

T.C.

UNIVERSITY OF TURKISH AERONAUTICAL ASSOCIATION

SCIENCE AND TECHNOLOGY INSTITUTE

MINIMUM-TIME PATH PLANNING FOR ROBOT MANIPULATORS

USING PATH PARAMETER OPTIMIZATION WITH EXTERNAL

FORCE AND FRICTIONS

Ph.D. THESIS

Majdi F. Abdulati

1303947013

GRADUATE SCHOOL OF AERONAUTICAL AND ASTRONAUTICS

Mechanical and Aeronautical Engineering (English) (Doctorate)

February 2019



**UNIVERSITY OF TURKISH AERONAUTICAL ASSOCIATION
SCIENCE AND TECHNOLOGY INSTITUTE**

**MINIMUM-TIME PATH PLANNING FOR ROBOT MANIPULATORS
USING PATH PARAMETER OPTIMIZATION WITH EXTERNAL
FORCE AND FRICTIONS**

A Thesis

Presented to the Graduate School

Faculty of Engineering, Turkish Aeronautical Association University

In partial fulfilment of the

requirements for the Degree

Doctor of Philosophy in Mechanical Engineering

By

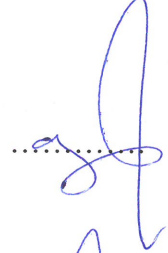
Majdi Farag Abdulati

1303947013

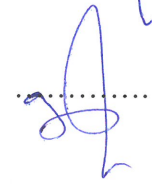
Thesis Supervisor: Asst. Prof. Dr. Habib GHANBARPOURASL

Majdi F. Abdulati has student number 1303947013 and he is enrolled in the Ph.D. Program at the Institute of Science and Technology, University of Turkish Aeronautical Association. After meeting all the required conditions of the related regulations, he has successfully accomplished the presentation of the thesis in front of the jury. The title of the thesis is: "MINIMUM-TIME PATH PLANNING FOR ROBOT MANIPULATORS USING PATH PARAMETER OPTIMIZATION WITH EXTERNAL FORCE AND FRICTIONS".

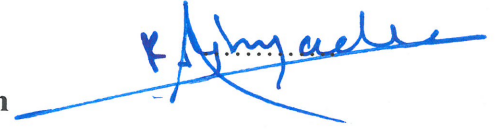
Thesis Supervisor: Asst. Prof. Dr. Habib GHANBARPOURASL
University of Turkish Aeronautical Association



Jury Members: Asst. Prof. Dr. Habib GHANBARPOURASL
University of Turkish Aeronautical Association




: Asst. Prof. Dr. Reza AGHAZADEH
University of Turkish Aeronautical Association



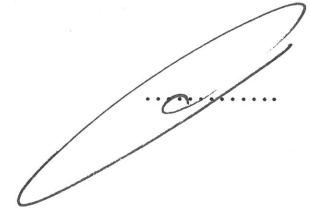
: Asst. Prof. Dr. Mohammad Mehdi GOMROKI
University of Turkish Aeronautical Association



: Asst. Prof. Dr. Munir ELFARRA
Ankara Yildirim Beyazit University



: Asst. Prof. Dr. Babek ERDEBILLI
Ankara Yildirim Beyazit University



Thesis Defense Date: 20. 2. 2019

UNIVERSITY OF TURKISH AERONAUTICAL ASSOCIATION

SCIENCE AND TECHNOLOGY INSTITUTE

STATEMENT OF NON-PLAGIARISM

I hereby declare that all information in my Ph.D. thesis titled: *Minimum-time path planning for robot manipulators using path parameter optimization with external force and frictions*, has been presented in accordance with the academic rules and ethical conduct. I also declare and certify with my honor that I have fully cited all the references of the literature sources utilized in this study.

20. 2. 2019

Majdi Farag Abdulati

1303947013



إهداء

بسم الله الرحمن الرحيم

قال الله تعالى: وقضى ربك ألا تعبدوا إلا إياه وبالوالدين احساناً

أهدي هذا العمل المتواضع إلى الوالدين الكريمين وزوجتي وأبنائي حفظهما الله

إلى إخوتي وأخواتي

حفظهم الله جميعاً وأجزل لهم المثوبة وأسأل الله أن يتولاهم في الدنيا والآخرة وان يجعلهم ممن إذا

أعطي شكر وإبنتي صبر وإذا أذنب استغفر وأن يسبغ عليهم نعمه ظاهرة وباطنه

إلى كل الأصدقاء والنفوس الطيبة

إلى كل أساتذتي إلى كل من درسني وعلمني وأعانني من قريب أو من بعيد

إلى من وسعتهم ذاكرتي ولم تسعهم منكرتي.

ACKNOWLEDGEMENT

I am and would always be deeply indebted to my supervisor Asst. Prof. Dr. Habib Ghanbarpourasl for his guidelines, encouragement, suggestions, constructive criticism and motivation. He has helped me at various stages during my Ph.D. studies and willingly sharing his knowledge and ideas with me. I appreciate his patience and understanding because it helped me during the hard times. I am so grateful to him.

I extend my sincere appreciation and gratitude to my parents, brothers and sisters for their continuous support and patience during my study. I want to thank my wife, who supported me during my studies until the completion of my degree. I would also like to express my gratitude and appreciation to my extended family and my friends for their assistance and encouragement.

I want to thank everyone at my university, especially the professors, who played a very important role while mentoring me. I am thankful to the Turkish government and people.

Although I have tried to acknowledge the contributions of everyone, who helped me, I regret in case if I missed the name of someone, who has important contributions. I acknowledge that this work couldn't have been possible without help and favours, for which, I'll remain grateful, and humble.

ABSTRACT

MINIMUM-TIME PATH PLANNING FOR ROBOT MANIPULATORS USING PATH PARAMETER OPTIMIZATION WITH EXTERNAL FORCE AND FRICTIONS

Majdi Farag Abdulati

Ph.D., Department of Mechanical and Aeronautical Engineering

Thesis Supervisor: Asst. Prof. Dr. Habib Ghanbarpourasl

February-2019, 100 pages

In the past, very few researchers focused their efforts on the path planning problem for those robots, which are used in the industrial processes. The main purpose of path planning solution is that it has significant engineering applications in the fields such as industrial robotics and mechatronics because now robots are extensively used in several industries including aerospace and automotive industries. The aim of this thesis is to presents new planning method for minimum-time trajectory, which is applicable in robotic manipulators. They are used in Cartesian-space because they have dynamic constraints, which consist of a desired path from the Cartesian-space to the manipulator while the external forces are subject to the input voltage of the actuators. The existing processes have many limitations and several constraints; however, the current thesis shows how to develop methods for minimum-time path planning keeping in view dynamics of robotics including torque and force limits. For this purpose, we used path parameter optimization algorithm to calculate the minimum-time control in the presence of external force and frictions. The main matter of interest in this work is a path with unknown path parameter. This parameter is a function of time and the unknown vector, and it is used for optimization. In this thesis, the optimization of dynamic system has been converted into an optimization problem, which is subject to the equations of motion as well as limitations of angular velocity, angular acceleration, angular jerk, input actuator torques, input voltage, and final-time, respectively. In order to manage the all-time for the presented algorithm, the final-time of the task has been divided into steps for making the process easier. The final-time is an additional unknown variable in the optimization problem. The algorithm attempts to minimize the final-time by optimizing the path parameter, so it is a function of time

with some unknown parameters. In the nutshell, the algorithm can have a smooth input voltage in an allowable range, after which, all the motion parameters remain smooth. In addition, the simulation study shows that the presented approach is useful for the trajectory planning for a manipulator, which follows a Cartesian reference path. This point of view is new and provides interesting results, some of which have been already reported in the existing literature.

Keywords: Minimum-time path planning, robot manipulators, trajectory planning, path parameter, dynamic constraints, constrained optimization.



ÖZET

DIŞ KUVVET VE SÜRTÜNME İLE YOL PARAMETRESİ OPTİMİZASYONU KULLANARAK ROBOTİK İŞLEMCİLER İÇİN MİNİMUM-ZAMANLI YOL PLANLAMASI

Majdi Farag Abdulati

Doktora, Makine ve Havacılık Mühendisliği Bölümü

Tez Danışmanı: Yrd. Doç. Dr. Habib Ghanbarpourasl

Şubat-2019, 100 sayfa

Çok az sayıda araştırmacı geçmişte endüstriyel süreçlerde kullanılan robotlar için yol planlama problemine odaklanmıştır. Yol planlama çözümünün temel önemi endüstriyel robot teknolojisi ve Mekatronik gibi alanlarda birçok önemli mühendislik uygulamasına sahip olmasından kaynaklanmaktadır çünkü robotlar günümüzde uzay ve otomotiv endüstrileri de dâhil olmak üzere birçok endüstride yaygın olarak kullanılmaktadır. Bu tezin amacı, robotik işlemlerde uygulanabilen minimum-zamanlı gidiş yolu ve yörünge için yeni bir planlama yöntemi sunmaktır. Dış kuvvetler aktüatörlerin giriş gerilimine maruz kalırken, Kartezyen-uzaydan işlemciler istendik bir yoldan oluşan dinamik kısıtlamaları olduğu için Kartezyen-uzayda kullanılırlar. Mevcut süreçlerin birçok sınırlılığı ve kısıtlamaları vardır; bununla birlikte, mevcut tez, tork ve kuvvet limitleri dahil olmak üzere robot teknolojisinin dinamiklerini göz önünde bulundurarak minimum-zamanlı yol planlama için yöntemlerin nasıl geliştirileceğini göstermektedir. Bu amaçla, dış kuvvet ve sürtünmenin varlığında minimum zaman kontrolünü hesaplamak için yol parametresi optimizasyon algoritmasını kullandık. Bu çalışmadaki ana ilgi konusu, bilinmeyen yol parametresi olan bir yoldur. Bu parametre zamanın ve bilinmeyen vektörün bir fonksiyonudur ve optimizasyon için kullanılır. Bu tezde, optimizasyon algoritması, hareket denklemleri ve bunun yanı sıra sırasıyla açısal hız, açısal ivme, açısal sarsıntı, giriş aktüatör torkları, giriş voltajı ve final zamanı sınırlamalarına tabii bir optimizasyon problemine dönüştürülmüştür. Sunulan algoritmanın planlama sürecini yönetmek için, görevin final zamanı süreci kolaylaştırmak için bölünmüştür. Final zamanı, optimizasyon probleminde bilinmeyen ilave bir değişkendir. Algoritma, yol parametresini optimize ederek final zamanını en aza indirmeye çalışır, bu nedenle bilinmeyen bazı parametrelerle zamanın bir fonksiyonuna dönüşür. Özet olarak, algoritma izin verilen

bir aralıktaki düzgün bir giriş voltajına sahip olabilir, buradan sonrada tüm hareket parametreleri tahmin edilebilir kalır. Buna ek olarak, simülasyon çalışması sunulan yaklaşımın bir Kartezyen referans yolunu takip eden işlemci için yörünge planlaması bakımından yararlı olduğunu göstermektedir. Bu bakış açısı yenidir ve bazıları mevcut literatürde daha öncen de bildirilmiş olan ilginç sonuçlar vermektedir.

Anahtar Kelimeler: Minimum zaman yol planlama, robot işlemciler, yörünge planlama, yol parametresi, dinamik kısıtlamalar, kısıtlı optimizasyon.



TABLE OF CONTENTS

ACKNOWLEDGEMENT	I
ABSTRACT.....	II
TABLE OF CONTENTS	VI
LIST OF ABBREVIATIONS	X
NOMENCLATURE.....	XII
LIST OF FIGURES	XVI
LIST OF TABLES	XVIII
CHAPTER 1: GENERAL INTRODUCTION.....	1
1.1 Research Background.....	1
1.2 Basics of Path Planning.....	4
1.3 Trajectory Planning Algorithm	6
1.4 Way Point Tracking Limitations.....	7
1.5 Joint Control System.....	8
1.6 Problem Description and Method	9
1.7 Problem Statement	9
1.8 Objectives Methodologies of the Thesis	10
1.8.1 Obtaining the Dynamic Model.....	11
1.8.2 Designing the Model.....	11
1.8.3 Simulation and Comparison of Results.....	11
1.9 Thesis Overview.....	11
CHAPTER 2: LITERATURE REVIEW	12
2.1 Introduction.....	12
2.2 Path Planning Problem.....	13
2.3 Trajectory Planning and Tracking	14

2.4 Path Planning with Actuator Dynamics.....	15
2.5 Minimum-Time Trajectory Planning Method.....	15
2.6 Path Planning with External Force and Friction.....	17
2.7 Robot Dynamic Constraint Algorithms.....	19
2.8 Position Control for Robot Manipulators.....	21
2.9 Robotic Motor Control.....	30
2.10 Summary.....	31
 CHAPTER 3: MATHEMATICAL MODELLING AND KINEMATIC	
ANALYSIS.....	33
3.1 Introduction.....	33
3.2 Dynamic Modelling of Robotic Manipulator.....	34
3.3 Kinematics of Robot Manipulator.....	35
3.3.1 Forward (Direct) Kinematics.....	37
3.3.1.1 Derivation of Denavit–Hartenberg (DH) Parameters.....	38
3.3.1.2 Derivation of the Homogenous Transformation Matrices.....	39
3.3.2 Inverse Kinematics.....	41
3.4 Dynamic Modelling of Robot Manipulator in General.....	42
3.4.1 Newton–Euler Formulation.....	43
3.4.2 Euler–Lagrange Equations.....	43
3.5 General Expression for Kinetic Energy.....	44
3.6 General Expression for Potential Energy.....	47
3.7 Equation of Motion.....	48
3.8 SCARA Robot Arm Coordination System and (DH) Parameters.....	49
3.8.1 Kinematic Modelling of 3-DOF SCARA Robot.....	49
3.8.2 SCARA Configuration: Revolute-Revolute-Prismatic (RRP).....	50

3.8.3 General Form of Transformation Matrix.....	51
3.8.4 Properties of Inertia Matrix.....	52
3.8.5 The Matrix of Centrifugal and Coriolis Forces.....	54
3.8.6 The Gravitational Torque Vector.....	54
CHAPTER 4: MINIMUM-TIME PATH PLANNING FOR ROBOT MANIPULATORS USING PATH PARAMETER OPTIMIZATION WITH EXTERNAL FORCE AND FRICTIONS.....	
4.1 Introduction.....	57
4.2 Dynamic Modelling of the Robot Manipulator with Friction.....	57
4.3 Friction Modelling.....	58
4.4 Actuator Dynamics.....	58
4.5 DC Motor System Modelling.....	59
4.6 Constrained Dynamic System of the Robot Manipulator.....	61
4.7 Definition of the Parametric Trajectory Optimization Problem.....	62
4.7.1 The Cost Function.....	63
4.7.2 Subject of the Constraints.....	63
4.8 Dynamic Parameter of Robotic Manipulator.....	63
4.8.1 Objective Function.....	64
4.8.2 The Conditions.....	65
4.8.3 Initial and Final Conditions.....	65
4.9 Managing the Constraints of the Optimization Problem of Robot Path Planning.....	67
CHAPTER 5: RESULTS AND DISCUSSIONS.....	
5.1 Introduction.....	69
5.2 Robot Simulation Model.....	69

5.3 Discussion on Simulation of Results..... 71

CHAPTER 6: CONCLUSIONS AND RECOMMENDATIONS..... 79

6.1 Conclusions..... 79

6.2 Recommendations and Future Work..... 80

REFERENCES..... 88

APPENDIX 97

Appendix [A]: Tables..... 97

Resume/CV.....100



LIST OF ABBREVIATIONS

DOF.....	Degrees-of-Freedom
DH.....	Denavit–Hartenberg
<i>n</i> -DOF.....	Number of Degrees-of-Freedom
DC.....	Direct Current
DMP.....	Dynamic Movement Primitive
DP.....	Desired Path
PTP.....	Point-to-Point
PF.....	Path-Following
PSO.....	Particle Swarm Optimization
PMP.....	Pontryagin's Maximum Principle
SCARA.....	Self Compliant Articulated Robotic Arm
SQP.....	Sequential Quadratic Programming
S.....	Sine
C.....	Cosine
CTC.....	Computed Torque Control
COG.....	Centre of Gravity
C-space.....	Configuration-Space
GA.....	Genetic Algorithm
AIS.....	Artificial Immune System
FFC.....	Feed Forward Controller
FK.....	Forward Kinematics
IK.....	Inverse Kinematics
RRP.....	Revolute-Revolute-Prismatic
ODE.....	Ordinary Differential Equation
EE.....	End-Effector
EB.....	Euler–Bernoulli
EBT.....	Elementary Beam Theory
MTC.....	Minimum-Time Control
WMR.....	Wheeled Mobile Robots
BLDM.....	Brushless DC Motor
FEM.....	Finite Element Method
AMM.....	Assumed Modes Model

MCM..... Monte Carlo Method
TP Trajectory Planning
TSP Travelling Salesman Problem
T-space Task Space
NE Newton–Euler
NN..... Neural Network
LE..... Lagrange–Euler
2-D Two-Dimensional Space
3-D Three-Dimensional Space



NOMENCLATURE

T	Transformation matrix
T_i^{i-1}	Homogeneous transformation matrix between two joints from frame i to frame $i - 1$
r_{ij}	Element of the rotational matrix between the reference systems i and j
R_i	Rotational matrix of the center frame of the link- i
$a_{i-1}, \alpha_{i-1}, d_i, \theta_i$	Denavit–Hartenberg (DH) parameters
a_{i-1}	Link length
α_{i-1}	Twist angle
d_i	Joint distance
θ_i	Joint angle for revolute joint i of the robot manipulator
O_0	Origin of frame
q_i	Joint angle variable
x, y, z	Cartesian coordinates of the robot manipulator
x_i	x axis of the i^{th} moving frame
y_i	y axis of the i^{th} moving frame
z_i	z axis of the i^{th} moving frame
I_i	Inertia tensor matrix
I_{xx}, I_{yy}, I_{zz}	The principal moments of inertia
m_1	The mass of link-1
m_2	The mass of link-2
m_3	The mass of link-3
l_1	The length of link-1
l_2	The length of link-2
d	The position of the end-effector
l_{c1}	Position of the center of mass link-1
l_{c2}	Position of the center of mass link-2
I_{c1}	Inertia tensor of link-1
I_{c2}	Inertia tensor of link-2
I_{c3}	Inertia tensor of link-3
v	Total linear velocity at the manipulator

K	Total kinetic energy
L	Lagrangian function of the robotic manipulator
U	Total potential energy
M	Mass
q, \dot{q}	Joints of position and angular velocity
\ddot{q}	Joint of angular acceleration
\dddot{q}	Joint of angular jerk
g	Vector of gravity
r	Cartesian joint position
\dot{r}	Cartesian joint velocity
\ddot{r}	Cartesian joint acceleration
C	Coriolis centrifugal torque vector
G	Gravity torque vector
n	The number of joints
T_m	$[n \times 1]$ Vector of the motor torque constant
N	$[n \times n]$ Matrix of gear transmission ratio
T	$[n \times 1]$ Total torque
T_c	Control torque
T_{ext}	External torque
T_f	Friction torque
F_{ext}	External force
J	Jacobian matrix
v	Normal velocity
f_n	Normal force
μ	Friction coefficient
\hat{n}_v	Unit vector along the velocity vector
K_c	Coulomb friction coefficient of the joint
K_v	Viscous friction coefficient of the joint
A, B, C, D	Coefficients of the interpolation functions
$sign$	Sign function
s_i	Sine of the i^{th} joint angle of the manipulator ($i = 1, 2, 3 \dots n$)
c_i	Cosine of the i^{th} joint angle of the manipulator ($i = 1, 2, 3 \dots n$)
s_{ij}	Sine of the sum of the i^{th} and the j^{th} joint angles of the robot

c_{ij}	Cosine of the sum of the i^{th} and the j^{th} joint angles of the robot
C_{ijk}	Christoffel symbols
\dot{q}_m	Motor speed
l_i	Length of the i^{th} link of the manipulator
I	Armature current in the motor
L	Armature inductance
R	Armature resistance
U	Armature voltage of the motor
K_{bemf}	Back electromotive force of the DC motor
K_m	Motor torque constant matrix
\bar{U}	Maximum of the motor voltage
\bar{q}_m	Maximum of speed motor
\bar{I}	Maximum current in the motor
\bar{I}_c	Maximum armature current
\underline{T}_c	Minimum control torque
\underline{U}	Minimum motor voltage
$\underline{\dot{q}}$	Minimum joint velocity
J	Cost function in the objective function
v	Linear velocity of the end-effector
a	Acceleration vector
\mathcal{J}	Jerk vector (i.e., derivative of acceleration)
γ_0, γ_f	Path parameter at zero and final-times
r_0, r_f	Initial and final-positions
t_0, t_f	Initial and final-times
x	Unknown vector
e	Error parameter
Δe	Change of error
t	Time-increment
F	Force
ρ	Density
γ	Path parameter
ω	Total angular velocity vector

W Weight of neural network
 τ_i Torque applied to robot
 θ_i Joint angle of the manipulator
 $\dot{\theta}$ Angular velocity of revolute joint
 β, α, Φ Angle components of orientation
 ξ Damping ratio
 π Pi (constant = 3.14)
 n (n -DOF robot manipulator)
 i The number of links
 Δt Finite time-increment



LIST OF FIGURES

Figure 3.1: Proposed approach to position analysis of robot manipulator.....	34
Figure 3.2: Block diagram of the kinematics approach.....	36
Figure 3.3: The forward and inverse dynamic problems.....	37
Figure 3.4: Direct kinematics block diagram.....	38
Figure 3.5: Modified Denavit–Hartenberg (DH) parameter for a revolute joint.....	39
Figure 3.6: Denavit–Hartenberg (DH) frame assignment.....	40
Figure 3.7: Inverse kinematics.....	42
Figure 3.8: The SCARA manipulator.....	50
Figure 3.9: Coordinate frames attached to SCARA robot arm (RRP) type.....	50
Figure 3.10: Three-DOF (3-DOF) SCARA robot.....	51
Figure 4.1: DC motor system equivalent circuit.....	59
Figure 4.2: Trajectory planning along a parameterized path.....	63
Figure 5.1: Photo the IRCCyN SCARA robot.....	70
Figure 5.2: Desired path along the direction x and y -axis in a Cartesian-space.....	72
Figure 5.3: The corresponding optimal path parameter for moving end-effector....	73
Figure 5.4: The time derivatives of the path parameter with zero starting and ending.....	73
Figure 5.5: Time history of joint angles.....	74
Figure 5.6: The time derivative of the first joint with zero starting and ending.....	74
Figure 5.7: The time derivative of the second joint with zero starting and ending....	75
Figure 5.8: The profile for value acceleration, velocity, and jerk with zero starting and ending along the x direction in the Cartesian-space.....	75
Figure 5.9: The profile for value acceleration, velocity, and jerk with zero starting and ending along the y direction in the Cartesian-space.....	76
Figure 5.10: Computed torques profile of each joint angle with zero starting and ending.....	76
Figure 5.11: The time derivative of the first and second joints with zero starting and ending with computed torques.....	77
Figure 5.12: Actuator feeding voltages of the motor with zero starting and ending.....	77
Figure 6.1: Trajectory path parameterization.....	82

Figure 6.2: (a) The projection of the solution path (blue curve) in X–Y Cartesian-space. (b) The projection of the solution path (blue curve) in X–Y Cartesian-space..... 84

Figure 6.3: The projection of the end-effector's path in X–Y Cartesian-space.....84

Figure 6.4: (a) The end-effector trace when the manipulator is moving from the starting-point to the final-point. (b) Another point of view..... 85

Figure 6.5: Proposed methodology using neural network architecture..... 86



LIST OF TABLES

Table 3.1: Kinematics parameters.....	97
Table 3.2: Representing the SCARA robot parameter values.....	97
Table 3.3: The SCARA robot arm parameters.....	98
Table 4.1: Managing constraints for the optimization problem at time t_k	98
Table 5.1: Parameters of the IRCCyN SCARA robot.....	99
Table 5.2: Electro-mechanical constraints of the SCARA robot.....	99



CHAPTER 1

GENERAL INTRODUCTION

1.1 Research Background

In this chapter, we aim to define the most important terminologies pertaining to this thesis. Further details will be presented in the later chapters. Robotics is an interdisciplinary branch of engineering science that involves design, construction, and operation.

Nowadays, robotics has been acknowledged as an important field of industrial applications, and besides, it has important educational and medical applications. This field includes mechanical engineering, electrical engineering, electronics, computer science, artificial intelligence, and mechatronics. The multi-robot path planning problem can be avoided if the robots move from their starting-point to their end-point avoiding collisions with each other. The immense technological advancements during the past few decades have created the need for continuous development of new technological devices to cater the ever-increasing needs of human beings. This technological development was primarily observed in computer and electronic sciences: however; the technological development is no more confined to separate domains. In this scenario, the impact of technology is reflected in a vast array of fields, ranging from medical to educational to environmental fields of study. One such ever progressing field is robotics, which has the potential to facilitate the operations of a wide variety of domains. In the nutshell, a robotic manipulator is defined as a mechanical device, which is equipped with actuators and sensors, and they are controlled by a computing system in a workspace [1].

Robotic system manipulators have been extensively used in a wide array of robotics research because they perform human tasks more efficiently. Besides efficiency, robotic manipulators facilitate by reducing time and effort required to get a task done. This is the primary reason that robotic manipulators have utility in many significant applications such as medical sciences, space exploration, navigation and mining. This utility of robotic manipulators has created the need to optimize their performance in terms of speed. Hence, robotic path planning with optimum-time has become a

significant research area during the past few decades. The robotic manipulator workspaces are inhabited with certain obstacles because the workspaces are actually governed by certain laws of nature. When one or more robots are functioning in a particular workspace, their motions need to be controlled not only to avoid collisions but also for optimizing-time.

As mentioned earlier, robotic path planning is like devising an optimum strategy for the robots to initiate a move from their original-position to their final-position by avoiding obstacles in the way. The ultimate goal of path algorithm is to design a fully automated process for generating a series of movements to accomplish a task [2, 3]. It is a challenging task for robots to autonomously plan their paths considering the risks and constraints like collision, velocity limits, jerk and torque. This makes robotic path planning a challenging and interesting research area.

Another interrelated concept, which has been addressed in many important studies, is trajectory tracking. Although the terms path planning and trajectory tracking algorithms are sometimes interchangeably used, trajectory tracking is a slightly different concept. It is a sub-domain of path planning; so, only the constraints of velocity and acceleration generate a set of localized trajectories of a specified path [4].

In other words, trajectory generation means calculating a feasible way to let the robot move from an initial-point to the end-point. Generating a smooth and collision-free trajectory is imperative for efficient path planning. This complex challenge has been taken up by researchers during the past few decades, and they came up with valuable outcomes. The term "Robot" was first coined in 1921, when a Karel Capek, a Czech playwright in his play "Rossum's Universal Robots" mentioned robots as human-like machines. This play proved to be a pioneer and attracted widespread interest primarily in the genre of fiction, because the fictional artificial beings had a stark resemblance with human beings. The earliest works in robotics began during the World War II, when motorized manipulators were developed to control radioactive materials. Initially, these robotic manipulators required instructions for all their movements, but the first programmable robots were developed by George Devol in 1952, which were termed as programmed articulated transfer devices, and patented in 1966 [5]. This programmed device made way for robots. In 1962, Ernst developed a robot with force sensing mechanism, which was the first robot of its kind because it was capable of

functioning in an unstructured setting [6]. Then, the frequency of studies on robotics drastically increased.

Another significant progress was the development of the first robotic language called WAVE in 1977, which enabled robots to accept high-level instructions. This led to the introduction of first SCARA (Selective Compliant Articulated Robot for Assembly), which was patented in Japan in 1979, and then in the USA. In the initial research endeavors in the field of robotics, a robot was thought of as simply a constituting mechanical arm controlled by motor engines. Hence, path planning and motion planning problems were based on static surroundings without the underlying constraints [7]. During the early 1980s, there were many significant efforts to enhance the performance of industrial robotic manipulators.

Many of these studies focused on the understanding sensors as well as trajectory planning. For example, the first endeavor to understand the robotic path planning problem was to devise an obstacle avoidance strategy for a robot rover. On the other hand, there were several research endeavors in the early 1980s for understanding the dynamics of a multi-joint robot, using computation of torque. The robotic path planning problem attracted significant attention in that era when the researchers focused on finding a collision-free path for robotic manipulators [8]. Similarly, a significant obstacle avoidance strategy for a robotic path was devised in 1985. In the same year, the earliest robotic path planning strategy with time-optimization was conducted [9]. This study paved way for several significant advancements pertaining to time-optimized robotic path planning in the years to come. As the research in robotics progressed, its potential in different domains was identified. The first medical application of robotics was developed in 1987, and then in 1988, which comprised a robotic arm capable to perform a stereotactic brain surgery [10]. Since then, robotic applications have significant impact in the field of medicine to the extent that certain categories of robots have now been dedicated to the field of medicine. This includes micro-robots and bio-robots. Similarly, in the fields of research other than medicine, robotic applications are known to have a valuable impact. One such field is agriculture, in which, the utility of robotics was realized as early as 1985. Since then, robotics has largely automated agricultural operations and enhanced the efficiency of agricultural processes; however; other domains, which have seen significant advancements due to

robotic implementation, are construction [11], maintenance and repair, [12] and navigation [13] to name a few.

1.2 Basics of Path Planning

Path planning is a fundamental issue in robotics. This problem has been highlighted in many research works, for example, industrial engineering applications. Among all the types of path planning, optimization of the generated path may focus on different parameters such as time needed to move through a path, energy consumption during the process, geometric complexity of the path, and obstacle avoidance. The main idea behind this thesis is minimizing-time, which is needed for moving a robot from a starting-point to the end-point.

Path planning problem is, in general, important active research topic in robotics. There are many approaches to solve this problem. A robotic path is defined as the position of specified-points, on which, the robotic manipulator traverses, so that it can move from one-location to the other. For covering required distance in a workspace, different tasks must be decided. The best path planning solution must be found to control the robot's motion, for which, path planning is used. The algorithm of research is the path planning of the whole way from an initial to a final-point including stopping on a defined-location. In this context, finding a feasible path for moving a robot arm from a current-location to the destination is called path planning task. Although various studies have attempted to provide a feasible solution to the path planning for many years, it is still an open research problem. The main reason behind this is: It entails several complexities, particularly because of the existence of certain obstacles in the environment, which serve as obstructions in the path. The key issue in path planning involves devising a strategy to make a robot move from its current-position to the desired-position without collision with the obstacles.

Moreover, planning a robotic path can be more advanced if there are additional requirements as to how a robot traverses the path [14]. Considering these implications of robotic path planning, a more formal and parameterized definition of path planning problem for robotic manipulators was proposed in 1991 [15] in terms of formal representation called as configuration-space. According to this definition, the first part is to specify some "world" or space denoted by W . W can be defined in two ways, that

is, two-dimensional space, in which, $W = R^2$, and three-dimensional world/space, $W = R^3$.

The definition is as follows:

Given: $A, C_{free}, q_{init}, q_{goal}$; we'll find a path $\tau(s)$ which is valid, feasible, and provides a solution. The terms used in the definition are explained below:

- A : A single robotic manipulator is denoted by A .
- q : This is the position of the robotic manipulator, which consists of its linear dimensions as well as the geometric orientation or angles of the robotic joints. These angles or degrees-of-freedom (DOF) are constrained as well as linked. All the DOFs can be represented as vectors called as configuration.
- C : This is a group of entire q vectors within a certain range of values, which are termed as configuration-space, and denoted by C .
- C_{free} : At each position q , a robot A may or may not come across an obstacle. The configuration, in which, A does not come in contact with any obstacle, is denoted by C_{free} .
- q_{init}, q_{goal} : The starting position of the robotic path is represented by q_{init} , whereas the goal-position is denoted by q_{goal} .
- $\tau(s)$: The path that the robot A traverses in a configuration-space C is denoted by $\tau(s)$. The path $\tau(s)$ is a continuous function mapping, where $C \in [0, 1]$ mapped onto the configuration-space C .
- Validation: If the condition $\forall s. s \in [0, 1] \Rightarrow \tau(s) \in C_{free}$ is met, i.e. the path is not obstructed by any obstacle, such a path $\tau(s)$ is a valid path.
- Solution: If $\tau(0) = q_{goal}$, it means that the path $\tau(s)$ has reached the goal-position q_{goal} , so the path will be the solution.
- Feasible/Unfeasible: A valid path might not be a feasible path, despite successfully avoiding the obstacles, if it is not executable by the robot due to its constraints. This means when the path is beyond a robot's capabilities, it is not feasible, while if it is within its capabilities, it is termed as a feasible path.

As explained above, path planning is formally implemented in a space representation called as configuration-space, which is represented by C . A certain robotic manipulator, having k degrees-of-freedom (DOF), implies that each robotic

configuration can be denoted through the configuration-space C in terms of a set of real values $\{q_1, \dots, q_k\}$. The k values refer to the points in the configuration-space C . When the problem of finding a path from its first-point to the final-position evading the obstacles is visualized in the physical space, it seems a really complex one; however, it becomes easier when the configuration-space is considered because everything in a configuration-space is defined and formalized. Considering a configuration-space C , if the obstacle of the path O is defined as a subspace of C , it represents a position where the robotic manipulator collides with something. That collision-free space can be calculated as $C_{free} = C - O$. Hence, C_{free} is an obstacle-free path, on which, a robotic manipulator can freely move [14].

1.3 Trajectory Planning Algorithm

The trajectory planning for a dynamic system is a basic issue in industrial robotics research because it focuses on increasing productivity. Trajectory planning algorithm is based on the movement of a robot arm from one-position to another in a controlled manner. The position of a robot can be determined by knowing to what extent variables can be moved to achieve the desired-position or target. A path is defined as a sequence of robot motion without consideration of time.

In trajectory planning, path movement must be attained with specific timings keeping in view velocities and accelerations. In the scientific robotics, the problem of path planning entails optimization of certain parameters such as time, distance, and force etc., whereby, a path has to meet that optimization criterion. This subset of path planning, which involves optimization of certain parameters, is termed as trajectory planning or tracking [14].

Path planning algorithm and trajectory tracking problem are sometimes interchangeably used, which is wrong. A path simply defines the start-to-end-points whereas the trajectory defines the points as well as time, at which, the final-position is attained. A trajectory can be thought of as a path that has time constraints attached to it. Trajectory tracking algorithm formalizes motion planning keeping in consideration the temporal and mechanical limitations of a robotic manipulator [16]. Trajectory tracking is formally defined as the real-time transition of the robotic manipulator from the initial-point to the next within the robot's kinematic limits while avoiding obstacles [17]; therefore, a trajectory can be single-dimensional. On the other hand, if a

trajectory is multi-dimensional, it means that the robotic manipulator has multiple degrees-of-freedom, and the same is the case with most of the robotic manipulators. A single-dimensional trajectory is represented by the scalar function $q = q(t)$ whereas; a multi-dimensional trajectory is denoted in terms of a vector function of time: $P = P(t)$. There are several criteria, according to which, a trajectory might be generated. Most significant of these criteria include the time-optimization criterion, energy optimization criterion, and the minimum jerk criterion [18].

Moreover, there are several kinematic and dynamic constraints commonly accounted for during the trajectory planning process. These constraints can be of two types: the ones, which are implicit to the robotic manipulator, and the ones, which are also implicit to the constraints of the current task. The common constraints are acceleration, friction, torque/force, jerk and joint limits, which are primarily the underlying characteristics of a robot [19]. These constraints can be incorporated into the trajectory generation as follows:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + F(\dot{q}) + G(q) = \tau \quad (1.1)$$

Here, q denotes the position, τ represents the torque, the mass is expressed in terms of $M(q)$, the centrifugal force vector is denoted by $C(q, \dot{q})\dot{q}$, the friction vector is represented by $F(\dot{q})$, $G(q)$ that denote the gravity-torque vector, whereas, the derivatives of the position vector are denoted in terms of \dot{q} and \ddot{q} , respectively.

1.4 Way Point Tracking Limitations

A robotic path can be generated based on a set of way points. A way point represents a certain position in a trajectory at a particular time; hence, it denotes the most atomic level of a trajectory. Within a certain trajectory, a way point can also contain information about the other connected way points such as the position and velocity of the previous and subsequent way points. Furthermore, a trajectory may consist of a vector of n way points, on which, a robotic manipulator traverses so as to reach the final-position.

In addition, the way points in a 2-D configuration-space are represented as (x_k, y_k) , where $k \in \{1, \dots, n\}$. Each way point can also include the robotic constraints such as time, velocity and acceleration. Simply put, way points represent a framework that specifies whether a position is within a set of desired-positions or not. Way points of

a path can be generated using straight-lines and circular-arcs. Other than that, if a smoother path is desired, different interpolation techniques are applied; however, interpolation may make the process more complex. It is desirable in most path planning problems to maintain the simplicity of the straight-line path while adding smoothness and flexibility to the interpolation techniques [20]. A feasible solution is needed to maintain simplicity while adding smoothness that derives a spline curve to generate lower order polynomials, which are joined with each other through control points. If a high number of way points are required, a higher order spline has to be chosen.

Bobrow (1988) undertook further advancements in this domain by devising a mechanism to discover a collision-free path to acquire the minimum-time path planning solution to a robotic manipulator. In this study, he proposed the formularization of B-spline polynomials as well as the equations of motion in a nonlinear form in order to produce near optimal trajectories for a robot in the Cartesian-space [9].

1.5 Joint Control System

In robotic manipulators, the robotic joints play a significant role for trajectory tracking problems. The movement, the position and the number of robotic joints determine their degree-of-freedom (DOF). Robotic manipulators are usually programmed to accomplish a particular task in a task space. In spite of this, the tasks of the robotic manipulator might be mapped onto the joint space, particularly in case of a robotic manipulator, which is controlled within a joint space. This mapping typically takes places with the help of inverse kinematic techniques. The robotic manipulator may be further mapped onto a torque space based on the information of the joint space.

The trajectory tracking algorithm is implemented by the robotic manipulator in a joint space. The robotic manipulators in the joint space can be represented through two control design categories, which include classic joint control and model-based control. The classic joint-control caters to the generic implications of trajectory tracking while the model-based control implies the robotic control in the joint space specific to a certain trajectory tracking model [21].

1.6 Problem Description and Method

In the last few decades, researchers have been working on robotic path planning problems [8]; however, optimization of time and velocity for the path tracking of robotic manipulators is something that is still a challenge in the domain of robotic studies, and it has gained significant attention during the last few years. The approach to time-optimized robotic path planning is very significant particularly for industrial usage.

Several techniques have been presented in the previous studies in this context; however, finding a minimum-time path planning solution still attracts significant attention of the researchers because of its complexities related to nonlinearity, force and torque limitations, and coupling dynamics, to name a few. Hence, in this research, a strategy is required that takes into account the constraints of torque, force and friction while tracking a time-optimal and velocity-optimal trajectory for robotic manipulators in the Cartesian-space. In the nutshell, a mechanism for path planning needs to be devised for robotic manipulators to traverse a pre-defined trajectory within a specified amount of time and velocity.

1.7 Problem Statement

In the recent years, robotic technology became one of the high importance scientific technologies; so, it is used in many research areas and experts believe that it is important to the future of mankind.

In this research, the target is how to resolve the minimum-time path planning issue for a robot to move end-effector from one-position to another. The path planning algorithm operation needs to obtain the path parameter optimization algorithm with external force and the dynamic models of the robot system, which derive both forward and inverse kinematics equations; so, it is an important step in robot modelling. The analytical solution to the robot manipulator has been focused in this thesis to obtain a path using forward and inverse kinematic methods. Using these methods, manipulator's joint angles are determined from the required target given in the Cartesian-space and finding the trajectory planning solution for the path using the path as a parametric and minimum-time method for a robot manipulator.

1.8 Objectives Methodologies of the Thesis

The principle research focuses to find the path planning problem for a robot arm. To achieve the aim of the project, its major objectives are formulated in the following way:

- The aim of the thesis derivation of kinematic model is finding both direct and inverse kinematics as a necessary step in robot modelling. In particular, a detailed presentation of the Lagrange formalism, which has been used to determine the model of a manipulator. All the motion approaches considered in this work will be validated through the simulation results obtained in the case study of the robot manipulator by simulation, which was carried out with MATLAB program using a mathematical model of a SCARA robot.
- This method helps solving the path planning issue with the help of new approaches to industrial robotic systems taking into consideration their limitations. The proposed algorithm develops approaches to path planning through mathematical models for the robot manipulator systems, its joints, driving motors, and their simulations. Moreover, the path planning is based on modelling and analysis of a robot, which is the goal of this study. Many cases have been tested in this research using a manipulator system with minimum-time path planning method to motion planning of a robot arm from an initial-point to the end-point. We developed and tested a mathematical model to evaluate the appropriate solutions to robot path from a starting-point to the target-point.
- In the current work, we made a strategy for the minimum-time, and given initial and end-points. Moreover, this study investigates a minimum-time path planning mechanism of parametric trajectory of robotic manipulators under the external force constraint. The research is conducted to develop a robot path planning algorithm to move robot from a defined point-towards the target-point by considering minimization of travelling time for industrial robots. One of the goals of the current project is to minimize the cost function that is subject to constraints such as angular accelerations, angular velocities, angular jerks, input torques, input voltage, torques of actuators and final-time

limitations/constraints; therefore, the key purpose of this investigation is to optimize the time taken for a manipulator to traverse a pre-defined path.

1.8.1 Obtaining the Dynamic Model

By using mathematical modelling equations, which are based on the concepts of generalized coordinates, energy and generalized force are obtained.

1.8.2 Designing the Model

MATLAB is used to simulate the robot manipulator as a case study in this research.

1.8.3 Simulation and Comparison of Results

The results will be obtained through the simulation program.

1.9 Thesis Overview

This thesis is laid out as follows: By now the reader must have realized that Chapter 1 introduces the issues pertaining to robotics path planning problems algorithm, definitions, trajectory tracking, planning algorithms, and the way point tracking technique. Chapter 2 reviews the literature survey of the robotic kinematics, general path planning problems, robot control system, minimum-time trajectory planning method, trajectory planning and tracking, and actuator of dynamic constraints. Chapter 3 describes mathematical modelling and kinematic analysis of the robot arm. The mathematical model consists of expressions, which discuss the kinematic analysis: the Denavit–Hartenberg (DH) parameters, forward and inverse kinematics, and the modelling of the robot arm using the Newton–Euler method and the derived Lagrange method. The model constitutes kinematics and inverse dynamic equations. Chapter 4 focuses on robot path planning technique. This research presents tests of the proposed minimum-time path planning algorithms using path parameter optimization algorithm with the help of external force and frictions for developing the permanent magnet DC motor system, which consists of an armature circuit driving a mechanical load. Chapter 5 presents the simulation study and its results. Finally, Chapter 6 consists of some concluding remarks, recommendations and perspectives, which may be useful for future studies.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In the field of robotics, there is a need for tracking an already known path for robot manipulators in the Cartesian-space with maximum velocity or minimum-time because it has economic benefits. Minimum-time path planning problem for robots have been widely studied in the past especially for industrial applications.

Many techniques have been proposed in the past to deal with this problem; however, due to factors such as non-linearity, coupling dynamics, complex systems, and torque limitation of an actuator, the task of finding a minimum-time path planning method for a manipulator has been quite complex; therefore, along with the advancements in the field of robotics as early as 1950s, many significant studies after that focused on the problem of path planning.

With the passage of time, more research has been focused on path planning approach during the last 30 years. This section seeks to develop a strong foundation, on which, the central theme of this research is based. It illustrates the significance of robotic research in general as well as throws light on the specific techniques and approaches applied to the minimum-time path planning method. In this context, a survey of the valuable studies related to robotic path planning has been carried out. The excerpts mentioned in this review have been collected from different research resources. It is mainly concerned with robotic path planning, its several approaches, and dynamics of constraint.

The review covers five main aspects of robotic path planning mechanism: Path planning problem with external force, trajectory planning and tracking algorithm, minimum-time path planning method, actuator of dynamic constraints approach and the literature of these aspects, which has been reviewed below.

2.2 Path Planning Problem

Path planning issue in robotic manipulators has been addressed since decades, starting from the earliest path planning study methods in this domain [22]. The methods suggested in the earlier studies proved to be detrimental to the advancements in the field of robotics particularly robotic path planning. Among the initial studies, path planning research attracted the most attention. It was called as generation of collision-free paths [23]; therefore, in the collision-free path generation, two main approaches were presented including global-approach and local-approach. The global-approach requires the development of an accurate depiction of connectivity to make a series of collision-free arrangement for a robotic manipulator in the form of connectivity graph.

On the other hand, the local-approach involves searching from a graph, which is already in place in the robotic manipulator's configuration-space. In another interesting study on path planning problem domain involved computation of heuristics to figure out the information about the geometrical structure of the robotic configuration-space. It was aimed at enabling the robotic manipulator to construct the grid by itself. One of the popular heuristic techniques for this scenario involves the searching of the path, which is guided towards the flow of cancelled gradient vector field, and it is produced by an artificial potential field [24].

Researchers have analyzed that although the global-approach promises to be efficient, it is computationally expensive. This is due to the fact that the global-approach requires the pre-calculation step, which is configuration of a connectivity graph before the path planning actually begins. This pre-computation makes the global-approach exponentially complex and expensive. Conversely, the local-approach excludes this pre-computation step because it only requires research in an already defined search space. Hence, the local-approach proves to be faster and less complex than the global-approach; however, it is unable to consider the additional constraints that might be present in the search space making it more simplistic than required. To address this issue, the researchers in the field of robotics have proposed powerful heuristic methods for guiding this path search. These studies paved way for further advancements in the coming years in the field of robotics. Researchers analyzed the application of B-spline polynomials for generating near-optimal trajectories in terms of minimization of the spline curve [9, 25], and [26]. In addition, other researchers have studied the usage of

spline curve, which was devised to connect straight-lines through circular-arcs by Fourier series and cubic Bezier splines. Another interesting approach proposed as a solution to the robotic path planning problem formulates a Genetic Algorithm (GA) of a continuous nature in order to develop a path planning in a Cartesian-space.

2.3 Trajectory Planning and Tracking

The path planning algorithm for the robotic manipulators, particularly in case of autonomous systems, consists of two primary phases termed as the trajectory planning and the trajectory tracking algorithm. Several strategies of robotic trajectory planning and tracking have been proposed in the literature.

Trajectory planning and tracking, in addition to obstacle avoidance, add dimensions to the path planning problem such as time-optimization, velocity-optimization, computational efficiency and the overall feasibility of the path planning mechanism. Significant studies, which added value to the field of robotic trajectory planning include a mechanism for letting the robot manipulator learn from the path it previously traversed [27]. This study specifies particular way points, using which, the robotic manipulator can get assistance to track a path.

In a related study, a mechanism was employed to identify a series of way points in order to plan the path of robotic manipulators. In this approach, the velocities of the previous path are approximated to generate a trajectory. Moreover, a number of continuous trajectories are taken into account from the start to end where the specified trajectories are blended together. The mentioned procedure is termed as point-to-point trajectory motion by several significant researchers. The researchers also examined the point-to-point approach to trajectory planning while the trajectories were generated for the point-to-point path in a joint space.

Several recent researchers have incorporated the type of path to be followed by a robotic manipulator in the path planning process; however, these types of path planning include straight-line path, cubic-spline and circular-path [28]. The researchers also considered the path types within the path planning process. The point-to-point planning incorporates certain constraints such as position, velocity, acceleration and jerk in the form of polynomial coefficients besides recognizing the path type [29].

Many past studies have emphasized the use of iterative and geometric methods as well as an optimized switching mechanism for trajectory planning in conformity with the Pontryagin's Maximum Principle (PMP). Furthermore, other relevant studies have employed the B-spline cubical mechanism in case of trajectory planning [30].

Another significant study uses cubic-splines for global trajectory planning. This study generates a trajectory for a robotic manipulator in a joint space incorporating the jerk criterion, which minimized the jerk. Although the study does not take into consideration the manipulator's dynamics, the minimum jerk approach seems very effective for the path. The approach is applied to an arbitrary six joint robotic manipulator. This trajectory tracking mechanism does not only promise to improve the performance, but also vows to increase the robotic manipulator's life span [31].

Trajectory tracking algorithm has attracted significant attention of researchers, particularly for formulating a trajectory tracking control mechanism in case of the flexible joint robotic manipulators. A few of these studies have presented an approach to trajectory tracking control without taking into account the actuator dynamics [31].

2.4 Path Planning with Actuator Dynamics

Most of the recent researchers have found it necessary to include actuator dynamics and constraints as a part of trajectory tracking control. For example, a trajectory tracking mechanism was proposed considering geometrical constraints, impulsive force constraints, torque constraints, maximum acceleration and velocity constraints.

Another interesting study in this regard [32] compares different frameworks of trajectory tracking controllers for unmanned vehicles, which consider nonlinearity of the dynamics, uncertainties, noise, disturbances and several other constraints. The actuating machines usually used for robotic manipulators are DC motors. Hence, the actuator dynamics mostly consist of the force, energy and friction, which are required for the DC motors attached to the manipulators.

2.5 Minimum-Time Trajectory Planning Method

The most notable earliest studies in the domain of minimum-time problem along a path were conducted by several notable researchers.

In another study of the same era, the authors presented an approach to compute paths in cases of closed-kinematic chain mechanisms. The earliest in this domain also included solutions to the problem of minimum-time trajectory planning in the joint space along a specified geometric path, considering the robot dynamics and constraints, which are subject to force and torque [33]. This was the first attempt of its kind to address the minimum-time problem solving constraints.

Bobrow took the research further and devised a technique to find a collision-free path to obtain a minimum-time motion planning [9]. He used non-linear equations of motion to produce optimal trajectories in the Cartesian-space. His use of B-spline polynomials produced optimal trajectories, which were further investigated by researchers [26]. Furthermore, minimization of the spline curve path was also studied in [25, 26, 34]. More researches who contributed to further advancements in the research added techniques for making the robot manipulator learn from the previous path devised [27] while a number of waypoints are determined with the help of a robot when it moves on the path. This procedure of specifying a set of way points for path planning is further elaborated by the researchers [35]. They estimated previous path velocities for trajectory generation and considered several continuous trajectories between points on a path. They were blended, which is also called as point-to-point motion. The method for generating a smooth and time-optimal trajectory has been presented, and the authors used the third derivative of the path parameter with respect to time as an input, which it limits the torque rate in order to achieve the smoothness of the path.

The point-to-point motion was investigated by the researchers [36]. They generated trajectories in a joint space for the point-to-point motion. In this study, the authors also considered the constraints of the actuators' velocity, acceleration and jerk limits while finding the minimum-time for the path planning. In some studies, the type of path leads through the given way points, such as a straight-line path, cubic-spline or a circular-path, which are included in the path planning.

In addition to taking the path type into consideration, the point-to-point motion planning also includes the polynomial coefficients of the constraints such as the position, velocity, acceleration and jerk constraints [37]. Moreover, a recent study [38] focused on user-defined trajectories as well as the development of commercial robotic

software. Many previous studies have also focused on iterative and geometric methods as well as optimal switching structures in accordance with Pontryagin's Maximum Principle. Some researchers have applied B-spline cubical methods for trajectory planning [30, 42]. The optimization of the minimum-time path problem has also been analyzed using the Sequential Quadratic Programming (SQP) method. For optimization, the above-mentioned method involves the determination of minimum transmission-time with electromagnetic constraints such as kinematics. Building up on that, researchers [27] used a similar dynamic programming algorithm in order to solve the minimum-time path planning problem. The dynamic programming approach has also been applied in a previous research [3]. After the early advancements in this domain, ample amount of research has been diverted towards the constrained motion of robots. For example, researchers [40, 48] proposed a method to solve the minimum-time problem considering the constraints of a jerk. Another study presented an algorithm for a minimum-time calculation, which was subjected to kinematic constraints [39, 41].

Another proposed method to deal with the path planning problem involves the connection of straight-lines with circular arcs, perturbations of a straight-line with Fourier series and cubic Bezier splines. Furthermore, solutions to the path planning with end-effector constraints have also been studied. In addition, another study on path with torque constraints includes either the bang-bang trajectory or the bang-singular-bang trajectory. Other interesting algorithms for solving the path planning include a continuous Genetic Algorithm for path generation in a Cartesian-space. More recently, the robotic methodologies of a point-to-point trajectory have also been applied in some applications [43].

2.6 Path Planning with External Force and Friction

There are many other approaches to using trajectory, which include agricultural field machines that utilize trajectory generation for animal movement. Most of the significant studies on robotic manipulators include those, which estimate the minimum path and generate trajectories while handling any kinematic constraints on the velocity and acceleration [44]. This study proposes a path planning mechanism while trajectories are generated in the operational space, which is subject to certain dynamic constraints. The studies, which have been mentioned here, do not usually consider the

factors such as external force and friction while calculating the minimum-time for the path planning of robot system manipulators.

There are some significant researches, which focus on external forces as well as friction as factors influencing the calculation of the minimum path for a robotic manipulator, which is controlled by DC motors [45] keeping in view the kinematic constraints of velocity and acceleration. Among the several aspects of path planning, the focus was on time-optimization of path planning. Traversing a given path in a limited amount of time has been a challenge for robotic manipulators and their developers. Path planning problem with time-optimization was first mentioned in a study [9], which paved way for several more studies in this domain including the minimum-time path planning problem along a specified path by some other researchers [33]. They took this research further and devised a time-optimized method of robotic path planning in a joint space. This study proposes a dynamic time-scaling algorithm as well as the graph search technique.

Another important study among the earlier studies formalized an algorithm for minimum-time trajectory tracking with different actuator limits. The study proved to be extremely useful as it presented a faster approach to trajectory tracking keeping in view the robotic control mechanism constraints and search approaches to this task. The study was among the pioneers, which proposed a parameterized path in the configuration-space as an effective approach to finding a minimum-time trajectory. The research considered the actuator's torque constraints and searched the potential paths to finally propose a time-optimal path. The smoothing and parameterization of the path were done using splines [46].

Another valuable study of this era proposed a method of calculating a robotic path in case of closed-kinematic chain mechanisms [47]. Similarly, another related study proposed an efficient technique for time-optimized robotic manipulator's path planning problem within a specific geometric path while taking into account the significant constraints of force [33]. Moreover, the problem of time-optimized path planning was addressed in another study [48]. In a significant study, a smooth and minimum-time trajectory generation mechanism has been proposed [49], which formalized a method for smoothing out the path by employing the third derivative of

the parameter. The derivative has been derived considering the torque rate and time as inputs.

The time-optimization of path planning and trajectory add productivity to the path planning mechanism; however, it was computationally expensive. Consequently, several studies have proposed a computationally cheaper methodology for a minimum-time path. Then, some researchers proposed a minimum-time path problem mechanism for an industrial robotic path, specifically for the robotic task of picking fruits [50]. In this study, a minimum-time for industrial robots was proposed while avoiding obstacles, which is computationally way less expensive as compared to what was previously proposed because the dynamic manipulators had disjointed or decoupled links without velocity constraints. In terms of computationally cheap time-optimal path planning, a time-optimal approach was presented in another study for a superior performance trajectory generation in case of omni-directional unmanned vehicles. The study uses a bang-bang control approach, which assures a balanced trade-off between time-optimality and computational efficiency because it comprises of a minimum number of computations by limiting the range of possible solutions [51]. Several valuable studies have devised a minimum-time path planning while considering the manipulator's dynamics. They proposed a less computationally expensive methodology for time-optimal motion planning. The study transforms the time-optimal motion problem into a convex optimization. For this, a disjointed approach to path planning has been presented as opposed to a direct approach [33]. This disjointed approach first solves the path planning problem, which must take into account the task characteristics and constraints as well as obstacle avoidance. The second phase of this approach carries out the time-optimal trajectory by taking into account the manipulator and actuator limitations [52].

2.7 Robot Dynamic Constraint Algorithm

Constraints of a robotic manipulator include natural constraints, which exist because of its mechanical and geometric specifications. In several significant studies, the researchers considered the constraints of the manipulators such as a jerk, acceleration and velocity while determining the time-optimum path for robotic manipulators. Another approach to path planning was implemented taking into account the constraints of the end-effectors [53].

In a study, path planning problem was researched keeping in view torque constraints and used the bang-bang path planning. Moreover, this significant study discusses point-to-point time optimization in path planning, which has been proposed using the Sequential Quadratic Programming method [54]. This study also incorporates the minimum transmission time mechanism keeping in view electromagnetic constraints and kinematics.

Moreover, researchers proposed a dynamic programming method for time-optimal path planning, which also discussed the early advancements in the domain of time-optimal robotic movements [27]. In this context, researchers proposed an approach to discover a minimum-time path planning keeping in view certain constraints, especially the jerk constraints in terms of higher order derivatives of the robotic position [3]. A relevant research [48] employs a method for determining a minimum-time path planning solution taking into account the dynamic and kinematic constraints. In more recent significant studies, the mechanisms have been proposed for point-to-point trajectory problem in case of various other applications; for example, researchers utilized the trajectory planning approach in the domain of agricultural machines and trajectory generation for animal movement [55]. The studies, which have been mentioned until now, do not usually consider the factors of external force and friction while calculating the minimum path for robotic manipulators; however, there are some significant researches, which focus on external force as well as friction as factors influencing the calculation of the minimum path for a robotic manipulator. For example, studies on robotic manipulators, which were controlled by DC motors, considered the force and friction generated by the motors, and they were dependent on the kinematic constraints of velocity and acceleration [45]. The position and force constraints of a robotic manipulator's trajectory in a coordinate system were also addressed.

In a related study, the authors proposed that the force constraints of robotic manipulators, when they interact with the environment, should add up to zero, in order to assure stability. This method employs a real-world interaction of forces within the configuration-space C . It also addresses the force constraints of robotic manipulators. In this study, the authors used a Dynamic Movement Primitive (DMP) to minimize the force in the process of interaction of a robot with its environment [56]. The study also proposes the approach to obtain the force constraints from kinaesthetic illustrations of

the robotic manipulator. The methodology employed in this study gets the robotic manipulator to tackle not only the force constraints in the path but the position and velocity constraints as well.

2.8 Position Control for Robot Manipulators

The research on the robotic arm control started as a part of space research, which caters to the requirements of a space robot manipulator. Some primary aspects of the robotic arm control for the applications of space studies were pointed out. After that, the robotic arm control was used in nuclear applications primarily to carry out decontamination operations, after which, the maintenance of the nuclear power plant was done through the robotic arm positioning. This proved to be a significant application, which required positioning precision of a robotic manipulator [57].

Flexible robotic arm manipulators have been proposed since they were first used for surgical operations. Later, such robotic arm manipulators were developed for boosting the precision in micro-surgery. A similar application was developed for robotic positioning precision for the treatment of cancer patients in Massachusetts General Hospital. In addition to that, the researchers studied biped walking machines through the physical dynamics of the robotic arm manipulator [58]. The aspect of a robotic arm manipulator that has been studied extensively is the control.

Robotic control refers to making a robotic manipulator carry out a task, keeping in view how familiar a robot is to the physical space, modelling and controlling factors. There are five major types of a robotic arm controls: Vibration control, position control, motion control, force control and the joint tracking control. Joint trajectory tracking control involves the robotic arms' joint control that is required for the motion along a specified trajectory. Moreover, vibration control of a robotic arm manipulator is important, as vibration causes a decline in the robot's performance in terms of its position [59]. Motion control in the robotic arm manipulator mainly makes sure that the robotic path is flexible as well as smooth. The force control in a robotic arm manipulator implies that the force, with which a manipulator interacts with the environment, can be controlled, and also with the help of the knowledge of the position and environment of a robotic manipulator [60]. Flexible robotic arm manipulators have been modeled in several significant researchers.

Researchers have focused on flexible robotic manipulators as opposed to rigid ones, so that they mentioned the benefits of portability, less complexity and less resource consumption. For example, the researchers analyzed the flexible robotic arm in case of collision-avoidance, which gives rise to other robotic applications, such as drawing robots and pattern recognition robots [61].

Several other utilities of flexible robotic arm manipulators have been studied by different researchers; however, the focus of the problem remained the controlling of arm vibrations. This problem has been taken up by some researchers, who mostly employed dynamic models and other control techniques as a solution to this problem. As is the case with other robotic manipulators, the robotic arm manipulator also encounters several constraints. One of the errors of control arises when the torque requirement of the motor is not met; hence flexibility is not taken care of. Secondly, the challenge is to maintain the precision and accuracy of the position of end-effector. To solve these problems, a robotic arm is required to have minimum vibration.

Robotic arm manipulators have been modeled in different ways in different studies. The mathematical models primarily work based on the energy principles. The robotic arm's kinetic energy is stored in terms of their inertia, and the potential energy is stored as a function of its position and gravitational field. If a robotic arm manipulator is flexible, it is able to store its potential energy by means of its links or joints. The joints or links may be exposed to twisting, turning or compression. Twisting joints store more potential energy whereas compression stores less potential energy because stiffness emerges as a consequence of compression. The bending of the joints stores potential energy by means of deflection. The linear models are only able to capture the single-link robotic arm manipulators whereas; they are unable to capture the multi-link manipulators [62]. Due to the non-linearity of multi-link robotic models, it becomes more complicated to capture the multi-link model. A Lagrangian-based model was proposed [63] in order to model multi-link flexible robotic arms. For modelling an n -link robotic arm model, the method for Assumed Mode Model (AMM) was analyzed applying a special moving coordinate system, which is termed as virtual rigid-link coordinate, and it helped solving this problem [64].

Another significant approach to solve this problem was proposed when the multi-link robotic arm manipulator was modeled with a linear dynamic model. This approach

involves a random number of flexible robotic links. The flexible links of a robotic arm model are treated using the Euler–Bernoulli (EB) beams. The researchers formularized the dynamic equations for the n -link robotic arm manipulators with the help the Newton–Euler (NE) method. Typically, flexible robotic manipulators are represented with mathematical models to analyze flexible models of robotic arm manipulators with the help of Euler–Bernoulli beams [65].

The dynamics of the flexible manipulator have been represented using partial equations. The differential equations are interpolated into finite models mainly using three models including Assumed Modes Model, finite element method, and lumped parameter. Moreover, the assumed mode approach transforms the n -dimensional system model into a finite dimensional model with a compulsion that the finite mode amplitude as well as the positional eigenfunctions must be taken into account. For designing the Assumed Mode Model, the eigenfunctions as well as the boundary condition of the robotic arm's link should be chosen [66].

In this study, the researchers explained that by using the Assumed Mode Model, the dynamic robotic model has been illustrated highlighting the vibration modes. Hence, the robotic link is denoted by a set of linear models of eigenfunctions. Moreover, it has been stipulated that the boundary conditions for the Assumed Mode Model can be selected in a number of ways; however, a limited set of Assumed Mode Model boundary conditions can prove to be significant for the robotic manipulators for some specified applications. This set of boundary conditions need optimization for the Assumed Mode Model in a way that is closest to the natural modes. The natural modes are determined by a number of factors of the robotic environment, particularly inertia and size of payload mass. The study concludes that it is extremely important to choose the suitable number of boundary conditions for the Assumed Mode Model to yield better results. This choice of the number of boundary conditions must be assessed based on the structure of robotic manipulator and the natural modes. The Assumed Model primarily decomposes the infinite link into a set of eigenfunctions, which are also known as mode shapes or time-specific coordinates. In studies, several ways of choosing the boundary condition for the Assumed Mode Models have been specified. Mostly, the selection of boundary conditions is based on the closeness to the natural modes of the robotic environment [67].

Many researchers have used Assumed Mode Model for designing the flexible link robotic manipulators in order to analyze the relationship between the vibration modes and the non-linearity of the manipulators. A significant method of boundary condition modelling is to use Euler–Bernoulli's beam, which is primarily used for assuring the model flexibility as well as its natural mode, which is computed according to the load of the manipulator's link. This approach to the Assumed Modes Method ensures that the estimated deflection of the link depends on the diagonal vibrations of the elastic manipulator's beam, which can be represented as a set of functions of the shape of the mode and the time-based displacement.

In this study, a significant amount of nonlinearity was added to the two-link flexible manipulator considering the inertial constraints, using an equation for developing an Assumed Modes Model also known as Lagrange's equation. In this study, a control mechanism has been applied comprising four actuators. Two of them are the DC motors coupled with the structures of the flexible links of the robotic arm manipulator, which help obtaining the required end-point motion.

The researchers have also proposed a mechanism of modelling multiple flexible links and joints in a dynamic manner, making use of the Euler–Lagrange model in addition to the Assumed Modes Model. In this study, the unitary flexible-links are influenced by the disturbances of the robotic arm manipulator's environment, so that the designed robotic controller belongs to the lower order [68]. The flexible robotic arm has been proposed by using the Assumed Mode Model. A flexible-link model has been used to explicitly develop a dynamic equation for carrying out the pre-specified motion but in this case, the researchers considered possibility of a robotic arm joint getting locked. They also examined the consequences of the stiffness or rigidness in the wrist force sensor of the flexible robot arm.

Furthermore, the Assumed Mode Model and force controlled robotic arm manipulator has been investigated. The study proposed a closed-force model in terms of the two-link (2-DOF) flexible robotic arm manipulator making use of the Assumed Mode Model. The study derives the two modes of modelling the two-link flexible robotic arm manipulator using Assumed Mode Model with the help of mathematical equations for a predefined motion, and formalization of hybrid force law. The study converts the

nonlinear closed-loop mathematical equations into the linear ones. The study also calculates the eigenvalues of the linear set of mathematical equations [69].

Many years later, another valuable study in this considered considered the eigenvalues of linear equations considering time variations of the boundary conditions. The experimental justification of modelling a flexible 3 meters long robotic arm manipulator was found with the help of Assumed Mode Model keeping in view several constraints and limitations. It proposed dynamic modelling of flexible robotics by using the Assumed Mode Model keeping in view the payload factor [70]. This study was mathematically as well as experimentally justified and verified. It was followed by a lightweight closed-loop model for highly flexible arm manipulator specifically designed for space applications. This approach has been derived from the Newton's–Euler model, and it was simulated and verified using MATLAB/Simulink as well as the MSC/Adams software [70].

The two-link flexible robotic arm can be developed by deriving combination of Euler–Lagrange model and Assumed Mode Model but geometrical, inertial and payload constraints must be kept in mind. As an alternative approach, nonlinear autoregressive moving average can also be used coupled with the algorithm of extended recursive least squares.

Researchers faced many problems during modelling and controlling a flexible arm manipulator. It was found that the stiffening of the robotic arm took place because of centrifugal force in more cases as compared to geometrical stiffening. Further investigations were carried out applying the Assumed Mode Model with a pin-free link, open-loop feedback-control, and open-loop flexible robotic control. Later, the Assumed Mode Model was used to understand elastic deformation of the robotic arm. It was tested with the help of Euler–Bernoulli beams; so, the possibilities of robotic link inertia or the twisting of the flexible robotic arm link were ignored. The equations for the motion of the robotic manipulator were formalized using the Lagrange's model.

It was believed earlier that the Finite Element Method (FEM) characterizes the factor of elasticity in case of flexible robotic arm manipulators. It can be explained as: In finite element, the elasticity factor is accounted for in rigid bodies while the robotic arm link is then overlaid on it. A major benefit of the finite element method is that it tackles the nonlinearity with simplicity and less complexity. The finite element method

tackles the design and boundary condition uncertainties and disturbances in the robotic environment. Although finite element model provides a lot of advantages as compared to other models, however, it fails to consider the natural frequency.

In addition to that, the finite element method might result in complex over-projected stiffness in the flexible robotic arm manipulators. The mathematical simulation of the finite element method can be exhaustive because it has many wide-range state space equations. A flexible robotic arm manipulator employs finite element method; however, it considers the links of the flexible robotic arm as a sequence of n elements which have the same size. In most cases, the finite element method formulizes a flexible robotic arm manipulator as a nonlinear dynamic model by a combination of Lagrange approach as well as the Euler–Bernoulli beam approach, taking into account more than one node. The more the number of nodes is, the more will be the dimensions of the stiffness matrix. This clearly means that the increased nodes increase the length and complexity of the equations. This implies that for an optimal performance of the flexible element method, the number of nodes must be wisely selected [71].

In the same study, a mechanism has been proposed to dynamically model the complex spatial aspects, particularly for the applications of flexible robotic arm manipulators. In this study, the analysis has been carried out for employing the finite element method in combination with other dynamic analysis methods as well as the coordinate reduction mechanisms for nonlinear and complex system analysis. A set of dynamic equations was applied, which mathematically represent the multi-link flexible robotic arm manipulators. By formalizing an equation for the finite element method, the coordinates of the specified nodes were represented as positional constraints [71].

In another relevant research, the dynamic and kinematic constraints of the flexible-link robotic manipulator have been analyzed in case of a generalized three-dimensional motion. This study takes into account the mass as well as flexibility of the flexible robotic arm manipulator without making it discrete. Furthermore, the model of two-link flexible robotic arm manipulator was presented, which could move the two-joints on a horizontal plane. The study formulized accurate equations for partial differentiation to capture the modes of the flexible robotic arm by comparing the equations of the boundary conditions with the equations of partial differentiation centered on the elbow. This comparison of the boundary condition equations with the

equations of partial differentiation was used to produce precise eigenfrequencies. The flexible-link robotic arm manipulator's movement was thirty percent because of the use of eigenfrequencies. Hence, eigenfrequencies have proved to be a significant measure of the configuration of the flexible link robotic arm manipulator. Later, a segregation of the elastic deformity and the rigid body motion was carried out using the predefined path of the rigid body.

Further work on geometric stiffness for a flexible robotic arm shows that the first order equation of motion was derived to explain the movement of a flexible robotic arm [72]. Another researcher suggested a mechanism to incorporate the finite element method for deriving a dynamic equation for a planar two-link flexible robotic arm with the help of Elementary Beam Theory (EBT). Moreover, the study used the finite element method to determine the system force and bending of joints of a flexible robotic arm. While examining the forward dynamics, the inertial loads, also called the acceleration vector, was kept in view and in the reverse dynamic examination. The dynamic equations were used to find out the acceleration of the specified coordinates. The study employed the coupling effects, which were caused by the elastic deformation by formularizing the equations, and considering the impacts of inertia, deformation and nonlinearity of the link's motion. Furthermore, this study incorporates the dynamics of the manipulator in the form of the actuator-servo effect. Later, a mechanism was employed to determine the dynamic strength as well as the dynamic stiffness of the flexible robotic. Lagrange's dynamic differential equations were used for modelling flexible robotics by incorporating the integrated model and the robot system dynamics. This study estimates the dynamic strength as well as the dynamic stiffness using the dynamic deformation model, dynamic stress model and the reliability model. Furthermore, the study applies the Monte Carlo technique for acquiring the random dynamic parameters for flexible robotics.

This mechanism for determining dynamic stiffness of a flexible robotic device can be verified by using it for a multi-link flexible robotic arm manipulator. This examination of dynamic stiffness of a flexible robotic was further investigated. The study used the Assumed Mode Model to capture the dynamics of the manipulator.

A significant study on robotic modelling shows that the minimum energy was produced while tracking the trajectory of the joints of a flexible robotic by employing

a Genetic Algorithm. In addition to that, this study applies the principle of Extended Hamilton as well as the Euler–Bernoulli's beam in order to obtain the numerical model of the flexible robotic. Furthermore, this technique was verified through the mathematical simulation. This study proposed the incorporation of the specified model into the finite element method in order to verify its efficiency in case of a two-link as well as multi-link flexible robotic arm manipulators. Like other significant studies, the results of this study have also been verified using the mathematical simulation [73].

In another significant research, a mechanism was proposed to minimize vibrations in a flexible robotic arm with two-flexible modes. This can also be accomplished by employing the method of command input pre-definition. This method takes into account several pre-conditions and constraints, but its results were verified through mathematical simulation [74].

A study formalized a filter for adaptive disturbance rejection by incorporating the input shaping control law to minimize the vibration by applying an unmanned spacecraft for (2-DOF) flexible robotic arm manipulator. The results were verified through mathematical simulations. The third and final type of model for the flexible robotic arm manipulator is the lumped parameter method. The lumped parameter method develops the robotic manipulator model in terms of a lump of masses as well as massless springs. The lumped parameter method is considered as an easy method among the three types of flexible manipulator models; however, its accuracy is compromised in some cases. This method consists of two approaches: It is applied in an experimental way, and a series of experiments are conducted to find out the parameters [75].

The researchers also proposed a method for two conjoined flexible robotics for tackling a rigid object employing the lumped parameter. A novel method was suggested for analyzing the robotic arm manipulator using the lumped model. Moreover, the lumped parameter was proposed for estimating the elastic mechanical process in order to illustrate the required trajectory for a flexible robot [75].

Some similar studies were published in 1996. One of them suggested the variables should be disjoined from the joint for the elastic variables for illustrating the trajectory control of a flexible robot. In the same year, another study applied this method to handle the force of the constrained flexible robotic arm manipulator. Another approach was, as mentioned earlier, the lumped parameter method, which helps analyzing the

stability of a flexible robotic device. In this algorithm, the authors concluded: When multiple links of the manipulator are taken into account, many control issues can be curbed as opposed to the non-flexible or rigid manipulators. A number of implicit complexities were faced while developing a flexible robotic arm manipulator. One of them was the non-minimum phase, which is the consequence of the zeroes of the system in the half s-plane. At the same time, there can be several causes behind non-minimum phase in a flexible robotic arm manipulator. The firstly is the occurrence of non-collocated actuation of a flexible robotic arm manipulator. In this case, the transfer function involves calculating the position output, in which, torque acts as an input. Similarly, mapping inputs from the flexible link robotic arm manipulators can be made linear. In this case, the input torque produced on the joint represents a non-minimum phase, which has several negative effects on the function of a flexible robot. It may diminish the trajectory tracking performance because of the constrained control input, and also causes increased error in trajectory tracking.

The non-minimum phase makes the functioning of a robotic arm control more complex and limits the bandwidth. For flexible robotic manipulators, under-actuation means that the flexible robotic has less actuators, so it is very significant. The mechanism has its own dynamic coupling needs. The under-actuated robotic manipulator control was studied in terms of degrees-of-freedom, which are controlled by the actuated degree-of-freedom. Another problem with the minimum-phase manipulators is the non-holonomic constraints. The non-holonomic constraints describe the coordinates, which are not dependent on the time-derivative. In some studies, the modelling of n -link robotic arm manipulators have been carried out with the help of hierarchical control concept. This study acquires the equation of motion by applying the multi-body system technique [76].

Another significant approach was formulating dynamic equations for a 3-D multi-link robotic arm manipulator, which caters to the prismatic as well as the revolute joints. Moreover, the dynamic and kinematic implications of the flexible multi-link robotic arm model were analyzed using a free-moving platform in a cosmic space. Other strategies for capturing the control of a robotic arm model include end-effector regulation, joint tracking of trajectory as well as trajectory tracking of the end-effector [77].

2.9 Robotic Motor System Control

The main control task is start-to-end motion of a robot arm that is possible by designing an algorithm to solve the robot path planning, which is the goal of this thesis. To generate power in robots, an electrical DC or AC motor is required. Robotic applications typically require a good velocity, high torque and high accuracy to be considered as efficient robots. A motor transfers torque to the points on the robotic arm links using mechanical links like chains. Mostly, motor friction is thought to inflict certain limitations on the performance of a robotic manipulator. Among the earliest studies on this topic, a study [78] modelled a robotic manipulator, monitored the effects of a DC motor particularly motor friction, and analyzed the overall performance of the robot. This study suggests the usage of an adaptive controller to recompense the friction effects of the motor on the robotic manipulator. Designing a control system usually involves modelling different types and sizes of DC motors for various robotic applications. For example, a study [79], modelled the usage of a DC motor in Wheeled Mobile Robots (WMR) as a wheel driving machine while controlling the rotation of the motor. Wheeled Mobile Robots are mobile robots, which perform with the help of sensors. The sensors help robots avoid obstacles on the robotic motion path. In this case, the wheels of a Wheeled Mobile Robots are steered by a DC motor. This study [79] models a DC motor with the help of control mechanisms and models the DC motor by first employing its electrical circuit diagram.

After that, the DC motor system is characterized in terms of a set of mathematical equations. Based on that, a suitable transformation function for the equations is devised. Finally, the model is simulated with simulation software such as MATLAB. The robotic motor modelling proposes an adaptive fuzzy approach with a brushed DC motor system in case of a flexible joint robotic manipulator but most of the researchers preferred modelling of Brushless DC Motor (BLDM) over the brushed motor for several robotic applications. It has been shown that Brushless DC Motor is connected to a flexible robotic arm also called as direct-drive Brushless DC Motor, which has so far yielded a much-improved performance for force-controlled actuators [80]. Among more notable studies in this regard, the researchers [81] used Brushless DC Motor for direct-drive for tracking control in the robotic applications. Direct-drive motors refer to the motors which are directly attached to the robotic links without any gear.

Similarly, brushless DC motors were also considered in robotic manipulators in a mathematical form.

In the studies, manipulator-controlled direct-drive robots have been extensively proposed, as they ensure high performance of a robotic arm manipulator, since they do not require gears for interacting with a motor. For obtaining the required amount of torque, direct-drive manipulators are effective, so that the friction is minimized [80]. The reasons for using a Brushless DC Motor have been discussed in the study, among which, the most notable factor is the performance boost. The incorporation of direct-drive brushless DC motor in robotic manipulators has been studied particularly in case of commercial robots. The designers designed a low-cost commercial micro robot named Alice using two direct-drive and watch type DC motors because of its demand in the toy market. It had low power consumption and easy control of the watch-type DC motors, which assured efficient design.

In this scenario, the robotic motors drove the robotic manipulators along a particular path assuring obstacle avoidance. Another significant study analyzed the speed and power consumption characteristics of a DC motor in the mathematical form as well as in terms of mechanical energy consumption of a DC motor. The study proposes a method to minimize the mechanical energy consumption of the DC motors for robotic manipulators [81]. The minimization of energy of robotic motors was also studied extensively by some other researchers [82, 83].

2.10 Summary

The literature pertaining to this field is extensive, and it has many studied cases of motion planning and trajectory planning for a robot arm. A majority of researchers used the optimization algorithm technique. Some others have used the pseudo-inverse analytical process for optimization. The researchers observed many constraints like precedence, geometric and connectivity constraints, cost of the assembly and the least stability criteria, which were considered during assembly sequence generation that has been reported in many studies.

In many cases, numerical optimization methods were used. Our research on robots focused on finding a minimum-time trajectory planning method for the path of a robot

manipulator. The modelling was done to limit constraints in terms of kinematic constraint equations, which are essential to solve.

Most of the research work deals with the derivation of the robot equations depending on several strategies. This study focuses on algorithm for direct/inverse kinematic analysis and modelling. Most of the relevant approaches focus on logical and sensing for obstacle avoidance. As far as the path planning problem is concerned, several studies deal with this subject based on logical computation, spline polynomials, and B-splines to drive the robot path in the joint space. This study focuses on Cartesian-space information for the motion planning for taking the end-effector from the starting-point to end-point.



CHAPTER 3

MATHEMATICAL MODELLING AND KINEMATIC ANALYSIS

3.1 Introduction

The conventional solution approach to kinematics is important in various fields of post-modern technology from computer graphics (e.g. character animation) to space exploration. All these fields of application are fundamentally required to evaluate both orientation and the position of the Cartesian-coordinates of end-effector and joint variables of a robot manipulator. To evaluate the position and orientation of end-effector and its joint variables, a researcher should apply homogeneous transformation matrix method. This method is a conventional tool to describe the kinematic relationship between the joint and the links. Moreover, this method of representation has been used for many decades for tracing the end-effector position of robot manipulators.

In the third current chapter, we mentioned the essential mathematical information to help readers understand the study and the analysis. We have used necessary mathematical tools, which are needed in robotic systems. The modelling problem should be applied before applying any control method to resolve a path planning problem. In this section, we are concerned with the development of the model for a SCARA robot (Self Compliant Articulated Robotic Arm), its kinematics, and dynamics of its formulation systems.

The goal of this chapter is applying modelling equations of a robot arm, which show the forces needed for the movement of robots. For effectively controlling the position of a robotic manipulator, the dynamics of the manipulator should be known and explicitly specified in terms of the force, which is exerted on it to cause motion. The dynamic modelling involves the dynamic movement of a robotic arm manipulator, which is generated by the actuators and the force, which is applied on them. To control the application of force on a manipulator is a significant challenge because a weaker force causes the manipulator to respond slowly while too much force might crack or

destroy it. This force is applied in terms of the torque generated by the actuators for the dynamic motion of a robotic manipulator. The structure of a robotic arm will be fully described along with the parameters of its links.

The structure of the robotic manipulator should enable the derivation of the kinematic and dynamic equations, which will be used later in the design process of the controllers. There are two different approaches to modelling of a robot in general: The Euler–Lagrange (EL) formulation and the Newton–Euler (NE) method. The former treats the manipulator as a whole; so, the dynamic analysis is based on Lagrangian function, which uses the description of both kinetic and potential energies of the system. This formulation only states the differential equations that determine forces and torques of individual actuators. As a result of using Newton–Euler approach through dynamic equations, the required forces and torques of actuators help computing the forces and moments acting on the joints. This formulation separately treats each link of a manipulator. It means that the equations describing the linear and angular motion of the links are separately expressed for each body with respect to its coordinate frames.

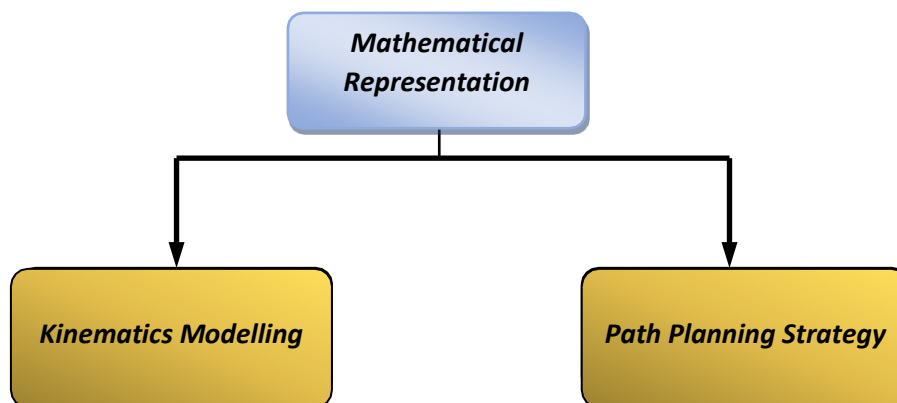


Figure 3.1: Proposed approach to position analysis of robot manipulator.

3.2 Dynamic Modelling of Robotic Manipulator

This section presents the geometric structure description of manipulator robots using the (DH) representation, named after Denavit and Hartenberg [85]. It is the first step to compute the dynamic model for simple open-chain manipulator robots. We also presented the Lagrange formalism, using which; we established the general equation

of motion for manipulator robots. Finally, to illustrate the dynamic model of manipulator robots, we used a manipulator robot that was used in the simulation to validate the theoretical study presented in this thesis. To operate such an articulated system, we must resort to laws for stability of this system. These control laws sometimes use some elements of the dynamics of the robot, so the parameters are known. The dynamic equation of a manipulator consists of mathematical models of equations of motion of the mentioned robot. There are two types of a model; the inverse model has been used in the control applications, however, this model provides particular torques, which are exerted by the actuators, according to positions, speeds, accelerations, and the direct dynamic model used in the simulation, which provides joint acceleration based on positions, speeds and joint torques.

Several methods have been used to obtain the inverse dynamic model. We formulated Newton–Euler and Lagrange–Euler processes. We used the Lagrange–Euler formulation because it is simple and systematic, and it describes the dynamic model of the system in terms of work and energy using generalized coordinates. The first part of this chapter presents features of basic manipulative robots such as the degree-of-freedom and the notion of singularity, etc. In the second part, we will present geometric description of the systems articulated for particularly manipulative robots; a description based on two methods, one of them is the Denavit–Hartenberg (DH) standard; however, this method is not powerful enough especially in the case of articulated tree structures or parallel. The other more modern and more powerful method is modified (DH). In the third part of this chapter, the formalism of Lagrange will be applied to a model robot. This particular approach is quite simple to implement, and it is implementable both through computer-assisted and manual calculations. In this part, we will treat the different structural properties of the dynamic model, which are very effective in the correctors' synthesis. By the end of this chapter, the calculations of the model of the SCARA robot will be accomplished and used for validation.

3.3 Kinematics of Robot Manipulator

Kinematic modelling is an important section of robot technology. This section deals with the kinematic position of a mechanical system. Robot kinematic refers to the analytical study of motion and structure of robot manipulators. Formulation of the

suitable kinematics models for a robot mechanism is very crucial for analyzing the behaviour of industrial manipulators. There are two types of problems in the kinematic analysis of robot manipulators: Forward and Inverse Kinematics.

1. Calculating the Forward Kinematic (FK).
2. Calculating the Inverse Kinematic (IK).

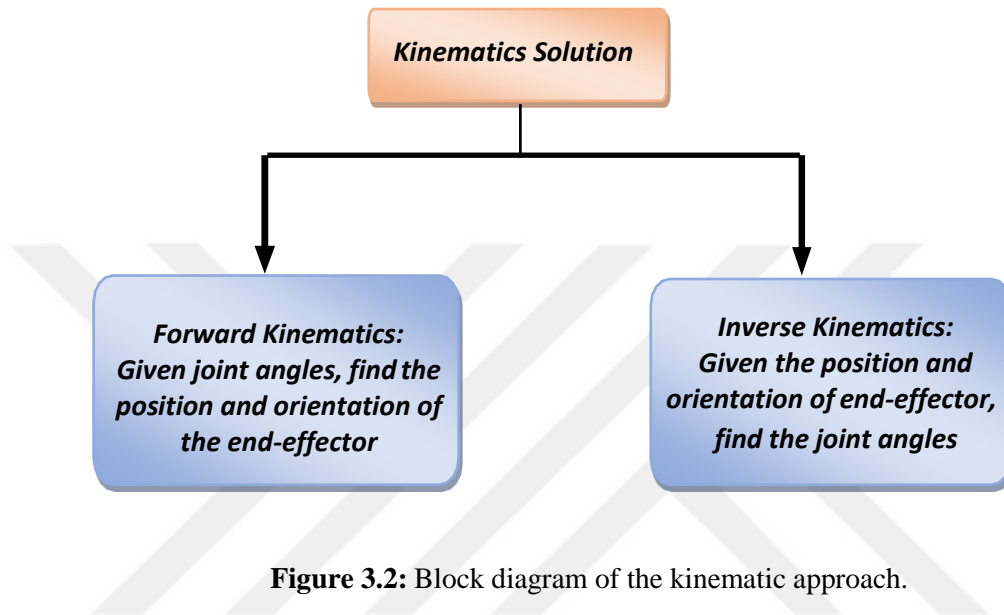


Figure 3.2: Block diagram of the kinematic approach.

The kinematics of a robotic refers to the mathematical equations that describe the forward and inverse relationships between the Cartesian position coordinates of the manipulator end-effector and the angular positions of the revolute joints. These equations are very important in the process of mapping the manipulator's desired trajectories from the Cartesian-space to the joint space and vice versa. The robotic manipulator kinematics can be divided into inverse and direct kinematics.

The direct kinematics describes the Cartesian position coordinates of the end-effector as functions of the joint angular positions. On the other hand, the inverse kinematics describes the joint's angular positions as functions of the end-effector Cartesian-coordinates. Figure 3.1 shows a simplified block diagram of kinematics modelling.

In this thesis, the following steps describe a general analytical procedure for deriving the direct and inverse kinematics of a manipulator, which help deriving the kinematics of the robotic arm, as shown in Figure 3.5.

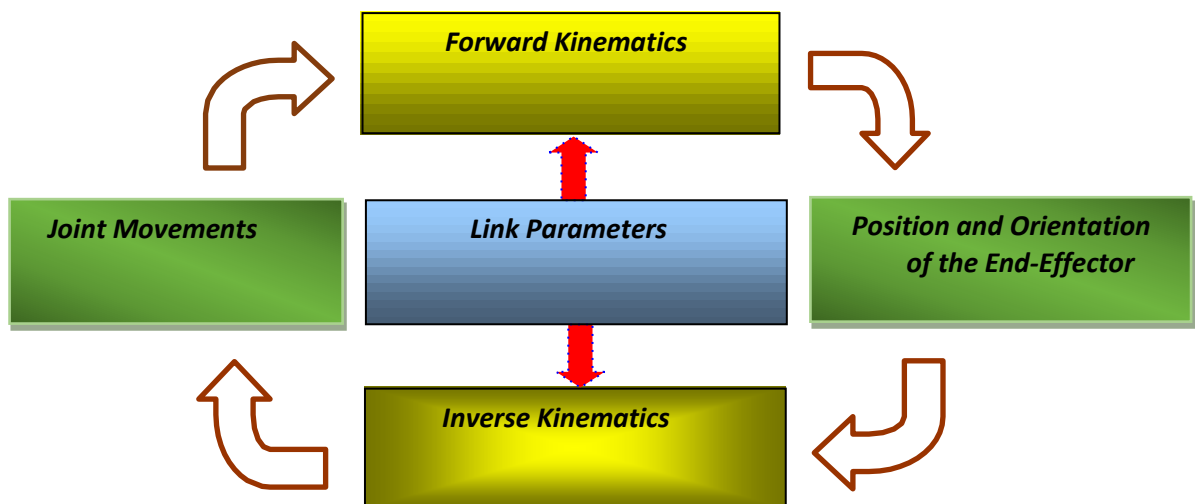


Figure 3.3: The forward and inverse dynamic problems.

3.3.1 Forward (Direct) Kinematics

A manipulator is a series of links connected by joints either 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 direct kinematics. To obtain the direct kinematic equations for the manipulator, the following steps must be accomplished. For forward kinematics, all the link lengths and joint angles of the robot system must be available.

In forward kinematics, the values of joint and link variables are substituted in a set of equations that define a particular configuration of the robotic system. Forward kinematic problem is one of the most significant obstacles faced in the dynamic motion of a serial-link robotic manipulator. The joint variables are the angles between the links in the case of revolute or rotational joints, and the link extension in the case of prismatic or sliding joints [84]. The forward kinematic problem has been addressed by several studies; however, it is still an open-ended research question. The problem of forward kinematics occurs when a robot moves from a position A and B with respect to a common coordinate system.

Forward Kinematic (FK) problem defines the collective impact of a set of joint angles between the links. Consequently, it determines the position and angle of the end-effector if the values of joint variables are given. For the representation of a forward

kinematics problem, a fixed coordinate system is created, which is termed as the base frame, and it serves as a reference to the robotic manipulator.

$$x = f(\theta) \quad (3.1)$$

Forward kinematics is used to solve the end-effector position and orientation given in the joints' values. Giving specific values to the joints is the only solution to the problem and hence, there is only one-position and orientation for the end-effector.

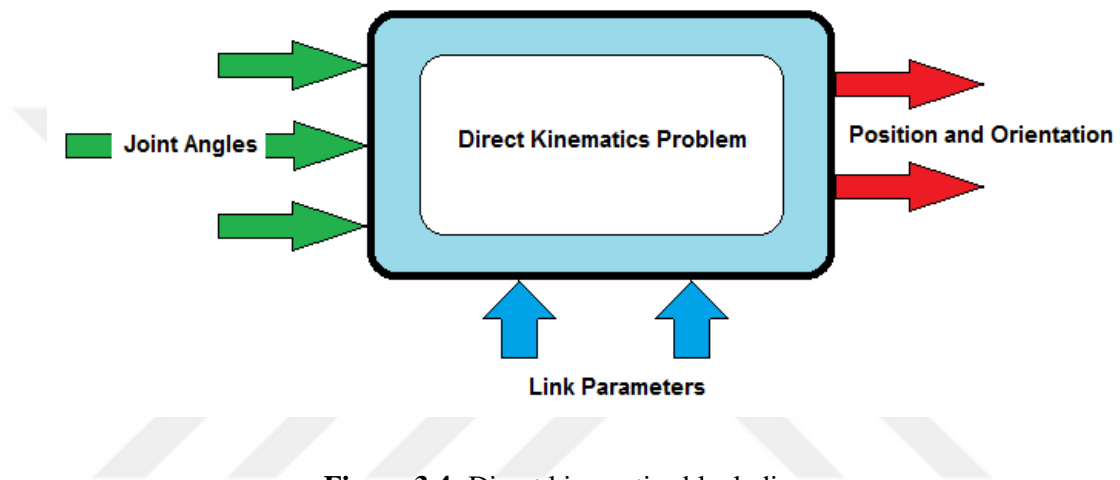


Figure 3.4: Direct kinematics block diagram.

In order to derive the forward kinematics of a robotic manipulator, the following steps must be followed:

3.3.1.1 Derivation of Denavit–Hartenberg (DH) Parameters

Denavit and Hartenberg (DH) have proposed a systematic method for performing the passage between adjacent joints of a mechanical system. This method is called the Denavit–Hartenberg [85]. The standard Denavit–Hartenberg is mainly about kinematic chains while each joint has a degree-of-freedom of rotation or translation. The translation and the rotation are so-called kinematic assemblies of lower order, which means that the adjacent surfaces remain in contact during their movement. The six possibilities thus offered are the hinge, the slide, the cylindrical bearing, spherical ball joint, torque screw-nuts, and plane-to-plane motion. There are many ways to describe the robot configuration. One way is to use Denavit–Hartenberg notation of describing the link and its connection to the neighboring link which is the joint variable.

Denavit–Hertenberg uses the four-parameter method for the manipulator kinematics. These parameters allow the calculation of the vector for different links and the rotation matrix, as illustrated in Figure 3.5.

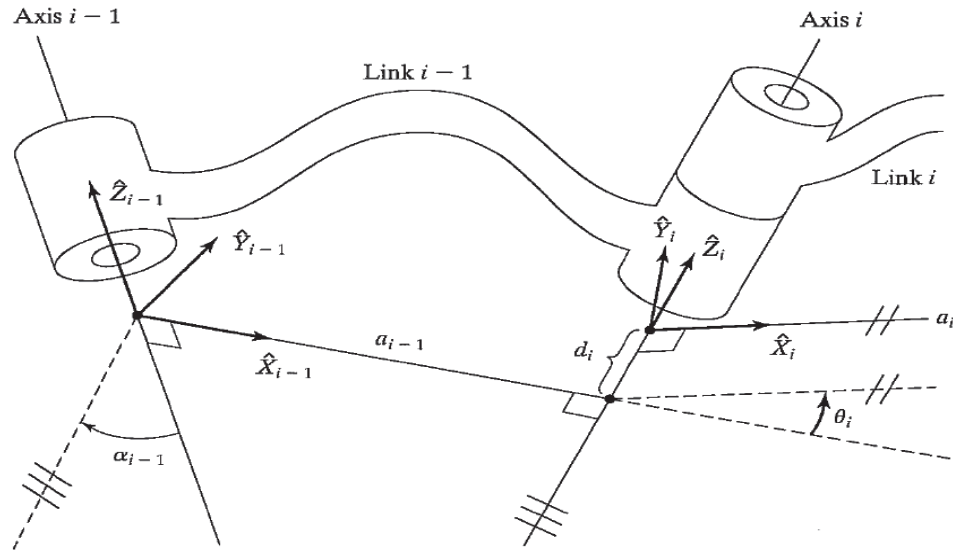


Figure 3.5: Modified Denavit–Hertenberg (DH) parameter for a revolute joint.

There are four variables in Denavit–Hertenberg method notation. They are as follows [86]:

1. a_{i-1} : (Length) the measured distance along X_i from Z_i to Z_{i-1} .
2. α_{i-1} : (Twist) the measured angle about X_i from Z_i to Z_{i-1} .
3. d_i : (Offset) the measured distance along Z_i from X_{i-1} to X_i .
4. θ_i : (Angle) the measured angle about Z_i from X_{i-1} to X_i .

These parameters are illustrated in Figure 3.5 for a revolute joint. It is important to notice that d_i and α_{i-1} do not change unless the robot configuration changes. Here, a_{i-1} is the link length and α_{i-1} is the link twist. After assigning the frames, we can make a table for representing the four parameters for the three-link (3-DOF), SCARA robot manipulator. Table 3.1 shows the parameter type associated with revolute or prismatic parameter joints. (See appendix A)

3.3.1.2 Derivation of the Homogenous Transformation Matrices

We now present the matrices of passage from link i to $i - 1$ after constructing the Denavit–Hertenberg (DH) parameters. The next step is to construct the transformation matrix for each link from the following equation. The parameters α_{i-1} and d_i are

constants. The geometry of the link determines them. One of the other two parameters θ_i and a_{i-1} vary as the joint moves. For a revolute joint, variable θ_i represents the joint displacement, while a_{i-1} is a constant. On the other hand, for a prismatic joint, parameter a_{i-1} is the variable representing joint displacement and θ_i is a constant [88]. The joint variable q_i associated to the i^{th} joint is defined as:

$$q_i = (1 - \sigma_i)\theta_i + \sigma_i d_i \quad (3.2)$$

Where:

$$\sigma_i = \begin{cases} 0 & \text{if joint } i \text{ is rotational} \\ 1 & \text{if } i \text{ joint prismatic} \end{cases} \quad (3.3)$$

A commonly used convention for selection of frames of references in robotics is the application of the Denavit–Hertenberg [85], which has been shown in Figure 3.6. It can be seen that the homogenous transformation matrix T_i^{i-1} defines frame i with respect to frame $i - 1$, which can be obtained as:

$$T_i^{i-1} = Rot(X, \alpha_{i-1})Tran(X, a_{i-1})Tran(Z, d_i)Rot(Z, \theta_i) \quad (3.4)$$

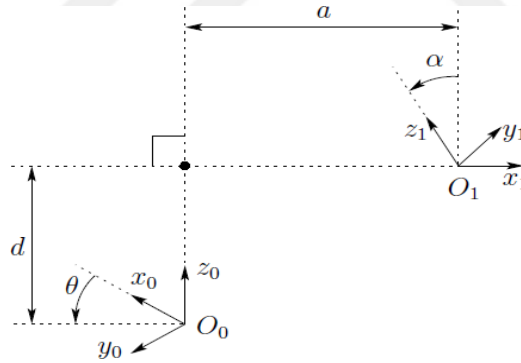


Figure 3.6: Denavit–Hertenberg (DH) frame assignment.

With:

$$Rot(X, \alpha_{i-1}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_{i-1}) & -\sin(\alpha_{i-1}) & 0 \\ 0 & \sin(\alpha_{i-1}) & \cos(\alpha_{i-1}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

$$Tran(X, a_{i-1}) = \begin{bmatrix} 1 & 0 & 0 & a_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

$$Tran(Z, d_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.7)$$

$$Rot(Z, \theta_i) = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.8)$$

Then,

The overall transformation matrix between the end-effector frame and the base frame will be:

$$T_i^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_{i-1} \\ \sin(\theta_i)\cos(\alpha_{i-1}) & \cos(\alpha_{i-1})\cos(\theta_i) & -\sin(\alpha_{i-1}) & -d_i\sin(\alpha_{i-1}) \\ \sin(\alpha_{i-1})\sin(\theta_i) & \sin(\alpha_{i-1})\cos(\theta_i) & \cos(\alpha_{i-1}) & d_i\cos(\alpha_{i-1}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.9)$$

for $(i = 1, \dots, n)$

We can notice that the homogenous transformation matrix (3.9) has the general form:

$$T_i^{i-1} = \begin{bmatrix} A_i^{i-1} & O_o^{i-1} \\ 0 & 1 \end{bmatrix} \quad (3.10)$$

Here, T_i^{i-1} matrix is linked with the end of the arm frame, $i - 1$ to its base i , A_i^{i-1} is the (3×3) matrix defining the orientation of frame i with respect to the orientation of frame $i - 1$, and O_o^{i-1} defining the origin of frame i with respect to frame $i - 1$.

3.3.2 Inverse Kinematics

This section is concerned with the "Inverse Problem" in terms of the end-effector position and orientation. This is the problem of inverse kinematics, and it is, in general, more difficult to resolve than the forward kinematic problem [86]. For inverse kinematic problem, a researcher should place the robot manipulator at a desired-location, and then calculate the values of joint of the hand of the robot, which is called as inverse kinematics.

The inverse kinematics defines the particular configuration of the robotic system, so it is possible to calculate the values of joint and link variables of the robot. It is relevant

to most of the real applications and tasks than forward kinematics. The problem is non-linear, so we must check whether there is a solution, and then check whether there are multiple solutions, and finalize the strategy to find the solution. The existence of a solution mainly depends on the manipulator's workspace and whether the end-effector's desired position lies in the accessible workspace or not. Another possibility for the problem is to find multiple solutions and decide about the preferable solution. A criterion to build the decision on is finding the closest solution with obstacle avoidance. The analytical equation of inverse kinematics is given by:

$$\theta = f^{-1}(x) \quad (3.11)$$

Inverse Kinematics (IK), as the name suggests, can be thought of as an inverse of direct kinematics, that is, if the position and angle of the end-effector is given, it determines the joint angles of both the links. Clearly, the problem of finding the joint angles is more complex than the forward kinematics. Inverse kinematic approach can be approximated as shown in Figure 3.7.



Figure 3.7: Inverse kinematics.

3.4 Dynamic Modelling of Robot Manipulator

Robot manipulators can be described mathematically in different ways. For robot design purposes, it is necessary to make a mathematical model that reveals the dynamic behaviour of the manipulator. This mathematical model has been derived using the Lagrangian mechanics [89].

In this section, we have analyzed the dynamic behavior of manipulator arms, which we have described in terms of the rate of change with respect to time in the arm configuration in relation to the joint torque exerted by the actuators. This relationship can be expressed by a set of differential equations called as equations of motion, which govern the dynamic response of the arm linkage to input joint torque.

The dynamics describe the relationship between forces, torques and motion. The kinematics describes the motion without the consideration of the forces and torques, while the dynamic equations describe the relationship between forces and motion. The model of motion is important for designing a robot, simulation, animation of robot motion and designing control algorithm.

Euler–Lagrange equation is a known method to describe the evaluation of mechanical model. The Lagrangian system must be solved in order to determine the Euler–Lagrange function. The two main formalisms that are generally used in the model equation of the robot manipulator dynamics are described below:

1. Newton–Euler (NE) Formulation.
2. Lagrangian–Euler (LE) Equation.

3.4.1 Newton–Euler Formulation

A method used for analysing the dynamics of robot is recursive Newton–Euler formulation, which is described in this section. More detailed derivation of this formulation has been given in the literature [84]. The inverse dynamics of an open kinematic chain structure can be calculated in three steps:

1. Find the acceleration and the velocity of each body in the structure.
2. Find forces required to produce computed accelerations.
3. Find the forces transmitted across the joints from the forces acting on the bodies.

The derivation of the Newton–Euler formulation for an n -link manipulator is completely based on a previous research [84]. It is necessary to choose the frames: $\{0, \dots, n\}$. Here, frame 0 stands for an inertial frame, and frame i is rigidly attached to link i [84]. The following list presents several vectors and scalars expressed in the frame i , which are required for the Newton–Euler process.

3.4.2 Euler–Lagrange Equations

In this work, we have presented the Lagrange–Euler and we do not consider that for simple open-chain robots. Lagrange's formalism exists in the first calculation of the Lagrange function of the manipulator robot; therefore, the difference between its kinetic energy K and its potential energy U is calculated and then applied the dynamic model by applying derivatives and partial derivatives. The n scalar equations are

obtained called as Lagrange equations. To obtain the dynamic model with the Lagrange–Euler formalism, we must first determine the kinetic energy $K(q, \dot{q})$ and the potential energy $U(q)$ because the Lagrangian $L(q, \dot{q})$ is given by:

$$L(q, \dot{q}) = K(q, \dot{q}) - U(q) \quad (3.12)$$

The Lagrange–Euler equation for conservative systems has been applied [87]; so, the manipulator robot's movement equations are given by:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \tau \quad (3.13)$$

Where, q and $\dot{q} \in R^n$ are the coordinates and the generalized velocities, respectively. $\tau \in R^n$ is the vector of generalized forces or torques. The Lagrangian approach can be expressed in the following form:

$$L = K - U \quad (3.14)$$

Here, L is the Lagrangian of the robotic manipulator, K is the kinetic energy of the system, and U is the potential energy of the system.

To obtain the general arm dynamic equations of motion, we will compute the total kinetic and potential energies of the system, the Lagrangian, and then substitute them into the Lagrange's equation (3.13) to obtain the final result.

3.5 General Expression for Kinetic Energy

First, we'll derive expression for the kinetic energy of the robot noting that the kinetic energy of any rigid object consists of translational kinetic energy due to linear velocity of the center mass, and the rotational kinetic energy that emerges due to angular velocity of the link.

Given a point on link i with coordinates of r_i with respect to frame i have been attached to the link. The base coordinate of this point is:

$$r = T_i^0 r_i \quad (3.15)$$

Here, $(T_i^0 = T_1^0 T_2^1 \dots T_i^{i-1})$ is a homogeneous transformation $\in R^{(4 \times 4)}$ that is a function of the joint variables $\{q_1, q_2, \dots, q_i\}$. Consequently, the velocity of the point in the base coordinates is:

$$v = \frac{dr}{dt} = \sum_{i=1}^j \left\{ \frac{\partial T_i^0}{\partial q_i} \dot{q}_i \right\} r_i + T_i^0 \sum_{i=1}^j \left\{ \frac{\partial r_i}{\partial q_i} \dot{q}_i \right\} \quad (3.16)$$

Hence, ($i = 1, 2, \dots, n$)

Since $\frac{\partial r_i^0}{\partial q_i} = 0$ for $i > j$, we can replace the upper limit of the summation with n which represents the number of joints.

$\left\{\frac{\partial r_i}{\partial q_i}\right\} = 0$ because it is constant with respect to frame i that is attached to the link.

The kinetic energy of an infinitesimal mass dm at r_i has a velocity vector, which is described by:

$v = [v_x \ v_y \ v_z]^T$ is defined as:

$$dK_i = \frac{1}{2} (v_x^2 + v_y^2 + v_z^2) dm \quad (3.17)$$

$$dK_i = \frac{1}{2} \text{trace}(vv^T) dm \quad (3.18)$$

Using the expression of the velocity v given by the equation (3.16), we have obtained:

$$dK_i = \frac{1}{2} \text{trace} \left\{ \sum_{j=1}^n \sum_{k=1}^n \left(\frac{\partial r_i^0}{\partial q_j} \right) (r_i r_i^T dm) \frac{\partial r_i^0}{\partial q_k} \dot{q}_j \dot{q}_k \right\} \quad (3.19)$$

Thus, the total kinetic energy for link i is given by:

$$K_i = \int_{link\ i} dK_i \quad (3.20)$$

By substituting dK_i by the expression (3.19), we can move the integration symbol inside the summations. So, the inertia matrix $\in R^{(4 \times 4)}$ for the link i is given by:

$$I_i = \int_{link\ i} r_i r_i^T dm \quad (3.21)$$

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix} = \text{constant} \quad (3.22)$$

Here, the integrals are taken on the body volume i . It is a constant matrix that is evaluated once for each body. It depends on the geometry and distribution of the body mass i . It is expressed as follows:

$$\begin{aligned} I_{xx} &= \iiint (y^2 + z^2) \rho(x, y, z) dx dy dz = \int (y^2 + z^2) dm \\ I_{yy} &= \iiint (x^2 + z^2) \rho(x, y, z) dx dy dz = \int (x^2 + z^2) dm \\ I_{zz} &= \iiint (y^2 + x^2) \rho(x, y, z) dx dy dz = \int (y^2 + x^2) dm \end{aligned} \quad (3.23)$$

Cross products of inertia:

With:

$$\begin{aligned}
 I_{xy} = I_{yx} &= \iiint -xy \rho(x, y, z) dx dy dz = - \int xy dm \\
 I_{xz} = I_{zx} &= \iiint -xz \rho(x, y, z) dx dy dz = - \int xz dm \\
 I_{yz} = I_{zy} &= \iiint -yz \rho(x, y, z) dx dy dz = - \int yz dm
 \end{aligned} \tag{3.24}$$

The function $\rho(x, y, z)$ is the density of the body.

and first moments:

$$m\bar{x} = - \int x dm, m\bar{y} = - \int y dm, m\bar{z} = - \int z dm \tag{3.25}$$

It is equal to:

$$I_i = \begin{bmatrix} \int x^2 dm & \int yx dm & \int zx dm & \int x dm \\ \int xy dm & \int y^2 dm & \int zy dm & \int y dm \\ \int xz dm & \int yz dm & \int z^2 dm & \int z dm \\ \int x dm & \int y dm & \int z dm & \int dm \end{bmatrix} \tag{3.26}$$

Here, m is the total mass of the body, so $r_i = [\bar{x} \ \bar{y} \ \bar{z} \ 1]^T$ represents the vector of coordinates based on the body's center of gravity i in the reference R_i , we can write:

I_i is the inertia matrix as shown below:

$$I_i = \begin{bmatrix} \frac{-I_{xx}+I_{yy}+I_{zz}}{2} & -I_{xy} & -I_{xz} & m\bar{x} \\ -I_{xy} & \frac{I_{xx}-I_{yy}+I_{zz}}{2} & -I_{yz} & m\bar{y} \\ -I_{xz} & -I_{yz} & \frac{I_{xx}+I_{yy}-I_{zz}}{2} & m\bar{z} \\ m\bar{x} & m\bar{y} & m\bar{z} & m \end{bmatrix} \tag{3.27}$$

In equation (3.27), m is the mass, $(\bar{x}, \bar{y}, \bar{z})$ is the location vector of the center of mass, expressed in terms of $\{i\}$ local coordinate frame, and $I_{xx}, \dots, I_{xy}, \dots$ are moments/products of inertia pertaining to link i with respect to $\{i\}$ local coordinate frame. These quantities are either given by the manufacturer's specifications or they can be calculated from the other quantities:

This is a constant matrix, which is evaluated once for each link. It depends on the geometry and mass distribution link i .

After defining I_i , we may write the kinetic energy of link i as described by:

$$K_i = \frac{1}{2} \text{trace} \left\{ \sum_{j=1}^n \sum_{k=1}^n \left(\frac{\partial T_i^0}{\partial q_j} \right) I_i \frac{\partial T_i^{0T}}{\partial q_k} \dot{q}_j \dot{q}_k \right\} \quad (3.28)$$

The total arm kinetic energy of the manipulator is then written as:

$$K = \sum_{i=1}^n K_i = \frac{1}{2} \sum_{i=1}^n \text{trace} \left\{ \sum_{j=1}^n \sum_{k=1}^n \left(\frac{\partial T_i^0}{\partial q_j} \right) I_i \left(\frac{\partial T_i^{0T}}{\partial q_k} \right) \dot{q}_j \dot{q}_k \right\} \quad (3.29)$$

Since the trace of a sum of matrices is the sum of individual traces, we may interchange the summation and the trace operator to obtain:

$$K = \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n m_{jk}(q) \dot{q}_j \dot{q}_k \quad (3.30)$$

or

$$K = \frac{1}{2} \dot{q}^T M(q) \dot{q} \quad (3.31)$$

Here, the arm inertia matrix $M(q)$ has the elements defined as follows:

$$m_{jk}(q) = \sum_{i=1}^n \text{trace} \left\{ \left(\frac{\partial T_i^0}{\partial q_j} \right) I_i \frac{\partial T_i^{0T}}{\partial q_k} \right\} \quad (3.32)$$

Since the kinetic energy is a scalar quantity, so:

$$\begin{aligned} K &= \frac{1}{2} \dot{q}^T M(q) \dot{q} = K^T = \left(\frac{1}{2} \dot{q}^T M(q) \dot{q} \right)^T \\ K &= K^T, M(q) = M(q)^T \\ \text{or } m_{jk}(q) &= m_{kj}(q) \end{aligned} \quad (3.33)$$

3.6 General Expression for Potential Energy

If link i has mass m_i and center of gravity \bar{r}_i expressed in the coordinates of its frame i , the potential energy of the link is given by \bar{r}_i .

Let the potential energy of the manipulator be U and let each of its link potential be U_i , it is expressed by:

$$U_i = m_i g^T T_i^0 \bar{r}_i \quad (3.34)$$

Representing the coordinates of the center of gravity in the coordinates, we have the total potential energy of the robot arm, which is:

$$U = m_i g^T T_i^0 \bar{r}_i \quad (3.35)$$

Letting:

$$T_i^0 \bar{r}_i = P_i^0 \quad (3.36)$$

$$U = \sum_{i=1}^n U_i = \sum_{i=1}^n m_i g^T P_i^0 \quad (3.37)$$

Where:

$$g^T = [g_x \ g_y \ g_z]$$

g is the gravitational constant ($g = 9.8062 \text{ m/sec}^2$)

3.7 Equation of Motion

From equations (3.12) and (3.13), and considering the fact that the potential energy does not depend on the joint velocity of the equation of motion of the manipulator arm, we get the following expression:

$$\frac{d}{dt} \frac{\partial K(q, \dot{q})}{\partial \dot{q}_i} - \frac{\partial K(q, \dot{q})}{\partial q_i} + \frac{\partial U}{\partial q_i} = \tau_i \quad (3.38)$$

$$\frac{\partial K(q, \dot{q})}{\partial \dot{q}} = \frac{1}{2} \left(\frac{\partial \dot{q}^T M(q) \dot{q}}{\partial \dot{q}} \right) = \frac{1}{2} (M(q) \dot{q} + M(q)^T \dot{q}) = M(q) \dot{q} \quad (3.39)$$

$$\frac{\partial K(q, \dot{q})}{\partial \dot{q}_i} = \sum_{j=1}^n m_{ij}(q) \dot{q}_j \quad (3.40)$$

$$\frac{d}{dt} \frac{\partial K(q, \dot{q})}{\partial \dot{q}_i} = \sum_{j=1}^n m_{ij}(q) \ddot{q}_j + \sum_{j=1}^n \sum_{k=1}^n \left(\frac{\partial m_{ij}(q) q_k}{\partial q_k} \right) \dot{q}_j \quad (3.41)$$

$$\sum_{j=1}^n m_{ij}(q) \ddot{q}_j + \sum_{j=1}^n \sum_{k=1}^n \left(\frac{\partial m_{ij}(q) q_k}{\partial q_k} \right) \dot{q}_j - \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n \left(\frac{\partial m_{ij}(q) q_k}{\partial q_i} \right) \dot{q}_j + \frac{\partial U(q)}{\partial q_i} = \tau_i \quad (3.42)$$

Using the symmetry property of the matrix $M(q)$ we have:

$$\frac{\partial m_{ij}(q) q_k}{\partial q_k} \dot{q}_j - \frac{1}{2} \frac{\partial m_{kj}(q)}{\partial q_i} \dot{q}_k \dot{q}_j = \frac{1}{2} \left\{ \frac{\partial m_{ij}(q)}{\partial q_k} \dot{q}_k \dot{q}_j + \frac{\partial m_{ki}(q)}{\partial q_j} \dot{q}_k \dot{q}_j - \frac{\partial m_{kj}(q)}{\partial q_i} \dot{q}_k \dot{q}_j \right\} \quad (3.43)$$

$$M(q) = \frac{1}{2} \left\{ \frac{\partial m_{ij}(q)}{\partial q_k} + \frac{\partial m_{ki}(q)}{\partial q_j} - \frac{\partial m_{kj}(q)}{\partial q_i} \right\} \dot{q}_k \dot{q}_j \quad (3.44)$$

Using of Christoffel symbols:

$$C_{ij}(q, \dot{q}) = \sum_{k=1}^n C_{ijk} \dot{q}_k \quad (3.45)$$

Here:

$$C_{ij,k} = \frac{1}{2} \left\{ \frac{\partial m_{kj}(q)}{\partial q_i} + \frac{\partial m_{ki}(q)}{\partial q_j} - \frac{\partial m_{ij}(q)}{\partial q_k} \right\} \quad (3.46)$$

$$C_{kj} = C_{ikj}(q) \dot{q}_i \quad (3.47)$$

$$C_{kj} = \sum_i \frac{1}{2} \left\{ \frac{\partial m_{kj}(q)}{\partial q_i} + \frac{\partial m_{ki}(q)}{\partial q_j} - \frac{\partial m_{ij}(q)}{\partial q_k} \right\} \dot{q}_i \quad (3.48)$$

Substituting it into the equation of motion, we get:

$$\sum_{j=1}^n m_{ij}(q)\ddot{q}_j + \sum_{j=1}^n \sum_{k=1}^n C_{ijk}\dot{q}_k\dot{q}_j + \frac{\partial U(q)}{\partial q_i} = \tau_i \quad (3.49)$$

Now let $G(q)$ be the vector having i^{th} coordinates: $\frac{\partial U(q)}{\partial q_i}$. We can write equation (3.49) in the compact form as follows:

The modelling equation of a robotic manipulator system will be as follows [86].

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (3.50)$$

Where, $C(q, \dot{q})\dot{q}$ is the Coriolis/centripetal vector and $G(q)$ represent the gravity vector. Equation (3.50) is the final form of the dynamic equation for a robot.

Where:

τ : Joints torque $\in R^n$

\ddot{q} : Joint accelerations $\in R^n$

\dot{q} : Joint velocities $\in R^n$

q : Joint positions $\in R^n$

$M(q)$: Inertia matrix $\in R^{n \times n}$

$G(q)$: Vector of gravity forces $\in R^n$

$C(q, \dot{q})$: Matrix of centrifugal forces and Coriolis $\in R^{n \times n}$

3.8 SCARA Robot Arm Coordination System and DH Parameters

SCARA robot has high repetition accuracy ($<0.025 \text{ mm}$), fast acceleration and highest speed (2000-5000 mm/s) of movement to meet the demands of short cycle times in automated assembly. A typical SCARA robot structure is shown in Figure 3.8. It is compact, and the working envelopes are relatively limited (ranges $<1000 \text{ mm}$). The range of payloads that can be supported by this robot is (10-100 kg). These robots are best used for planar type tasks such as pick and place or assembly line sorting. Table 3.2 summarizes data for typical SCARA robots (See appendix A).

3.8.1 Kinematics Modelling of 3-DOF SCARA Robot

The SCARA robot is used in many applications such as pick and place, assembly, and packaging. The SCARA robot is a nonlinear dynamic system, which has some uncertainties including friction. In this section, the application of the equations of motion has been applied to simulate and illustrate the mathematical method. For the

sake of simplification, we have chosen the first (3-DOF) of the SCARA robot [90]. It is sufficient to position the terminal organ (the effector) at any point in the accessible space of the robot. First, we represent the articular configuration of the SCARA robot of the three degrees-of-freedom Revolute-Revolute-Prismatic (RRP), and then we'll calculate the dynamic model of this robot.

3.8.2 SCARA Configuration: Revolute-Revolute-Prismatic (RRP)

In order to support the theoretical development presented in this thesis, simulation studies have been performed using (3-DOF) mechanical structure moving in 3 dimensional-space of the 3-D SCARA robot. This robot model has already been considered by many modern researches. SCARA robot is shown in Figure 3.8. As its name implies, it is specially designed for assembly operations. Although the SCARA robot has a Revolute-Revolute-Prismatic (RRP) manipulator structure (Figure 3.9), it is quite different from the spherical configuration in the appearance or application, which is quite unlike the spherical design which has (Z_0, Z_1, Z_2) in perpendicular while the SCARA robot has (Z_0, Z_1, Z_2) in parallel.



Figure 3.8: The SCARA manipulator.

The values of the SCARA robot parameters are given by the following Table. 3.2. (See appendix A).

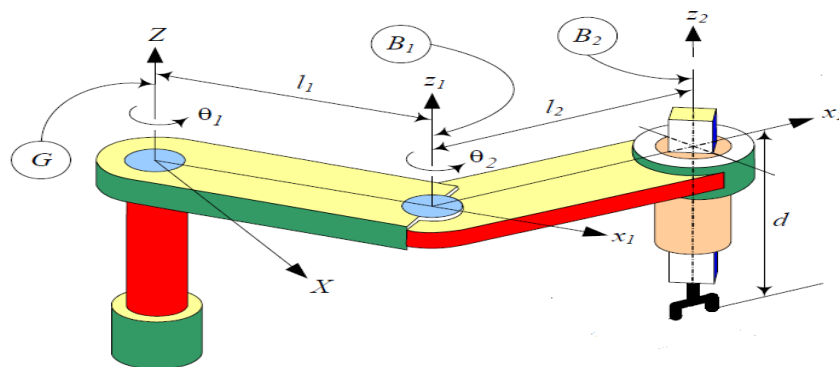


Figure 3.9: Coordinate frames are attached to SCARA robot arm (RRP) type.

Assign $(Z_0, Z_1, \dots, Z_{n-1})$ axes along the motion axes of joint 1, joint 2, ..., joint n , respectively, Therefore, the Z_i axis coincides with the motion axis of the joint $i + 1$.

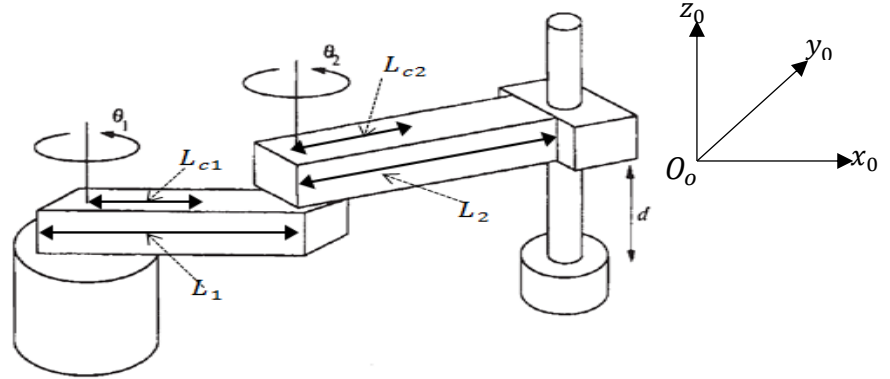


Figure 3.10: Three-degree-of-freedom (3-DOF) SCARA robot.

Here, m_1 : The mass of link-1; m_2 : The mass of link-2; m_3 : The mass of link-3; l_1 : The length of link-1; l_2 : The length of link-2; d : The position of the end-effector; l_{c1} : Position of the center of mass link-1; and l_{c2} : Position of the center of mass link-2. Link-1 has mass m_1 and inertia tensor I_{c1} , Link-2 has mass m_2 and inertia tensor I_{c2} , Link-3 has mass m_3 and inertia tensor I_{c3} .

3.8.3 General Form of Transformation Matrix

In this chapter, we will calculate the dynamic equation of the SCARA robot based on the formulas presented in this study using the mathematical formulation to determine the elements of the matrices M and C and the elements of G . The kinematics was described using Denavit–Hartenberg algorithm. Applying the rules of Denavit–Hartenberg (DH) notions presented in section 3.3.1.1, we can easily represent the manipulator robot and thus calculate all the corresponding transition matrices.

Transformation matrix for points in frame B_1 to the base frame G :

$$T_1^0 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & l_1 \cos(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) & 0 & l_1 \sin(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.51)$$

Transformation matrix for points in frame B_2 to the base frame B_1 :

$$T_2^1 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & l_2 \cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & l_2 \sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.52)$$

Transformation matrix for points in frame B_3 to the base frame B_2 :

$$T_3^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.53)$$

Using the relations $T_e = T_3^0 = T_1^0 T_2^1 T_3^2$, we can calculate the following matrices:

The overall transformation matrix between the end-effector frame and the base frame is:

$$T_e = T_3^0 = \begin{bmatrix} c_{12} & -s_{12} & 0 & l_1 c_1 + l_2 c_{12} \\ s_{12} & c_{12} & 0 & l_1 s_1 + l_2 s_{12} \\ 0 & 0 & 1 & -d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.54)$$

Using: $c_1 = \cos(\theta_1)$, $s_1 = \sin(\theta_1)$, $c_2 = \cos(\theta_2)$, $s_2 = \sin(\theta_2)$,

$c_{12} = \cos(\theta_1 + \theta_2)$, $s_{12} = \sin(\theta_1 + \theta_2)$

End-effector's (x, y, z) coordinates were placed in the reference frames as a function of joint parameters (forward kinematics):

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\ l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \\ -d \\ 1 \end{bmatrix} \quad (3.55)$$

Joint parameters as function of (x, y, z) were measured at the reference frame (inverse kinematics):

$$x^2 + y^2 = (l_2 \sin(\theta_2))^2 + ((l_1 + l_2 \cos(\theta_2)))^2 \quad (3.56)$$

$$\theta_2 = \pm \frac{x^2 + y^2 - [(l_2)^2 + (l_1)^2]}{2l_1 l_2} \quad (3.57)$$

$$\theta_1 = \tan^{-1}\left(\frac{y}{x}\right) - \tan^{-1}\left(\frac{l_2 \sin(\theta_2)}{(l_1 + l_2 \cos(\theta_2))}\right) \quad (3.58)$$

$$d = -Z \quad (3.59)$$

3.8.4 Properties of Inertia Matrix

Let us first find the inertia matrices. For simplification of the calculation and since there is absence of the parameters' values, we consider that the bodies of the arm are stems. So, all the products of inertia are null as well as the moments of inertia tensors of three links with respect to the axes (x, y, z) .

The inertia tensor matrix of the first link-1 is:

$$I_1 = \begin{bmatrix} I_{xx1} & I_{xy1} & I_{xz1} \\ I_{xy1} & I_{yy1} & I_{yz1} \\ I_{xz1} & I_{yz1} & I_{zz1} \end{bmatrix} \quad (3.60)$$

The inertia tensor matrix of the second link-2 is:

$$I_2 = \begin{bmatrix} I_{xx2} & I_{xy2} & I_{xz2} \\ I_{xy2} & I_{yy2} & I_{yz2} \\ I_{xz2} & I_{yz2} & I_{zz2} \end{bmatrix} \quad (3.61)$$

The inertia tensor matrix of the third link-3 is:

$$I_3 = \begin{bmatrix} I_{xx3} & I_{xy3} & I_{xz3} \\ I_{xy3} & I_{yy3} & I_{yz3} \\ I_{xz3} & I_{yz3} & I_{zz3} \end{bmatrix} \quad (3.62)$$

- Evaluation of the Dynamic Parameters: $M(\theta), C(\theta, \dot{\theta}), G(\theta)$

The elements of the inertia matrix $M(\theta)$ are:

$$M(\theta) = \begin{bmatrix} m_{11}(\theta) & m_{12}(\theta) & m_{13}(\theta) \\ m_{21}(\theta) & m_{22}(\theta) & m_{23}(\theta) \\ m_{31}(\theta) & m_{32}(\theta) & m_{33}(\theta) \end{bmatrix} \text{symmetric matrix} \quad (3.63)$$

Now we are ready to form the inertia matrix $M(\theta)$.

Here, m_{ij} where $(i = 1, 2, 3)$ and $(j = 1, 2, 3)$ are expressed, using equation (3.32):

$$m_{11}(\theta) = m_1 l_{c1}^2 + I_{zz1} + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(\theta_2)) + I_{zz2} + m_3 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos(\theta_2)) + I_{zz3} \quad (3.64)$$

$$m_{12}(\theta) = m_{21}(\theta) = m_2 (l_{c2}^2 + l_1 l_{c2} \cos(\theta_2)) + m_3 (l_{c2}^2 + l_1 l_{c2} \cos(\theta_2)) + I_{zz2} + I_{zz3} \quad (3.65)$$

$$m_{22}(\theta) = m_2 l_{c2}^2 + m_3 l_2^2 + I_{zz2} + I_{zz3} \quad (3.66)$$

$$m_{33}(\theta) = m_3 \quad (3.67)$$

$$m_{13}(\theta) = m_{31}(\theta) = m_{23}(\theta) = m_{32}(\theta) = 0 \quad (3.68)$$

3.8.5 The Matrix of Centrifugal and Coriolis Forces

We now determine the elements of the following matrix of centrifugal forces and Coriolis. They are as follows:

The Coriolis/centripetal matrix $C(\theta, \dot{\theta})$:

$$C_{ij,k} = \sum_i^n \frac{1}{2} \left\{ \frac{\partial m_{ij}(\theta)}{\partial \theta_k} + \frac{\partial m_{ki}(\theta)}{\partial \theta_j} - \frac{\partial m_{kj}(\theta)}{\partial \theta_i} \right\} \dot{\theta}_i, (i, j, k = 1, 2, \dots, n) \quad (3.69)$$

Therefore,

$$C(\theta, \dot{\theta}) = \begin{bmatrix} c_{11}(\theta, \dot{\theta}) & c_{12}(\theta, \dot{\theta}) & c_{13}(\theta, \dot{\theta}) \\ c_{21}(\theta, \dot{\theta}) & c_{22}(\theta, \dot{\theta}) & c_{23}(\theta, \dot{\theta}) \\ c_{31}(\theta, \dot{\theta}) & c_{32}(\theta, \dot{\theta}) & c_{33}(\theta, \dot{\theta}) \end{bmatrix} \quad (3.70)$$

Computing Christoffel symbols for using the formulas in equation (3.46), from where:

$$C_{11}(\theta, \dot{\theta}) = -l_1 \sin(\theta_2)(m_3 l_2 + m_2 l_{c2}) \dot{\theta}_2 \quad (3.71)$$

$$C_{12}(\theta, \dot{\theta}) = -l_1 \sin(\theta_2)(m_3 l_2 + m_2 l_{c2}) \dot{\theta}_1 \quad (3.72)$$

$$C_{21}(\theta, \dot{\theta}) = l_1 \sin(\theta_2)(m_3 l_2 + m_2 l_{c2}) \dot{\theta}_1 \quad (3.73)$$

$$C_{13}(\theta, \dot{\theta}) = C_{31}(\theta, \dot{\theta}) = C_{22}(\theta, \dot{\theta}) = C_{23}(\theta, \dot{\theta}) = C_{32}(\theta, \dot{\theta}) = C_{33}(\theta, \dot{\theta}) = 0 \quad (3.74)$$

For simplicity and more systematic dynamical model, we consider the next assumptions:

$$I_{zz1} = \frac{1}{12} m_1 l_1, I_{zz2} = \frac{1}{12} m_2 l_2, I_{zz3} = 0, l_{c1} = \frac{1}{2} l_1, l_{c2} = \frac{1}{2} l_2 \quad (3.75)$$

3.8.6 The Gravitational Torque Vector

To calculate the gravity, we must find the potential energy of the robot arm, as expressed by equations (3.35) and (3.37):

$$U_1 = m_1 g^T T_1^0 \bar{r}_1 = 0$$

$$U_2 = m_2 g^T T_2^0 \bar{r}_2 = 0 \quad (3.76)$$

$$U_3 = m_3 g^T T_3^0 \bar{r}_3 = -m_3 g$$

When:

$$g^T = [0 \ 0 \ -g]$$

$$\bar{r}_1 = \begin{bmatrix} \frac{1}{2} a_1 & 0 & 0 & 1 \end{bmatrix}$$

$$\bar{r}_2 = \left[\frac{1}{2}a_2 \ 0 \ 0 \ 1 \right] \quad (3.77)$$

$$\bar{r}_3 = \left[0 \ 0 \ \frac{1}{2}a_1 \ 1 \right]$$

Where:

$$U(\theta) = U_1(\theta) + U_2(\theta) + U_3(\theta) \quad (3.78)$$

$$U(\theta) = -m_3g \quad (3.79)$$

Using this expression of total potential energy $G_k = \frac{\partial U(\theta)}{\partial \theta_k}$, we can calculate the elements of the gravity vector with the following equations. The gravity matrix $G(\theta)$ is as follows:

$$G_1(\theta) = \frac{\partial U}{\partial \theta_1} = 0, \quad G_2(\theta) = \frac{\partial U}{\partial \theta_2} = 0, \quad G_3(\theta) = \frac{\partial U}{\partial \theta_3} = -m_3g \quad (3.80)$$

The function $G(\theta)$ has been defined in a next equation as:

$$G(\theta) = \frac{\partial U(\theta)}{\partial \theta} = \begin{bmatrix} G_1(\theta) \\ G_2(\theta) \\ G_3(\theta) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -m_3g \end{bmatrix} \quad (3.81)$$

$$G(\theta) = [0 \ 0 \ -m_3g]^T \quad (3.82)$$

We can write down the dynamic equations of the robot as:

$$\begin{aligned} & \left(\left(\frac{1}{3}m_1 + m_2 + m_3 \right) l_1^2 + \left(\frac{1}{3}m_2 + m_3 \right) l_2^2 + (m_2 + 2m_3)l_1l_2 \cos(\theta_2) \right) \ddot{\theta}_1 \\ & + \left(\left(\frac{1}{3}m_2 + m_3 \right) l_2^2 + \left(\frac{1}{2}m_2 + m_3 \right) l_1l_2 \cos(\theta_2) \right) \ddot{\theta}_2 - l_1l_2(m_2 + 2m_3) \sin(\theta_2) \dot{\theta}_1 + \\ & l_1l_2 \left(\frac{1}{2}m_2 + m_3 \right) (\sin(\theta_2) \dot{\theta}_2) = \tau_1 \end{aligned} \quad (3.83)$$

$$\begin{aligned} & \left(\left(\frac{1}{3}m_2 + m_3 \right) l_2^2 + \left(\frac{1}{2}m_2 + m_3 \right) l_1l_2 \cos(\theta_2) \right) \ddot{\theta}_1 + \left(\left(\frac{1}{3}m_2 + m_3 \right) l_2^2 \right) \ddot{\theta}_2 \\ & + l_1l_2 \left(\frac{1}{2}m_2 + m_3 \right) (\sin(\theta_2) \dot{\theta}_1) = \tau_2 \end{aligned} \quad (3.84)$$

$$m_3 \ddot{\theta}_3 - m_3g = \tau_3 \quad (3.85)$$

The final-dynamic model of the manipulator can be expressed in the matrix form as follows:

$$M(\theta) \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} + C(\theta, \dot{\theta}) \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} + G(\theta) = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \quad (3.86)$$

We can write the matrices in component form:

$$M(\theta) = \begin{bmatrix} m_{11} & m_{12} & 0 \\ m_{21} & m_{22} & 0 \\ 0 & 0 & m_{33} \end{bmatrix}, C(\theta, \dot{\theta}) = \begin{bmatrix} c_{11} & c_{12} & 0 \\ c_{21} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.87)$$

$$G(\theta) = [0 \ 0 \ -m_3g]^T, \tau_1 = [\tau_1 \ \tau_2 \ \tau_3]^T \quad (3.88)$$

Finally, we can write down the dynamic equations of three degrees-of-freedom (3-DOF) for SCARA robot. They are as follows:

$$\begin{bmatrix} \left(\left(\frac{1}{3}m_1 + m_2 + m_3 \right) l_1^2 + \left(\frac{1}{3}m_2 + m_3 \right) l_2^2 + (m_2 + 2m_3)l_1l_2 \cos(\theta_2) \right) & \left(\left(\frac{1}{3}m_2 + m_3 \right) l_2^2 + \left(\frac{1}{2}m_2 + m_3 \right) l_1l_2 \cos(\theta_2) \right) & 0 \\ \left(\left(\frac{1}{3}m_2 + m_3 \right) l_2^2 + \left(\frac{1}{2}m_2 + m_3 \right) l_1l_2 \cos(\theta_2) \right) & \left(\left(\frac{1}{3}m_2 + m_3 \right) l_2^2 \right) & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} + \begin{bmatrix} -l_1l_2(m_2 + 2m_3) \sin(\theta_2) & l_1l_2\left(\frac{1}{2}m_2 + 2m_3\right) \sin(\theta_2) & 0 \\ l_1l_2\left(\frac{1}{2}m_2 + m_3\right) \sin(\theta_2) & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -m_3g \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \quad (3.89)$$

CHAPTER 4

MINIMUM-TIME PATH PLANNING FOR ROBOT MANIPULATORS

USING PATH PARAMETER OPTIMIZATION WITH

EXTERNAL FORCE AND FRICTIONS

4.1 Introduction

During recent years, robotics has become a significant topic of scientific research because of its application in many fields and in many research areas. It is one of the technologies that have the potential to shape/reshape the future of mankind. In many engineering applications, parametric trajectory is tracked when the robot moves through the desired path at a high velocity. In the path planning method, the path smoothens, which is very significant because if it is not smooth, a robot might make and unusual tilt, crash or make dangerous movements. The path planning algorithm of manipulative robots has become a very important and vast field of research. During the last few decades, several laws have been established. For an arm of a manipulator robot to reach a desired position regulation or follow a pre-defined trajectory continuation, so obviously, the motion planning must have certain properties robustness, speed, convergence or stability.

In this research, the target is how to solve minimum-time path planning problem for the robot manipulator and finding the trajectory planning solution for the path using path parameter. In this chapter, we have presented the main algorithm for trajectory planning, and focused on minimum-time path planning method for robot manipulators using path parameter optimization algorithm with external force and frictions.

4.2 Dynamic Modelling of the Robot Manipulator with Friction

In this section, we have supposed that the manipulator's joint angles are represented by a vector q , so, the joint angle velocity will be \dot{q} , while \ddot{q} will be the angular acceleration. The dynamic modelling equation of motion of the robot manipulator can be written as follows [89]:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = T \quad (4.1)$$

In this case, the total torque T will be as follows:

$$T = T_{ext} + T_c + T_f \quad (4.2)$$

Here, $M(q) \in R^{n \times n}$ is the inertia matrix and $C(q, \dot{q}) \in R^{n \times n}$ is a matrix that contains the information of centrifugal and Coriolis torques. Here, C is not a unique matrix but $C(q, \dot{q})\dot{q}$ is a unique vector. $G(q) \in R^{n \times n}$ is the gravity torque, T is the summation of total torques, and n is a number of joints' angles. T_{ext} , T_c , and T_f are the external force, control and friction torques, respectively. The external torque T_{ext} has been expressed in following equation [89]:

$$T_{ext} = J^T F_{ext} \quad (4.3)$$

Here, F_{ext} is the external force at the robot's end-effector, and J^T is the transpose of the Jacobian matrix.

4.3 Friction Modelling

We consider that the external force is generated by friction between the end-effector and task plane, and its direction is along the opposite direction of the end-effector's velocity. We consider a model as follows:

$$F_{ext} = -\mu v f_n \hat{n}_v \quad (4.4)$$

In the mentioned expression, v is the norm of the velocity vector of the end-effector, f_n is a normal force which is perpendicular to the surface that has velocity vector and the end-effector's link, μ is the friction coefficient, and \hat{n}_v is a unit vector along the velocity tangent. The joint angle friction torque has been modeled [89]:

$$T_f = K_c \text{sign}(\dot{q}) + K_v \dot{q} \quad (4.5)$$

Here, K_c is a diagonal matrix that consists of the Coulomb's friction coefficients for any joint, and K_v is another diagonal matrix having the viscous friction coefficients of the joint angles.

4.4 Actuator Dynamics

This section studies and analyzes the dynamic behaviour of the permanent-magnet DC motor system. Many researchers consider \dot{q} and \ddot{q} to have constraints, but when the actuator is able to generate more torque, it will be able to generate higher jerk and

acceleration, so the limitations on the time derivatives of the joint angles come from the actuators.

Consequently, the added limitations on the time derivatives of the joint angles will cause more limitation for the robot, so we will not use the actual potential of the robot. Here, we will consider the transmission of rotation between the actuators and the arm of the robot as being guaranteed by the mechanical transmission system of the gears. Although this mechanism reduces the angular velocity of the motor, it increases the generated torque of the motor [89]:

$$T_c = NT_m$$

$$N = \text{diag}([N_1, \dots, N_n]) \quad (4.6)$$

$$\dot{q} = N^{-1}\dot{q}_m$$

Here, N is the diagonal matrix of the transmission gear system, T_m the vector of motor's torque, \dot{q}_m is the speed of the motor.

4.5 DC Motor System Modelling

This section analyzes the dynamic behavior of the permanent-magnet DC motor, which is the driving force of the robotic manipulator. For studying it, first a mathematical model has been developed for the DC motor. The joint angle motors are generally DC motors. The use of Kirchhoff's voltage law for armature windings has been shown in Figure 4.1. The DC motor system equations are as follows [90]:

$$\dot{I} = L^{-1}(-RI - K_{bemf}\dot{q}_m + U) \quad (4.7)$$

$$T_m = K_m I$$

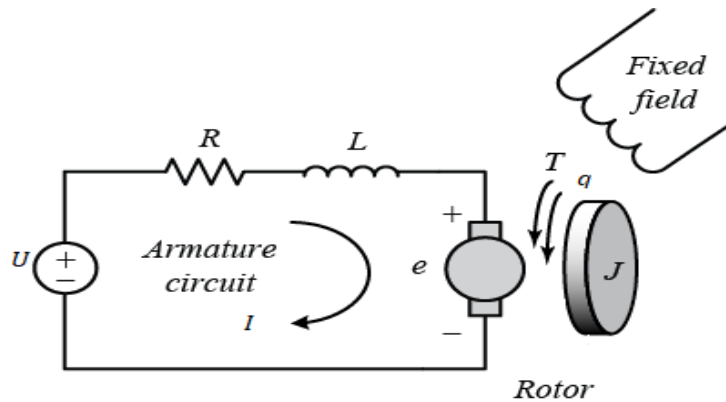


Figure 4.1: DC motor system equivalent circuit.

Here, $L = \text{diag}([L_1, L_2, \dots, L_n])$ is a diagonal matrix that contains the motor inductance elements and $R = \text{diag}([R_1, R_2, \dots, R_n])$ is a matrix that contains the armature resistances, $K_{bemf} = \text{diag}([K_{bemf_1}, K_{bemf_2}, \dots, K_{bemf_n}])$ shows the back electromotive force constant matrix, $U = (U_1, U_2, \dots, U_n)$ is the input voltage vector, $I = (I_1, I_2, \dots, I_n)$ is the armature current of each DC motor system, and $K_m = \text{diag}([K_1, K_2, \dots, K_n])$ is the motor torque constant matrix.

By substituting equation (4.7) into the equation (4.6), we get:

$$\dot{T}_c = AT_c + B\dot{q} + DU \quad (4.8)$$

$$A = \text{diag} \left(\left[-\frac{R_1}{L_1}, -\frac{R_2}{L_2}, \dots, -\frac{R_n}{L_n} \right] \right)$$

$$B = \text{diag} \left(\left[-\frac{K_{m_1}K_{bemf_1}N_1^2}{L_1}, -\frac{K_{m_2}K_{bemf_2}N_2^2}{L_2}, \dots, -\frac{K_{m_n}K_{bemf_n}N_n^2}{L_n} \right] \right) \quad (4.9)$$

$$D = \text{diag} \left(\left[\frac{K_{m_1}N_1}{L_1}, \frac{K_{m_2}N_2}{L_2}, \dots, \frac{K_{m_n}N_n}{L_n} \right] \right)$$

As a result, the augmented equations of the motion of the robot manipulator will be as follows:

$$T_c = M\ddot{q} + C\dot{q} + G(q) - T_{ext} - T_f \quad (4.10)$$

By calculating the time derivative of equation (4.10), we have:

$$\dot{T}_c = \dot{M}\ddot{q} + M\ddot{q} + \dot{C}\dot{q} + C\ddot{q} + \frac{\partial G}{\partial q}\dot{q} - \dot{T}_{ext} - \dot{T}_f \quad (4.11)$$

Equation (4.8) has been rearranged as follows:

$$U = D^{-1}(\dot{T}_c - AT_c - B\dot{q}) \quad (4.12)$$

Here, \dot{T}_{ext} and \dot{T}_f are calculated using equation (4.3), thus:

$$\dot{T}_{ext} = j^T F_{ext} + J^T \dot{F}_{ext} \quad (4.13)$$

When the normal force is constant, by calculating the time derivative of F_{ext} using equation (4.4), we obtain the following:

$$\dot{F}_{ext} = -\mu\dot{v}f_n\hat{n}_v - \mu v f_n \dot{\hat{n}}_v \quad (4.14)$$

Here, \hat{n}_v and $\dot{\hat{n}}_v$ are the unit vectors of velocity and its time derivative, respectively.

In conclusion, when the desired joint angles and their first, second and third-time

derivatives were known, in order to calculate U , we needed to calculate T_c using the desired path, after which, equation (4.11) was used to calculate the \dot{T}_c . Finally, by using equation (4.12), U was calculated.

4.6 Constrained Dynamic System of the Robot Manipulator

This section introduces and discusses benefits of constraints. There are a number of constraints, which emerge because of the limitations of the robot. These constraints can be divided into two-groups, out of which, the first-group is related to kinematic constraints such as constraints in the joint angles, angular velocity, angular acceleration, jerk and higher time derivatives of the joint angles. We can consider limitations for the higher time derivatives of the joint angles. The second-group is related to the constraints on the actuators. The equations for the constrained dynamic system of the motors are as follows [90]:

$$|\dot{q}_{m_i}| \leq \bar{q}_{m_i} \quad (4.15.a)$$

$$|I_i| \leq \bar{I}_i \quad (4.15.b)$$

$$|U_i| \leq \bar{U}_i \quad (4.15.c)$$

$$|\dot{I}_i| \leq \bar{\dot{I}}_i \quad (4.15.d)$$

$$\sqrt{\frac{1}{t_f} \int_0^{t_f} I_i^2(t) dt} \leq \bar{I}_{c_i}, \text{ for } (i = 1, 2, \dots, n) \quad (4.15.e)$$

Here, n is the number of joint angles, and \bar{q}_{m_i} , \bar{I}_i , \bar{U}_i , $\bar{\dot{I}}_i$, \bar{I}_{c_i} represent the maximum admissible motor speed, current, feeding voltage, the time derivative of current and braked motor current, respectively. Finally, we converted these constraints into robot manipulator constraints as follows:

$$|\dot{q}_i| \leq N^{-1} \bar{q}_{m_i} \quad (4.16.a)$$

$$|T_{c_i}| \leq NK_m \bar{I}_i \quad (4.16.b)$$

$$|\dot{T}_{c_i}| \leq NK_m \bar{\dot{I}}_i \quad (4.16.c)$$

$$\sqrt{\frac{1}{t_f} \int_0^{t_f} T_{c_i}^2(t) dt} \leq N^2 K_m^2 \bar{I}_{c_i}^2, \text{ for } (i = 1, 2, \dots, n) \quad (4.16.d)$$

Where:

$$\dot{q}_i, T_{ci}, \dot{T}_{ci}, \text{ and } \sqrt{\frac{1}{t_f} \int_0^{t_f} T_{ci}^2(t) dt}$$

The four above mentioned variables represent the angular velocity of joint angles, the control torque, the time derivative of the torque, and guaranteed term for harmless overtaking of the permanent operating range [91].

4.7 Definition of the Parametric Trajectory Optimization Problem

In robotic applications, we need to track a desired trajectory in the Cartesian-space. There are two ways of finding a trajectory that must be considered as a part of path planning solution.

The first one is called as a trajectory of robot manipulators, which are defined by the way point method to solve the path planning problem using the optimization method, and it does not have analytical formulation. Some researchers suggested a second approach, which is a technique related to a path parameter, and it is defined by the independent parameter. In this case, the path parameter is parameterized by using a time-independent scalar-path parameter.

In this study, when the path parameter has discontinuities in k^{th} time derivatives, then the path and all the dynamics will have discontinuities at the k^{th} and the higher k^{th} derivative, for example, when the second-time derivative of path parameter had a discontinuity between the intermediate-points of the linear acceleration, angular acceleration, jerk (a derivative of acceleration), and other time derivatives will also have discontinuities. Sometimes, this path is analytically formulated and parametrized with an independent parameter, which is a function of time. This independent parameter approach given in this chapter is also called as path parameter algorithm. In the following discussion, there is a free parameter for the path that has to be defined as a function of time, when the free parameter is known as a function of time and some parameters of a manipulator like time history of joint angles, angular velocities, or torque of the joint are known.

We are interested in performing a task with a robot keeping in view the minimum-time path planning problem. Thus, the cost function will be as follows:

4.7.1 The Cost Function

In this case, the cost function will be:

$$\text{Minimization of the cost function } J = \int_0^{t_f} dt = t_f \quad (4.17.a)$$

Where, J is the cost function in the objective function, and t_f is the final-time.

4.7.2 Subject to the Constraints

$$\dot{T}_c = \dot{M}\ddot{q} + M\ddot{q} + \dot{C}\dot{q} + C\ddot{q} + \frac{\partial G}{\partial q}\dot{q} - \dot{T}_{ext} - \dot{T}_f$$

$$U = D^{-1}(\dot{T}_c - AT_c - B\dot{q})$$

$$|\dot{q}_i| \leq N^{-1}\bar{q}_{m_i} \quad (4.17.b)$$

$$|T_{c_i}| \leq NK_m\bar{I}_i$$

$$|\dot{T}_{c_i}| \leq NK_m\bar{I}_i$$

$$\sqrt{\frac{1}{t_f} \int_0^{t_f} T_{c_i}^2(t) dt} \leq N^2 K_m^2, \text{ for } (i = 1, 2, \dots, n)$$

Where, M and C are functions of q and \dot{q} , respectively; therefore, the time derivative of these matrices is as follows:

$$\dot{M} = \sum_{i=1}^n \frac{\partial M}{\partial q_i} \dot{q}_i, \dot{C} = \sum_{i=1}^n \frac{\partial C}{\partial q_i} \dot{q}_i + \sum_{i=1}^n \frac{\partial C}{\partial \dot{q}_i} \ddot{q}_i \quad (4.18)$$

4.8 Dynamic Parameter of the Robotic Manipulator

Suppose the purpose of path planning is to track a desired path in the Cartesian-space that is parameterized by γ . The position of end-effector r is a function of the path.

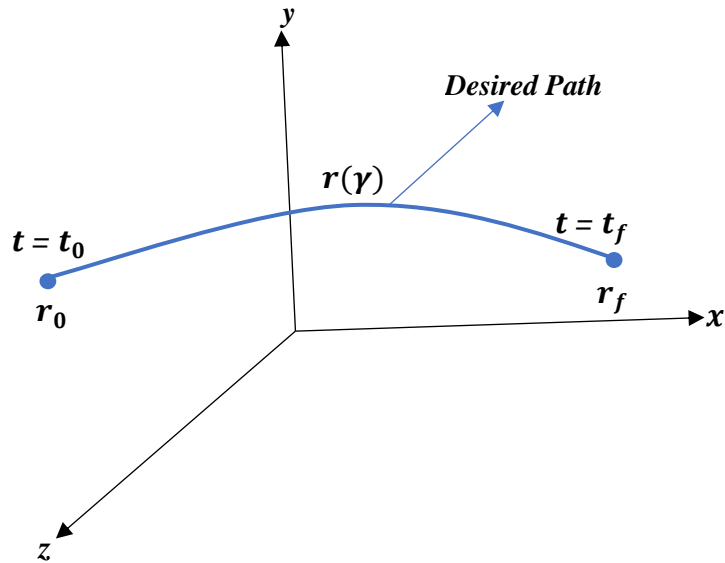


Figure 4.2: Trajectory planning along a parameterized path.

The sample trajectory has been shown in Figure 4.2, the coordinates of the start and stop-points are r_0 and r_f . When the robot is at the starting-point, for simplicity, γ can be represented by $\gamma(t_0) = 0$, and at the stopping-point, it is $\gamma(t_f) = \gamma_f$. It is necessary to specify the initial-position and the final-position of the end-effector.

4.8.1 Objective Function

The aim of this optimization algorithm technique is to find $\gamma(t)$ to minimize the cost function equation (4.17.a) by considering the constraints equation (4.17.b). In this case, polynomial approach is taken to convert the function optimization into a parameter optimization problem; therefore, the approach will provide a sub-optimal solution; however, $\gamma(t)$ has a number of constraints because of the manipulator limitations. First, we know γ has two constraints based on the definition of the path at the first and the end-times as follows:

$$\begin{aligned}\gamma(t_0) &= \gamma(0) = 0 \\ \gamma(t_f) &= \gamma_f\end{aligned}\tag{4.19}$$

Where, $\gamma(t)$ is a function of time and the unknowns (x, y, z) are components of position vector r for minimum-time path planning. We have to find the shape of function $\gamma(t)$ such that the robot manipulator is able to start from point r_0 and move through the desired path towards the r_f in the minimum-time. This desired path is fixed, and it will not change with time.

When the robot wants to move through the desired path with high velocity, the actuators might be unable to generate the desired torques, so there are some constraints that we have to take into account.

To find other constraints, we define the relation between the kinematic constraints in the Cartesian-space and the joint-space; therefore, we start by finding the relation between the velocity, acceleration, and jerk as functions of time in the Cartesian-space, thus:

$$\begin{aligned}\dot{r} &= v = r'\dot{\gamma} \\ \ddot{r} &= a = r''\dot{\gamma}^2 + r'\ddot{\gamma} \\ \dddot{r} &= J = r'''\dot{\gamma}^2 + 2r''\ddot{\gamma} + r'\dddot{\gamma}\end{aligned}\tag{4.20}$$

Here, v, a , and J represent velocity, acceleration, and jerk of the end-effector, respectively. The relation between kinematic parameters in the Cartesian-space and the joint space are given as follows:

$$\begin{aligned} v &= J\dot{q} \\ a &= \dot{J}\dot{q} + J\ddot{q} \\ J &= \ddot{J}\dot{q} + 2\dot{J}\ddot{q} + J\ddot{q} \end{aligned} \tag{4.21}$$

Here, J, \dot{J} , and \ddot{J} are the Jacobian, first and second-time derivatives of the Jacobian matrix, respectively. It is desirable for the velocity, acceleration and jerk to be zero at the initial-point.

4.8.2 The Conditions

Thus, by using equation (4.20); $\dot{q}, \ddot{q}, \ddot{q}$, will be zero. These conditions are simple in the view of the actuators for starting because the commands of the actuator will not jump to the maximum.

4.8.3 Initial and Final Conditions

Hence, if at the starting-point, the velocity, acceleration and jerk are zero in the Cartesian-space, the time derivatives of the path parameter $\dot{\gamma}(0), \ddot{\gamma}(0)$ and $\ddot{\gamma}(0)$ will be zero; therefore, the initial-conditions for $\gamma(t)$ will be:

$$\begin{aligned} \gamma(0) &= 0 \\ \dot{\gamma}(0) &= 0 \\ \ddot{\gamma}(0) &= 0 \\ \ddot{\gamma}(0) &= 0 \end{aligned} \tag{4.22}$$

It is necessary to define the final-conditions for the path parameter. We know that γ_f is known by the definition of the path, so again, in order to have zero velocity, zero acceleration, and zero jerk in the Cartesian-space and the joint spaces, the final-condition for $\gamma(t)$ will be:

$$\begin{aligned} \gamma(t_f) &= \gamma_f \\ \dot{\gamma}(t_f) &= 0 \\ \ddot{\gamma}(t_f) &= 0 \\ \ddot{\gamma}(t_f) &= 0 \end{aligned} \tag{4.23}$$

For a sub-optimal solution, suppose the function $\gamma(t)$ is approximated by time-series as follows:

$$\gamma(t) = \sum_{i=4}^n a_i t^i \quad (4.24)$$

In the above format, all the initial-conditions were satisfied but the final-conditions were not. For maintaining simplicity to meet the final-conditions, we will consider a model for $\gamma(t)$ as follows:

$$\gamma(t) = A\gamma_1(t) + B\gamma_2(t) + C\gamma_3(t) + D\gamma_4(t) \quad (4.25)$$

Here, $A, B, C,$ and D are constant and unknown parameters, and $\gamma_1, \gamma_2, \gamma_3$ and γ_4 have a format as given in equation (4.25); therefore, they will satisfy the initial-conditions:

$$\begin{aligned} \gamma_1(t) &= \sum_{i=4}^n a_i t^i \\ \gamma_2(t) &= \sum_{i=4}^n b_i t^i \\ \gamma_3(t) &= \sum_{i=4}^n c_i t^i \\ \gamma_4(t) &= \sum_{i=4}^n d_i t^i \end{aligned} \quad (4.26)$$

When, $a_i, b_i, c_i,$ and $d_i, (i = 4, \dots, n),$ and t_f are known, we can calculate $A, B, C,$ and D as follows:

$$\begin{aligned} A\gamma_1(t_f) + B\gamma_2(t_f) + C\gamma_3(t_f) + D\gamma_4(t_f) &= \gamma_f \\ A\dot{\gamma}_1(t_f) + B\dot{\gamma}_2(t_f) + C\dot{\gamma}_3(t_f) + D\dot{\gamma}_4(t_f) &= 0 \\ A\ddot{\gamma}_1(t_f) + B\ddot{\gamma}_2(t_f) + C\ddot{\gamma}_3(t_f) + D\ddot{\gamma}_4(t_f) &= 0 \\ A\ddot{\gamma}_1(t_f) + B\ddot{\gamma}_2(t_f) + C\ddot{\gamma}_3(t_f) + D\ddot{\gamma}_4(t_f) &= 0 \end{aligned} \quad (4.27)$$

In equation (4.27), there are four unknown parameters $A, B, C,$ and D when $\gamma_1, \gamma_2, \gamma_3$ and γ_4 are known at the final-time. For the satisfaction of the final-conditions, we can write:

$$\begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} \gamma_1(t_f) & \gamma_2(t_f) & \gamma_3(t_f) & \gamma_4(t_f) \\ \dot{\gamma}_1(t_f) & \dot{\gamma}_2(t_f) & \dot{\gamma}_3(t_f) & \dot{\gamma}_4(t_f) \\ \ddot{\gamma}_1(t_f) & \ddot{\gamma}_2(t_f) & \ddot{\gamma}_3(t_f) & \ddot{\gamma}_4(t_f) \\ \ddot{\gamma}_1(t_f) & \ddot{\gamma}_2(t_f) & \ddot{\gamma}_3(t_f) & \ddot{\gamma}_4(t_f) \end{bmatrix}^{-1} \begin{bmatrix} \gamma_f \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.28)$$

Hence, when A, B, C , and D are selected from equation (4.28), the initial and final-conditions will always be satisfied. Consequently, $\gamma(t)$ will be the function of the unknown vector as given below:

$$\gamma(t) = \gamma(x, t)$$

$$x = [a_4, \dots, a_n, b_4, \dots, b_n, c_4, \dots, c_n, d_4, \dots, d_n, t_f]^T \quad (4.29)$$

Where, x is an unknown vector that is calculated by minimizing the cost function through equation (4.17.a), which is subject to the dynamic constraints of the robot equation (4.17.b). Therefore, the function of this optimization problem is to minimize the equation (4.17.a); so, the constraints equation (4.17.b) will be as follows:

- Cost Function:

$$\text{Minimization of the cost function } J(x) = [0_{(1 \times 4)(n-4)} \ 1]x \quad (4.30)$$

- The Constraint Equations:

$$T_c = M\ddot{q} + C\dot{q} + G(q) - T_{ext} - T_f$$

$$\dot{T}_c = \dot{M}\ddot{q} + M\ddot{\ddot{q}} + \dot{C}\dot{q} + C\ddot{q} + \frac{\partial G}{\partial q}\dot{q} - \dot{T}_{ext} - \dot{T}_f$$

$$U = D^{-1}(\dot{T}_c - AT_c - B\dot{q})$$

$$|\dot{q}_i| \leq N^{-1}\bar{q}_{m_i} \quad (4.31)$$

$$|T_{c_i}| \leq NK_m\bar{I}_i$$

$$|\dot{T}_{c_i}| \leq NK_m\bar{I}_i$$

$$\sqrt{\frac{1}{t_f} \int_0^{t_f} T_{c_i}^2(t) dt} \leq N^2 K_m^2 \bar{I}_{c_i}^2, \text{ for } (i = 1, 2, \dots, n)$$

4.9 Managing the Constraints of the Optimization Problem of Robot Path Planning

The purpose behind the algorithm of the optimization problem is to minimize the final-time t_f subjected to the dynamics of the robot manipulator. As mentioned previously, unknown parameters are collected in the vector x ; therefore, the objective function can be written as follows:

$$\text{Minimize } J(x) = x_{\bar{N}+1} = t_f \quad (4.32)$$

Where, \hat{N} is the number of unknown parameters for modelling $\gamma(t)$; therefore, $4(n - 4) = \hat{N}$; however, many constraints exist due to the limitations pertaining to angular velocity, motor input voltages, and the torques of the motors. These constraints may appear at different times.

We can manage the constraints of the problem at any time by adding the previous constraints. A simple method of doing this is to divide the time between the initial and final-points with known and constant m incremental-times. The constraints listed in Table 4.1 have two types. The first type consists of differential equations and the second type consists of non-linear or linear equations of inequality.

In this study, we suggest rewriting every constraint without solving the differential equations. We know that some constraints may appear at all times, so we can manage the constraints of the problem at any time by adding them to the previous constraints. A simple algorithm has been used to generate new constraints for any time-interval; therefore, when the vectors x and t_f are known, we can divide the t_f by m incremental-times as follows:

$$\Delta t = \frac{t_f}{m}, t_k = \Delta t k, (k = 0, \dots, m) \quad (4.33)$$

To prepare the constraints as a function of the unknown vector x , first the initial-time x_0 is generated as a random vector. This random vector will guarantee the first and end-conditions of the path parameter, but it is very important for it to adjust it for other constraints as well. Table 4.1 shows the processes of generating of constraints. (See appendix A).

In Table 4.1, \bar{a} and \underline{a} are respectively used to represent the maximum and minimum of parameter a . Therefore, the size of the constraints at each step will increase. There are ten constraints in each t_k , so the size of the constraint vector at the end will be $10(m + 1)$. In equation (4.31), there is an integral equation for any motor.

This integral can be written as follows:

$$\Delta t \sum_{i=1}^N T_{c_i}^2 = N \Delta t \bar{T}_{c_i}^2 \quad (4.34)$$

Thus, by the end of every iteration, another constraint is added to the previous constraints.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 Introduction

This chapter discusses the simulations and their results pertaining to the robotic manipulator. For simulation, MATLAB has been used for the proposed algorithm, which we applied to the manipulator test and trials. We used dynamic robot manipulator model for synthesizing the motion.

Moreover, we have presented theoretical results, which were obtained through the mentioned simulation program in order to understand and analyze the overall impact. We used MATLAB because now it is a universally acceptable process for simulation.

5.2 Robot Simulation Dynamic Model

The SCARA robot and its kinematic and dynamic models have been used in a previous simulation study as well [91] but in the current thesis, we have considered some significant additional parameters and considered even small changes to obtain accurate simulations.

The robot manipulator model has been shown in Figure 5.1, which we have used for experiments. The mathematical form of simulation is given below:

$$\begin{aligned}
 & \begin{bmatrix} (3.78 + 0.272\cos q_2 + 0.022\sin q_2) + (0.08 + 0.136\cos q_2 + 0.011\sin q_2) & 0 \\ (0.08 + 0.136\cos q_2 + 0.011\sin q_2) & 0.08 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \\
 & \begin{bmatrix} (0.011\cos q_2 - 0.136\sin q_2) + (\dot{q}_2 + \dot{q}_1) + 0.07 & (0.011\cos q_2 - 0.136\sin q_2)\dot{q}_1 \\ (0.011\cos q_2 - 0.136\sin q_2)\dot{q}_1 & 0.013 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \\
 & \begin{bmatrix} T_{c1} \\ T_{c2} \end{bmatrix} + \begin{bmatrix} (0.62\text{sign}\dot{q}_1) \\ (0.17\text{sign}\dot{q}_2) \end{bmatrix} + \begin{bmatrix} K_{v1} & 0 \\ 0 & K_{v2} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} T_{ext1} \\ T_{ext2} \end{bmatrix} \quad (5.1)
 \end{aligned}$$

For two-link or (2-DOF) robot manipulator, the kinematics equations are based on the end-effector's position, and they are a function of joint angles and the robot manipulator's length. It can be expressed in matrix form as follows:

$$r = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} l_1 \cos q_1 + l_2 \cos(q_1 + q_2) \\ l_1 \sin q_1 + l_2 \sin(q_1 + q_2) \end{bmatrix} \quad (5.2)$$

Now applying the given direct kinematic, we can express the inverse kinematic equations in the following form:

$$q_1 = \tan^{-1} \left(\frac{y}{x} \right) - \tan^{-1} \left(\frac{K_2}{K_1} \right) \quad (5.3)$$

$$q_2 = \tan^{-1} \left(\frac{\sin(q_2)}{\cos(q_2)} \right) \quad (5.4)$$

In this case, the formulas for K_1 , K_2 , $\sin(q_2)$ and $\cos(q_2)$ are as follows:

$$K_1 = l_1 + l_2 \cos(q_2), \quad K_2 = l_2 \sin(q_2) \quad (5.5)$$

$$\sin q_2 = \pm \sqrt{1 - \cos q_2^2} \quad (5.6)$$

$$\cos q_2 = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2L_1 L_2} \quad (5.7)$$



Figure 5.1: Photo of the IRCCyN SCARA robot.

The robot manipulator parameters have been mentioned in Table 5.1. (See appendix A).

Here, the normal force is $f_n = 0.1N$. We have shown the actuators' electro-mechanical constraints in Table 5.2. (See appendix A).

5.3 Discussion on Simulation of Results

This section shows that the simulations were performed for plotting Figure 5.1 with the help of a pencil, which is attached to the end-effector. By doing this, we will draw the desired path while the end-effector must track the given path. In this case, the Cartesian path is a function of the path parameter, which can be expressed in matrix form as given below:

$$r = \begin{bmatrix} \rho \sin(\gamma) \\ 0.5\rho \cos(\gamma) \end{bmatrix} \quad (5.8)$$

If we take the derivatives of r based on path parameter γ , the derivatives will be as follows:

$$r' = \begin{bmatrix} \rho' \sin(\gamma) + \rho \cos(\gamma) \\ 0.5\rho' \cos(\gamma) - 0.5\rho \sin(\gamma) \end{bmatrix} \quad (5.9)$$

$$r'' = \begin{bmatrix} \rho'' \sin(\gamma) + 2\rho' \cos(\gamma) - \rho \sin(\gamma) \\ 0.5\rho'' \cos(\gamma) - \rho' \sin(\gamma) + 0.5\rho \cos(\gamma) \end{bmatrix} \quad (5.10)$$

$$r''' = \begin{bmatrix} \rho''' \sin(\gamma) + 3\rho'' \cos(\gamma) - 3\rho' \sin(\gamma) + \rho \cos(\gamma) \\ 0.5\rho''' \cos(\gamma) - 1.5\rho'' \sin(\gamma) - 0.5\rho' \cos(\gamma) - \rho \sin(\gamma) \end{bmatrix} \quad (5.11)$$

Here, ρ , ρ' , ρ'' and ρ''' can be mathematically expressed as follows:

$$\rho = 0.4 - 0.1\gamma \cos(\gamma), \quad (0 \leq \gamma \leq 200^\circ) \quad (5.12)$$

$$\rho' = -0.1 \cos(\gamma) + 0.1\gamma \sin(\gamma) \quad (5.13)$$

$$\rho'' = 0.2 \sin(\gamma) + 0.1\gamma \cos(\gamma) \quad (5.14)$$

$$\rho''' = -0.4 \sin(\gamma) - 0.1\gamma \cos(\gamma) \quad (5.15)$$

We have presented the desired path in Figure 5.2 with clearly marked starting and stopping-points, which are as follows:

$$r_0 = \begin{bmatrix} 0 \\ 0.2 \end{bmatrix} m$$

$$r_f = \begin{bmatrix} -0.1965 \\ -0.2699 \end{bmatrix} m \quad (5.16)$$

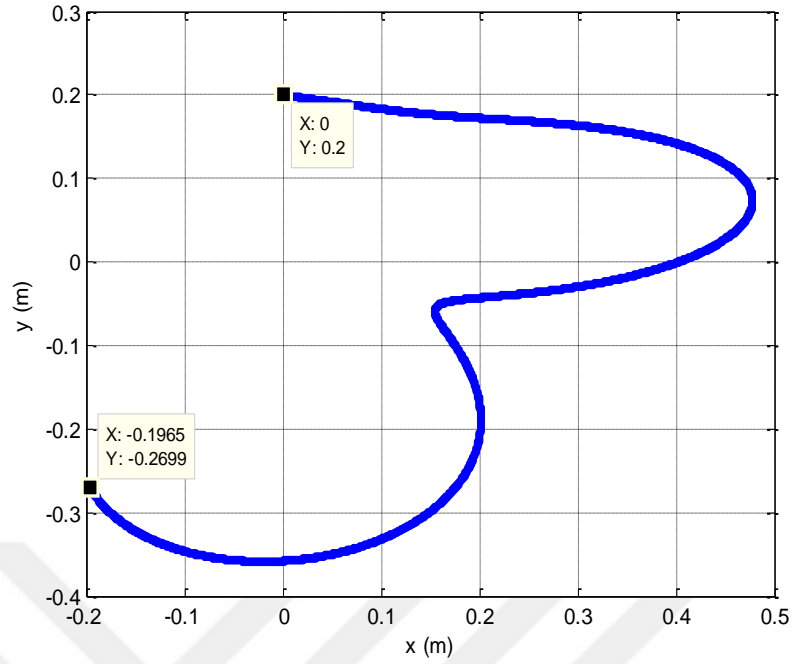


Figure 5.2: Desired path in a 2-D Cartesian-space. The blue curve shows the motion along the x -axis and y -axis, respectively.

The outcomes of the simulations and the desired path from the starting-point to the end-point in 2-D Cartesian-space have been illustrated in Figure 5.2. For approximating γ as a function of time, the value of n in equation (4.24) was taken as $n = 10$. The Taylor expansion contains terms up to t^{10} . Thus $a_i = b_i = c_i = d_i = 0$ while ($i = 1, \dots, 4$).

Now we'll find the number of unknown parameters. Since γ is (4×6); we added t_f to it so it will be ($4 \times 6 + 1 = 25$) unknown parameters; therefore, we used normal distribution for generating the path parameter coefficients. Here, the final-time for the movement was initially estimated as (17) seconds.

The time ranges between (0) to (17) seconds, which has been further divided into (5000) points. They are shown in Table 5.1 (See appendix A). Hence, the total number of constraints is (50,000). In case of iteration, the number of constraints will increase if the time is increasing. When the optimization algorithm was applied, the final-time was (14.4296) seconds, and the results were found, which are given below. The path parameter, its first, second, and third-time derivatives have been shown in Figures 5.3 and 5.4, which are showing simulation results.

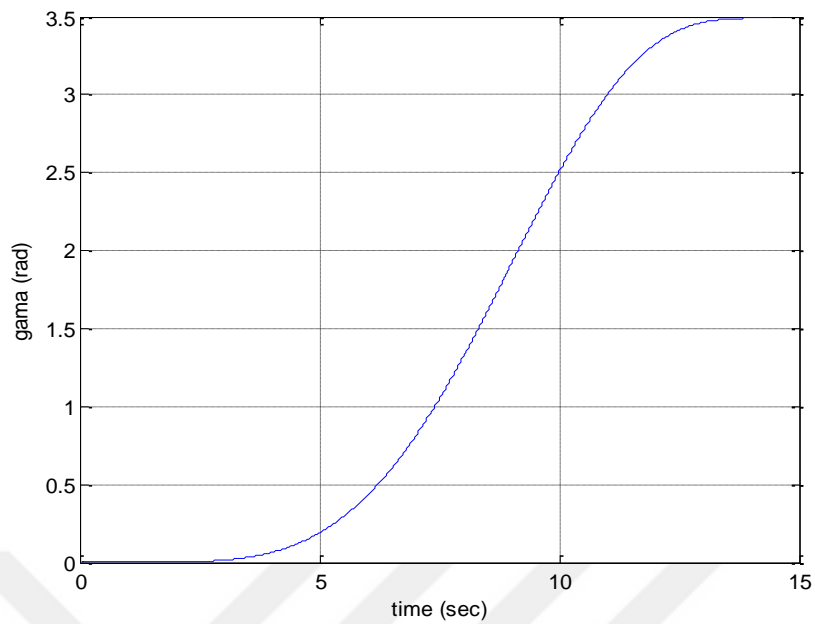


Figure 5.3: The corresponding optimal path parameter for moving end-effector.

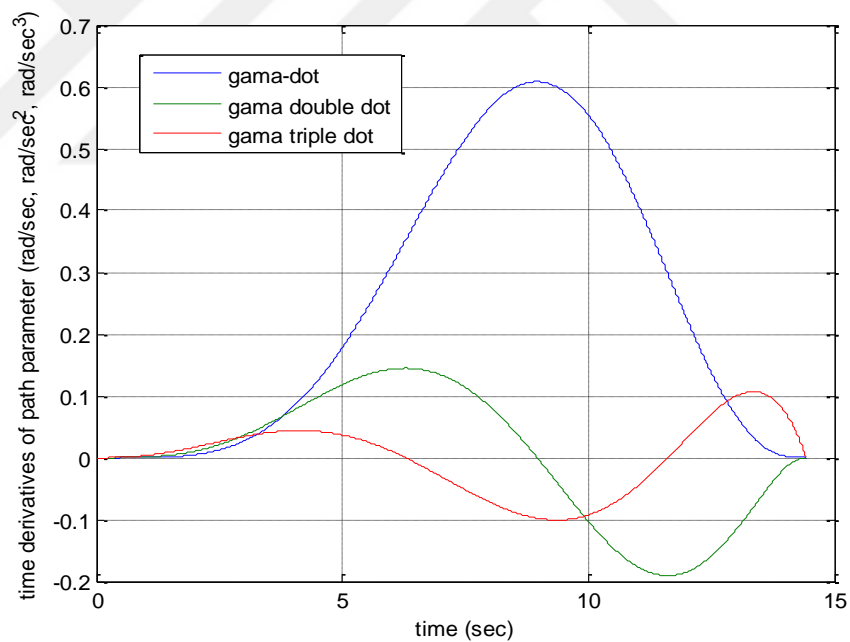


Figure 5.4: The time derivatives of the path parameter with zero starting and ending have been shown in green, blue, and red curves respectively.

Now we know the end-point trajectory. The results shown in Figure 5.3 demonstrate that the path parameter starts at zero and ends at ($200 \text{ deg} = 3.4907 \text{ rad}$). The test results show that the path parameters result in smooth graph formation while Figure 5.4 depicts the first three-time derivatives of the path parameter. These parameters have zero values both at the starting and ending-points. In addition, the

first-time derivative assumes positive value. We discovered that γ is an increasing function; therefore, the end-effector always goes towards the final-point when the time increases, and it does not move in the backward direction. Some of the other derivatives were positive while some of them were negative because the system changes the acceleration along the Cartesian reference path.

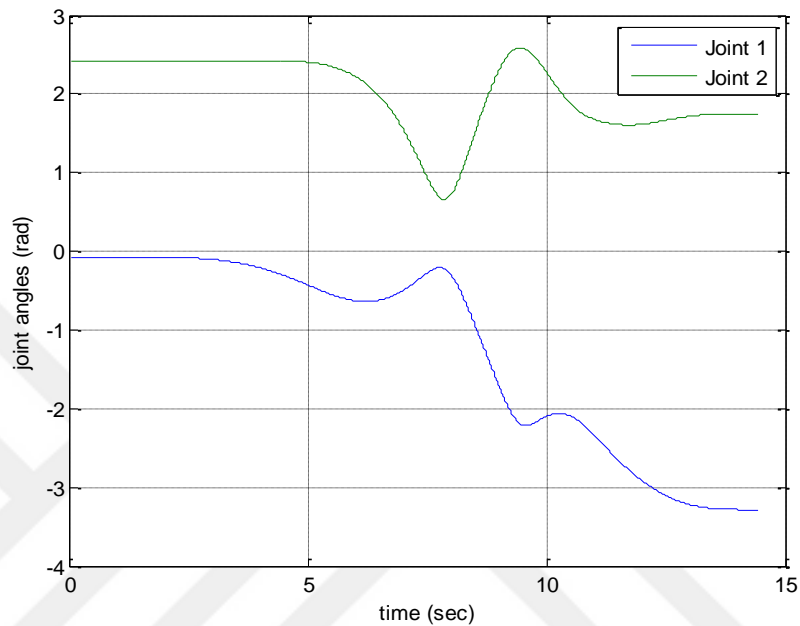


Figure 5.5: The time history of each joint angle. The blue and green curves show the values for joint 1 and joint 2, respectively.

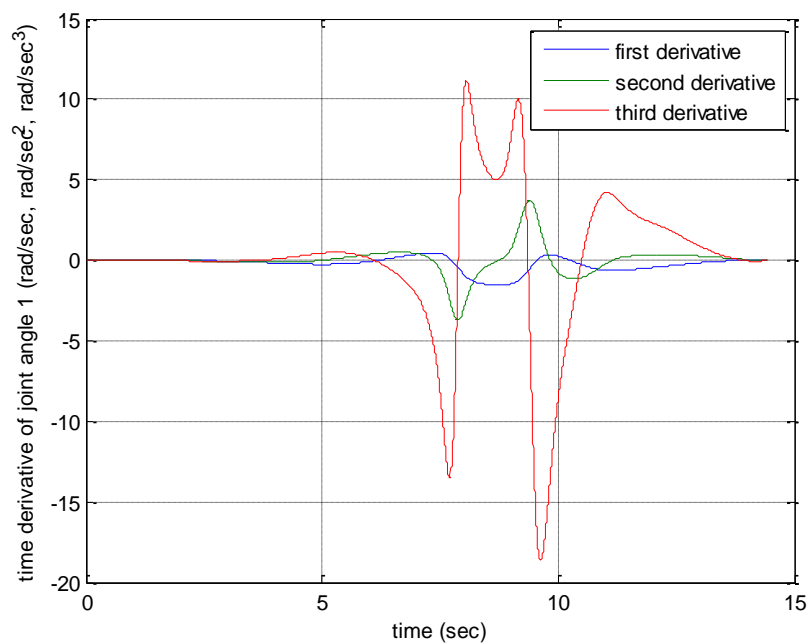


Figure 5.6: The time derivative of the first joint with zero starting and ending shown by green, blue, and red lines, respectively.

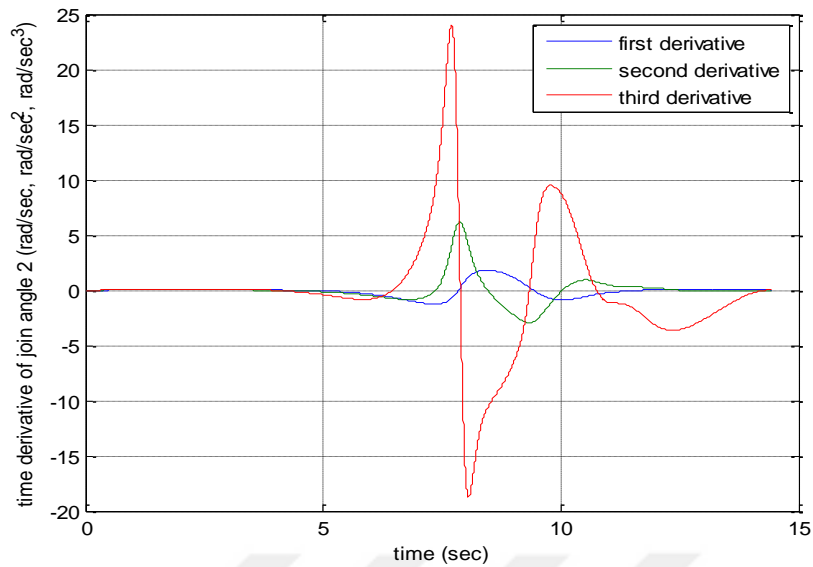


Figure 5.7: The time derivative of the second joint with zero starting and ending shown in green, blue, and red, respectively. The curve exists in the 2-D space.

Figure 5.5 shows the joint angles while Figures 5.6 and 5.7 show the first second and third time-derivatives of the joint angles, respectively. It is obvious that the time derivatives of the joint angles show smoothness at the beginning and end-points where they are equal to zero. The simulations and the figures show that the angular velocity of the joint angles has both positive as well as negative values; therefore, the robot changes its angular velocity. If we take the absolute values of the maximum angular velocities, they are $\begin{bmatrix} 1.5 \\ 1.757 \end{bmatrix} rad/sec$, so they exist within the defined range, which has been shown in Table 5.2 (See appendix A).

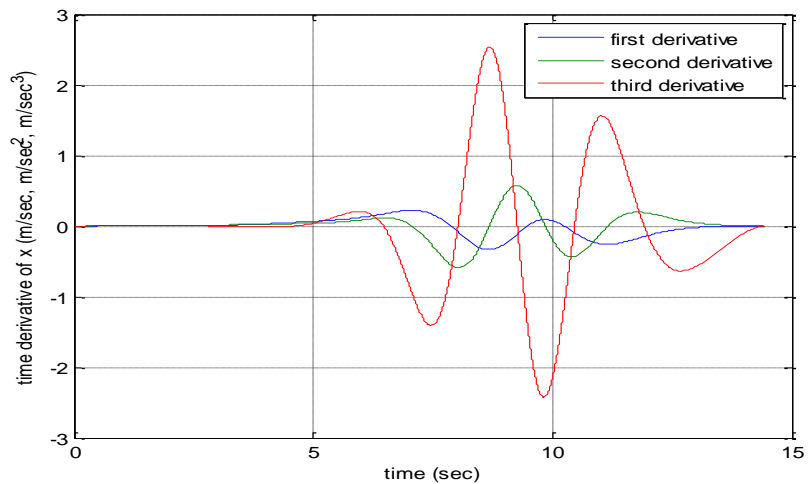


Figure 5.8: The value, velocity, acceleration, and jerk with zero starting and ending points along the x -direction in 2-D Cartesian-space.

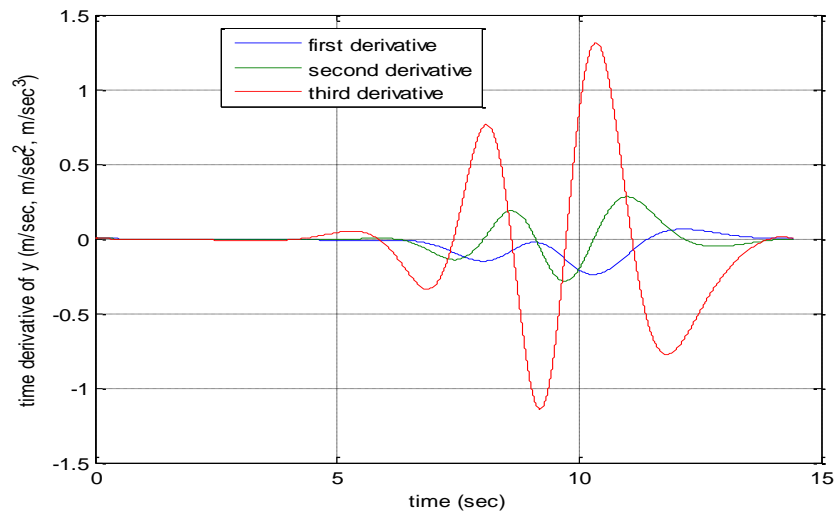


Figure 5.9: The value, velocity, acceleration, and jerk with zero starting and ending along the y -direction in the Cartesian-space.

Figures 5.8 and 5.9 show that the third time derivatives of the joint positions have been used for calculating the joint velocity, acceleration, and jerk, which have zero values at the first and the end-times; therefore, their graphs are smooth, while the end-effector smoothly moves through the Cartesian-space. So far, the functions that described the end-point trajectory either moved in the x direction or the y direction, and we differentiated them with respect to time.

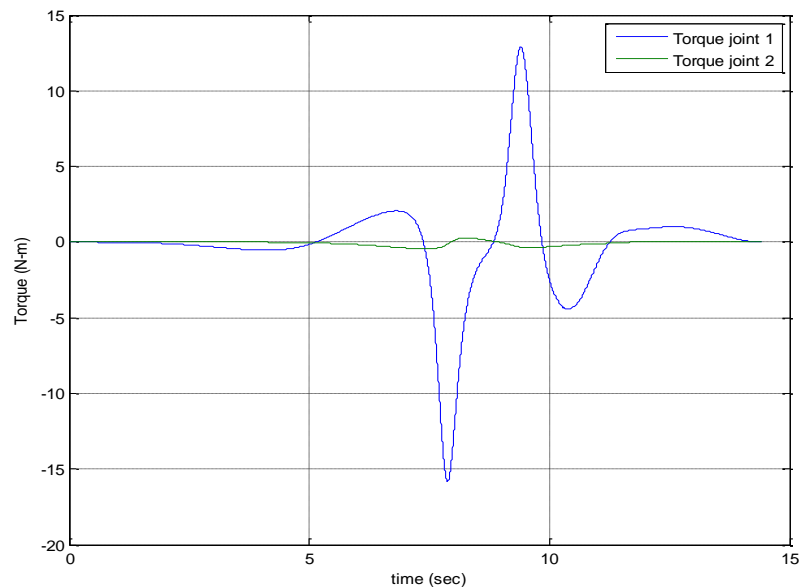


Figure 5.10: Computed torques for each joint angle with zero starting and ending points. The blue and green curves show torque at joint 1 and joint 2, respectively.

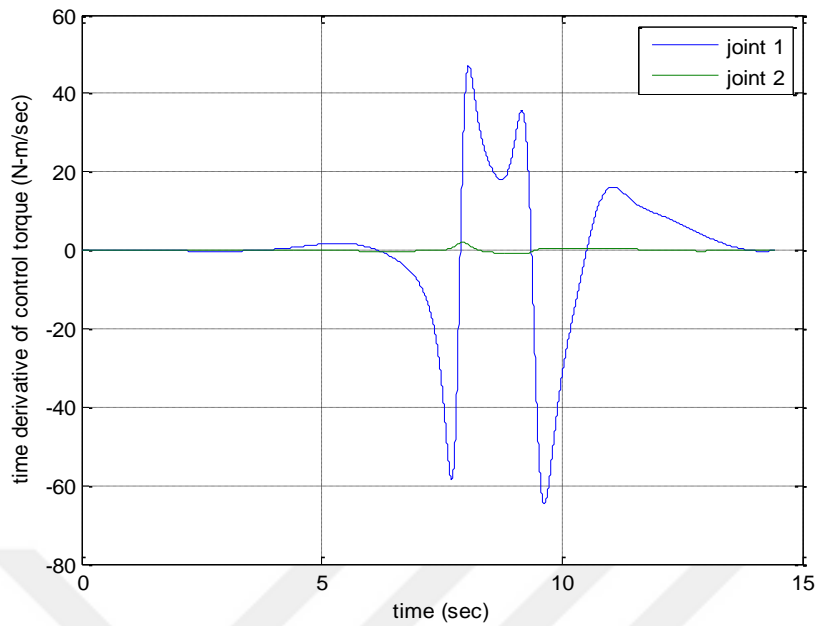


Figure 5.11: The time derivative of the first and second joints with zero starting and ending with torque computations. The blue and green curves correspond to the first and the second derivative of control torque, respectively.

Figures 5.10 and 5.11 show the torque of the joint angles and time derivatives of the torques. Again, the torque, and the first time derivative are smooth, and have zero values at the start and stop-times.

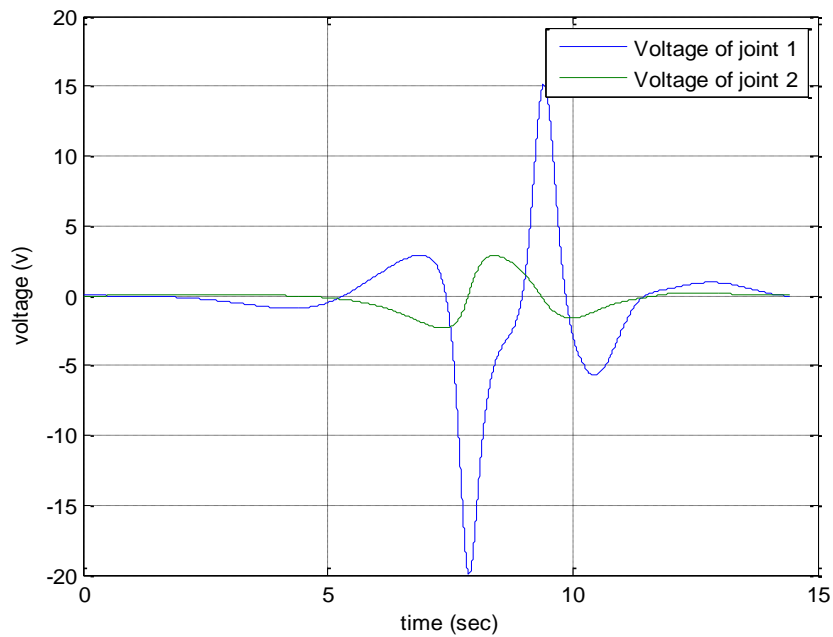


Figure 5.12: Actuator feeding voltages of the motor with zero starting and ending. The blue and green curves show voltage at joint 1 and joint 2, respectively.

The simulations illustrated in Figure 5.12 show the feeding voltage of the motors. They show smooth curves and have zero values both at the starting and stopping points. In this case, it is obvious that the robot uses maximum voltages for link-one, which is (20) volts and uses (15.161) volts for the second-link; so, the feeding voltages are within the range. The algorithm has a smooth input voltage, and smooth path that reduces the dynamic reactions of a robot when the motor operations start and stop.



CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

After considering all the important factors, and conducting the experiments, we have summarized the following conclusions:

The main purpose of this research was to investigate the possibility of improving the path planning algorithm performance for an industrial robot's manipulator. For our research work, we used a new technique based on path parameter optimization algorithm to resolve the path planning problem in the Cartesian-space in the presence of external forces and friction. The actuators of the robot are modelled for permanent-magnet DC motors keeping in view their constraints. The goal of this thesis is to develop a minimum-time path planning mechanism to move the industrial robot from one-point or position to the desired end-point minimizing the travelling time. The aim is to minimize the cost function, which is subjected to constraints such as angular velocities, angular accelerations, angular jerks, input torques, input voltage and final-time; therefore, the key purpose of this research is to optimize the time taken for the robotic manipulator system to move on a pre-defined path. By dividing the time between the start and stop, the path parameter optimization algorithm is converted to optimize the cost function, which is subjected to some equality nonlinear constraints and inequality linear constraints.

MATLAB simulation has been used for path planning solution of a manipulator with a desired path within the Cartesian-space. The path parameter is used for the formulation of the desired path as a function of time. A polynomial model has been considered for the path parameter so that it can guarantee constraints at the starting and ending-points. Consequently, the initial three-time derivatives of the joint angles and the position of the end-effector are equal to zero. In addition, the overall outcome in terms of voltages, torques, and time derivatives of the torques are equal to zeros in the boundary conditions. It is obvious that the approach can hold all the constraints

pertaining to the actuators including other kinematics constraints between the initial and final-times.

Consequently, the method used in this research was sub-optimal because we used polynomial model for the path parameter; therefore, it isn't the global optimum-point. The best advantages of the method include smoothness of all the dynamic and kinematic parameters, and more importantly, we were able to automatically control the start and stop-conditions. The computer-aided simulation study shows satisfactory performance responses of the method for the robotic manipulator's path planning. We introduced a method for path planning problem and parametric trajectories under the external force of the robot manipulator, which is driven by permanent-magnet DC motors. For this purpose, path planning for a robot arm was optimized for finding a minimum-time path planning solution. The path planning was carried out by simulating a model in the MATLAB function, which is based on a given input to the robot along the path for the minimum-time consumption between the start and stop-points. Since this problem is important because it has industrial implications, new operating procedures should be introduced to improve the quality and the functionality of the presented approaches to robotics. Efforts should be made to generate smooth trajectories with minimum jerk and other constraints.

6.2 Recommendations and Future Work

Some possible directions for future work include addition of some new factors and components: Future research efforts should be focused on resolving the path planning problem with better solutions and improvement of the already developed models. Improvement of models by adding the complete modelling of the selected factors in the model and simulating the model in different spaces is highly recommended. For the future work, some research work pertaining to this study can be organized as follows:

- Applying the optimization strategy has been proposed using algorithm for solving minimum-time path planning method for multi-point manufacturing tasks. Using the start-stop movements, the path planning problem has been converted into a Travelling Salesman Problem (TSP) and a series of point-to-point minimum-time transfer path planning problems can be used to parameterize the transfer path and then the path parameters are optimized to

obtain the minimum point-to-point transfer time. A new Travelling Salesman Problem with minimum-time index should be constructed and then solved by using a classical Genetic Algorithm (GA) according to the proposed approach.

- In future, optimization algorithms should be designed and used to solve the path planning problem of robot manipulators by finding the shortest path in real-time in the presence of obstacles. The problem should be approached from the control perspective, and the configuration-space should be searched for path points, which optimize the cost function. This method should be implemented via a multi-pass sequential localized search via a backtracking technique. Further, the proposed technique converges to a global-optimal or a sub-optimal algorithm solution. The specified algorithm should be implemented on a (3-DOF) manipulator arm, and it should be analyzed for cost and time of execution.
- This thesis strongly recommends applying other types of inputs like cubic-spline trajectory planning. The optimal trajectory planning algorithm has been developed for planning minimum-time smooth motion trajectories for manipulators, and it also controls uncertainty, so it is recommended to be included. Optimal trajectory planning algorithm has been divided into two phases. The first phase encompasses derivation of minimum-time optimal trajectory using cubic-spline because of its less vibration and other characteristics. Although cubic-splines are widely used in robotics, velocity and acceleration ripples in the first and last knots can worsen the manipulator trajectory. The second phase includes changing cubic-spline interpolation in the first and last knots of optimized trajectory with 7th order polynomial for having zero jerks at the beginning and end-points of the trajectory. Particle Swarm Optimization (PSO) method has been chosen as optimization algorithm because of its easy implementation and successful optimization performance. Using the 7th order, the mean of 7th order derivative, and the seventh-order spline will require that the control points must satisfy several constraints. For manipulators, the path is generally planned for some-point on the platform. Assuming that it is the centre-point of the platform, the control points require less than eight constraints for positioning include velocity and accelerations, then 7th order derivative should be given some initial-condition. In the

following lines, we'll turn to each parameter of our path planning algorithm and discuss its possible impact on the performance. First of all, we would like to know how the performance changes with the type of chosen trajectory. We have presented and analyzed the techniques for improving trajectory quality by changing the trajectory. Until now, we have implemented two representative trajectory types, and the polynomial point-to-point motion.

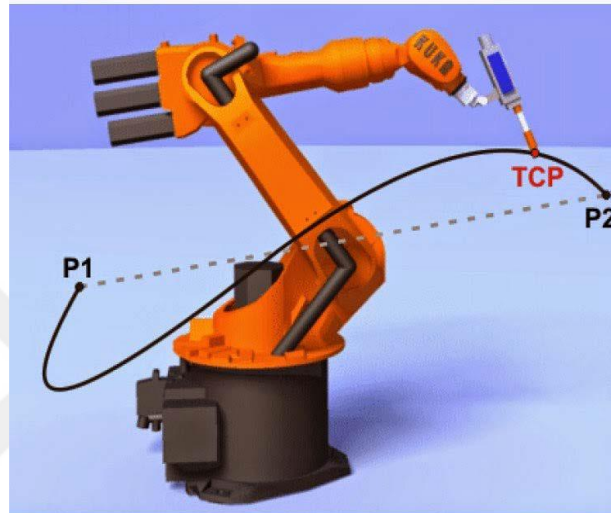


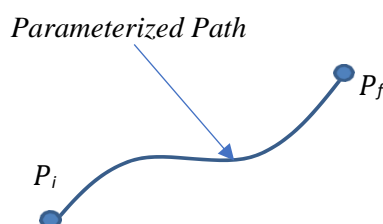
Figure 6.1: Trajectory path parameterization.

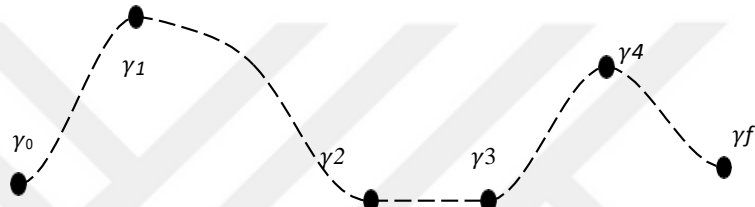
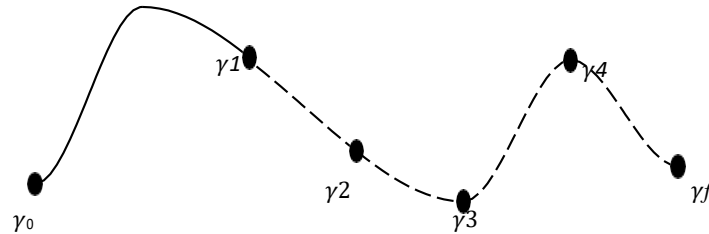
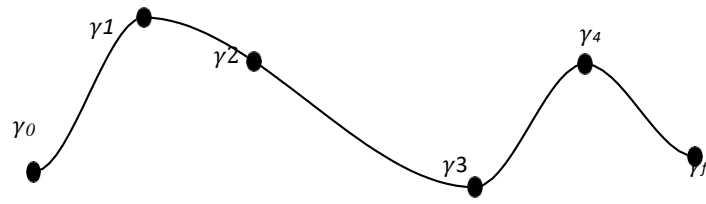
- Polynomial Trajectories

In this section, we'll focus on the trajectory generation with a polynomial. In this setting, the movement of each dimension is a polynomial movement. Firstly, the case, in which, the robot stops at each base point has been analyzed, and afterwards, we'll look into the path motion, on which, the robot does not stop at every base point. Polynomials with a higher-degree can be used but we only need to get a continuous function for calculating the values and the velocity.

- Requirements for Trajectories

Generate a time-optimal path independent of trajectories/band limits.





- Other ways based on the conclusions of this research can be used for time-optimal path planning algorithm for a manipulator to find the best path according to the cost function and taking into account the kinematical constraints of the manipulator. Efforts should be made to assure time efficiency and cost minimization. The manipulator's end-effector should move in a complex 3-D workspace from an initial-configuration to a given final-configuration. Then, the path planning algorithm should be formulated as a global-optimization problem, which is solved using a Genetic Algorithm, and it can be applied to solve the optimization problem with goals and constraints consisting of planning the end-effector trajectory, avoiding collisions between the robots and the obstacles (with small magnitude), and minimizing energy consumption with multiple populations. The path planning problems that we focused on were constrained, since we want to stay within the dynamic limits of the robot. This situation can occur often enough, which can decrease the amount of computations the robot needs to do. The main innovations and contributions of the proposed approach are following:

- Any shape of the robot can be accommodated, and the collision checking is not combinatorial.
- Some kind of constraints for path selection might emerge. These constraints are known as path constraints.
- The manipulator's kinematical constraints (path, maximum velocity, acceleration, and jerk) must be considered.

By suggesting this, we assume that a (2-DOF) manipulator operating in a 3-D workspace is cluttered with static obstacles. The overall requirements that must be satisfied are as follows:

- The obstacles have known position and shape, and they are static.
- The desired initial-point and final-point are fixed and known.

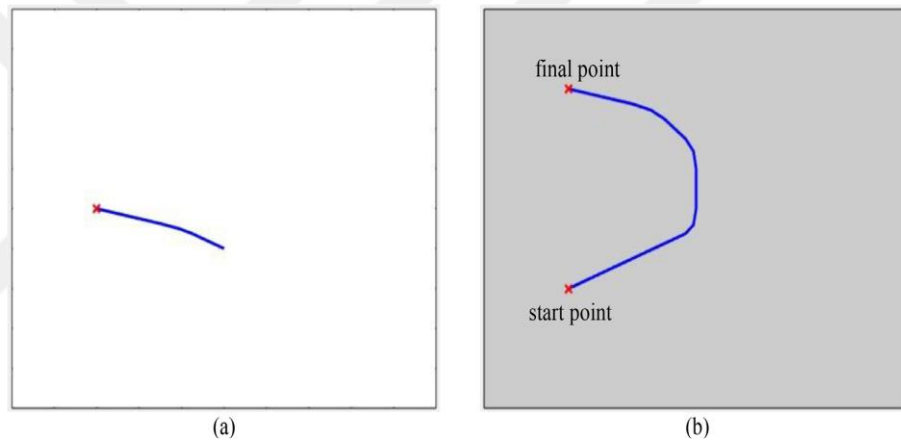


Figure 6.2: (a) The projection of the solution path (blue curve) taking place in X-Y Cartesian-space. (b) The projection of the solution path (blue curve) taking place in X-Y Cartesian-space.

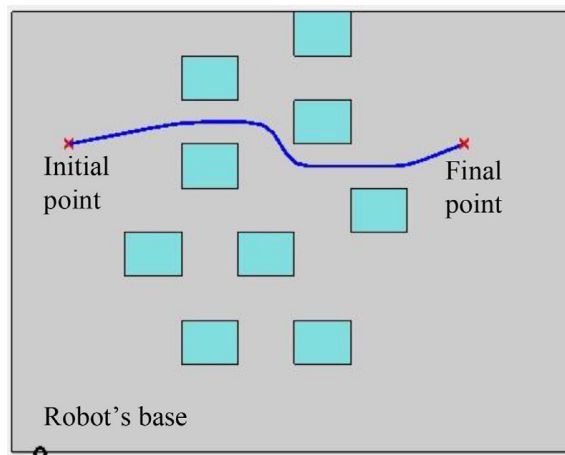


Figure 6.3: The projection of the end-effector's path in X-Y Cartesian-space.

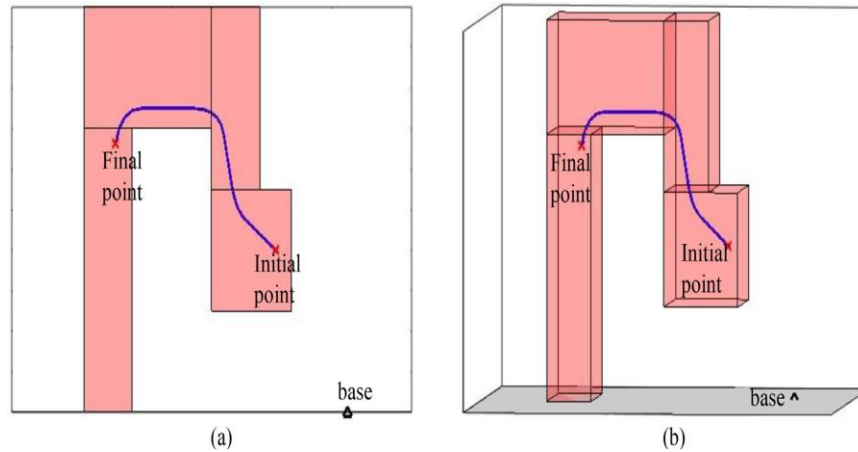


Figure 6.4: (a) The end-effector trace while the manipulator is moving from the starting-point to the final-point. (b) Another point of view.

- The plan is to employ the designed Neural Network (NN) controllers to control the performance of path planning algorithms of robotic manipulators with or without a mathematical model, which would make them effective for both planned and unplanned trajectory tracking problems of any degree-of-freedom. The robotic manipulator can be used to design inverse kinematic controller of the robot arm. A neural network consists of an interconnected group of artificial neurons, and it processes information using the connectionist approach to computation. There is little research that has been done for applying the neural networks for the purpose of path planning for robot manipulators.

Modern neural networks are non-linear statistical data modelling tools. They are usually used to model complex relationships between inputs and outputs or to find patterns in the obtained data. To simulate the given method of path planning, the initial-point and the destination-point were randomly defined while the camera sensor mounted in the environment was used to detect the points of collision and the destination. The neural network used in this method was back-propagation feed-forward network. The path planning method implemented in this work can be extended to use for multi-robot environment, which contains more than one object. By using this method, the path planning algorithm can be accomplished without any prior knowledge of the environment while it also ensures that minor changes in the environment may not affect effective path planning of the robot for future research on this topic.

Examples are: planning for robots, minimal-time planning method, and real-time motion planning.

Neural network is an important area of application in the field of robotic motion planning. It is used as a cost function with some techniques to increase the degree-of-freedom and higher dimensional work spaces but that increases complexity. Thus, the neural network used for solving the inverse kinematics solves path planning problems for robots at a higher degree-of-freedom, which is necessary to improve the quality of the path. It can produce off-line data of position and end-effector of a robot, which helps formulating a relation between position of the joint and the end-effector. A neural network can determine the relation between the joint position and the end-effector.

$$\gamma = f(t_i, \omega, t_f), \gamma(t, P) \quad (6.1)$$

$$P = \begin{bmatrix} t_i \\ \gamma \\ \omega \\ t_f \end{bmatrix} \quad (6.2)$$

Some other parameters can also be considered such as mathematical complexity, stochastic nature of the path planner, local minima solving, and uncertainties etc.

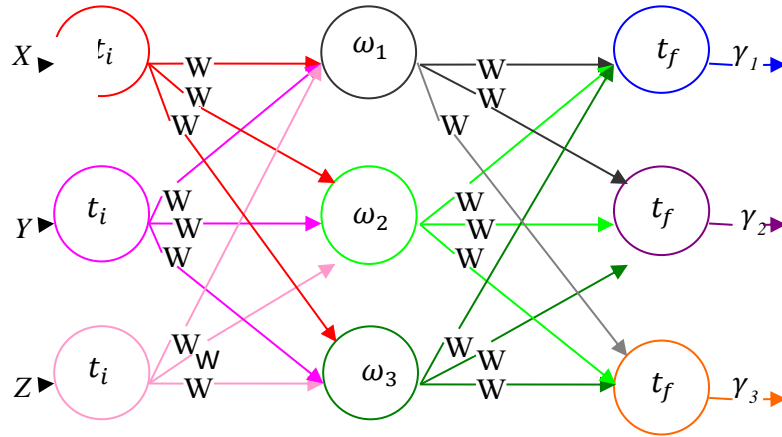


Figure 6.5: Proposed methodology of using neural network architecture.

The proposed neural network will be able to follow the desired joint position and speed trajectories with an acceptable error even in disturbance. The structure of the network, which contained a large number of neurons in the hidden layer, can lengthen the time needed for training and weight adjustment process. This large structure of the network

increases with the degree-of-freedom (DOF) of the manipulator. As mentioned earlier, a number of neurons can be placed in different configurations (layers) to form a neural network (perceptron). A perceptron network may consist of a single layer or multiple layers of neurons. A single layer perceptron consists of a number of neurons placed in a parallel structure. Figure 6.5 shows a single layer perceptron, which consists of a number of neurons while each of them has its own activation function, which might be similar or different from the functions of other neurons, and its own weight vector that connects it to the input vector. The complexity of the input connections with the neurons of a single layer perceptron's is evident from Figure 6.5. Since there is no limit to the number of inputs applied to a single layer perceptron or the number of neurons that may be placed in a layer, a more simplified representative notation of a single layer perceptron has been used, as shown in Figure 6.5. We have applied a cognitive neural network, which is used to determine a collision-free shortest path of a robot from the initial-point to the destination in an unknown environment. Moreover, a neural network has been created, which is used by its non-linear functional approximation.

REFERENCES

1. Latombe, J. C. "Robot Motion Planning," Vol. 124. *Springer Science and Business Media*. 2012.
2. Nottensteiner, K., Bodenmueller, T., Kassecker, M., Roa, M. A., Stemmer, A., Stouraitis, T., and Thomas, U. "A Complete Automated Chain for Flexible Assembly using Recognition, Planning and Sensor-Based Execution," *In ISR 47st International Symposium on Robotics; Proceedings of*, pp. 1-8. 2016.
3. Kröger, T., and Wahl, F. M. "Online Trajectory Generation: Basic Concepts for Instantaneous Reactions to Unforeseen Events," *IEEE Transactions on Robotics*, Vol. 26(1), pp. 94-111. 2010.
4. Røv, M. N. "Time-Optimal Motion Planning and Motion Control for Industrial Manipulators," *Master's Thesis, Institute for Teknisk Kybernetikk*. 2014.
5. Devol, G. C. U.S. Patent No. 3,283,918. *Washington, DC: U.S. Patent and Trademark Office*. 1966.
6. Ernst, A. U.S. Patent No. 3,033,604. *Washington, DC: U.S. Patent and Trademark Office*. 1962.
7. Cook, D. J. "Adding Intelligence to Robot arm Path Planning Using A Graph-match Analogical Reasoning System," *In Intelligent Robots and Systems, 1992., Proceedings of the 1992 IEEE/RSJ International Conference on*, Vol. 1, pp. 657-662. 1992.
8. Chien, R. T., Zhang, L., and Zhang, B. "Planning Collision-Free Paths for Robotic Arm Among Obstacles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. (1), pp. 91-96. 1984.
9. Bobrow, J. E., Dubowsky, S., and Gibson, J. S. "Time-Optimal Control of Robotic Manipulators Along Specified Paths," *The International Journal of Robotics Research*, Vol. 4(3), pp. 3-17. 1985.
10. Kwoh, Y. S., Hou, J., Jonckheere, E. A., and Hayati, S. "A Robot with Improved Absolute Positioning Accuracy for CT Guided Stereotactic Brain Surgery". *IEEE Transactions on Biomedical Engineering*, Vol. 35(2), pp. 153-160. 1998.

11. Skibniewski, M. J., and Russell, J. S. "Robotic Applications to Construction," *Cost Engineering*, Vol. 31(6), pp. 10. 1989.
12. Parker, L. E., and Draper, J. V. "Robotics Applications in Maintenance and Repair," *Handbook of Industrial Robotics*, pp. 1023-1036. 1998.
13. Elfes, A. "Sonar-Based Real-World Mapping and Navigation," *IEEE Journal on Robotics and Automation*, Vol. 3(3), pp. 249-265. 1987.
14. Raja, P., and Pugazhenthii, S. "Path Planning for Mobile Robots in Dynamic Environments using Particle Swarm Optimization," *In Advances in Recent Technologies in Communication and Computing, 2009. ARTCom09. International Conference on*, pp. 401-405. 2009.
15. Barraquand, J., and Latombe, J. C. "Robot Motion Planning: A Distributed Representation Approach," *The International Journal of Robotics Research*, Vol. 10(6), pp. 628-649. 1991.
16. Howard, A., Matarić, M. J., and Sukhatme, G. S. "An Incremental Self-Deployment Algorithm for Mobile Sensor Networks," *Autonomous Robots*, Vol. 13(2), pp. 113-126. 2002.
17. Katrakazas, C., Quddus, M., Chen, W. H., and Deka, L. "Real-Time Motion Planning Methods for Autonomous on-Road Driving: State-of-The-Art and Future Research Directions," *Transportation Research Part C: Emerging Technologies*, Vol. 60, pp. 416-442. 2015.
18. Gasparetto, A., Boscariol, P., Lanzutti, A., and Vidoni, R. "Trajectory Planning in Robotics," *Mathematics in Computer Science*, Vol. 6(3), pp. 269-279. 2012.
19. Spong, M. W., and Vidyasagar, M. "Robot Dynamics and Control," *John Wiley and Sons*. 2008.
20. Li, T., Yang, Y., Hong, B., Ren, J., and Du, J. "A Robust Adaptive Nonlinear Control Approach to Ship Straight-Path Tracking Design," *In American Control Conference, 2005. Proceedings of the 2005*, pp. 4016-4021. 2005.
21. Cook, G. "Mobile Robots: Navigation, Control and Remote Sensing," 2011.
22. Moravec, H. P. "Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover," (No. STAN-CS-80-813). *Stanford University*

Department of Computer Science. 1980.

23. Schwartz, J. T., Sharir, M., and Hopcroft, J. E. "Planning, Geometry, and Complexity of Robot Motion," Vol. 4. *Intellect Books.* 1987.
24. Faverjon, B., and Tournassoud, P. "A Local Based Approach to Path Planning of Manipulators with a High Number of Degrees-of-Freedom," *In Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, Vol. 4, pp. 1152-1159. 1987.
25. Chang, Y. H., Lee, T. T., and Liu, C. H. "Online Approximate Cartesian Path Trajectory Planning for Robotic Manipulators," *IEEE Transactions on Systems Man and Cybernetics*, Vol. 22(3), pp. 542-547. 1992.
26. Cao, B., Doods, G. I., and Irwin, G. W. "Time-Optimal and Smooth Constrained Path Planning for Robot Manipulators," *In Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pp. 1853-1858. 1994.
27. Murray, J. J., and Lovell, G. H. "Dynamic Modelling of Closed-Chain Robotic Manipulators and Implications for Trajectory Control," *IEEE Transactions on Robotics and Automation*, Vol. 5(4), pp. 522-528. 1989.
28. Szádeczky-Kardoss, E., and Kiss, B. "Continuous-Curvature Paths for Mobile Robots," *Technical report, Dept. of Control Engineering and Information Technology, Budapest University of Technology and Economics.* 2008.
29. Bruyninckx, H., Klotzbücher, M., Hochgeschwender, N., Kraetzschmar, G., Gherardi, L., and Brugali, D. "The BRICS Component Model: A Model-Based Development Paradigm for Complex Robotics Software Systems," *In Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pp. 1758-1764. 2013.
30. Macfarlane, S., and Croft, E. A. "Jerk-Bounded Manipulator Trajectory Planning: Design for Real-Time Applications," *IEEE Transactions on Robotics and Automation*, Vol. 19(1), pp. 42-52. 2003.
31. Piazzzi, A., and Visioli, A. "Global Minimum-Jerk Trajectory Planning of Robot Manipulators," *IEEE Transactions on Industrial Electronics*, Vol. 47(1), pp. 140-149. 2009.

32. Calzolari, D., Schürmann, B., and Althoff, M. "Comparison of Trajectory Tracking Controllers for Autonomous Vehicles," *In Intelligent Transportation Systems (ITSC), 2017 IEEE 20th International Conference on*, pp. 1-8. 2017.
33. Shin, K., and McKay, N. "Selection of Near-Minimum-Time Geometric Paths for Robotic Manipulators," *IEEE Transactions on Automatic Control*, Vol. 31(6), pp. 501-511. 1986.
34. Lepetič, M., Klančar, G., Škrjanc, I., Matko, D., and Potočnik, B. "Time-Optimal Path Planning Considering Acceleration Limits," *Robotics and Autonomous Systems*, Vol. 45(3-4), pp. 199-210. 2003.
35. Lloyd, J., and Hayward, V. "Trajectory Generation for Sensor-Driven and Time-Varying Tasks," *The International Journal of Robotics Research*, Vol. 12(4), pp. 380-393. 1993.
36. Constantinescu, D., and Croft, E. A. "Smooth and Time-Optimal Trajectory Planning for Industrial Manipulators along Specified Paths," *Journal of Robotic Systems*, Vol. 17(5), pp. 233-249. 2000.
37. Kim, K. W., Kim, H. S., Choi, Y. K., and Park, J. H. "Optimization of Cubic Polynomial Joint Trajectories and Sliding Mode Controllers for Robots using Evolution Strategy," *In Industrial Electronics, Control and Instrumentation, 1997. IECON 97. 23rd International Conference on*, Vol. 3, pp. 1444-1447. 1997.
38. Incerti, G. "Motion Planning of SCARA Robots for Trajectory Tracking," *World Academy of Science, Engineering and Technology, International Journal of Mechanical Aerospace Industrial Mechatronic and Manufacturing Engineering*, Vol. 9(5), pp. 827-836. 2015.
39. Dongmei, X., Daokui, Q., and Fang, X. "Path-Constrained Time-Optimal Robot Control," *In Robotics and Biomimetics, 2006. ROBIO'06. IEEE International Conference on*, pp. 1095-1100. 2006.
40. Haschke, R., Weitnauer, E., and Ritter, H. "On-Line Planning of Time-Optimal, Jerk-Limited Trajectories," *In Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 3248-3253. 2008.

41. Bianco, C. G. L., and Piazzzi, A. "A Semi-Infinite Optimization Approach to Optimal Spline Trajectory Planning of Mechanical Manipulators," *In Semi-Infinite Programming, Springer, Boston*, pp. 271-297. MA. 2001.
42. Sakamoto, K., and Kawamura, A. "Trajectory Planning using Optimum Solution of Variational Problem," *In Power Conversion Conference, 1993. Yokohama 1993., Conference Record of the*, pp. 666-671. 1993.
43. Ardeshiri, T., Norrlöf, M., Löfberg, J., and Hansson, A. "Convex Optimization Approach to Time-Optimal Path Tracking of Robots with Speed-Dependent Constraints," *IFAC Proceedings Volumes*, Vol. 44(1), pp. 14648-14653. 2011.
44. Park, F. C., Bobrow, J. E., and Ploen, S. R. "A Lie Group Formulation of Robot Dynamics," *The International Journal of Robotics Research*, Vol. 14(6), pp. 609-618. 1995.
45. Bianco, C. G. L., and Ghilardelli, F. "Real-Time Planner in The Operational Space for the Automatic Handling of Kinematic Constraints," *IEEE Transactions on Automation Science and Engineering*, Vol. 11(3), pp. 730-739. 2014.
46. Rajan, V. "Minimum Time Trajectory Planning," *In Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, Vol. 2, pp. 759-764. 1985.
47. Luh, J., and Zheng, Y. F. "Computation of Input Generalized Forces for Robots with Closed Kinematic Chain Mechanisms," *IEEE Journal on Robotics and Automation*, Vol. 1(2), pp. 95-103. 1985.
48. Sahar, G., and Hollerbach, J. M. "Planning of Minimum-Time Trajectories for Robot Arms," *The International Journal of Robotics Research*, Vol. 5(3), pp. 90-100. 1986.
49. Sciavicco, L., and Siciliano, B. "Modelling and Control of Robot Manipulators," *Springer Science and Business Media*. 2012.
50. Van Willigenburg, L. G., Hol, C. W. J., and Van Henten, E. J. "On-Line Near Minimum-Time Path Planning and Control of an Industrial Robot for Picking Fruits," *Computers and Electronics in Agriculture*, Vol. 44(3), pp. 223-237. 2004.

51. Kalmár-Nagy, T., D'Andrea, R., and Ganguly, P. "Near-Optimal Dynamic Trajectory Generation and Control of an Omnidirectional Vehicle," *Robotics and Autonomous Systems*, Vol. 46(1), pp. 47-64. 2004.
52. Shiller, Z. "Time-Energy Optimal Control of Articulated Systems with Geometric Path Constraints," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 118(1), pp. 139-143. 1996.
53. Martin, B. J., and Bobrow, J. E. "Determination of Minimum-Effort Motions for General Open Chains," *In Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, Vol. 1, pp. 1160-1165. 1995.
54. Altintas, Y., and Erkorkmaz, K. "Feedrate Optimization for Spline Interpolation in High-Speed Machine Tools," *CIRP Annals-Manufacturing Technology*, Vol. 52(1), pp. 297-302. 2003.
55. Technitis, G., Othman, W., Safi, K., and Weibel, R. "From A to B, Randomly: A Point-to-Point Random Trajectory Generator for Animal Movement," *International Journal of Geographical Information Science*, Vol. 29(6), pp. 912-934. 2015.
56. Rozo, L., Bruno, D., Calinon, S., and Caldwell, D. G. "Learning Optimal Controllers in Human-Robot Cooperative Transportation Tasks with Position and Force Constraints," *In Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 1024-1030. 2015.
57. Meggiolaro, M. A., and Dubowsky, S. "Improving the Positioning Accuracy of Powerful Manipulators with Application in Nuclear Maintenance," *In Proceedings of the 16th Brazilian Congress of Mechanical Engineering on Robotics and Control*, Vol. 15, pp. 210-219. 2001.
58. Gruber, S., Schiehlen, W., Mang, H. A., Rammerstorfer, F. G., and Eberhardsteiner, J. "Biped Walking Machines: A Challenge to Dynamics and Mechatronics," *In Fifth World Congress on Computational Mechanics Wien: Vienna University of Technology*, pp. 10-27. 2002.
59. Mohamed, Z., and Tokhi, M. O. "Command Shaping Techniques for Vibration Control of a Flexible Robot Manipulator," *Mechatronics*, Vol. 14(1), pp. 69-90. 2004.

60. Eppinger, S., and Seering, W. A. R. R. E. N. P. "Understanding Bandwidth Limitations in Robot Force Control," *In Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, Vol. 4, pp. 904-909. 1987.
61. Dwivedy, S. K., and Eberhard, P. "Dynamic Analysis of Flexible Manipulators, A Literature Review," *Mechanism and Machine Theory*, Vol. 41(7), pp. 749-777. 2006.
62. Cannon Jr, R. H., and Schmitz, E. "Initial Experiments on The End-Point Control of a Flexible One-Link Robot," *The International Journal of Robotics Research*, Vol. 3(3), pp. 62-75. 1984.
63. Book, W. J. "Recursive Lagrangian Dynamics of Flexible Manipulator Arms," *The International Journal of Robotics Research*, Vol. 3(3), pp. 87-101. 1984.
64. Asada, H., Ma, Z. D., and Tokumaru, H. "Inverse Dynamics of Flexible Robot Arms: Modelling and Computation for Trajectory Control," *Journal of Dynamic Systems Measurement and Control*, Vol. 112(2), pp. 177-185. 1990.
65. Lochan, K., Roy, B. K., and Subudhi, B. "A Review on Two-Link Flexible Manipulators," *Annual Reviews in Control*, Vol. 42, pp. 346-367. 2016.
66. Rahimi, H. N., and Nazemizadeh, M. "Dynamic Analysis and Intelligent Control Techniques for Flexible Manipulators: A Review," *Advanced Robotics*, Vol. 28(2), pp. 63-76. 2014.
67. Korayem, M. H., Rahimi, H. N., and Nikoobin, A. "Mathematical Modelling and Trajectory Planning of Mobile Manipulators with Flexible Links and Joints," *Applied Mathematical Modelling*, Vol. 36(7), pp. 3229-3244. 2012.
68. Subudhi, B., and Morris, A. S. "Dynamic Modelling, Simulation and Control of a Manipulator with Flexible Links and Joints," *Robotics and Autonomous Systems*, Vol. 41(4), pp. 257-270. 2002.
69. Chiou, B. C., and Shahinpoor, M. "Dynamic Stability Analysis of a Two-Link Force-Controlled Flexible Manipulator," *Journal of Dynamic Systems Measurement and Control*, Vol. 112(4), pp. 661-666. 1990.
70. Bascetta, L., Ferretti, G., Fossati, F., Magnani, G., Rusconi, A., and Scaglioni, B. "Modelling Identification and Control of a Flexible Lightweight Robot for

- Space Applications,” *In IAA Conference on Dynamics and Control of Space Systems DyCoSS 2014*, pp. 961-976. 2015.
71. Kalra, P., and Sharan, A. M. “Accurate Modelling of Flexible Manipulators using Finite Element Analysis,” *Mechanism and Machine Theory*, Vol. 26(3), pp. 299-313. 1991.
 72. Abedi, E., Nadooshan, A. A., and Salehi, S. “Dynamic Modelling of Tow Flexible Link Manipulators,” *International Journal of Natural Sciences and Engineering*, Vol. 2(4). 2008.
 73. Faris, W. F., Ata, A. A., and Sa'Adeh, M. Y. “Energy Minimization Approach to a Two-Link Flexible Manipulator,” *Journal of Vibration and Control*, Vol. 15(4), pp. 497-526. 2009.
 74. Zohoor, H., and Khorsandijou, S. M. “Dynamic Model of a Flying Manipulator with Two Highly Flexible Links,” *Applied Mathematical Modelling*, Vol. 32(10), pp. 2117-2132. 2008.
 75. Kim, D., and Uchiyama, M. “A Force/Torque Sensor-Less Realization of Fast and Dexterous Tasks with a Parallel Robot,” *In Industrial Electronics Society, 2000. IECON 2000. 26th Annual Conference of the IEEE*, Vol. 1, pp. 223-228. 2000.
 76. Chen, W., Yu, Y., Zhao, X., Zhao, L., and Sun, Q. “Position Control of a (2-DOF) Underactuated Planar Flexible Manipulator,” *In Mechatronics and Automation (ICMA), 2011 International Conference on*, pp. 464-469. 2011.
 77. Benosman, M., and Le Vey, G. “Control of Flexible Manipulators: A Survey,” *Robotics*, Vol. 22(5), pp. 533-545. 2004.
 78. Canudas, C., Astrom, K., and Braun, K. “Adaptive Friction Compensation in DC Motor Drives,” *IEEE Journal on Robotics and Automation*, Vol. 3(6), pp. 681-685. 1987.
 79. Aung, W. P. “Analysis on Modelling and Simulink of DC Motor and its Driving System used for Wheeled Mobile Robot,” *World Academy of Science Engineering and Technology*, Vol. 32, pp. 299-306. 2007.
 80. Robinson, D. W., Pratt, J. E., Paluska, D. J., and Pratt, G. A. “Series Elastic Actuator Development for a Biomimetic Walking Robot,” *In Advanced*

- Intelligent Mechatronics*, 1999. *Proceedings. 1999 IEEE/ASME International Conference on*, pp. 561-568. 1999.
81. Mei, Y., Lu, Y. H., Hu, Y. C., and Lee, C. G. "Energy-Efficient Motion Planning for Mobile Robots," *In Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, Vol. 5, pp. 4344-4349. 2004.
 82. Hanitsch, R. "Energy Efficient Electric Motors," *Energy*, Vol. 645, pp. 227. 2002.
 83. De Almeida, A., Bertoldi, P., and Leonhard, W. "Energy Efficiency Improvements in Electric Motors and Drives," *Springer Science and Business Media*, 2012.
 84. Mark W. Spong, Seth Hutchinson and M. "Vidyasagar, Robot Modelling and Control," *John Wiley and Sons*, 2006.
 85. Denavit J. and R. S. Hartenberg, "A Kinematic Notation for Lower-pair Mechanisms based on Matrices," *Journal of Applied Mechanics*, pp. 215-221. 1955.
 86. Craig J. J. "Introduction to Robotics Mechanics and Control," *USA: Addison-Wesley Publishing Company*. 1989.
 87. Dombre E. and W. Khalil "Modelisation et Commande des Robots," *Hermes*, Paris. 1988.
 88. F. L. Lewis and C. T. Abdullah and D. M. Dawson "Control of Robot Manipulators," *New York, Macmillan*, 1993.
 89. Khalil, W., and Dombre, E. "Modelling Identification and Control of Robots," *London, U.K. HPS*, 2002.
 90. Moojin, K., Wonkyu, M., Daesung, B., and Ilhan, P. "Dynamic Simulations of Electro-Mechanical Robotic Systems Driven by DC Motors," *Robotica* 2004, Vol. 22, pp. 523-531. 2004.
 91. Chettibi, T., and Lemoine, P. "Generation of Point-to-Point Trajectories for Robotic Manipulators Under Electro-Mechanical Constraints," *International Review of Mechanical Engineering*, ISSN 1970-8734, Vol. 1(2), pp. 131. 2007.

APPENDIX

Appendix [A]: TABLES**1. Kinematics Parameters:****Table 3.1:** Kinematics parameters.

<i>Arm Parameters</i>	<i>Symbol</i>	<i>Revolute Joint</i>	<i>Prismatic Joint</i>
<i>Joint Angle</i>	θ	<i>Variable</i>	<i>Fixed</i>
<i>Joint Distance</i>	d	<i>Fixed</i>	<i>Variable</i>
<i>Link Length</i>	a_{i-1}	<i>Fixed</i>	<i>Fixed</i>
<i>Link Twist Angle</i>	α_{i-1}	<i>Fixed</i>	<i>Fixed</i>

2. Dynamic Parameters:**Table 3.2:** Representing the SCARA robot parameter values.

<i>Parameters</i>	<i>Symbol</i>	<i>Values</i>
<i>Mass of the first link-1</i>	m_1	<i>1 kg</i>
<i>Mass of the second link-2</i>	m_2	<i>1 kg</i>
<i>Mass of the third link-3</i>	m_3	<i>1 kg</i>
<i>Length of the link-1</i>	$l_1 = a_1$	<i>1 m</i>
<i>Length of the link-2</i>	$l_2 = a_2$	<i>1 m</i>
<i>Length of the link-3</i>	d	<i>1 m</i>

3. Denavit-Hartenberg (DH) Parameters:

Table 3.3: The SCARA robot arm parameters.

Link	Joint Angle θ	Twist Angle α_{i-1}	Link Length a_{i-1}	Link Offset d
1	θ_1	0	l_1	0
2	θ_2	0	l_2	0
3	θ_3	π	0	$-d$

*For joint variable

4. Generating Constraints:

Table 4.1: Managing constraints for the optimization problem at time t_k .

<i>Initial-time</i>	<ul style="list-style-type: none"> - Initial estimate of the final-time $t_f(0)$ - Generating random vector x
<i>For $k = (1, \dots, m)$</i>	<ul style="list-style-type: none"> - Using of x, calculate $\gamma_k, \dot{\gamma}_k, \ddot{\gamma}_k, \ddot{\gamma}_k$ - Using equation (4.20), calculate $r_k, \dot{r}_k, \ddot{r}_k, \ddot{\gamma}_k$ - Calculate: $q_k = q(r_k)$ J_k and \dot{J}_k $\dot{q}_k = J_k^{-1} r_k' \dot{\gamma}_k \dot{J}_k$ $v = J_k \dot{q}_k$ - Calculate: $M_k(q_k), C_k(q_k, \dot{q}_k), G_k, T_{ext_k}, T_{f_k}$ $\ddot{q}_k = J_k^{-1} (-\dot{J}_k \dot{q}_k + r_k'' \dot{\gamma}_k^2 + r_k' \ddot{\gamma}_k)$ $T_{c_k} = M_k \ddot{q}_k + C_k \dot{q}_k + G_k q_k - T_{ext_k} - T_{f_k}$ - Calculate: $\dot{M}_k(q_k, \dot{q}_k), \dot{C}_k, \dot{G}_k, \dot{T}_{ext_k}, \dot{T}_{f_k}$ $\ddot{q}_k = J_k^{-1} (-\dot{J}_k^{-1} \dot{q}_k - 2\dot{J}_k \ddot{q}_k + r_k''' \dot{\gamma}_k^2 + 2r_k'' \ddot{\gamma}_k + r_k' \ddot{\gamma}_k)$ $\dot{T}_{c_k} = \dot{M} \ddot{q} + M \ddot{q} + \dot{C} \dot{q} + C \ddot{q} + \frac{\partial G}{\partial q} \dot{q} - \dot{T}_{ext} - \dot{T}_f$ $U_k = D^{-1} (\dot{T}_{c_k} - A \dot{T}_{c_k} - B \dot{q}_k)$ - Constraints are:

$$C_k = \begin{cases} U_k - \bar{U} \\ \underline{U} - U_k \\ T_{ck} - \bar{T}_c \\ \underline{T}_c - T_{ck} \\ \dot{q}_k - \bar{q} \\ \underline{\dot{q}} - \dot{q}_k \\ \dot{T}_{ck} - \bar{\dot{T}}_c \\ \underline{\dot{T}}_c - \dot{T}_{ck} \\ \gamma_k \geq 0 \\ \dot{\gamma}_k \geq 0 \end{cases} \leq 0$$

5. Parameters of the Robot Manipulator:

Table 5.1: Parameters of the IRCCyN SCARA robot.

<i>Task</i> N^0	μ	K_v	K_c	$l(m)$
1	0.2	0.4	0.01	0.21
2	0.2	0.3	0.01	0.3

6. The Electro-Mechanical Constraints of the Actuators:

Table 5.2: Electro-mechanical constraints of the SCARA robot.

<i>Axis</i>	\bar{U} (volt)	q (deg)	$\bar{\dot{q}}$ (rad.s ⁻¹)
1	20	0-270	1.5
2	20	0-180	2

Resume/CV

Name:

Majdi Farag Abdulati

Omar Al-Mukhtar University, Libya.

Tel: +218 (91) 3998753

Email: magdyfarag38@yahoo.com



Born on the 5 November in 1983 in Libya–Libyan nationality.

Education:

- BSc in Mechanical Engineering, Omar Al-Mukhtar University/ Libya / College of Science / Department of Mechanical Engineering, 2005.
- MSc in Department of Automatic Control and Flight Dynamics, Alexandria University, Egypt, 2008-2011.

Experience:

Omar Al-Mukhtar University, 2011-present

Working as a lecturer teaching following courses:

- Nonlinear Vibrations.
- Fundamental of Mechanical Vibrations.
- Mechanical Vibrations.

Languages:

- Excellent oral and written communication skills in Arabic and English.
- Intermediate oral and written communication skills in Turkish.

Research:

- Majdi. F. Mohamed, E. M. Attia, N. A. Maharem, “Effect of Road Excitation on the Vehicle Vibration Suspended by Electro-Rheological (ER) Damper”, *International Review of Mechanical Engineering IREME, Praise Worthy Prize S.R.L Naples*, Vol. 7, No. 3, March 2013. pp. 529-537, 2013.

Activities:

- Sports
- Reading