-1} & a_{i-1} \\ S\theta_i C\alpha_{i-1} & C\theta_i C\alpha_{i-1} & -S\alpha_{i-1} & -S\alpha_{i-1} d_i \\ S\theta_i S\alpha_{i-1} & C\theta_i S\alpha_{i-1} & C\alpha_{i-1} & C\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.25}$$

The derivation of the end-effector transformation matrix with respect to any joint variable (q_j) can be written as

$$\frac{\partial T_i^0}{\partial q_j} = \frac{\partial}{\partial q_j}(T_1^0 T_2^1 \ldots T_j^{j-1}) = T_1^0 T_2^1 \ldots \frac{\partial}{\partial q_j}(T_j^{j-1}) \ldots T_i^{i-1} \tag{2.26}$$

Then the general form is

$$\frac{\partial}{\partial q_j}(T_1^0) = \begin{cases} T_1^0 T_2^1 \ldots T_{j-1}^{j-2} Q_j T_j^{j-1} T_i^{i-1} & , j \le i \\ 0 & , j \ge 0 \end{cases} \tag{2.27}$$

Define the quantity $U_{ij} \equiv \dfrac{\partial T_i^0}{\partial q_i}$ which expresses the movement effect of joint *j* on the segment

$$U_{ij} = \begin{cases} T_j^0 Q_j T_i^{j-1} & j \le i \\ 0 & j \ge i \end{cases} \tag{2.28}$$

replacing the notation U_{ij} in equation (2.22)

$$V_i = \left[\sum_{j=1}^{i} U_{ij} \dot{q}_j \right] r_i^i \tag{2.29}$$

In the case of prismatic joint $(q_i = d_i)$

$$\frac{\partial T_i^{i-1}}{\partial d_i} = \frac{\partial}{\partial d_i} \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_{i-1} & S\theta_i S\alpha_{i-1} & 0 \\ S\theta_i C\alpha_{i-1} & C\theta_i C\alpha_{i-1} & -S\alpha_{i-1} & 0 \\ 0 & C\theta_i S\alpha_{i-1} & C\alpha_{i-1} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.30}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.31}$$

Equation (2.31) can be written as, multiplication between equation (2.30) and pre-multiplication matrix known as Q_i

$$\frac{\partial T_i^{i-1}}{\partial d_i} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_{i-1} & S\theta_i S\alpha_{i-1} & 0 \\ S\theta_i C\alpha_{i-1} & C\theta_i C\alpha_{i-1} & -S\alpha_{i-1} & 0 \\ 0 & C\theta_i S\alpha_{i-1} & C\alpha_{i-1} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.32}$$

The main advantage of using the pre-multiplication matrix is to avoid the repeated derivation of the transformation matrix T_i^{i-1}.

Define the quantity $U_{ijk} \equiv \frac{\partial U_{ij}}{\partial q_k}$ which expresses the velocity intersection effect which created by the different velocities of the joints. This quantity can be calculated according to the form

$$U_{ijk} \equiv \frac{\partial U_{ij}}{\partial q_k} \begin{cases} T_{j-1}^{0} Q_j T_{k-1}^{j-1} T_i^{k-1} & ,i \geq k \geq j \\ T_{k-1}^{0} Q_k T_{j-1}^{k-1} Q_j T_i^{j-1} & ,i \geq j \geq k \\ 0 & ,i \leq k \text{ or } i \leq k \end{cases} \tag{2.33}$$

The kinetic energy can be expressed as:

$$\mathcal{K} = \frac{1}{2} m v^T v + \frac{1}{2} w^T I w \tag{2.34}$$

Where m is *n x n* matrix called manipulator mass matrix, v and w are the linear and angular velocity vectors, and I is a symmetric 3 X 3 matrix called the Inertia Tensor [12,14,15].

a) Inertia Tensor

It is necessary to express the inertia tensor, I and it is relative to the inertial reference frame and depends on the configuration of the object. The form of inertia tensor is

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \tag{2.35}$$

Where

$$\begin{aligned} I_{xx} &= \iiint (y^2 + z^2)\rho(x,y,z)\ dx\ dy\ dz \\ I_{yy} &= \iiint (x^2 + z^2)\rho(x,y,z)\ dx\ dy\ dz \\ I_{zz} &= \iiint (x^2 + y^2)\rho(x,y,z)\ dx\ dy\ dz \end{aligned} \tag{2.36}$$

$$I_{xy} = I_{yx} = -\iiint xy\rho(x,y,z)\ dx\ dy\ dz$$

$$I_{xz} = I_{zx} = -\iiint xz\rho(x,y,z)\ dx\ dy\ dz \qquad (2.37)$$

$$I_{yz} = I_{zy} = -\iiint yz\rho(x,y,z)\ dx\ dy\ dz$$

Where $\rho(x,y,z)$ is the mass density, the diagonal elements I_{xx}, I_{yy}, I_{zz} are called the principal moment of Inertia, and the elements $I_{xy}, I_{yx}\ldots\ etc.$ are called the cross product of Inertia. The form general for computing the matrix of inertia tensor is:

$$J_i = \begin{bmatrix} \frac{-I_{xx}+I_{yy}+I_{zz}}{2} & -I_{xx} & -I_{xz} & -m_i\bar{x}_i \\ -I_{xy} & \frac{I_{xx}-I_{yy}+I_{zz}}{2} & -I_{yz} & -m_i\bar{y}_i \\ -I_{xz} & -I_{yz} & \frac{I_{xx}+I_{yy}-I_{zz}}{2} & -m_i\bar{z}_i \\ -m_i\bar{x}_i & -m_i\bar{y}_i & -m_i\bar{z} & m_i \end{bmatrix} \qquad (2.38)$$

The vector $(\bar{x}_i, \bar{y}_i, \bar{z}_i)$ is the vector of center of mass.

b) Kinetic energy for an *n*-link Robot

The kinetic energy of link i of mass m_i is expressed as:

$$K = \frac{1}{2}\dot{q}^T \sum_{i-1}^{n}\left[m_i J_{vi}(q)^T J_{vi}(q) + J_{wi}(q)^T R_i(q) I_i R_i(q)^T J_{wi}(q)\right]\ \dot{q} \qquad (2.39)$$

Then the kinetic energy of manipulator is:

$$K = \frac{1}{2}\dot{q}^T D(q)\dot{q} \qquad (2.40)$$

Where $J(q)$ is the Jacobian matrix and $D(q)$ is a symmetric positive definite matrix is called inertia matrix [12,14,16].

2.3.3 General Expression for Potential Energy

In the case of rigid dynamics, the source of potential energy is gravity. The potential energy of i^{th} link can be computed by assuming that the mass of the object is concentrated at its center of mass, it is expressed by:

$$P_i = g^T r_{ci} m_i \tag{2.41}$$

Where, g is the vector of gravity, and r_{ci} the coordinate of the center of mass of link i then the potential energy of manipulator is:

$$P = \sum_{i=1}^{n} P_i = \sum_{i=1}^{n} g^T r_{ci} m_i \tag{2.42}$$

The potential energy is a function of generalized coordinates not their derivative, the potential energy depends on the configuration of robot not on its velocity [14].

2.3.4 Motion Equations

After obtaining the total kinetic and potential energy, then the Euler-Lagrange equation can be written as:

$$L = K - P = \frac{1}{2} \sum_{ij} d_{ij}(q) \dot{q}_i \dot{q}_j - P(q) \tag{2.43}$$

By taking the derivative of equation (2.25)

$$\frac{\partial L}{\partial \dot{q}_k} = \sum_j d_{kj} \dot{q}_j$$

and (2.44)

$$\frac{d}{dt} \frac{\partial}{\partial \dot{q}_k} = \sum_i d_{kj} \ddot{q}_j + \sum_j \frac{d}{dt} d_{kj} \dot{q}_j = \sum_j d_{kj} \ddot{q}_j + \sum_{ij} \frac{\partial d_{kj}}{\partial q_i} \dot{q}_i \dot{q}_j$$

Also

$$\frac{\partial L}{\partial q_k} = \frac{1}{2} \sum_{ij} \frac{\partial d_{ij}}{\partial q_k} \dot{q}_i \dot{q}_j - \frac{\partial P}{\partial q_k} \tag{2.45}$$

Thus the Euler-Lagrange equation can be written:

$$\sum_j d_{kj}\ddot{q}_j + \sum_{ij}\left\{\frac{\partial d_{kj}}{\partial q_i} - \frac{1}{2}\frac{\partial d_{ij}}{\partial q_k}\right\}\dot{q}_i\dot{q}_j - \frac{\partial P}{\partial q_k} = \Gamma_k \qquad (2.46)$$

The term

$$\sum_{i,j}\left\{\frac{\partial d_{kj}}{\partial q_i}\right\}\dot{q}_i\dot{q}_j = \frac{1}{2}\sum_{i,j}\left\{\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j}\right\}\dot{q}_i\dot{q}_j$$

and (2.47)

$$\sum_{i,j}\left\{\frac{\partial d_{kj}}{\partial q_i} - \frac{1}{2}\frac{\partial d_{ij}}{\partial q_k}\right\}\dot{q}_i\dot{q}_j = \sum_{i,j}\frac{1}{2}\left\{\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k}\right\}\dot{q}_i\dot{q}_j$$

Put

$$c_{ijk} = \frac{1}{2}\left\{\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k}\right\} \qquad (2.48)$$

Then:

$$c_{kj} = \sum_i^n c_{ijk}(q)\dot{q}_i = \frac{1}{2}\left\{\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k}\right\} \qquad (2.49)$$

The term in equation (2.48) is called Christofell symbols, and then the Lagrange-Euler equations can be written as:

$$\sum_j d_{kj}(q)\ddot{q}_j + \sum_{ij} c_{ijk}(q)\dot{q}_i\dot{q}_j + \frac{\partial P}{\partial q_k} = \Gamma_k \qquad k=1,2,\ldots,n \qquad (2.50)$$

There are three types of terms. The first one involves the second derivative of generalized coordinates. The second type involves the first derivative of generalized coordinates; this type is classified into two terms, terms involving a product of the type $\dot{q}_i^2$ are called centrifugal while those involving a product of the type $\dot{q}_i\dot{q}_j$ where $i \neq j$ are called coriolis terms. The third type is those involving only the generalized coordinates.

Finally, the form of Euler-Lagrange Equations is expressed as:

$$\left[D(q)+J\right]\ddot{q}+C(q,\dot{q})\dot{q}+f(\dot{q})+g(q)=\Gamma \qquad (2.51)$$

Where

Γ : *Vector of dimension (n X 1), called the vector of generalized forces applied on the joints.*

q: *Vector of dimension (n X 1), called the vector of joints variables of the manipulator.*

$\dot{q}$: *Vector of dimension (n X 1), called the vector of angular velocity.*

$\ddot{q}$: *Vector of dimension (n X 1), called the vector of angular acceleration.*

$\left[D+J\right](q)$: *Matrix of dimension (n X n), called the matrix of inertia,* $J(q)$ *is the motor proper inertia.*

$h(q,\dot{q})$: *Vector of dimension (n X 1), called the vector of centrifugal and coriolis.*

$f(\dot{q})$: *Vector of dimension (n X 1), called the vector of friction coefficients.*

$g(q)$: *Vector of dimension (n X 1), called the vector of gravity.*

Chapter III

Fuzzy Control, Approach and Design

3.1 Background:

The fuzzy set theory was put earliest in 1965 by Professor Zadeh when he presented his paper on fuzzy sets and introduced the concept of linguistic variable.

Fuzzy Logic is a logic system that uses imprecision, was first invented as a representation scheme and calculus for uncertain or vague notions. It is basically a multi-valued logic that allows more human-like interpretation and reasoning in machines by resolving intermediate categories between notations, such as true/false, hot/cold etc, used in Boolean logic. This was seen as an extension of the conventional Boolean Logic that was extended to handle the concept of partial truth or partial false rather than the absolute values and categories in Boolean logic.

Fuzzy logic categories objects into sets which are described by linguistic variables such as long, fast, cool, heavy, middle-aged and so on. Objects can have varying degrees of membership of such fuzzy sets, ranging from a crisp “definitely not a member, denoted by 0” to a crisp definitely a member, denoted by “1”. The crucial distinction is that between these crisp extremes, objects can have less certain degree of membership such as “not really a member, perhaps denoted by a “0.1” and “pretty much a member, 0.9 possibly” [17].

Fuzzy Logic can be applied to control, thus assumes the name Fuzzy Control. Fuzzy Control is made up of control rules which simulate those used by humans when they control or operate the machines. It can be especially effective way of controlling non-linear systems when expert human knowledge of the system is available. Many researches and applications have been performed since Mamdani and his colleague presented the first FLC work [18].

Fuzzy logic uses rules with antecedents and consequents to produce outputs from inputs. The antecedents are the inputs that are used in the decision-making process or the “IF” parts of the rules, while the consequents are the implications of the rules or the “THEN” parts [17].

As mentioned before, Fuzzy Control applies fuzzy logic to the control of process by utilizing different categories, usually the "*Error*" and the "*Change of error*", for the process state and applying rules to decide a level of output [17,19].

In modern, the classical control systems have been replaced by the FLC, which means that the IF-THEN rules and fuzzy membership functions replaces the mathematical models to control the system, the main advantage of fuzzy controllers, especially when the obtainment of the mathematical models of the system may be too complex.

When designing a Fuzzy Logic Controller (FLC) expert knowledge of the process to be controlled can be used to design the membership functions and rule base, but unfortunately there is no general procedure for designing a FLC.

3.2 Fundamentals on Fuzzy Logic:

Classical logic deals with prepositions (conclusion or decision) that are either true or false. The main content of classical logic is the study of rules that allow new logical variables to be produced as function of certain existing variables. While the main idea in fuzzy set theory is that the element has a degree of membership to a fuzzy set [11,20]. An introduction to fuzzy set theory and some definitions will be discussed.

3.2.1 Fuzzy Sets:

Fuzzy set theory provides means for representing uncertainties; it uses linguistic variables, rather than quantitative variables to represent imprecise concepts. A fuzzy set is a set containing elements that have varying degrees of membership in the set. The elements in fuzzy set, because their membership need not be complete, can also be members of other fuzzy sets on the same universe [21].

If U is a collection of objects denoted generically by x, then a fuzzy set A in U (universe of discourse) is defined as a set of ordered pairs:

$$A = \{(x, \mu_A(x)) / x \in U\}$$
$$\mu_A(x) \in [0,1] \tag{3.1}$$

Where $\mu_A(x)$ is the membership for the fuzzy set A. The output of the membership for a given input is called degree of the membership.

The classical set theory is built on the fundamental concept "set" of which an individual is either a member or not a member.

As an example, anyone between 175 and 190 cm is considered as tall, in crisp set the membership function is defined as:

$$\mu_A(x) = \begin{Bmatrix} 1 \ \ if \ tall \in [175, 190] \\ 0 \ \ if \ tall \notin [175, 190] \end{Bmatrix} \tag{3.2}$$

But in fuzzy set, the membership function is defined as a function of the variable (*tall*) and according to the definition in equation (3.1); the membership function is defined as:

$$\mu_A(x) = func(tall) \tag{3.3}$$

In classical, to indicate that c is a subset of ***S***; we write $c \subset S$, that means all the elements of the set s are contained in the set S, in fuzzy set theory c is a subset of ***S*** if:

$$\mu_c(x) \le \mu_S(x) \tag{3.4}$$

In classical theory an empty set is denoted by ϕ , that means the set contains no element, but in fuzzy theory the empty set ***S*** is defined as

$$\mu_c(x) = 0 \quad \forall x \in S \qquad (3.5)$$

that means if *c* is a fuzzy set then no element in ***S*** has a member in *c* [6,21].

3.2.2 Features of Membership Function:

A membership function for a fuzzy set *A* on the universe of discourse *X* is defined as $\mu_A : X \to [0,1]$, each element of *X* is mapped to a value between 0 and 1. This value, called membership value or degree of membership, quantifies the grade of membership of the element in X to the fuzzy set A. Membership functions allow us to graphically represent a fuzzy set.

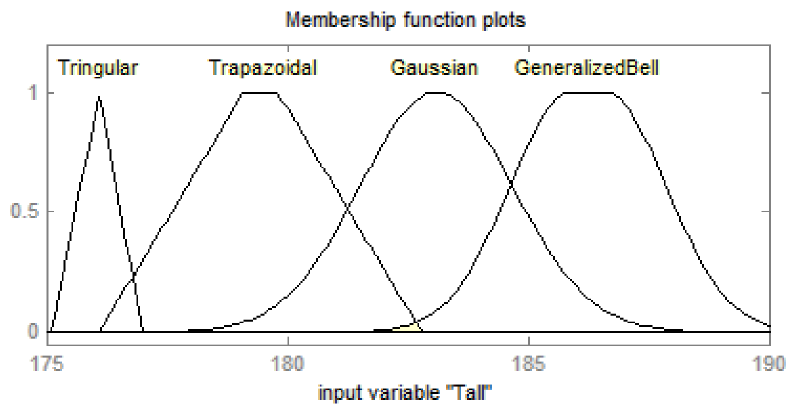


Figure (3.1): *Some typical membership functions*

The membership function is said to be normal if one element at least or more in the universe has a value 1; the center of a fuzzy set is the mean value of all points that achieves the maximum value of $\mu_A(x)$. The height of fuzzy set is the largest value of $\mu_A(x)$. Membership Functions characterize the fuzziness of fuzzy sets.

It is essentially embodies all fuzziness for a particular fuzzy set. Its description is essential to fuzzy property or operation [6,20,21].

3.2.3 Linguistic Variables:

Linguistic variables are the variables expressed as human language, which represents imprecise information. Linguistic expressions are needed for the inputs and outputs and the characteristics of the inputs and outputs [6,19].

For example, if we study the case of the error of the angle of the pendulum using the fuzzy logic, then the error is linguistic variable that takes different fuzzy sets, as shown in figure (3.2)

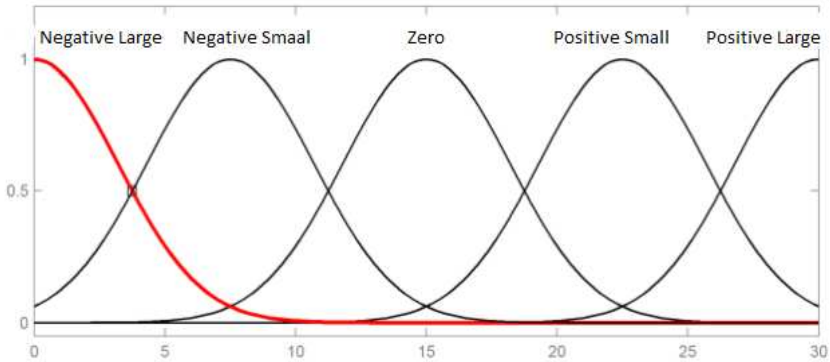


Figure (3.2): *Example of Linguistic Variables*

It is clear that the linguistic variables can be naturally represented by fuzzy sets and logical connectives "*and, or*" of these sets. Each element of figure (3.2) is defined as a mathematical function.

For example, the statement "*error is positive small*" can represents the situation where the pendulum is at a significant angle to the *left* of the vertical.

Many books and papers used the notation of linguistic variables in the form:

$$\{X, N(X), U, S_x\} \tag{3.6}$$

Where, *X* designates linguistic variable name such as error and change of error N(X) is the set of all linguistic names of the linguistic variable, like "*Negative Large, Negative Small, Zero, Positive Small and Positive Large*" S_x is the meaning of the variable that returns to the linguistic variable, such as the part "*Positive Large*" means the linguistic variable "*error*" is "*Positive Large*". Finally, U is the universe of discourse of the variables where *X* takes a crisp value [6,18,20].

3.2.4 Linguistic Values

Just as u_i and y_i take on values over each universe of discourse U_i and Y_i respectively, linguistic variables take "*Linguistic values*" that are used to describe characteristics of the variables [19].

Let U_i^n denote the n^{th} linguistic value of the linguistic variable u_i defined over the universe of discourse U_i. If we assume that there exist many linguistic values defined over U_i, then the linguistic variable u_i takes on the elements from the set of linguistic values denoted by:

$$U_i = \{U_i^n : n = 1, 2, \ldots, N_i\} \qquad (3.7)$$

Linguistic values are generally descriptive term such as "*Positive Large*", "*Zero*", and "*Negative Big*". For example, if we assume that the linguistic variable denotes "*Speed*" then we may assign the linguistic values as, "U_1= *Slow*", "U_2= *Medium*" and "U_3= *Fast*" [6,18,20,21].

3.2.5 Linguistic Rules:

As well as specifying the membership functions, the rule base also needs to be designed. The mapping of the inputs and outputs for a fuzzy system is in part characterized by a set of *condition* (*action rules*), usually the inputs of the fuzzy system are associated with the premise, and the outputs are associated with the consequent.

The If-Then rule takes the form, **IF** *premise* **THEN** *consequent* [17,19,22].

As example, for two inputs error e and the change of error Δe and one output u with a universe of discourses $E, \Delta E$ and U respectively, and the "**IF-THEN**" rule has the form:

$$
\begin{aligned}
R_1 &: \ IF\ e\ is\ A_1\ AND\ \Delta e\ is\ B_1\ THEN\ u\ is\ C_1 \\
R_2 &: \ IF\ e\ is\ A_2\ AND\ \Delta e\ is\ B_2\ THEN\ u\ is\ C_2 \\
& \quad \dots \quad \dots \quad \dots \quad \dots \\
R_n &: \ IF\ e\ is\ A_n\ AND\ \Delta e\ is\ B_n\ THEN\ u\ is\ C_n
\end{aligned}
$$

More accurate, if we choose the membership of the error and the change of error "*Negative Large, Zero, Positive Large*", then we can write the rules as follow:

$$
\begin{aligned}
R_1 &: \ IF\ e\ is\ NL\ AND\ \Delta e\ is\ NL\ THEN\ u\ is\ NL \\
R_2 &: \ IF\ e\ is\ NL\ AND\ \Delta e\ is\ ZE\ THEN\ u\ is\ NL \\
& \quad \dots \quad \dots \quad \dots \quad \dots \\
R_n &: \ IF\ e\ is\ PL\ AND\ \Delta e\ is\ PL\ THEN\ u\ is\ PL
\end{aligned}
$$

The experience of the human controller is usually expressed as linguistic "IF-THEN" rules that state in what situations which actions should be taken [23].

3.2.6 Operation on Fuzzy Sets:

Professor Zadah suggests some operations on fuzzy set theory like, intersection, union and complement.

3.2.6.1 Intersection "*AND*"

The intersection of fuzzy sets A and B which are defined on the universe of discourse U_i is a fuzzy set denoted by $A \cap B$ with a membership function defined by the minimum of the membership values as in:

$$\mu_C(u) = \mu_{A\cap B}(u) = \min\{\mu_A(u), \mu_B(u)\}$$
$$or\ \mu_C(u) = \mu_A(u) \cap \mu_B(u) \tag{3.8}$$

In fuzzy logic, intersection is used to represent the "*and*" operation. For example, if we use minimum to represent the "*and*" operation, then the shaded membership function in Figure (3.3) is $\mu_{A\cap B}$ which is formed from the intersection of the two fuzzy sets A and B [19,20,21].

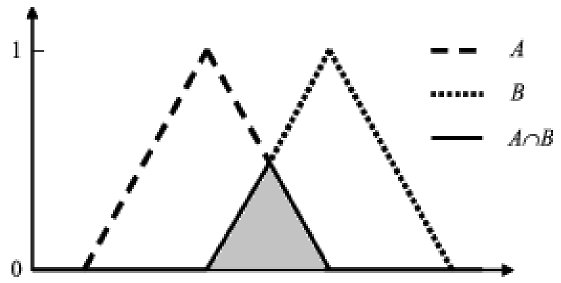


Figure (3.3): *A membership function for the Intersection of two Fuzzy sets*

3.2.6.2 Union "*OR*"

The union of fuzzy sets A and B, which are defined on the universe of discourse U, is a fuzzy set denoted by $A \cup B$, with a membership function defined by the maximum of the membership values as in:

$$\mu_C(u) = \mu_{A\cup B}(u) = \max\{\mu_A(u), \mu_B(u)\}$$
$$or\ \mu_C(u) = \mu_A(u) \cup \mu_B(u) \tag{3.9}$$

In fuzzy logic, union is used to represent the "*Or*" operation. For example, if we use maximum to represent the "*Or*" operation, then the lined membership function in figure (3.4) is $\mu_{A\cup B}$ which is formed from the intersection of the two fuzzy sets A and B [19,20,21].

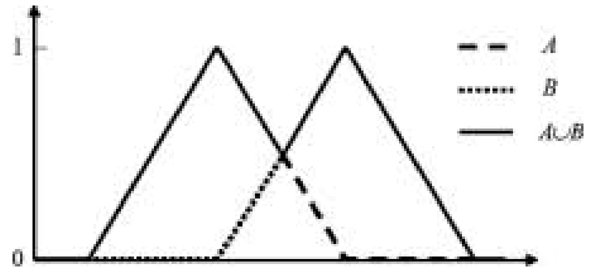


Figure (3.4): *A membership function for the Union of two Fuzzy sets*

3.2.6.3 Complement

The complement "*not*" of a fuzzy set A with a membership function $\mu_A(u)$ has a membership function given by:

$$\mu_{\bar{A}}(u) = 1 - \mu_A(u) \qquad (3.10)$$

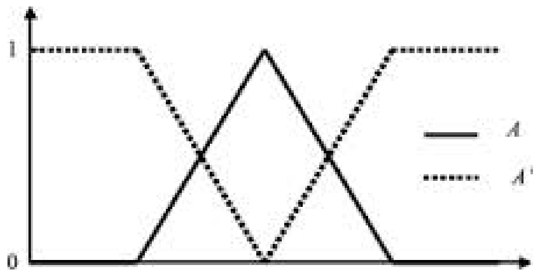


Figure (3.5): *A membership function for the complement of Fuzzy sets*

The dashed membership function is the complement of the lined membership function [19,20,21].

3.2.6.4 Cartesian product

The fuzzy Cartesian product is used to quantify operations on many universes of discourse. If $A_1, A_2, \ldots, A_n$ are fuzzy sets defined on the universes of discourse $U_1, U_2, \ldots, U_n$ respectively, their Cartesian product is a fuzzy set denoted by $A_1 \times A_2 \times \ldots \times A_n$ with a membership function defined by

$$\mu_{A_1\times A_2\times\ldots\times A_n}(u_1,u_2,\ldots,u_n)=\mu_{A_1}(u_1)\times\mu_{A_2}(u_2)\times\ldots\times\mu_{A_n}(u_n) \qquad (3.11)$$

[10,11,12].

3.2.6.5 Algebraic product

The algebraic product of two fuzzy sets A and B is defined as [10]

$$A.B=\left\{\left(u,\mu_A(u)\right).\left(u,\mu_B(u)\right)\right\} \qquad (3.12)$$

3.2.6.6 Compositional Rule of Interface

If R is a fuzzy set relation $U\times V$ and A is a fuzzy set in U, then the fuzzy set B in V includes A is given by: [6,19]

$$B=A*R \qquad (3.13)$$

Which read, *A* composition *R*. There are two cases in compositional rule; the first is maximum-minimum (max-min) operation:

$$\mu_A(u)=\max_{A\in U}\left\{\min(\mu_A(u),\mu_B(v))\right\} \qquad (3.14)$$

The other case is maximum-product (max-product) operation:

$$\mu_A(u)=\max_{A\in U}\left\{\mu_A(u)\bullet\mu_B(v)\right\} \qquad (3.15)$$

3.3 Structure of Fuzzy Logic Controller

The input and the output of the Fuzzy logic controller are non-fuzzy (crisp) values. FLC consists of four main components, the fuzzifier, rule base, inference mechanism and defuzzifier. The *fuzzifier* converts the crisp input to a linguistic variable using the membership functions stored in the fuzzy knowledge base. The *rule base* holds the knowledge, in the form of a set of rules, of how best to control the system. The inference *mechanism* determines the extent to which each rule is relevant to the current situation as characterized by the input and draws decisions using the current inputs and the information in the rule-base. And the *defuzzifier* converts the fuzzy output of the inference mechanism to crisp using membership functions into crisp values [19,20,21,24,25].

3.3.1 Fuzzification

Fuzzification is the process of making a crisp quantity fuzzy. It is scale the input crisp value into a normalized universe of discourse U, then converts each crisp input to a degree of membership function. For example, the fuzzifier converts the input value 10 into a linguistic variable. For each input and output variables selected, determined number of membership functions and a qualitative category for each one of them, must be defined. As shown in figure (3.1) the shape of these membership functions can be diverse.

3.3.2 Rule Base

Once the input and output variables and memberships are defined, we have to design the rule-base or "*decision matrix of the fuzzy knowledge-base*". Rule base is the core of FLC; it is combined human expertise with a series of logical rules for using the knowledge. The rules connecting the input and the output are based on the understanding of the system. It composed of "*expert*" If (antecedents) Then (conclusion) rules. These rules transform the input variables to an output that, the input combined by "*And*" or "*Or*" operator. Depending on the number of memberships for the input and the output variables, it will be able to define more or less potential rules. The easier case is a rule-base concerning only one input and one output variable, the more variables the more rules have to define in order to make the inference reliable [26,27,28,29].

Once a variable is fuzzified, it takes a value between 0 and 1 indicating the degree of membership to a given membership of that specified variable. The degree of membership of the input variables has to be combined to get the degree of membership of the output variable.

Rule base takes conditional statement that have the following form:

$$\text{If } e \text{ is } PS, \text{ And } \Delta e \text{ is } NL \text{ Then } u \text{ is } NS \qquad (3.16)$$

The most popular methods to form the ***If-Then*** rules are, Mamdani and Takagi, Sugeno & Kang (TSK). The two methods have the same antيecedent evaluation of the ***If-Then*** statement. But the main different between the two is the consequent, where the consequent in Mamdani depends on the designer or the human operator, but in TSK, the consequent is a function of real value.

3.3.3 Inference Mechanism

Fuzzy inference mechanism process is obtaining the relevant control rule at the current time then decides the behavior of the output. Fuzzy inference system uses a collection of fuzzy membership functions (MFs) and rules; it uses ***If-Then*** fuzzy rules to convert the fuzzy input to the fuzzy output. It consist of three parts, a rule base containing a selection of fuzzy rules, a data base defining the fuzzy values used in fuzzy rules, and a reasoning mechanism [16], figure (3.6) describes the model of the inference mechanism [19,21,24].

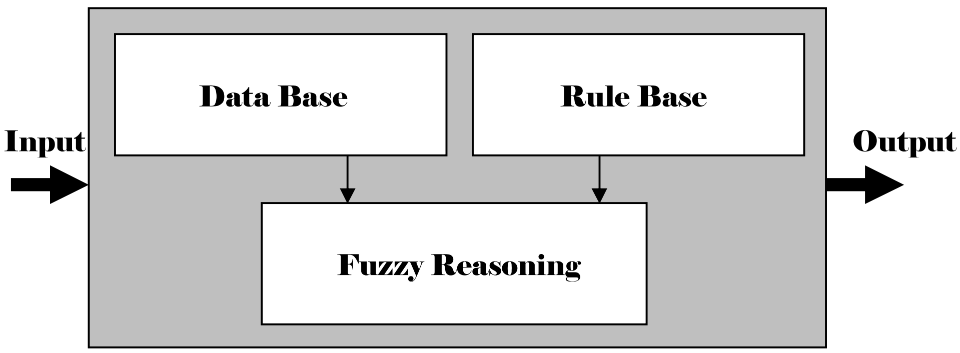


Figure (3.6): *Model of Fuzzy Inference Processing*

The value of membership function for the rules is calculated using fuzzy inference mechanism "*Implication*". There are several ways used to implement fuzzy inference methods, the most used are Mamdani, Larsen, and Takagi, Sugeno & Kang (TSK). Those are briefly described as follow:

3.3.3.1 Mamdani FIS "*max-min*"

A linguistic model that describes the system by means of linguistic ***If-Then*** rules with fuzzy preposition in the antecedent as well as in the consequent, implication is modeled by means of minimum operator and the resulting output membership functions are combined using maximum operator.

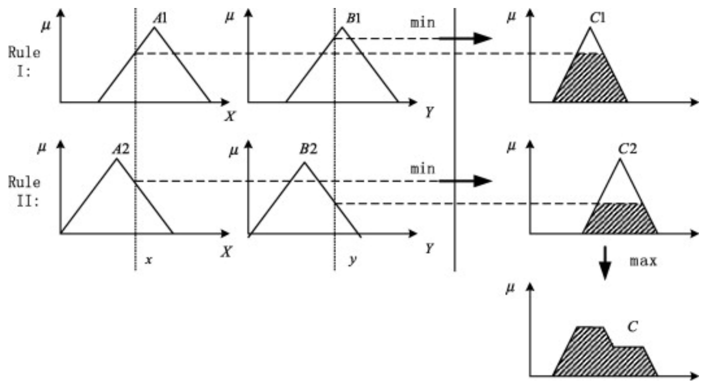


Figure (3.7): *Fuzzy Inference Processing using Mamdani Model*

3.3.3.2 Larsen FIS "*max-product*"

Implication is modeled using the product operator, while preposition are defined by the maximum operator.

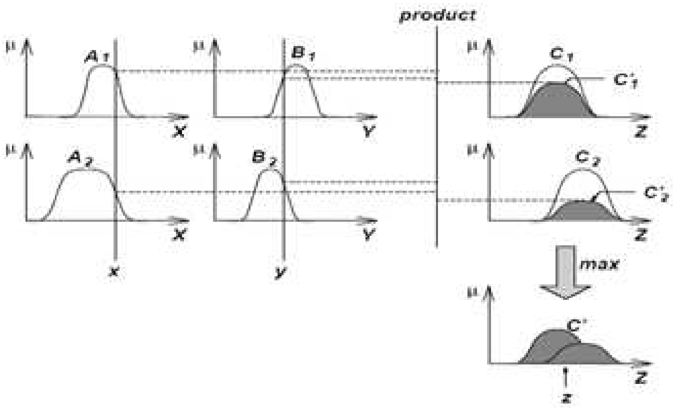


Figure (3.8): *Fuzzy Inference Processing using Larsen Model*

3.3.3.3 Takagi, Sugeno & Kang (TSK) FIS

Developed to reduce the number of rules required by the Mamdani model. It replaces the fuzzy sets, then part, of Mamdani rule with function or equation of input variables. Sometimes the function is a constant; overall output via is always crisp. It is avoids the time-consuming method of defuzzification necessary in the Mamdani model.

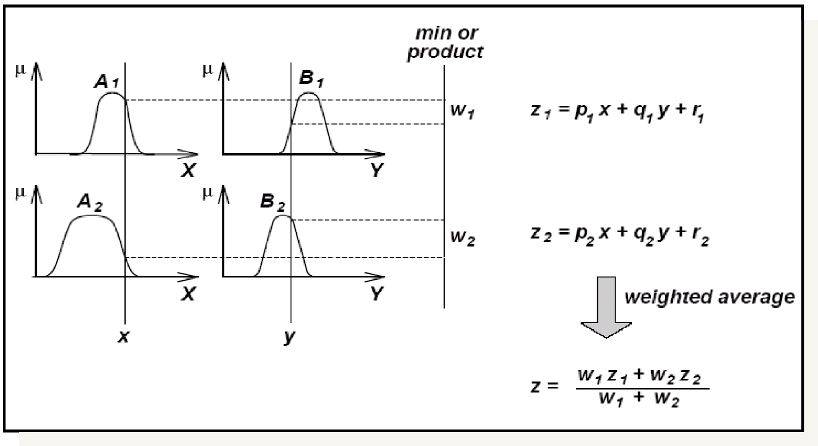


Figure (3.9): *Fuzzy Inference Processing using TSK Model*

3.3.4 Defuzzification

It is the final step in FLC, it means convert the fuzzy set into a crisp values that can be sent to the plant as a control signal. The conclusion or control output derived from the combination of input, output membership functions and fuzzy rules is still a vague or fuzzy element, and this process is called fuzzy inference. To make that conclusion or fuzzy output available to real application, a difuzzification process is needed. A number of defuzzification strategies exist but the most popular are, center of gravity "*COG*", mean of maximum "*MOM*" and weighted average "*WT*" [30].

3.3.5 Fuzzy Controller Design

Fuzzy control system design essentially amount to:

1. Identifying the fuzzy controller input and output.

2. Partitioning the universe of discourse or the interval spanned by each variable into a number of subsets, assigning each a linguistic label.

3. Choosing the preprocessing that is needed for the controller inputs and outputs.

4. Designing each of the four components "*Fuzzification, Knowledge base, Inference engine, and Defuzzification*" of the fuzzy controller, as shown in figure 3.13.

5. Validate the model, if it does not meet the expected performance; iterate on the above design steps.

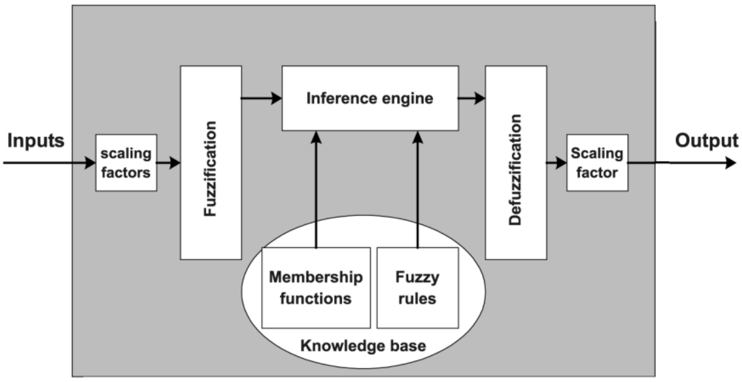


Figure (3.10): *Four Components of the Fuzzy Logic Controller*

It should be noted that the success of designing, depends on the problem at hand, and the extent and quietly of the available knowledge. For some problems, the knowledge-base design may lead fast, while for others it may be a very time-consuming. There for it is useful to combine the knowledge based design with a data-driven tuning for the model parameters.

3.3.6 Fuzzy Controller for 3-DOF Robot

3.3.6.1 3-DOF Robot

The 3-DOF configuration, shown in figure (3.11), is a popular manipulator which, as its name suggests, is tailored for assembly operations. The class was proposed as a means to provide motion capabilities to the end-effector that are required by the assembly of printed-board circuits and other electronic devices with a flat geometry [15].

It has two parallel revolute joints (allowing it to move and orient in a plane), with a third prismatic joint for moving the end-effector normal to the plane. The main advantage is that the first two joints don't have to support any of the weight of the manipulator or the load. In addition, the base of the manipulator can easily house the actuators for the first two joints. The actuators can be made very large, so the robot can move very fast.

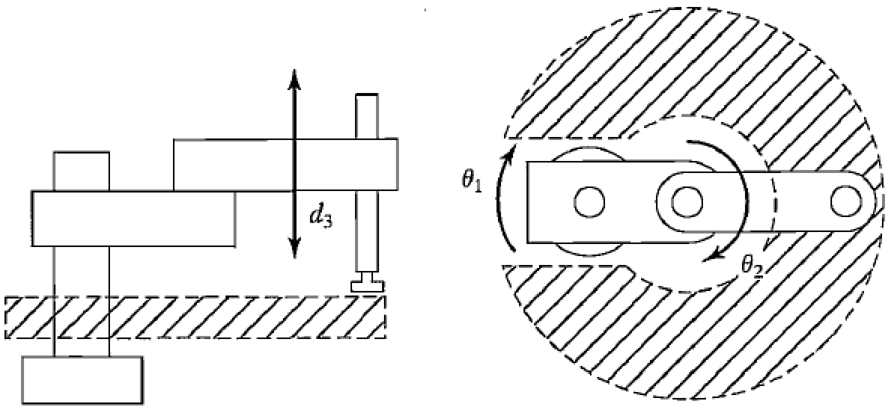


Figure (3.11): *Side view and Top view for 3-link Manipulator*

A 3-link robot manipulator will be represented by a mathematical model "as shown in Appendix A" with three input driving torques "*tau*" and six output variables, three angle displacements for the three joints, and three angular velocities, in practice the final goal of controlling a manipulator is to put the end-effector at some specific positions. These positions are previously determined by the operator to achieve specified functions.

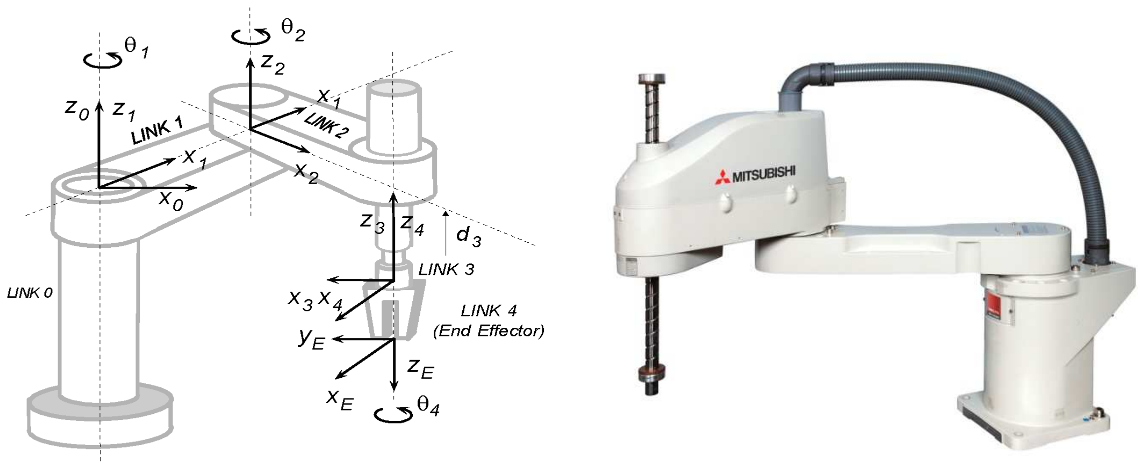


Figure (3.12): *Three-link SCARA Manipulator*

3.3.6.2 Designing the Fuzzy Controller

a) Identifying the fuzzy controller input and output.

In the proposed FLC, the measured "*Error*" and "*Derivative of error*" of the position are the inputs of FLC. They are scaled to some numbers in the interval [-90 90], these values indicate to the angle of rotation, and are mapped to linguistic variables by fuzzification operator. The values of linguistic variables are composed of linguistic terms, Negative Large "*NL*", Negative Small "*NS*", Zero "*ZE*", Positive Small "*PL*", and Positive Large "*PL*".

While the FLC output is the position which scaled in the interval [-90 90], and mapped to linguistic variables, the values of linguistic variables are composed of linguistic terms, Negative Large "*NL*", Negative Small "*NS*", Zero "*ZE*", Positive Small "*PL*", and Positive Large "*PL*", which are all fuzzy sets as shown in figure (3.15). The Fuzzy Control Model used here is Takagi and Sugeno (TSK) model.

Figure (3.13) shows the memberships of the position error, which indicate to the error of angle of rotation.

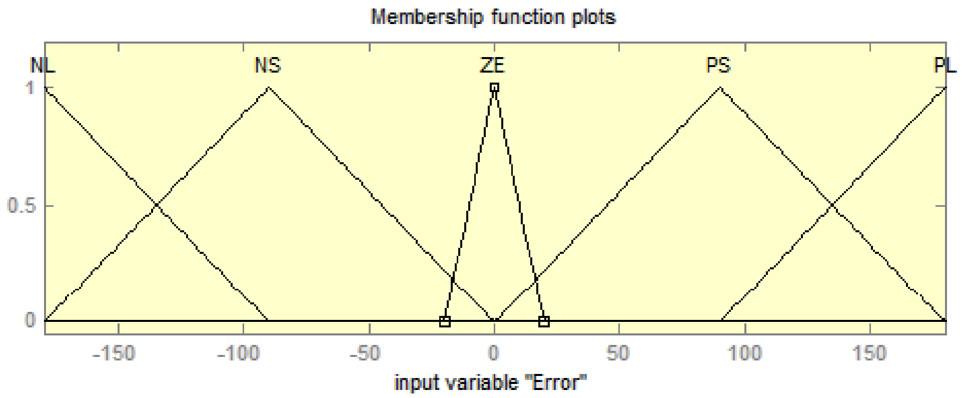


Figure (3.13): *Membership Functions for the input Variable Position Error*

Figure (3.14) shows the memberships of the position derivative of error, which indicate to derivative of error of the angle of rotation.

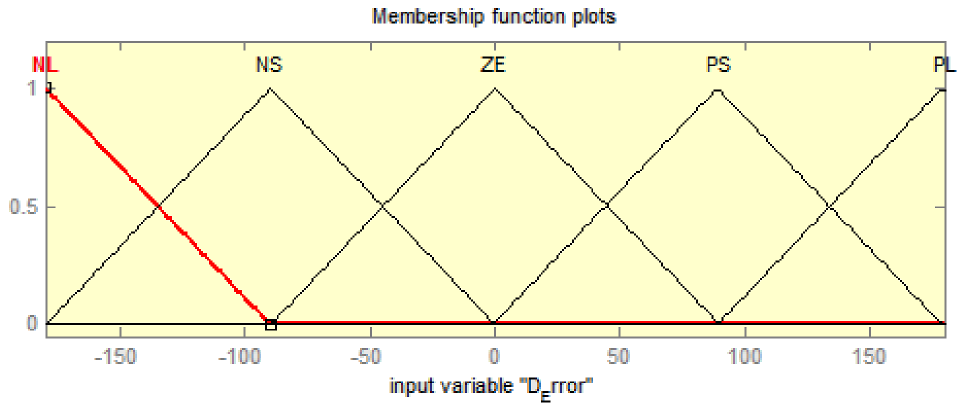


Figure (3.14): *Membership Functions for the input Variable Position Derivative of Error*

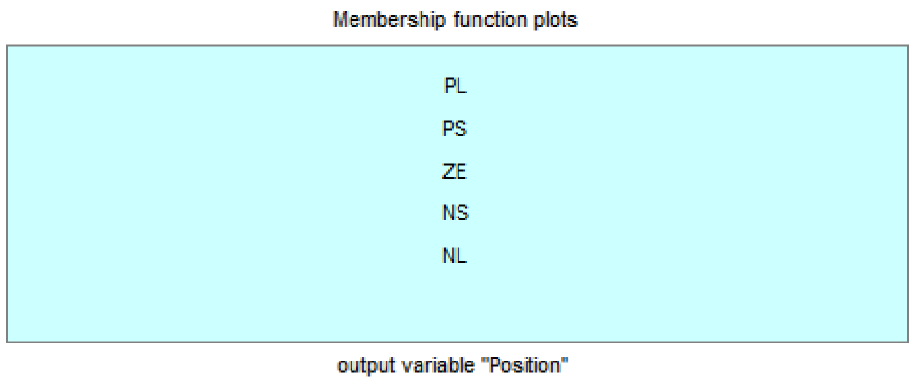


Figure (3.15): *Membership Functions for the Output Variable Position*

Figure (3.15) shows the memberships functions for the output, which indicate to the exact position of the each link. For example if the error of the position is *positive small* and the change of error is *zero* then the link will be at the position 0.0431 at the Cartesian coordinates.

b) *Recommended Control Rule Base.*

The fuzzy rule is represented by a sequence of the form ***If-Then***, leading to algorithms describing what action or output should be taken. In the proposed FLC, twenty-five rules recommended to achieve the tracking. Fuzzy control rules for the designed controller are listed in table (3.1). For example, a rule base has the following form:

$$\textit{If (Error is PS) and (D-Error is ZE) then (Position} = 0.0431) \qquad (3.19)$$

Table (3.1): *Fuzzy Control Rules*

Error / D-Error	**NL**	**NS**	**ZE**	**PS**	**PL**
NL	NL	NL	NL	NS	ZE
NS	NL	NL	NS	ZE	PS
ZE	NL	NS	ZE	PS	PL
PS	NS	ZE	PS	PL	PL
PL	ZE	PS	PL	PL	PL

The fuzzy control rules of the proposed controller have been derived experimentally from studying and observe the response of the process to be controlled. The twenty-five rules have been derived by the multiplication of the numbers of membership functions of the two inputs.

Figure (3.16) shows the block diagram of the fuzzy inference process, that is the process of formulating the mapping from a given input to an output using fuzzy logic.

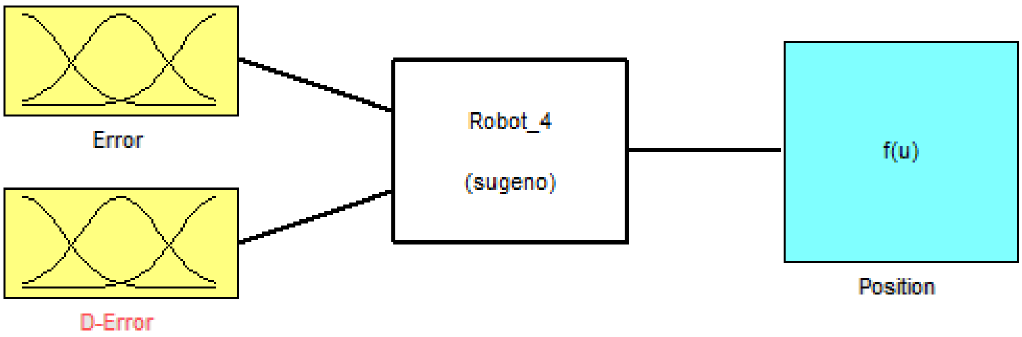


Figure (3.16): *Fuzzy Inference Block*

3.3.6.3 Summary

A review of the fundamental of fuzzy sets, fuzzification, fuzzy rules, fuzzy inference and defuzzification is discussed in this chapter. These operations are described using a three-link robot manipulator. The effectiveness of the control system depends on the system response which is achieved via Matlab Simulation.

Chapter IV

Simulation And Results

4.1 Introduction

This chapter presents simulations and results for the manipulator, SIMULINK MATLAB is used to simulate and evaluate the performance of the proposed controller that applied on the robot. Dynamic model of the three-DOF robot manipulator has been utilized in the synthesis process of control.

4.2 MATLAB SIMULINK

MATLAB "*Matrix Laboratory*" is a commercial package which operates as an interactive programming environment. It is a high-level language and interactive environment for numerical computation, visualization, and programming which created by Math-work Inc.

SIMULINK is a block diagram environment for multi-domain simulation and Model-Based design. It supports system-level design, simulation, automatic code generation, and continuous test and verification of embedded systems.

This part presents the SIMULINK diagrams for the manipulator and the designed controller; figure (4.1) shows MATLAB embedded function, which let to compose a MATLAB function within a SIMULINK model, of the nonlinear robot manipulator, as seen the inputs of the embedded function are the angles, the positions and the velocities of the three joints, while the outputs are the accelerations of the three joints.

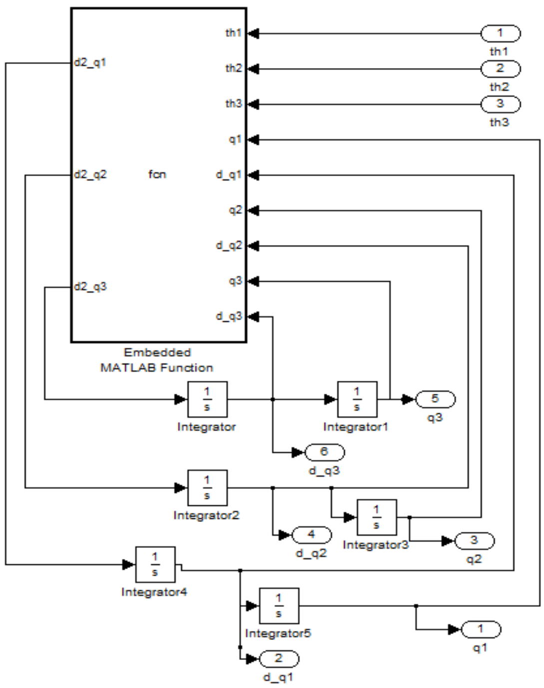


Figure (4.1): *Embedded MATLAB Function of the Nonlinear System*

Figure (4.2) shows MATLAB embedded function of the nonlinear feedback system, which allows applying the control low.

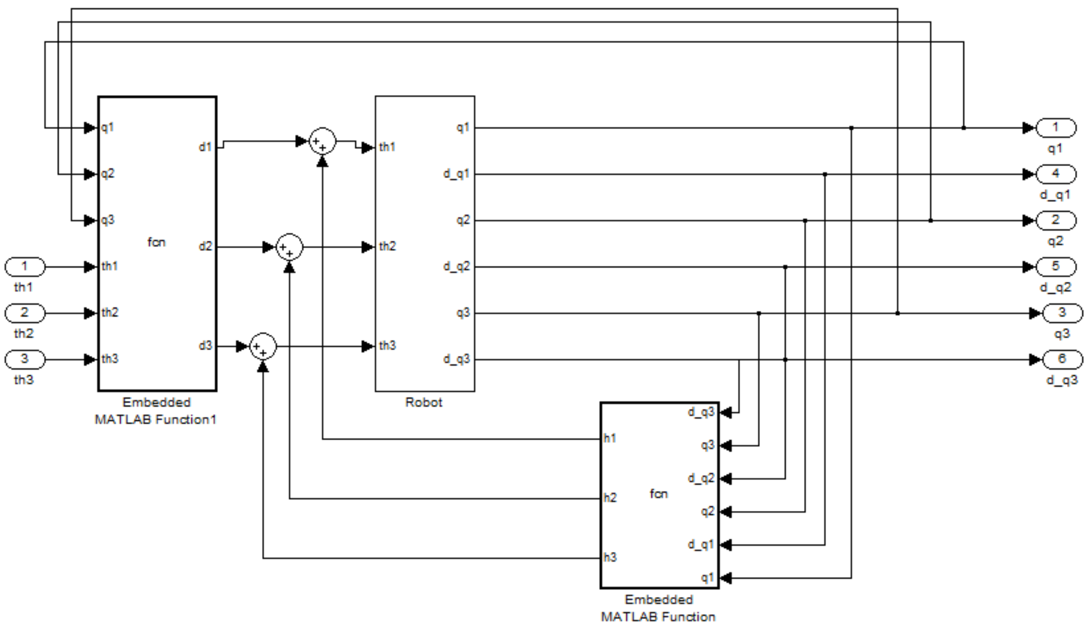


Figure (4.2): *Embedded MATLAB Function of the Nonlinear Feedback System*

Figure (4.3) depicts the block diagram of the subsystem of the three-link robot with fuzzy position controller.

The gains (G1,G3,G5) are the gains of the errors of the three-links, while the gains (G2,G4,G6) are the gains of the changes of errors. The gains (G7,G8,G9) are the gains of the designed fuzzy controller.

The tracking curves for positions "*Tracking 1, Tracking 2, and Tracking 3*" and velocities "*Velocity tracking 1, Velocity tracking 2, Velocity tracking 3*" can be shown in the scoops which are named to signify the display of each one.

Also the errors in positions and velocities are shown in figure (4.3) to define the difference between the desired and the actual trajectories. The scoops which display the errors are named as, "*Error Position 1, Error Position 2, and Error Position 3*",

"*Error Velocity 1, Error Velocity 2, and Error Velocity 3*"

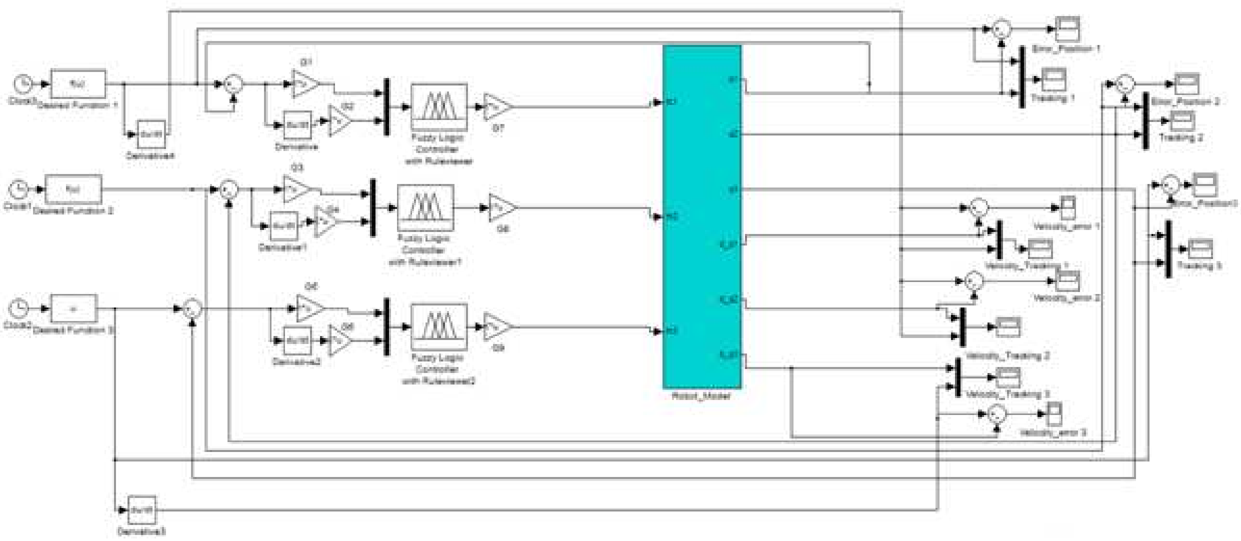


Figure (4.3): *Block Diagram of the Three-link SCARA Robot with Fuzzy Controller*

The desired trajectories are named as "*Desired Function 1, Desired Function 2, Desired Function 3*", finally the controllers appear as "*Fuzzy Logic Controller with Rule viewer*".

4.3 3-DOF Robot as Case Study

The proposed fuzzy controller will be used to control the 3-links robot as case study. Designed to mimic the action of a human arm and can be used in jobs from automobile factories to underwater construction. This tool is frequently utilized because of its speed, efficiency and low cost. It can be programmed to perform precise jobs repetitively, such as installing a pin or carrying items from one location to another within its range of motion.

It is consists of two revolute and one prismatic joints as shown in figure (3.15). The vertical motion is usually an independent linear axis at the wrist or in the base. Table (4.1) shows the values of the parameters of the manipulator.

Table (4.1): *Values of the Parameters of 3-DOF Robot*

$link$	$Link Parameters$			
i	q_i^*	α_i	a_i	d_i
1	θ_1^*	0	$l_1 = 0.5\ m$	0
2	θ_2^*	0	$l_2 = 0.5\ m$	0
3	d_3^*	0	0	$d_3 = 0.3 + \Delta d_3\ \ m$

Simulation and numerical results prove that the performance of the fuzzy logic controller is good and the obtained figures illustrate that the tracking of the joints demonstrate that, as shown in the next.

Figure (4.4) shows the position tracking curve of the first joint, the blue curve is the desired path while the red is the actual. Sinusoidal function was chosen as a desired trajectory.

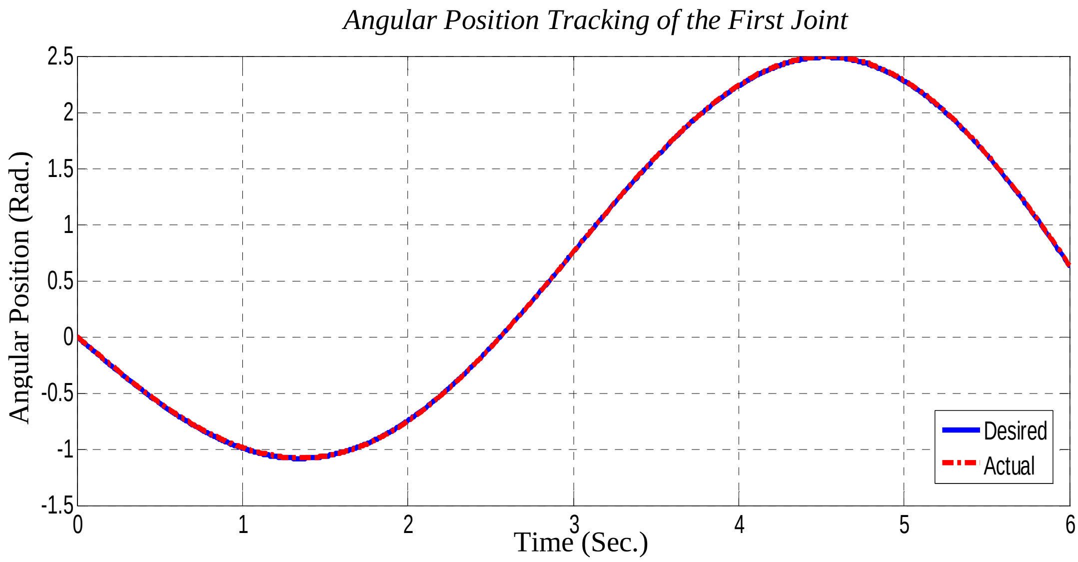


Figure (4.4): *Position Tracking Curve of the First Joint*

The figure shows that the first joint tracks the desired path accurately, with very small tracking error as shown in figure (4.5).

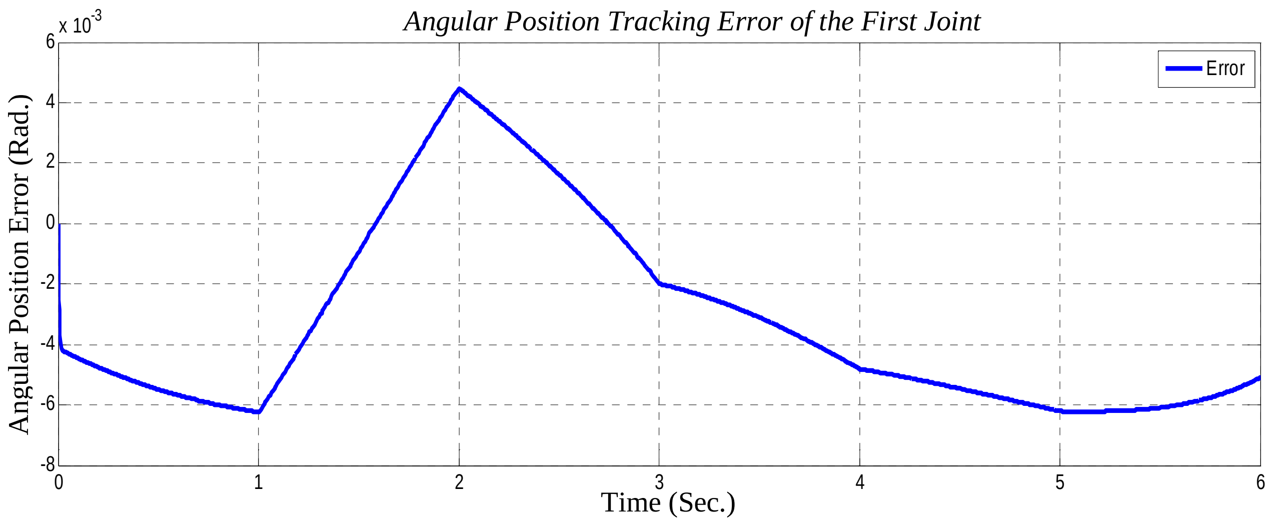


Figure (4.5): *Position Tracking Error Curve of the First Joint*

Figure (4.6) shows the position tracking curve of the second joint, the same desired trajectory for the first joint chosen for second joint.

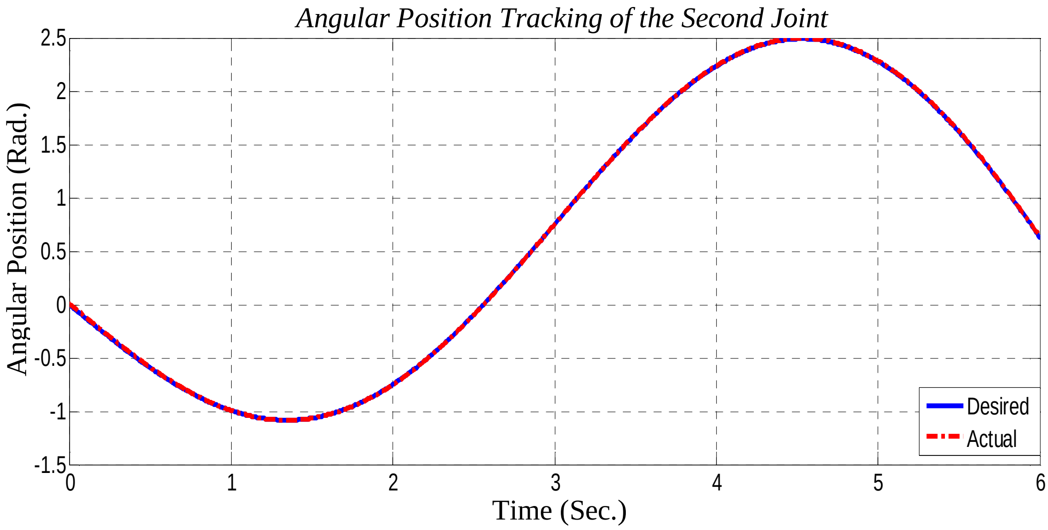


Figure (4.6): *Position Tracking Curve of the Second Joint*

Also the figure shows that the second joint tracks the desired path accurately, with very small tracking error as shown in figure (4.7).

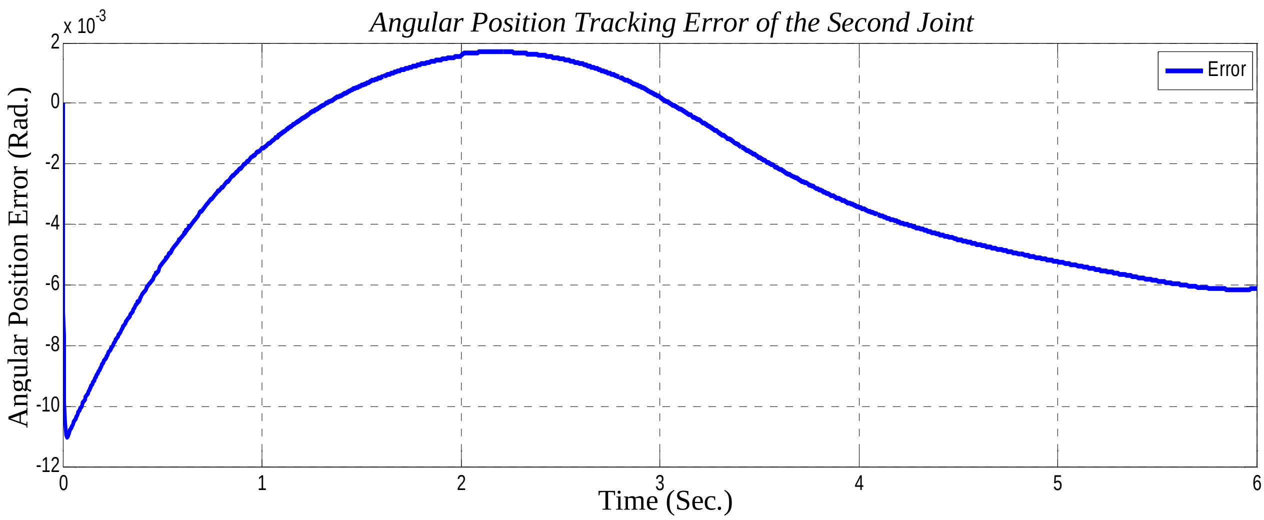


Figure (4.7): *Position Tracking Error Curve of the Second Joint*

Figure (4.8) shows the position tracking curve of the third joint, a linear function was chosen as a desired trajectory.

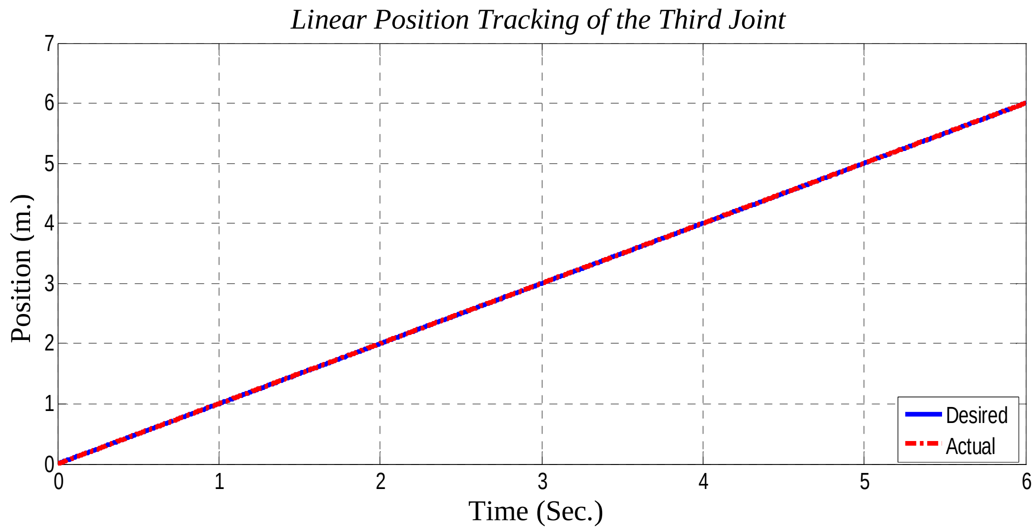


Figure (4.8): *Position Tracking Curve of the Third Joint*

As the figure show, the third joint tracks the desired path accurately, with very small tracking error as shown in figure (4.9).

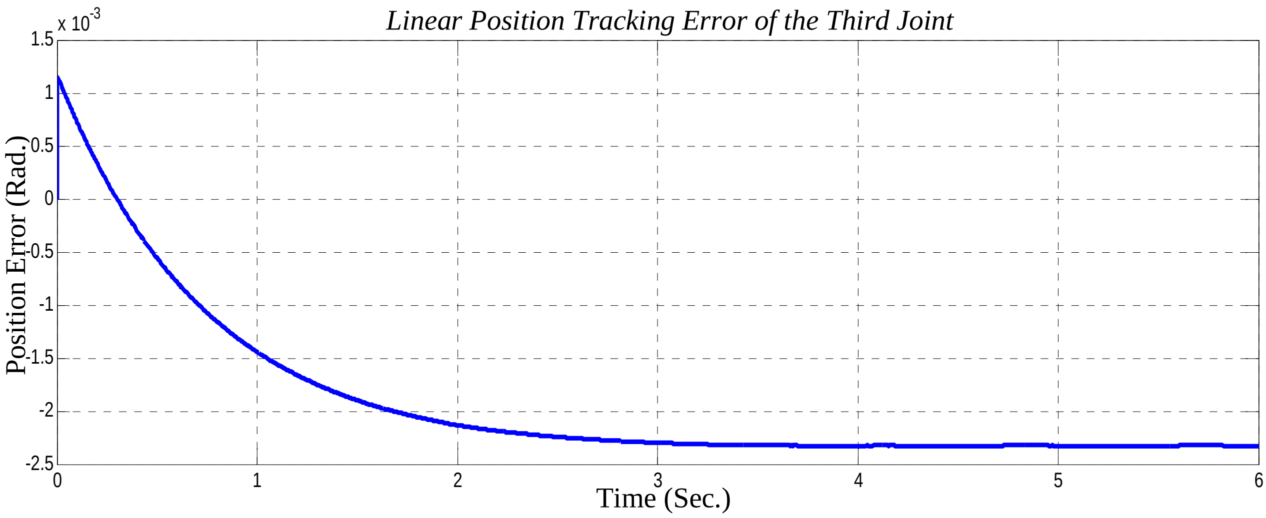


Figure (4.9): *Position Tracking Error Curve of the Third Joint*

Figure (4.10) shows the velocity tracking curve of the first joint, the derivative of the position desired function was taken as a desired velocity trajectory.

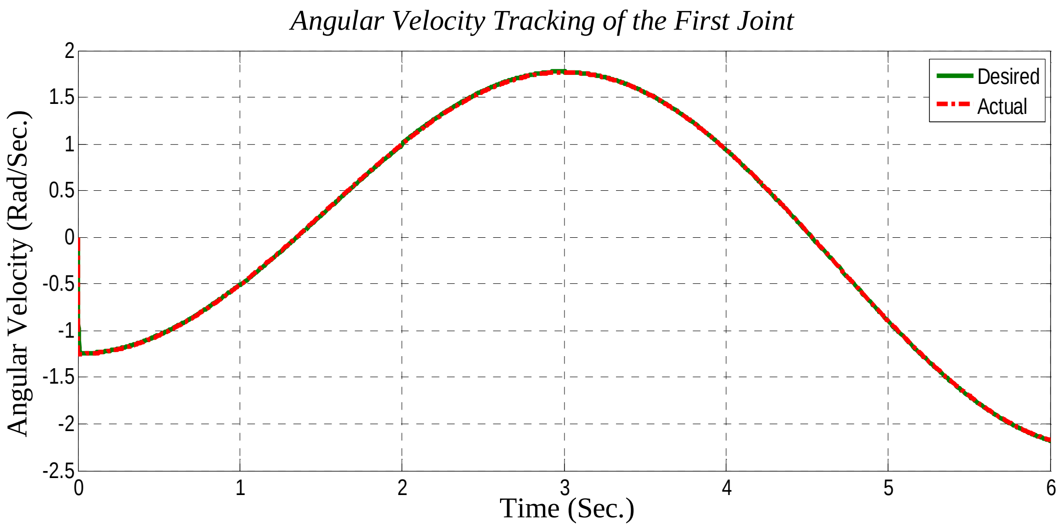


Figure (4.10): *Velocity Tracking Curve of the First Joint*

The velocity tracking error is very small as shown in figure (4.11)

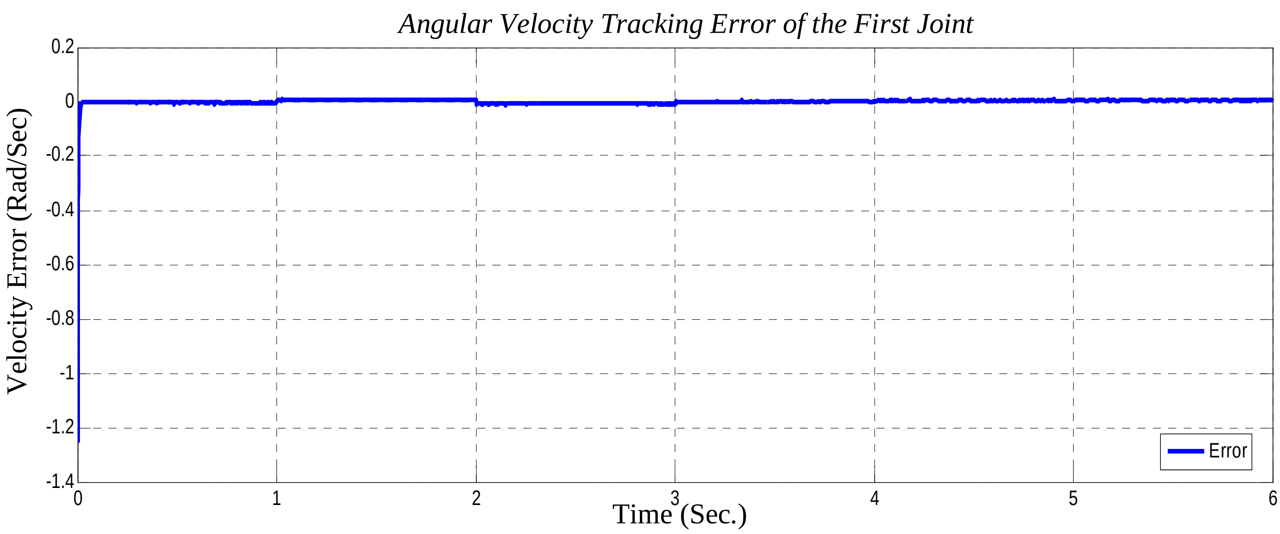


Figure (4.11): *Velocity Tracking Error Curve of the First Joint*

Figure (4.12) shows the velocity tracking curve of the second joint, the derivative of the position desired function was taken as a desired velocity trajectory.

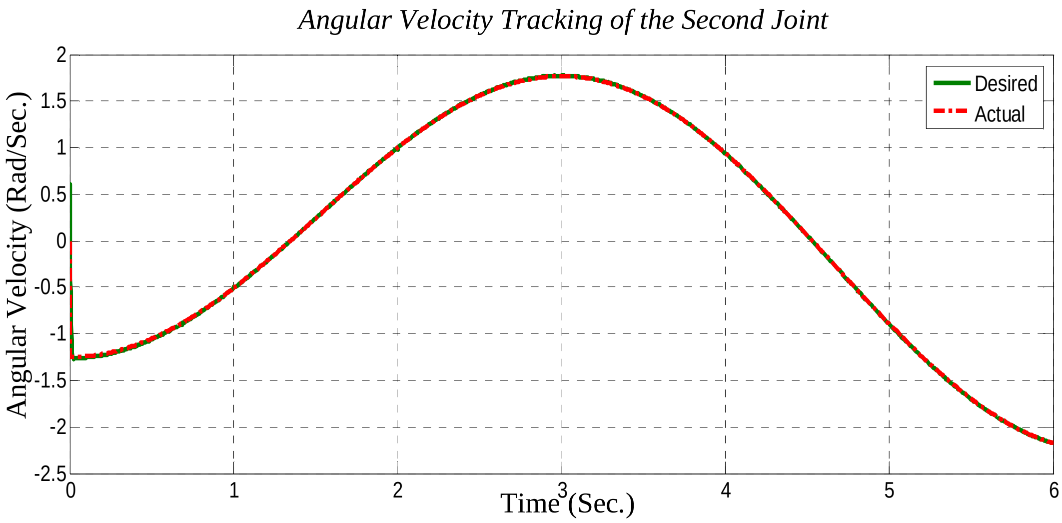


Figure (4.12): *Velocity Tracking Curve of the Second Joint*

The velocity tracking error of the second joint is very small as shown in figure (4.13)

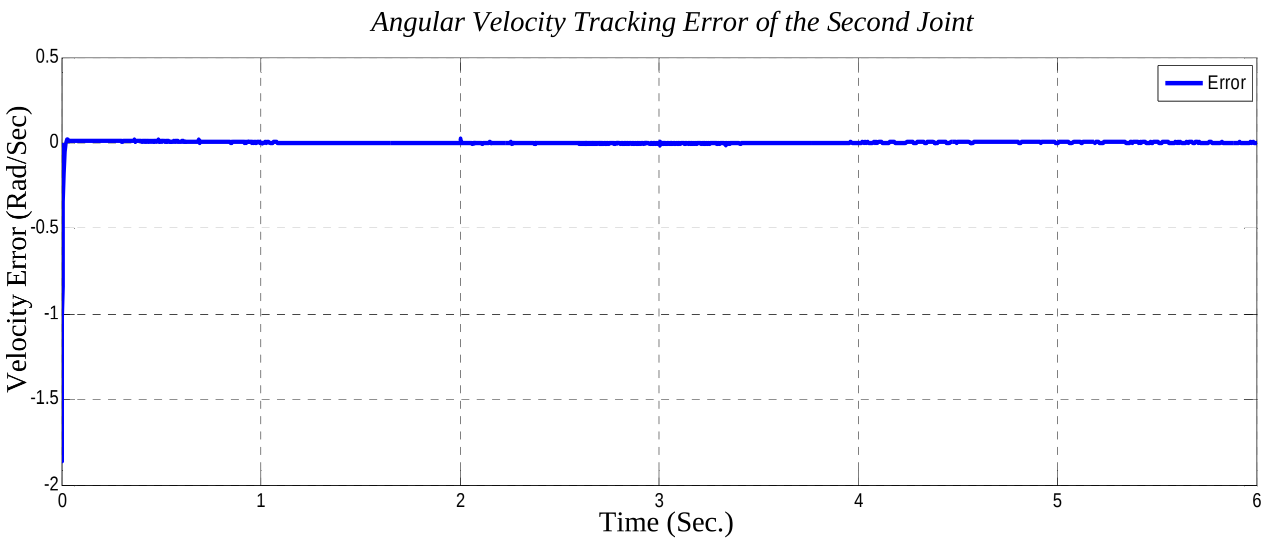


Figure (4.13): *Velocity Tracking Error Curve of the Second Joint*

Figure (4.14) shows the velocity tracking curve of the third joint, also the derivative of the position desired function was taken as a desired velocity trajectory.

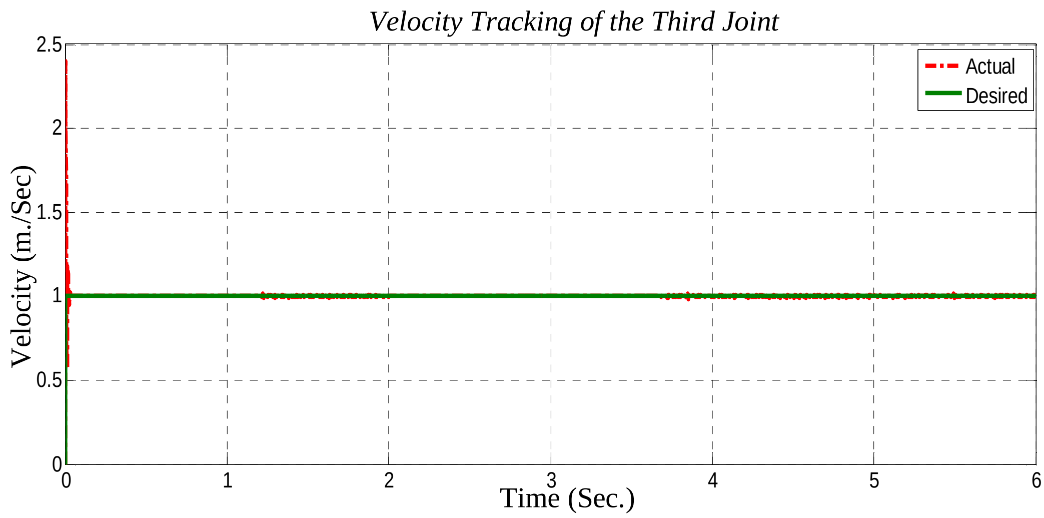


Figure (4.14): *Velocity Tracking Curve of the Third Joint*

The velocity tracking error of the second joint is very small as shown in figure (4.15)

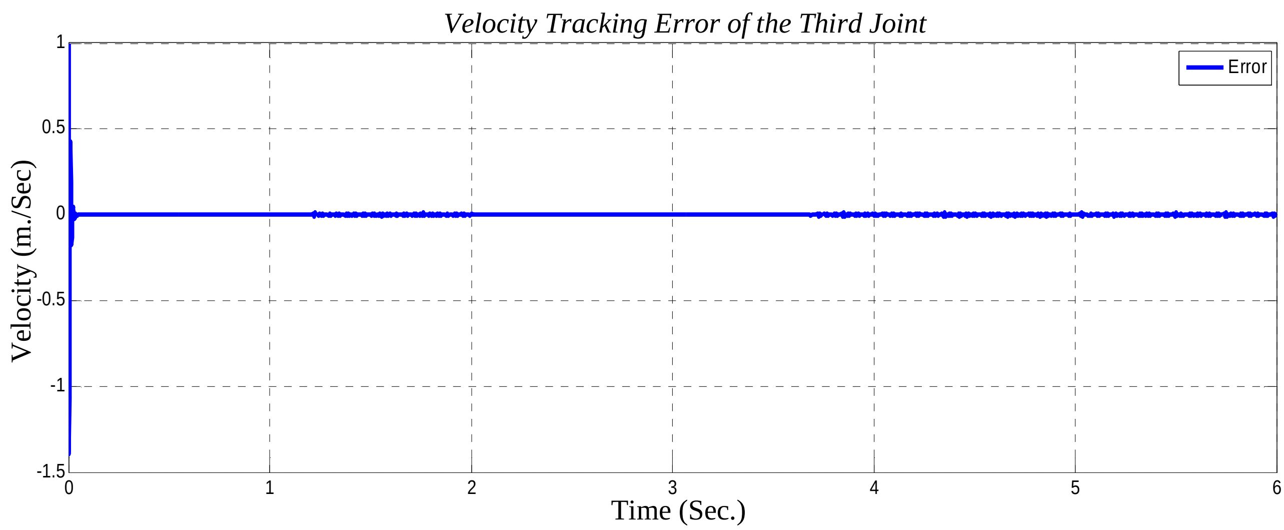


Figure (4.15): *Velocity Tracking Error Curve of the Third Joint*

The control surface of (ANNIT2FL) controller proposed in [1] shows that the error of tracking is big when compare with the tracking error in this thesis, the error in [1] is "0.1". The results in [4] shows that the position tracking error is about "0.2" while the velocity tracking error is about "0.1", and the work presented in [5] shows that the error of (NFC) is unacceptable. But in this work the error is very small; it is “0.012” at most.

The objective of this thesis was to control 3-DOF robot arm to track a desired path with minimum error that means the tracking path from the initial position to the final position was considered. It is clear that the error generates from the use of TSK fuzzy model is very small when comparing with the errors in the other methods.

The work presented in [6] was interested in control Lynx6 robot arm to reach the specified location with minimum error without regarding the tracking path from the initial position to the final position. But infinite number of trajectories to move from initial point to final one, then the main difference between this thesis and the work presented in [6] is the consideration of the tracking path.

The work presented in [4] was described a fuzzy position control scheme for precise tracking of robot manipulator. This work similar to the work presented in this thesis, but the results proved that the algorithm used here is more effective than the one used in [4].

The precisely tracking and small errors is very important task which allow implementing and achieving all the functions and tasks that intended and required, like pick and place, assembly, and packaging applications or any application requires good tracking and precise automation.

Chapter V

Conclusion and

Future Work

This report presented a complete study for controlling robot manipulator. This process depends on two main sides; the first one is modeling the manipulator while the second one is controlling the manipulator.

The modeling process includes complete kinematics "*forward and inverse kinematics*" analyses of the robot. A complete mathematical model of three-link robot was developed.

Controlling process requires the designing of all constituents of controllers; this means identifying the fuzzy controller input and output, choosing the recommended control rule base.

The objective of this thesis was to control three-link robot manipulator to trace desired trajectory using Takagi-Sugeno fuzzy model. This model was applied using twenty-five rules. The controller use min-max inference mechanism.

The simulation results proved that the proposed fuzzy controller has good performances such as fast response and small errors for different desired trajectory functions and it can be extended to more degree-of-freedom robotic arm systems.

The proposed fuzzy controller was applied to tracking control of a three-link robot manipulator to verify the validity and effectiveness of the control scheme. From the simulation results, we conclude that the suggested controller can effectively achieve the trajectory tracking even for the system with very small tracking error.

TSK fuzzy model has many advantages; such as it is reduce the number of rules required by the Mamdani model, it has guaranteed continuity of the output surface and it is well-suited to mathematical analysis.

A future work can focuses on different types of controllers like adaptive fuzzy controller, and extended the system to more degree-of-freedom. It is recommended to use the tuning of fuzzy controller using adaptive techniques to improve the performance.

It is also recommended to use different types of models such as Larsen and Tsukamoto fuzzy models.

References

[1] Y. Sahin, M. Tinkir and A. Ankarali, "*Trajectory Planning and Adaptive Neural Network Based Interval Type-2 Fuzzy Logic Controller Design of 3-DOF Robot*" IEEE Journal of Robotics and Automation, pp 114-119, 2011.

[2] Young-Wan Cho, Eung-Sun Kim, Ki-Chul Lee and Mignon Park, "*Tracking Control of a Robot Manipulator Using a Direct Model Reference Adaptive Fuzzy Control*" International Conference on Intelligent Robots and Systems, pp 100-105, 1999.

[3] Rong-Jong Wai and Zhi-Wei Yang, "*Adaptive Fuzzy-Neural-Network Control of Robot Manipulator Using T-S Fuzzy Model Design*" IEEE International Conference on Fuzzy Systems, pp 90-97, FUZZ 2008.

[4] St.Joseph's, "*Precise Tracking Control of Robot Manipulator Using Fuzzy Logic*" DARH2005 Conference, Session 4.1, pp 1-8, 2005.

[5] Jafar Tavoosi, Majid Alaei and Behrouz Jahani, "*Neuro – Fuzzy Controller for Position Control of Robot Arm*" 5th SASTech, Khavaran Higher-education Institute, Mashhad, Iran. May 12-14, 2011.

[6] A. Alassar, "*Modeling and Control of 5DOF Robot Arm Using Supervisory Control*" Master thesis, Faculty of Electrical Engineering, Islamic University, Gaza, March 2010.

[7] H. K. Lam, T. H. Lee, F. H. F. Leung and P. K. S. Tam, "*Fuzzy Model Reference Control of Wheeled Mobile Robots*" The 27th Annual Conference of the IEEE Industrial Electronics Society, pp 570-573, 2011.

[8] Wen-Jer Chang, Wei-Han Huang and Wei Chang, "*Fuzzy Control of Inverted Robot Arm with Perturbed Time-Delay Affine Takagi-Sugeno Fuzzy Model*" IEEE International Conference on Robotics and Automation Roma, Ital, pp 4380-4385 10-14 April 2007.

[9] M. I. H. Nour, J. Ooi, , and K. Y. Chan, "*Fuzzy Logic Control vs. Conventional PID Control of an Inverted Pendulum Robot*" International Conference on Intelligent and Advanced Systems, pp 209-214, 2007.

[10] M. AbuQassem, "*Simulation and Interfacing of 5 DOF Educational Robot Arm*" Master thesis, Faculty of Electrical Engineering, Islamic University, Gaza, June 2010.

[11] G.Chen and T. T. Pham, "*Introduction to Fuzzy sets, Fuzzy Logic and Fuzzy Control System*" By CRC press LLC, 2011.

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

[13] S. Kucuk and Z. Bingul, "*Industrial Robotics Theory Modeling and Control*" Edition by S. Cubero, pp 119-120, Jan. 2007.

[14] M. W. Spong, S. Hutchinson, and M. Vidyasagar, "*Robot Modeling and Control*" First edition, pp 68

[15] J. Angeles, "*Fundamental of Robotics Mechanical Systems, Theory, Methods and Algorithms*" Third edition, pp 269-285

[16] J.M. Hollerbach, "*A Recursive Lagrangian Formulation of Manipulator Dynamics and Comparative Study of Dynamics Formulation Complexity*" IEEE Transaction on Systems, Man and CYBERNETCS, vol. SMC-10 no.11 November 2011

[17] J. Foran, "*Optimization of a Fuzzy Logic Controller using Genetic Algorithms*" summer 2010.

[18] E.H. Mamdani, "*Applications of fuzzy logic to approximate reasoning using linguistic synthesis*" IEEE Transaction on Computers, vol. 26, No. 12, pp. 1182-1191, Dec. 1977.

[19] K. M. Passino and S. Yurkovich, "*Fuzzy Control*" Addison Wesely Longman, 1998.

[20] T. Terano, K. Asia and M. Seugeno, "*Fuzzy Systems Theory and its Applications*" Academic Press, pp. 34-59, 1992.

[21] T. J. Ross "*Fuzzy Logic with Engineering Applications*" 3ed edition, University of new Mexico, USA 2010.

[22] Y. J. Park, H. S. Cho and D. H. Cha "*Genetic Algorithm-Based Optimization of Fuzzy Logic Controller Using Characteristic Parameters*" IEEE International Conference on Evolutionary Computation, pp 831-836, 1995.

[23] L. X. Wang, J. M. Mendel "*Generating Fuzzy Rules by Learning from Examples*" IEEE Transaction on Systems, Man and Cybernetics, vol. 22, No. 6, pp 1414-1427, 1992

[24] B. Hamed "*Fuzzy Control Lectures*" Electrical Engineering Department, the Islamic university of Gaza, 2011

[25] O.A.S. Youssef "*Application of Fuzzy Inference Mechanisms to Power System Relaying*" Sues Canal University, Egypt, 2004

[26] R. Kayalvizhi, S.P. Natarajan, and S. Anbumalar "*Development of Fuzzy Logic Control for Paralleled Positive Output Elementary Luo Converter*" Annamalai University, India, 2006

[27] R. Kayalvizhi, S.P. Natarajan, and S. Anbumalar "*Fuzzy Control of Paralleled Negative Output Luo Converter*" Annamalai University Journal of Engineering and Technology, pp. 90-93,2005

[28] E.H. Mamdani, H.J. Efstathiou and K. Sugiyama "*Development of Fuzzy Logic Control*" The 23rd IEEE Conference on decision and control, vol. 23, pp. 888-893, Dec. 1984.

[29] H. Ying "*The Takagi-Sugeno Fuzzy Controller Using the Simplified Linear rules Are Nonlinear variable Gain Controllers*" Automatica, vol. 34, pp. 157-167, 1998.

[30] M. Jamshidi, N. Vadiee, "*Introduction to Fuzzy Logic Using Matlab*" Springer, pp. 97-107, 2007.

[31] S.N. Sivanandam, S. Sumathi and S.N. Deepa "*Introduction to Fuzzy Logic Using Matlab*" Springer, pp. 97-107,

Appendices

APPENDICES

APPENDIX A: FORWARD AND INVERSE KINEMATICS ANALYSIS

A.1 Denavit-Hertenberg Convention Parameters

According to *Denavit-Hertenberg* convention, the table of the parameters display as follow:

Joint	*Type*	q_i	α_i	a_i	d_i
1	*Revolute*	θ_1	*0*	*0.5*	*0*
2	*Revolute*	θ_2	*0*	*0.5*	*0*
3	*Prismatic*	d_3	*0*	*0*	*0.3*

A.2 Link Transformation of Three-link Robot Manipulator

The derivation of link transformation matricies $T^{i-1}_{\ \ i}$

$$T^0_1 = \begin{bmatrix} C_1 & -S_1 & 0 & l_1C_1 \\ S_1 & C_1 & 0 & l_1C_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (A.1)$$

$$T^2_3 = \begin{bmatrix} C_1 & -S_1 & 0 & l_1C_1 \\ S_1 & C_1 & 0 & l_1C_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (A.2)$$

$$T^1_2 = \begin{bmatrix} C_2 & -S_2 & 0 & l_2C_2 \\ S_2 & C_1 & 0 & l_2C_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (A.3)$$

$$H_0^2 = T_1^0 . T_2^1 = \begin{bmatrix} C_{12} & -S_{12} & 0 & l_1C_1 + l_2C_{12} \\ S_{12} & C_{12} & 0 & l_1C_1 + l_2C_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (A.4)$$

The final transformation matrix T_3^0 for three-link robot:

$$H_3^0 = H_2^0 . T_3^2 = T_1^0 . T_2^1 . T_3^2 = \begin{bmatrix} C_{12} & -S_{12} & 0 & l_1C_1 + l_2C_{12} \\ S_{12} & C_{12} & 0 & l_1C_1 + l_2C_{12} \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (A.5)$$

According to the equation (2.4),

$$T_{end-effector}^{base} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Then

$$r_{11} = C_{12}\ , \qquad r_{12} = -S_{12}, \qquad r_{13} = 0$$

$$r_{21} = S_{12}\ , \qquad r_{22} = C_{12}, \qquad r_{23} = 0 \qquad (A.6)$$

$$r_{31} = S_{12}\ , \qquad r_{32} = C_{12}, \qquad r_{33} = 1$$

With

$$C_i = Cos\ \theta_i \quad , \quad C_{ij} = Cos(\theta_i + \theta_j)$$
$$S_i = Sin\ \theta_i \quad , \quad S_{ij} = Sin(\theta_i + \theta_j)$$

$$P_x = l_2C_{12} + l_1C_1\ , \qquad P_y = l_2C_{12} + l_1S_1\ , \qquad P_z = d_3$$

A.3 Inverse of link Transformation Matrices

$$(A_1^0)^{-1}=\begin{bmatrix} C_1 & S_1 & 0 & -C_1 l_1(C_1+S_1) \\ -S_1 & C_1 & 0 & -C_1 l_1(C_1-S_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (A.7)$$

$$(A_2^1)^{-1}=\begin{bmatrix} C_2 & -S_2 & 0 & C_2 l_2 \\ S_2 & C_1 & 0 & C_2 l_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (A.8)$$

$$(A_3^2)^{-1}=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (A.9)$$

Low of inverse matrices: For 3x3 matrix

$$A = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \qquad (A.10)$$

The inverse of A

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

One of the simple ways to solve the inverse kinematics problems is by using geometric solution. With this method, cosine law can be used. A tow planer manipulator will be used to review this kinematic problem as in the Figure (A.1).

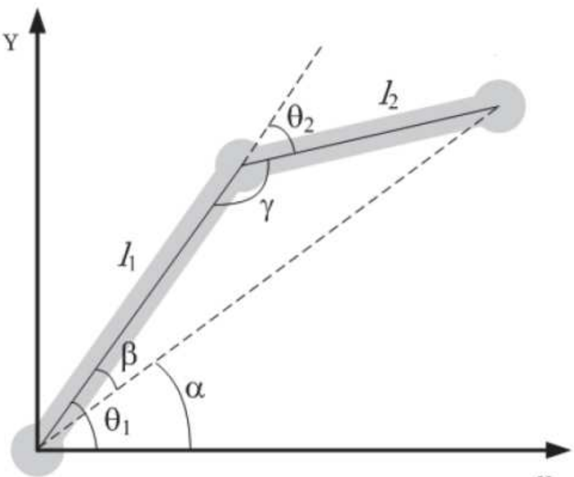


Figure (A.1): *Geometric of two revolute links of the manipulator*

By applying the cosines law, the following form obtained

$$(x^2 + y^2) = l_1^2 + l_2^2 - 2l_1l_2\cos(180 - \theta_2) \qquad (A.12)$$

Since $\cos(180 - \theta_2) = -\cos(\theta_2)$ then the equation (A.12) become,

$$(x^2 + y^2) = l_1^2 + l_2^2 - 2l_1l_2\cos(\theta_2) \qquad (A.13)$$

Then θ_2 determined by taking the inverse cosine,

$$\theta_2 = acos\left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2}\right) \qquad (A.14)$$

By applying the sinus law, the following form obtained

$$\frac{\sin(\beta)}{l_2} = \frac{\sin(\gamma)}{\sqrt{x^2+y^2}}$$

and (A.15)

$$\alpha = atan\left(\frac{y}{x}\right)$$

Where $\sin(\gamma) = \sin(180-\theta_2) = \sin(\theta_2)$. Then the equation (A.15) become

$$\beta = asin\left(\frac{l_2 \sin(\theta_2)}{\sqrt{x^2+y^2}}\right) \qquad (A.16)$$

Since $\theta_1 = \alpha + \beta$, then

$$\theta_1 = asin\left(\frac{l_2 \sin(\theta_2)}{\sqrt{x^2+y^2}}\right) + atan\left(\frac{y}{x}\right) \qquad (A.17)$$

A.4 Movement effect matrices

According to the equation (2.28) the movement effect matrices are

$$U_{11} = \begin{bmatrix} -S_1 & -C_1 & 0 & -l_1S_1 \\ C_1 & -S_1 & 0 & l_1C_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (A.18)$$

$$U_{22} = \begin{bmatrix} -S_{12} & -C_{12} & 0 & -l_2S_{12} \\ C_{12} & -S_{12} & 0 & l_2C_{12} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (A.19)$$

$$U_{33} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (A.20)$$

$$U_{12}=U_{21} = \begin{bmatrix} -S_{12} & -C_{12} & 0 & -l_{12}C_{12} - l_1S_1 \\ C_{12} & S_{12} & 0 & l_{12}C_{12} + l_1C_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (A.21)$$

$$U_{13}=U_{31} = \begin{bmatrix} -S_{12} & -C_{12} & 0 & -l_2S_{12} - l_1S_1 \\ C_{12} & -S_{12} & 0 & l_2C_{12} + l_1C_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (A.22)$$

$$U_{23}=U_{32} = \begin{bmatrix} -S_{12} & -C_{12} & 0 & -l_2S_{12} \\ C_{12} & -S_{12} & 0 & l_2C_{12} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (A.23)$$

A.5 velocities effects matrices

According to the equation (2.33), the velocities intersection effects matrices between the joints can be calculated as follow

$$U_{111} = Q_1.Q_1.T_1^0 = \begin{bmatrix} -C_1 & S_1 & 0 & -l_1C_1 \\ -S_1 & C_1 & 0 & -l_1S_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (A.24)$$

$$U_{211} = Q_1.Q_1.T_2^0 = \begin{bmatrix} -C_{12} & S_{12} & 0 & -l_1C_1 - l_2C_{12} \\ -S_{12} & -C_{12} & 0 & -l_1S_1 - l_2S_{12} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (A.25)$$

$$U_{311} = Q_1.Q_1.T_3^0 = \begin{bmatrix} -C_{12} & S_{12} & 0 & -l_1C_1 - l_2C_{12} \\ -S_{12} & -C_{12} & 0 & -l_1S_1 - l_2S_{12} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.26}$$

$$U_{212} = Q_1.T_1^0.Q_2.T_2^1 = \begin{bmatrix} -C_{12} & S_{12} & 0 & -l_2C_{12} \\ -S_{12} & -C_{12} & 0 & -l_2S_{12} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.27}$$

$$U_{222} = T_1^0.Q_1.Q_2.T_2^1 = \begin{bmatrix} -C_{12} & S_{12} & 0 & -l_2C_{12} \\ -S_{12} & -C_{12} & 0 & -l_2S_{12} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.28}$$

$$U_{312} = Q_1.T_2^0.Q_2.T_3^1 = \begin{bmatrix} -C_{12} & S_{12} & 0 & -l_2C_{12} \\ -S_{12} & -C_{12} & 0 & -l_2S_{12} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.29}$$

$$U_{313} = Q_1.T_2^0.Q_3.T_3^2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.30}$$

$$U_{322} = T_1^0.Q_2.Q_2.T_3^2 = \begin{bmatrix} -C_{12} & S_{12} & 0 & -l_2C_{12} \\ -S_{12} & -C_{12} & 0 & -l_2S_{12} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.31}$$

$$U_{323} = T_1^0.Q_2.T_2^1.Q_3.T_3^2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.32}$$

$$U_{333} = T_2^0.Q_3.Q_3.T_3^2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.33}$$

A.6 Christofell symbols vector

According to the equation (2.48), Christofell symbols can be computed as follow

1. For $i = 1$

$$\begin{array}{lll} c_{111} = 0 & c_{112} = -\frac{1}{2} m_2 l_1 l_2 S_2 - m_3 l_1 l_2 S_2 & c_{113} = 0 \\ c_{121} = -\frac{1}{2} m_2 l_1 l_2 S_2 - m_3 l_1 l_2 S_2 & c_{122} = -\frac{1}{2} m_2 l_1 l_2 S_2 - m_3 l_1 l_2 S_2 & c_{123} = 0 \\ c_{131} = 0 & c_{132} = 0 & c_{133} = 0 \end{array} \tag{2.34}$$

2. For $i = 2$

$$\begin{array}{lll} c_{211} = \frac{1}{2} m_2 l_1 l_2 S_2 + m_3 l_1 l_2 S_2 & c_{212} = 0 & c_{213} = 0 \\ c_{221} = 0 & c_{222} = 0 & c_{223} = 0 \\ c_{231} = 0 & c_{232} = 0 & c_{233} = 0 \end{array} \tag{2.35}$$

3. For $i = 3$

$$\begin{array}{lll} c_{311} = 0 & c_{312} = 0 & c_{313} = 0 \\ c_{321} = 0 & c_{322} = 0 & c_{323} = 0 \\ c_{331} = 0 & c_{332} = 0 & c_{333} = 0 \end{array} \tag{2.36}$$

According to the equation (2.49),

$$\begin{aligned} h_{11} &= (-2m_3 l_1 l_2 S_2 - m_2 l_1 l_2 S_2)\dot{\theta}_1 \dot{\theta}_2 - (\frac{1}{2} m_2 l_1 l_2 S_2 - m_3 l_1 l_2 S_2)\dot{\theta}_2^2 \\ h_{21} &= (\frac{1}{2} m_2 l_1 l_2 S_2 - m_3 l_1 l_2 S_2)\dot{\theta}_1^2 \\ h_{31} &= 0 \end{aligned} \tag{A.37}$$

Then the centrifugal and coriolis vector can be written as

$$h = \begin{bmatrix} (-2m_3l_1l_2S_2 - m_2l_1l_2S_2)\dot{\theta}_1\dot{\theta}_2 - (\frac{1}{2}m_2l_1l_2S_2 - m_3l_1l_2S_2)\dot{\theta}_2^2 \\ (\frac{1}{2}m_2l_1l_2S_2 - m_3l_1l_2S_2)\dot{\theta}_1^2 \\ 0 \\ \end{bmatrix} \tag{A.38}$$

A.6 Inertia Tensor

According to the equation (2.38), the inertia moment matrices can be written as follow. For the two rotate joints, the inertia moments over the x and y axis are nil, but over the z axis، $I_{zz} = \frac{1}{3}m_2l_2^2$ then the inertia tensor expressed as

$$J_1 = \begin{bmatrix} \frac{1}{3}m_1l_1^2 & 0 & 0 & -\frac{1}{2}m_1l_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{2}m_1l_1 & 0 & 0 & m_1 \end{bmatrix} \tag{A.39}$$

$$J_2 = \begin{bmatrix} \frac{1}{3}m_2l_2^2 & 0 & 0 & -\frac{1}{2}m_2l_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{2}m_2l_2 & 0 & 0 & m_2 \end{bmatrix} \tag{A.40}$$

$$J_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & m_3 \end{bmatrix} \tag{A.41}$$

For the prismatic joint, there is no rotation, then the inertia moment over x, y an z axis are nil, but over the mass center equals m_3, then the inertia matrix

$$D(q)=\begin{bmatrix} \frac{1}{3}m_1l_1^2+\frac{1}{3}m_2l_2^2+m_2l_1^2+m_2l_1l_2C_2+m_2l_2^2+2m_3l_1l_2C_2+m_3l_1^2 & \frac{1}{3}m_2l_2^2+\frac{1}{2}m_2l_1l_2C_2+m_3l_2^2+m_3l_1l_2C_2 & 0 \\ \frac{1}{3}m_2l_2^2+\frac{1}{2}m_2l_1l_2C_2+m_3l_2^2+m_3l_1l_2C_2 & \frac{1}{3}m_2l_2^2+m_3l_2^2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \quad (A.42)$$

The motor proper inertia

$$J=\begin{bmatrix} J_1 & 0 & 0 \\ 0 & J_2 & 0 \\ 0 & 0 & J_3 \end{bmatrix}=\begin{bmatrix} N_1^2J_1 & 0 & 0 \\ 0 & N_2^2J_2 & 0 \\ 0 & 0 & N_3^2J_3 \end{bmatrix} \quad (A.43)$$

Then the final inertia matrix can be written as

$$D(q)=\begin{bmatrix} \frac{1}{3}m_1l_1^2+\frac{1}{3}m_2l_2^2+m_2l_1^2+m_2l_1l_2C_2+m_2l_2^2+2m_3l_1l_2C_2+m_3l_1^2+N_1^2J_1 & \frac{1}{3}m_2l_2^2+\frac{1}{2}m_2l_1l_2C_2+m_3l_2^2+m_3l_1l_2C_2 & 0 \\ \frac{1}{3}m_2l_2^2+\frac{1}{2}m_2l_1l_2C_2+m_3l_2^2+m_3l_1l_2C_2 & \frac{1}{3}m_2l_2^2+m_3l_2^2+N_2^2J_2 & 0 \\ 0 & 0 & m_3+N_3^2J_3 \end{bmatrix} (A.44)$$

A.7 Friction vector

The friction vector can be written as follow

$$f(\dot{q})=\begin{bmatrix} N_1^2(f_1\dot{q}_1+f_4 sign(\dot{q}_1)) \\ N_2^2(f_2\dot{q}_2+f_5 sign(\dot{q}_2)) \\ N_3^2(f_3\dot{q}_3+f_6 sign(\dot{q}_3)) \end{bmatrix} \quad (A.45)$$

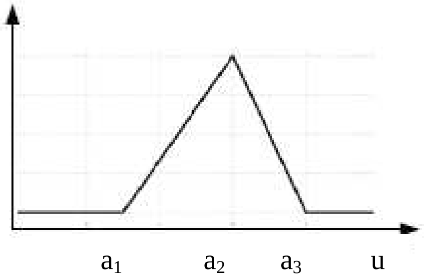
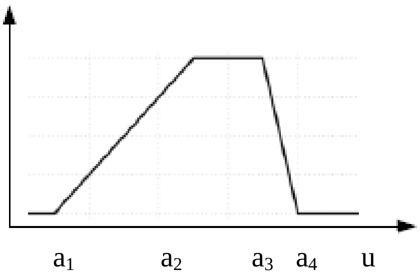
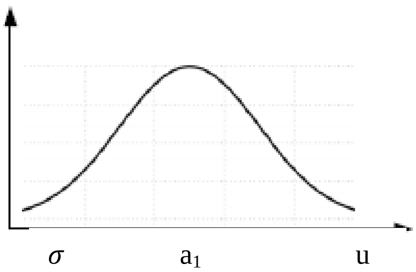
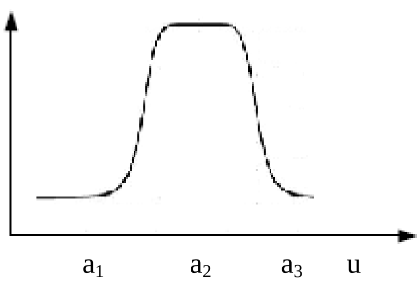
A.8 Gravity vector

For the two revolute joints, the gravity term equal zero, but for the prismatic joint the gravity term equal 9.8062 m/s^2 then the gravity vector can be written as:

$$g=\begin{bmatrix} 0 \\ 0 \\ -9.8062 \end{bmatrix} \quad (A.46)$$

APPENDIX B: FUZZY MEMBERSHIP FUNCTIONS AND RULE BASE

B.1 Membership Functions Types

Function Name	Example	Mathematical Description
Triangle		$\mu(u)=\begin{cases}0 & ,if\ \ u\le a_1\\ \dfrac{u-a_1}{a_2-a_1} & ,if\ \ a_1\le u\le a_2\\ \dfrac{a_3-u}{a_3-a_2} & ,if\ \ a_2\le u\le a_3\\ 0 & ,if\ \ u\ge a_3\end{cases}$
Trapezoidal		$\mu(u)=\begin{cases}0 & ,if\ \ u\le a_1\\ \dfrac{u-a_1}{a_2-a_1} & ,if\ \ a_1\le u\le a_2\\ 1 & ,if\ \ a_2\le u\le a_3\\ \dfrac{a_3-u}{a_3-a_2} & ,if\ \ a_3\le u\le a_4\\ 0 & ,if\ \ u\ge a_4\end{cases}$
Gaussian		$\mu(u)=e^{\frac{-(u-a)}{2\sigma^2}}$
Generalized Bell		$\mu(u)=\dfrac{1}{1+\left\lvert\dfrac{u-a_3}{a_1}\right\rvert^{2a_2}}$

B.2 Rule Base

$$
\begin{array}{l}
If\ \ e\ is\ NL\ \ and\ \ \Delta e\ is\ NL\ \ Then\ \ Position\ is\ NL \\
If\ \ e\ is\ NL\ \ and\ \ \Delta e\ is\ NS\ \ Then\ \ Position\ is\ NL \\
If\ \ e\ is\ NL\ \ and\ \ \Delta e\ is\ ZE\ \ Then\ \ Position\ is\ NL \\
If\ \ e\ is\ NL\ \ and\ \ \Delta e\ is\ PS\ \ Then\ \ Position\ is\ NS \\
If\ \ e\ is\ NL\ \ and\ \ \Delta e\ is\ PL\ \ Then\ \ Position\ is\ ZE
\end{array}
$$

$$
\begin{array}{l}
If\ \ e\ is\ NS\ \ and\ \ \Delta e\ is\ NL\ \ Then\ \ Position\ is\ NL \\
If\ \ e\ is\ NS\ \ and\ \ \Delta e\ is\ NS\ \ Then\ \ Position\ is\ NL \\
If\ \ e\ is\ NS\ \ and\ \ \Delta e\ is\ ZE\ \ Then\ \ Position\ is\ NS \\
If\ \ e\ is\ NS\ \ and\ \ \Delta e\ is\ PS\ \ Then\ \ Position\ is\ ZE \\
If\ \ e\ is\ NS\ \ and\ \ \Delta e\ is\ PL\ \ Then\ \ Position\ is\ PS
\end{array}
$$

$$
\begin{array}{l}
If\ \ e\ is\ ZE\ \ and\ \ \Delta e\ is\ NL\ \ Then\ \ Position\ is\ NL \\
If\ \ e\ is\ ZE\ \ and\ \ \Delta e\ is\ NS\ \ Then\ \ Position\ is\ NS \\
If\ \ e\ is\ ZE\ \ and\ \ \Delta e\ is\ ZE\ \ Then\ \ Position\ is\ ZE \\
If\ \ e\ is\ ZE\ \ and\ \ \Delta e\ is\ PS\ \ Then\ \ Position\ is\ PS \\
If\ \ e\ is\ ZE\ \ and\ \ \Delta e\ is\ PL\ \ Then\ \ Position\ is\ PL
\end{array}
$$

$$
\begin{array}{l}
If\ \ e\ is\ PS\ \ and\ \ \Delta e\ is\ NL\ \ Then\ \ Position\ is\ NS \\
If\ \ e\ is\ PS\ \ and\ \ \Delta e\ is\ NS\ \ Then\ \ Position\ is\ ZE \\
If\ \ e\ is\ PS\ \ and\ \ \Delta e\ is\ ZE\ \ Then\ \ Position\ is\ PS \\
If\ \ e\ is\ PS\ \ and\ \ \Delta e\ is\ PS\ \ Then\ \ Position\ is\ PL \\
If\ \ e\ is\ PS\ \ and\ \ \Delta e\ is\ PL\ \ Then\ \ Position\ is\ PL
\end{array}
$$

$$
\begin{array}{l}
If\ \ e\ is\ PL\ \ and\ \ \Delta e\ is\ NL\ \ Then\ \ Position\ is\ ZE \\
If\ \ e\ is\ PL\ \ and\ \ \Delta e\ is\ NS\ \ Then\ \ Position\ is\ PS \\
If\ \ e\ is\ PL\ \ and\ \ \Delta e\ is\ ZE\ \ Then\ \ Position\ is\ PL \\
If\ \ e\ is\ PL\ \ and\ \ \Delta e\ is\ PS\ \ Then\ \ Position\ is\ PL \\
If\ \ e\ is\ PL\ \ and\ \ \Delta e\ is\ PL\ \ Then\ \ Position\ is\ PL
\end{array}
$$

e is the error, and Δe the change of error.