Università degli Studi di Padova

(UNIPD)

DIPARTIMENTO DI TECNICA E GESTIONE DEI SISTEMI INDUSTRIALI (DTG)

Master thesis in Management Engineering

ENERGY MODELLING AND SIMULATION FOR INDUSTRIAL ROBOTS

Supervisor

Prof. Boscariol Paolo

Master Candidate

Bushra Ahmed Mohamed Ahmed

Student ID:

2043127

Academic Year

2023-2024

بسم الله الرحمن الرحيم

سورة طه

فَتَعَالَى اللَّهُ الْمَلِكُ الْحَقُّ ۗ وَلَا تَعْجَلْ بِالْقُرْآنِ مِنْ قَبْلِ أَنْ يُقْضَىٰ إِلَيْكَ وَحْيُهُ ۖ وَقُلْ رَبِّ زِدْنِي عِلْمًا ﴿١١٤﴾

Surah Ta-Ha

114. So Exalted is Allah, the True Sovereign. Do not be hasty with the Quran before its inspiration is concluded to you, and say, "My Lord, increase me in knowledge."

Abstract

This thesis explores energy modelling and simulation techniques tailored for industrial robots, with a primary objective of advancing energy efficiency. Focusing on the ABB-IRB-140 robot, the study utilizes MATLAB to develop comprehensive energy models for three distinct motions. The research unfolds through various objectives, including formulating kinematics, developing motion planning algorithms, conducting simulations, and constructing energy consumption models for individual robot joints.

A pivotal aspect of this research lies in the development of a robust motion planning algorithm, recognized as a fundamental pillar that underpins the entire endeavour. This algorithm serves as a critical mechanism for optimizing energy efficiency and seamlessly integrating energy modelling techniques into real-world industrial applications. While MATLAB customization caters to specific robot characteristics, the developed algorithm boasts versatility, enabling its adaptation across a spectrum of industrial contexts and robot configurations.

By elucidating the intricate relationship between motion planning and energy consumption in industrial robots, this research contributes to a deeper understanding of energy dynamics within the industrial landscape. Moreover, the insights gleaned hold the promise of significant advancements in energy-efficient robotics, fostering sustainable practices and mitigating the environmental impact associated with industrial operations. Ultimately, this thesis represents a crucial step forward in the quest for energy optimization, highlighting the transformative potential of interdisciplinary research at the nexus of engineering and sustainability.

Dedication

In the Name of Allah, the Compassionate, the Merciful and prayers and peace upon our Prophet Muhammed.

This work is dedicated to my family, whose unwavering support has been a constant throughout every phase of my life. In particular, I dedicate it to my mother, whose presence I feel watching over me from the heavens with a caring eye.

I also dedicate this work to my friends scattered across the globe; your presence in my life has been a source of immense strength and comfort.

To all my fellow Sudanese, I dedicate this work. Being Sudanese is a mark of resilience and strength, and I urge you to continue your fight with unwavering determination.

In solidarity with the people of Palestine, I dedicate this work, praying that Allah may swiftly alleviate your suffering.

To the oppressed souls in Congo and around the world, this work is dedicated. May my words serve as a beacon of hope and awareness in the ongoing struggle for justice and freedom.

May this work contribute, even if in a small way, to making the world a better place for all.

Ahmed Bushra

Padova, Italy

February 2024

Acknowledgment

In the name of Allah, the Most Gracious, the Most Merciful. Alhamdulillah, all praises to Allah for the strength and blessings that enabled me to complete this thesis.

I extend special appreciation to my supervisor, Prof. Paolo Boscariol, for his unwavering guidance and support. His invaluable feedback and suggestions throughout the thesis have greatly contributed to its success. This project has provided me with invaluable real-world experience, and I am grateful for the opportunity.

I would also like to express my gratitude to UNIPD and specially the DTG department for their efforts in providing us with essential information and facilitating our journey, enabling us to effectively implement our education into real-time project design and analysis.

List of Contents

Abstract V

List of Figures XIV

List of Tables XVII

Listing of acronyms XIX

1. Introduction 1
   - 1.1 Background 2
   - 1.2 Industrial Robots (IR) 3
   - 1.3 Research Significance 6
   - 1.4 Research Objectives 6
   - 1.5 Thesis Structure 6
2. Theoretical Background & Literature Review 8
   - 2.1 Introduction 9
   - 2.2 Brief History of Industrial Robotics 9
     - 2.2.1 The First Industrial Robotic Generation (1950-1967) 10
     - 2.2.2 The Second Industrial Robotic Generation (1968-1977) 11
     - 2.2.3 The Third Industrial Robotic Generation (1978-1999) 13
     - 2.2.4 The Fourth Industrial Robotic Generation (2000-today) 16
   - 2.3 Structure of Industrial Robots 16
   - 2.4 ABB IRB 140 ROBOT 17
     - 2.4.1 Technical Specifications 18
   - 2.5 Robots Modelling 19
   - 2.6 Robots' Kinematics 20
     - 2.6.1 Direct Kinematics (D.K) 21
     - 2.6.2 Inverse Kinematics (I.K) 23
     - 2.6.3 Differential Kinematics 24
   - 2.7 Motion Planning and Trajectory Generation 26

2.8 Energy Modelling and Simulation 30
3. Methodology 33
3.1 Introduction 34
3.2 Assumptions 36
3.3 Direct Kinematics 37
3.4 Inverse Kinematics 39
3.5 Jacobian Matrix 41
3.6 Motion Planning 42
3.7 Robot Dynamics and Path Tracking 43
3.8 Energy Modelling and Simulation 45
4. Results and Discussions 47
4.1 Introduction 48
4.2 Data Given 48
4.3 Numerical Results 51
4.4 Graphical Results 53
5. Conclusion and Recommendations 70
5.1 Conclusion 71
5.2 Recommendations 72
6. References & Bibliography 74

List of Figures

3.1 Major Reasons to invest in Industrial Robots 1

3.2 Annual Installation of Industrial Robots 5

3.3 Annual Installation of Industrial Robots by Customer Industry 5

2.1 UNIMATE Robot 10

2.2 STANDFORD Arm 11

2.3 PUMA Robot 11

2.4 IRB-6 developed by ASEA (ABB now) 12

2.5 First SCARA Robot by Hiroshi Makino 14

2.6 Direct-Driven SCARA Robot 14

2.7 Six Delta in Swiss Company Demaurex's Production line 15

2.8 Fastest Picking Robot in 1998 ABB Flex-Picker 15

2.9 Mechanical Structure of a Robotic Arm 17

2.10 The 3D-Model of ABB-IRB-140 18

2.11 Axis and Dimensions of ABB-IRB-140 18

2.12 Several References Frames 19

2.13 The Denavit_Hartenberg (DH) Parameters 22

2.14 Four Different Solutions 23

2.15 Linear Path with Parabolic Blends 27

2.16 Multisegment Linear Path with Parabolic Blends 27

3.1 Methodology Flowchart 35

3.2 ABB-IRB-140 Axis and Links 36

3.3 Projection of the Wrist Centre onto the XY Plane 40

3.4 Projection onto the Plane formed by Link 2 and 3 40

3.5 A Path with Three Via Points 44

3.6 A Path with Triangular Shape 44

3.7 A Path with Pick and Place Task 45

4.1 End-Effector Positions, Speeds, and Accelerations vs. Time for Via_Points Path 54

4.2 Joint Angles Positions, Speeds, and Accelerations vs. Time for Via_Points Path 55
4.3 End-Effector Positions, Speeds, and Accelerations vs. Time for Trinagular_Shape Path 56
4.4 Joint Angles Positions, Speeds, and Accelerations vs. Time for Trinagular_Shape Path 57
4.5 End-Effector Positions, Speeds, and Accelerations vs. Time for Pick_and_Place Path 58
4.6 Joint Angles Positions, Speeds, and Accelerations vs. Time for Pick_and_Place Path 59
4.7 3D Representation of the ABB-IRB-140 Robot Movement in the Via_Point Path. 60
4.8 3D Representation of the ABB-IRB-140 Robot Movement in the Triangular_Shape Path. 61
4.9 3D Representation of the ABB-IRB-140 Robot Movement in the Pick_and_Place Path. 61
4.10 Joint and Motor Torque for Via_Points Path 62
4.11 Joint and Motor Torque for Triangular_Shape Path 63
4.12 Joint and Motor Torque for Pick_and_Place Path 64
4.13 Electrical Power Dissipation and Total Energy Consumption in Via_Points Path 66
4.14 Electrical Power Dissipation and Total Energy Consumption in Triangular_Shape Path 67
4.15 Electrical Power Dissipation and Total Energy Consumption in Pick_and_Place Path 68

List of Tables

3.1 Denavit-Hartenburg Parameters for the ABB-IRB-140 Robot 37

4.1 The Shared Data Among the Three Paths 49

4.2 The Specific Data Tailored to each Path 50

4.3 The numerical results derived from modelling and simulating the energy for three distinct paths 51

Listing of Acronyms

IR/IRB	Industrial Robots
IRM	Industrial Robotic Manipulator
ISO	International Organisation of Standardisation
SCARA	Selective Compliance Assembly Robot Arm
IFR	International Federation of Robotics
DOF	Degree Of Freedom
DH	Denavit-Hartenberg
D.K	Direct Kinematic
I.K	Inverse Kinematic
M.P	Motion Planning

CHAPTER ONE:

INTRODUCTION

1.1 Background

Robotics is a captivating field focused on constructing machines capable of executing tasks, emulating living organisms, and understanding their surroundings. The concept of creating such devices has intrigued humans since the early stages of object construction. Handcrafted devices mimicking human movements, such as the clockwork figures in Venice's San Marcos clock tower, the narrative-telling figurines in Prague's fifteenth-century astronomical clock, and the intricate systems envisioned by Renaissance artist Leonardo da Vinci in his notebooks, serve as early examples of this fascination [1].

Scientists often feel a profound connection to their work, gaining insights into themselves through their research. This connection is apparent not only in the realm of physics but also in disciplines like psychology and chemistry. The study of robotics, in particular, highlights this relationship, aligning with the engineering inclination towards synthesis rather than a purely analytical scientific approach. Perhaps it is this interdisciplinary nature that contributes to the widespread interest in the field [2].

Today, robots have become integral components of our daily lives, omnipresent across industries, homes, oceans, and even outer space. These technological marvels, driven by intricate programming and artificial intelligence, are reshaping industries, pushing creative boundaries, and influencing our daily existence [3].

The impact of robots on manufacturing is profound, revolutionizing production quality, efficiency, and flexibility while reducing costs. Equipped with the right hardware, industrial robots (IR) can perform an extensive array of tasks, including material handling, welding, painting, assembly, and palletizing. Their capabilities surpass those of humans, reaching farther, moving faster, and lifting heavier loads, all with superior reliability and accuracy. Industrial robots are not only efficient but are also adept at undertaking tasks perceived as repetitive, messy, hazardous, or requiring precision beyond human capabilities [4].

1.2 Industrial Robots (IR)

The International Organisation for Standardisation (ISO) defines an "industrial robot" as an "*automatically controlled, reprogrammable multipurpose manipulator, programmable in three or more axes, which can be either fixed in place or attached to a mobile platform for use in automation applications in an industrial environment*" [5].

Key terms in this definition include:

- Reprogrammable: Capable of modifying programmed motions or auxiliary functions without physical alterations to the mechanical system.
- Multipurpose: Adaptable to new applications through physical modifications.
- Axis: The direction determining whether a robot moves linearly or rotationally.
- Manipulator: A mechanism composed of segments connected or sliding with respect to each other.

Industrial robots are categorized based on their mechanical configuration:

- Cartesian robot (gantry or rectangular robot): Possesses three prismatic joints, forming a Cartesian coordinate system.
- SCARA robot: Features two parallel rotary joints ensuring adherence to a predetermined plan.
- Articulated robot: Has three or more rotational joints.
- Parallel or delta robot: Exhibits a closed-loop structure formed by linked arms.
- Cylindrical robot: Comprises at least one prismatic and one rotary joint, forming a cylindrical coordinate system.
- Polar robot (spherical robot): Contains one prismatic joint, two rotary joints, and axes forming a polar coordinate system.

Industrial Robots stand out as key enablers in the industry 4.0 revolution, offering benefits that extend beyond bolstering productivity and competitiveness. They bring increased flexibility, facilitating rapid adaptation to production changes, responding

to fluctuating demand, and handling smaller batch sizes. Additionally, these robots enhance resilience, navigating production peaks, and demonstrating resilience in the face of systemic shocks, such as witnessed during the COVID-19 pandemic. Contributing to energy and resource efficiency, they optimize performance by reducing energy consumption, minimizing material waste, and increasing overall yield. Furthermore, their impact includes improved productivity, providing crucial support to manufacturing employees by elevating work quality and ensuring adherence to health and safety regulations. This transformative technology also brings about cost reductions, both in operating and capital expenses, while simultaneously enhancing product quality, increasing production output rates, and optimizing space utilization, particularly in high-value manufacturing areas. [5].

Figure (1.1) shows major reasons to invest in industrial Robots.

Due to the factors outlined earlier, the installation of Industrial Robots has reached a historic high, totalling 553,052 units, according to the statistical department of the International Federation of Robotics (IFR). Notably, this marks the second consecutive year where annual installations have surpassed the 500,000-unit threshold. The automotive and electronics sectors, being major customer industries, significantly increased their robot installations compared to the previous year.

Figures (1.2) and (1.3) depict the annual installation of industrial robots overall and by customer industry, respectively. [6].

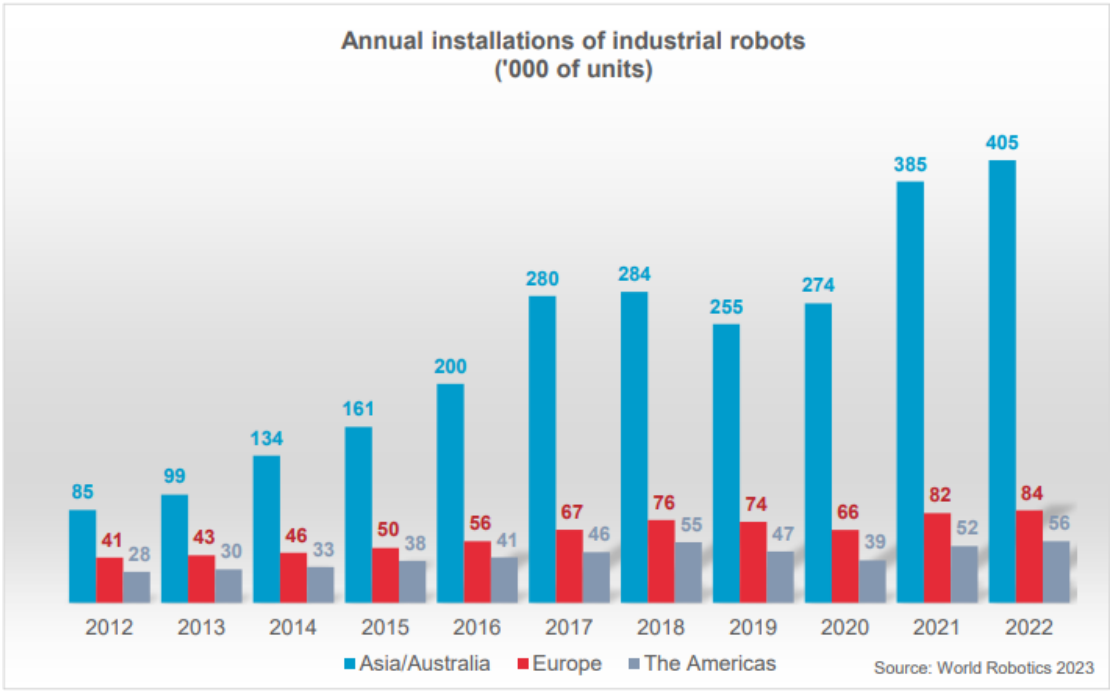


Figure (1.2) shows annual installation of industrial robots.

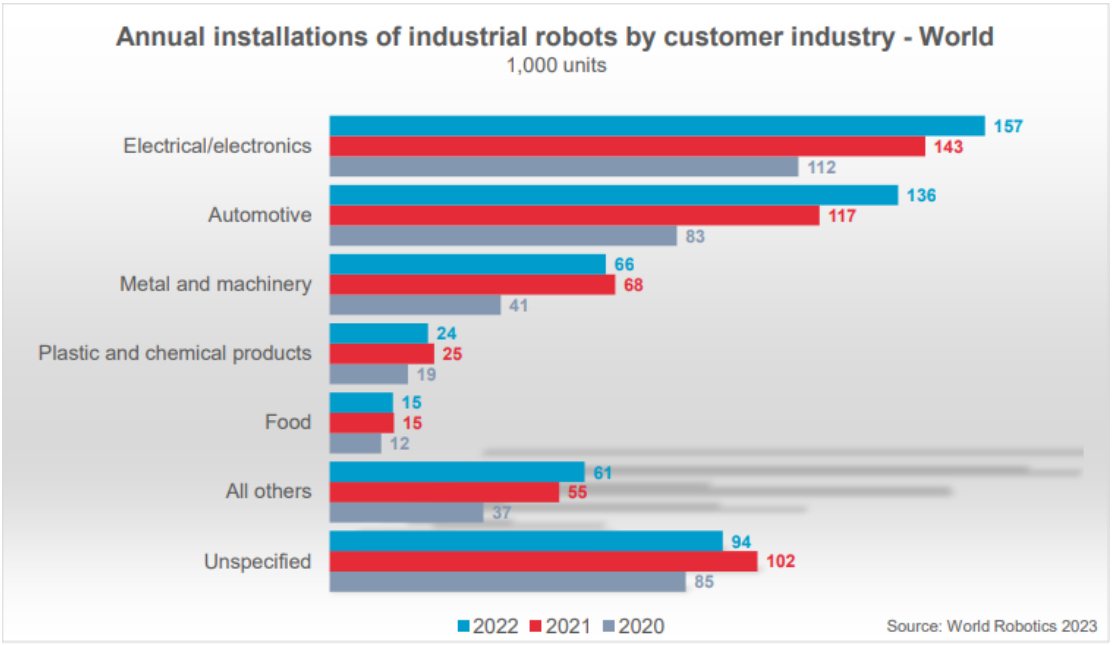


Figure (1.3) shows annual installation of industrial robots by customer industry.

1.3 Research Significance

The industrial sector currently accounts for a significant portion, around a quarter, of the total energy consumption [7]. The imperative to reduce energy usage has become more critical than ever, with both economically developed and developing nations committing to energy reduction initiatives [8]. Among the contributors to this energy consumption in the industrial sector are industrial robots, which play various roles in different industrial processes. Enhancing the energy efficiency of these robots can significantly impact the overall energy consumption of the industrial sector. While numerous measures and practices exist to enhance the energy efficiency of industrial robots (IR), a foundational step involves modelling and simulating their energy consumption to understand their energy consumption patterns and to facilitate further improvements. In this thesis project, the focus is on an industrial robot model, specifically the ABB-IRB-140, where energy modelling and simulation using MATLAB will be implemented. Three distinct motions have been planned and selected to develop comprehensive energy models.

1.4 Research Objectives

- Formulating the kinematics for the examined robot, encompassing both position kinematics and differential kinematics.
- Creating an algorithm for motion planning with the aim of enhancing the energy efficiency of robots.
- Conducting simulations for diverse motions performed by the robot under investigation.
- Constructing models to estimate the energy consumption associated with each joint of the robot.

1.5 Thesis Structure

This is the structure of the thesis. In Chapter 2, the geometry of the examined manipulator (ABB-IRB-140) will be presented along with a comprehensive review of

all the literature and previous studies utilized to derive the final outcome. Chapter 3 will provide an in-depth exploration of the methodology. Chapter 4 focuses primarily on the analysis and the generated outcomes. The results collected will be examined, and corresponding recommendations will be formulated in Chapter 5. The thesis concludes with a bibliography and a list of references.

CHAPTER TWO:

THEORETICAL BACKGROUND &

LITERATURE REVIEW

2.1 Introduction

This chapter will conduct a thorough analysis of robotics research, with a specific focus on modelling robots and their energy usage. It aims to illuminate key areas within robotics and enhance understanding of the thesis objectives. Beginning with a concise overview of industrial robotics' historical development and classification, it will outline typical industrial robot structures. The specific robot under study, central to the thesis objectives, will then be introduced. Subsequent sections will delve into various topics, including robot kinematics, motion planning, robot dynamics, and energy modelling, each supported by relevant theoretical frameworks and previous studies.

2.2 Brief History of Industrial Robotics

As mentioned in chapter one, according to ISO (The International Organization for Standardization) defines an “industrial robot” as an “*automatically controlled, reprogrammable multipurpose manipulator, programmable in three or more axes, which can be either fixed in place or attached to a mobile platform for use in automation applications in an industrial environment*” [5].

Some of these machines, known as automata, date back to the Greek-Hellenistic era. Over the ages, clever inventors from a variety of cultures have invented and made them. The word "robot" has a broader definition, whereas the term "automata" mostly refers to technologies that resemble humans. The word "robot" has a more modern origin; specifically, it derives from the Czech word "robota," which means "forced labour" or "heavy work." The phrase "R.U.R.: Rossum's Universal Robots" was first used in 1920 by Czech novelist Karel Čapek (1890–1938), who is credited with introducing it [10]. However, Isaac Asimov (1920–1992) used the term "robotics" for the first time in his 1942 novel "Runaround," which is part of the well-known "I, Robot" series. He outlined three guidelines for robot behaviour and human interaction in that book; these guidelines would subsequently be known as the three Laws of Robotics [11].

Certain writers suggested classifying industrial robots into four "generations" in order of chronological order. [12]

2.2.1 The First Industrial Robotic Generation (1950–1967)

Between 1950 and 1967, the first wave of industrial robots emerged. These machines were essentially programmable but lacked the ability to adapt task methods or interact with their environment. They operated using basic hardware without advanced servo-controllers. A distinctive feature was the loud noise generated when their arms collided with mechanical stops limiting movement. Pneumatic actuators were common, controlled by basic "logic gates" acting as automatic regulators. These gates, often in the form of cams activating pneumatic valves or relays controlling solenoid valves, directed the robot's actions. Initially, these robots were limited to simple tasks like loading, unloading, or basic material handling operations. However, a significant advancement occurred in 1954 when George Devol conceived the "Programmable Article Transfer," laying the groundwork for Unimate, recognized as the first true industrial robot [14]. Unimate, powered hydraulically, was first deployed at a General Motors factory in Trenton, New Jersey, to extract parts from a die-casting machine. Despite its complexity in reprogramming, it found use mainly in automotive factories for spot-welding and workpiece handling [13]. Figure (2.1) depicts an image of the Unimate Robot.

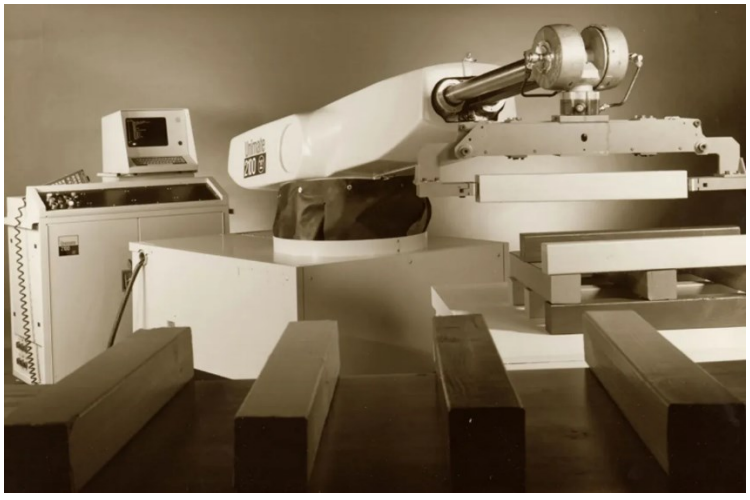


Figure (2.1) shows an image for the UNIMATE Robot.

2.2.2 The second Industrial Robotic Generation (1968-1977)

The second generation of industrial robots, spanning from 1968 to 1977, represented a significant advancement in robotics technology. These robots were programmable and capable of both point-to-point and continuous motion, facilitated by servo-controllers and either microprocessors or Programmable Logic Controllers (PLCs) for control purposes [12]. However, despite their improved capabilities compared to earlier models, their versatility remained limited due to the bespoke nature of their application-specific software. Diagnostic capabilities were also constrained, offering only rudimentary failure reports [15].

The transition from hydraulic to electric actuators in the 1970s, driven by technological advancements and economic factors such as rising oil prices, accelerated the development and adoption of electrically driven robots. Victor Scheinman's Stanford Arm (figure (2.2)) and later Vicarm played crucial roles in this shift, with Vicarm's design influencing the renowned PUMA (figure (2.3)) robot produced by Unimation in 1978 [16].

Figure (2.2) the Standford arm.

Figure (2.3) shows the PUMA Robot.

Moreover, various companies globally contributed to the evolution of industrial robotics during this period. KUKA introduced the Famulus robot in 1973 [4], while Cincinnati Milacron developed the T3, the first commercially available minicomputer-controlled industrial robot. One year after, Hitachi made significant strides with the HI-T-HAND Expert, renowned for its precision in insertion operations and advanced control systems. In the same year 1974 ASEA (now ABB) launched the IRB series in 1974, starting with the IRB-6 (figure 2.4), recognized for its versatility in complex tasks such as machining and arc-welding [14].

Figure (2.4) IRB-6 developed by ASEA (ABB now).

2.2.3 The Third Industrial Robotic Generation (1978-1999)

The third generation of industrial robots, which were typically manufactured between 1978 and 1999, were distinguished by a greater degree of interaction with the environment and the operator via sophisticated interfaces (such vision or voice). Additionally, they have a certain degree of self-programming ability, allowing them to slightly alter their programming to perform various jobs. With the help of servo controls, these robots were able to move continuously or from point to point while performing intricate jobs. By connecting to a PLC or PC, they could be programmed both online and offline. This allowed for the use of a high-level language for programming motion and allowed the robots to be interfaced with databases or CADs [12]. The ability to programme the robots at a high level offline expanded their operating possibilities. Furthermore, the diagnostic capabilities might be much improved: these robots could generate a report on the nature and location of the failure in addition to an indication of failure detection. Furthermore, the third generation of robots possessed a certain level of “intelligence” along with some degree of limited adaptive capability. These abilities could be used in some more difficult tasks, where the data from vision or perception systems could be used to locate the workpieces and objects and guide joint movements in accordance with the task at hand, accounting for the possibility of slight adjustments to the objects' positions [14].

The spread of robots was aided by additional scientific and technological advancements throughout the period between the end of the 1970s and the start of the 1980s.

In 1978, a breakthrough kinematic structure was introduced by the Japanese scientist Hiroshi Makino from Yamanashi University, this structure was made of three revolute joints and one prismatic joint lying at the end of kinematic chain, this manipulator was named SCARA (figure 2.5) an acronym from “Selective Compliance Assembly Robot Arm” and since that time it’s employed in tasks such as the assembly of small objects [17].

The development of direct drive actuated robots was a significant technological advancement in industrial robotics. The CMU Direct Drive Arm was the first prototype of this type, created in 1981 at Carnegie Mellon University (Pittsburgh, USA) by Asaka and Kanade. Because there was no need for an intermediate gear or chain system because the motors were directly attached to the arms, this type of robot operated more quickly and with more accuracy [18].

The AdeptOne (Figure 2.6), the first commercially produced direct-driven SCARA robot (1984), incorporated both previously described findings [19].

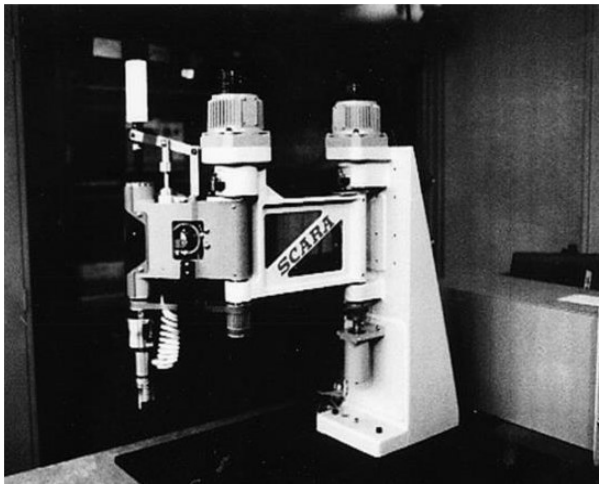


Figure (2.5) First SCARA Robot by Hiroshi Makino.

Figure (2.6) Direct-Driven SCARA robot.

Even with the tremendous advancements made in the 1980s, scientific research was driven to create novel kinematic structures by the demand for robots that could perform tasks quickly. The concept of using parallel kinematic chains in place of traditional serial kinematic chains was proposed, and the result was a class of lightweight dynamic robot. The Delta robot, created in 1992 by Swiss scientist Reymond Clavel at the Ecole Polytechnique Fédérale de Lausanne (EPFL), served as the prototype for this type of robot. Three translational DOFs and one rotating DOF were present in this kind of robot that Clavel constructed for his PhD thesis. Parallel robots could operate at a far higher speed than serial robots, but they had a narrower work area. For high-speed pick-and-place tasks, numerous parallel manipulators have imitated the kinematic architecture of the Delta robot [20].

Six Delta robots were working inside a work cell to put pretzels into trays when the Swiss company Demaurex created the first use for them in 1992 (Figure 2.7). Based on the Delta robot's structure, ABB created the Flex-Picker, the fastest picking robot in the world, a few years later (1998) (Figure 2.8) [21]

Figure (2.7) Six Delta in Swiss Company Demaurex's Production Line.

Figure (2.8) Fastest Picking Robot in 1998 ABB Flex-picker.

2.2.4 The Fourth Industrial Robotic Generation (2000-today)

Conventionally, the third generation terminates with the end of the century; industrial robots are regarded as part of the fourth generation, which began in 2000 and continues to this day. These robots possess high-level "intelligent" qualities (e.g., deep learning, logical reasoning, complicated strategy, advanced computation, cooperative behaviour) [15].

2.3 Structure of Industrial Robots

It is worthwhile to present the mechanical components of industrial robots in order to advance our understanding of them. This is an important step since it helps in the process of choosing and modelling the industrial robot that is the subject of our study. Additionally, several academic works [22] categorise industrial robots based on their mechanical structure because this approach offers a framework for comprehending the numerous robot kinds and their applicability for diverse activities, as well as assistance in addressing issues that are common to all classifications.

An industrial robot is composed by the following elements (figure 2.9):

- Links: rigid body connected to one or more rigid bodies [23].
- Joints: mechanical part that connects two rigid bodies and enables constrained relative motion between them [23] and include the actuators, gearings, and sensors.
- A Power supply
- A Control unit

- An end-effector: device specifically designed for attachment to the mechanical interface to enable the robot to perform its task [23].

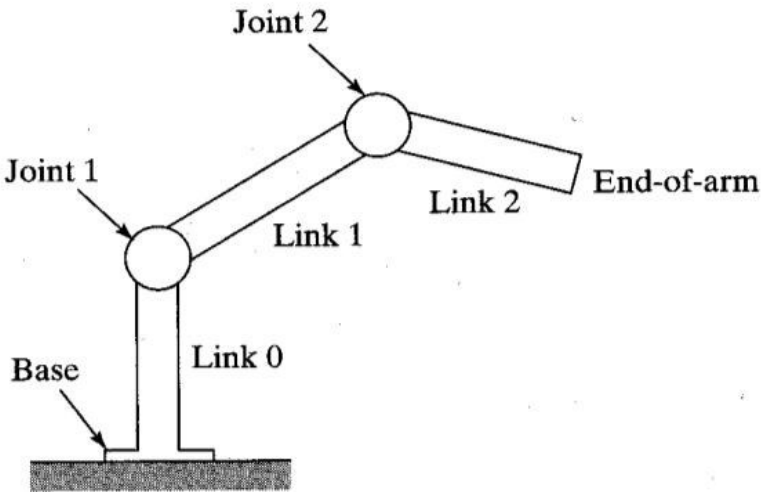


Figure (2.9) Mechanical Structure of a Robotic Arm.

2.4 ABB IRB 140 ROBOT

The ABB COMPANY developed the 6-axis ABB IRB 140 industrial robot, which has a 6 kg payload and is intended primarily for manufacturing companies that use flexible robot-based automation. The robot can communicate extensively with external systems and has an open structure specifically designed for flexible use [24]. The ABB IRB-140 robot is a compact, industrial manipulator with six degrees of freedom (DOF). Because of its robust build, it may be positioned at any angle on the wall, the floor, or an inverted roof [25]. Moreover, it features integrated cabling, which gives it some flexibility and makes it suitable and simple to integrate into any robotic process.

In the past, a few researchers used this robot to carry out certain specialist duties like:

- high precision drilling operations, which make use of the ABB IRB-140 robot arm's 6 DOF [26]

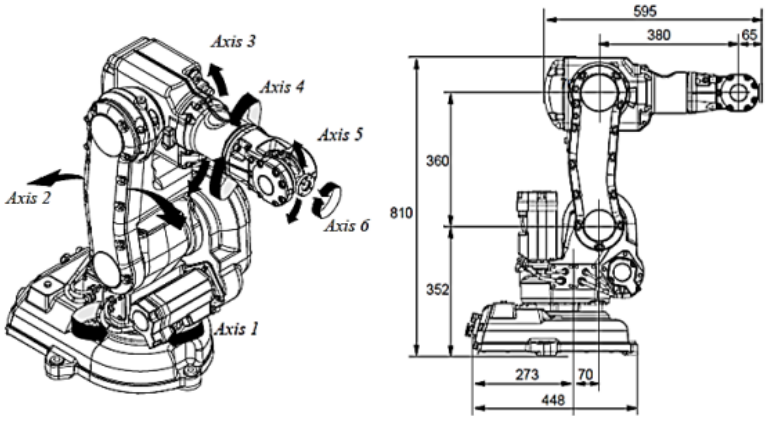
- adaptive multimodal applications, which give this robot artificial vision and speech control via sophisticated word recognition algorithms [27]
- and the design of a 3D print head that works with the ABB IRB 140 robotic arm in order to create a device that can print [28] larger objects than those produced by conventional 3D printers.

2.4.1 Technical Specifications

The product specification, which is released by ABB PRODUCTS, is where all technical details, CAD models, and videos of the ABB IRB-140 robot can be obtained [24].

The two figure below are intended to present the 3D-Model of ABB-IRB-140 (2.10) and the technical drawings showing the axis and the dimensions of the same robot (2.11), respectively.

Figure (2.10) The 3D-Model of ABB-IRB-140.



(2.11) Axis and Dimensions of ABB-IRB-140.

2.5 Robots Modelling

Robot modelling is the process of simulating or representing robots mathematically in order to comprehend their kinematics, dynamics, and behaviour. It includes many different things, like kinematic modelling, dynamic modelling, geometric modelling control modelling, sensor modelling, and environment modelling. These models allow engineers to predict and enhance the performance of robotic systems in a variety of jobs and settings by providing a framework for analysis, design, and optimisation.

Robots modelling involves studying the localization of objects in three dimensions, including the manipulator's linkages, parts, tools, and surrounding objects. Understanding the mechanical components helps determine which sections require modelling to achieve specific objectives. Key characteristics for modelling include position and orientation. Position refers to an object's location relative to a chosen reference point or coordinate system (is also called frame figure 2.12), often represented using Cartesian coordinates (x, y, z). Orientation describes an object's direction or alignment relative to a reference frame, commonly represented using rotation matrices (a mathematical representation ($_{i+1}^{i}R$) used to describe the orientation of an object in three-dimensional space. It's a square matrix (3x3) that represents a rotation transformation). Numerical representation and management of these properties are crucial research topics. Typically, a coordinate system is attached to an object, allowing its position and orientation to be defined relative to a reference coordinate system [2].

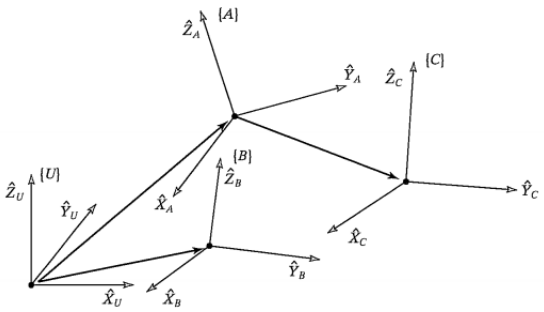


Figure (2.12) Several reference Frames [2].

Translation and rotation are integrated into a larger idea known as transformation to represent the location and orientation of an item or coordinate frame in relation to another. Changes in locations, orientations, or sizes in space are described by this phrase, which also includes additional processes. The transformation matrix, sometimes referred to as the homogeneous transformation matrix, is the mathematical embodiment of this idea. Robotics and computer graphics use this 4x4 matrix to express changes in three-dimensional space [2]. (Equation 2-1).

$$ {}^{i}_{i+1}\boldsymbol{T} = \left[\begin{array}{c|c} {}^{i}_{i+1}\boldsymbol{R} & {}^{i}\boldsymbol{P}_{i+1} \\ \hline \mathbf{0\ 0\ 0} & \mathbf{1} \end{array}\right] \qquad \textbf{(2.1)} $$

${}^{i}_{i+1}\boldsymbol{T}$ is the homogeneous transformation matrix that relates $\boldsymbol{i+1}$ to $\boldsymbol{i}$, ${}^{i}_{i+1}\boldsymbol{R}$ is the rotation matrix that describes the rotation of $\boldsymbol{i+1}$ in respect to $\boldsymbol{i}$, and finally ${}^{i}\boldsymbol{P}_{i+1}$ locates $\boldsymbol{i+1}$ origin to $\boldsymbol{i}$.

2.6 Robots' Kinematics

When robot kinematics is addressed in this thesis work, it refers to positional kinematics (forward and inverse kinematics) and differential kinematics. The distinction between positional kinematics and differential kinematics has significance. Positional kinematics pertains to the analytical link that exists between the end-effector position and orientation and joint positions in a robot manipulator. While differential kinematics uses the manipulator Jacobian to explain the analytical relationship in terms of velocities between the joint motion and the end-effector motion [29]. In general, the study of motion without consideration for the forces causing it is known as kinematics. Position, velocity, acceleration, and all higher order derivatives of the position variables (with respect to time or any other variable(s)) are studied within the field of kinematics science. Because of this, the study of manipulator kinematics encompasses all geometrical and temporal aspects of motion [2].

2.6.1 Direct Kinematics (D.K):

The direct or forward kinematic's purpose is to determine the tool frame's orientation and location in relation to the base frame. This is sometimes conceptualised as converting the manipulator position representation from a joint space description into a Description of Cartesian space [2]. In this manner, any robot might be described by first determining the values of four quantities for each link; two of these quantities describe the link directly, while the other two explain the link's relationship to a neighbouring link [2]. The following four variables are (figure 2.13):

- **Link length (a_{i-1}):** is the length of the common perpendicular between the Z_{i-1} and Z_i axes, measured along the X_{i-1}. This parameter represents the distance along the previous joint's axis to the intersection point of the common perpendicular.
- **Link twist (d_i):** is the displacement along the Z_{i-1} axis to align the origin of frame **I** with the Z_{i-1} axis. This parameter represents the distance along the previous joint's axis to the origin of frame i.
- **Joint angle (Θ_i):** is the angle about the Z_{i-1} axis required to align the $X_{i-1}$and X_i axes. This parameter represents the rotation about the previous joint's axis to align the coordinate frames of consecutive links.
- **Joint Twist (α_i):** is the angle about the $\mathbf{X_i}$ axis required to align the Z_{i-1} and Z_i axes. This parameter represents the rotation about the common perpendicular between consecutive links to align the Z_{i-1} and Z_i axes.

The convention that is using the above-mentioned parameters for direct (forward) kinematics to describe the geometric relationship between the consecutive links in any manipulator is called Denavit-Hartenberg (DH) convention [30]. This convention is used widely because it's simple and has a standardized method of representing geometric relationships between robot links and joints.

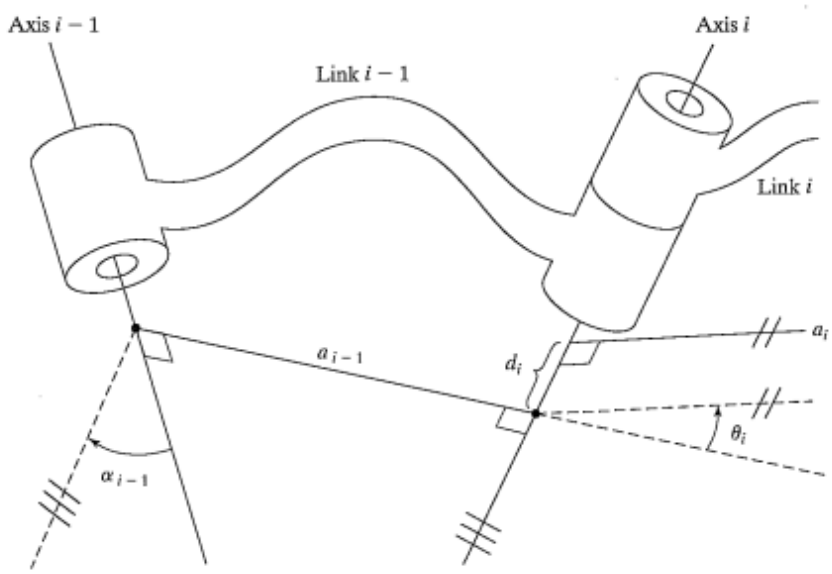


Figure (2.13) the Denavit-Hartenberg (DH) parameters.

After defining the four parameters for each link, the transformations matrices could be obtained by applying equation (2.2), [2]:

$$ {}_{i}^{i-1}T = Rot(X_{i-1}, \alpha_{i-1}) \; x \; Trans(X_{i-1}, \alpha_{i-1}) \; x \; Rot(Z_i, \Theta_i) \; x \; Trans(0,0, d_i) \qquad (2.2) $$

The result of this products should be the following the homogenous transformation matrix (equation (2-3)) Where S means sin$e$ and C means cosine:

$$ {}_{i}^{i-1}T = \begin{bmatrix} C\Theta_i & -S\Theta_i & 0 & a_{i-1} \\ S\Theta_i\, C\alpha_{i-1} & C\Theta_i\, C\alpha_{i-1} & -S\alpha_{i-1} & -d_i S\alpha_{i-1} \\ S\Theta_i\, S\alpha_{i-1} & C\Theta_i\, S\alpha_{i-1} & C\alpha_{i-1} & d_i C\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (2.3) $$

The forward kinematics were derived from the majority of earlier research using the same convention, for instance,

- Juan S. Toquica and Jose Mauricio S.T. Motta's work from 2021 proposed a calibration methodology for industrial robots [31].
- In addition, a kinematics analysis and trajectory planning of an Industrial robotic manipulator (IRMs) based on the hybrid optimisation algorithms using

the same DH convention was developed by Gurjeet Singh and Vijay Kumar Banga [37] in 2022.

2.6.2 Inverse Kinematics (I.K):

It requires the determination of every set of joint angles that could be used to achieve the specified position and orientation of the manipulator's end-effector given its location and orientation [2].

In human and other biological systems, millions of times a day, this relatively complex geometrical issue is handled. For an artificial system such as a robot, it’s needed to develop an algorithm within the control computer to perform this computation. The answer to this issue is, in a sense, the most crucial component of a manipulator system. This problem can be conceptualised as a mapping from "locations" in 3-dimensional Cartesian space to "locations" in the internal joint space of the robot [2].

The following factors make the inverse kinematics problem significantly more difficult to solve [29]:

- Since the equations involved are typically nonlinear, a closed-form solution is not always attainable.
- There might be more than one answer figure (2.13).

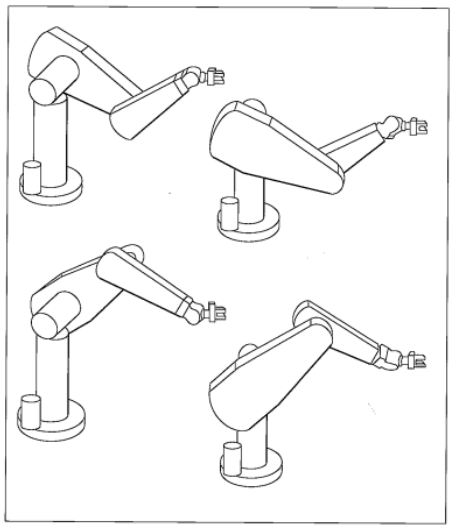


Figure (2.14) Four Different Solutions [2].

- There might not be any acceptable solutions given the kinematic structure of the manipulator.
- There might be infinite solutions, such as in the case of a kinematically redundant manipulator.

One could wonder why engineers, scientists, and other professionals are interested in creating inverse kinematics models after taking into account the aforementioned difficulties. One explanation would be that it's employed in robotics for robot programming, motion planning, and control, which enables robots to arrive at particular locations in space. Additionally, it is essential for activities like trajectory generation, obstacle avoidance, and path planning, which allow for precise control and manipulation of robotic arms.

The solution of each robot depends mainly on its structure. There are two main categories of solutions for inverse kinematics problems: closed-form solutions and numerical solutions. Since numerical solutions are iterative in nature, they are significantly slower. Afterwards, there is a further division of the closed-form solutions into geometric (which utilise geometric reasoning to discover joint angles) and algebric solutions (which solve equations derived from the robot's kinematic restrictions). Eventually, Pieper presents a solution [32] that simplifies the issue, particularly in cases when the robot's three successive axes intersect at a single location. This is known as the spherical wrist. Most industrial robots that are sold commercially can be used with the support of Pieper's method.

2.6.3 Differential Kinematics:

As mentioned earlier, Differential kinematics establishes the link between joint velocities and the corresponding linear and angular velocities of the end-effector. This connection is defined by a matrix known as the geometric Jacobian (equation 2-4), which relies on the configuration of the manipulator. Alternatively, if the end-effector's position is represented using a minimal form in operational space, the Jacobian matrix can be computed by differentiating the direct kinematics function with respect to the joint variables. This resulting Jacobian, known as the analytical

Jacobian, generally differs from the geometric one. The Jacobian serves as a pivotal tool for characterizing manipulators, aiding in identifying singularities, analysing redundancy, determining inverse kinematics algorithms, describing the relationship between forces applied to the end-effector and resulting torques at the joints (statics), as well as deriving dynamic equations of motion and designing operational space control strategies [29].

$$J = \begin{bmatrix} \widehat{Z}_i \, x \, (p - p_i) \ldots & \widehat{Z}_j \\ \hline \widehat{Z}_i \quad \ldots & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \end{bmatrix} \qquad \textbf{(2.4)}$$

Revolute Prismatic

Generally, the jacobian matrix has up to 6 rows and columns equal to the number of joints of the manipulator. It's a square matrix that could be used directly to find cartesian velocities (equation 2-5) and in the inverse form to find joint rates (equation 2-6) if the position is not at singularity [29].

$$\dot{X} = J \, \dot{Q} \qquad \textbf{(2.5)}$$

$$\dot{Q} = J^{-1} \, \dot{X} \qquad \textbf{(2.6)}$$

Where $\dot{X}$ is the cartesian velocity, J is the jacobian matrix, and $\dot{Q}$ is the joints rates.

An alternative approach to determining the Jacobian matrix is known as the screw-based method. This method relies on the concept of twists, which are vectors representing both rotational and translational motion in a rigid body. Each joint of the robot corresponds to a twist, and their combination forms a twist matrix, capturing the overall motion of the robot. From this twist matrix, the Jacobian matrix is derived, illustrating the relationship between joint velocities and end-effector velocity. It

elucidates how changes in joint positions impact the speed of the end-effector and vice versa [33].

This methodology has been utilized in previous studies, including "Kinematics, Dynamics, and Evaluation of Energy Consumption for ABB IRB 140 Serial Robots in the Tracking of a Path" by Maruo Baquero-Suarez in 2013 [25], and "Screw-Based Relative Jacobian for Manipulators Cooperating in a Task" by Luiz Ribeiro, Raul Guenther, and Daniel Martins in 2008 [34].

2.7 Motion Planning and Trajectory Generation

A manipulator's fundamental requirement is transitioning from its starting position to a predetermined final one while adhering to motion laws to ensure smooth movement without exceeding force limits or exciting structural resonances. Planning algorithms are crucial for generating these smooth trajectories.[29]

To clarify terminology, it's important to distinguish between a path and a trajectory. A path denotes the points in joint or operational space that the manipulator must traverse during motion, serving as a geometric representation. In contrast, a trajectory is a path with a specified timing law, typically incorporating velocities and/or accelerations at each point.[29]

In robotics, path planning and trajectory generation are vital for enabling effective and autonomous navigation and manipulation. These techniques facilitate tasks such as motion control, navigation, and obstacle avoidance, enhancing efficiency and safety. Viewing manipulator motions as tool frame motions relative to the station frame offers advantages, such as modularity for different manipulators or tool sizes and the ability to plan motions relative to moving workstations by updating the station frame definition during runtime.

Moreover, the primary goal is to transition the manipulator from its initial position to a desired final one, involving changes in both orientation and position of the tool relative to the station. Detailed motion specifications may require specifying intermediate via points between the initial and final positions, representing tool

position and orientation relative to the station. Temporal attributes, like elapsed time between via points, can also be specified [2].

In the past years, scientists have investigated a variety of techniques to determine the via points, such as employing splines or cubic polynomials, high order polynomials, linear functions with parabolic blends, and other parametric equations.

For instance, linear interpolation is used to move from the present joint position to the final position when linear functions with parabolic blends are taken into account, as shown in Figure 2.14. It should be emphasised that while every joint in this system moves linearly, the end-effector's movement in space is typically nonlinear. At each path point, we add a parabolic blend zone to the linear function to produce a smooth path with continuous position and velocity. The exact case could be applied when the user defines an arbitrary number of middle points (waypoints or via-points) as shown in figure 2.15. [2].

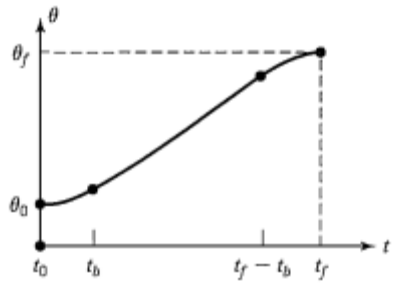


Figure (2.15) Linear Path with Parabolic Blends.

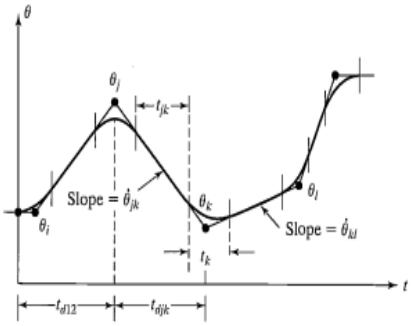


Figure (2.16) Multisegment Linear Path with Parabolic Blends.

A series of equations (equations 2.7–2.10) that can be found in Carig's book [2] need be carefully applied in order to construct an algorithm that can apply the previously indicated method.

The first 2.7 ought to be utilised for the first segment:

$$\frac{\boldsymbol{\Theta_2-\Theta_1}}{\boldsymbol{t_{12}-\frac{1}{2}t_1}} = \boldsymbol{\ddot{\Theta}_1 t_1} \qquad \textbf{(2.7)}$$

The set of 2.8 equations is used to find $\boldsymbol{\dot{\Theta}_1}$ and $\boldsymbol{t_{12}}$

$$\boldsymbol{\ddot{\Theta}_1 = SGN\,(\Theta_2 - \Theta_1)\left|\ddot{\Theta}_1\right|}$$

$$\boldsymbol{t_1 = t_{d12} - \sqrt{t_{d12}^2 - \frac{2(\Theta_2-\Theta_1)}{\ddot{\Theta}_1}}}$$

$$\boldsymbol{\dot{\Theta}_1 = \frac{\Theta_2-\Theta_1}{t_{d12}-\frac{1}{2}t_1}} \qquad \textbf{(2.8)}$$

$$\boldsymbol{t_{12} = t_{d12} - t_1 - \frac{1}{2}t_2}$$

For the last segment 2.9 is used:

$$\frac{\boldsymbol{\Theta_{n-1}-\Theta_n}}{\boldsymbol{t_{d(n-1)n}-\frac{1}{2}t_n}} = \boldsymbol{\ddot{\Theta}_n t_n} \qquad \textbf{(2.9)}$$

Which guides to the solution of 2.10:

$$\boldsymbol{\ddot{\Theta}_n = SGN\,(\Theta_{n-1} - \Theta_n)\left|\ddot{\Theta}_n\right|}$$

$$\boldsymbol{t_n = t_{d(n-1)n} - \sqrt{t_{d(n-1)n}^2 + \frac{2(\Theta_n-\Theta_{n-1})}{\ddot{\Theta}_n}}}$$

$$\boldsymbol{\dot{\Theta}_{(n-1)n} = \frac{\Theta_n-\Theta_{n-1}}{t_{d(n-1)n}-\frac{1}{2}t_n}} \qquad \textbf{(2.10)}$$

$$\boldsymbol{(t_{(n-1)n} = t_{d(n-1)n} - t_n - \frac{1}{2}t_{n-1}}$$

Typically, the initial point, final point, and via-points are provided in Cartesian space as 3-dimensional locations (X, Y, Z). The equations mentioned above can be utilized to determine the necessary values for displacements, velocities, and accelerations for each of the X, Y, and Z coordinates. Alongside the configuration of points, an essential input is the duration for each segment. Occasionally, the user may specify the maximum acceleration that dictates the motion, while at other times, it is left to the algorithm to decide based on defined constraints. Applying this kind of algorithm may result in energy-efficient trajectories.

The same methodology was employed by Ghulam Farid and colleagues in their study titled "Computationally Efficient Algorithm to Generate a Waypoints-Based Trajectory for a Quadrotor UAV," where they discovered that linear interpolation is computationally faster compared to alternative methods [35]. As previously mentioned, various other approaches could yield favourable results. For instance:

- In 2021, Y. Feng and colleagues utilized a 5th-order polynomial to construct joint trajectories in their paper titled "An Energy-Saving Optimization Method for Cyclic Pick-and-Place Tasks Based on Flexible Joint Configurations" [36].
- Gurjeet Singh and Vijay Kumar Banga employed multi-objective functions, including collision-free motion, jerk, travel time, and acceleration, for planning the trajectory of the robot manipulator [37].

Trajectory generation significantly impacts the energy consumption of robots. Efficient trajectory planning can minimize unnecessary movements, reduce accelerations and decelerations, and optimize path selection, ultimately leading to lower energy usage. Conversely, inefficient trajectory generation can result in increased energy consumption due to suboptimal movements and excessive acceleration/deceleration. Therefore, optimizing trajectory generation can contribute to significant energy savings in robotic systems.

2.8 Energy Modelling and Simulation

Taking into account the aforementioned factors, particularly the endeavour of modelling and simulating the energy usage of robots and manipulators divides into three main areas [38]:

- Creating a dynamic model that incorporates friction for mechanical energy dissipation.
- Assessing the power consumed by each motor.
- Estimating and simulating energy consumption.

Dynamics constitutes a vast area of research dedicated to analysing the forces necessary for inducing motion. To propel a manipulator from a standstill, maintain a constant end-effector velocity, and eventually bring it to a halt, a sophisticated series of torque functions must be exerted by the joint actuators. The specific characteristics of these required torque functions hinge upon the spatial and temporal attributes of the end-effector's path, as well as the mass properties of the links and payload, joint friction, and other relevant factors. One approach to guiding a manipulator along a desired trajectory involves determining these actuator torque functions through the dynamic equations of motion governing the manipulator [2].

Another application of the dynamic equations of motion lies in simulation. By reconfiguring these equations to compute acceleration based on actuator torque, it becomes feasible to simulate the motion of a manipulator under the influence of a specified set of actuator torques [2].

Prior to addressing the dynamics of robots, it is essential to depict the mass distribution of each link to consider the inertial characteristics of the robot. Manufacturers often lack precise knowledge of these inertial properties, yet various methods exist to address this issue. It is crucial to highlight that the inertia tensors for each link should be computed relative to the centre of mass of each link [39].

Upon establishing the Jacobian and inertial matrices, two primary methods emerge for solving a robot's dynamic equations concerning motion, forces, and torques: the

Newton-Euler method and the Lagrange-Euler method. The Newton-Euler method offers the advantage of being applicable to all robot links along the entire arm, providing an efficient means of determining the necessary forces and torques. Following the completion of the Newton-Euler iterations, the joint torque τ_i will have been calculated for all joints. Subsequently, the dynamic equation of motion can be expressed in the following equation [40]:

$$\boldsymbol{\tau = M\,(q)\ddot{q} + C\,(q\,,\dot{q}) + G(q)} \qquad \textbf{(2.11)}$$

Where $\boldsymbol{M\,(q)}$ is the mass matrix of the manipulator, $\boldsymbol{C\,(q\,,\dot{q})}$ + is a vector of centrifugal and Coriolis terms, and $\boldsymbol{G(q)}$ is a vector of gravity terms. $\boldsymbol{q}, \boldsymbol{\dot{q}}, \boldsymbol{\ddot{q}}$ are joints' positions, velocities, and accelerations respectively.

Finally, the dynamic model of the robot under investigation can be depicted utilizing the Lagrangian formalism, resulting in the below formulation [41]:

$$\boldsymbol{M\,(q)\ddot{q} + C\,(q\,,\dot{q}) + f_v\dot{q} + \ F_c\,sign(\dot{q}) = \ \tau_m} \qquad \textbf{(2.12)}$$

Where $\boldsymbol{f_v}$ is viscous friction coefficients, $\boldsymbol{F_c}$ is the Coulomb friction forces, and $\boldsymbol{\tau_m}$ is the motor torque.

As commonly practiced in the literatures, the process of determining motor power consumption, estimating energy usage, creating an electrical model, and simulating the robot's energy are typically conducted concurrently using the following set of equations [42]:

$$\boldsymbol{\tau_m(t) = K_t I(t)} \qquad \textbf{(2.13)}$$

$$\boldsymbol{V(t) = RI(t) + \ K_b\dot{q}_m(t)} \qquad \textbf{(2.14)}$$

$\boldsymbol{K_t}$ is the motor torque constant, $\boldsymbol{K_b}$ is the back-emf constants, $\boldsymbol{R}$ is the motor winding resistance, and $\boldsymbol{\dot{q}_m}$ is the motor speed.

The electric power consumption $\boldsymbol{W_m}$ equation is:

$$\boldsymbol{W_m(t) = V(t)^T I(t)} \qquad \textbf{(2.15)}$$

$\boldsymbol{W_m}$ represents the electric power drawn by each joint. The overall consumption of the robot is straightforwardly calculated by summing the individual consumptions of each of the six joint motors, as follows:

$$\boldsymbol{W_r(t) = \sum_{i=1}^{n} W_{m,i}(t)} \quad \textbf{(2.16)}$$

Where n is the number of joints under consideration.

Ultimately, the total energy consumption $\boldsymbol{E_{robot}}$ linked to a task defined within the time frame [ta, tb] can be calculated as follows:

$$\boldsymbol{E_{robot} = \int_{ta}^{tb} W_r(t)dt} \quad \textbf{(2.17)}$$

The energy model utilized in [38], [41], and [42] is more comprehensive compared to other energy models. For example, in [25], energy is solely represented as the total integration between the period (ta, tb) for the multiplication of motor torque and motor speed, without taking into account heat dissipations resulting from friction or electric power consumption.

CHAPTER THREE:

METHODOLOGY

3.1 Introduction

In this chapter, we delve into the process of simulating and constructing the energy model for industrial robots in thorough detail. The techniques outlined in chapter two will primarily be applied to the ABB-IRB-140 industrial robot. Before delving into any calculations or algorithm development, it is imperative to meticulously describe the kinematics and dynamics models of the robot being studied, laying a strong groundwork for subsequent analyses.

Given the potential complexity of the calculations involved, a fundamental assumption is initially made regarding which joints will be included in the study. These assumptions play a crucial role in simplifying the energy assessment process and ensuring the accuracy of the energy model for various applications.

Following this, we proceed to model and validate the direct kinematics (D.K) of the ABB-IRB-140 robot, followed directly by the inverse kinematics (I.K) model. Subsequently, the Jacobian matrix is determined.

A pivotal aspect of this thesis work involves developing the motion planning algorithm, which is discussed as the fifth point in this chapter. Designing a trajectory that optimizes energy usage positively impacts the overall energy consumption of the robot. The algorithm aims to be efficient while considering parameters essential for operator safety and task completion effectiveness.

Subsequently, three paths are selected and visually presented to apply the motion planning algorithm and calculate joint torques using Newton-Euler iterations. These paths include multiple points through which the end-effector must pass to accomplish specific tasks.

Following this, the dynamic model for each path, including the friction model, is developed to yield more realistic results using the Lagrangian formalism, resulting in motor torques. Subsequently, the electric power consumption for each motor and path is calculated individually. Finally, the energy model is derived by integrating the total power consumption over a specific time frame.

All the aforementioned steps are executed in MATLAB, primarily through the development of MATLAB functions, which are then consolidated into distinct script files for each selected path. Figure (3-1) illustrates the flowchart outlining the methodology employed in this thesis to simulate and construct energy models for industrial robots.

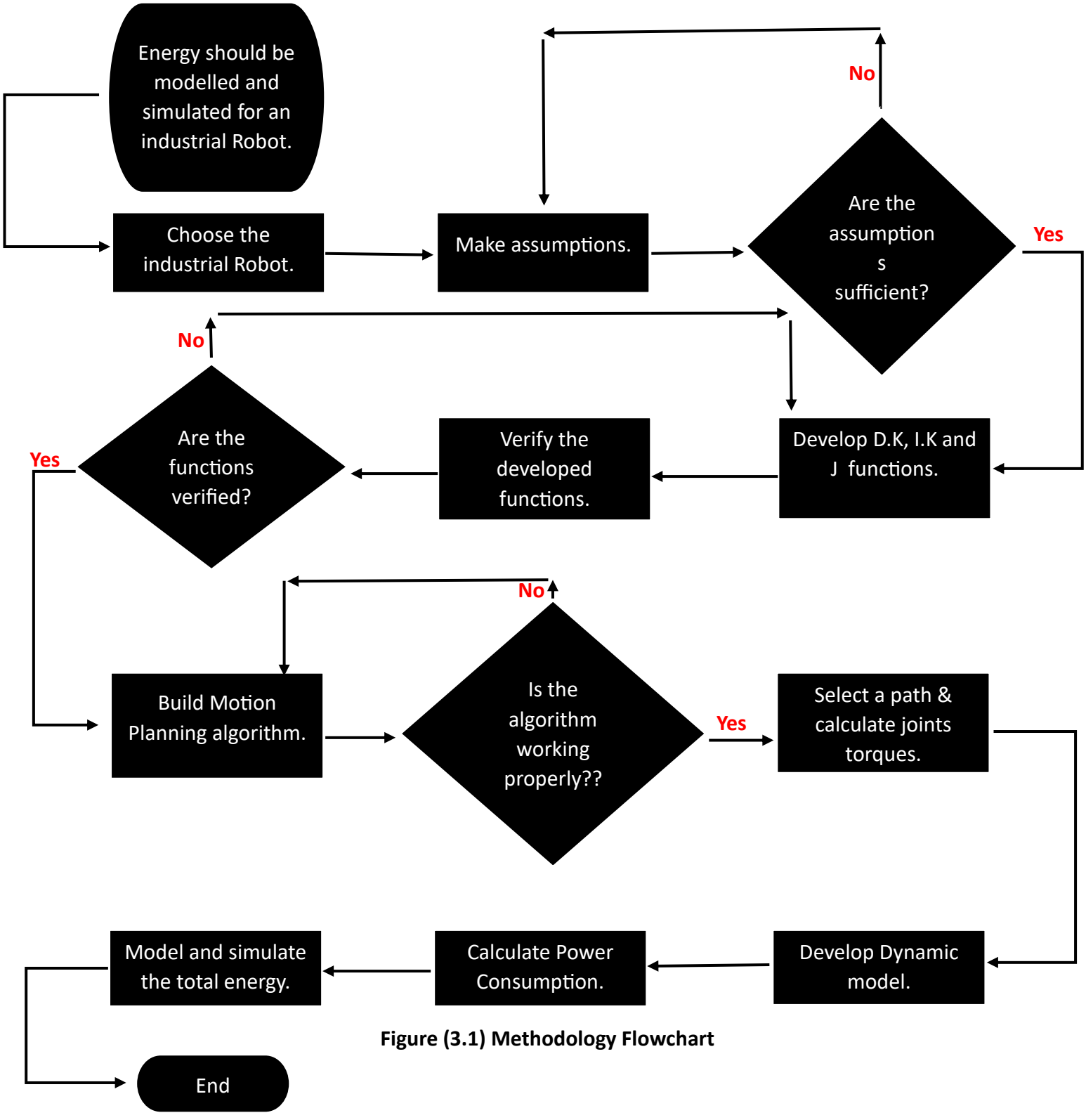


Figure (3.1) Methodology Flowchart

3.2 Assumptions

Assumptions play a fundamental role in applied science by aiding in the simplification of the problem under study and enabling researchers to focus on more practical aspects of their research.

In the case being investigated here, assumptions are made for similar reasons. The ABB-IRB-140 Industrial Robot consists of six joints (as shown in figure 3-2), with three of them intersecting at a single point known as the wrist centre. This centre allows for the manipulation of the robot's tip in a manner akin to a human wrist, with movements such as roll, pitch, and yaw. It is important to note that determining the position of the wrist centre (p_{xw}, p_{yw}, p_{zw}) can be calculated solely based on the lengths of the robot's links and the angles of its first three joints. These assumptions are crucial for simplifying both the direct and inverse kinematics calculations and, more importantly, significantly reducing the computational workload required for analysing robot dynamics, especially the iterative process needed for assessing joint torque. As a result, this study will focus exclusively on the first three links of the robot. This simplification is justified by the fact that there is not much moving mass beyond the wrist centre compared to the first three links, and any remaining mass beyond the wrist centre will be accounted for in the calculations for the centre of mass and inertia of the third link [43].

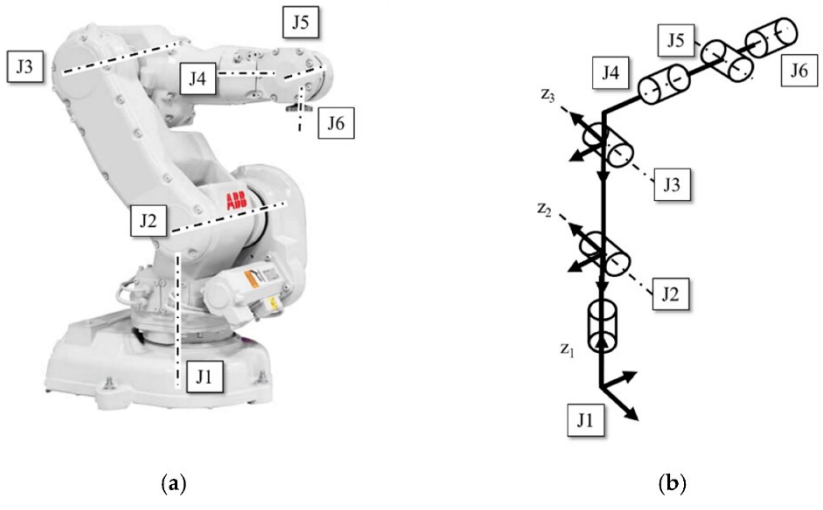


Figure (3.2) ABB-IRB-140 Axis and Links.

3.3 Direct Kinematics

As previously discussed in Chapter Two, direct or forward kinematics involves determining the position and orientation of the end-effector by considering the joint angles or lengths of the robot's links. The Denavit-Hartenburg (DH) convention is a widely used method for describing a robot in this manner. Table 3-1 below presents the four DH parameters for each link, corresponding to the configuration shown in Figure (3.2).

Table 3.1: Denavit-Hartenburg Parameters for the ABB-IRB-140 Robot

i	a_{i-1}	α_i	d_i	Θ_i
1	0	0	d1	Θ_1
2	π/2	a1	0	Θ_2
3	0	a2	0	Θ_3
4	π/2	0	d4	Θ_4
5	-π/2	0	0	Θ_5
6	π/2	0	0	Θ_6

a1 = 70 mm, a2 = 360 mm, d1 = 352 mm, d4 = 380 mm

Employing equation (2.2) iteratively for every row of table 3-1 facilitates the derivation of several transformation matrices. Subsequently, equation (2.3) is applied to multiply all of these matrices, yielding the calculation of T60, denoting the transformation matrix depicting both translation and rotation from the robot's base to the wrist centre. This process yields:

$$ {}_{6}^{0}\boldsymbol{T} = \begin{bmatrix} \boldsymbol{r_{11}} & \boldsymbol{r_{12}} & \boldsymbol{r_{13}} & \boldsymbol{p_{xw}} \\ \boldsymbol{r_{21}} & \boldsymbol{r_{22}} & \boldsymbol{r_{23}} & \boldsymbol{p_{yw}} \\ \boldsymbol{r_{31}} & \boldsymbol{r_{32}} & \boldsymbol{r_{33}} & \boldsymbol{p_{zw}} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} \end{bmatrix} $$

Where:

$$r_{11} = C_1[C_{23}(C_4C_5C_6 - S_4S_6) - S_{23}S_5C_6] + S_1(S_4C_5C_6 + C_4S_6)$$

$$r_{12} = C_1[-C_{23}C_4C_5C_6 + S_{23}S_5S_6] - S_1S_4C_5S_6$$

$$r_{13} = C_1[C_{23}C_4S_5 + S_{23}C_5] + S_1S_4S_5$$

$$r_{21} = S_1[C_{23}(C_4C_5C_6 - S_4S_6) - S_{23}S_5C_6] - C_1(S_4C_5C_6 + C_4S_6)$$

$$r_{22} = S_1[C_{23}C_4C_5C_6 + S_{23}S_5S_6] + C_1S_4C_5S_6$$

$$r_{23} = S_1[C_{23}C_4S_5 + S_{23}C_5] - C_1S_4S_5 \quad \textbf{(3.1)}$$

$$r_{31} = S_{23}(C_4C_5C_6 - S_4S_6) + C_{23}S_5C_6$$

$$r_{32} = -S_{23}C_4C_5C_6 - C_{23}S_5C_6$$

$$r_{33} = S_{23}C_4S_5 - C_{23}C_5$$

$$P_{xw} = C_1[S_{23}d_4 + C_2a_2 + a_1]$$

$$P_{yw} = S_1[S_{23}d_4 + C_2a_2 + a_1]$$

$$P_{zw} = -C_{23}d_4 + S_2a_2 + d_1$$

In order to elucidate the notation, let's define, $C_1 = \cos(\Theta_1)$ and $C_{12} = \cos(\Theta_1 + \Theta_2)$. Additionally, as outlined in the assumption section, our analysis only considers the initial three links. Consequently, the equation governing the wrist centre's position (P_{xw}, P_{yw}, P_{zw}) solely incorporates the joint positions from joints 1, 2, and 3. Should one aim to compute the absolute tip of the manipulator, it merely involves a translation of length d6 along the z-axis of frame 6 relative to the base frame 0:

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} p_{xw} + d_6r_{13} \\ p_{yw} + d_6r_{23} \\ p_{zw} + d_6r_{33} \end{bmatrix} \quad \textbf{(3.2)}$$

The outcome from this step is a MATLAB function called: ***dk_ABB_IRB_140.***

Where the input is a scalar vector (q = 6x1) contains the joints angles values in Radians and the outputs are; End-Effector position (P=1x3) and orientation (O = 3x3).

```
[P,O] = dk_ABB_IRB_140(q)
```

3.4 Inverse Kinematics

Chapter two elucidated that inverse kinematics stands in contrast to forward kinematics. While forward kinematics entails determining the end-effector's position and orientation based on the robot's joint angles or link lengths, inverse kinematics involves computing the joint angles necessary to achieve a desired position and orientation of the end-effector.

For a six-degree-of-freedom robot with a wrist, it's customary to divide the inverse kinematics conundrum into two parts: the first part, a geometrical solution, focuses on finding the joint angles corresponding to the wrist centre's position, while the second part involves an analytical solution to determine the angles corresponding to the wrist's orientation.

Referring to the frame assignment depicted in figure (3.3), the x and y components of frame {1} coincide with those of frame {0} due to the sole Z-directional disparity between the two frames. Consequently, the projection of the wrist components onto the x-y plane of frame {0} mirrors the components in frame {1}. Furthermore, given that both links two and three are planar, the position vector in the y-direction varies solely with respect to Θ_1. Thus, employing the arctangent function can yield two feasible solutions for Θ_1.

$$\boldsymbol{\theta_1 = \tan^{-1}\frac{p_{yw}}{p_{xw}} = \operatorname{atan2}(p_{yw}, p_{xw})} \quad \textbf{(3.3)}$$

Or

$$\boldsymbol{\theta_1 = \pi + \tan^{-1}\frac{p_{yw}}{p_{xw}} = \operatorname{atan2}(-p_{yw}, -p_{xw})} \quad \textbf{(3.4)}$$

The plane formed by links 2 and 3 is considered when seeking to find the second and third joint angles, as depicted in Figure 3.4.

$$\cos \Theta_3 = \frac{(p_{xw} + a_1 C_1)^2 + (p_{yw} + a_1 S_1)^2 + (p_{zw} - d_1)^2 - a_2^2 - d_4^2}{2 a_2 d_4} = CZ3 \quad \textbf{(3.5)}$$

The solution for Θ_3 will be obtained through the expression $\Theta_3 = \cos^{-1}(CZ3)$. Given that Θ_1 has two solutions, one corresponding to the elbow up and the other to the elbow down configuration, this will be represented by:

$$\Theta_3 = \tan^{-1}\left(\frac{\pm\sqrt{1-D^2}}{D}\right) = atan2(\pm\sqrt{1-D^2}, D^2) \quad \textbf{(3.6)}$$

In the same manner,

$$\Theta_2 = atan2\left(p_{zw} - d_1, \sqrt{(p_{xw} + a_1 C_1)^2 + (p_{yw} + a_1 S_1)^2}\right) - atan2(d_4 S_3, a_2 + d_4 C_3) \quad \textbf{(3.7)}$$

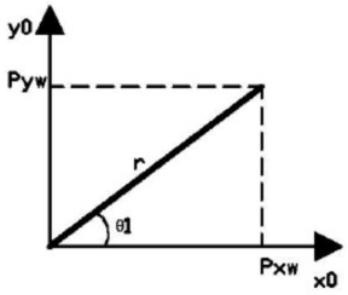


Figure (3.3): Projection of the wrist center onto the xy plane [43].

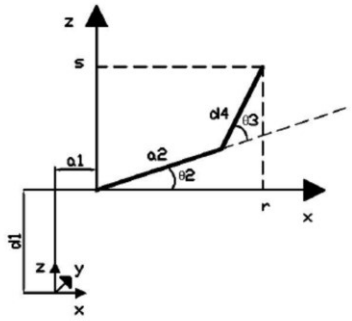


Figure (3.4): Projection onto the plane formed by links 2 and 3 [43].

The outcome from this step is a matlab function called: ***ik_ABB_IRB_140.***

Where the input for this function is the end-effector position matrix (W) and the output is a matrix containing all the possible joints angles configuration solutions (s) to achieve the inserted position.

```
[s] = ik_ABB_IRB_140(W,d6)
```

3.5 Jacobian Matrix

The responsibility for establishing a connection between joint velocities and the associated linear and angular velocities of the end-effector, as well as vice versa, lies with the Jacobian matrix. Its significance extends beyond these relationships, encompassing the determination of joint torques, which, in turn, aids in calculating the overall energy consumption of the robot.

One approach to obtaining the Jacobian matrix involves directly computing the partial derivatives over time of the equations describing Cartesian position as functions of joint angles (Equations 3.1).

$$\frac{\delta p_{xw}}{\delta t} = [-S_1(S_{23}d_4 + C_2a_2 + a_1) \quad C_1(C_{23}d_4 - S_2a_2) \quad C_1C_{23}d_4] \begin{bmatrix} \dot{\Theta}_1 \\ \dot{\Theta}_2 \\ \dot{\Theta}_3 \end{bmatrix} \tag{3.8}$$

$$\frac{\delta p_{yw}}{\delta t} = [C_1(S_{23}d_4 + C_2a_2 + a_1) \quad S_1(C_{23}d_4 - S_2a_2) \quad S_1C_{23}d_4] \begin{bmatrix} \dot{\Theta}_1 \\ \dot{\Theta}_2 \\ \dot{\Theta}_3 \end{bmatrix} \tag{3.9}$$

$$\frac{\delta p_{Zw}}{\delta t} = [0 \quad S_{23}d_4 + C_2a_2 \quad S_{23}d_4] \begin{bmatrix} \dot{\Theta}_1 \\ \dot{\Theta}_2 \\ \dot{\Theta}_3 \end{bmatrix} \tag{3.10}$$

Transforming these equations into matrix representation yields the Jacobian matrix (equation 2.4) for linear velocity:

$$J_{3x3} = \begin{bmatrix} -S_1(S_{23}d_4 + C_2a_2 + a_1) & C_1(C_{23}d_4 - S_2a_2) & C_1C_{23}d_4 \\ C_1(S_{23}d_4 + C_2a_2 + a_1) & S_1(C_{23}d_4 - S_2a_2) & S_1C_{23}d_4 \\ 0 & S_{23}d_4 + C_2a_2 & S_{23}d_4 \end{bmatrix} \tag{3.11}$$

Similarly, by completing another set of partial derivatives, one can determine the derivative of J, denoted as J_dot.

The outcome from this step is a MATLAB function called: ***j_ABB_IRB_140.***

Where the input is a scalar (q) containing the joints angle values and the output is the Jacobian matrix (J).

```
[J] = j_ABB_IRB_140(q)
```

3.6 Motion Planning

Motion planning represents the primary challenge within this thesis, as it dictates the trajectory points through which the manipulator must navigate, mirroring real-world scenarios where specific tasks necessitate precise point-to-point traversal.

The algorithm developed here employs a linear function methodology augmented with parabolic blending to facilitate path planning incorporating via points (waypoints). Beyond initial and final coordinates, users also input intermediary points through which the manipulator is intended to traverse. Additionally, users are required to specify the temporal duration for each task. The determination of acceleration can either be user-specified from the outset or left to the algorithm to calculate based on the available trajectory, considering maximum speed and acceleration thresholds for each task. Utilizing inverse kinematics and the Jacobian matrix, the algorithm converts Cartesian path points into corresponding joint angles.

Furthermore, the algorithm undertakes the responsibility of ensuring that all points along the path are realistically reachable, avoiding unattainable velocities or accelerations, and adhering to predefined constraints encompassing task completion, operator safety, and energy efficiency considerations. Lastly, the algorithm offers a straightforward approach for selecting among multiple inverse kinematics solutions.

The outcome from this step is a MATLAB function called: ***LFVP***, it stands for **L**inear **F**unction with parabolic blends for a path with **V**ia **P**oints.

The foundation of this algorithm stems from the series of equations (2.7-2.10) outlined in Chapter Two, sourced from Carig's publication [2]. Here are how these equations are implemented in MATLAB.

```
th = VP(:,j);
    % first: speeds thp - there are n-1 of them
    thpp(1) = sign(th(2)-th(1))*amax;
    bt(1) = h(1)-sqrt(h(1)^2-2*(th(2)-th(1))/thpp(1));
    thp(1) = (th(2)-th(1))/(h(1)-.5*bt(1));

    for k=2:n-2
        thp(k) = (th(k+1) -th(k))/h(k);
    end

    thpp(n) = sign(th(n-1)-th(n))*amax;
    bt(n) = h(n-1)-sqrt(h(n-1)^2+2*(th(n)-th(n-1))/thpp(n));
    thp(n-1) = (th(n)-th(n-1))/(h(n-1)-.5*bt(n));

    for k = 2:n-1
        thpp(k) = sign(thp(k)-thp(k-1))*amax;
    end

    for k=2:n-1
        bt(k) = (thp(k)-thp(k-1))/thpp(k);
    end

    bt = round(bt/(2*dt))*2*dt;

    if ~isreal(bt)
        Message = 'Constarints are not satisfied for Z, change either
acceleration or tasks duration time'
        quit(LFVP_PB)
    end
```

Where the inputs are Cartesian path points (th), tasks durations (td) and maximum acceleration (a, optional). In the other hand, the outputs are overall time duration (T) of the manipulator journey, joint angles positions (Q), speeds (V), and accelerations (A) for X, Y, & Z.

```
[Tx,Qx,Vx,Ax,Ty,Qy,Vy,Ay,Tz,Qz,Vz,Az,Message] = LFVP_PB(th,td,a)
```

3.7 Robot Dynamics and Path Tracking

The application of the Newton-Euler method (as per Equation 2.11) to robots with a high number of degrees of freedom poses a significant challenge. Therefore, in this study, simplification of this method for calculating joint torques is achieved through the assumptions outlined earlier in this chapter, aided by the MATLAB Robotics

System Toolbox. The function is used here is called: ***irb140.rne_mdh,*** the outcome from this function will be the joints torques values.

```
joint_torque = irb140.rne_mdh(q,q_dot,q_ddot,[0,0,-9.81],zeros(1,6))';
```

Subsequently, three specific paths are selected for examination within this thesis: the Via_Points Path (depicted in Figure 3.5), the Triangular Path (illustrated in Figure 3.6), and the Pick_and_Place Path (outlined in Figure 3.7).

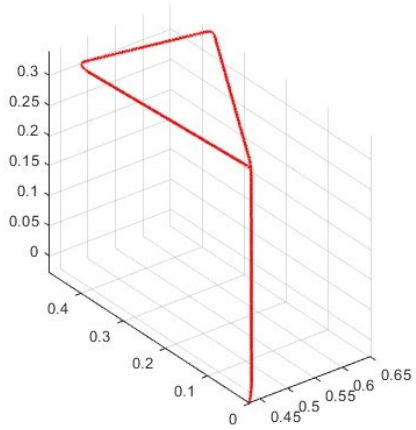


Figure (3.5): A Path with Three Via Points.

The primary objective of path tracking is to utilize the equations and expressions from preceding sections on kinematics and dynamics. This facilitates the computation of positions, velocities, accelerations, and resultant torques on the manipulator's joints, relative to an equivalent time corresponding to the arm's tracking velocity.

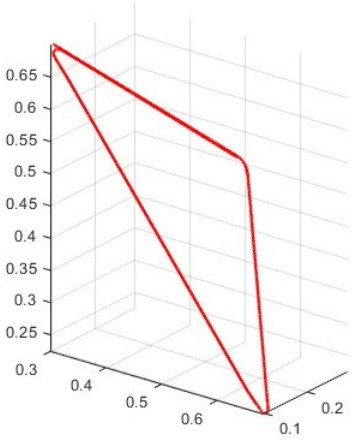


Figure (3.6): A Path with Triangular Shape.

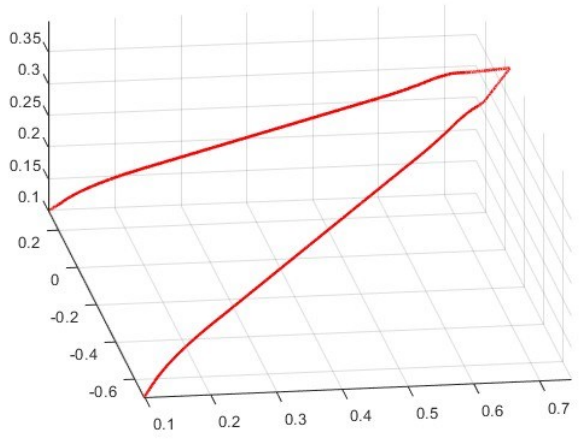


Figure (3.7): A Path with Pick and Place Task.

Following this, estimation of the energy consumed by the manipulator can be derived from the aforementioned data.

3.8 Energy Modelling and Simulation

The process of Modelling and simulating the energy of any robot involves three key steps:

- Creating a dynamic model with friction for mechanical energy dissipation.

Initially, a dynamic model is developed, accounting for friction to dissipate mechanical energy. To determine the motor speed and acceleration for each joint, the following equations are utilized:

$$Motor\ Speed = Joint\ Speed\ x\ Gear\ Ratio \quad \textbf{(3.12)}$$

$$Motor\ Acceleration = Joint\ Acceleration\ x\ Gear\ Ratio \quad \textbf{(3.13)}$$

Subsequently, the Lagrangian equation outlined in Chapter Two (Equation 2.12) is applied to establish the dynamic model, where ***Tm*** represents the joint torque divided by the gear ratio, ***Tc*** denotes the Coulomb friction torque, ***th_dot_m'*** signifies the motor speed, and ***Kv*** denotes the velocity constant.

```
Tm = joint_torque/GR + Tc*sign(th_dot_m') + Kv*th_dot_m;
```

- Assessing the power consumed by each motor.

The power consumed by each motor (joint) is evaluated by applying Equation (2.15), resulting in ***Wem*** (watts), which indicates the electric power absorbed by each motor. Equation 2.15 incorporates terms such as ***R*** (resistance), ***Kt*** (torque constant), and ***Tm*** (motor torque) to calculate ***Wem***.

```
Wem = (R/Kt^2)*Tm^2 + 0.9*Tm*th_dot_m';
```

- Estimating and simulating energy consumption.

Finally, to derive the overall energy model, Equation (2.17) is employed, integrating the power consumed by each motor over time using the following integration function:

```
E_tot = cumtrapz (t,Wem)
```

Here, ***cumtrapz*** computes the integral of ***Wem*** utilizing the trapezoidal method within the defined time frame ***t***.

These processes are executed for each path individually to determine their energy consumption, followed by plotting the results for analysis.

CHAPTER FOUR:

RESULTS AND DISCUSSIONS

4.1 Introduction

This chapter presents the results obtained from applying the methodology outlined in Chapter Three, followed by a comprehensive discussion.

Initially, given data pertinent to all paths (Via_Points, Triangular_Shape, Pick_and_Place) is introduced and displayed in a table. Subsequently, for each of the selected three paths, two tables are presented: one summarizing specific given data and the other summarizing numerical results for key parameters such as total elapsed time, maximum joint torque, maximum motor torques, maximum electrical power dissipation, and total energy consumption.

Following the tabulated data, graphical representations are provided for each case, including: the development of end-effector positions, speeds, and accelerations with time in (X,Y,Z); the development of the first three joint angles' positions, speeds, and accelerations with time; joints torque values; 3D graphical representation of the ABB-IRB-140 movement in the described path; the time evolution of the absorbed electrical power by each motor; and the time evolution of the energy consumed for each motor, representing the energy simulation.

These graphical representations serve to visually elucidate the performance and behaviour of the manipulator throughout the prescribed paths, aiding in the interpretation and analysis of the obtained results.

4.2 Data Given

The data provided in this section is categorized into two types: general data, which describes the robot itself, and specific data, which pertains to the assigned task or path.

The acquisition of this data primarily involves referencing manufacturers for information regarding the ABB-IRB-140 robot itself or its motors. Additionally, some data may be sourced from previous studies to ensure adherence to best practices.

Table 4.1 illustrates the shared data among the three paths:

i	Given Data (abbreviation-Units)	Link 1	Link 2	Link 3
1	Gravitational Acceleration (g-m^2/sec)	**9.81**	**9.81**	**9.81**
2	Gear Ratio (GR-nondimensional)	**100**	**100**	**100**
3	Coulomb friction forces (F_C-N.m)	**0.1**	**0.1**	**0.1**
4	Viscous friction coefficients (f_v- N.m.s/rad)	**0.5 x 10^{-3}**	**0.5 x 10^{-3}**	**0.5 x 10^{-3}**
5	Motor Torque Constant (K_t-N.m/A)	**0.63**	**0.97**	**2.3**
6	Electric Resistance (R-Ω)	**9**	**20**	**25**

All the data illustrated above are equally utilized across all paths; however, the values may vary for the individual links but remain consistent for the assigned task.

Table 4.2 presents the specific data tailored to each path:

i	Inserted Data (abbreviation-Units)	Via_Points	Triangular_Shape	Pick_and_Place
1	Cartesian Path Points (X,Y,Z) (th-m)	**[0.43, 0, -0.028; 0.6342, 0.2627, 0.1765; 0.6527, 0.3768, 0.3443; 0.4831, 0.4831, 0.3007; 0.6342, 0.2627, 0.1765]**	**[0.3, 0.1, 0.7; 0.5, 0.3, 0.5; 0.7, 0.1, 0.2; 0.3, 0.1, 0.7; 0.5, 0.3, 0.5]**	**[0.1, 0.3, 0.1; 0.7, 0.1, 0.35; 0.75, -0.05, 0.4 0.7, -0.08, 0.35; 0.1, -0.7, 0.1]**
2	Task Duration (t_D-seconds)	**[2, 2, 2, 2]**	**[2, 2, 2, 2]**	**[2, 2, 2, 2]**
3	Acceleration (a-m^2/second)	**0.5**	**0.9**	**0.5**

Precisely describing a path or task necessitates inserting Cartesian path points in a specific manner to closely replicate the desired trajectory. However, the task duration may vary depending on how the operator intends the manipulator to execute tasks, consequently affecting acceleration requirements. As previously mentioned, the developed Motion Planning algorithm offers flexibility for users to either input acceleration values manually or delegate this task to the robot, adhering to predefined constraints to ensure operator safety, task completion, and energy efficiency.

4.3 Numerical Results

Now is the time for the numerical results to be presented and discussed for each path. Specifically, attention will be given to five results, namely: the total elapsed time, maximum joint torques, maximum motor torques, maximum electrical power dissipation, and the total consumed energy. These numerical findings will offer insights into the performance and efficiency of the manipulator across various paths.

Table 4.3 displays the numerical results derived from modelling and simulating the energy for three distinct paths.

i	Numerical Results (abbreviation-Units)	Via_Points			Triangular_Shape			Pick_and_Place		
1	Joints	Joint one	Joint Two	Joint Three	Joint one	Joint Two	Joint Three	Joint one	Joint Two	Joint Three
2	Total Elapsed Time (t-seconds)	**34.7**			**34.12**			**40.44**		
3	Maximum Joint Torques (τ - N.m)	**3.08**	**-113.8**	**-15.17**	**2.62**	**-106.9**	**16.66**	**2.93**	**-115.3**	**-15.74**
4	Maximum Motor Torques (τM - N.m)	**-0.14**	**-1.22**	**-0.25**	**0.1252**	**-1.17**	**0.2681**	**-0.14**	**-1.26**	**-0.26**
5	Maximum Electrical Power Dissipation (Wem - Watt)	**1.52**	**34.98**	**0.7506**	**1.12**	**36.68**	**1.30**	**3.96**	**40.02**	**2.07**
6	Total Consumed Energy (E_tot - Joule)	**6.38**	**181.73**	**2.57**	**4.34**	**80.31**	**4.44**	**9.87**	**173.85**	**3.29**

Firstly, the total elapsed time represents the duration required for executing the MATLAB code dedicated to calculating the energy consumed in each path. As depicted in Table 4.3, this value is notably higher for the Pick_and_Place task, primarily due to the relatively larger distance travelled compared to other paths.

Next, the maximum joint torque consistently manifests as highest for the second joint, followed by the third and finally the first. This pattern aligns with the fact that the joint with higher mass exhibits greater inertia, necessitating higher torques during acceleration or deceleration. When considering the total maximum joint torques calculated for each path, the Pick_and_Place task exhibits the highest torque requirements, followed by the Via_Points path, and finally the Triangular_Shape path. This reflects the increased trajectory complexity experienced by the joints in the Pick_and_Place and Via_Points paths, particularly due to sudden changes in directions, which elevate dynamic loads on the joints, necessitating greater torque to overcome inertia and other forces.

Subsequently, the maximum motor torques exhibit a similar trend as the joint torques, with the second joint requiring the highest torque, followed by the third and then the first. This consistency underscores the linear relationship between joint torque and motor torque as expressed in Equation 2.12.

Moving on to the maximum electrical dissipation, the order varies slightly across different paths. For the Triangular_Path, the order mirrors that of joint torque, with the joint requiring the highest torque also dissipating the most electrical power. However, for the Pick_and_Place and Via_Points paths, the order shifts due to additional factors such as speed, acceleration profiles, and motor characteristics. Nevertheless, the Pick_and_Place task consistently exhibits higher electrical power dissipation overall.

Finally, the total energy consumed by each joint in each path follows the same order as the electrical power dissipation. This alignment is attributed to the fact that energy is the integration of power over a time frame or the area under the curve. Notably, the second joint consistently demonstrates the highest energy demand across all paths, primarily due to its responsibility for most of the robot's reaching

motion. Additionally, the high velocity of the second joint at points of high torque necessitates a significant amount of power, resulting in a larger energy requirement.

4.4 Graphical Results

Each time the MATLAB code is executed for one of the selected paths, it generates eleven figures representing various variables plotted against time. These include the cartesian end-effector positions, speeds, and accelerations; the first three joint angles' positions, speeds, and accelerations; joint torques; a 3D representation of the ABB_IRB_140 robot's movement in the selected path; motor torques; the electric power absorbed by each motor; and the total energy consumed by each motor.

The graphs for end-effector and joint angles' positions, speeds, and accelerations for the Via_Points, Triangular_Shape, and Pick_and_Place paths will be presented in Figures 4.1, 4.2, 4.3, 4.4, 4.5, and 4.6, respectively.

To generate the End-effector position graph, the cartesian positions resulting from the motion planning algorithm are combined into one matrix and plotted against time. Then, these values are differentiated once with respect to time to obtain speeds and twice to obtain accelerations.

Subsequently, applying inverse kinematics for the end-effector positions maps these values to joint angles' positions. Further, applying the Jacobian matrix yields joint angles' speeds, which, when differentiated with respect to time, provide joint angles' accelerations.

These graphs serve primarily for performance verification purposes, allowing comparison of expected motion profiles with actual motion profiles and ensuring that the robot joints are accelerating within specified limits. Additionally, they aid in the analysis of motion profiles by providing graphical insights into the kinematics and dynamics of the robot system.

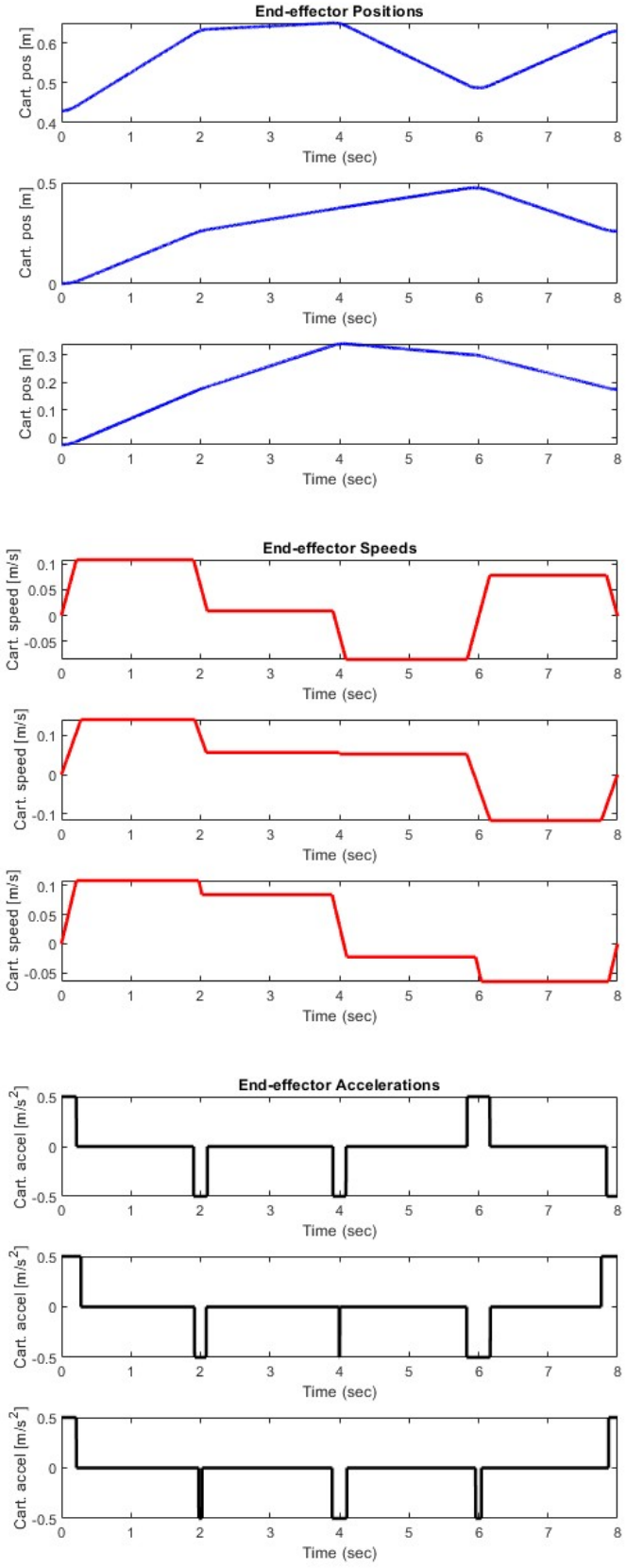


Figure (4.1): End-Effector Positions, Speeds, and Accelerations vs. Time for Via_Points Path.

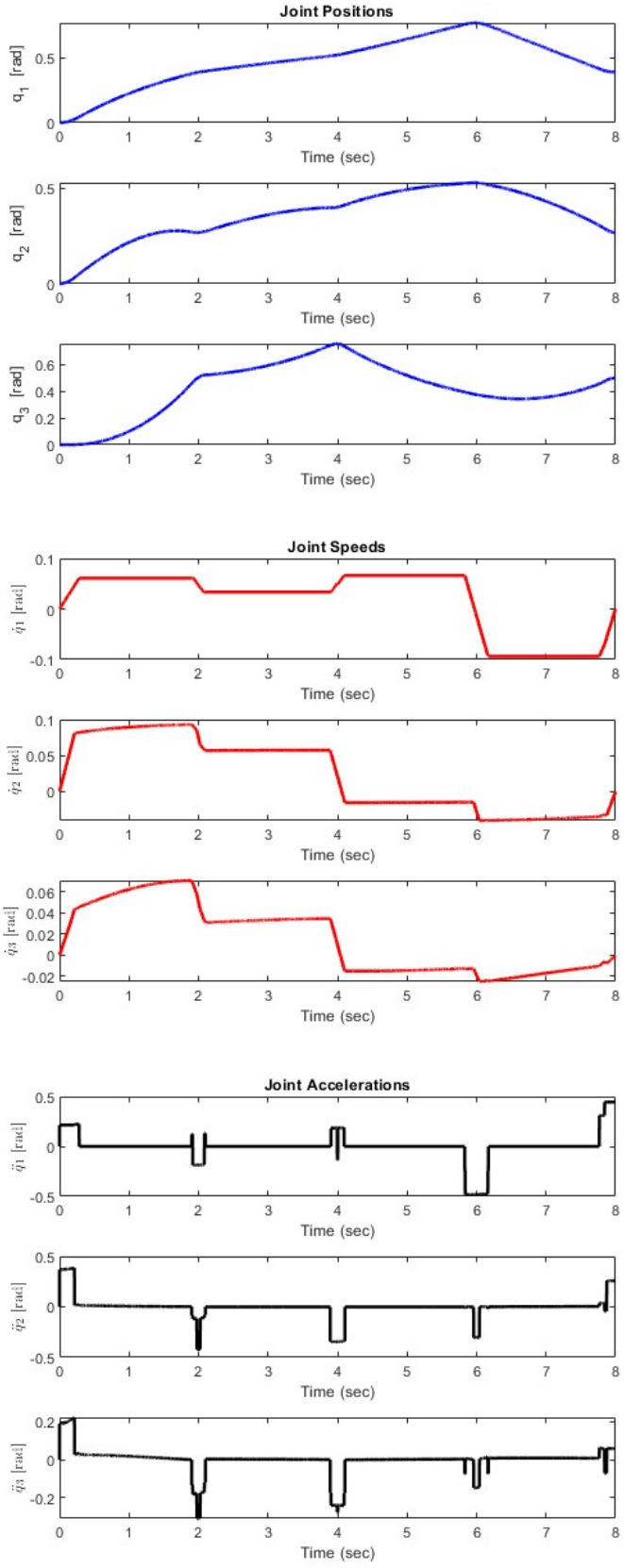


Figure (4.2): Joint Angles Positions, Speeds, and Accelerations vs. Time for Via_Points Path.

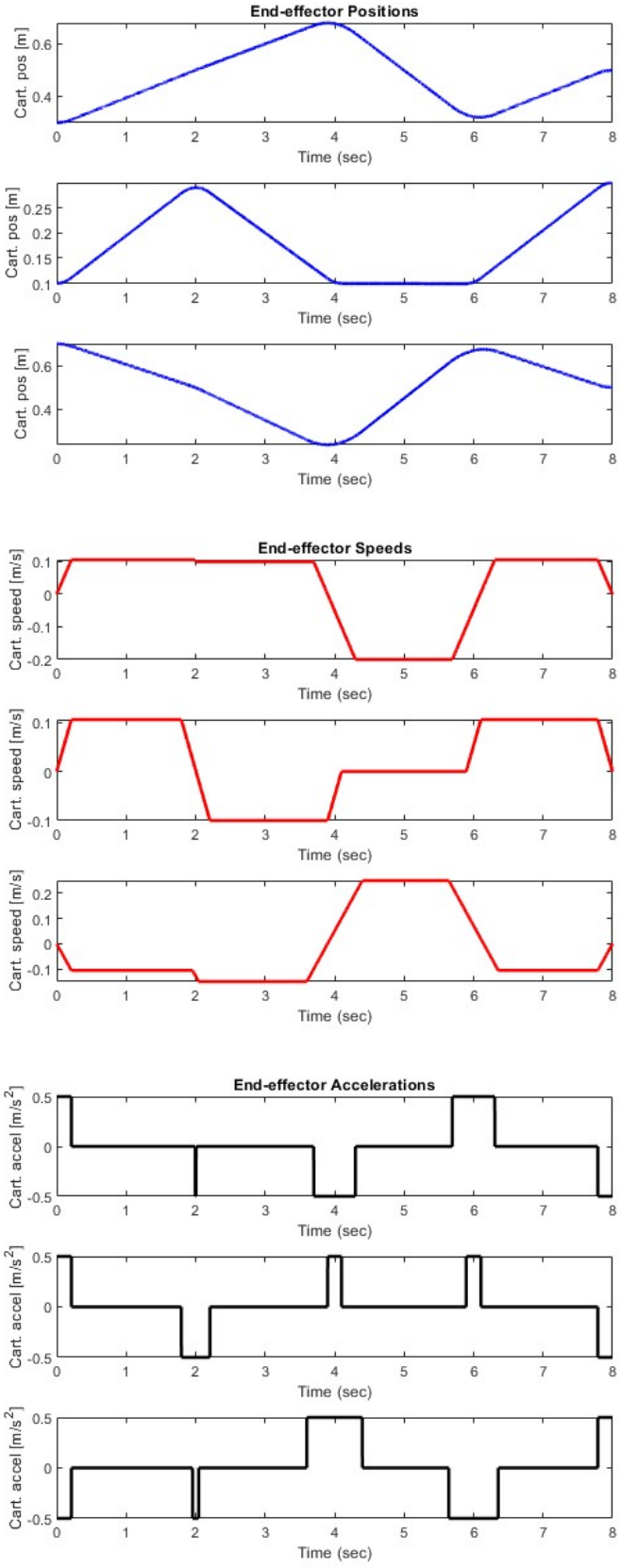


Figure (4.3): End-Effector Positions, Speeds, and Accelerations vs. Time for Triangular_Shape Path.

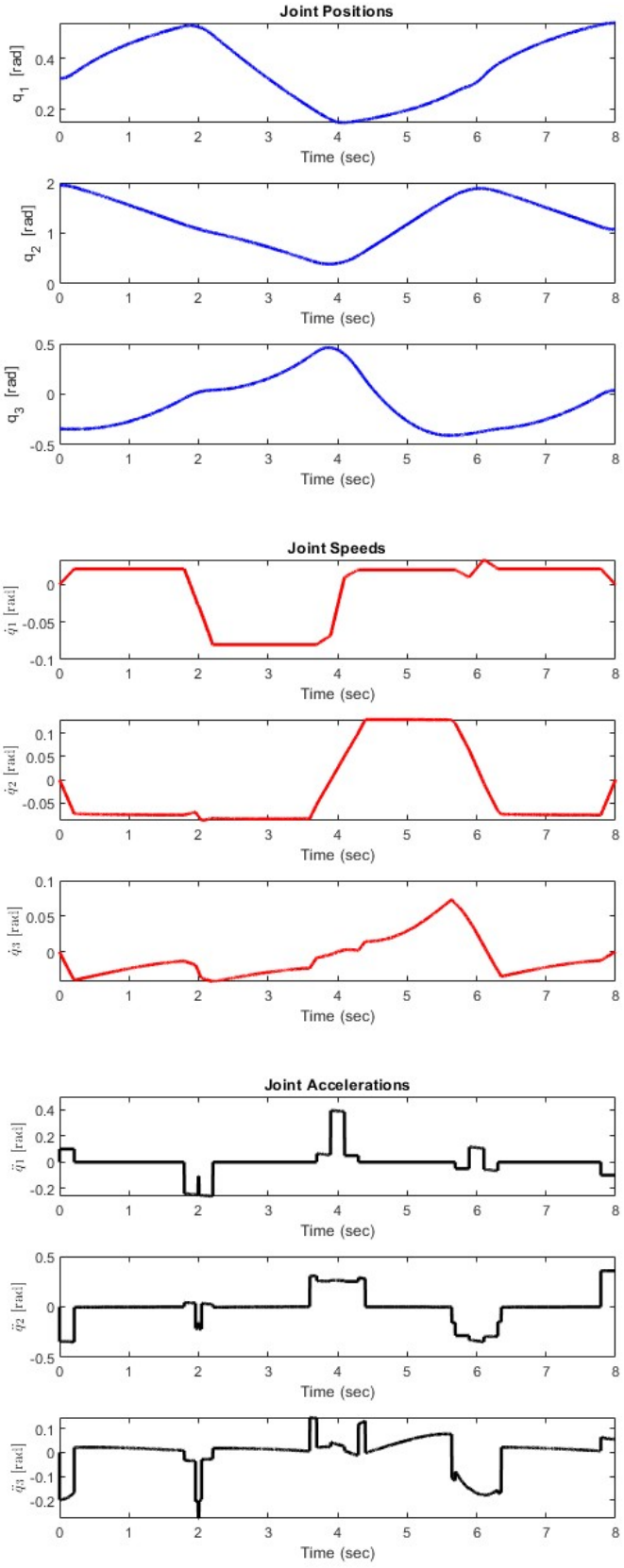


Figure (4.4): Joint Angles Positions, Speeds, and Accelerations vs. Time for Triangular_Shape Path.

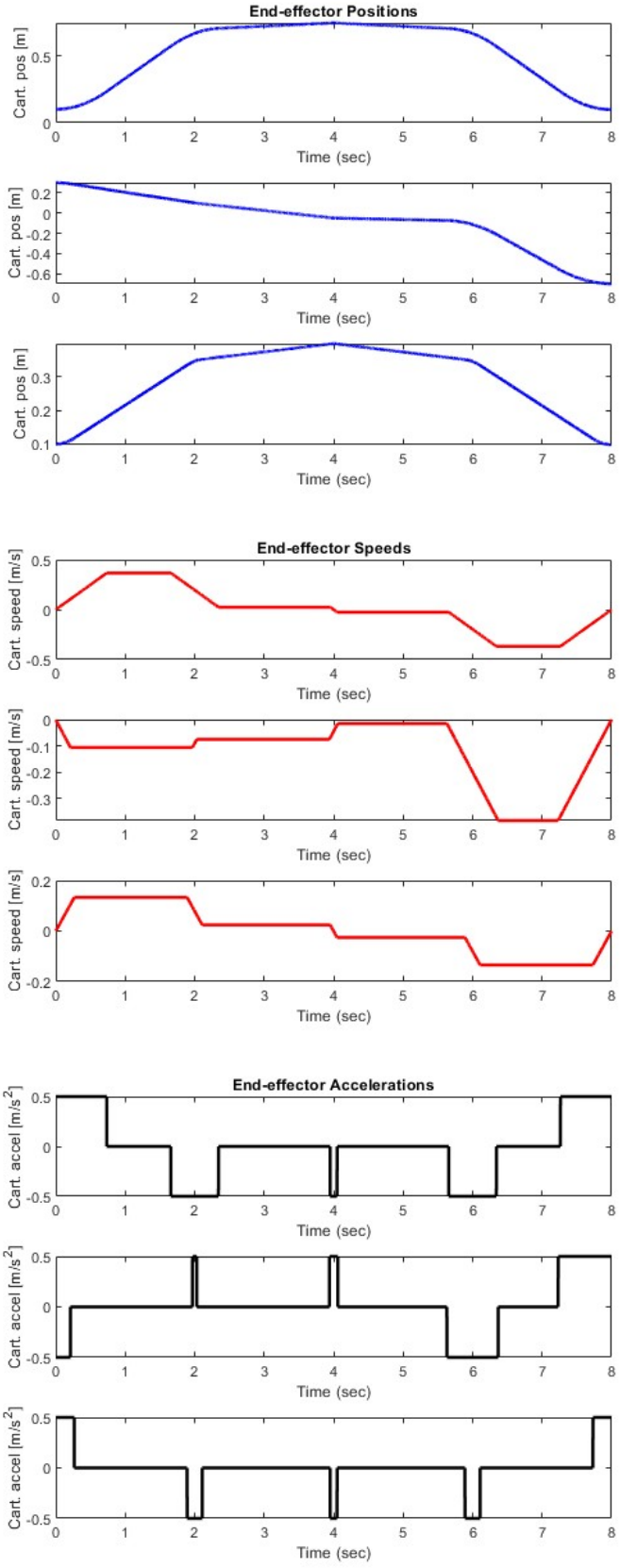


Figure (4.5): End-Effector Positions, Speeds, and Accelerations vs. Time for Pick_and_Place Path.

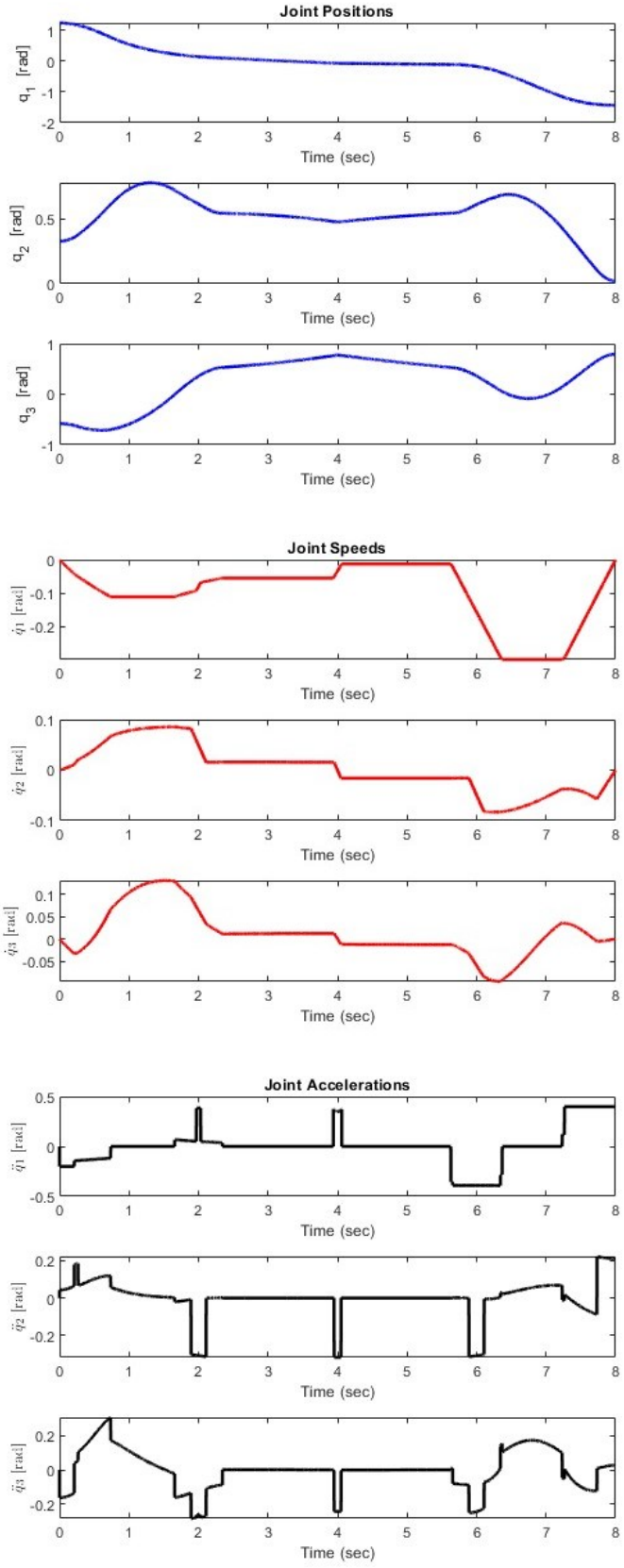


Figure (4.6): Joint Angles Positions, Speeds, and Accelerations vs. Time for Pick_and_Place Path.

Figures 4.7, 4.8, and 4.9 depict the ABB_IRB_140 robot's movement along the described trajectories: Via_Points, Triangular_Shape, and Pick_and_Place paths, respectively.

These graphs are instrumental in trajectory planning and optimization for robotic motion. Engineers and researchers can utilize these visualizations to design motion profiles that adhere to specific performance criteria, such as minimizing jerk (rate of change of acceleration) or ensuring smooth transitions between positions.

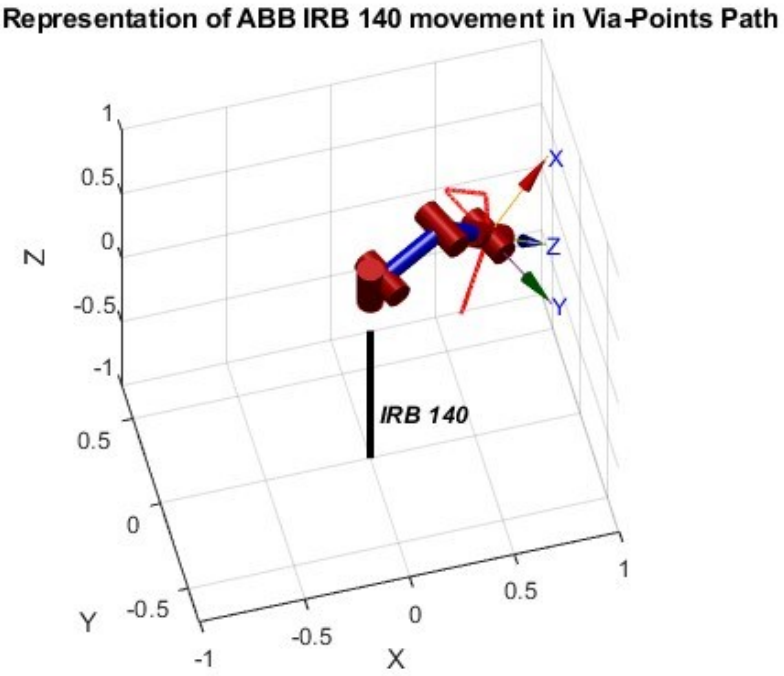


Figure 4.7: 3D Representation of the ABB_IRB_140 Robot Movement in the Via_Points Path.

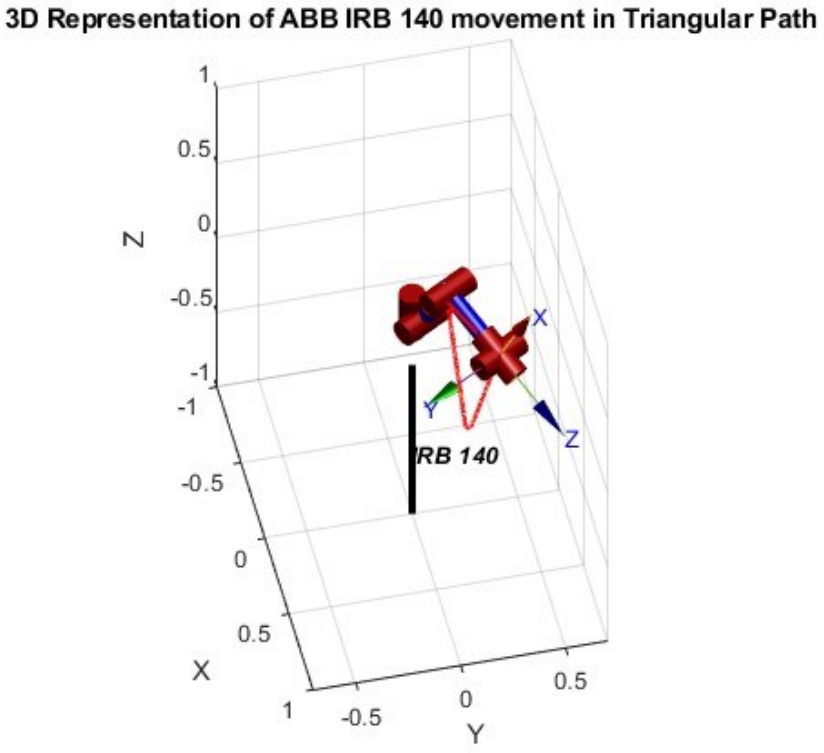


Figure 4.8: 3D Representation of the ABB_IRB_140 Robot Movement in the Triangular_Shape Path.

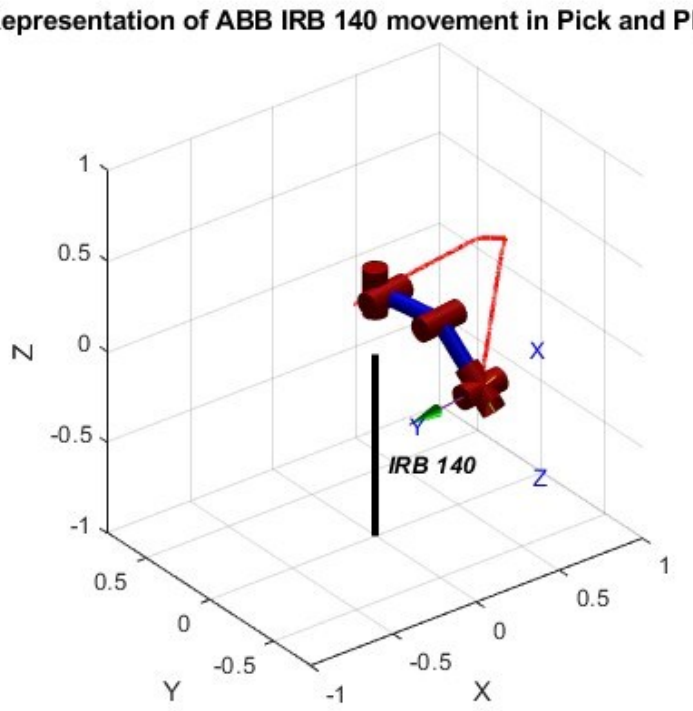


Figure 4.9: 3D Representation of the ABB_IRB_140 Robot Movement in the Pick_and_Place Path.

Then for each path in the same order a graph (figure 4.10, 4.11, and 4.12 respectively) will be presented showing the Joint torques and motor torques.

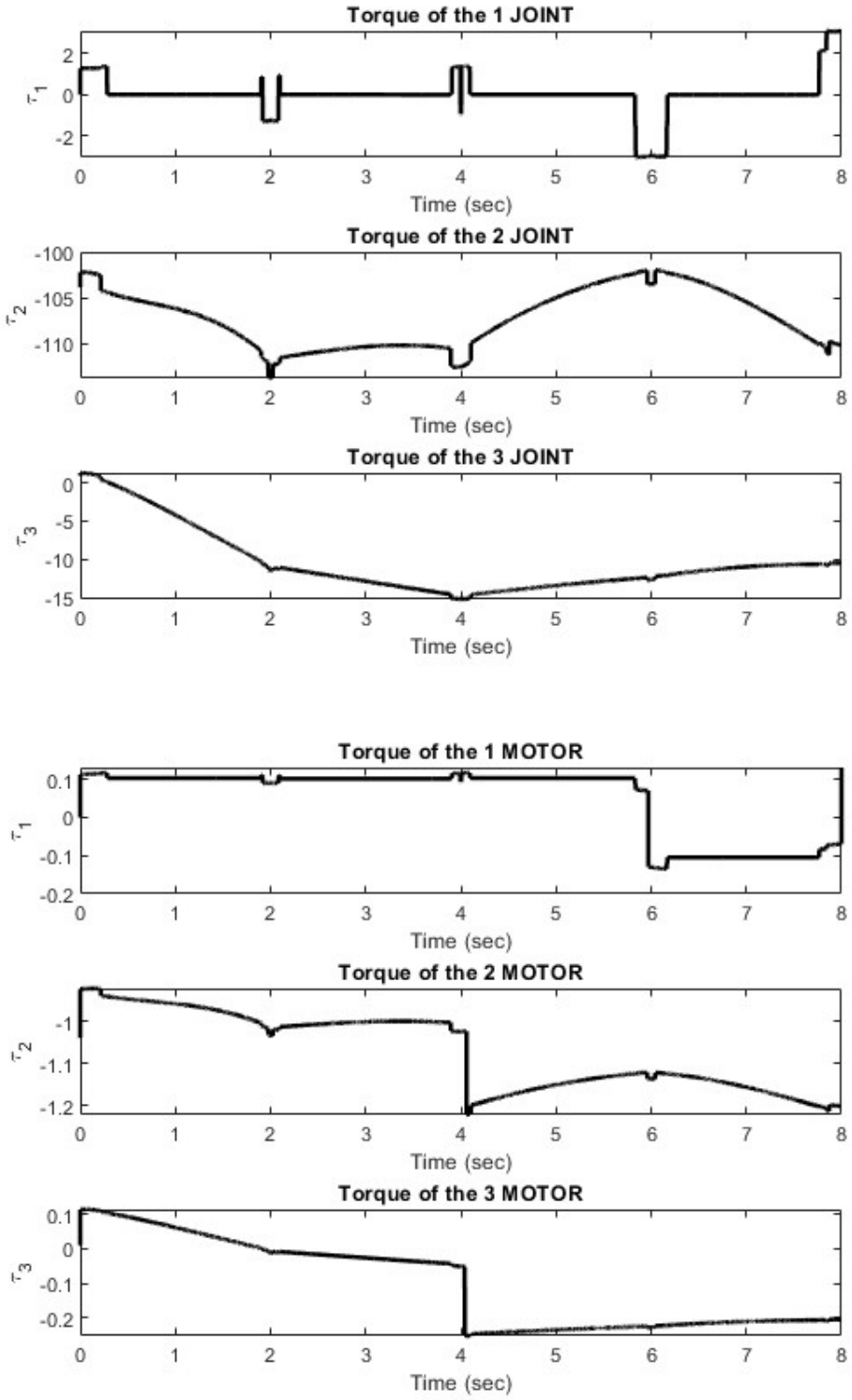


Figure 4.10: Joint and Motor Torques for Via_Points Path.

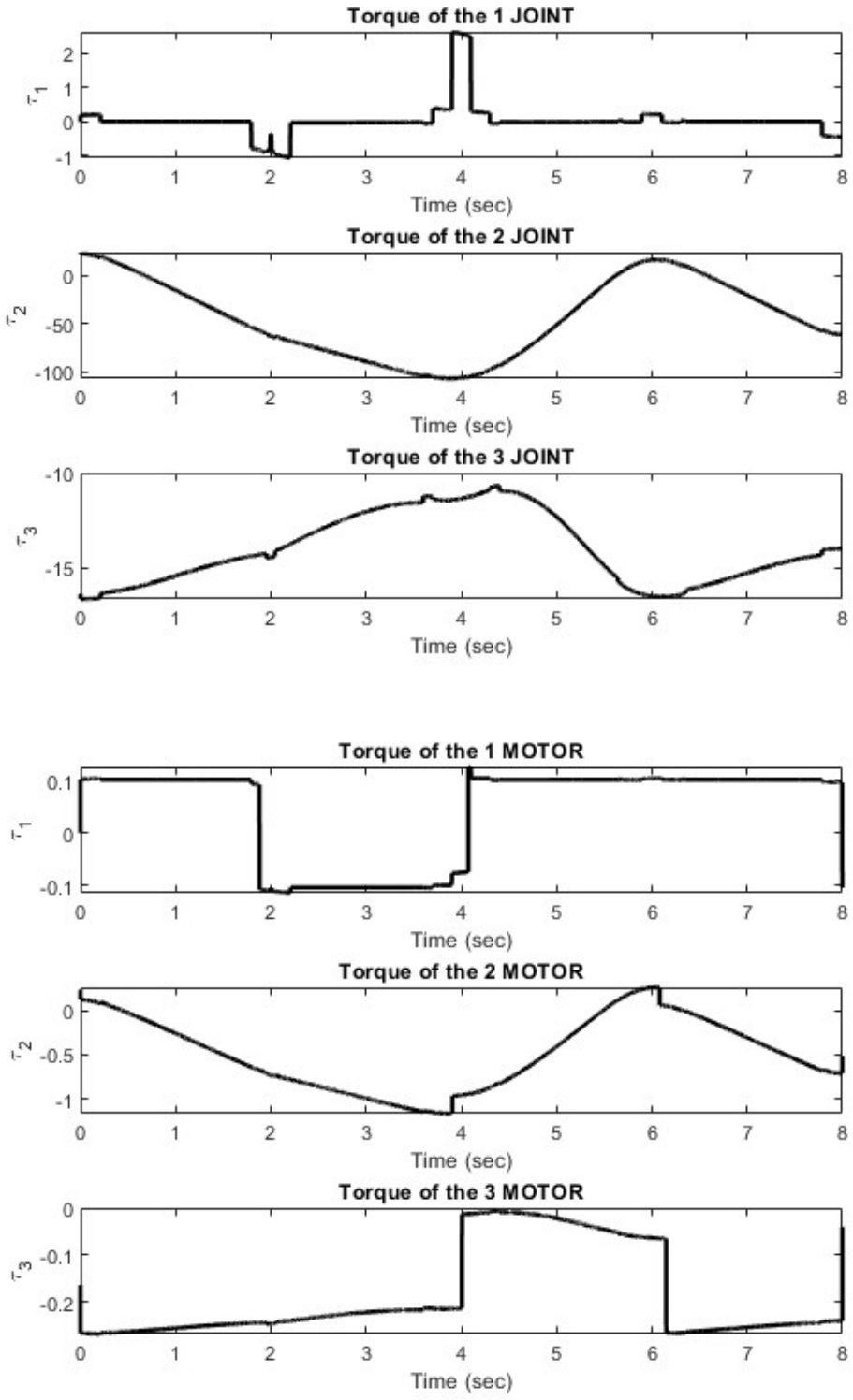


Figure 4.11: Joint and Motor Torques Triangular_Shape Path.

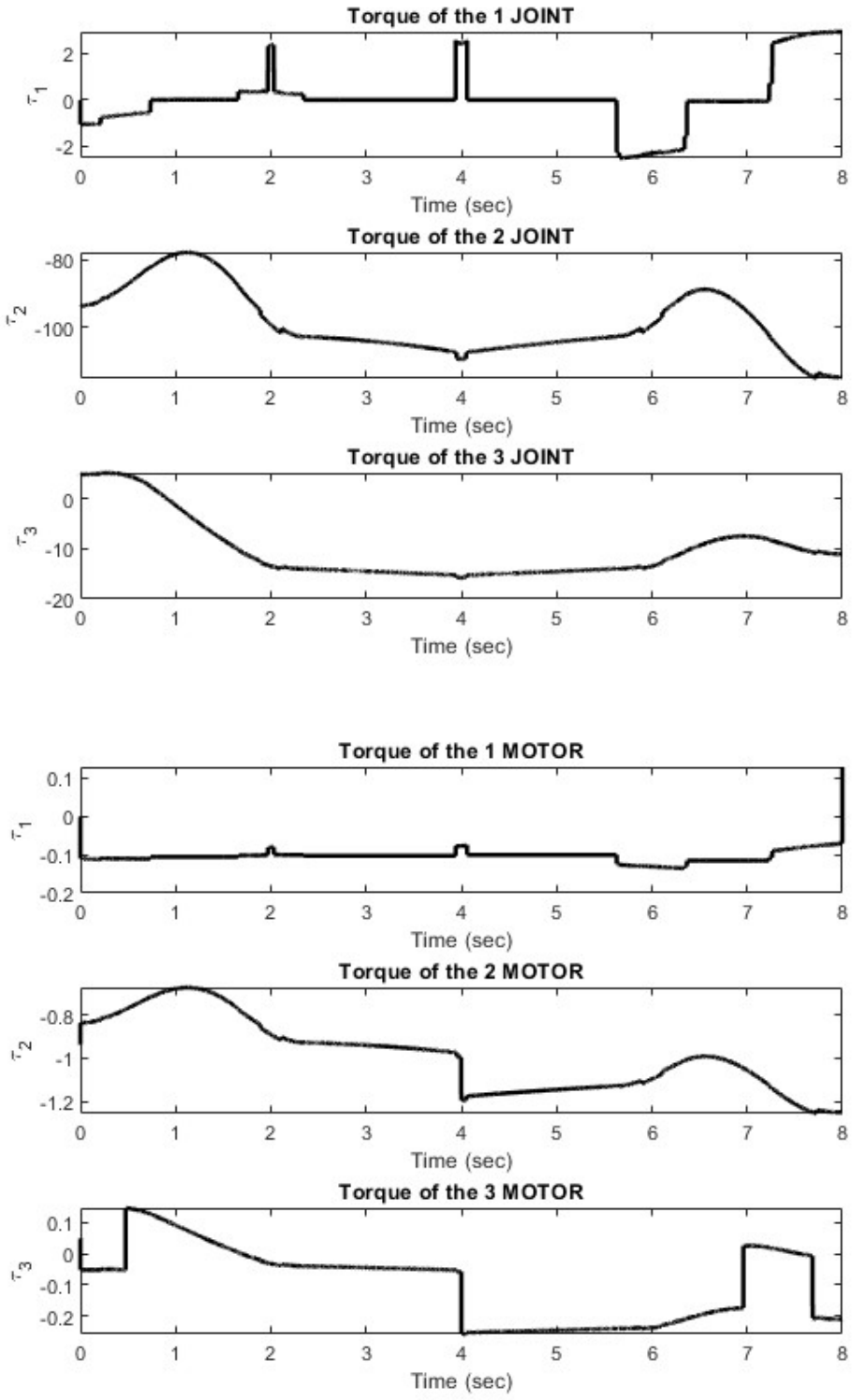


Figure 4.12: Joint and Motor Torques Pick_and_Place Path.

As elucidated in the Numerical Results section (Section 4.3), joint two and Motor two consistently exhibit the maximum torque values, followed by the third and finally the first. This observation underscores the principle that joints with higher mass demonstrate greater inertia, requiring higher torques during acceleration or deceleration.

Moreover, these graphs serve a crucial role in the selection and sizing of motors with appropriate torque ratings to meet the demands of the system effectively. By analysing torque profiles, engineers can ensure that selected motors operate within their specified limits, avoiding both overloading, which can lead to motor damage, and underutilization, which can result in inefficient system performance.

Following this, the joint power dissipation graphs and energy consumption graphs will be presented together as Figure 4.13, 4.14, and 4.15, respectively. This combined presentation offers a comprehensive understanding of the relationship between power dissipation and energy consumption. As energy is the integral of the power graph, analysing these graphs jointly provides deeper insights into both phenomena.

Power dissipation graphs highlight the sections of the paths that demand the most power. While much of the power graphs are predictable, there are instances where torque is high but power is unexpectedly low. This discrepancy often occurs when joints move very slowly, resulting in lower power consumption. Notably, this phenomenon is observable, particularly in joint three of the Pick_and_Place and Via_Points paths.

Furthermore, the energy graphs offer valuable information about the areas where significant energy consumption occurs. By examining peaks in the power requirement graphs, one can obtain a rough estimate of these energy-intensive sections along the path. This holistic analysis aids in identifying critical points in the trajectory where optimization efforts may be focused to enhance overall system efficiency and performance.

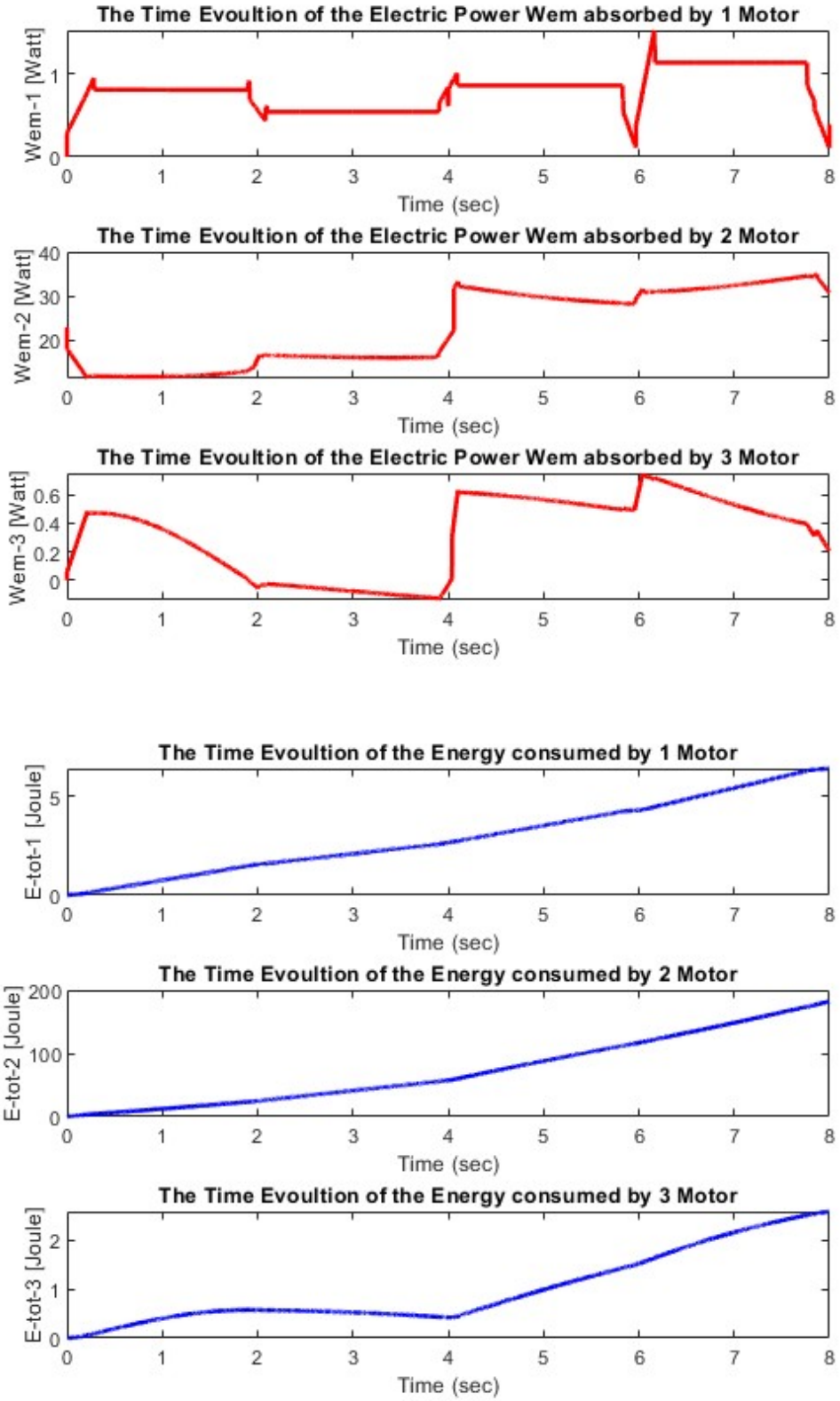


Figure 4.13: Electrical Power Dissipation and Total Energy Consumption in Via_Points Path.

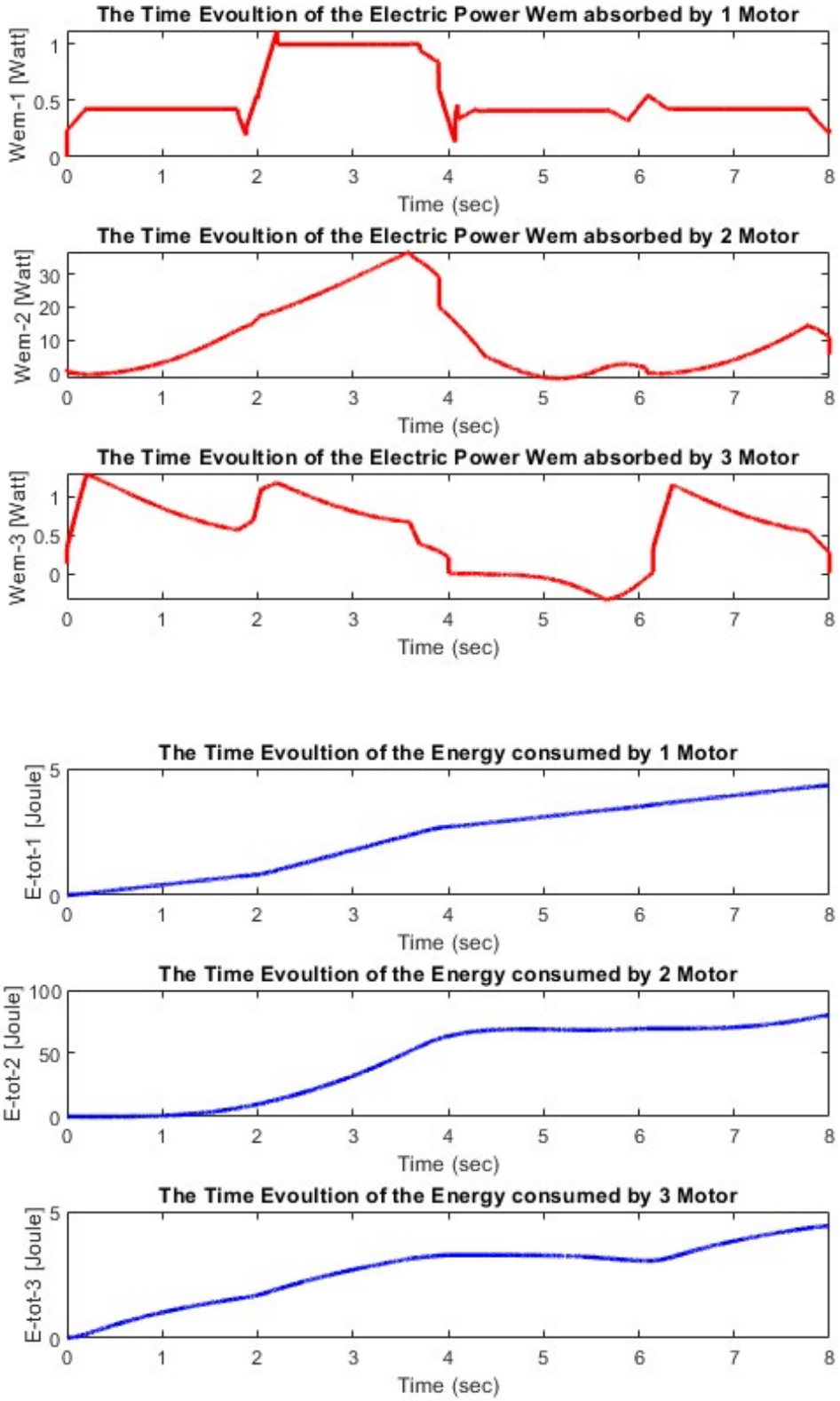


Figure 4.14: Electrical Power Dissipation and Total Energy Consumption in Triangular_Shape Path.

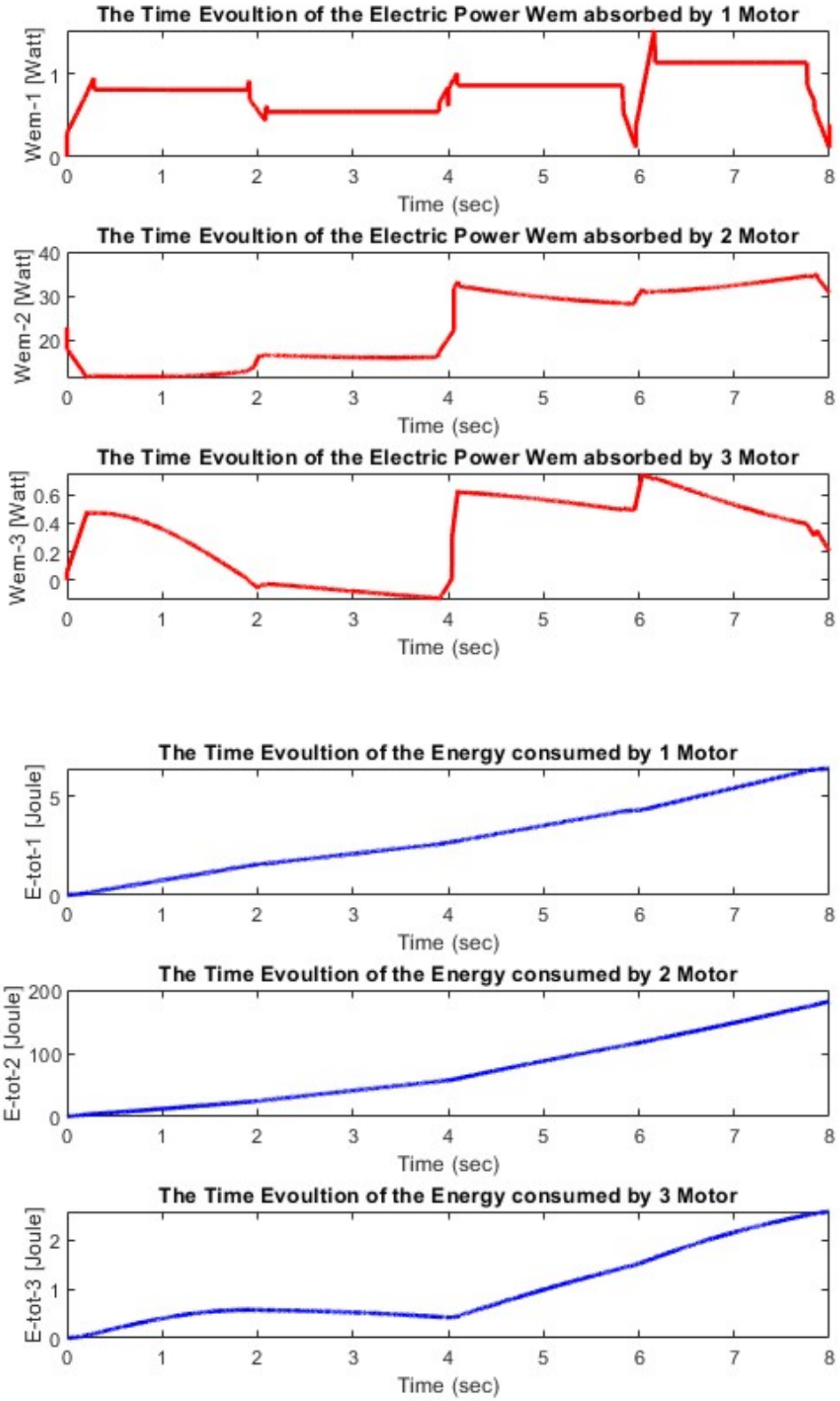


Figure 4.15: Electrical Power Dissipation and Total Energy Consumption in Pick_and_Place Path.

In the previous three figures, joint two consistently exhibits the highest energy demand: 181.73 Joules for Via_Points, 173.85 Joules for Pick_and_Place, and 80.31 Joules for Triangular_Path. This consistent trend can be attributed to its pivotal role in facilitating most of the robot's reaching motion. Notably, its velocity profile displays numerous spikes, indicative of rapid changes in velocity, while its torque profile also features many spikes, reflecting significant fluctuations in torque. These combined factors result in a substantial amount of power being consumed, consequently leading to a larger energy requirement for this particular joint.

Energy consumption and power dissipation graphs of robot joints provide engineers and researchers with valuable insights into the performance and efficiency of robotic systems. By analysing these graphs, they can evaluate how energy is consumed across different joints during operation. This analysis enables them to identify areas for optimization and improvement, such as reducing energy waste, optimizing motion trajectories, and implementing energy-saving strategies.

Engineers can also use these graphs to detect anomalies or fluctuations in energy consumption, which may indicate potential faults or malfunctions in robot components. By monitoring energy consumption patterns, engineers can perform preventive maintenance to address issues before they escalate, minimizing downtime and improving system reliability. Furthermore, these graphs serve as a benchmark for comparing the energy efficiency of different robot designs, control strategies, or operational modes. They also help quantify the environmental impact of robot operations by assessing energy consumption and associated emissions, contributing to sustainable engineering practices.

Overall, energy models and simulations play a crucial role in optimizing performance, ensuring reliability, and promoting sustainability in robotic systems.

CHAPTER FIVE:

CONCLUSION AND RECOMMENDATIONS

This chapter provides a comprehensive summary of the work conducted throughout the thesis, highlighting the achieved results that align with the thesis objectives. By presenting the obtained results and their relevance to the thesis goals, this chapter underscores the successful execution of the thesis's aims and objectives.

Furthermore, drawing from the insights gained during the thesis, several recommendations are proposed for potential areas of future research. These recommendations stem from identified gaps, limitations, or opportunities discovered during the course of the thesis. By outlining these suggestions for future investigation, this chapter not only consolidates the thesis's findings but also sets the stage for continued advancement and innovation in the field.

In essence, this chapter serves as a culmination of the thesis's endeavours, offering a reflection on the outcomes achieved and paving the way for further exploration and development in related areas. Through the dissemination of findings and recommendations, the thesis contributes to the ongoing discourse and progression of research in its respective domain.

5.1 Conclusion

The thesis work presents an application modelling and simulating approach to the energy consumption of industrial robots. It's noteworthy that the roadmap followed is applicable to all industrial robots when the final objective is modelling and simulating consumed energy. While some MATLAB codes and functions are tailored exclusively to certain types of robots, especially those requiring specific data such as link sizes and masses, the motion planning algorithm (which serves as the bottleneck for this study) relies on user-inserted data related to path descriptions. Consequently, the procedure for modelling energy consumption can be generalized for all industrial robots. This method and developed algorithm can thus play a vital role in understanding industrial robots' behaviour from an energy standpoint.

Here are some key achievements of this thesis:

- The construction of a comprehensive simulation model has been outlined from both kinematic and dynamic perspectives.

- Simultaneously, the formulation of corresponding mathematical equations describing the simulation model has been introduced.
- A simple yet robust motion planning algorithm has been developed, which adheres to constraints such as energy-effectiveness, task completion, and operator safety.
- Based on this algorithm, the equation for energy consumption has been derived.
- The proposed approach for determining energy consumption has been demonstrated through a series of simulation experiments, showcasing the evaluation of energy consumption for three selected motion trajectories.

5.2 Recommendations and Future Research

In addition to the achievements outlined in the thesis, several recommendations and avenues for future research can be explored:

1. Paths Optimization Study: Further investigation could be conducted using the same motion planning algorithm to optimize paths and trajectories, aiming to reduce overall energy consumption. By refining path planning strategies, it may be possible to identify more energy-efficient routes for industrial robots, thereby enhancing their operational efficiency.
2. Development of Control Algorithms: Another potential area for research involves the development of control algorithms or scheduling policies that minimize energy consumption while maintaining performance and productivity. By integrating advanced control techniques, such as model predictive control or adaptive control, energy-efficient operation of industrial robots can be achieved without compromising on task execution.
3. Extension to Multi-Robot Systems: Expanding the energy modelling and simulation framework to analyse the energy consumption of multi-robot systems operating collaboratively in industrial environments presents an intriguing opportunity. Understanding the energy dynamics of coordinated

robot teams can lead to more efficient resource allocation and task allocation strategies, ultimately improving overall system performance.

4. Data-Driven Approaches: Leveraging data-driven approaches, such as machine learning or data analytics, holds promise for optimizing energy efficiency in industrial robot systems. By utilizing historical energy consumption data, sensor measurements, and operational logs, predictive models and adaptive control strategies can be developed to optimize energy usage in real-time. This proactive approach enables robots to adapt their behaviour based on changing environmental conditions and operational requirements, thereby maximizing energy efficiency.
5. Integration of Digital Twins: Exploring the integration of digital twins across multiple domains, including mechanical, electrical, and software systems, offers significant potential for creating comprehensive digital representations of industrial robots. By simulating and analysing the behaviour of digital twins in virtual environments, engineers can gain valuable insights into energy consumption patterns and identify opportunities for optimization and improvement. This integrated approach enables holistic optimization of industrial robot systems, leading to enhanced performance, reliability, and energy efficiency.

By pursuing these recommendations and avenues for future research, the field of industrial robotics can continue to advance, leading to more energy-efficient and sustainable manufacturing processes.

REFERENCES & BIBLIOGRAPHY

1. Niku, S.B. (2020). Introduction to robotics: analysis, control, applications. Hoboken, Nj: Wiley.
2. Craig, J.J. (2018). Introduction to robotics: mechanics and control. Upper Saddle River, Nj: Pearson.
3. Dubai Sensor. (n.d.). Introduction to Robotics: Robots, History and Structure. [online] Available at: https://www.dubai-sensor.com/blog/introduction-to-robotics-robots-history-and-structure/ [Accessed 7 Jan. 2024].
4. Singh, Balkeshwar. “Evolution of Industrial Robots and Their Applications.” Www.academia.edu, vol. 3, no. 5, 1 Jan. 2013, www.academia.edu/71682433/Evolution_of_Industrial_Robots_and_their_Applications.
5. IFR. “International Federation of Robotics.” Ifr.org, 2019, ifr.org/industrial-robots.
6. Müller, Christopher: World Robotics 2023 – Industrial Robots, IFR Statistical Department, VDMA Services GmbH, Frankfurt am Main, Germany, 2023.
7. Eurostat. “Energy Statistics - an Overview.” Ec.europa.eu, Feb. 2022, ec.europa.eu/eurostat/statistics-explained/index.php?title=Energy_statistics_-_an_overview.
8. Widuto, Agnieszka. EN BRIEFING Energy Saving and Demand Reduction. Sept. 2022.
9. Pellegrinelli, Stefania, et al. “Minimization of the Energy Consumption in Motion Planning for Single-Robot Tasks.” Procedia CIRP, vol. 29, no. 2212-8271, 1 Jan. 2015, pp. 354–359, www.sciencedirect.com/science/article/pii/S2212827115004886, https://doi.org/10.1016/j.procir.2015.02.174.
10. Čapek, Karel. (Rossum´S Universal Robots) a Play in Introductory Scene and Three Acts DRAMATIS PERSONAE. 2006.
11. ASIMOV, ISAAC. “RUNAROUND.” WJCC Schools, 1942, wjccschools.org/wp-content/uploads/sites/2/2016/01/Runaround-by-Isaac-Asiimov.pdf. Accessed 2024.

12. Kojcev, Risto, et al. Dissecting Robotics -Historical Overview and Future Perspectives Dissecting Robotics -Historical Overview and Future Perspectives. 2017.
13. Wallén, Johanna. The History of the Industrial Robot. 2008.
14. Malone, Bob. “George Devol: A Life Devoted to Invention, and Robots.” IEEE Spectrum, 26 Sept. 2011, spectrum.ieee.org/george-devol-a-life-devoted-to-invention-and-robots.
15. Gasparetto, A., and L. Scalera. “A Brief History of Industrial Robotics in the 20th Century.” Advances in Historical Studies, vol. 08, no. 01, 2019, pp. 24–35, https://doi.org/10.4236/ahs.2019.81002. Accessed 27 Jan. 2024.
16. Victor David, Scheinman. “DESIGN of a COMPUTER CONTROLLED MANIPULATOR.” Apps.dtic.mil, 1 June 1969, apps.dtic.mil/sti/citations/AD0708069. Accessed 27 Jan. 2024.
17. Makino, H., and Furuya N. “SELECTIVE COMPLIANCE ASSEMBLY ROBOT ARM.” Pascal-Francis.inist.fr, 1980, pascal-francis.inist.fr/vibad/index.php?action=getRecordDetail&idt=PASCAL8130152643. Accessed 27 Jan. 2024.
18. Asada, Haruhiko, and Takeo Kanade. “Design of Direct-Drive Mechanical Arms.” Journal of Vibration and Acoustics, vol. 105, no. 3, 1 July 1983, pp. 312–316, https://doi.org/10.1115/1.3269106. Accessed 26 Dec. 2020.
19. IFR. From the First Installation until Today Milestones of Technology and Commercialization History of Industrial Robots Compiled by the International Federation of Robotics -IFR 2012.
20. Clavel, R. (1991). Contribution à l’étude et à la réalisation d’un robot delta [PhD thesis, Ecole Polytechnique Fédérale de Lausanne]. Retrieved from https://infoscience.epfl.ch/record/31403/files/EPFL_TH925.pdf
21. Bogue, Robert. “The First Half Century of Industrial Robot: 50 Years of Robotic Developments.” Emerald Insights, no. 0143-991x, 9 Nov. 2022, https://doi.org/10.1108/IR-10-2022-0251]. Accessed 27 Jan. 2024.
22. International Federation of Robotics. (2022). World Robotics - Industrial Robots. Retrieved from

https://ifr.org/img/worldrobotics/WR_Industrial_Robots_2022_Chapter_1.pdf In-text citation: (International Federation of Robotics, 2022)

23. ISO. (2021). ISO 8373:2021 Robots and robotic devices -- Vocabulary. Retrieved from https://www.iso.org/standard/67639.html In-text citation: (ISO, 2021)
24. ABB PRODUCTS. Product Specification IRB 140.
25. Baquero-Suárez, Mauro, et al. Kinematics, Dynamics and Evaluation of Energy Consumption for ABB IRB- 140 Serial Robots in the Tracking of a Path Autonomous Bicycles View Project Simulación de Modelos Matemáticos Para Ingeniería Química. View Project Kinematics, Dynamics and Evaluation of Energy Consumption for ABB IRB-140 Serial Robots in the Tracking of a Path. 2013, https://doi.org/10.13140/2.1.3436.5448.
26. Zanchettin, Andrea Maria, et al. "Exploiting Task Redundancy in Industrial Manipulators during Drilling Operations." 2011 IEEE International Conference on Robotics and Automation, May 2011, https://doi.org/10.1109/icra.2011.5979624. Accessed 30 Jan. 2024.
27. Gnjatovic, Milan, et al. "Adaptive Multimodal Interaction with Industrial Robot." 2012 IEEE 10th Jubilee International Symposium on Intelligent Systems and Informatics, Sept. 2012, https://doi.org/10.1109/sisy.2012.6339538.
28. Pollák, Martin, et al. The Structural Design of 3D Print Head and Execution of Printing via the Robotic Arm ABB IRB 140. Apr. 2018.
29. Siciliano, Bruno, et al. Robotics. Advanced Textbooks in Control and Signal Processing, London, Springer London, 2009, link.springer.com/book/10.1007/978-1-84628-642-1.
30. Denavit, J., and R. S. Hartenberg. "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices." Journal of Applied Mechanics, vol. 22, no. 2, 1 June 1955, pp. 215–221, https://doi.org/10.1115/1.4011045.
31. Toquica, Juan S, and T Motta. "A Methodology for Industrial Robot Calibration Based on Measurement Sub-Regions." The International Journal of Advanced Manufacturing Technology, vol. 119, no. 1-2, 22 Nov. 2021, pp. 1199–1216, https://doi.org/10.1007/s00170-021-08308-4. Accessed 3 Feb. 2024.

32. Pieper, Donald, and quot; Jwn. CO CO O O 00 Q the KINEMATICS of MANIPULATORS under COMPUTER CONTROL. 1968.
33. Hunt, K. H. “Special Configurations of Robot-Arms via Screw Theory.” Robotica, vol. 4, no. 3, 1 July 1986, pp. 171–179, www.cambridge.org/core/journals/robotica/article/abs/special-configurations-of-robotarms-via-screw-theory/9DCB2DCECB99811794C44DC2652DD14E, https://doi.org/10.1017/S0263574700009358. Accessed 4 Feb. 2024.
34. Ribeiro, Luiz, et al. SCREW-BASED RELATIVE JACOBIAN for MANIPULATORS COOPERATING in a TASK. 2008.
35. Farid, Ghulam, et al. “Computationally Efficient Algorithm to Generate a Waypoints-Based Trajectory for a Quadrotor UAV.” IEEE, 1 June 2018, https://doi.org/10.1109/ccdc.2018.8407894. Accessed 4 Feb. 2024.
36. Feng, Yixiong, et al. “An Energy-Saving Optimization Method for Cyclic Pick-And-Place Tasks Based on Flexible Joint Configurations.” Robotics and Computer-Integrated Manufacturing, vol. 67, 1 Feb. 2021, pp. 102037–102037, https://doi.org/10.1016/j.rcim.2020.102037. Accessed 4 Feb. 2024.
37. Singh, Gurjeet, and Vijay Kumar Banga. “Kinematics and Trajectory Planning Analysis Based on Hybrid Optimization Algorithms for an Industrial Robotic Manipulators.” Soft Computing, vol. 26, no. 21, 26 Aug. 2022, pp. 11339–11372, https://doi.org/10.1007/s00500-022-07423-y. Accessed 8 May 2023.
38. Boscariol, Paolo, and Dario Richiedei. “Energy-Efficient Design of Multipoint Trajectories for Cartesian Robots.” The International Journal of Advanced Manufacturing Technology, vol. 102, no. 5-8, 14 Jan. 2019, pp. 1853–1870, https://doi.org/10.1007/s00170-018-03234-4. Accessed 5 Feb. 2024.
39. Hollerbach, J. (1989). Kinematics and dynamics for control. In: Robotics Science, edited by MIT. Cambridge.
40. Nagurka, M.L. 2005. Newton-Euler Dynamics of Robots. In: Kurfess, T.R. (ed.) Robotics and Automation Handbook. New York: CRC Press.
41. Boscariol, Paolo, and Dario Richiedei. “Energy Saving in Redundant Robotic Cells: Optimal Trajectory Planning.” Mechanism Design for Robotics, 31 Aug.

2018, pp. 268–275, https://doi.org/10.1007/978-3-030-00365-4_32. Accessed 5 Feb. 2024.

42. Boscariol, Paolo, et al. “Energy Optimization of Functionally Redundant Robots through Motion Design Special Issue Optimization of Motion Planning and Control for Automatic Machines, Robots and Multibody Systems Edited by Energy Optimization of Functionally Redundant Robots through Motion Design.” MDPI, 26 Apr. 2020, https://doi.org/10.3390/app10093022. Accessed 5 Feb. 2024.
43. Carter, Tyler. THE MODELING of a SIX DEGREE-OF-FREEDOM INDUSTRIAL ROBOT for the PURPOSE of EFFICIENT PATH PLANNING a Thesis in Industrial Engineering. 2009.