FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

DESIGN AND DEVELOPMENT OF AN OPEN-SOURCE ARTICULATED ROBOT MANIPULATOR FOR EDUCATION-BASED INDUSTRIAL APPLICATIONS

Mehmet Can TÜRK

Master's Thesis

DEPARTMENT OF MECHANICAL ENGINEERING

JULY 2022

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

Department of Mechanical Engineering

Master's Thesis

DESIGN AND DEVELOPMENT OF AN OPEN-SOURCE ARTICULATED ROBOT MANIPULATOR FOR EDUCATION-BASED INDUSTRIAL APPLICATIONS

Author
Mehmet Can TÜRK

Supervisor
Prof. Dr. Servet SOYGÜDER

JULY 2022
ELAZIG

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

Department of Mechanical Engineering

Master's Thesis

Title:	Design and Development of an Open-Source Articulated Robot Manipulator for Education-Based Industrial Applications
Author:	Mehmet Can TÜRK
Submission Date:	31 May 2022
Defense Date:	04 July 2022

THESIS APPROVAL

This thesis, which was prepared according to the thesis writing rules of the Graduate School of Natural and Applied Sciences, Fırat University, was evaluated by the committee members who have signed the following signatures and was unanimously approved after the defense exam made open to the academic audience.

		Signature
Supervisor:	Prof. Dr. Servet SOYGÜDER Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences	Approved
Chair:	Prof. Dr. İlyas ÇANKAYA Yıldırım Beyazıt University, Faculty of Engineering and Natural Sciences	Approved
Member:	Prof. Dr. Orhan ÇAKAR Fırat University, Faculty of Engineering	Approved

This thesis was approved by the Administrative Board of the Graduate School on

.... / .... / 20 ....

Signature

Prof. Dr. Kürşat Esat ALYAMAÇ
Director of the Graduate School

DECLARATION

I hereby declare that I wrote this Master's Thesis titled "Design and Development of an Open-Source Articulated Robot Manipulator for Education-Based Industrial Applications" in consistent with the thesis writing guide of the Graduate School of Natural and Applied Sciences, Fırat University. I also declare that all information in it is correct, that I acted according to scientific ethics in producing and presenting the findings, cited all the references I used, express all institutions or organizations or persons who supported the thesis financially. I have never used the data and information I provide here in order to get a degree in any way.

04 July 2022

Mehmet Can TÜRK

PREFACE

During this study, one of the main difficulties was to obtain a profound inverse kinematic solution since the problem was quite complex and many nonlinear equations had to be handled carefully. And the literature on the subject was not easy to follow and to validate since they were mostly theoretical solutions based on certain toolboxes. The unclarity of the literature was both one of the main problems and one of the main motivations to complete the study with an applied solution.

The fund required for the prototype manufacturing has been provided mainly by a scientific research project submitted to and accepted by OSTIM Technical University in Ankara/TURKEY.

On the second phase of the study where an industrial robotic arm was customized both on the software and hardware levels, a collaboration with an automation company called KNet Mechatronic Systems has been established and the implementation process has been completed in cooperation. They deserve special thanks for being easy to collaborate and their consent on the publication of some of the information on the implementation phase.

There are two people that I admire and respect so much who have contributed to this study indirectly, Angela Sodemann and Chris Annin. Without them, this study would not be as it is now. I'm very grateful to both!

I want to express my gratitude towards my supervisor Prof. Dr. Servet SOYGÜDER as he always supported and guided me during the study and pushed me even further in tough times.

And finally, special thanks to my dear wife Zehra for her understanding and patience with me during this study.

Mehmet Can TÜRK
ELAZIG, 2022

TABLE OF CONTENTS

Page

PREFACE IV
TABLE OF CONTENTS V
ABSTRACT VI
ÖZET VIII
LIST OF FIGURES IX
LIST OF TABLES XII
LIST OF APPENDICES XIII
SYMBOLS AND ABBREVIATIONS XIV

1. INTRODUCTION 1
1.1. Statement of the Problem 1
1.2. Objectives 2
1.3. Scope of the Study 3
1.4. Outline of the Thesis 5

2. GENERAL INFORMATION AND LITERATURE REVIEW 6
2.1. General Information 6
2.1.1. Definition of the Word “Robot” 6
2.1.2. Advantages of Robots 6
2.1.3. Statistics on Robots Used in Industry 7
2.1.4. Types of Robots 9
2.1.5. Use of Articulated Robots 12
2.2. Background in the Field 16
2.3. Related Academic Studies 17

3. MATERIAL AND METHOD 20
3.1. Preparation for Development 20
3.1.1. Technical Specifications 22
3.2. Method and Theoretical Solution 23
3.2.1. Denavit-Hartenberg Notation 24
3.2.2. Forward Kinematics 27
3.2.3. Inverse Kinematics 30
3.2.4. Mechanical Design 43
3.2.5. Simulation and Control 48
3.2.6. Workspace Estimation 51
3.3. Prototype Development 52
3.3.1. Static Load Calculations 52
3.3.2. Component Selection 54
3.3.3. Test Setups 55
3.3.4. System Hardware Topology 56
3.3.5. Manufacture and Physical Tests 57

4. STUDY ON THE CUSTOM KUKA KR5 ROBOT 58
4.1.1. Scope of the Study 58
4.1.2. Partial Inverse Kinematic Solution 58
4.1.3. Mechanical Design 61

4.1.4. Simulation 62
4.1.5. Workspace Estimation 62
4.1.6. Implementation on the Physical Model 63
5. RESULTS AND DISCUSSION 65
5.1. Results From the Designed Robot 65
5.1.1. Simulation Results for Linear Motion Along Z-Axis 65
5.1.2. Simulation Results for Linear Motion Along X-Axis 66
5.1.3. Simulation Results for Circular Motion on YZ-Plane 67
5.1.4. Simulation Results for Circular Motion on XY-Plane 68
5.1.5. Simulation Results for Sinusoidal Motion on XZ-Plane 69
5.1.6. Simulation Results for Interplanar Motion on Multiple Planes 70
5.2. Results from the KUKA KR5 Robot 71
5.2.1. Simulation Results for Linear Motion Along Z-Axis 71
5.2.2. Simulation Results for Linear Motion Along X-Axis 72
5.2.3. Simulation Results for Circular Motion on XY-Plane 73
5.2.4. Physical Test Results for Linear Motion Along Z-Axis 74
5.2.5. Physical Test Results for Linear Motion Along Z-Axis 75
6. CONCLUSIONS 76
RECOMMENDATIONS 78
REFERENCES 79
APPENDICES 82
CURRICULUM VITAE

ABSTRACT

Design and Development of an Open-Source Articulated Robot Manipulator for Education-Based Industrial Applications

Mehmet Can TÜRK

Master's Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences

Department of Mechanical Engineering

July 2022, Page: xv + 121

Articulated robotic manipulators with 6 degrees of freedom are commonly used in industry and their importance is growing since they are capable of replacing human effort in many areas. Yet, the theoretical background and the practical use of this technology don't come together as often since many academics prefer to work on the theoretical frame while many companies prefer to carry commissioning tasks of the robots manufactured by a very limited number of high-tech companies who present their robots as a closed-architecture. This situation dramatically slows down the progress of making the essence of this technology more accessible to wider communities.

In order to contribute to making this technology more accessible to anyone professionally interested in robotics; this study focused on the development process of a 6-DoF robotic manipulator from the preliminary design phase to the fully-functioning prototype phase including inverse kinematic analysis, simulation development, etc.

As an extra study to validate the results, the customization of an actual industrial robotic manipulator KUKA KR5 which didn't have any electronic components provided by the manufacturer has been accomplished. A similar solution to the originally designed robotic arm has been obtained for this model as well and implemented on the actual KUKA robot. In the end, the KUKA KR5 robot has been controlled based on the theoretical solution obtained within the scope of this study and the custom electronic components mounted on the mechanical body. So, the actual industrial robotic arm ran totally independently from the manufacturer's control software and electronic hardware.

Keywords: Articulated robot manipulator, 6 degrees-of-freedom robotic arm, Analytical inverse kinematics, Open-Source robot manipulator

ÖZET

Eğitim Amaçlı Endüstriyel Uygulamalar için Açık Kaynaklı Mafsallı Bir Robot Manipülatörün Tasarımı ve Geliştirilmesi

Mehmet Can TÜRK

Yüksek Lisans Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Makine Mühendisliği Anabilim Dalı

Temmuz 2022,   Sayfa: xv + 121

Altı serbestlik dereceli mafsallı robot manipülatörler endüstride yaygın olarak kullanılmaktadır ve birçok alanda insan emeğinin yerini alabilecek kapasitede oldukları için önemleri giderek artmaktadır. Ancak birçok akademisyenin teorik çerçeve üzerinde çalışmayı tercih etmesi ve birçok robotik/otomasyon şirketinin ise çok az sayıdaki yüksek teknolojili robot üreticisi firmanın piyasaya kapalı bir mimari halinde sundukları robotları devreye alma görevlerini yerine getirmeyi tercih etmesi gibi nedenlerden dolayı bu teknolojinin teorik arka planı ve pratik kullanımı pek sık bir araya gelememektedir. Bu durum, bu teknolojinin özünü geniş topluluklar için daha erişilebilir hale getirme sürecini önemli ölçüde yavaşlatmaktadır.

Bu teknolojinin robotik alanı ile profesyonel olarak ilgilenen herkes için daha erişilebilir olmasına katkıda bulunmak amacıyla bu çalışmada; ön tasarım aşamasından ters kinematik analiz, simülasyon geliştirme vb. dahil olmak üzere tam işlevsel prototip aşamasına kadar özel bir 6 serbestlik dereceli robot kolun geliştirme sürecine odaklanılmıştır.

Sonuçları doğrulamak için ek bir çalışma olarak, üzerinde üretici tarafından sağlanan herhangi bir elektronik bileşeni olmayan KUKA KR5 isimli gerçek bir endüstriyel robot kolun özelleştirilmesi gerçekleştirilmiştir. KUKA robot için; bu çalışmanın ana odağı olan 6 serbestlik dereceli özel robot kol için geliştirilen çözüme benzer benzer bir çözüm elde edilmiş ve bu teorik çözüm gerçek robot üzerinde uygulanmıştır. Çalışmanın sonunda, KUKA KR5 robotu, çalışma kapsamında geliştirilen özel teorik çözüm ve mekanik gövdeye monte edilen özel elektronik bileşenler temelinde, üreticinin kontrol yazılımı ve elektronik donanımından bağımsız bir şekilde başarıyla kontrol edilmiştir.

Anahtar Kelimeler: Eklemli robot manipülatörü, 6 serbestlik dereceli robot kol, Analitik ters kinematik, Açık kaynak robot manipülatörü

LIST OF FIGURES

Page

Figure 1.1. Industrial Robotic Arms in a Factory 1
Figure 1.2. A Commercial User Interface to Control Industrial Robots by STAUBLI 2
Figure 1.3. A Commercial Desktop-size Robotic Manipulator 4
Figure 1.4. An Industrial Robotic Arm KUKA KR5 ARC-C With no Electronic Components 5
Figure 2.1. Top 10 Countries which Employ the Highest Number of Robots 7
Figure 2.2. Top 10 Industries that Employ the Highest Number of Robots 8
Figure 2.3. Operational Stock of Robots in the World 8
Figure 2.4. Annual Installation of Robots by Industry in the World between 2018-2020 9
Figure 2.5. Annual Installation of Robots between 2015-2020 and Forecast for 2021-2024 9
Figure 2.6. An Articulated Robot 10
Figure 2.7. An Articulated Robot Mounted on a Linear Tray 11
Figure 2.8. An Articulated Robot Mounted on a Mobile Robot 11
Figure 2.9. An Articulated Arc Welding Robot 12
Figure 2.10. An Articulated Spot Welding Robot 12
Figure 2.11. An Articulated Painting Robot 13
Figure 2.12. An Articulated Material Handling Robot 13
Figure 2.13. An Articulated Robot in Forging Industry 14
Figure 2.14. Articulated Robots on a Production Line 14
Figure 2.15. The Famous Articulated Surgical Robot Called “Da Vinci” - General View 15
Figure 2.16. The Famous Articulated Surgical Robot Called “Da Vinci” - Close Shot 15
Figure 3.1. Axis Assignment of the Designed Robotic Manipulator 26
Figure 3.2. Visual Representation of Orientation Values nx, sy, az 32
Figure 3.3. Visual Representation of Orientation Values ax, ay, az 32
Figure 3.4. Definition of Axis Assignments of the Robotic Manipulator 38
Figure 3.5. Geometric Model of the Designed Robot when Wrist is Above Shoulder 39
Figure 3.6. Geometric Model of the Designed Robot when Wrist is Below Shoulder 40
Figure 3.7. Technical Drawing of the Designed Robotic Arm 44
Figure 3.8. CAD Model of the Designed Robotic Arm 44
Figure 3.9. Robotic Arm Bended Side Views 45
Figure 3.10. Robotic Arm Horizontal Orthogonal Views 45
Figure 3.11. Robotic Arm Horizontal Side Views 46

Figure 3.12. Robotic Arm Horizontal Front and Top Views 46

Figure 3.13. Robotic Arm Vertical Front and Top Views 46

Figure 3.14. Robotic Arm Vertical Side Views 47

Figure 3.15. Robotic Arm Vertical Orthogonal Views 47

Figure 3.16. Robotic Arm Vertical Orthogonal Back Views 48

Figure 3.17. Generated 3D Model in Simulink 49

Figure 3.18. Simulink Model of the Designed Robot 49

Figure 3.19. User Interface for the Inputs into Simulink Model 50

Figure 3.20. Workspace Estimation of the Designed Robot 51

Figure 3.21. Free Body Diagram of the Robotic Arm for Static Load Calculations 52

Figure 3.22. Master Distances of the Robot Based on the Free Body Diagram 53

Figure 3.23. General View of the Electronics Components 55

Figure 3.24. Motor Test Setup 55

Figure 3.25. Control Board Setup 56

Figure 3.26. System Hardware Topology Diagram 57

Figure 4.1. KUKA KR-5 ARC-C Robot Mechanical Body 58

Figure 4.2. Geometric Model of KUKA KR5 when Wrist is Above Shoulder 59

Figure 4.3. Geometric Model of KUKA KR5 when Wrist is Below Shoulder 60

Figure 4.4. KUKA KR5 Robotic Arm Orthogonal Views 61

Figure 4.5. KUKA KR5 Robotic Arm Side Views 61

Figure 4.6. KUKA KR5 Generated 3D Model in Simulink 62

Figure 4.7. Workspace Estimation of the Custom KUKA KR5 Robot 62

Figure 4.8. KUKA KR-5 ARC-C Robot with Motors Mounted 63

Figure 4.9. KUKA KR-5 ARC-C Robot Control Panel 64

Figure 5.1. Visual Simulation for the Linear Motion Along the Z-Axis 65

Figure 5.2. Graphs for the Linear Motion Along the Z-Axis from 599mm to 300mm 66

Figure 5.3. Visual Simulation for the Linear Motion Along the Z-Axis 66

Figure 5.4. Graphs for the Linear Motion Along the X-Axis from -300mm to +300mm 67

Figure 5.5. Visual Simulation for the Circular Motion on the Y-Z Plane 67

Figure 5.6. Graphs for the Circular Motion on the Y-Z Plane 68

Figure 5.7. Visual Simulation for the Circular Motion on the X-Y Plane 68

Figure 5.8. Graphs for the Circular Motion on the X-Y Plane 69

Figure 5.9. Visual Simulation for the Sinusoidal Motion on the X-Z Plane 69

Figure 5.10. Graphs for the Sinusoidal Motion on the X-Z Plane 70

Figure 5.11. Visual Simulation for the Diagonal Interplanar Motion 70
Figure 5.12. Graphs for the Diagonal Interplanar Motion.. 71
Figure 5.13. KUKA KR5 Visual Simulation for the Linear Motion Along the Z-Axis 72
Figure 5.14. KUKA KR5 Graphs for the Linear Motion Along the Z-Axis from 1100mm to -300mm...... 72
Figure 5.15. KUKA KR5 Visual Simulation for the Linear Motion Along the X-Axis.............................. 73
Figure 5.16. KUKA KR5 Graphs for the Linear Motion Along the X-Axis from 0mm to 800mm............. 73
Figure 5.17. KUKA KR5 Visual Simulation for the Circular Motion on the XY-Plane............................. 74
Figure 5.18. KUKA KR5 Graphs for the Circular Motion on the XY-Plane 74
Figure 5.19. Physical Test of KUKA KR5 for the Linear Motion Along the Z-Axis 75
Figure 5.20. Physical Demonstration of KUKA KR5 at an Expo 75

LIST OF TABLES

Page

Table 3.1. General Specifications of the Designed Robotic Manipulator 22

Table 3.2. Technical Specifications of Similar Industrial Robotic Arms 22

Table 3.3. Axis Specifications of the Designed Robotic Manipulator 23

Table 3.4. D-H Table of the Designed Robotic Manipulator 26

Table 3.5. Maximum Torque Requirements for each Motor 53

Table 3.6. Maximum Discrete Torque Requirements for each Motor 54

Table 3.7. Final Motor Configuration for the Robot 54

Table 5.1. Comparison Table for Different Robots 76

LIST OF APPENDICES

Page

Appendix-1: Work Plan 82

Appendix-2: Budget List........ 84

Appendix-3: MATLAB Codes for Forward Kinematics 85

Appendix-4: MATLAB Codes for Inverse Kinematics 89

Appendix-5: Matlab Codes for User Interface........ 93

Appendix-6: MATLAB Codes for Workspace Estimation........ 102

Appendix-7: MATLAB Codes to Simulate Continuous Motion and to Generate Graphs via Timeseries.. 104

Appendix-8: Technical Drawings of the Designed Robot 111

SYMBOLS AND ABBREVIATIONS

Symbols

A_i	: Homogeneous transformation matrix of the axis i
a_i	: Distance between joints
a_x	: Angle between Z_5 and X_0
a_y	: Angle between Z_5 and Y_0
a_z	: Angle between Z_5 and Z_0
C_i	: Cosine of the angle θ for the axis i
d_i	: Length of the link
n_x	: Angle between X_5 and X_0
n_y	: Angle between X_5 and Y_0
n_z	: Angle between X_5 and Z_0
o_x	: Angle between Y_5 and X_0
o_y	: Angle between Y_5 and Y_0
o_z	: Angle between Y_5 and Z_0
S_i	: Sine of the angle θ for the axis i
${}^{m}T_n$	: Homogeneous transformation matrix from the axis m to the axis n
α_i	: Angle between Z_{i-1} and Z_i
θ_i	: Rotation angle of the axis i which is the angle between X_{i-1} and X_i

Abbreviations

DoF	: Degree(s) of Freedom
D-H	: Denavit-Hartenberg
FK	: Forward Kinematics
G	: Gravitational Acceleratiton
HTM	: Homogeneous Transformation Matrix
IFR	: International Federation of Robotics
IK	: Inverse Kinematics
P	: Prismatic Joint
PID	: Proportional-Integral-Derivative
R	: Revolute Joint
TCP	: Tool Center Point

1. INTRODUCTION

An overall view of the problem, the solution offered within the scope of this study, main objectives and the general outline of the thesis are presented in this chapter.

Six-degrees-of-freedom (DoF) robotic manipulators are machines that resemble a human arm and can do most of the tasks a human arm can do. These machines can determine the required motor angles for all desired X, Y, Z coordinates in real world thanks to the high-level mathematical operations in the background and they are widely used in industry. To give a few examples, they can; tighten screws, pick-and-place objects from and to the specified positions, weld many surfaces, paint complex parts, and can carry many tasks that people do physically in industry. Figure 1.1 shows an example of robotic manipulators used in the automotive industry.

Figure 1.1. Industrial Robotic Arms in a Factory [1]

1.1. Statement of the Problem

As seen in figures in the previous section, robots are widely used in the world and their use keeps growing both by number and industries. Although the popularity and number of robotic manipulators continues to increase day by day, thanks to the many advantages they provide to the industry, the science behind this technology is not as well-known as the technology itself. Very few high-tech companies in the world can manufacture these machines, and these companies usually present their products in a closed box to prevent outside interference. This way, the system may not be intervened by unauthorized people in any way. Users mostly must follow the instructions given by the manufacturer and use the robot through the graphical user interface developed by the manufacturer itself. Figure 1.2 shows an example industrial user interface developed by the company STAUBLI to control and program their robots.

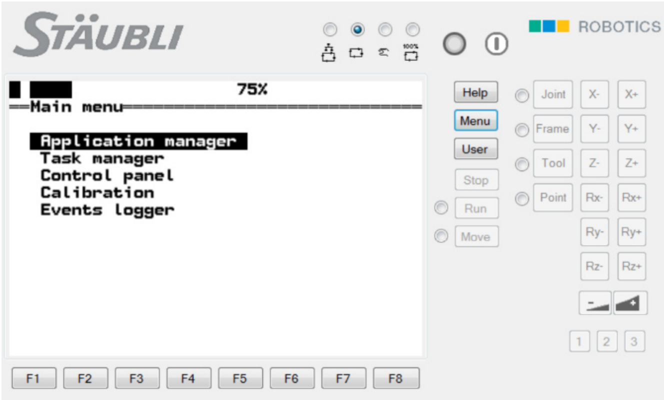


Figure 1.2. A Commercial User Interface to Control Industrial Robots by STAUBLI [2]

While the industry keeps using closed architectures understandably, the academic world is generally not focused on applied studies in this field. The academic literature, in addition to being difficult to properly grasp, is heavily focused on theoretical solutions with no prototype development and they are mostly based on academic toolboxes that limit the solution to a single environment. And this creates another barrier on the way for students and even academics to relate the theory to the real-world and fully grasp the subject.

Although robotic manipulators have been used extensively in the industry for many years, the human resource with the theoretical knowledge behind this technology and the applied experience in the field is very limited. Moreover; various reasons such as this technology being quite costly, requiring a serious amount of effort to understand the subject properly, and the robot manufacturers selling their products in a closed box for commercial concerns, keeping the user's intervention into the system very limited, slow down the progress of making the essence of this technology more accessible to wider communities.

1.2. Objectives

In order to contribute to making this technology more accessible to anyone professionally interested in robotics; development process of a 6-DoF robotic manipulator from the preliminary design phase to the fully-functioning prototype phase shall be executed within the scope of this study. The robotic manipulator prototype to be developed will have both an experimental function suitable for educational purposes and the ability to perform small-scale industrial tasks such as welding, painting, etc. when the proper end effector is attached to the wrist of the manipulator. The thesis with all details will be published and be accessible in online dissertation databases. In addition to this, results of the study, video explanations on the critical subjects, and further improvements on the model will be published on platforms such as YouTube and GitHub following the publication of the thesis.

Moreover, implementation of the solution on another industrial robot shall be accomplished. On this part of the study, an industrial robotic manipulator KUKA KR5's mechanical body will be used to implement the solution. Custom motors and electronic components shall be mounted on the mechanical body, and the theoretical solution and the control software to run this new custom KR5 robot shall be developed.

1.3. Scope of the Study

Within the scope of the study, the development of a desktop-size, 6 degrees-of-freedom robotic manipulator shall be accomplished which covers the following phases:

- Preliminary Robotic Design
- Forward Kinematic Analysis
- Inverse Kinematic Analysis
- Static Load Analysis
- Detailed 3D Mechanical Design
- Simulation Development
- Component Selection
- Procurement of the Components
- Manufacturing
- Assembly
- Control Software Development
- User Interface Development
- Optimization Studies
- Open-Source Publication (Github-YouTube)

Figure 1.3 is an example of a robotic manipulator of similar size to the robot to be developed.

Figure 1.3. A Commercial Desktop-size Robotic Manipulator [3]

As an extra study, the customization of an actual industrial robotic manipulator KUKA KR5 ARC-C which doesn't have any electronic components provided by the manufacturer shall be accomplished which covers the following phases:

- Preliminary Robotic Design
- Forward Kinematic Analysis
- Inverse Kinematic Analysis
- 3D Mechanical Design
- Simulation Development
- Real-Time Simulation Development on Hardware
- Component Selection
- Procurement of the Components
- Assembly
- Control Software Development
- User Interface
- Optimization Studies

A picture of the KUKA robot at the beginning of the study is shown in Figure 1.4.

Figure 1.4. An Industrial Robotic Arm KUKA KR5 ARC-C With no Electronic Components

1.4. Outline of the Thesis

This study consists of 5 chapters and extra sections for references and appendices. The contents of each chapter are follows. Chapter 1 contains a brief summary of the thesis regarding the problem, objectives, and the scope of the study. In Chapter 2, general information on robots, their advantages and types are presented then background in the field and the literature review are covered. Chapter 3 is where; overall method, preparation phase, all theoretical subjects, solutions and simulations, prototyping phase are explained in detail. Chapter 4 contains the details of the extra study on the KUKA KR5 robot which includes overall method, preparation phase, theoretical subjects, solutions and simulations, physical tests. Chapter 5 presents the results from the two robots developed within the scope of this study. In Chapter 6 conclusions, and recommendations for further improvements are introduced.

2. GENERAL INFORMATION AND LITERATURE REVIEW

In this section, general information about robots, their types, the focus of this study, the background in the field and academic literature review will be presented.

2.1. General Information

2.1.1. Definition of the Word “Robot”

The Robotic Industries Association (RIA) defines robot as follows:

"A robot is a reprogrammable, multifunctional manipulator designed to move materials, parts, tools or specialized devices through variable programmed motions for the performance of a variety of tasks." [4]

In simple terms; any machine that replaces human effort, is not dependent on the human input and can make decisions on their own can be called as a robot.

2.1.2. Advantages of Robots

Robots used in industry are commonly called industrial robots. These robots provide a wide variety of advantages to the business world. To summarize the benefits, these categories can be used:

- **Improvement on Quality**
  While they work at high speed, robots are also well-engineered to maintain a high level of accuracy and repeatability which ensures that a certain quality in the output product will be preserved each time. This also makes it possible for companies to spend less resources on the quality control processes.
- **Extended Working Hours**
  Since people can only work for a certain number of hours per day with limited physical and mental capacity and very volatile focus ability, robots provide an incomparable performance over people for certain tasks. They can work almost non-stop and maintain their physical and mental energies while not losing focus on the task.
- **Better Working Environment**
  Some indoor and outdoor tasks are considered inconvenient for humans as they are; dangerous, repetition-based, or labor-intensive. Robots can help improve working conditions for people.
- **Improvement on Efficiency**

Industrial robots are designed to carry specific tasks fast and accurately. In almost all cases, robots can do majority of the tasks much faster and much more accurately than humans. And this allows businesses to do the same tasks much faster while using less resources.

- **Improvement on Profitability**

  Since efficiency is improved, as a consecutive positive outcome, industrial robots can increase profit margins for companies. Using less resources while maintaining a decent level of quality decreases unit costs greatly.

- **Prestige**

  Having higher level of industrial capabilities is always a plus in the eye of customers and competitors. In addition to all others benefits robots provide companies, leaving a better impression on other business partners is another positive outcome of having robots within the business organization.

2.1.3. Statistics on Robots Used in Industry

Robotic manipulators are widely used in a great variety of sectors, especially in developed countries. Figure 2.1 and Figure 2.2 below present the number of industrial robots in countries with the highest use of robots and the distribution of robots by industries, according to 2017 data provided by International Federation of Robotics (IFR).

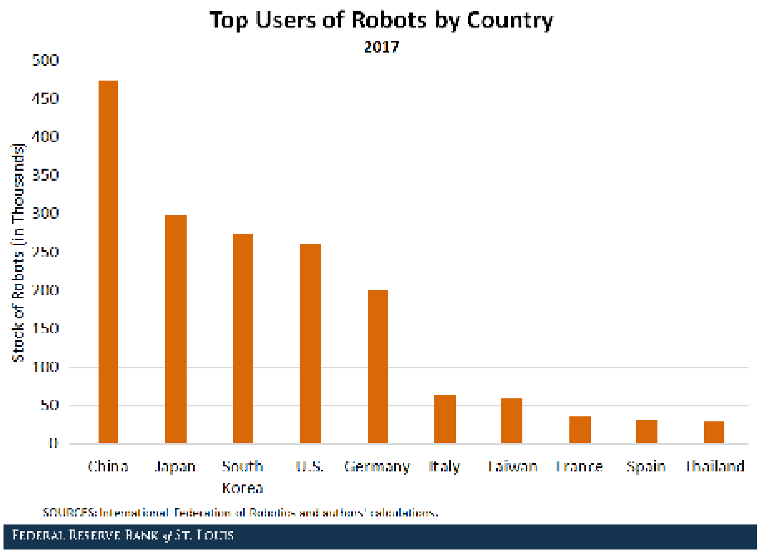


Figure 2.1. Top 10 Countries which Employ the Highest Number of Robots [5]

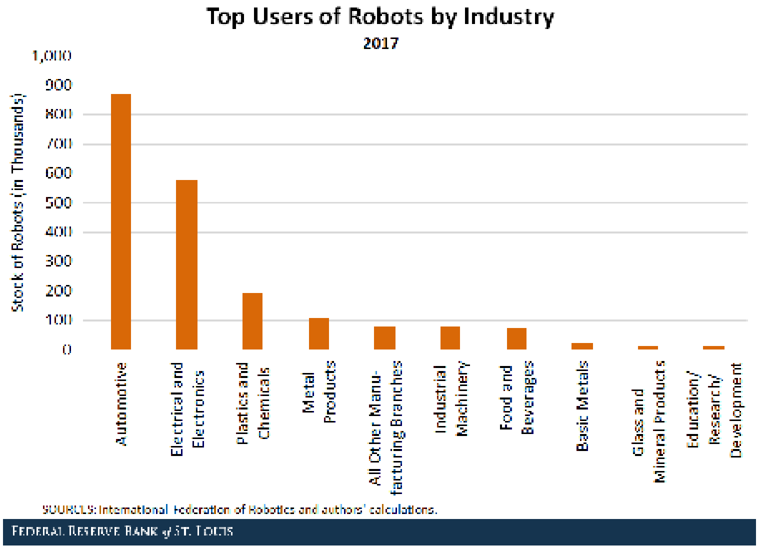


Figure 2.2. Top 10 Industries that Employ the Highest Number of Robots [5]

Below are the graphs regarding the current status and future forecasts on robots provided by IFR and presented in Press Conference World Robotics 2021 in Figure 2.3, Figure 2.4, and Figure 2.5.

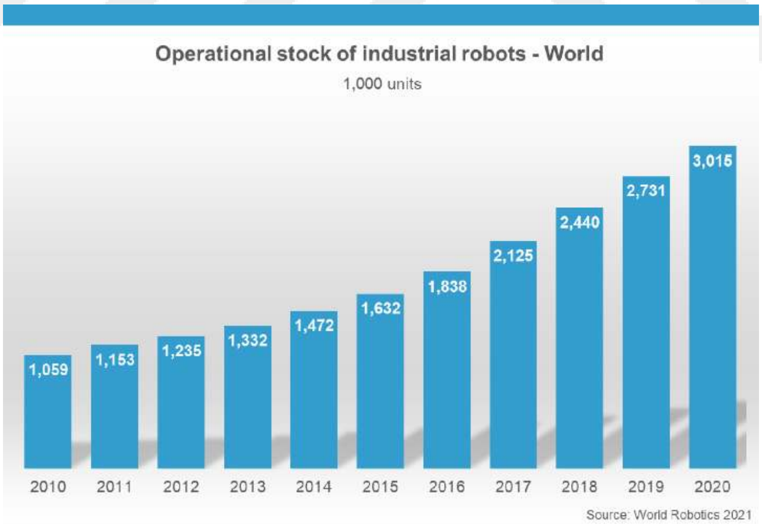


Figure 2.3. Operational Stock of Robots in the World [6, 7]

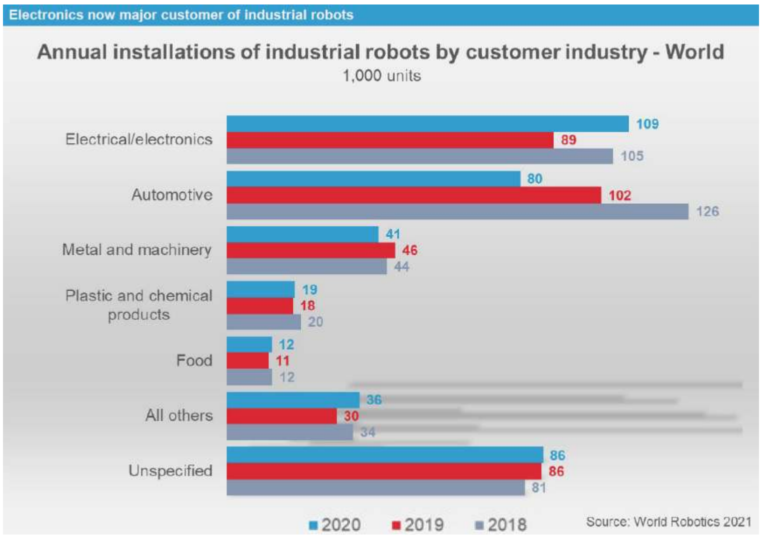


Figure 2.4. Annual Installation of Robots by Industry in the World between 2018-2020 [6, 7]

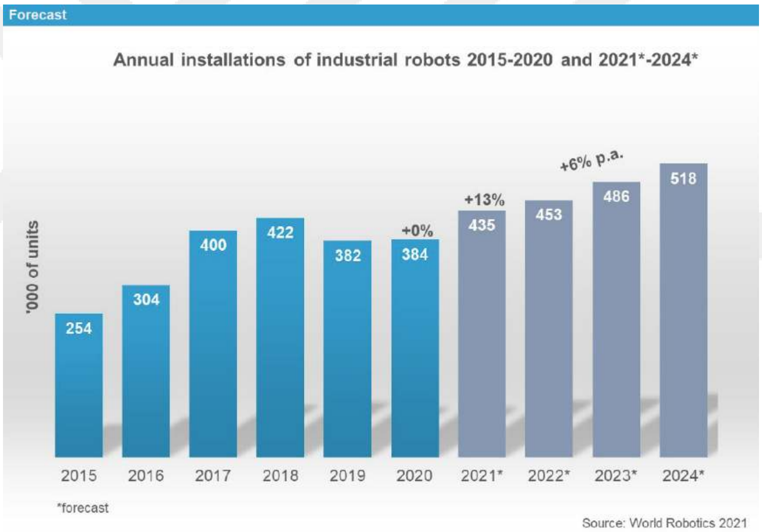


Figure 2.5. Annual Installation of Robots between 2015-2020 and Forecast for 2021-2024 [6, 7]

2.1.4. Types of Robots

Since the definition of the word "Robot" has a wide range, the types of robots also vary tremendously. Robot types can be divided into two main categories and many other sub categories. Some major types of robots are listed below. Since this study mainly focuses articulated robots, other types will not be explained in detail, instead they will be mentioned shortly.

1) Mobile Robots

- **Industrial Mobile Robots**
- **Aerospace Robots**

- **Consumer/Service Robots**
- **Telepresence Robots**
- **Research Robots and Others**

  Most robots are developed in university laboratories and R&D departments of companies[8–14]. This category consists of all other robot types that are not mentioned in this list and covers some of the types mentioned here also. It covers a great variety of robots like; snake-shaped robots, underwater robots, legged robots, entertainment robots, military robots, disaster response robot, etc. To encapsulate the robot types in general while keeping track of the focus of this study, many other sub categories will no mentioned here on purpose.

2) Stationary Robots

- **Cartesian/Linear Robots**
- **SCARA Robots**
- **Delta/Parallel Robots**
- **Spherical Robots**
- **Articulated Robots**

  This type of robots is one of the most widely used robot types as their design is pretty much like a human arm and they can do most tasks that people can do with their arm. They only have rotational axes and no prismatic/linear axis, yet they can still perform tasks that require linear motion. They might have 2 to 10 joints, or even more if necessary, depending on the purpose, but the most common types have either 4 or 6 degrees of freedom [15]. The term "degree(s) of freedom" stands for the number of joints on a robot that can move independently from other joints and it will be referred as "DoF" hereupon. A picture of an articulated robot is shown in Figure 2.6.

Figure 2.6. An Articulated Robot [16]

They can also be mounted on different mechanisms or robots to increase mobility and functionality such as linear trays that have 1-DoF, 3-DoF gantry robots, wheeled mobile robots, etc. Below are Figure 2.7 and Figure 2.8 that represent examples for different combinations of articulated robots.

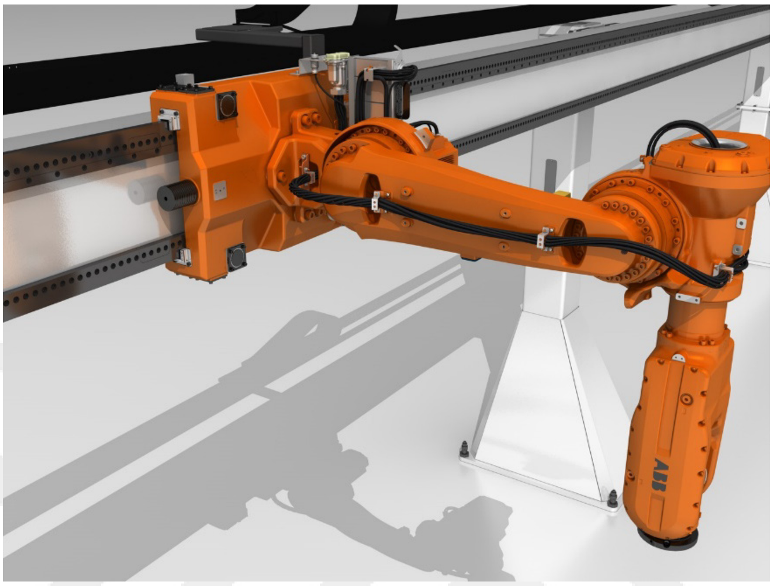


Figure 2.7. An Articulated Robot Mounted on a Linear Tray [17]

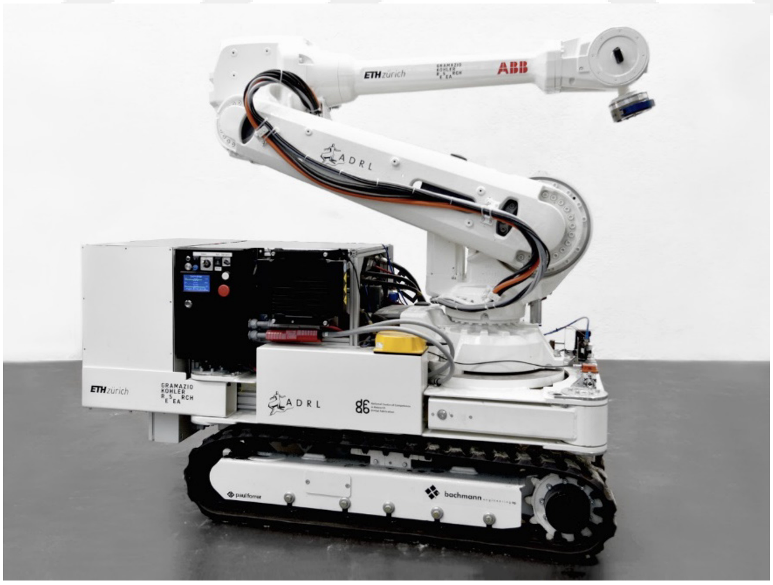


Figure 2.8. An Articulated Robot Mounted on a Mobile Robot [18]

2.1.5. Use of Articulated Robots

This type of robots are the most common ones and they are the focus of this study. Especially in industry, they are used very commonly in a variety of sectors and departments. Some major, but not all, use cases of articulated robots are presented below.

Welding

Welding is one of most popular areas for articulated robots since it requires high precision and consistency for a high volume of repetitive tasks, especially in the automotive industry. Two welding robots are presented in Figure 2.9 and Figure 2.10.

Figure 2.9. An Articulated Arc Welding Robot [19]

Figure 2.10. An Articulated Spot Welding Robot [20]

Painting

Painting is also a popular area for articulated robots to serve since it requires minimal resource usage and standard quality by nature. An example of a painting robot is shown in Figure 2.11.

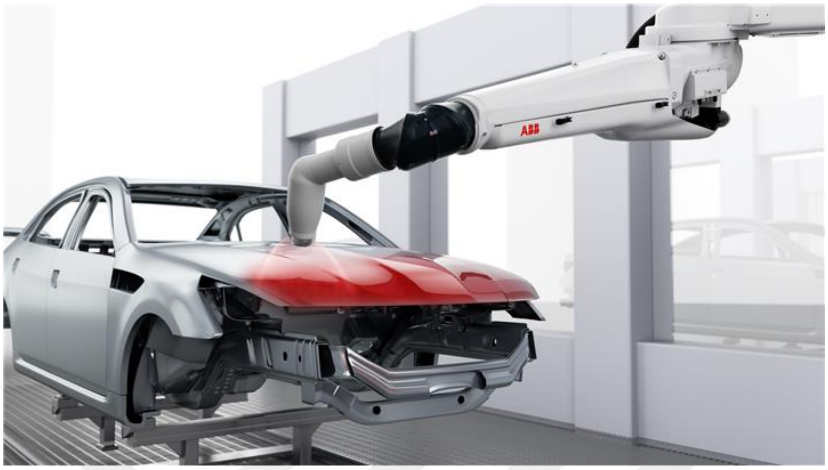


Figure 2.11. An Articulated Painting Robot [21]

Material Handling

In a factory, it’s one of the most common tasks to transport an item to the next phase of the production line. There are several transport-related tasks that robotic manipulators come in handy such as; sorting, packaging, unloading, palletizing, etc. Articulated robots can help people with these tasks successfully whether the load is big and heavy or small, non-uniformly shaped and light-weighted. These needs both apply to heavy or light-weight items. They can be used in automotive industry, food industry, medical industry, etc. An example of a material handling robot can be seen in Figure 2.12.

Figure 2.12. An Articulated Material Handling Robot [22]

Besides providing a more ergonomic working environment for people by carrying heavy loads, articulated robots can also be used for the dangerous tasks in various industries to provide a safer environment for people such as cutting metal parts, transporting extremely hot forged metals, various tasks in the foundry industry, etc. An articulated robot manipulator used in forging industry is shown in Figure 2.13.

Figure 2.13. An Articulated Robot in Forging Industry [23]

Assembly Lines

In many industries, the final product consists of numerous parts and the assembly phase of these parts cause a heavy load on the work force. In most cases, especially in the automotive industry, the assembly of these parts do not require human intervention and they are basically a group of high-volume repetitive tasks that need to be handled the same exact way each time. Thus, articulated robots provide a perfect solution for these operations. A demonstrative picture of this kind of use case is presented below in Figure 2.14.

Figure 2.14. Articulated Robots on a Production Line [24]

Surgery

Articulated robots can also be used for super-precise surgical operations. One of the best examples of these applications is the famous Da Vinci Surgical Robotic System developed by Intuitive Surgical. The system operates based on articulated robots that are equipped with very precise sensors and perception system and functions in the form of master and slave robots. The master robot is directly controlled by the surgeon in person and the slave robot imitates any motion that happens in the master robot at extremely high precision and very low latency at a distant place. The Da Vinci system is presented below in Figure 2.15 and Figure 2.16.

Figure 2.15. The Famous Articulated Surgical Robot Called "Da Vinci" - General View [25]

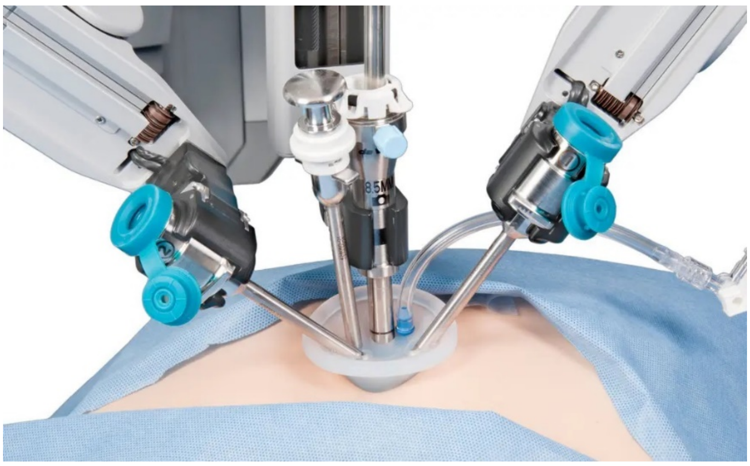


Figure 2.16. The Famous Articulated Surgical Robot Called "Da Vinci" - Close Shot [26]

Since articulated robots are very good at replacing human effort while providing serious advantages, they are widely used in industry and it’s not easy to cover all use cases. Only some of the major use cases have been explained in this section.

2.2. Background in the Field

With the development of digital control and remote-control concepts, important developments have occurred in robotics studies. The machine, which was remotely controlled by John Parson, was used by the United States Air Force in the 1940s and then by the Atomic Energy Commission. It was used in important processes in studies on radioactive materials. This system started to be used in industry. Together with Cyril Walter Kenward, they patented it in March 1954. Thus, a system that could be called the first industrial robot was designed.

One of the major developments was made by Joseph F. Engelberger (Physics Engineer) and George C. Devol. They made a part transfer robot called the father of robots. And as a result of their work, they established a company called "Unimate" 1949, this company is the first company established on robotics.

After this date, many companies around the world, especially in America, Europe, and Japan, started to work on robotics. And this interest increased the development with it. Among these developments, the first robot application language "WAVE" was developed by the Stanford Academy and brought to the science of robotics. In 1974, "AL" and commercially "VAL" were made. It was developed by Val Ultimate and accordingly "PUMA (Programmable Universal Machine for Assembly)" was developed and applied on it. It was a relatively short articulated robot, but was based on the General Motors assembly line. In 1979, "SCARA (Selective Compliance Arm for Robotic Assembly)" was developed by Yamanashi University for assembly purposes. This system was commercially introduced in 1981.

By the 90s, robots have taken their place in human life by doing flawless works in a wide variety of fields and especially those that people cannot easily do. I think robots are the best examples of these developments, such as assembly robots that can conduct research in deep waters where human beings cannot descend, that can work for years without stopping, surgical robots that prevent surgeons from making mistakes and are even ambitious enough to eliminate the surgical profession, and an artificial arm that can be fitted instead of a human arm.

But the noble development should be the robot dog "Aibo", who was stung to our homes by Sony. Although the robot has the capabilities of perception, decision making and application in the direction of the command, there are many features such as identifying its owner with a wide-angle memory: As a result, the work in robotics technology is now largely based on developments in computer technology. Although computers were available when the robotics industry was born, they were not suitable for robot control until the late 1970s due to their size. All robots in the market today are used computer control. However, it is certain that the field of robotics science still includes machine and computer science.

2.3. Related Academic Studies

G. Shouchao et al. (2022) proposed a novel 7 degrees-of-freedom (DoF) articulated manipulator with a special type of active spherical joint. Their claim is to achieve a better robotic design in terms of kinematic capability and workspace volume by using a design that is like a spherical-revolute-spherical kinematic chain [27].

A.H. Rajpar et al. (2021) developed a 3 DoF articulated robot that is controlled via Bluetooth using an Android device. The robot was designed to carry material handling tasks for very light objects. Along with the necessary mathematical calculations, they have also developed a prototype to prove their mathematical solution [28].

E. A. Chukwudi et al. (2020) proposed an H-infinity controller for a 3 DoF robotic manipulator. The purpose of the study was to achieve better stabilization for the robot. They designed an H-infinity controller along with a PID controller and compared the results. The H-infinity controller provided significantly better results than the PID controller in terms of disturbance rejection and noise rejection [29].

H. Karci and A. Tangel (2016) declared that they have developed a system where they can distinguish objects according to their size with ultrasonic sensors with a robot arm, they placed on a mobile vehicle. According to the data they received from the ultrasonic sensor, they stated that they performed the numerical design of the position of the axis robot arm that they designed with the position and speed control of the mobile tool with an FPGA. They declared that the motors on the robot arm on the mobile vehicle were DC motors and that they designed DC motor drivers, provided the speed of the motors and the VHDL code and PWM signals in their position control. In these studies, they used the frequency division technique for PWM signals, and they indicated the control tags of the units on the mobile robot at the same time by making use of the parallel operations of the FPGA [30].

S. Sabarigirish and S.J. Mija (2016) presented the determination of the trajectory between the start and end points when obstacles were created for a robot manipulator with 5 degrees-of-freedom. Trajectory planning is one of the most important aspects of the selection and placement task performed by robotic manipulators. Various possibilities are considered for the robotic manipulator with 5-DoF when the barriers are created. An algorithm has been created for situations with and without obstacles. The orbital motion of the robot is determined using a cubic polynomial. For this purpose, 5-DOF Vi Microsystems Robots have been modeled and analyzed. Different paths were obtained so that the obstacles do not pass through the same points depending on their location, and the trajectory determination algorithm and calculations were made using MATLAB simulations [31].

Z. Xuefeng et al. (2015) focused on the rehabilitation of excitatory limbs, nonlinear and uncertainty problems in robot trajectory tracking control. Considering that the rehabilitation robot system should be done on the repetitive task and repetitive learning control features, a nonlinear saturation function class is introduced and a nonlinear iterative learning control algorithm is proposed. Linear PID robot dynamics control, which is widely used, is used in this algorithm. Good quality control is achieved under conditions where model information is not accurate and only position feedback can be measured. The Lyapunov stability theory, together with the rehabilitation robot dynamics model features, was applied to prove the global asymptotic stability of the closed loop system. The five degrees of freedom simulation results of the rehabilitation robot system show that the proposed nonlinear iterative learning control is a good control [32].

In the study of F. Pehlivan et al. in 2015; They stated that they obtained the inverse and flat kinematic equations of the AL5A robot and achieved this with a geometric approach. They stated that they made kinematic calculations in tracking the tip trajectory of the robot. They also used PID controllers in trajectory tracking and had them tracked precisely [33].

C.Y. Tsai et al. (2015) addressed the servo control issue for a robot manipulator with 5 degrees of freedom to perform the pick and drop task. A new hybrid switched reagent-based visual servo control structure, inspired by a sensing and reaction behavior without counter-interaction matrix computation, is proposed to effectively address this issue. The proposed structure is like logic-based approaches, but fuzzy modeling and/or fuzzy rule-based learning requirements are ignored. To achieve this, a new hardware sensitive reactive function approach is introduced to directly map an image position error vector to a desired end effector velocity vector, considering hardware limitations. This approach helps simplify visual servo systems with improved reliability. Also, the proposed control system is a hybrid switch controller that includes both image-based and position-based reactive planning schemes, allowing to increase the durability and efficiency of the visual servo system. [34].

C. D. Bellicoso et al. (2015) carried out a study on design, modeling, and control of a light robot manipulator with 5 degrees of freedom. The robot arm, called Prisma Ultra-Lightweight 5-arm (PUL5AR), is designed to perform vertical take-off and manipulation missions equipped with unmanned aerial vehicle landing gear. The handle is compact and lightweight. Its mechanical parts are designed to fold itself during landing maneuvers. In addition, this design limits the arm's center of gravity to the vehicle floor as much as possible, reducing the overall inertia and static instability of the system. [35].

R. Amir and K. Reihaneh (2015) used a fuzzy motion method controller for position tracking of 5 DOF upper limb exoskeleton robots. In the proposed control method, firstly, the control input is designed using the inverse dynamic method. Then PID control was used to eliminate the undesirable effects of uncertainties and it was mathematically proven that the closed loop system is asymptotically stable. Finally, T-S fuzzy motion method has been applied to prevent undesirable behavior events compared to the classical motion method. In the simulation results; It has been observed that the fuzzy motion method gives the desired result. [36].

3. MATERIAL AND METHOD

In this chapter, information on all resources and materials used for the study will be provided and all methods used will be explained in detail.

3.1. Preparation for Development

Before performing the robotic design and development, it is necessary to act in accordance with certain criteria that should be done in general regarding the robot development. A list of these criteria is as follows;

- Determining the purpose of the robot to be designed
- Determination of suitable sensors
- Calculations for inverse kinematics solution
- Designing a detailed CAD model of the robot
- Simulation of the robot to test inverse kinematic solution
- Development of algorithm for the robot
- Designing an electronic circuit for the robot
- Software development for the robot
- Assembling and testing the parts of the designed robot
- Detection and elimination of errors during the tests

Robot arms are widely used in almost every field of the industrial industry. People have made their life easier by building special robots such as a painter robot arm designed to paint the products in the paint shops, a robot arm designed for packing, a robot arm designed for material handling operations, a robot arm capable of separating the parts very quickly according to their colors in mass production. The robot arm to be designed in this project is meant to carry an object from one point to another using an attached gripper to the end of the robot. But it is also perfectly capable of carrying various tasks if the tool that is attached at the end is below the maximum payload capacity.

Once the purpose of the robot arm is specified, it is necessary to decide what elements should be detected in its surroundings (color separation, temperature value detection, humidity value detection, proximity, etc.). The robot arm uses sensors to provide the ability to make decisions in line with the reactions from its environment in accordance with this purpose and to perform its task in line with these decisions.

Then necessary calculations must be performed to obtain a valid theoretical inverse kinematics solution which is explained in detail in the following sections.

Another very important step when developing a robot is the algorithm. The algorithm is a logical sequence of operations that the robot will perform. The algorithm is of great importance when deciding what inputs to be taken from the human operator, what will happen in urgency cases, what objects around it should be detected, how many sensors to use, and which of them, if any, will function as the primary feedback, etc.

The design of the robot arm should be made in the simplest way by choosing the most suitable material for its purpose, without complicating the manufacturing and assembly stages. If there are some criteria that the designed robot must comply with, considering these criteria (payload, TCP, dimensions of the robot arm, etc.) provides a correct design. Motor selection should be made according to the designed material. For the mechanical part of the robot arm to be designed correctly, the motor must first be selected. While choosing the motor, it is necessary to determine the rotation speed of the motor, the current it draws, the operating voltage and the size of the motor for the robot arm design.

An electronic circuit design should be made in order to make it physically possible for the system to perform the desired tasks. This designed circuit; It will prepare the necessary infrastructure for functions such as transferring the data from the sensors to the microcontroller, driving the motors through the microcontroller, power supplying the motor drivers, and operator intervention.

One of the most important steps that enables the robotic arm to work as desired is the algorithmic structure of the robot and the effective use of the selected microcontroller. Each programming language and microcontroller has its own characteristics and therefore may require different commands. Before starting this software design, preliminary research on the microcontroller and programming language ensures that good software can be developed. Then, the obtained theoretical solution is transferred to the microcontroller and physical tests on unassembled components are performed.

If no problems are encountered in the above-mentioned steps, it is possible to proceed to the assembly of the mechanical parts, electronic components, motors, and sensors of the robot to the robot arm. The components that were tested without assembly before, are now performed with the robot assembled.

After the robot is assembled, more detailed physical tests are carried out and if a problem is encountered, the cause of the error is determined, the problem is resolved and retests are performed.

Before starting to work on the development of the robotic manipulator a detailed work plan has been set based on time and work packages to keep track of the study. The detailed work plan is presented in Appendix-1.

3.1.1. Technical Specifications

Technical specifications of the designed manipulator can be found below in Table 3.1:

Table 3.1. General Specifications of the Designed Robotic Manipulator

Technical Specifications		
Assembly Type	Desktop, Ground	-
Number of Axes	6	-
Rated Reach	550	mm
Maximum Reach	950	mm
Rated Payload	2	kg
Maximum Payload	3	kg
Total Weight	20	kg

To have comparable models in the same class as the designed robot within the scope of this study, technical specifications of other industrial robots are listed below in Table 3.2:

Table 3.2. Technical Specifications of Similar Industrial Robotic Arms [37–40]

Brand	Model	No. of Axes	Weight [kg]	Payload [kg]	Maximum Reach [mm]
Staubli	TX2- 40	6	29	2	515
Staubli	TX2-60L	6	53	3.7	920
Kuka	KR 3 R540	6	26.5	3	541
Kuka	KR 6 R900-2	6	55	6.7	901
Kawasaki	RS003N-A	6	20	3	620
Kawasaki	RS005L-A	6	37	5	903

Axis specifications of the robot with physical restrictions and rated velocities are presented in Table 3.3:

Table 3.3. Axis Specifications of the Designed Robotic Manipulator

Axis Specifications		
	Angular Displacement (deg)	Rated Velocity (deg/s)
Axis 1	-170 / +170	60
Axis 2	-60 / +120	60
Axis 3	-150 / +30	60
Axis 4	-355 / +355	120
Axis 5	-130 / +130	120
Axis 6	-355 / +355	120

3.2. Method and Theoretical Solution

Quick summary of the method is given in this section and related details are covered below under subheadings.

In robot manipulation, advanced mathematical expressions are used to describe the relationship between the internal physical components (links and joints) of a robot and the relationship between the robot's end effector and object the robot is supposed to manipulate. Although there are several methods to follow, only the most widely used method will be mentioned here, the well-known Denavit-Hartenberg notation. This notation provides a very convenient way of expressing the positions and orientations of robot elements in relation to each other. Using this method, how far an object is translated from another object, or how much they are rotated in any axis may be expressed mathematically.

Once the identification of the overall motion-related parameters separately for each axis using the D-H notation is complete, then the next step is taken to define all robot elements at once as a whole system to find position and orientation of a specific part of the robot in reference to any other part of the robot using the input angles of each joint and this process will be referred as Forward Kinematics.

After having a successful forward kinematics model completed, the next step will be the real core problem in robotics which is finding required angles or translations to satisfy the desired position and orientation of the end effector. This part is much harder to solve in most cases than the previous steps and it is called Inverse Kinematics.

Then 2D and 3D models shall be crated to represent the physical robot on computer to run multiple tests and conduct improvements, if necessary, on the inverse kinematic solution in simulation phase till it's proven fully functional. Then the prototype development will begin.

3.2.1. Denavit-Hartenberg Notation

Denavit-Hartenberg notation is a common way to solve inverse kinematics problems. It's basically a compact way of defining homogeneous transformation matrices for all rotation and linear motion types [41].

D-H notation is simply a multiplication of 4 transformation matrices that act on a single joint. These 4 matrices describe; rotation and translation around and along the Z and X axes of the joint. Equation 3.1 and Equation 3.4 represents the rotation matrices around the Z and X axes in the same order. Equation 3.2 and Equation 3.3 stands for the translation matrices along the Z and X axes in the same order.

$$R_{z,\theta_i} = \begin{bmatrix} Cos(\theta_i) & -Sin(\theta_i) & 0 & 0 \\ Sin(\theta_i) & Cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 3.1$$

$$T_{z,d_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 3.2$$

$$T_{x,a_i} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 3.3$$

$$R_{x,\alpha_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & Cos(\alpha_i) & -Sin(\alpha_i) & 0 \\ 0 & Sin(\alpha_i) & Cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 3.4$$

Equations above are used to describe motion of an object, but robotic manipulators are quite complex and generally have multiple kinds of motions occurring simultaneously. So, it's a must to consider all kinds of motions that are possible for a joint at the same time. That's where the famous D-H notation comes in. It's a tool that combines all these separate transformation matrices and creates a single unifying matrix called Homogeneous Transformation Matrix (HTM) which covers all possible motions for the joint.

$$^{i-1}A_i = R_{z,\theta_i}\, T_{z,d_i} T_{x,a_i} R_{x,\alpha_i} \quad 3.5$$

$$
{}^{i-1}A_i = \begin{bmatrix} Cos(\theta_i) & -Sin(\theta_i) & 0 & 0 \\ Sin(\theta_i) & Cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & Cos(\alpha_i) & -Sin(\alpha_i) & 0 \\ 0 & Sin(\alpha_i) & Cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 3.6
$$

$$
{}^{i-1}A_i = A_i = T_i = HTM \quad 3.7
$$

$$
A_i = \begin{bmatrix} Cos(\theta_i) & -Sin(\theta_i)Cos(\alpha_i) & Sin(\theta_i)Sin(\alpha_i) & a_iCos(\theta_i) \\ Sin(\theta_i) & Cos(\theta_i)Cos(\alpha_i) & -Cos(\theta_i)Sin(\alpha_i) & a_iSin(\theta_i) \\ 0 & Sin(\alpha_i) & Cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 3.8
$$

Here in the homogeneous transformation matrix, 9 cells ranging from the first row and first column to the third row and third column represent rotation of the axis only and it will be referred as Rotation Matrix will be expressed as R_i hereupon. 3 cells in the last column up to the third row represent translation or position of the axis and it will be referred as Position Vector and will be expressed as P_i from now onward. The first 3 values in the fourth row represent the Perspective Transform while the element in the cell where the last column and the last row coincide represents the Scaling Factor and these two terms will receive little attention throughout this study since they are not the core elements in the HTM [42].

$$
R_i = \begin{bmatrix} Cos(\theta_i) & -Sin(\theta_i)Cos(\alpha_i) & Sin(\theta_i)Sin(\alpha_i) \\ Sin(\theta_i) & Cos(\theta_i)Cos(\alpha_i) & -Cos(\theta_i)Sin(\alpha_i) \\ 0 & Sin(\alpha_i) & Cos(\alpha_i) \end{bmatrix} \quad 3.9
$$

$$
P_i = \begin{bmatrix} a_iCos(\theta_i) \\ a_iSin(\theta_i) \\ d_i \end{bmatrix} = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \quad 3.10
$$

Using the DH notation, a special matrix called homogeneous transformation matrix as shown in Equation 3.8 is obtained. This is a standard matrix to be used for any kind of robotic manipulator. Inside the HTM, there are some special parameters used to define translations and rotations.

θ_i: Angle between X_{i-1} and X_i around axis Z_{i-1}

d_i: Distance between X_{i-1} and X_i along axis Z_{i-1}

a_i: Distance between Z_{i-1} and Z_i along axis X_i

α_i: Angle between Z_{i-1} and Z_i around axis X_i

In addition to countless amount of text-based resources, some visual resources that explain the D-H parameters are also available online [43]. Using the D-H notation, coordinate frames on the robot's joints shall be assigned. Z axes will be the primary axes that represent motion of the joint and they can be assigned freely depending on the robot's desired configuration to satisfy certain motion capabilities. For X axes, X_i must be normal to Z_{i-1}. Y axes will obey the right-hand rule when assigning. In accordance with the D-H notation's requirements, all coordinate frames are assigned as in Figure 3.1.

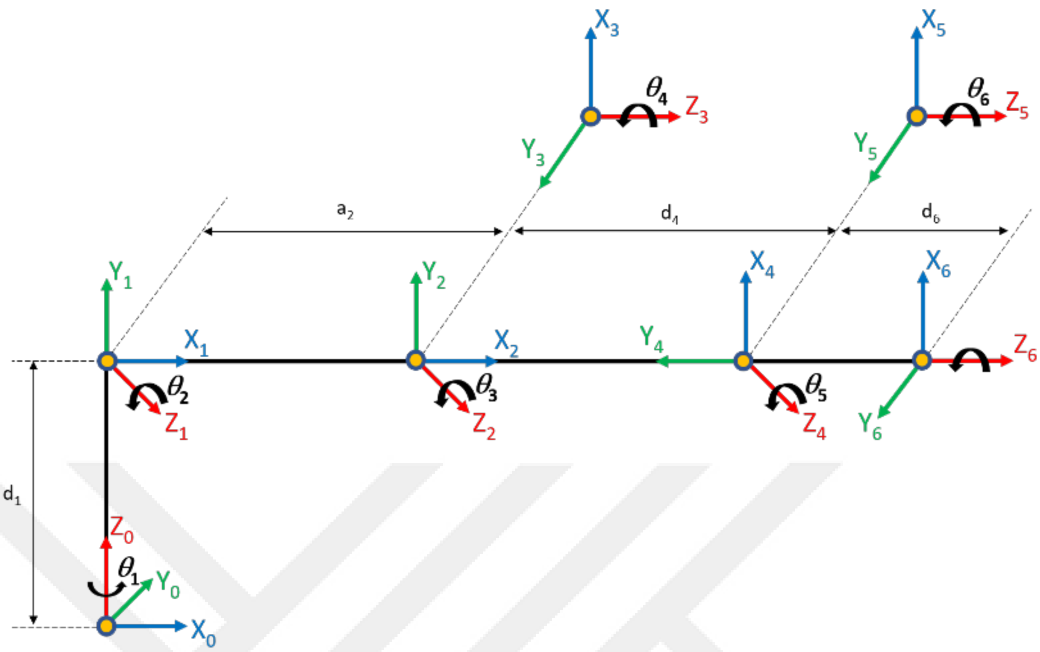


Figure 3.1. Axis Assignment of the Designed Robotic Manipulator

Once all frames are assigned, then it is possible to create a special parameter table to cover all motion-related properties of the robot in a single table. Using the assigned frames, D-H parameter table is extracted as in Table 3.4.

Table 3.4. D-H Table of the Designed Robotic Manipulator

LINK	$\theta(rad)$	d(mm)	a(mm)	$\alpha(rad)$
1	θ_1	d_1	0	$\frac{\pi}{2}$
2	θ_2	0	a_2	0
3	$\theta_3 + \frac{\pi}{2}$	0	0	$\frac{\pi}{2}$
4	θ_4	d_4	0	$-\frac{\pi}{2}$
5	θ_5	0	0	$\frac{\pi}{2}$
6	θ_6	d_6	0	0

Once the D-H parameter table is obtained then forward kinematic analysis may be conducted.

3.2.2. Forward Kinematics

The forward kinematic analysis method makes it possible to determine the position by means of serial matrix multiplications for cases where the angle and orientation values of the robot's axes are known. Based on the D-H table, which is created according to Denavit-Hartenberg notation and defines the motion capabilities of all axes of the robot, position information of any point on the robot can be obtained if the relevant angle values are given as input. The most common use of this method is to obtain the position information of the wrist center or tool center point (TCP) of the robot arm. Since the input of this method is the angle values of the robot that are not relevant to the purpose in real life, it draws little attention. Yet, it is still essential as the forward kinematic analysis is required to be able to start working on the inverse kinematic analysis which is much more useful in real life.

Based on the D-H table created in the previous section, a homogenous transformation matrix will be obtained for each axis of the robotic manipulator. In total, 6 HTMs will be present and they will be serially connected to each other to create the mathematical expression to define all motions of the whole body by matrix multiplications [44]. The process to conduct forward kinematic analysis on the robotic arm is explained below.

Now the HTM for each axis shall be calculated and named as $A_1, A_2, A_3, A_4, A_5, A_6$. These homogeneous transformation matrices may be shown using different notations such as $A_i = {}^{i-1}A_i = {}^{i-1}T_i$ which can appear in the text as $A_1 = {}^{0}A_1 = {}^{0}T_1$ or $A_6 = {}^{5}A_6 = {}^{5}T_6$. Using the standard form in Equation 3.8, HTM of A_i for each axis is derived as shown below.

$$A_1 = \begin{bmatrix} Cos(\theta_1) & 0 & Sin(\theta_1) & 0 \\ Sin(\theta_1) & 0 & -Cos(\theta_1) & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 3.11$$

$$A_2 = \begin{bmatrix} Cos(\theta_2) & -Sin(\theta_2) & 0 & a_2Cos(\theta_2) \\ Sin(\theta_2) & Cos(\theta_2) & 0 & a_2Sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 3.12$$

$$A_3 = \begin{bmatrix} Cos(\theta_3 + \frac{\pi}{2}) & 0 & Sin\left(\theta_3 + \frac{\pi}{2}\right) & 0 \\ Sin(\theta_3 + \frac{\pi}{2}) & 0 & -Cos\left(\theta_3 + \frac{\pi}{2}\right) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 3.13$$

$$A_4 = \begin{bmatrix} Cos(\theta_4) & 0 & -Sin(\theta_4) & 0 \\ Sin(\theta_4) & 0 & Cos(\theta_4) & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 3.14$$

$$A_5 = \begin{bmatrix} Cos(\theta_5) & 0 & Sin(\theta_5) & 0 \\ Sin(\theta_5) & 0 & -Cos(\theta_5) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 3.15$$

$$A_6 = \begin{bmatrix} Cos(\theta_6) & -Sin(\theta_6) & 0 & 0 \\ Sin(\theta_6) & Cos(\theta_6) & 0 & 0 \\ 0 & 0 & 1 & \mathrm{d}_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 3.16$$

These A_i matrices define the transformation of each axis. But they need to be serially connected so that their relative motion to each other can be represented in a single matrix. This matrix shall be called as the overall homogeneous transformation matrix of the robot from base to end effector and use the notation ${}^0T_6$ to represent it. Since this matrix will have many components, it will be more convenient to represent it in the proper notation symbolically first, then give the details below. As explained in the previous chapter, here in Equation 3.17, $r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23}, r_{31}, r_{32}, r_{33}$ represent the rotational part of the homogeneous transformation matrix only. They are mainly responsible for orientation. P_x, P_y and P_z however, stand for the positions in related axes X , Y and Z.

$${}^0T_6 = A_1A_2A_3A_4A_5A_6 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \mathrm{P_x} \\ r_{21} & r_{22} & r_{23} & \mathrm{P_y} \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 3.17$$

In which;

$$\boldsymbol{r_{11}} = \mathrm{Sin}(\theta_6)\big(\mathrm{Cos}(\theta_4)\mathrm{Sin}(\theta_1) + \mathrm{Cos}(\theta_1)\mathrm{Cos}(\theta_2)\mathrm{Sin}(\theta_3)\mathrm{Sin}(\theta_4) + \mathrm{Cos}(\theta_1)\mathrm{Cos}(\theta_3)\mathrm{Sin}(\theta_2)\mathrm{Sin}(\theta_4)\big) - \mathrm{Cos}(\theta_6)\big(\mathrm{Cos}(\theta_2 + \theta_3)\mathrm{Cos}(\theta_1)\mathrm{Sin}(\theta_5) - \mathrm{Cos}(\theta_5)\mathrm{Sin}(\theta_1)\mathrm{Sin}(\theta_4) + \mathrm{Cos}(\theta_1)\mathrm{Cos}(\theta_2)\mathrm{Cos}(\theta_4)\mathrm{Cos}(\theta_5)\mathrm{Sin}(\theta_3) + \mathrm{Cos}(\theta_1)\mathrm{Cos}(\theta_3)\mathrm{Cos}(\theta_4)\mathrm{Cos}(\theta_5)\mathrm{Sin}(\theta_2)\big) \quad 3.18$$

$$\boldsymbol{r_{12}} = \mathrm{Cos}(\theta_6)\big(\mathrm{Cos}(\theta_4)\mathrm{Sin}(\theta_1) + \mathrm{Cos}(\theta_1)\mathrm{Cos}(\theta_2)\mathrm{Sin}(\theta_3)\mathrm{Sin}(\theta_4) + \mathrm{Cos}(\theta_1)\mathrm{Cos}(\theta_3)\mathrm{Sin}(\theta_2)\mathrm{Sin}(\theta_4)\big) + \mathrm{Sin}(\theta_6)\big(\mathrm{Cos}(\theta_2 + \theta_3)\mathrm{Cos}(\theta_1)\mathrm{Sin}(\theta_5) - \mathrm{Cos}(\theta_5)\mathrm{Sin}(\theta_1)\mathrm{Sin}(\theta_4) + \mathrm{Cos}(\theta_1)\mathrm{Cos}(\theta_2)\mathrm{Cos}(\theta_4)\mathrm{Cos}(\theta_5)\mathrm{Sin}(\theta_3) + \mathrm{Cos}(\theta_1)\mathrm{Cos}(\theta_3)\mathrm{Cos}(\theta_4)\mathrm{Cos}(\theta_5)\mathrm{Sin}(\theta_2)\big) \quad 3.19$$

$$\boldsymbol{r_{13}} = \mathrm{Cos}(\theta_2 + \theta_3)\mathrm{Cos}(\theta_1)\mathrm{Cos}(\theta_5) - \mathrm{Sin}(\theta_5)\big(\mathrm{Cos}(\theta_1)\mathrm{Cos}(\theta_2)\mathrm{Cos}(\theta_4)\mathrm{Sin}(\theta_3) - \mathrm{Sin}(\theta_1)\mathrm{Sin}(\theta_4) + \mathrm{Cos}(\theta_1)\mathrm{Cos}(\theta_3)\mathrm{Cos}(\theta_4)\mathrm{Sin}(\theta_2)\big) \quad 3.20$$

$$\boldsymbol{P_x} = \mathrm{d}_4\mathrm{Cos}(\theta_2 + \theta_3)\mathrm{Cos}(\theta_1) + \mathrm{a}_2\,\mathrm{Cos}(\theta_1)\mathrm{Cos}(\theta_2) + \mathrm{d}_6\mathrm{Cos}(\theta_2 + \theta_3)\mathrm{Cos}(\theta_1)\mathrm{Cos}(\theta_5) + \mathrm{d}_6\mathrm{Sin}(\theta_1)\mathrm{Sin}(\theta_4)\mathrm{Sin}(\theta_5) - \mathrm{d}_6\mathrm{Cos}(\theta_1)\mathrm{Cos}(\theta_2)\mathrm{Cos}(\theta_4)\mathrm{Sin}(\theta_3)\mathrm{Sin}(\theta_5) - \mathrm{d}_6\mathrm{Cos}(\theta_1)\mathrm{Cos}(\theta_3)\mathrm{Cos}(\theta_4)\mathrm{Sin}(\theta_2)\mathrm{Sin}(\theta_5) \quad 3.21$$

$$\boldsymbol{r_{21}} = \mathrm{Sin}(\theta_6)\big(\mathrm{Cos}(\theta_2)\mathrm{Sin}(\theta_1)\mathrm{Sin}(\theta_3)\mathrm{Sin}(\theta_4) - \mathrm{Cos}(\theta_1)\mathrm{Cos}(\theta_4) + \mathrm{Cos}(\theta_3)\mathrm{Sin}(\theta_1)\mathrm{Sin}(\theta_2)\mathrm{Sin}(\theta_4)\big) - \mathrm{Cos}(\theta_6)\big(\mathrm{Cos}(\theta_2 + \theta_3)\mathrm{Sin}(\theta_1)\mathrm{Sin}(\theta_5) + \mathrm{Cos}(\theta_1)\mathrm{Cos}(\theta_5)\mathrm{Sin}(\theta_4) + \mathrm{Cos}(\theta_2)\mathrm{Cos}(\theta_4)\mathrm{Cos}(\theta_5)\mathrm{Sin}(\theta_1)\mathrm{Sin}(\theta_3) + \mathrm{Cos}(\theta_3)\mathrm{Cos}(\theta_4)\mathrm{Cos}(\theta_5)\mathrm{Sin}(\theta_1)\mathrm{Sin}(\theta_2)\big) \quad 3.22$$

$$\boldsymbol{r_{22}} = \mathrm{Cos}(\theta_6)\big(\mathrm{Cos}(\theta_2)\mathrm{Sin}(\theta_1)\mathrm{Sin}(\theta_3)\mathrm{Sin}(\theta_4) - \mathrm{Cos}(\theta_1)\mathrm{Cos}(\theta_4) + \mathrm{Cos}(\theta_3)\mathrm{Sin}(\theta_1)\mathrm{Sin}(\theta_2)\mathrm{Sin}(\theta_4)\big) + \mathrm{Sin}(\theta_6)\big(\mathrm{Cos}(\theta_2 + \theta_3)\mathrm{Sin}(\theta_1)\mathrm{Sin}(\theta_5) + \mathrm{Cos}(\theta_1)\mathrm{Cos}(\theta_5)\mathrm{Sin}(\theta_4) + \mathrm{Cos}(\theta_2)\mathrm{Cos}(\theta_4)\mathrm{Cos}(\theta_5)\mathrm{Sin}(\theta_1)\mathrm{Sin}(\theta_3) + \mathrm{Cos}(\theta_3)\mathrm{Cos}(\theta_4)\mathrm{Cos}(\theta_5)\mathrm{Sin}(\theta_1)\mathrm{Sin}(\theta_2)\big) \quad 3.23$$

$$\boldsymbol{r_{23}} = \mathrm{Cos}(\theta_2 + \theta_3)\mathrm{Cos}(\theta_5)\mathrm{Sin}(\theta_1) - \mathrm{Sin}(\theta_5)\big(\mathrm{Cos}(\theta_1)\mathrm{Sin}(\theta_4) + \mathrm{Cos}(\theta_2)\mathrm{Cos}(\theta_4)\mathrm{Sin}(\theta_1)\mathrm{Sin}(\theta_3) + \mathrm{Cos}(\theta_3)\mathrm{Cos}(\theta_4)\mathrm{Sin}(\theta_1)\mathrm{Sin}(\theta_2)\big) \quad 3.24$$

$$\boldsymbol{P_y} = d_4\left(Cos(\theta_2)Sin(\theta_1)Sin\left(\theta_3 + \frac{pi}{2}\right) + Cos\left(\theta_3 + \frac{pi}{2}\right)Sin(\theta_1)Sin(\theta_2)\right) - d_6\left(Sin(\theta_5)\left(Cos(\theta_1)Sin(\theta_4) - Cos(\theta_4)\left(Cos(\theta_2)Cos\left(\theta_3 + \frac{pi}{2}\right)Sin(\theta_1) - Sin(\theta_1)Sin(\theta_2)Sin\left(\theta_3 + \frac{pi}{2}\right)\right)\right) - Cos(\theta_5)\left(Cos(\theta_2)Sin(\theta_1)Sin\left(\theta_3 + \frac{pi}{2}\right) + Cos\left(\theta_3 + \frac{pi}{2}\right)Sin(\theta_1)Sin(\theta_2)\right)\right) + a_2\,Cos(\theta_2)Sin(\theta_1) \quad 3.25$$

$$\boldsymbol{r_{31}} = -\mathrm{Cos}(\theta_6)\big(\mathrm{Sin}(\theta_2 + \theta_3)\mathrm{Sin}(\theta_5) - \mathrm{Cos}(\theta_2 + \theta_3)\mathrm{Cos}(\theta_4)\mathrm{Cos}(\theta_5)\big) - \mathrm{Cos}(\theta_2 + \theta_3)\mathrm{Sin}(\theta_4)\mathrm{Sin}(\theta_6) \quad 3.26$$

$$\boldsymbol{r_{32}} = \mathrm{Sin}(\theta_6)\big(\mathrm{Sin}(\theta_2 + \theta_3)\mathrm{Sin}(\theta_5) - \mathrm{Cos}(\theta_2 + \theta_3)\mathrm{Cos}(\theta_4)\mathrm{Cos}(\theta_5)\big) - \mathrm{Cos}(\theta_2 + \theta_3)\mathrm{Cos}(\theta_6)\mathrm{Sin}(\theta_4) \quad 3.27$$

$$\boldsymbol{r_{33}} = \mathrm{Sin}(\theta_2 + \theta_3)\mathrm{Cos}(\theta_5) + \mathrm{Cos}(\theta_2 + \theta_3)\mathrm{Cos}(\theta_4)\mathrm{Sin}(\theta_5) \quad 3.28$$

$\boldsymbol{P_z} = \mathrm{d_1} + \mathrm{a_2}\,\mathrm{Sin}(\theta_2) + \mathrm{d_4}\mathrm{Cos}(\theta_2)\mathrm{Sin}(\theta_3) + \mathrm{d_4}\mathrm{Cos}(\theta_3)\mathrm{Sin}(\theta_2) +$
$\mathrm{d_6}\mathrm{Cos}(\theta_2)\mathrm{Cos}(\theta_5)\mathrm{Sin}(\theta_3) + \mathrm{d_6}\mathrm{Cos}(\theta_3)\mathrm{Cos}(\theta_5)\mathrm{Sin}(\theta_2) +$
$\mathrm{d_6}\mathrm{Cos}(\theta_2)\mathrm{Cos}(\theta_3)\mathrm{Cos}(\theta_4)\mathrm{Sin}(\theta_5) - \mathrm{d_6}\mathrm{Cos}(\theta_4)\mathrm{Sin}(\theta_2)\mathrm{Sin}(\theta_3)\mathrm{Sin}(\theta_5)$ 3.29

Since all the required elements are present now, forward kinematic analysis can easily be applied on the HTM and its relevant components to obtain position and orientation values. The homogeneous transformation matrix above is specified as ${}^{0}T_6$, which means that it represents all the motion related information of the end effector relative to the base frame of the robot. So, the matrix element P_z represents the distance of the end effector in the Z axis relative to the base frame. Hence if all the elements on the right-side of Equation 3.29 are known, then the value of P_z can easily be calculated. This also applies to other elements in Equation 3.17, such as P_x, P_y, r_{11}, r_{23}, and so forth. This gives us a firm solution to find desired positions and orientations, however in real world, that is not the case almost every time. Therefore, this solution alone is not enough to make a manipulator function properly. To sum up, for the designed robot, in the forward kinematic analysis method the outputs are $r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23}, r_{31}, r_{32}, r_{33}, P_x, P_y, P_z$ while inputs are $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, a_2, d_1, d_4, d_6$. MATLAB codes generated to conduct forward kinematic analysis are presented in Appendix-3.

In industry, most of the time the exact desired position of the robot's end effector and its orientation at that point is known, but no information available about the angles of the motors to satisfy the conditions. This is where another method called inverse kinematic analysis [45] comes in handy.

3.2.3. Inverse Kinematics

Inverse kinematics in robotics can be defined as finding the variable values required for a particular terminal position and orientation which is the exact opposite of forward kinematic analysis. In the inverse kinematic analysis method, the parametric solution for the related axis variables is obtained by using nonlinear equations derived from the total homogeneous transformation matrix of the system that is previously obtained using the forward kinematic analysis. The solution of the nonlinear equations obtained from the total transformation matrix, which is formed by multiplying the homogeneous transformation matrices of the 6-degree-of-freedom robotic arm, is quite complex, and sometimes unsolvable. Solving these equations directly in closed form is often very difficult, so they require efficient and systematic methods [46]. There are two main methods to solve such equations.

I. Closed Form Solutions: The closed form approach provides the general solution of common variables depending on the equations obtained from the homogeneous transformation matrix. Thus, it provides a practical and efficient approach for real-time robotic applications where fast calculations are required to provide quick responses to robot's environment. In addition, in cases where there is more than one solution, like the elbow joint being curved towards the ground or the sky for instance, it eliminates the complexity of the solution by choosing only one of these solutions that will satisfy the requirements of the application. The closed form approach is examined under two categories.

- Geometric approach: This method uses the geometric shape caused by the manipulator's posture. It reduces the complexity in the problem by converting the problem into a series of simpler plane geometry problems. Therefore, it is commonly preferred to solve inverse kinematic problems.
- Algebraic approach: This method uses algebraic relations between the parameters of the manipulator and the axis variables. It is also a common approach to solve inverse kinematic problems.

II. Numerical (Iterative) Solutions: This approach provides a more general, robot-independent solution to the inverse kinematics problem, and it's much easier to apply, but it comes with some drawbacks such as; high computation requirement resulting in slower outputs, providing only point solutions, requires an initial guess for positions, etc.

In this method, inputs will be the positions in the axes X, Y, and Z, and the orientation values of the end effector relative to the robot's base frame while the output is the motion variables, which are the angles $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$ for the designed robot. The desired position shall be set as P_x, P_y, P_z while the desired orientation is set as $n_x, n_y, n_z, s_x, s_y, s_z, a_x, a_y, a_z$ inside the homogeneous transformation matrix ${}^0T_6$ shown in Equation 3.30

$$ {}^0T_6 = \begin{bmatrix} n_x & s_x & a_x & P_x \\ n_y & s_y & a_y & P_y \\ n_z & s_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 3.30 $$

In which;

n is short for normal vector,

s is short for sliding/side vector,

a is short for approach vector,

Px, Py, Pz represent the desired position of the tool center point (TCP),

n_x is the angle between X_5 and X_0,

n_y is the angle between X_5 and Y_0,

n_z is the angle between X_5 and Z_0,

s_x is the angle between Y_5 and X_0,

s_y is the angle between Y_5 and Y_0,

s_z is the angle between Y_5 and Z_0,

a_x is the angle between Z_5 and X_0,

a_y is the angle between Z_5 and Y_0,

a_z is the angle between Z_5 and Z_0 [47].

A visual representation of these values is provided below in Figure 3.2 and Figure 3.3.

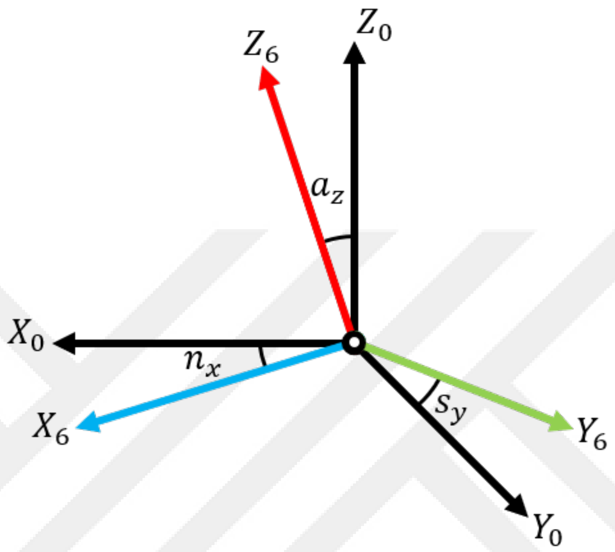


Figure 3.2. Visual Representation of Orientation Values n_x, s_y, a_z

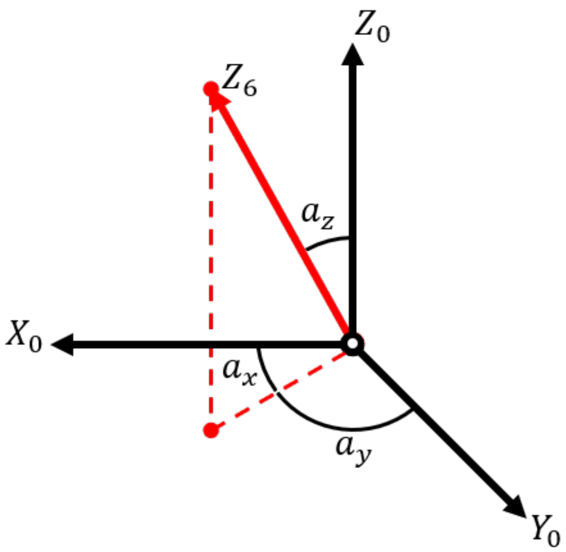


Figure 3.3. Visual Representation of Orientation Values a_x, a_y, a_z

Two overall homogeneous transformation matrices were mentioned so far. They look like each other, but they can be distinguished as the desired transformation matrix and the actual transformation matrix.

$$\text{Actual HTM: } {}^{0}\mathrm{T}_6 = \mathrm{A}_1\mathrm{A}_2\mathrm{A}_3\mathrm{A}_4\mathrm{A}_5\mathrm{A}_6 = \begin{bmatrix} \mathrm{r}_{11} & \mathrm{r}_{12} & \mathrm{r}_{13} & \mathrm{P_x} \\ \mathrm{r}_{21} & \mathrm{r}_{22} & \mathrm{r}_{23} & \mathrm{P_y} \\ \mathrm{r}_{31} & \mathrm{r}_{32} & \mathrm{r}_{33} & \mathrm{P_z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 3.17$$

$$\text{Desired HTM: } {}^{0}\mathrm{T}_6 = \begin{bmatrix} \mathrm{n_x} & \mathrm{s_x} & \mathrm{a_x} & \mathrm{P_x} \\ \mathrm{n_y} & \mathrm{s_y} & \mathrm{a_y} & \mathrm{P_y} \\ \mathrm{n_z} & \mathrm{s_z} & \mathrm{a_z} & \mathrm{P_z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 3.30$$

To be able to solve the inverse kinematics problem, these two matrices will be matched and the overlapping cells will be treated as equal. The values in the rotational part of the desired HTM will simply follow the trigonometric rules and take values ranging from -1 to $+1$. In the rotation matrix, sum of squares of the elements in each row and in each column always must be $+1$. The variables P_x, P_y, P_z in the position vector can take a great variety of values unless they don’t exceed the physical limits of the robot’s capabilities [47].

$${}^{0}T_6 = A_1A_2A_3A_4A_5A_6 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & s_x & a_x & P_x \\ n_y & s_y & a_y & P_y \\ n_z & s_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 3.31$$

In the homogeneous transformation matrix above, there are 12 equations and 6 unknowns as only 3 of the 9 rotation terms are independent. Therefore, there are 6 non-linear equations present with 6 unknowns.

A numeric example of the desired homogeneous transformation matrix derived from the kinematic model look as shown in Equation 3.32.

$${}^{0}T_6 = \begin{bmatrix} n_x & s_x & a_x & P_x \\ n_y & s_y & a_y & P_y \\ n_z & s_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -0.4619 & 0.3827 & 0.8001 & 560.0722 \\ -0.1913 & -0.9239 & 0.3314 & 231.9895 \\ 0.8660 & 0 & 0.5000 & 649.0000 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 3.32$$

As mentioned earlier, the values in the actual HTM which is on the left side in Equation 3.31 are quite long and complicated since 6 transformation matrices are multiplied to properly represent the whole system. Therefore, as an example, when the two matrices are overlapped by setting the desired values as shown in Equation 3.32, P_x and a_x values will be expressed as shown below:

$$\boldsymbol{P_x} = d_4 Cos(\theta_2 + \theta_3)Cos(\theta_1) + a_2 Cos(\theta_1)Cos(\theta_2) + d_6 Cos(\theta_2 + \theta_3)Cos(\theta_1)Cos(\theta_5) + d_6 Sin(\theta_1)Sin(\theta_4)Sin(\theta_5) - d_6 Cos(\theta_1)Cos(\theta_2)Cos(\theta_4)Sin(\theta_3)Sin(\theta_5) - d_6 Cos(\theta_1)Cos(\theta_3)Cos(\theta_4)Sin(\theta_2)Sin(\theta_5) \quad 3.33$$

$$\boldsymbol{a_x} = Cos(\theta_2 + \theta_3)Cos(\theta_1)Cos(\theta_5) - Sin(\theta_5)\big(Cos(\theta_1)Cos(\theta_2)Cos(\theta_4)Sin(\theta_3) - Sin(\theta_1)Sin(\theta_4) + Cos(\theta_1)Cos(\theta_3)Cos(\theta_4)Sin(\theta_2)\big) \quad 3.34$$

As seen in equations 3.33 and 3.34, this problem is quite complex since the parameters are heavily dependent on each other. To solve this problem the standard process offered by Richard Paul in 1981, also known as Paul's Solution [41], works as follows:

- Try to overlap the terms directly or in the division form within the desired and actual transformation matrices.
- If possible, find a simple Equation such as: $\frac{d_1 a_z}{n_x - s_x} = \frac{Sin(\theta_2)}{Cos(\theta_2)}$ thus $\theta_2 = \tan^{-1}\left(\frac{d_1 a_z}{n_x - s\_x}\right)$ using the Equation 3.35.
- When no solution is available for any angle values by overlapping the two matrices, the next step will be pre-multiplying the both sides of the equation by the inverse of the first element on the right side as shown in Equation 3.36 and try overlapping the new two matrices again. Instead of pre-multiplication, post-multiplication of both sides by the inverse of the last term on the right side of the equation is also possible.
- If all the required angle values cannot be obtained using the new matrices only, then the process shall be continued by pre-multiplying the left side of the new equation by the inverse of the first element on the right side of the new equation as shown in Equation 3.37 and the newest two matrices shall be overlapped to find the other angle values.
- This process shall be repeated till all the required angle values are obtained parametrically.

Since there are 6 joints available within the system, the complexity has been increased dramatically, so it will not be easy to apply this standard method directly, but this method is still useful to the solution and it will be used in the first step and then combined with another method.

$${}^0T_6 = A_1A_2A_3A_4A_5A_6 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & s_x & a_x & P_x \\ n_y & s_y & a_y & P_y \\ n_z & s_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 3.31$$

$${}^0T_6 = \begin{bmatrix} n_x & s_x & a_x & P_x \\ n_y & s_y & a_y & P_y \\ n_z & s_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_1A_2A_3A_4A_5A_6 \quad 3.35$$

In Equation 3.35, no useful relation is available so both sides by A_1 shall be pre-multiplied.

$$(A_1)^{-1} \times {}^0T_6 = (A_1)^{-1} \times A_1A_2A_3A_4A_5A_6 = A_2A_3A_4A_5A_6 \qquad 3.36$$

Equation 3.36 also doesn't provide the required relations. Hence, the pre-multiplication of both sides of the equation by the first element on the right side continues.

$$(A_1A_2)^{-1} \times {}^0T_6 = (A_1A_2)^{-1} \times A_1A_2A_3A_4A_5A_6 = A_3A_4A_5A_6 \qquad 3.37$$

Equation 3.37 provide a relation that can be used. But, simplify the equation first.

$$(A_1A_2)^{-1} \times {}^0T_6 = A_3A_4A_5A_6 \qquad 3.38$$

New names for the elements in these two matrices as shown in equations 3.39 and 3.40 may be assigned. In these equations all the overlapping terms will be equal such as $a_{11} = b_{11}$, $P_y = P_y$ and so forth.

$$(A_1A_2)^{-1} \times {}^0T_6 = \begin{bmatrix} a_{11} & a_{12} & a_{13} & P_x \\ a_{21} & a_{22} & a_{23} & P_y \\ a_{31} & a_{32} & a_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad 3.39$$

$$A_3A_4A_5A_6 = \begin{bmatrix} b_{11} & b_{12} & a_{13} & P_x \\ b_{21} & a_{22} & a_{23} & P_y \\ b_{31} & a_{32} & a_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad 3.40$$

Using the equations above, this relation may be obtained:

$$\frac{P_z}{a_{33}} = \frac{P_z}{b_{33}} \qquad 3.41$$

Which is:

$$\frac{P_yCos(\theta_1) - P_xSin(\theta_1)}{(a_yCos(\theta_1) - a_xSin(\theta_1))} = \frac{d_6Sin(\theta_4)Sin(\theta_5)}{Sin(\theta_4)Sin(\theta_5)} \qquad 3.42$$

Simplify this equation till parametric solution to angle θ_1 is obtained:

$$\frac{P_yCos(\theta_1) - P_xSin(\theta_1)}{(a_yCos(\theta_1) - a_xSin(\theta_1))} = d_6 \qquad 3.43$$

$$P_yCos(\theta_1) - P_xSin(\theta_1) = d_6(a_yCos(\theta_1) - a_xSin(\theta_1)) \qquad 3.44$$

Divide both sides by $Cos(\theta_1)$:

$$\mathrm{P_y} - \frac{\mathrm{P_x Sin(\theta_1\,)}}{\mathrm{Cos(\theta_1)}} = \mathrm{d_6 a_y} - \frac{\mathrm{d_6 a_x Sin(\theta_1)}}{\mathrm{Cos(\theta_1)}} \qquad 3.45$$

Group the similar terms:

$$\frac{\mathrm{d_6 a_x Sin(\theta_1)}}{\mathrm{Cos(\theta_1)}} - \frac{\mathrm{P_x Sin(\theta_1\,)}}{\mathrm{Cos(\theta_1)}} = \mathrm{d_6 a_y - P_y} \qquad 3.46$$

$$\frac{\mathrm{Sin(\theta_1)}}{\mathrm{Cos(\theta_1)}}(\mathrm{d_6 a_x - P_x}) = \mathrm{d_6 a_y - P_y} \qquad 3.47$$

$$\frac{\mathrm{Sin(\theta_1)}}{\mathrm{Cos(\theta_1)}} = \frac{\mathrm{d_6 a_y - P_y}}{\mathrm{d_6 a_x - P_x}} \qquad 3.48$$

$$\theta_1 = \tan^{-1}\left(\frac{\mathrm{d_6 a_y - P_y}}{\mathrm{d_6 a_x - P_x}}\right) \qquad 3.49$$

In Equation 3.49, there is a parametric solution available for the angle θ_1 that includes the terms d_6, a_y, a_x, P_y, P_x all of which are known. So, when the desired HTM is defined in the proper form, the solution to angle θ_1 will be present. But it’s very unlikely to find all angle values as simple as shown above for a 6-DoF robotic arm.

Now it’s time to try to find the other 5 angle values parametrically. When the same homogeneous transformation matrices are analyzed without further pre-multiplying, the relationship below is obtained:

$$\frac{P_y}{P_x} = \frac{P_y}{P_x} \qquad 3.50$$

$$\frac{(\mathrm{pz\ Cos(\theta_2) - d_1 Cos(\theta_2) - px\ Cos(\theta_1)Sin(\theta_2) - py\ Sin(\theta_1)Sin(\theta_2)})}{\mathrm{pz\ Sin(\theta_2) - d_1 Sin(\theta_2) - a2 + px\ Cos(\theta_1)Cos(\theta_2) + py\ Cos(\theta_2)Sin(\theta_1)}} =$$

$$\frac{\mathrm{d4\ Sin(\theta_3) + d_6 Cos(\theta_5)Sin(\theta_3) + d_6 Cos(\theta_3)Cos(\theta_4)Sin(\theta_5)}}{\mathrm{d4\ Cos(\theta_3) + d_6 Cos(\theta_3)Cos(\theta_5) - d_6 Cos(\theta_4)Sin(\theta_3)Sin(\theta_5)}} \qquad 3.51$$

On the left side of the equation above, only the term θ_2 is unknown since the parametric solution to angle θ_1 is already found. So, using the left side of this equation, a parametric expression for θ_2 may be created, but it will not provide the final solution since there are 3 unknowns on the right side of the equation which are $\theta_3, \theta_4, \theta_5$. Thus, with the help of trigonometric identities, it is needed to degrade the equation to a solvable level and this process is extremely time-consuming and requires a lot of focus. After many steps, it is technically possible to find a solution where θ_2 is represented in the form of a $\cos^{-1}(A)$ expression as follows:

$$k_1\big(Cos(\theta_2)\big)^2 + k_2 Cos(\theta_2) + k_5 = 0 \qquad 3.52$$

$$(Cos(\theta_2))^2 k_1 + Cos(\theta_2) k_2 + k_5 = 0 \qquad 3.53$$

$$Cos(\theta_2) = \frac{-k_1+\sqrt{k_2^2-4k_1k_3}}{2k_1} \textbf{ or } Cos(\theta_2) = \frac{-k_1-\sqrt{k_2^2-4k_1k_3}}{2k_1} \quad 3.54$$

$$\theta_2 = \cos^{-1}\left(\frac{-k_1+\sqrt{k_2^2-4k_1k_3}}{2k_1}\right) \textbf{ or } \theta_2 = \cos^{-1}\left(\frac{-k_1-\sqrt{k_2^2-4k_1k_3}}{2k_1}\right) \quad 3.55$$

Where the terms k_1, k_2, k_3 are all known. But this method is very demanding and there are still 4 other angle values to be found. Therefore, another method called Pieper's Solution [48] shall be used to solve this problem. This method is only applicable for 6-DoF robotic arms where the last 3 joints intersect at a single point and act as a spherical wrist and the designed robot satisfies this criterion. Thus, Pieper's Solution is convenient to solve the concerned inverse kinematics problem for the system developed within the scope of this study. In this method, a geometrical approach shall be used to solve the problem. Before diving into the geometrical solution, it is good to go through the axis assignment that was explained earlier and presented in Figure 3.1.

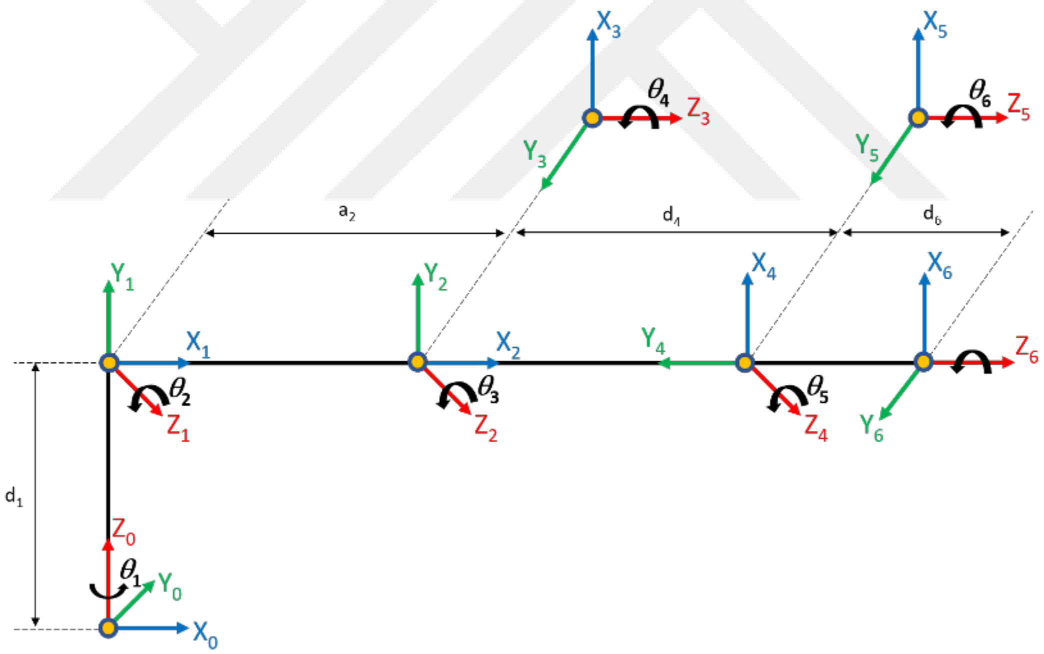


Figure 3.1. Axis Assignment of the Robotic Manipulator

In the figure above, there are some crucial coordinate centers that will be referred as; base, shoulder, elbow, wrist, tool center point (TCP). A simplified model that covers these references is provided below in Figure 3.4:

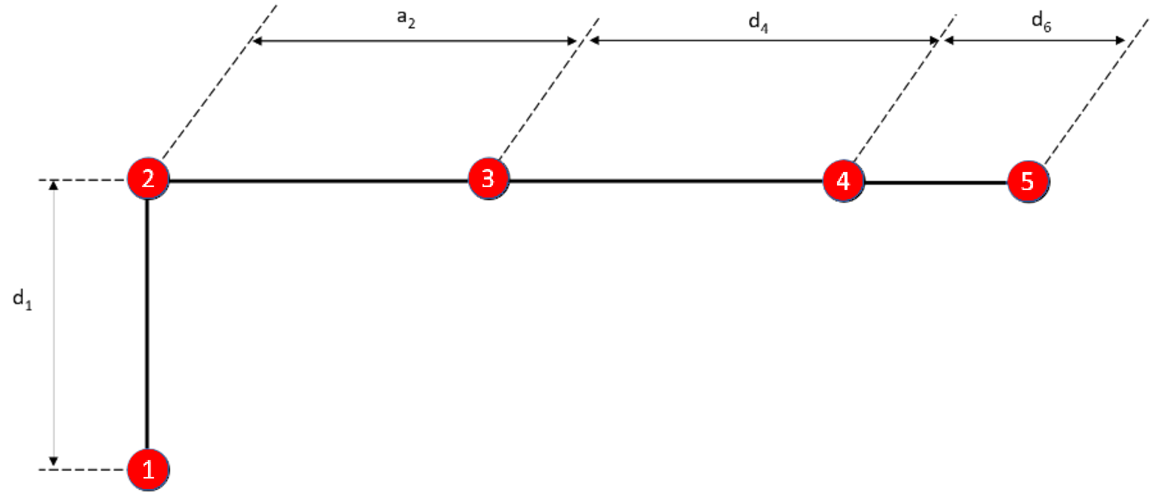


Figure 3.4. Definition of Axis Assignments of the Robotic Manipulator

In which;

Point 1 represents the center of the base frame,

Point 2 represents the center of the shoulder frame,

Point 3 represents the center of the elbow frame,

Point 4 represents the center of the wrist frame,

Point 5 represents the center of the TCP frame.

The last 3 joints intersect at point 4 in Figure 3.4 and create a spherical joint at the end of the robotic manipulator that is mainly responsible for the orientation of the TCP relative to the base frame. This will feature come very handy when solving the inverse kinematics problem.

Overall strategy to obtain an inverse kinematic solution for the whole system will be as follows:

- Divide the robot into two parts at point 4, which is the wrist center.
- Solve the inverse kinematics for the first 3 joint using the geometrical approach. This will give us a profound solution to calculate the required angles to satisfy the desired position of the wrist center.
- Then the second half of the robot will be reattached at point 4.
- Compute HTM for the first 3 joints. And using the relations in the equations below, determine the HTM from the wrist to the TCP.
- Solve the inverse kinematics for the last 3 joints.

$$A_1A_2A_3A_4A_5A_6 = {}^0T_6 \rightarrow {}^0T_6 = {}^0T_3\,{}^3T_6 \quad 3.56$$

$$({}^0T_3)^{-1}\ {}^0T_6 = ({}^0T_3)^{-1}\ {}^0T_3\,{}^3T_6 \rightarrow {}^3T_6 = ({}^0T_3)^{-1}\ {}^0T_6 \quad 3.57$$

In the geometric approach, there are two configurations available based on the wrist's position being above or below the shoulder frame. The geometric model used to calculate the position for the first 3 joints while the wrist is above the shoulder is represented in Figure 3.5.

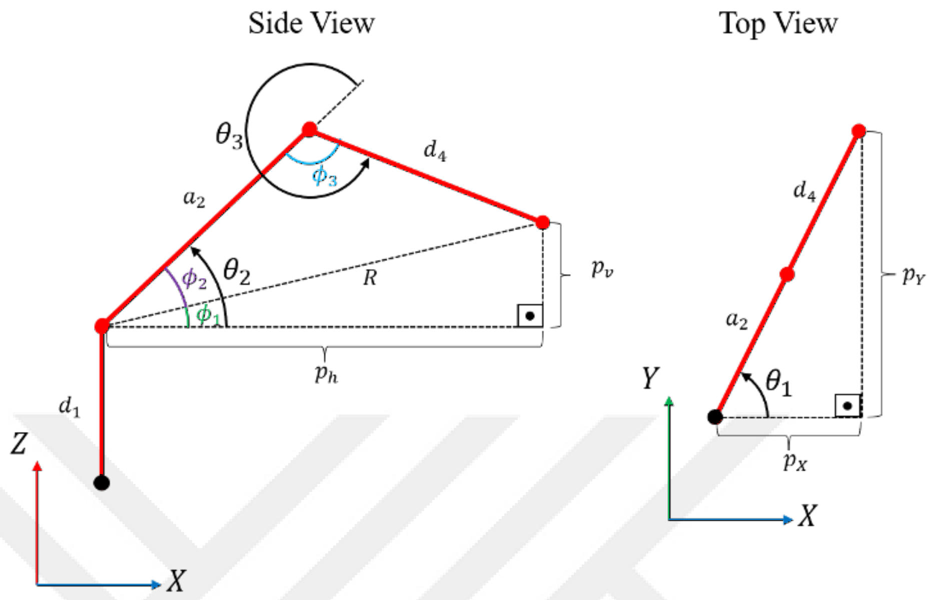


Figure 3.5. Geometric Model of the Designed Robot when Wrist is Above Shoulder

Equations derived from the geometric model for the wrist-above configuration are as follows:

$$p_v = p_z - d_1 \tag{3.58}$$

$$p_h = \sqrt{p_x^2 + p_y^2} \tag{3.59}$$

$$R = \sqrt{p_h^2 + p_v^2} \tag{3.60}$$

$$\phi_1 = \tan^{-1}\left(\frac{p_v}{p_h}\right) \tag{3.61}$$

$$\phi_2 = \cos^{-1}\left(\frac{d_4^2 - a_2^2 - R^2}{-2a_2R}\right) \tag{3.62}$$

$$\phi_3 = \cos^{-1}\left(\frac{R^2 - d_4^2 - a_2^2}{-2d_4a_2}\right) \tag{3.63}$$

$$\theta_1 = \frac{p_y}{p_x} \tag{3.64}$$

$$\theta_2 = \phi_2 - \phi_1 \tag{3.65}$$

$$\theta_3 = \pi + \phi_3 \tag{3.66}$$

The geometric model used to calculate the position for the first 3 joints while the wrist is below the shoulder is represented in Figure 3.6.

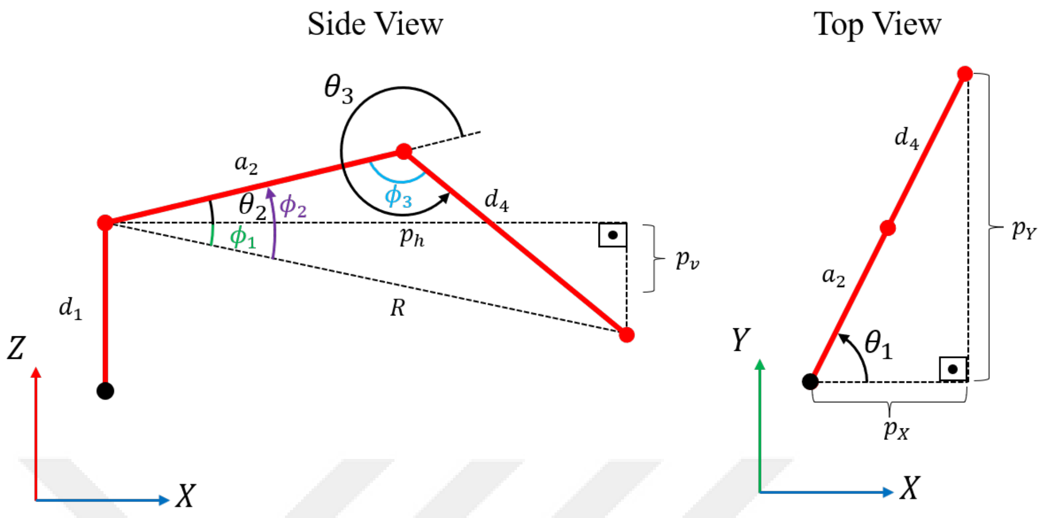


Figure 3.6. Geometric Model of the Designed Robot when Wrist is Below Shoulder

Equations derived from the geometric model for the wrist-below configuration are as follows:

$$p_v = d_1 - p_z \tag{3.67}$$

$$p_h = \sqrt{p_x^2 + p_y^2} \tag{3.68}$$

$$R = \sqrt{p_h^2 + p_v^2} \tag{3.69}$$

$$\phi_1 = \tan^{-1}\left(\frac{p_v}{p_h}\right) \tag{3.70}$$

$$\phi_2 = \cos^{-1}\left(\frac{d_4^2 - a_2^2 - R^2}{-2a_2R}\right) \tag{3.71}$$

$$\phi_3 = \cos^{-1}\left(\frac{R^2 - d_4^2 - a_2^2}{-2d_4a_2}\right) \tag{3.72}$$

$$\theta_1 = \frac{p_y}{p_x} \tag{3.73}$$

$$\theta_2 = \phi_2 - \phi_1 \tag{3.74}$$

$$\theta_3 = \pi - \phi_3 \tag{3.75}$$

Now that solutions to the first 3 angle values are available for the given input coordinate values of the wrist, the last 3 angle values can be computed so that they can be combined with the current solution to form the final solution.

Since frame 2 and frame 3 are intersected at point 3 in Figure 3.4, it may seem confusing and in some cases for similar configurations it may be needed to make slight adjustments to the standard method. In the present model, the relation ${}^{3}T_6 = ({}^{0}T_3)^{-1}\ {}^{0}T_6$ does not give enough parameters to calculate the position of the wrist properly as it doesn't include the length d_4 and the angle θ_3. But since θ_4 only effects the orientation of the wrist and not the position, and as the frame 3 can be translated along the Z_3 axis towards the wrist, it makes no difference for the solution. The last three joints still count as intersecting at the wrist center. Even if it didn't work, it would still be possible to use the pre-multiplication method as explained earlier and obtain another relation as ${}^{4}T_6 = ({}^{0}T_4)^{-1}\ {}^{0}T_6$.

Luckily, the equation ${}^{3}T_6 = ({}^{0}T_3)^{-1}\ {}^{0}T_6$ is enough to find a useful relation for θ_4 and the HTM for this configuration looks as follows:

$$(A_1A_2A_3)^{-1} \times {}^{0}T_6 \ = A_4A_5A_6 \quad 3.76$$

New names for the elements in these two matrices may be assigned as shown in equations 3.77 and 3.78. In these equations all the overlapping terms will be equal such as $c_{11} = d_{11}$, $P_z = P_z$ and so forth.

$$(A_1A_2A_3)^{-1} \times {}^{0}T_6 \ = \begin{bmatrix} c_{11} & c_{12} & c_{13} & P_x \\ c_{21} & c_{22} & c_{23} & P_y \\ c_{31} & c_{32} & c_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 3.77$$

$$A_4A_5A_6 = \begin{bmatrix} d_{11} & d_{12} & d_{13} & P_x \\ d_{21} & d_{22} & d_{23} & P_y \\ d_{31} & d_{32} & d_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 3.78$$

Using the equations above, this relation may be obtained:

$$\frac{d_{23}}{d_{13}} = \frac{c_{23}}{c_{13}} \quad 3.79$$

$$\frac{\mathrm{Sin}(\theta_4)\mathrm{Sin}(\theta_5)}{\mathrm{Cos}(\theta_4)\mathrm{Sin}(\theta_5)} = \frac{a_x\mathrm{Sin}(\theta_1) - a_y\mathrm{Cos}(\theta_1)}{c_{13}} \quad 3.80$$

$$\frac{\mathrm{Sin}(\theta_4)}{\mathrm{Cos}(\theta_4)} = \frac{a_x\mathrm{Sin}(\theta_1) - a_y\mathrm{Cos}(\theta_1)}{c_{13}} \quad 3.81$$

$$\theta_4 = \tan^{-1}\left(\frac{a_x\mathrm{Sin}(\theta_1) - a_y\mathrm{Cos}(\theta_1)}{c_{13}}\right) \quad 3.82$$

Where the term c_{13} is known and its long form is expressed as follows;

$$c_{13} = a_z Cos(\theta_2)Cos(\theta_3) - a_z Sin(\theta_2)Sin(\theta_3) - a_x Cos(\theta_1)Cos(\theta_2)Sin(\theta_3) - a_x Cos(\theta_1)Cos(\theta_3)Sin(\theta_2) - a_y Cos(\theta_2)Sin(\theta_1)Sin(\theta_3) - a_y Cos(\theta_3)Sin(\theta_1)Sin(\theta_2) \quad 3.83$$

Once the θ_4 is obtained, the matrices are no more useful. So it is needed to premultiply both sides of the Equation 3.76 one more time to find θ_5 and θ_6.

$$(A_1 A_2 A_3 A_4)^{-1} \times {}^0T_6 = A_5 A_6 \quad 3.84$$

New names for the elements in these two new matrices shall be assigned as shown in equations 3.85 and 3.86.

$$(A_1 A_2 A_3 A_4)^{-1} \times {}^0T_6 = \begin{bmatrix} e_{11} & e_{12} & e_{13} & P_x \\ e_{21} & e_{22} & e_{23} & P_y \\ e_{31} & e_{32} & e_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 3.85$$

$$A_5 A_6 = \begin{bmatrix} f_{11} & f_{12} & f_{13} & P_x \\ f_{21} & f_{22} & f_{23} & P_y \\ f_{31} & f_{32} & f_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 3.86$$

Using the equations and matrices above, the following relation is obtained:

$$\frac{e_{13}}{e_{23}} = \frac{f_{13}}{f_{23}} \quad 3.87$$

$$\frac{e_{13}}{e_{23}} = \frac{Sin(\theta_5)}{-Cos(\theta_5)} \quad 3.88$$

Thus;

$$\theta_5 = \tan^{-1}\left(\frac{e_{13}}{e_{23}}\right) \quad 3.89$$

Where;

$$e_{13} = a_x Sin(\theta_1) Sin(\theta_4) - a_y Cos(\theta_1) Sin(\theta_4) + a_z Cos(\theta_2)Cos(\theta_3)Cos(\theta_4) - a_z Cos(\theta_4) Sin(\theta_2) Sin(\theta_3) - a_x Cos(\theta_1)Cos(\theta_2)Cos(\theta_4) Sin(\theta_3) - a_x Cos(\theta_1)Cos(\theta_3)Cos(\theta_4) Sin(\theta_2) - a_y Cos(\theta_2)Cos(\theta_4) Sin(\theta_1) Sin(\theta_3) - a_y Cos(\theta_3)Cos(\theta_4) Sin(\theta_1) Sin(\theta_2) \quad 3.90$$

$$e_{23} = -\left(a_x Cos(\theta_1)\,Sin(\theta_2)\,Sin(\theta_3) - a_z Cos(\theta_3)\,Sin(\theta_2) - a_x Cos(\theta_1) Cos(\theta_2) Cos(\theta_3) - a_y Cos(\theta_2) Cos(\theta_3)\,Sin(\theta_1) - a_z Cos(\theta_2)\,Sin(\theta_3) + a_y\,Sin(\theta_1)\,Sin(\theta_2)\,Sin(\theta_3)\right) \tag{3.91}$$

Using the same matrices 3.85 and 3.86, the following relation for θ_6 may be obtained:

$$\frac{e_{31}}{e_{32}} = \frac{f_{31}}{f_{32}} \tag{3.92}$$

$$\frac{e_{31}}{e_{32}} = \frac{Sin(\theta_6)}{Cos(\theta_6)} \tag{3.93}$$

Thus;

$$\theta_6 = \tan^{-1}\left(\frac{e_{31}}{e_{32}}\right) \tag{3.94}$$

Where;

$$e_{31} = n_x Cos(\theta_4)\,Sin(\theta_1) - n_y Cos(\theta_1) Cos(\theta_4) - n_z Cos(\theta_2) Cos(\theta_3)\,Sin(\theta_4) + n_z\,Sin(\theta_2)\,Sin(\theta_3)\,Sin(\theta_4) + n_x Cos(\theta_1) Cos(\theta_2)\,Sin(\theta_3)\,Sin(\theta_4) + n_x Cos(\theta_1) Cos(\theta_3)\,Sin(\theta_2)\,Sin(\theta_4) + n_y Cos(\theta_2)\,Sin(\theta_1)\,Sin(\theta_3)\,Sin(\theta_4) + n_y Cos(\theta_3)\,Sin(\theta_1)\,Sin(\theta_2)\,Sin(\theta_4) \tag{3.95}$$

$$e_{32} = o_x Cos(\theta_4)\,Sin(\theta_1) - o_y Cos(\theta_1) Cos(\theta_4) - o_z Cos(\theta_2) Cos(\theta_3)\,Sin(\theta_4) + o_z\,Sin(\theta_2)\,Sin(\theta_3)\,Sin(\theta_4) + o_x Cos(\theta_1) Cos(\theta_2)\,Sin(\theta_3)\,Sin(\theta_4) + o_x Cos(\theta_1) Cos(\theta_3)\,Sin(\theta_2)\,Sin(\theta_4) + o_y Cos(\theta_2)\,Sin(\theta_1)\,Sin(\theta_3)\,Sin(\theta_4) + o_y Cos(\theta_3)\,Sin(\theta_1)\,Sin(\theta_2)\,Sin(\theta_4) \tag{3.96}$$

MATLAB codes prepared to conduct inverse kinematic analysis are presented in detail in Appendix-4.

3.2.4. Mechanical Design

After obtaining the full solution to the inverse kinematics problem, a real-size 3D model of the robotic manipulator was designed in the CAD software SolidWorks by Dassault Systemes. Technical drawing of the designed robot was generated in the software eDrawings. By using the software SolidWorks Visualize, the CAD model of the designed robot was rendered in high quality. Figure 3.7 is the technical drawing of the robot in which all units are in millimeters.

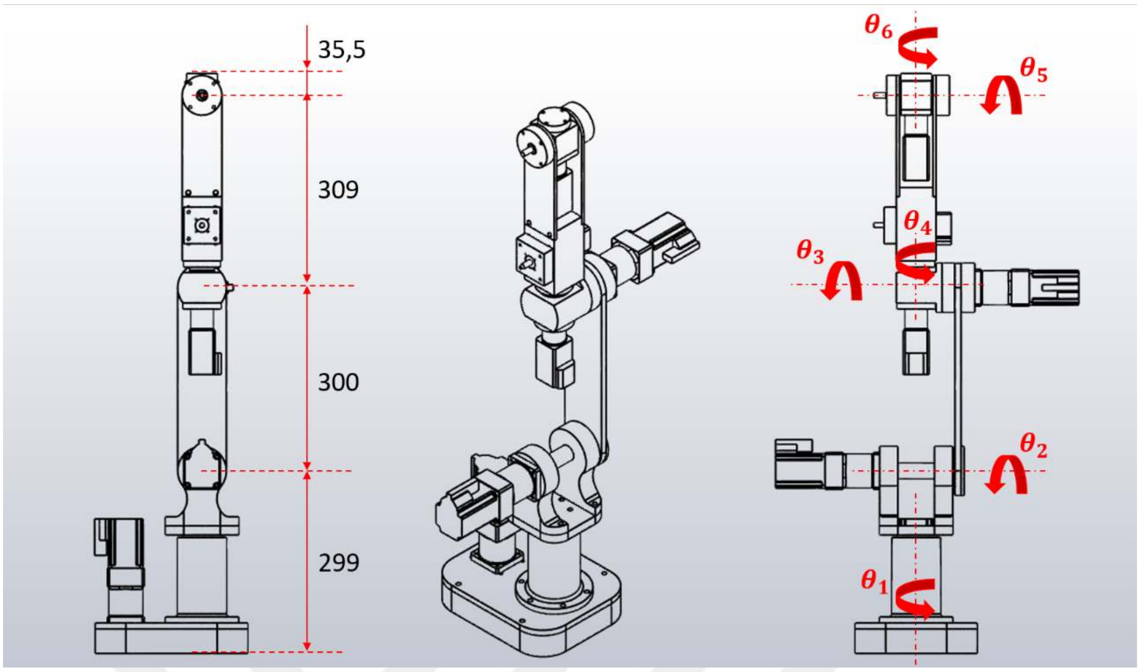


Figure 3.7. Technical Drawing of the Designed Robotic Arm

Figure 3.8 and Figure 3.9 are the views of the robotic arm's CAD model when the shoulder and the elbow bended which means that the angles $\theta_2 = +\frac{\pi}{2}$ and $\theta_3 = -\frac{\pi}{2}$ and all other angles are equal to zero.

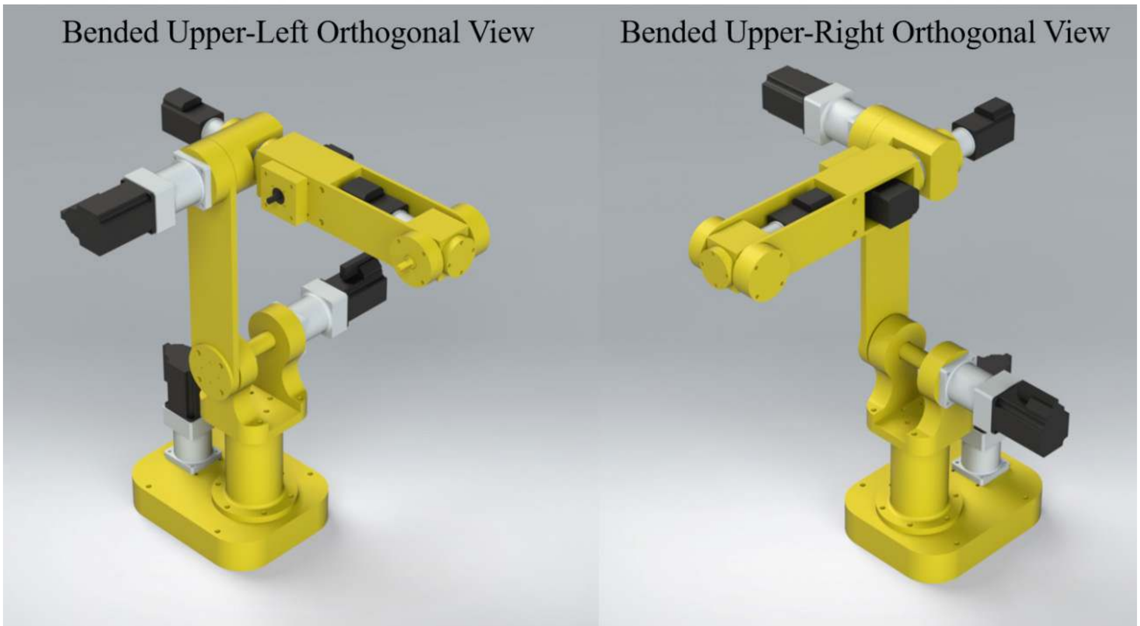


Figure 3.8. CAD Model of the Designed Robotic Arm

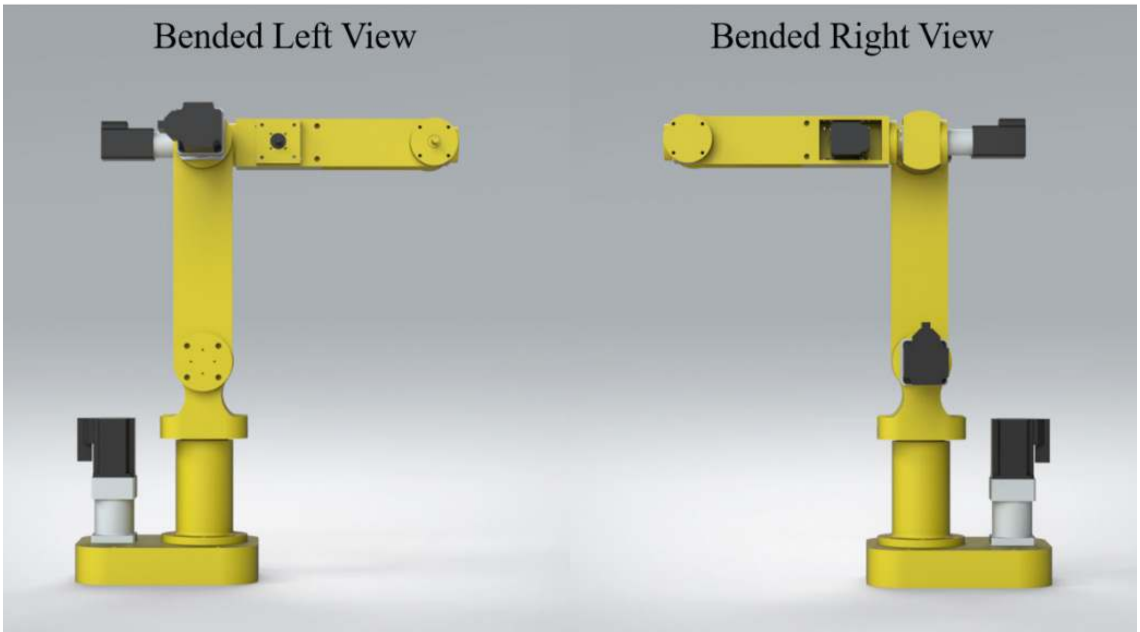


Figure 3.9. Robotic Arm Bended Side Views

Figure 3.10, Figure 3.11, and Figure 3.12 are the views of the robotic arm's CAD model when the robot is horizontally laid in the home position and all angles are equal to zero.

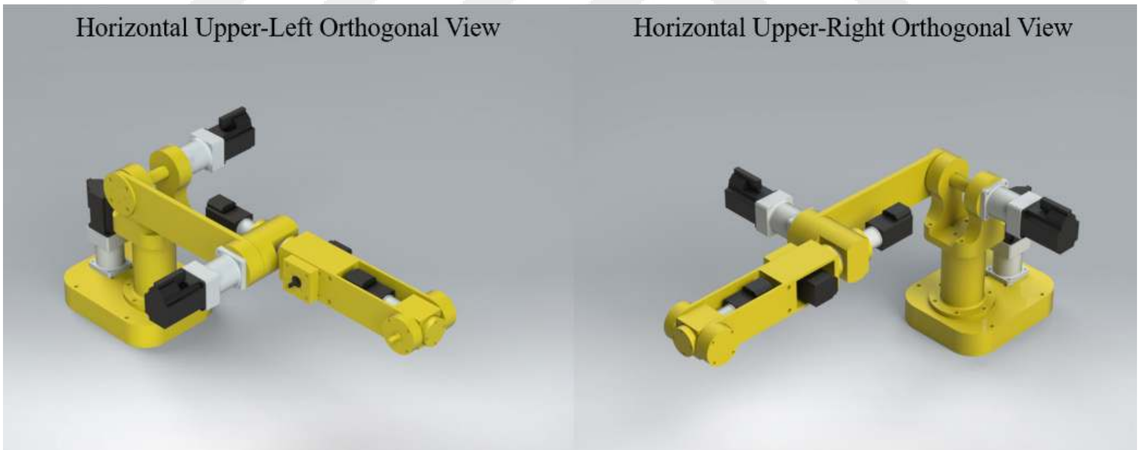


Figure 3.10. Robotic Arm Horizontal Orthogonal Views

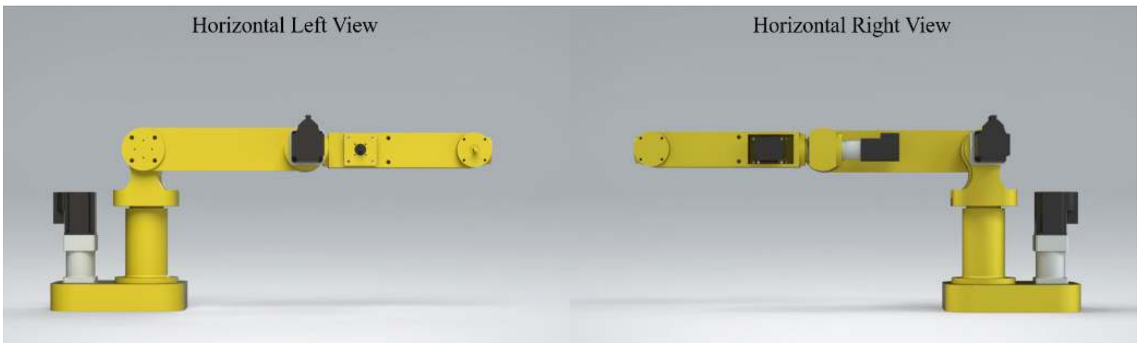


Figure 3.11. Robotic Arm Horizontal Side Views

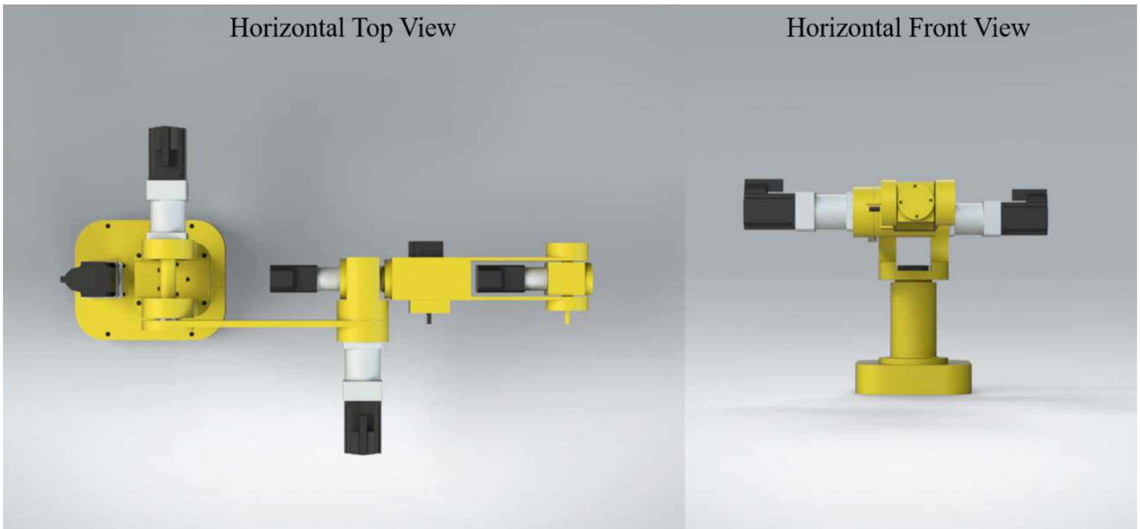


Figure 3.12. Robotic Arm Horizontal Front and Top Views

Figure 3.13, Figure 3.14, Figure 3.15 and Figure 3.16 are the views of the robotic arm's CAD model when the shoulder is rotated to keep to robot's body fully normal to the ground. In this position, the angle $\theta_2 = +\frac{\pi}{2}$ and all other angles are set to zero.

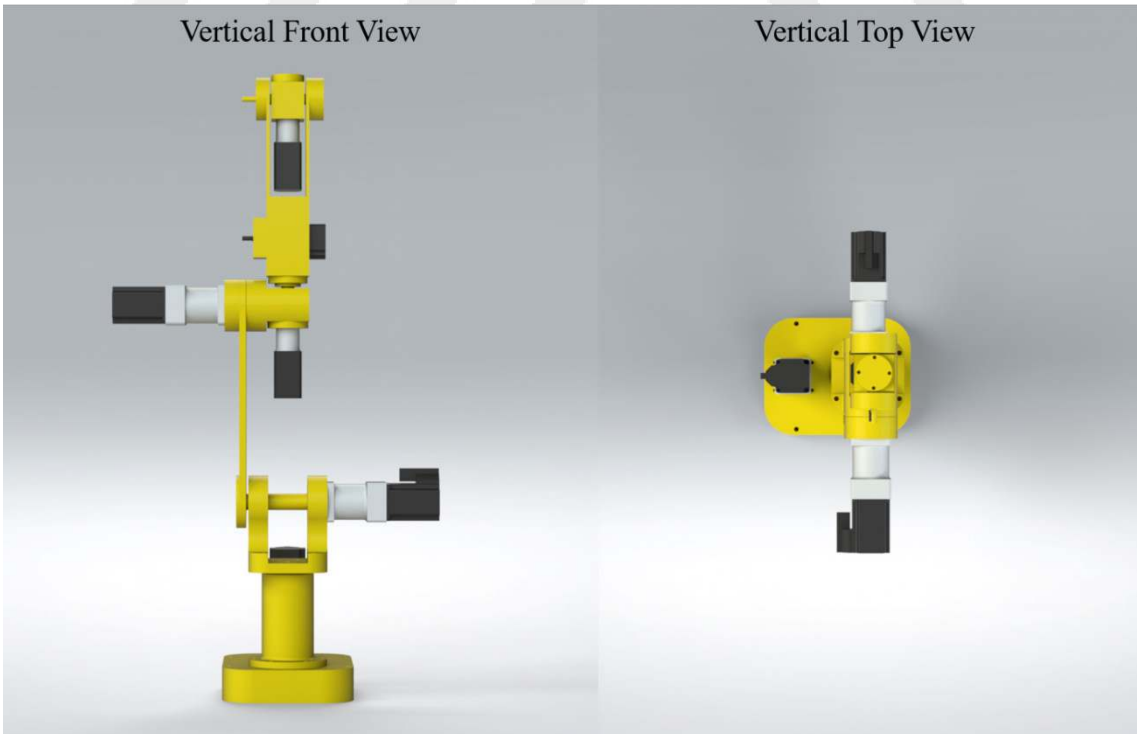


Figure 3.13. Robotic Arm Vertical Front and Top Views

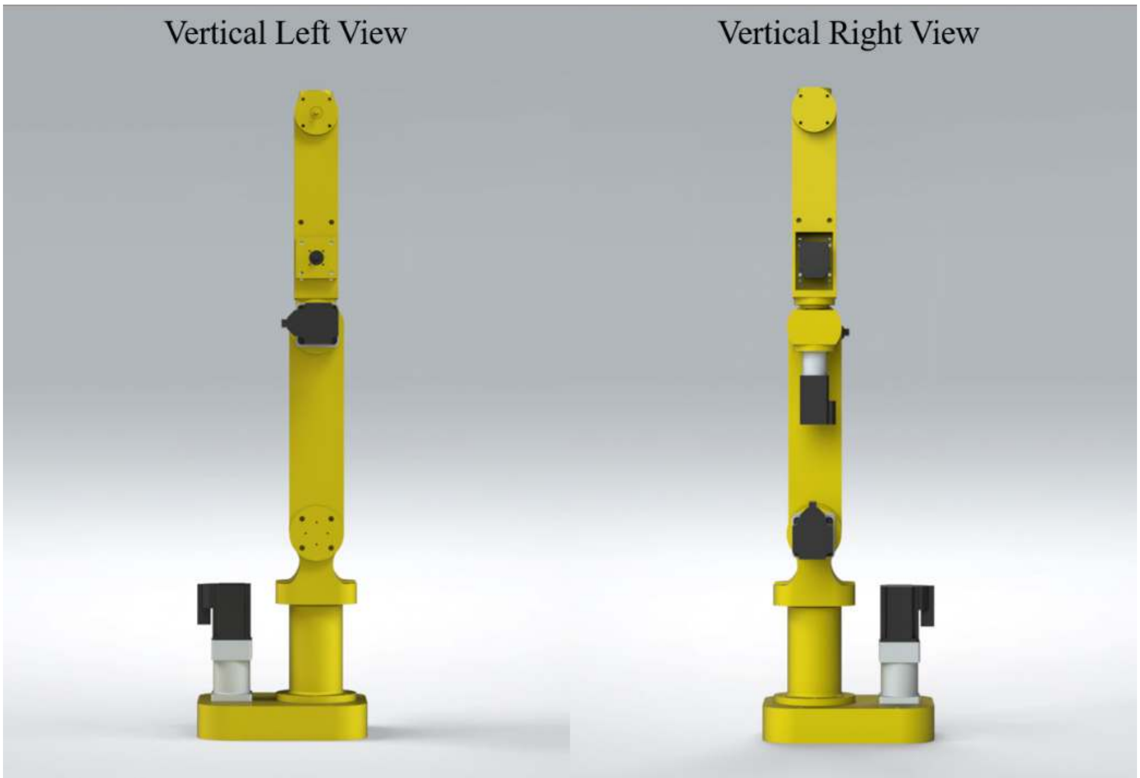


Figure 3.14. Robotic Arm Vertical Side Views

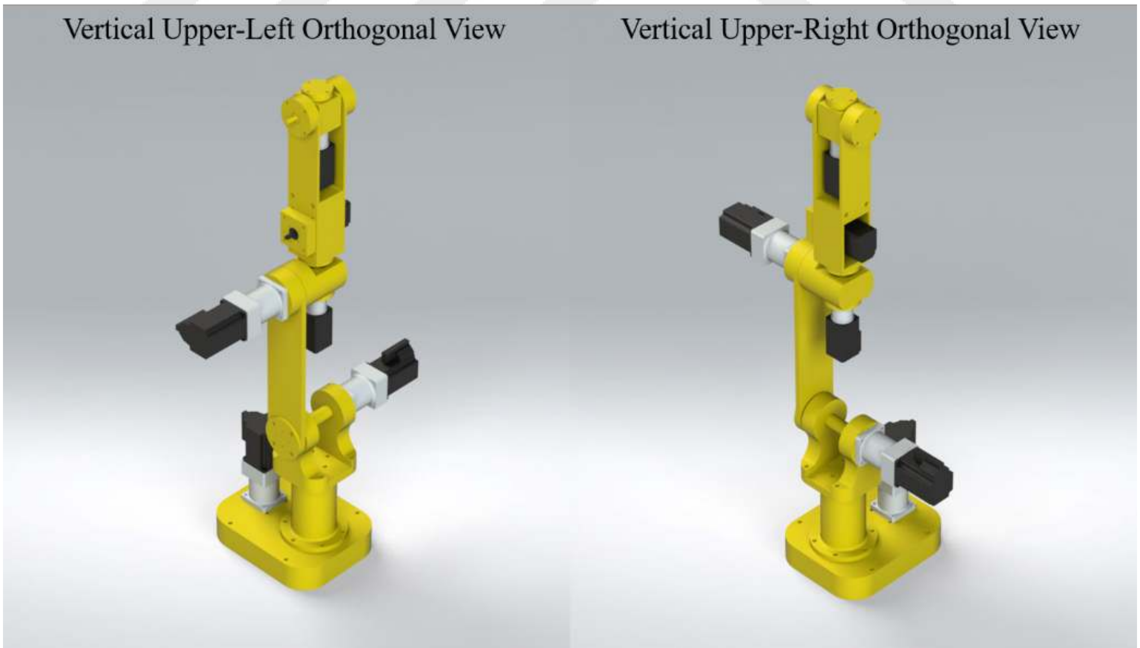


Figure 3.15. Robotic Arm Vertical Orthogonal Views

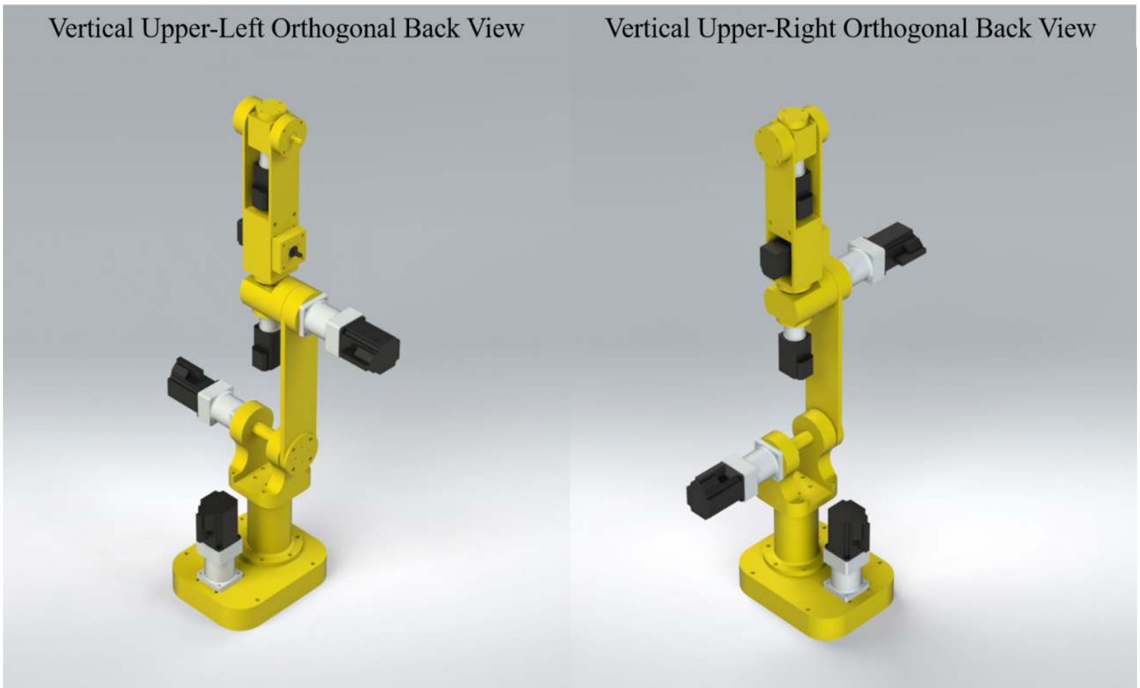


Figure 3.16. Robotic Arm Vertical Orthogonal Back Views

All the necessary design-related components such as; bearings, supports, mounting plates, grooves, pulleys, etc. were reflected to the design. Detailed technical drawings for each part are presented in Appendix-8.

3.2.5. Simulation and Control

Once the CAD model of the robotic arm was designed, a 3D simulation was prepared in the software MATLAB by MathWorks. The well-known MATLAB-based block diagram environment Simulink has been used to develop the simulation. Simscape Multibody (formerly known as SimMechanics) environment has been used to generate the 3D model within Simulink. 3D model generated within Simulink is shown below in Figure 3.17.

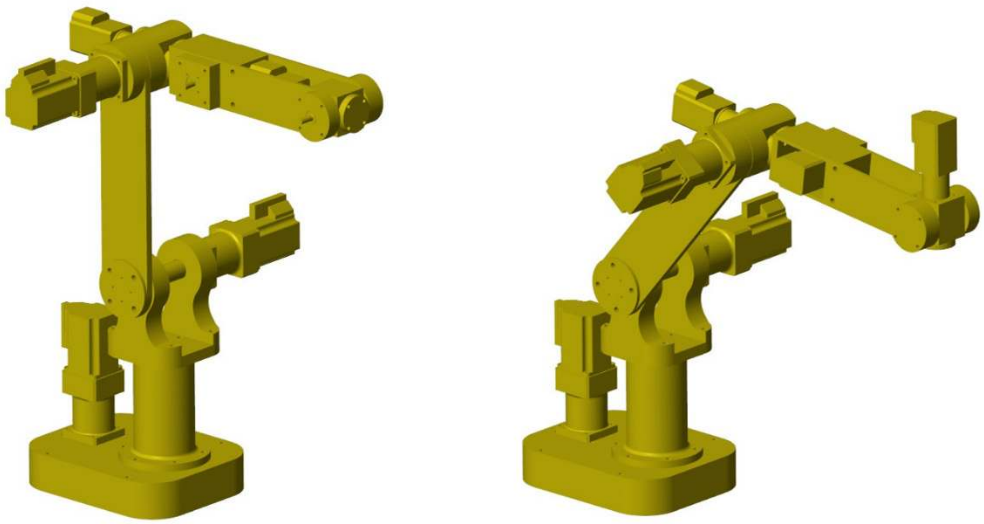


Figure 3.17. Generated 3D Model in Simulink

Overall view of the Simulink model is shown below in Figure 3.18.

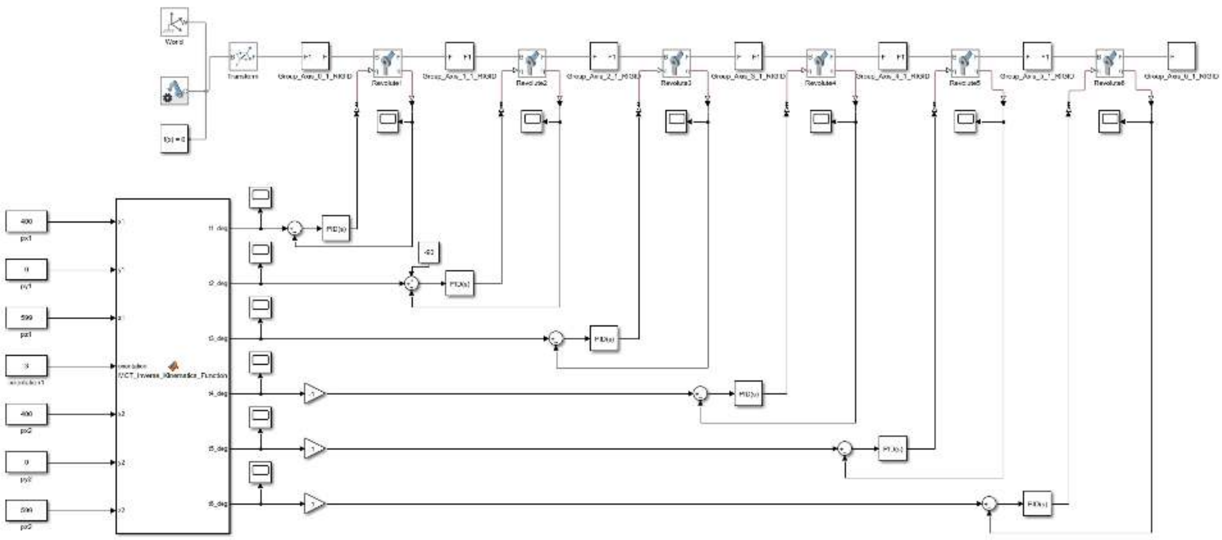


Figure 3.18. Simulink Model of the Designed Robot

In the Simulink model, in order to provide a proper and accurate movement for the virtual robot by following the calculated angle values, a PID (Proportional-Integral-Derivative) controller has been used [49]. This controller makes sure that the calculated theoretical angle values that are fed into the virtual robot through virtual actuators are always in line with the actual angle values on the robot end. It consists of 3 elements as K_p, K_i, K_d and is expressed as below.

$$u(t) = K_p e(t) + K_i \int_0^{\tau} e(t)dt + K_d \frac{d}{dt} e(t) \quad 3.97$$

In which;

K_p stands for the proportional control component, which will reduce the rise time, but will not eliminate the steady state error,

K_i represents the integral control and is effective in reducing the steady-state error, but it can have a negative effect on the transient error,

K_d is the derivative control element and is effective in increasing the stability of the system, reducing the maximum overshoot, and improving the transient reduction,

In order to pass the input data provided by the user into the Simulink model and visually crosschecking the numerical solution obtained, a user interface was developed in the MATLAB App Designer environment. Source codes of the user interface is presented in Appendix-5.

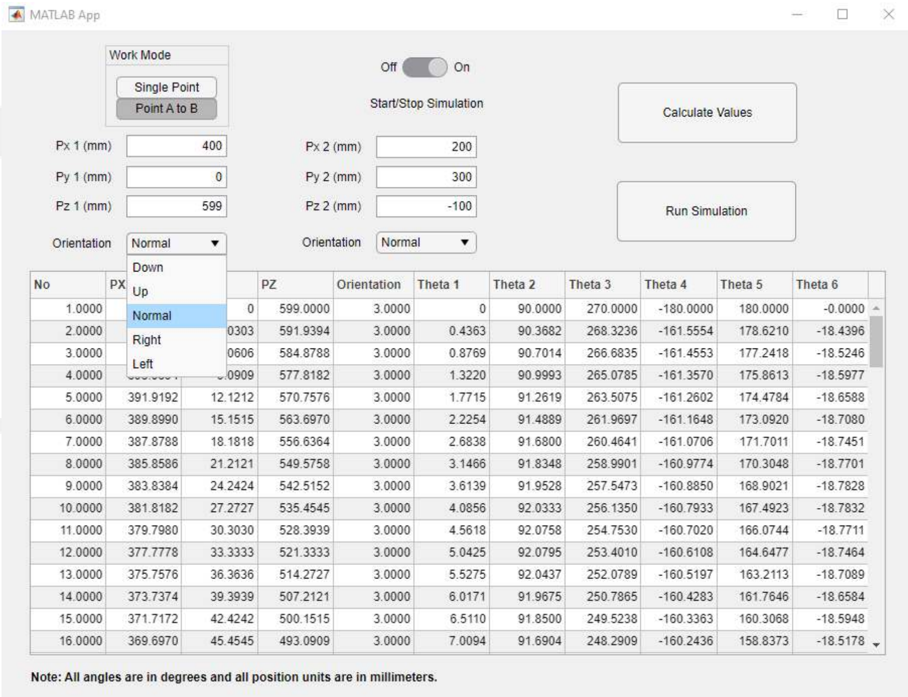


Figure 3.19. User Interface for the Inputs into Simulink Model

Through the interface above in Figure 3.19, user can provide input for the start and end points in the cartesian space along with an orientation for the TCP and can see solutions of all 6 joint values in the joint space. Coordinate inputs are provided by the textboxes on the interface and the orientation input is provided using the ComboBox objects that has options such as "Up, Down, Normal (to XY), Right (Normal to XZ), etc.". 100 different points are generated between the start and end points for all 3 axes (X, Y, Z) which are called waypoints and the solution is obtained for each waypoint along the path while keeping the orientation steady as specified

.

3.2.6. Workspace Estimation

Workspace of an articulated robot generally tends to have a sphere-like shape. The designed robot that is the focus of this study is a 6-DoF articulated robot with 6 rotational joints, therefore its workspace also is supposed to look like a sphere, but since there are physical limits in the actual prototype for each joint's motion that are caused by the design choice, the sphere looks slightly different. The angle limits for each axis have been specified in Table 3.3.

A famous and simple method called the Monte Carlo method was used to calculate the workspace of the robot [50]. In this method, random samples of the joints' angle values within the physical boundaries are generated. Then using the direct kinematic equations derived earlier, position values P_x, P_y, P_z for the TCP are calculated by feeding the random angle samples from all joints. Thus, the result of this operation creates a single point within the robot's reach. By using multiple samples, a very accurate estimation can be made for the robot's motion capabilities. In this study, 10.000 samples have been created to estimate the workspace using the formula below:

$$\theta_i = (\theta_{i\_max} - \theta_{i\_min}) * rand(\,) + \theta_{i\_min} \qquad 3.98$$

In which;

θ_i is the i_{th} joint's current angle,

θ_{i_max} is the i_{th} joint's maximum angle,

θ_{i_min} is the i_{th} joint's minimum angle,

$rand(\,)$ is an operator that generates a numerical value between 0 and 1.

Below is the estimated workspace of the designed robot generated using the method explained above:

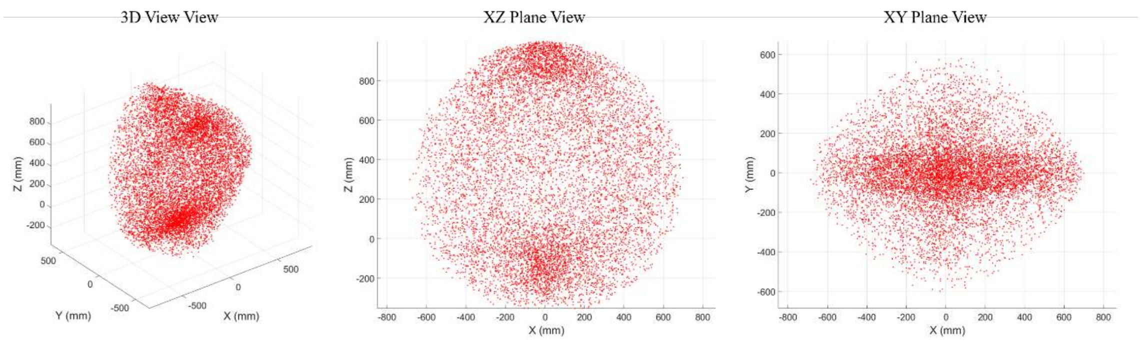


Figure 3.20. Workspace Estimation of the Designed Robot

The full MATLAB code that was used to generate the workspace estimation has been provided in Appendix-6.

3.3. Prototype Development

Once all the theoretical solutions are verified, it is time to start developing the actual robotic manipulator that would be manufactured.

3.3.1. Static Load Calculations

Free body diagram of the robot to calculate the loads on each axis is defined as follows in Figure 3.21:

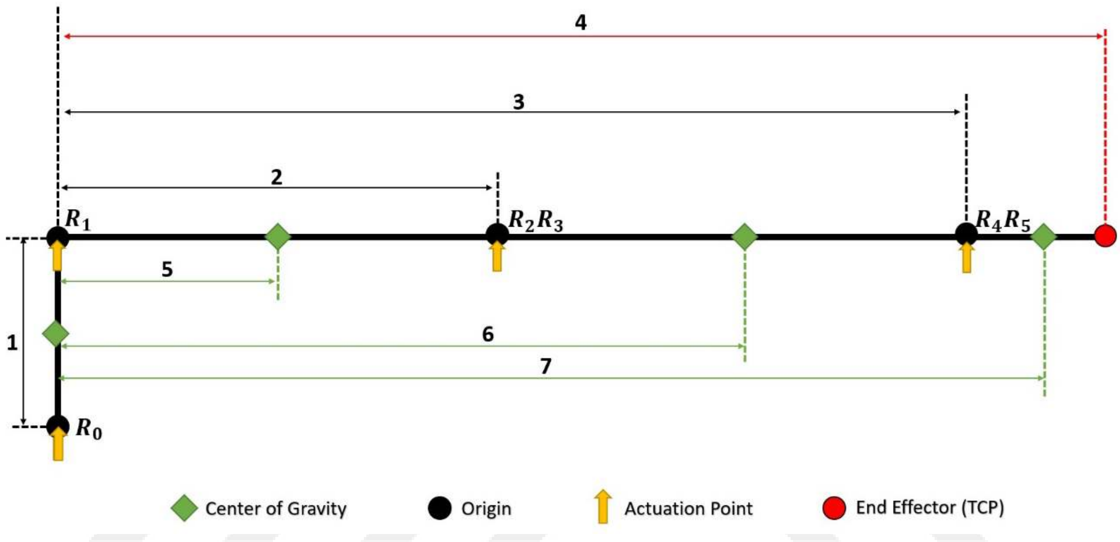


Figure 3.21. Free Body Diagram of the Robotic Arm for Static Load Calculations

In order to keep the figure clear and understandable since there are dozens of components, forces are not presented in the free body diagram. They were calculated in a separate sheet considering almost every single mass on the robot and their distances to each joint's actuation point.

Master distances to calculate actual static loads on each joint were set as shown below in Figure 3.22

		Length (mm)	Distance to R1 (mm)	Distance to R2 (mm)	Distance to R3 (mm)	Distance to R4 (mm)	Distance to R5 (mm)
1	R0 Origin		299				
	Link 1	299					
	Link 1 CoG		149,5				
	R1 Origin		0				
	Link 2	300					
5	Link 2 CoG		150				
2	R2 Origin		300				
	Link 3	0					
2	Link 3 CoG		300	0			
2	R3 Origin		300	0			
	Link 4	315					
6	Link 4 CoG		457,5	157,5	157,5		
3	R4 Origin		615	315	315		
	Link 5	0					
3	Link 5 CoG		615	315	315	0	
3	R5 Origin		615	315	315	0	
	Link 6	36					
7	Link 6 CoG		633	333	333	18	18
4	End Effector (TCP)		651	351	351	36	36

Figure 3.22. Master Distances of the Robot Based on the Free Body Diagram

Based on the calculations to find static loads on each axis, maximum torque requirements for each motor were obtained as follows in Table 3.5. The calculations consist of two parts, the static part where the robot tries to hold itself steady against gravitation when fully extended and then the dynamic part where the robot tries to lift itself upwards against gravitation at a fixed acceleration. Since the fully extended position is the worst-case scenario for the robot by means of torque requirement, once the robot can tolerate this load, it will have no problem at all when working within the rated reach at the rated velocity.

Table 3.5. Maximum Torque Requirements for each Motor

THEORETICAL MAXIMUM TORQUE REQUIREMENTS									
	Load Inertia (kg*m^2)	Angular Velocity (rad/s)	Time to accelerate (s)	Angular Acceleration (rad/s^2)	Motion Torque (Nm)	Holding Torque (Nm)	Total Torque (Nm)	Security Factor	Total Torque Requirement with Security Factor (Nm)
Motor 1	3,408	0,5	1	0,5	1,704	11,220	15,778	1,4	18,094
Motor 2	1,601	0,5	1	0,5	0,800	37,400	44,009	1,4	53,481
Motor 3	0,403	0,5	1	0,5	0,201	11,886	18,292	1,4	16,922
Motor 4	0,018	0,5	1	0,5	0,009	3,609	2,121	1,4	5,064
Motor 5	0,003	0,5	1	0,5	0,001	0,818	3,117	1,4	1,147
Motor 6	0,003	0,5	1	0,5	0,001	0,818	3,063	1,4	1,147

To see the effects of any component in the system on any axis, by dividing the overall torque calculation equations, a discrete load calculation table was obtained as shown in Table 3.6:

Table 3.6. Maximum Discrete Torque Requirements for each Motor

PARTIAL TORQUE REQUIREMENTS						
	Holding Torque R1 to R2 (Nm)	Holding Torque R2 to R3 (Nm)	Holding Torque R3 to R5 (Nm)	Load Inertia R1 to R2 (kg*m^2)	Load Inertia R2 to R3 (kg*m^2)	Load Inertia R3 to R5 (kg*m^2)
Sub-Calcs for R0	2,997	2,110	5,051	178,303	49,806	576,891
Sub-Calcs for R1	5,626	4,220	10,101	174,670	49,806	576,891
Sub-Calcs for R2	-	-0,120	5,033	-	3,262	161,234
Sub-Calcs for R3	-	-	5,055	-	-	161,772
Sub-Calcs for R4	-	-		-	-	
Sub-Calcs for R5	-	-		-	-	

Considering the load calculations, a final configuration of the 6 motors were chosen as shown below in Table 3.7.

Table 3.7. Final Motor Configuration for the Robot

SELECTED FINAL CONFIGURATION									
	Motor Type	Gearbox Type	Motor Torque (Nm)	Gearbox Ratio	Efficiency (%)	Max. Allowable Output Torque (Nm)	Total Output Torque (Nm)	Motor Inertia (kg*m^2)	Inertia Ratio
Motor 1	Nema 23 Closed Loop Geared Stepper Motor	Planetary 50:1	1,2	50	0,94	50	56,400	0,000028	24,342
Motor 2	Nema 23 Closed Loop Geared Stepper Motor	Planetary 50:1	1,2	50	0,94	50	56,400	0,000028	11,433
Motor 3	Nema 23 Closed Loop Geared Stepper Motor	Planetary 50:1	1,2	50	0,94	50	56,400	0,000028	2,877
Motor 4	Nema 17 Closed Loop Geared Stepper Motor	Planetary 14:1	0,52	13,734	0,81	5	5,785	0,0000077	6,098
Motor 5	Nema 17 Closed Loop Geared Stepper Motor	Planetary 14:1	0,52	13,734	0,81	5	5,785	0,0000077	1,028
Motor 6	Nema 17 Closed Loop Geared Stepper Motor	Planetary 14:1	0,52	13,734	0,81	5	5,785	0,0000077	1,028

3.3.2. Component Selection

Once the motors were selected then other components of the system were chosen accordingly. An overall list of all components and their prices is presented in Appendix-2.

A general view of some of the components is shown in Figure 3.23.

Figure 3.23. General View of the Electronics Components

3.3.3. Test Setups

A test setup for the motors was prepared and all the motors were tested as shown in Figure 3.24.

Figure 3.24. Motor Test Setup

Then the control board setup was prepared to enclose all the control-related electronics components as shown below in Figure 3.25:

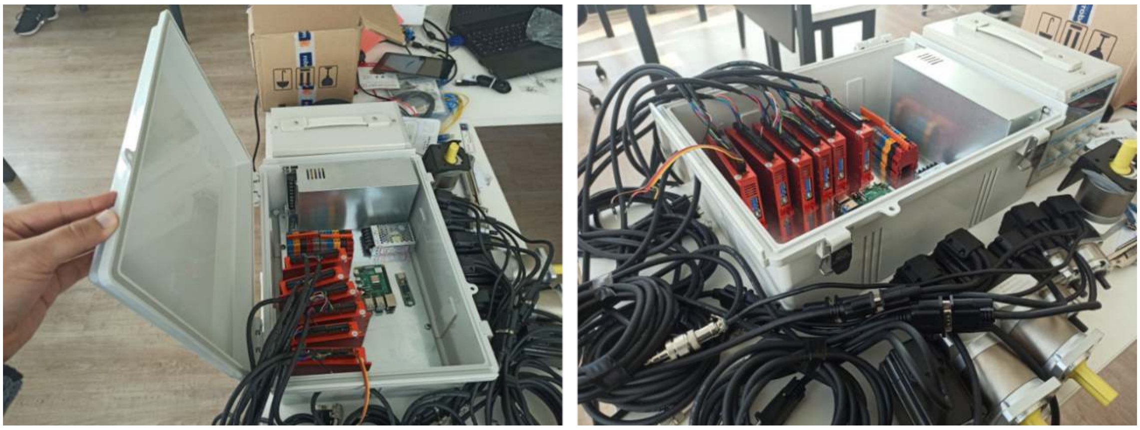


Figure 3.25. Control Board Setup

3.3.4. System Hardware Topology

The robotic arm has many electronic and mechanical components. A user provides the required inputs to the system's microcontroller via either the control panel or a computer. A power supply, provides the required power both to the microcontroller and the motor drivers which feed the actuators. The microcontroller communicates with the drive units in accordance with the user's commands. The drive units provide the required control signal for the stepper motors. The motors actuate the robotic arm's links. The encoders, limit switches and other electronic components provide smooth and accurate motion for the links when the robot is in operation. The system's overall hardware topology is presented as a diagram below in Figure 3.26.

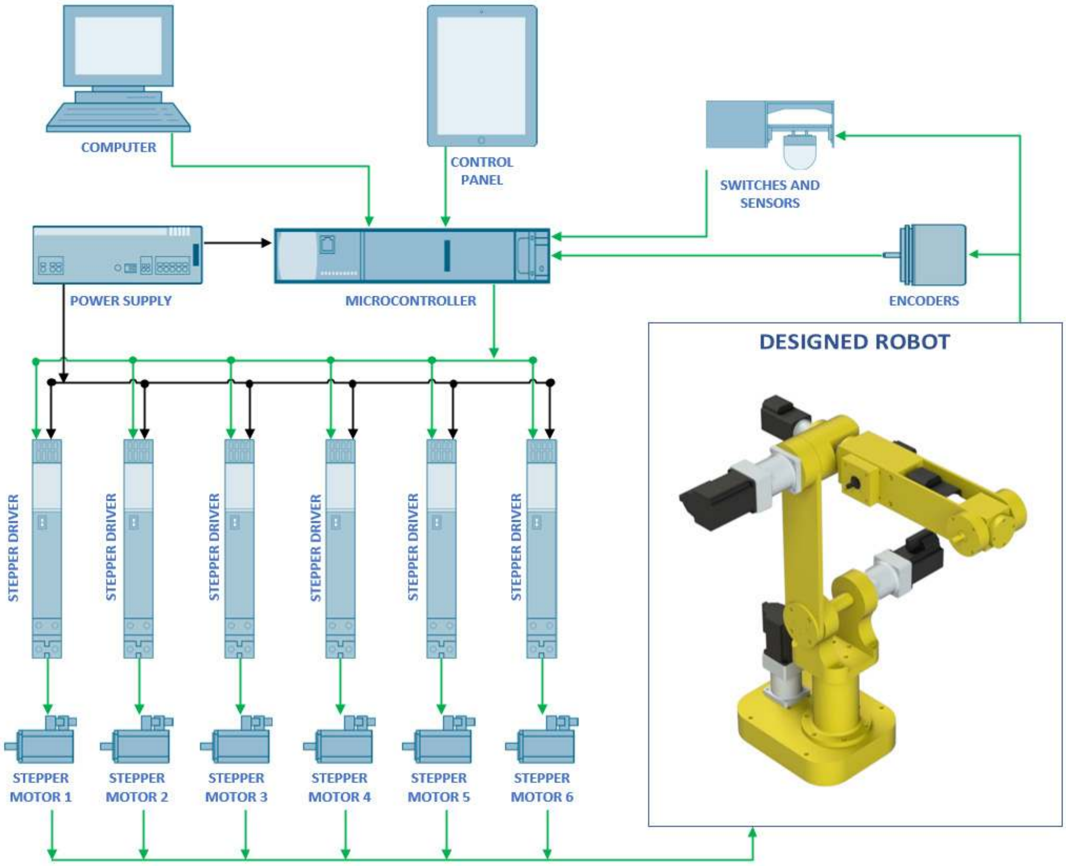


Figure 3.26. System Hardware Topology Diagram

3.3.5. Manufacture and Physical Tests

Within the scope of the study, the following tasks has been carried out so far; preliminary robotic design, forward kinematic analysis, analytical inverse kinematic analysis combined with the geometric approach, static load analysis, design of a 3D model of the robot on SolidWorks, simulation in MATLAB-Simulink environment, component selections for the prototype, procurement of the components, partial manufacturing and assembly, control software and user interface development on MATLAB-AppDesigner, optimization studies on the control software. The initial plan was to manufacture the designed robot on time and conduct physical tests on the actual prototype to validate the theoretical solutions. But unfortunately, some misfortunes happened and manufacture of the mechanical parts of the robot was delayed. Since it was unlikely to finish the prototype manufacturing on time, an alternative solution was implemented. The solution is presented in detail in the next chapter.

4. STUDY ON THE CUSTOM KUKA KR5 ROBOT

Collaboration has been established with a company called KNet Mechatronic Systems in Konya/TURKEY. Within the scope of this collaboration, a similar solution to the inverse kinematics problem explained in the previous chapter was implemented on an actual industrial robotic arm, KR5 ARC-C by KUKA.

4.1.1. Scope of the Study

KNet had an actual robotic arm consisting of only mechanical parts without any electronic components as shown in Figure 4.1. The inverse kinematics solution for this robot was obtained in MATLAB then tested in simulation mode in MATLAB, then the equations are simplified enough to be environment-independent so that they could be transferred to the ST Language that SIEMENS motion controller units can be programmed by.

Figure 4.1. KUKA KR-5 ARC-C Robot Mechanical Body

The inverse kinematic solution for KUKA KR5 is not the same as the solution for the designed robot this study was focused on developing. Most of the technical details of the work on the KUKA KR5 robot will not be shared here due to the confidentiality agreement with KNet. Below are the details of the KUKA KR5 study that KNet has allowed to be published.

4.1.2. Partial Inverse Kinematic Solution

The geometric model to start working on the inverse kinematic problem is not the same as the designed robot previously explained. The geometric model for the KUKA KR5 robot that is used to calculate the position for the first 3 joints while the wrist is above the shoulder is represented in Figure 4.2.

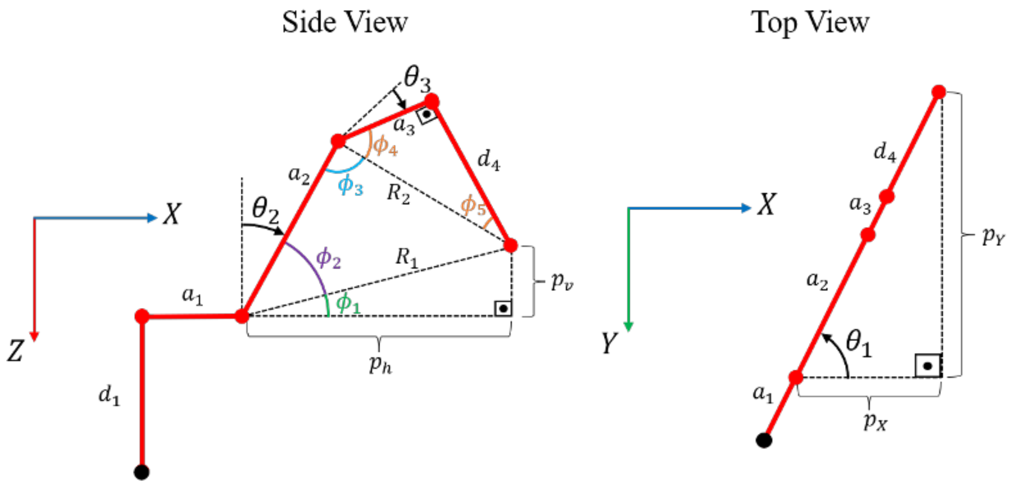


Figure 4.2. Geometric Model of KUKA KR5 when Wrist is Above Shoulder

Equations derived from the geometric model for the wrist-above configuration are as follows:

$$p_v = p_z - d_1 \quad 3.1$$

$$p_h = \sqrt{(p_x - a_1)^2 + (p_y - a_1)^2} \quad 3.2$$

$$R_1 = \sqrt{p_h^2 + p_v^2} \quad 3.3$$

$$R_2 = \sqrt{a_3^2 + d_4^2} \quad 3.4$$

$$\phi_1 = \tan^{-1}\left(\frac{p_v}{p_h}\right) \quad 3.5$$

$$\phi_2 = \cos^{-1}\left(\frac{R_2^2 - a_2^2 - R_1^2}{-2a_2R_1}\right) \quad 3.6$$

$$\phi_3 = \cos^{-1}\left(\frac{R_1^2 - R_2^2 - a_2^2}{-2R_2a_2}\right) \quad 3.7$$

$$\phi_5 = \tan^{-1}\left(\frac{a_3}{d_4}\right) \quad 3.8$$

$$\phi_4 = \frac{\pi}{2} - \phi_5 \quad 3.9$$

$$\theta_1 = \frac{p_y}{p_x} \quad 3.10$$

$$\theta_2 = \frac{\pi}{2} - \phi_2 - \phi_1 \quad 3.11$$

$$\theta_3 = \pi - \phi_3 - \phi_4 \quad 3.12$$

The geometric model for the KUKA KR5 robot that is used to calculate the position for the first 3 joints while the wrist is below the shoulder is represented in Figure 4.3.

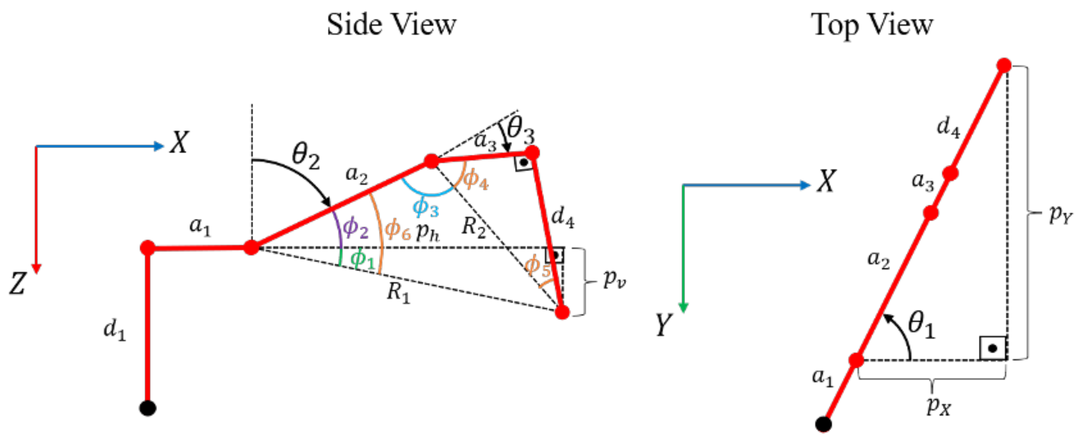


Figure 4.3. Geometric Model of KUKA KR5 when Wrist is Below Shoulder

Equations derived from the geometric model for the wrist-below configuration are as follows:

$$p_v = d_1 - p_z \quad 3.13$$

$$p_h = \sqrt{(p_x - a_1)^2 + (p_y - a_1)^2} \quad 3.14$$

$$R_1 = \sqrt{p_h^2 + p_v^2} \quad 3.15$$

$$R_2 = \sqrt{a_3^2 + d_4^2} \quad 3.16$$

$$\phi_1 = \tan^{-1}\left(\frac{p_v}{p_h}\right) \quad 3.17$$

$$\phi_6 = \cos^{-1}\left(\frac{R_2^2 - a_2^2 - R_1^2}{-2a_2R_1}\right) \quad 3.18$$

$$\phi_2 = \phi_6 - \phi_1 \quad 3.19$$

$$\phi_3 = \cos^{-1}\left(\frac{R_1^2 - R_2^2 - a_2^2}{-2R_2a_2}\right) \quad 3.20$$

$$\phi_5 = \tan^{-1}\left(\frac{a_3}{d_4}\right) \quad 3.21$$

$$\phi_4 = \frac{\pi}{2} - \phi_5 \quad 3.22$$

$$\theta_1 = \frac{p_y}{p_x} \quad 3.23$$

$$\theta_2 = \frac{\pi}{2} - \phi_2 \quad 3.24$$

$$\theta_3 = \pi - \phi_3 - \phi_4 \quad 3.25$$

4.1.3. Mechanical Design

After obtaining the full inverse kinematic solution, most steps of which were skipped here for confidentiality reasons, a real-size 3D CAD model of the robot KUKA KR5 ARC-C has been downloaded from the KUKA download center [51] and rearranged to create a simulation before applying the theoretical solution on the actual robot. Below in Figure 4.4 and Figure 4.5 are the rendered views of the KUKA robot's CAD model.

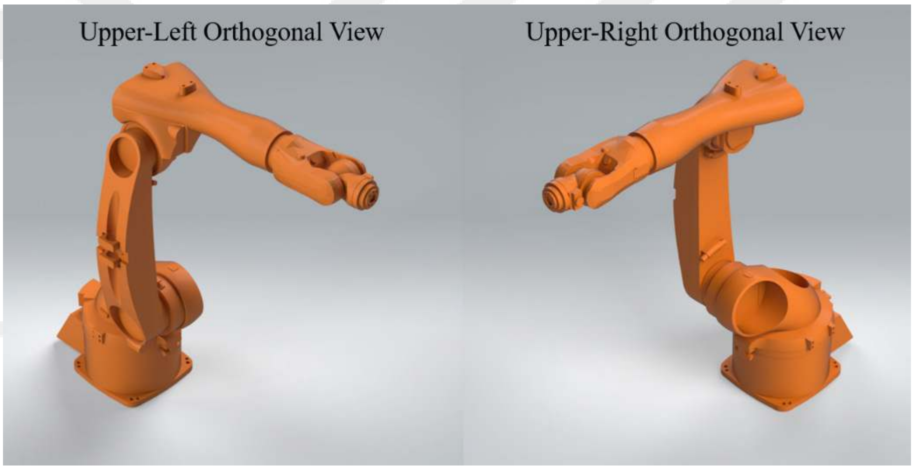


Figure 4.4. KUKA KR5 Robotic Arm Orthogonal Views

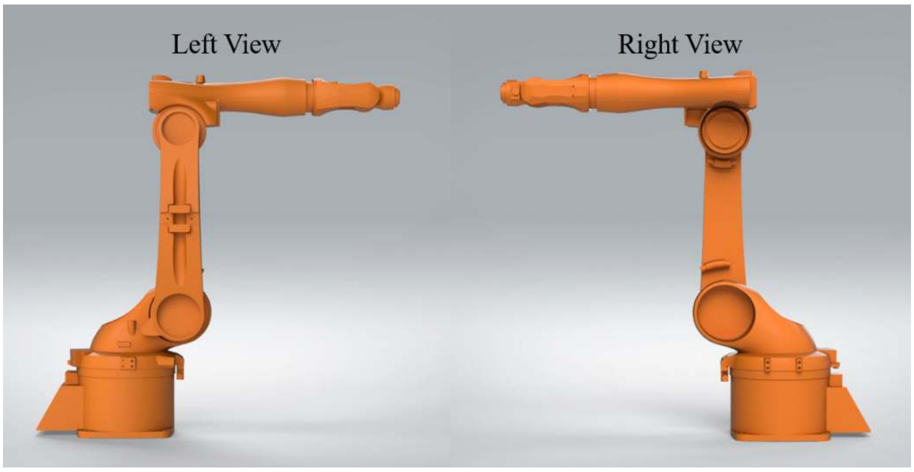


Figure 4.5. KUKA KR5 Robotic Arm Side Views

4.1.4. Simulation

Once the CAD model was ready then the model was regenerated in the MATLAB Simulink environment to verify the results both visually and numerically as shown below in Figure 4.6:

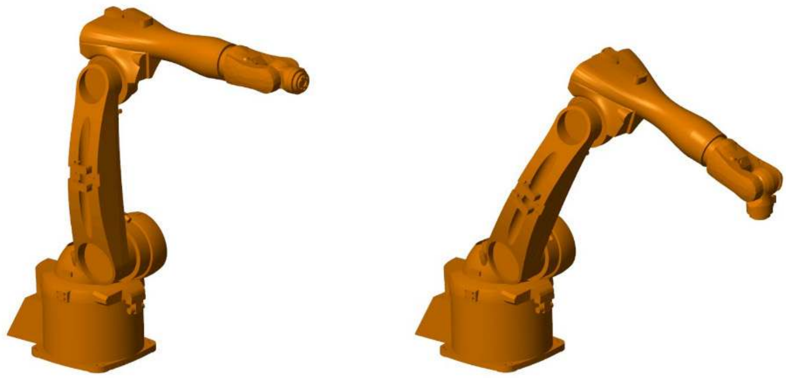


Figure 4.6. KUKA KR5 Generated 3D Model in Simulink

The solution was first tested numerically on MATLAB, then it was verified visually on the 3D model in Simulink. The tests on the Simulink model were close to perfect. Some glitches have been encountered for specific orientations of the robot for specific positions in the workspace and they have all been eliminated. Thus, a firmly verified solution has been obtained on MATLAB level. Some of the results from these tests are presented in the next chapter.

4.1.5. Workspace Estimation

As explained in the Workspace Estimation section for the designed, the famous Monte Carlo method was used to estimate the workspace of the custom KUKA robotic arm. Below in Figure 4.7 is the estimated workspace of the custom KUKA KR5 robot.

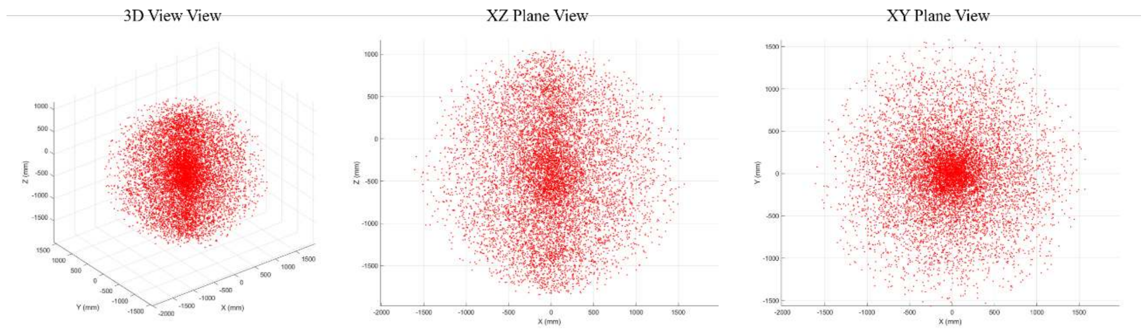


Figure 4.7. Workspace Estimation of the Custom KUKA KR5 Robot

4.1.6. Implementation on the Physical Model

Once everything was fine in the MATLAB simulation, then it was time to move the solution out of the MATLAB environment to the industrial level. Therefore, the expressions in the solution were simplified so that they could be transferred into the Structured Text (ST) language which is one of the programming languages defined by the standard IEC 61131-3 [52] and used for industrial control systems. Then the control software written in the ST language was first tested numerically on the software SIEMENS SCOUT. After that, a pre-validation process to the control software is applied by providing different input values into the controller and checking the numerical results. Once everything was fine, then a simulation environment called Scene Viewer by B&R was used to validate the results visually before working on the physical model. A real-time communication was established between SIEMENS motion controller and the software Scene Viewer through a local Open Platform Communications (OPC) server. After all the results were visually verified for different configurations, custom motors were mounted on the KUKA KR5's body shown in Figure 4.8, and the control panel was prepared to house the electronic components as shown in Figure 4.9.

Figure 4.8. KUKA KR-5 ARC-C Robot with Motors Mounted

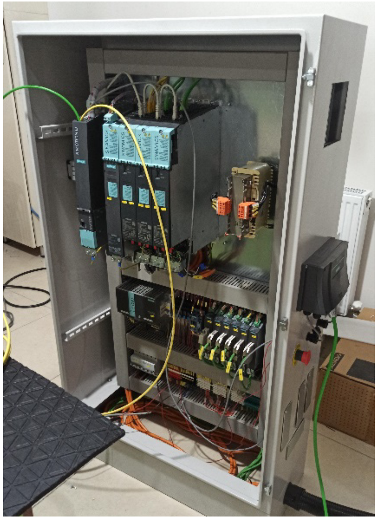


Figure 4.9. KUKA KR-5 ARC-C Robot Control Panel

Within the scope of the work in this section, most tasks that are related to the implementation of the solution to the KUKA KR5 robot were mainly KNet’s responsibility, hence after the MATLAB phase, most tasks have been carried mainly by KNet. Other technical details will remain confidential. The results of this work will be explained in the next chapter.

5. RESULTS AND DISCUSSION

As explained earlier, this study has two separate phases, so the results were divided into two sections. In the first section the results obtained from the designed robot were shared. And the results from the custom KUKA robot were shared in the second section.

5.1. Results From the Designed Robot

After obtaining a sound solution to the inverse kinematics problem, various visual and numerical tests have been conducted for different configurations. Some insignificant flaws for certain configurations have been encountered and fixed completely. A fully-verified model in simulation is available now. Some visual and numerical results obtained during tests are presented in this section. MATLAB codes used to simulate continuous motion of the robot is presented in Appendix-7.

5.1.1. Simulation Results for Linear Motion Along Z-Axis

Results of a test for the motion along the Z-Axis in which the robot starts moving from the coordinates $(X{:}\,400, Y{:}\,0, Z{:}\,599)$ to the coordinates $(X{:}\,400, Y{:}\,0, Z{:}\,-200)$ are presented in Figure 5.1 and Figure 5.2. Orientation of the TCP for these tests is fixed as normal to the plane YZ and all units are in mm. A linear path between the initial and the final coordinates was created by generating 100 equal stops in between the two points, which are called waypoints. Kinematic solution has been obtained for each waypoint along the path described and the resulting angle values were sent into the virtual actuators to move the simulation model.

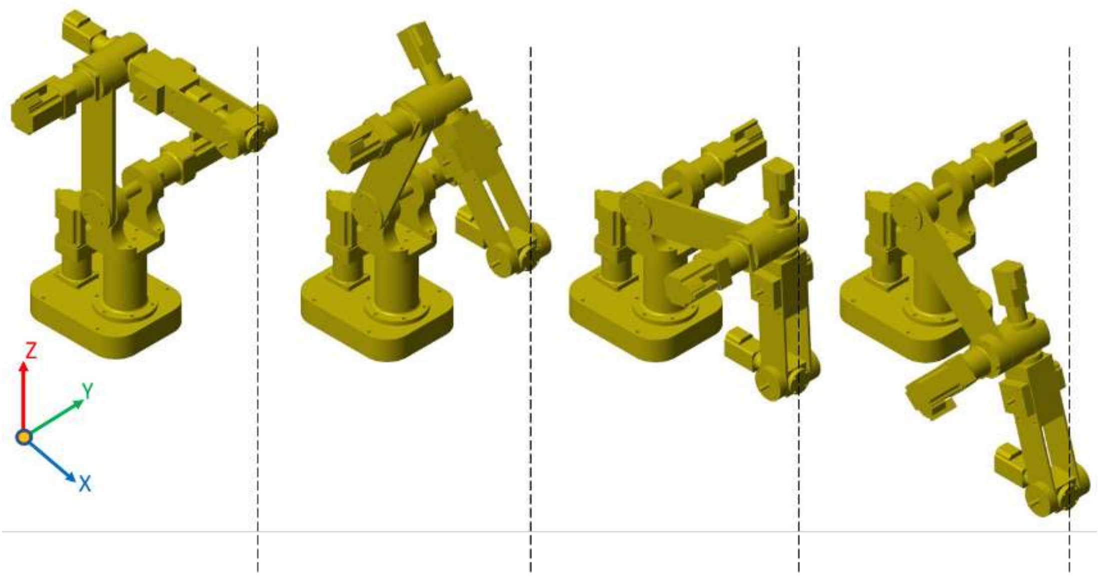


Figure 5.1. Visual Simulation for the Linear Motion Along the Z-Axis

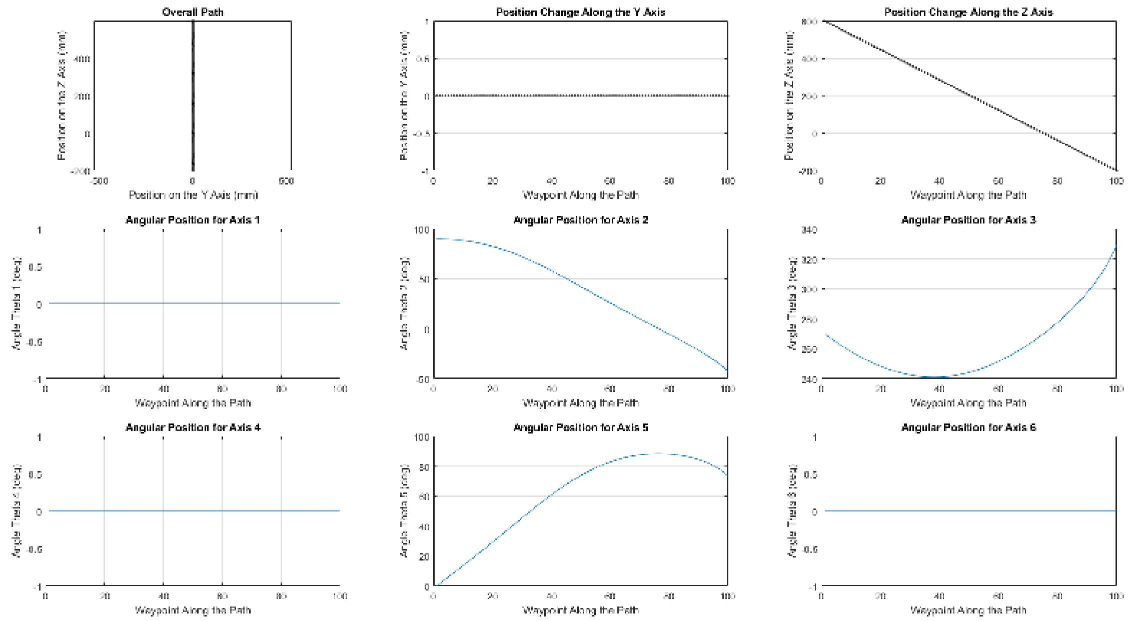


Figure 5.2. Graphs for the Linear Motion Along the Z-Axis from 599mm to 300mm

5.1.2. Simulation Results for Linear Motion Along X-Axis

Results of a test for the motion along the X-Axis in which the robot starts moving from the coordinates $(X\colon 300, Y\colon 250, Z\colon 400)$ to the coordinates $(X\colon -300, Y\colon 250, Z\colon 400)$ are shown in Figure 5.3 and Figure 5.4. Orientation of the TCP for these tests is fixed as normal to the plane XY and all units are in mm. A linear path between the initial and the final coordinates was created by generating 100 waypoints in between the two points.

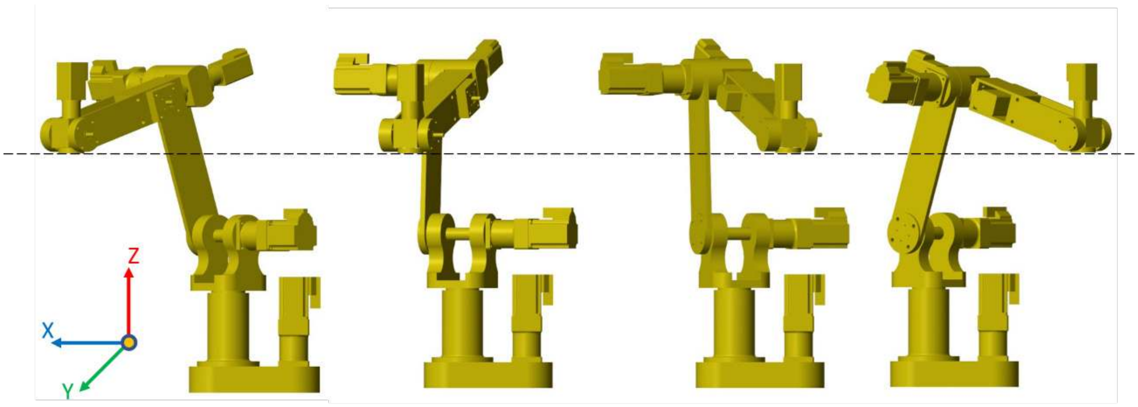


Figure 5.3. Visual Simulation for the Linear Motion Along the Z-Axis

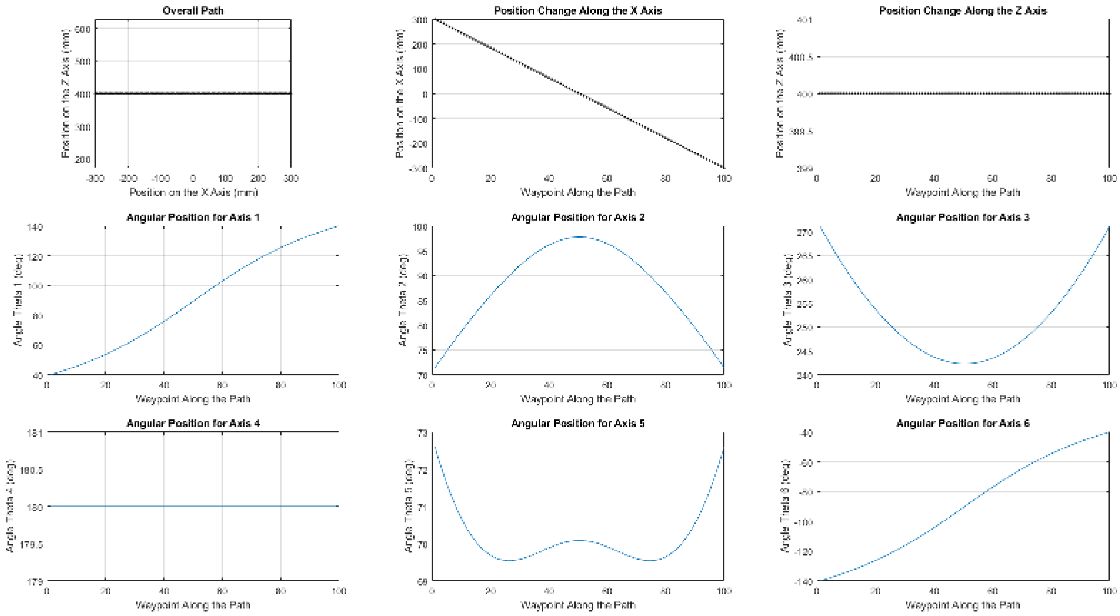


Figure 5.4. Graphs for the Linear Motion Along the X-Axis from -300mm to +300mm

5.1.3. Simulation Results for Circular Motion on YZ-Plane

Results of a test for the circular motion on the plane YZ in which the robot draws a circle with a radius of $r = 150mm$ around the center $(X{:}\,400, Y{:}\,250, Z{:}\,250)$ are in Figure 5.5 and Figure 5.6. Orientation of the TCP for these tests is fixed as normal to the plane YZ and all units are in millimeters. A circular path centered at the coordinates above with the radius provided was created by generating 100 waypoints along the outline of the circle.

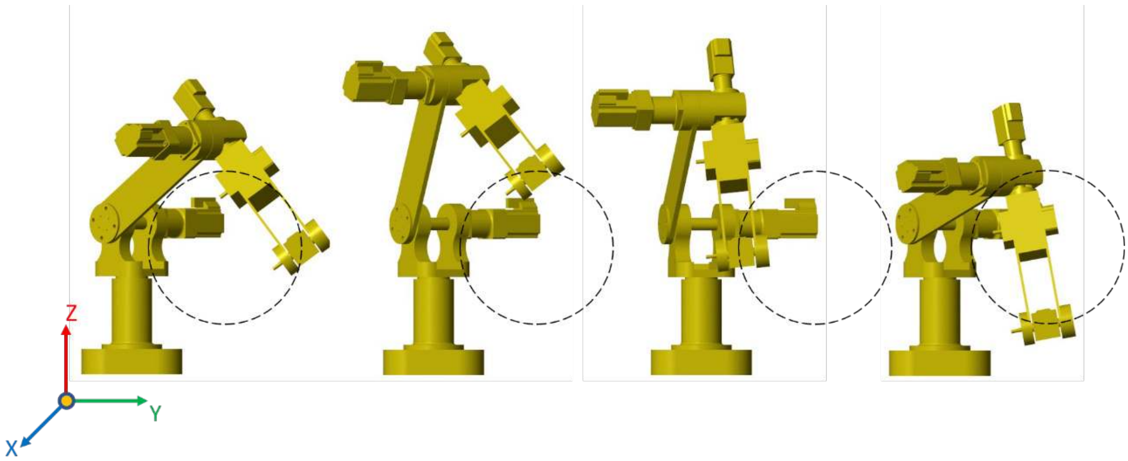


Figure 5.5. Visual Simulation for the Circular Motion on the Y-Z Plane

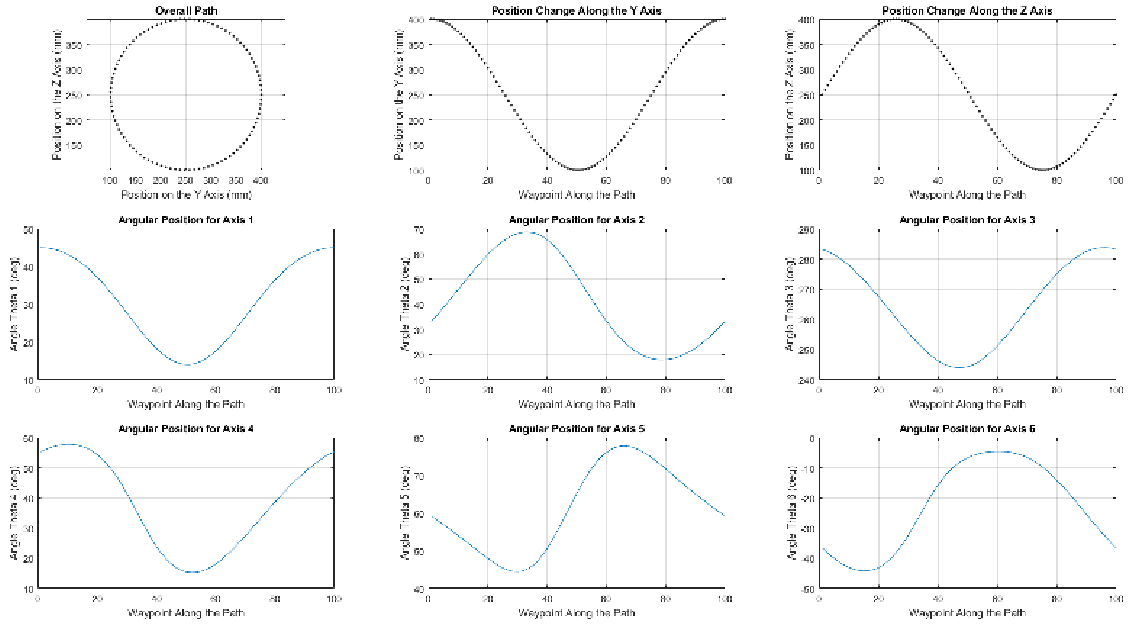


Figure 5.6. Graphs for the Circular Motion on the Y-Z Plane

5.1.4. Simulation Results for Circular Motion on XY-Plane

Results of a test for the circular motion on the plane XY in which the robot draws an arc with a radius of $r = 150mm$ around the center $(X{:}\,250, Y{:}\,250, Z{:}\,650)$ are presented in Figure 5.7 and Figure 5.8. Orientation of the TCP for these tests is fixed as normal to the plane XY facing towards sky and all units are in millimeters. A circular path centered at the coordinates above with the radius provided was created by generating 100 waypoints along the outline of the arc.

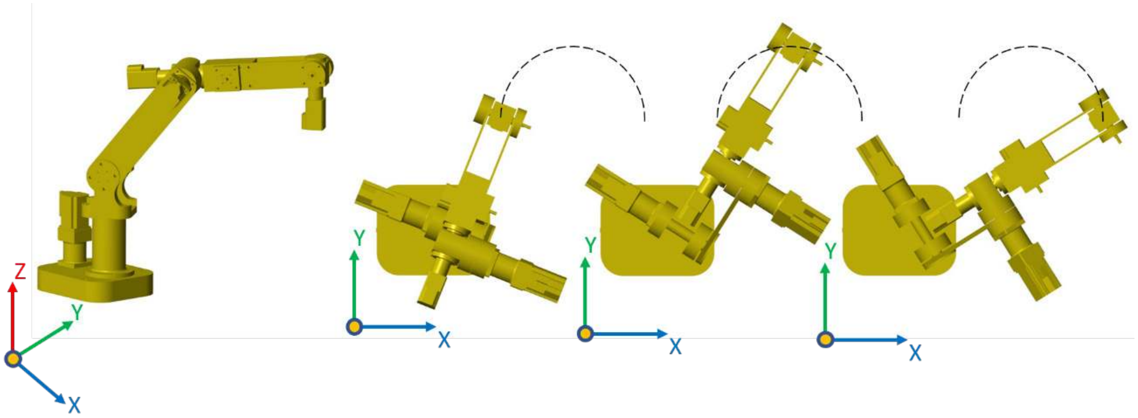


Figure 5.7. Visual Simulation for the Circular Motion on the X-Y Plane

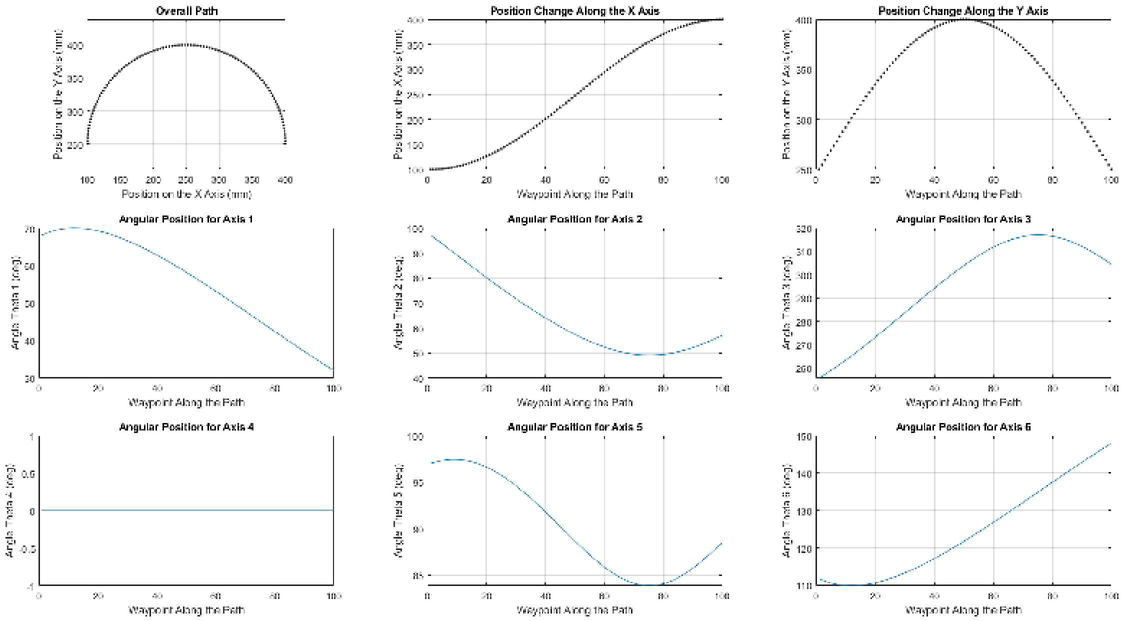


Figure 5.8. Graphs for the Circular Motion on the X-Y Plane

5.1.5. Simulation Results for Sinusoidal Motion on XZ-Plane

Results of a test for the sinusoidal motion on the plane XZ in which the robot draws a sinusoidal wave by drawing a circle with a radius of $r = 100mm$ while moving horizontally along the X axis which creates a sinusoidal wave with an amplitude of 100mm and a period of 200mm are presented in Figure 5.9 and Figure 5.10. The starting point of the sinus wave is defined as $(X{:}-200, Y{:}-400, Z{:}400)$ while the end point is $(X{:}200, Y{:}-400, Z{:}400)$. Orientation of the TCP for these tests is fixed as normal to the plane XZ facing further away from the center of the robot and all units are in millimeters. A sinusoidal path that starts and ends at points described above with the radius provided was created by generating 100 waypoints along the sinus wave.

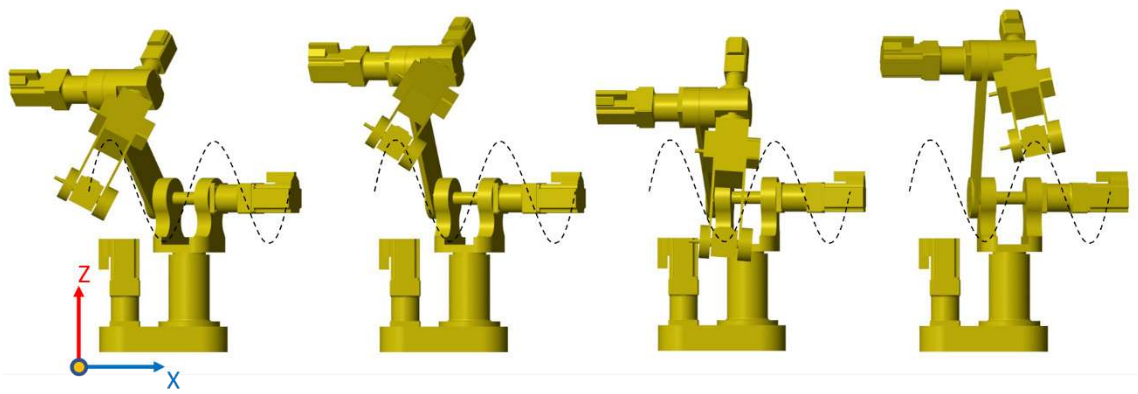


Figure 5.9. Visual Simulation for the Sinusoidal Motion on the X-Z Plane

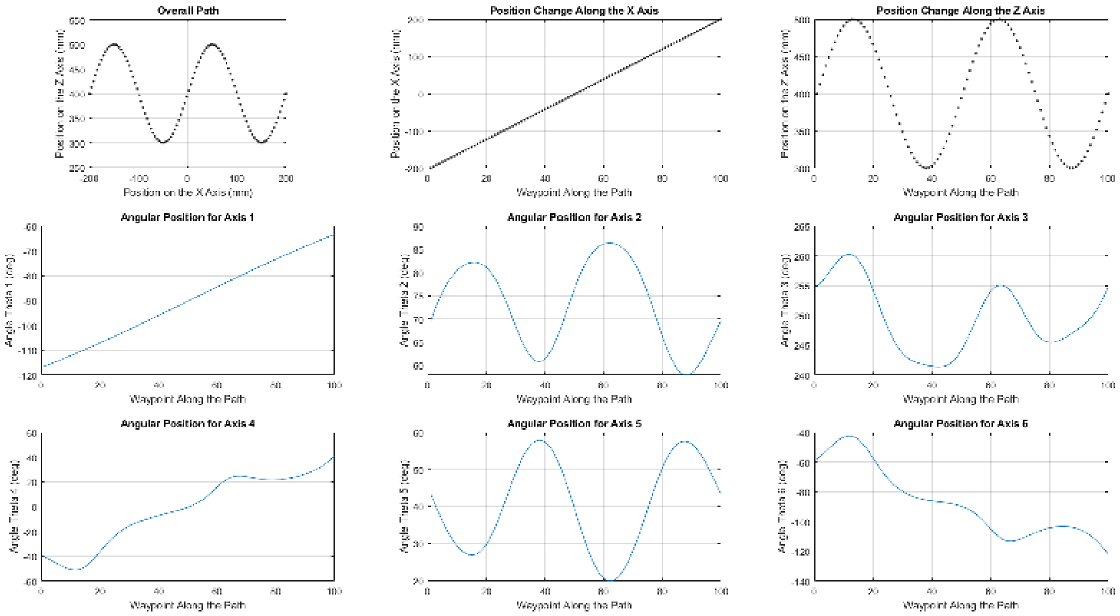


Figure 5.10. Graphs for the Sinusoidal Motion on the X-Z Plane

5.1.6. Simulation Results for Interplanar Motion on Multiple Planes

Results of a test for the interplanar motion in which the robot draws a diagonal straight line starting from the point $(X{:}\,200, Y{:}\,-300, Z{:}\,400)$ to the point $(X{:}\,550, Y{:}\,-50, Z{:}\,500)$ are presented below in Figure 5.11 and Figure 5.12. Orientation of the TCP for these tests is fixed as normal to the plane YZ facing further away from the center of the robot and all units are in millimeters. A linear path that starts and ends at points described above was created by generating 100 waypoints along the orthogonal line.

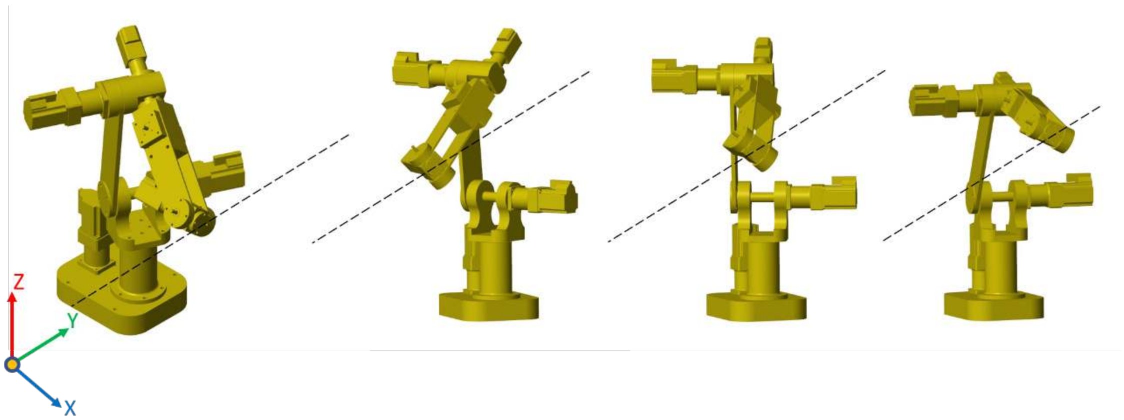


Figure 5.11. Visual Simulation for the Diagonal Interplanar Motion

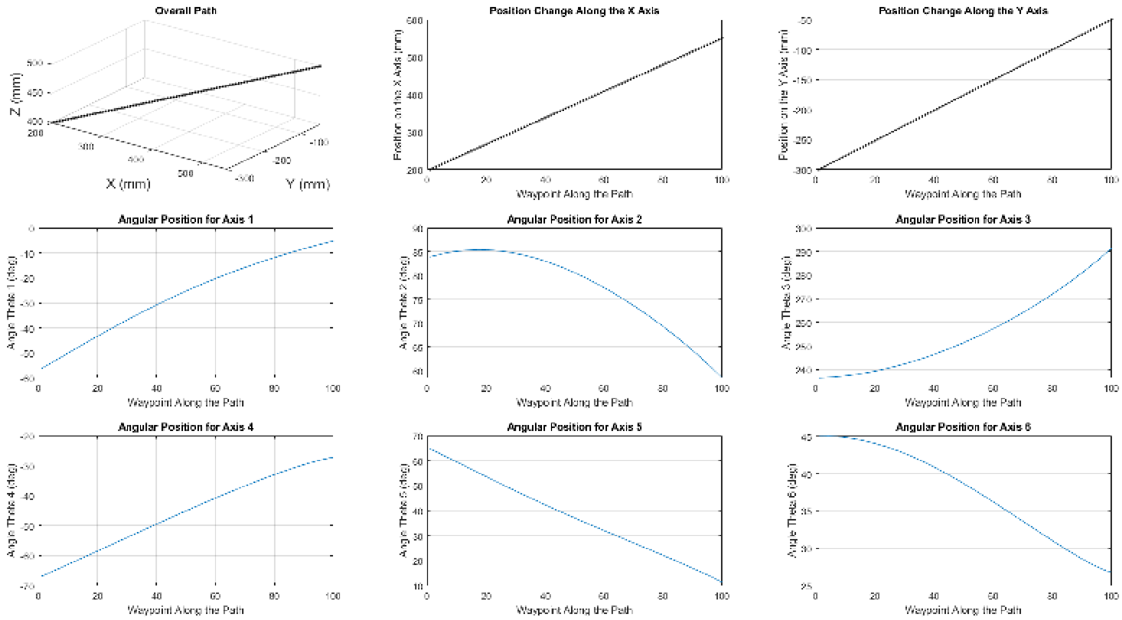


Figure 5.12. Graphs for the Diagonal Interplanar Motion

The solution has been completely validated, but unfortunately the prototype manufacturing couldn't be completed on time as explained earlier. Alternatively, a similar genuine solution has been developed and tested on an actual industrial robotic arm KR5 ARC-C by KUKA in cooperation with the company KNet Mechatronics.

5.2. Results from the KUKA KR5 Robot

After following similar steps as the designed robot, a unique inverse kinematic solution has been obtained, tested inclusively first in the simulation environment then in the real-time simulation environment using an actual motion controller unit by SIEMENS. Some figures and graphs from simulation-level tests and physical tests are presented below.

5.2.1. Simulation Results for Linear Motion Along Z-Axis

Results of a test for the motion along the Z-Axis in which the robot starts moving from the coordinates $(X{:}\,1015, Y{:}\,0, Z{:}\,1100)$ to the coordinates $(X{:}\,1015, Y{:}\,0, Z{:}\,-300)$ are below in Figure 5.13 and Figure 5.14. Orientation of the TCP for these tests is fixed as normal to the plane YZ facing further away from the robot's center and all units are in mm. A linear path between the initial and the final coordinates was created by generating 100 equal waypoints in between the two points.

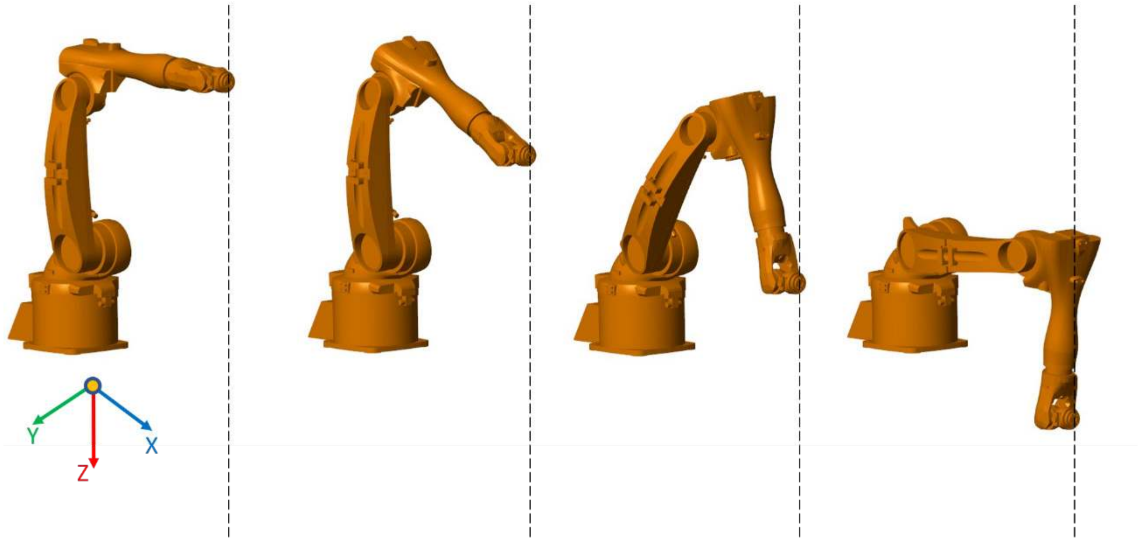


Figure 5.13. KUKA KR5 Visual Simulation for the Linear Motion Along the Z-Axis

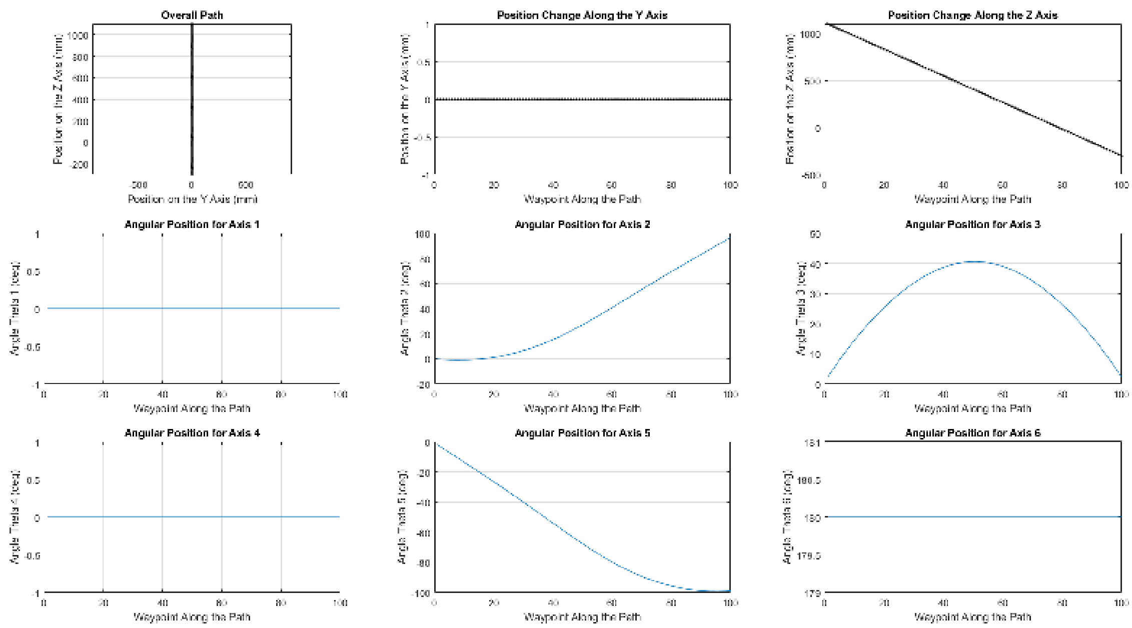


Figure 5.14. KUKA KR5 Graphs for the Linear Motion Along the Z-Axis from 1100mm to -300mm

5.2.2. Simulation Results for Linear Motion Along X-Axis

Results of a test for the linear motion along the X-Axis in which the robot starts moving from the coordinates $(X{:}\,0, Y{:}\,700, Z{:}\,1000)$ to the coordinates $(X{:}\,800, Y{:}\,700, Z{:}\,1000)$ are presented in Figure 5.15 and Figure 5.16. Orientation of the TCP for these tests is fixed as normal to the plane XY facing towards the ground and all units are in mm. A linear path between the initial and the final coordinates was created by generating 100 waypoints in between the two points.

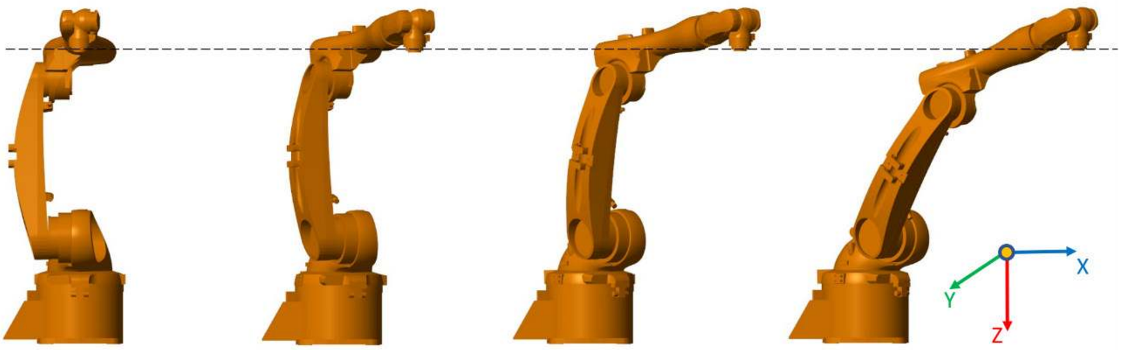


Figure 5.15. KUKA KR5 Visual Simulation for the Linear Motion Along the X-Axis

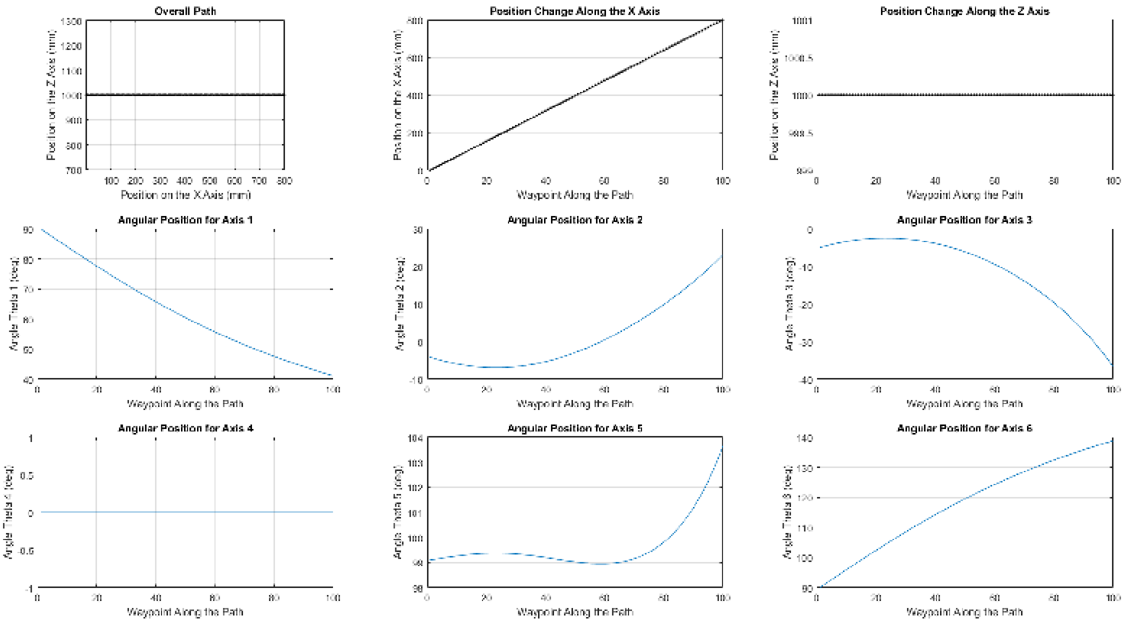


Figure 5.16. KUKA KR5 Graphs for the Linear Motion Along the X-Axis from 0mm to 800mm

5.2.3. Simulation Results for Circular Motion on XY-Plane

Figure 5.17 and Figure 5.18 are the results of a test for the circular motion on the plane XY in which the robot draws a circle with a radius of $r = 250mm$ around the center $(X: 500, Y: -500, Z: 1300)$. Orientation of the TCP for these tests is fixed as normal to the plane XY facing towards the sky and all units are in millimeters. A circular path centered at the coordinates above with the radius provided was created by generating 100 waypoints along the outline of the circle.

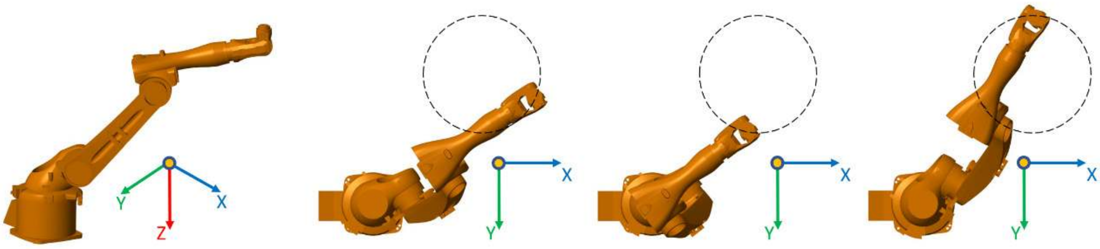


Figure 5.17. KUKA KR5 Visual Simulation for the Circular Motion on the XY-Plane

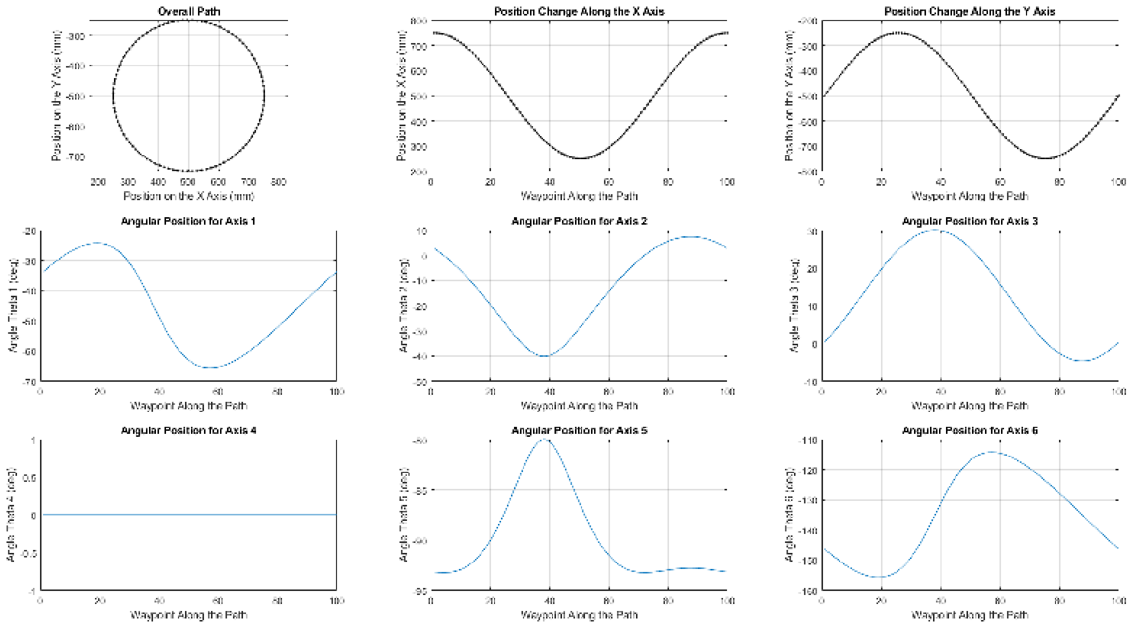


Figure 5.18. KUKA KR5 Graphs for the Circular Motion on the XY-Plane

Once the solution is verified, physical tests have been conducted on the actual industrial robotic arm KUKA KR5. A variety of tests have been conducted for different configurations. Some non-critical glitches for certain configurations have been encountered and troubleshot. A completely verified solution that works on a fully-functional robotic arm is available now whose inverse kinematic solution and control software has been developed peculiar to this study. Below are some figures from the actual physical tests of KUKA KR5.

5.2.4. Physical Test Results for Linear Motion Along Z-Axis

A test for the linear motion along the Z-Axis has been conducted and the results were shared in Figure 5.19. Orientation of the TCP for this test is fixed as normal to the plane YZ. A linear path consisting of many waypoints between the initial and the final coordinates was created and the waypoints with their solutions were fed into the motor drive units through the motion controller unit.

Figure 5.19. Physical Test of KUKA KR5 for the Linear Motion Along the Z-Axis

5.2.5. Physical Test Results for Linear Motion Along Z-Axis

Once the solution and the control software has been verified via several physical tests in a laboratory, the robotic arm KUKA KR5 was demonstrated at an expo in Konya/TURKEY. In Figure 5.20, the robot draws a rectangle on the plane YZ while keeping the orientation of the TCP as normal to the plane YZ. A set of linear paths consisting of many waypoints between the corner coordinates of the rectangle was created and the waypoints with their solutions were fed into the motor drive units through the motion controller unit. Since the following figure was created using a video that was shot at the expo with no stabilizer, the perspective is not fixed in the figure. Therefore, a red rectangle with dotted lines was used to indicate the path, but it was not stretched to fit into the perspective for the purpose of using identical rectangles on all three positions.

Figure 5.20. Physical Demonstration of KUKA KR5 at an Expo

The solution has been completely validated on a physical model which runs solely on the custom inverse kinematic solution and the software developed within the scope of this study. The solution is completely independent from environment since no toolbox or specific package of any software was used. It runs completely on basic mathematical expressions that can be represented in almost any programming language available.

6. CONCLUSIONS

The use of robots and especially industrial robotic arms grows day by day in order to; increase mass production, reduce costs, and reduce dependence on humans in the industry for improved stability in production. But the theory behind these industrial robots is not widely known by the public, or even by most robotic/automation companies. The academic literature in the field, in addition to being difficult to process, is heavily focused on theoretical solutions with no prototype development and they are mostly based on academic toolboxes that limit the solution to a single environment. In industry, the few robot manufacturers use custom programming languages to control their robots and access to the control software by the user is very limited as they manufacturer must keep their products as a closed box system whose critical backend software cannot be interfered by others.

Within the scope of the study; a detailed CAD model, advanced load calculations for the model, fully functional inverse kinematic solution, user interface, verified simulation, incomplete prototype of the desktop-size robotic manipulator have been developed. As the initial plan to develop a fully-functional prototype failed because of the delay in manufacturing of the mechanical parts, collaboration has been established with a company and the solutions have been implemented on an actual industrial robotic arm by KUKA. Within the context of the second part of the study; a similar method was used to obtain theoretical solutions, then the results were tested in simulation environment, verified by real-time simulation on the actual hardware, tested on the physical model and demonstrated at an expo. In the second part, the robot by KUKA didn’t have any electronic components on it, so only the mechanical body of the robot was used for the study. Then a custom control software was developed based the on theoretical solutions and custom electronic components have been mounted on the mechanical body. Details and results have been shared in related chapters. Table 5.1 is the comparison table for the two custom robots that are the focus of this study and two other robot manufacturers regarding the general specifications of the industrial robotic manipulators they manufacture.

Table 5.1. Comparison Table for Different Robots

	Designed Robot	**Custom KUKA KR5**	**KUKA KR4** [53-55]	**STAUBLI TX2-40** [56-58]
Programming Language	MATLAB	ST (Structured Text)	KRL	VAL3
Architecture	Open Source	Closed Source	Closed Source	Closed Source
Environment Dependency	Free	Free	Dependent	Dependent

The main difference between this study compared to most of the studies in the literature is that the results are not dependent on academic tools, instead, they are implementable on actual industrial robots as explained and proven throughout the thesis.

RECOMMENDATIONS

Based on the results of this thesis, further studies and improvements may be conducted as follows:

1) An even more detailed guide on the technical side that covers all aspects of the solution from the preliminary design phase to the fully functional prototype may be prepared. And this guide can be used for educational purposes to help graduate and undergraduate students who are interested in the field understand the subject clearly as the study offers a compound solution with both theory and application.
2) By mounting extra attachments on the TCP of the robot, such as welding torches, paint sprayers, grippers, etc. use cases of the robot may be extended and it can be used for lightweight industrial or educational purposes.
3) The practical implementation on the KUKA KR5 robot may be extended to many other commercial robots that are not functioning well enough thus stay idle or completely broken and not feasible to replace all original components provided by the manufacturer.

REFERENCES

[1] "Top 21 Industrial Robotics Companies in 2018 | Global Industrial Robotics Market," *Technavio*, Feb. 05, 2019. https://blog.technavio.org/blog/top-21-companies-in-the-industrial-robotics-market (accessed May 28, 2022).

[2] "Setting up Pickit with Stäubli — Pickit 3.1 documentation." https://docs.pickit3d.com/en/latest/robot-integrations/staubli/index.html (accessed May 28, 2022).

[3] D. Terdiman, "Autodesk's Lego model-building robot is the future of manufacturing," *Fast Company*, Jul. 20, 2018. https://www.fastcompany.com/90204615/autodesks-lego-model-building-robot-is-the-future-of-manufacturing (accessed May 22, 2022).

[4] D. J. Todd, *Fundamentals of Robot Technology*. Dordrecht: Springer Netherlands, 1986. doi: 10.1007/978-94-011-6768-0.

[5] "Which Countries and Industries Use the Most Robots?" https://www.stlouisfed.org/on-the-economy/2019/november/robots-affecting-local-labor-markets (accessed May 28, 2022).

[6] IFR, "International Federation of Robotics," *IFR International Federation of Robotics*. https://ifr.org (accessed Jul. 02, 2022).

[7] D. S. Bieller, "Director Statistical Dpt. Vice Chair IFR Service Robot Group," p. 43, 2021.

[8] S. Soyguder and T. Abut, "Haptic industrial robot control with variable time delayed bilateral teleoperation," *Industrial Robot: An International Journal*, vol. 43, no. 4, pp. 390–402, Jan. 2016, doi: 10.1108/IR-12-2015-0213.

[9] S. Soyguder and W. Boles, "SLEGS robot: development and design of a novel flexible and self-reconfigurable robot leg," *Industrial Robot: An International Journal*, vol. 44, no. 3, pp. 377–391, Jan. 2017, doi: 10.1108/IR-03-2016-0109.

[10] S. Soyguder and H. Alli, "Kinematic and dynamic analysis of a hexapod walking–running–bounding gaits robot and control actions," *Computers & Electrical Engineering*, vol. 38, no. 2, pp. 444–458, Mar. 2012, doi: 10.1016/j.compeleceng.2011.10.008.

[11] D. Korkmaz, Z. H. Akpolat, S. Soygüder, and H. Alli, "Dynamic simulation model of a biomimetic robotic fish with multi-joint propulsion mechanism," *Transactions of the Institute of Measurement and Control*, vol. 37, no. 5, pp. 684–695, May 2015, doi: 10.1177/0142331214565710.

[12] "SmartBird | Festo USA." https://www.festo.com/us/en/e/about-festo/research-and-development/bionic-learning-network/highlights-from-2010-to-2012/smartbird-id_33686/ (accessed May 22, 2022).

[13] "Honda Global | history." https://global.honda/innovation/robotics/ASIMO/history.html (accessed May 22, 2022).

[14] A. Bayram and A. S. Duru, "Design And Control Of A Rehabilitation Robot Manipulator For Head–Neck Orthopaedic Disorders," in *International Journal of Robotics and Automation 2022*, Jul. 2004, vol. 37, no. 6. doi: 10.2316/J.2022.206-0680.

[15] J. J. Craig, *Introduction to Robotics: Mechanics and Control*. Pearson Educación, 2005.

[16] "6 axis industrial robots TX2-140| Stäubli," *Staubli*. https://www.staubli.com/gb/en/robotics/products/industrial-robots/tx2-140.html (accessed May 22, 2022).

[17] "ABB introduces articulated robot, linear gantry combination delivering greater working range, cycle times and flexibility," *Automate*. https://www.automate.org/news/abb-introduces-articulated-robot-linear-gantry-combination-delivering-greater-working-range-cycle-times-and-flexibility (accessed Jun. 11, 2022).

[18] "Gramazio Kohler Research." https://gramaziokohler.arch.ethz.ch/web/e/about/aboutInfrastructure.html (accessed Jun. 11, 2022).

[19] B. Kelechava, "Industrial Robots Make Production Safe and Efficient," *The ANSI Blog*, Jan. 04, 2019. https://blog.ansi.org/2019/01/industrial-robots-safety-production-ria/ (accessed May 22, 2022).

[20] “Spot Welding Articulated robot by KUKA AG.” https://www.directindustry.com/prod/kuka-ag/product-17587-1873609.html (accessed Jun. 11, 2022).
[21] “Automotive paint robots market seeing ‘impressive growth’, says report,” *Robotics & Automation News*, Nov. 16, 2018. https://roboticsandautomationnews.com/2018/11/16/automotive-paint-robots-market-seeing-impressive-growth-says-report/19769/ (accessed May 22, 2022).
[22] R. Blake, “Industrial Robotic Arms Race Leaves Experts Asking ‘What Manufacturing Slowdown?,’” *Forbes*. https://www.forbes.com/sites/richblake1/2019/01/09/industrial-robotic-arms-race-leaves-experts-asking-what-manufacturing-slowdown/ (accessed May 22, 2022).
[23] “Roll Forging - Robotic Forge,” *Robot Partner*, Jun. 01, 2019. https://www.robotpartner.pl/en/roll-forging-robotic-forge/ (accessed Jun. 11, 2022).
[24] “Now factory robots may see better than you with 3D vision - asmag.com.” https://www.asmag.com/showpost/32032.aspx (accessed May 22, 2022).
[25] K. Speights, “How Intuitive Surgical, Inc. Makes Most of Its Money,” *The Motley Fool*, Aug. 12, 2017. https://www.fool.com/investing/2017/08/12/how-intuitive-surgical-inc-makes-most-of-its-money.aspx (accessed May 23, 2022).
[26] “As Intuitive Surgical continues to shine, competitors are entering the fray,” *The Robot Report*, Aug. 04, 2016. https://www.therobotreport.com/as-intuitive-surgical-continues-to-shine-competitors-are-entering-the-fray/ (accessed May 23, 2022).
[27] S. Guo, H. Liu, B. Liu, and Y. Wu, “A Novel Articulated Robot with Active Spherical Joints,” in *Advances in Mechanical Design*, Singapore, 2022, pp. 1891–1908. doi: 10.1007/978-981-16-7381-8_119.
[28] A. H. Rajpar, A. E. Eladwi, I. Ali, and M. B. Ali Bashir, “Reconfigurable Articulated Robot Using Android Mobile Device,” *Journal of Robotics*, vol. 2021, p. e6695198, Feb. 2021, doi: 10.1155/2021/6695198.
[29] C. Emmanuel Agbaraji, U. Henrietta Udeani, H. Chibueze Inyiama, and C. Chikodi Okezie, “Robust Control for a 3DOF Articulated Robotic Manipulator Joint Torque under Uncertainties,” *JERR*, pp. 1–13, Jan. 2020, doi: 10.9734/jerr/2019/v9i417022.
[30] H. Karci and A. Tangel, “Design and Prototype Implementation of a 5-Dof Mobile Robot Arm Based on FPGA,” *Journal of the Faculty Of Engineering and Architecture of Gazi University*, vol. 31, no. 2, 2016, Accessed: May 22, 2022. [Online]. Available: https://avesis.kocaeli.edu.tr/yayin/368bb999-3511-4b55-bf31-38110ffc638e/design-and-prototype-implementation-of-a-5-dof-mobile-robot-arm-based-on-fpga
[31] S. S and M. S.J, “Obstacle avoiding trajectory planning for 5 degree of freedom robot,” in *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, Jul. 2016, pp. 1–5. doi: 10.1109/ICPEICES.2016.7853202.
[32] Z. Xuefeng, W. Jianhui, and W. Xiaofeng, “Upper limb rehabilitation trajectory optimization based on artificial immune genetic algorithm,” in *The 27th Chinese Control and Decision Conference (2015 CCDC)*, May 2015, pp. 1598–1603. doi: 10.1109/CCDC.2015.7162174.
[33] F. Pehlivan and A. Ankarali, “5 Serbestlik Dereceli Robot Kolunun Kinematik Hesaplamaları Ve PID ile Yörünge Kontrolü,” *Proceedings of Trc-IFToMM Symposium on Theory of Machines and Mechanisms*, pp. 367–373, 2015.
[34] C.-Y. Tsai, C.-C. Wong, C.-J. Yu, C.-C. Liu, and T.-Y. Liu, “A Hybrid Switched Reactive-Based Visual Servo Control of 5-DOF Robot Manipulators for Pick-and-Place Tasks,” *IEEE Systems Journal*, vol. 9, no. 1, pp. 119–130, Mar. 2015, doi: 10.1109/JSYST.2014.2358876.
[35] C. D. Bellicoso, L. R. Buonocore, V. Lippiello, and B. Siciliano, “Design, modeling and control of a 5-DoF light-weight robot arm for aerial manipulation,” in *2015 23rd Mediterranean Conference on Control and Automation (MED)*, Jun. 2015, pp. 853–858. doi: 10.1109/MED.2015.7158852.
[36] A. Razzaghian and R. K. Moghaddam, “Fuzzy sliding mode control of 5 DOF upper-limb exoskeleton robot,” in *2015 International Congress on Technology, Communication and Knowledge (ICTCK)*, Nov. 2015, pp. 25–32. doi: 10.1109/ICTCK.2015.7582642.

[37] "Staubli Product Catalog." https://www.staubli.com/global/en/robotics/products/industrial-robots.html (accessed May 23, 2022).
[38] "KUKA KR 4 R600," *KUKA AG*. https://www.kuka.com/en-de/products/robot-systems/industrial-robots/kr-4-agilus (accessed May 23, 2022).
[39] "KUKA KR 6 R900-2," *KUKA AG*. https://www.kuka.com/en-de/products/robot-systems/industrial-robots/kr-agilus (accessed May 23, 2022).
[40] "Kawasaki Robot Catalog," *Kawasaki Robotics*. https://kawasakirobotics.com/asia-oceania/robots-category/small-medium-payloads/ (accessed May 23, 2022).
[41] R. P. Paul, *Robot Manipulators: Mathematics, Programming, and Control : the Computer Control of Robot Manipulators*. Richard Paul, 1981.
[42] R. D. Klafter, *Robotic Engineering : An Integrated Approach*. Englewood Cliffs, N.J. : Prentice Hall, 1989. Accessed: May 29, 2022. [Online]. Available: http://archive.org/details/robotic-engineering-an-integrated-approach
[43] *Denavit Hartenberg Notation (DH parameters)*, (Feb. 24, 2015). Accessed: May 27, 2022. [Online Video]. Available: https://www.youtube.com/watch?v=nuB_7BkYNMk
[44] Z. Bingül and S. Küçük, *Robot Tekniği-1*, 1st ed. Birsen Yayınevi, 2017.
[45] S. B. Niku, *Introduction to Robotics: Analysis, Control, Applications*. John Wiley & Sons Inc, 2010.
[46] M. Kemal Özgören, "Topological analysis of 6-joint serial manipulators and their inverse kinematic solutions," *Mechanism and Machine Theory*, vol. 37, no. 5, pp. 511–547, May 2002, doi: 10.1016/S0094-114X(02)00005-8.
[47] R. J. Schilling, *Fundamentals of robotics: analysis and control*. Simon & Schuster Trade, 1996.
[48] D. L. PIEPER, "The Kinematics of Manipulators Under Computer Control," Ph.D., Ann Arbor, United States. Accessed: May 29, 2022. [Online]. Available: https://www.proquest.com/docview/302481631/citation/A4782C023C33455APQ/1
[49] M. A. Johnson, "PID Control Technology," in *PID Control: New Identification and Design Methods*, J. Crowe, K. K. Tan, T. H. Lee, R. Ferdous, M. R. Katebi, H.-P. Huang, J.-C. Jeng, K. S. Tang, G. R. Chen, K. F. Man, S. Kwong, A. Sánchez, Q.-G. Wang, Y. Zhang, Y. Zhang, P. Martin, M. J. Grimble, D. R. Greenwood, M. A. Johnson, and M. H. Moradi, Eds. London: Springer, 2005, pp. 1–46. doi: 10.1007/1-84628-148-2_1.
[50] J. Zhu and F. Tian, "Kinematics Analysis and Workspace Calculation of a 3-DOF Manipulator," *IOP Conf. Ser.: Earth Environ. Sci.*, vol. 170, p. 042166, Jul. 2018, doi: 10.1088/1755-1315/170/4/042166.
[51] "KUKA Download Center," *KUKA AG*. https://www.kuka.com/en-de/services/downloads (accessed Jun. 05, 2022).
[52] R. Ramanathan, "The IEC 61131-3 programming languages features for industrial control systems," in *2014 World Automation Congress (WAC)*, Aug. 2014, pp. 598–603. doi: 10.1109/WAC.2014.6936062.
[53] "KR 4 AGILUS," *KUKA AG*. https://www.kuka.com/en-de/products/robot-systems/industrial-robots/kr-4-agilus (accessed Jul. 31, 2022).
[54] "KUKA robot controller KR C5," *KUKA AG*. https://www.kuka.com/en-de/products/robot-systems/robot-controllers/kr-c5 (accessed Jul. 31, 2022).
[55] "Application and robot programming from KUKA," *KUKA AG*. https://www.kuka.com/en-de/services/engineering/application-and-robot-programming (accessed Jul. 31, 2022).
[56] "6 axis industrial robots TX2-40 | Stäubli," *Staubli*. https://www.staubli.com/global/en/robotics/products/industrial-robots/tx2-40.html (accessed Jul. 31, 2022).
[57] "CS9 robot controller." https://www.staubli.com/br/en/robotics/products/robot-controllers/cs9-robot-controller.html (accessed Jul. 31, 2022).
[58] "VAL 3 language." https://www.staubli.com/br/en/robotics/products/robot-software/staeubli-robotics-controls/val-3-language.html#tabs-e17c3b5a40-item-c176cd9644-tab (accessed Jul. 31, 2022).

Appendices

Appendix-1: Work Plan

TIME (in Months) / WORK PACKAGES	M.1	M. 2	M. 3	M. 4	M. 5	M. 6	M. 7	M. 8	M. 9
1) Determination of Technical Specifications of Robotic Manipulator									
1.a) Determination of the maximum useful load carrying capacity									
1.b) Determining the dimensions of the working space									
1.c) Determination of link lengths									
1.d) Motor and motor driver selection									
1.e) Power source selection									
2) Modeling of the Mechanical Parts of the Manipulator									
2.a) Creating three-dimensional solid model in computer environment									
2.b) Designing the sub-parts									
2.c) Determining the mounting strategy									
3) Electronic Circuit Design									
3.a) Determination of electronic components to be used									
3.b) Electronic circuit set up and tested in computer environment									
3.c) Determining the integration strategy of the electronic circuit									
3.d) Determining the wiring plan									
4) Procurement of Materials and Components									
4.a) Supply of materials required for the manufacture of mechanical parts									
4.b) Supplying electronic components									
4.c) Providing the mechanical parts required for integration									
5) Manufacturing of Mechanical Parts and Establishment of Electronic Circuit									
5.a) Manufacture of non-standard mechanical parts									
5.b) Assembling all parts of the electronic circuit									
6) Mathematical Modeling and Simulation of the System									
6.a) Defining system parameters									
6.b) Deriving the mathematical model									
6.c) Finding all link angles parametrically with inverse kinematic analysis									
6.d) Simulation of the system in computer environment									
7) Development of the Software and User Interface to Control the Manipulator									
7.a) Development of software that will enable the system to communicate within itself and send the relevant angle information to the motors									
7.b) Development of the user interface required for the user to easily control the manipulator.									
8) Installation and Testing of the System									
8.a) Assembling the mechanical parts									
8.b) Integration of the electronic circuit to the system									

8.c) Completing the cabling system									
8.d) Mechanical tests									
8.e) Electronics tests									
8.f) Software testing									
8.g) Carrying out the necessary optimization studies									
8.h) Precision calibration									

APPENDIX-2: BUDGET LIST

Product Name	Detail	Quantity	Unit Price ($)	Total Price ($)
Nema 23 Stepper Motor	1.2 Nm, 1000CPR Encoder	3	25	75
Nema 17 Geared Stepper Motor	0.52 Nm, 14:1 Gear Ratio, 1000CPR	3	61	183
Planetary Gearbox for Nema 23 Stepper Motor	50:1 Gear Ratio	3	64	192
Stepper Driver	Closed-Loop, 0-7 A, 24-50VDC	6	44	264
Motor and Encoder Cable	1.7m, AWG20	3	10	30
Motor and Encoder Cable	2.7m, AWG20	3	14	42
Metal Body Material	Mainly aluminum	1	200	200
Plastic Body Material	ABS	1	50	50
Bearing, Belt, Chain, Bolt and Other Mechanical Parts	Miscellaneous	1	275	275
Regulator, Sensor, Switch, Cable, and Other Electronics	Miscellaneous	1	200	200
Teensy 4.1 Development Board	32-bit 180MHz ARM Cortex-M4 CPU	1	50	50
Raspberry Pi 4 4GB Development Board	64-bit 1.5GHz Quad core Cortex-A72	1	90	90
Raspberry Pi 7" Touch Screen	800x480 pixels, 10-point touch screen	1	110	110
Machining Service	Manufacturing of the metal parts	1	500	500
24VDC Power Supply	600W, 25A output, 100-240VAC input	1	80	80
			TOTAL ($):	**2341**

APPENDIX-3: MATLAB CODES FOR FORWARD KINEMATICS

```
% MATLAB 2021a, Version 9.10.0.1602886

syms theta alpha length_a length_d t1 t2 t3 t4 t5 t6 L1 L2 L3 d1 d2 d3 d4 d5 d6 a1 a2 a3 a4 a5 a6 nx ox ax
ny oy ay nz oz az px py pz

% a1 = 0;
% a2 = 300;
% d1 = 299;
% d4 = 309;
% tooltip = 55.5;
% d6 = 35.5 + tooltip;

% t1=pi/2;
% t2=pi/4;
% t3=-pi/2;
% t4=pi/4;
% t5=pi/4;
% t6=pi/2;

% t1=pi/8;
% t2=pi/6;
% t3=0;
% t4=0;
% t5=0;
% t6=0;
%
% nx = 0;
% ox = 0;
% ax = 1;
%
% ny = 0;
% oy = -1;
% ay = 0;
%
% nz = 1;
% oz = 0;
% az = 0;

DH_table = [t1,         d1,    0,    pi/2;
      t2,         0,     a2,   0;
      t3+(pi/2),  0,     0,    pi/2;
      t4,         d4,    0,    -pi/2;
      t5,         0,     0,    pi/2;
      t6,         d6,    0,    0];
```

```
for i=0: 1 :5
   if  i==0
      theta = DH_table(i+1,1);
      length_d = DH_table(i+1,2);
      length_a = DH_table(i+1,3);
      alpha = DH_table(i+1,4);
      A1= [  cos(theta),    -sin(theta)*cos(alpha),    sin(theta)*sin(alpha),     length_a*cos(theta);
             sin(theta),    cos(theta)*cos(alpha),     -cos(theta)*sin(alpha),    length_a*sin(theta);
             0,             sin(alpha),                cos(alpha),                length_d;
             0,             0,                         0,                         1];
   elseif  i==1
      theta = DH_table(i+1,1);
      length_d = DH_table(i+1,2);
      length_a = DH_table(i+1,3);
      alpha = DH_table(i+1,4);
      A2= [  cos(theta),    -sin(theta)*cos(alpha),    sin(theta)*sin(alpha),     length_a*cos(theta);
             sin(theta),    cos(theta)*cos(alpha),     -cos(theta)*sin(alpha),    length_a*sin(theta);
             0,             sin(alpha),                cos(alpha),                length_d;
             0,             0,                         0,                         1];
   elseif  i==2
      theta = DH_table(i+1,1);
      length_d = DH_table(i+1,2);
      length_a = DH_table(i+1,3);
      alpha = DH_table(i+1,4);
      A3= [  cos(theta),    -sin(theta)*cos(alpha),    sin(theta)*sin(alpha),     length_a*cos(theta);
             sin(theta),    cos(theta)*cos(alpha),     -cos(theta)*sin(alpha),    length_a*sin(theta);
             0,             sin(alpha),                cos(alpha),                length_d;
             0,             0,                         0,                         1];
   elseif  i==3
      theta = DH_table(i+1,1);
      length_d = DH_table(i+1,2);
      length_a = DH_table(i+1,3);
      alpha = DH_table(i+1,4);
      A4= [  cos(theta),    -sin(theta)*cos(alpha),    sin(theta)*sin(alpha),     length_a*cos(theta);
             sin(theta),    cos(theta)*cos(alpha),     -cos(theta)*sin(alpha),    length_a*sin(theta);
             0,             sin(alpha),                cos(alpha),                length_d;
             0,             0,                         0,                         1];
   elseif  i==4
      theta = DH_table(i+1,1);
      length_d = DH_table(i+1,2);
      length_a = DH_table(i+1,3);
      alpha = DH_table(i+1,4);
      A5= [  cos(theta),    -sin(theta)*cos(alpha),    sin(theta)*sin(alpha),     length_a*cos(theta);
             sin(theta),    cos(theta)*cos(alpha),     -cos(theta)*sin(alpha),    length_a*sin(theta);
             0,             sin(alpha),                cos(alpha),                length_d;
             0,             0,                         0,                         1];
   elseif  i==5
```

```
        theta = DH_table(i+1,1);
        length_d = DH_table(i+1,2);
        length_a = DH_table(i+1,3);
        alpha = DH_table(i+1,4);
        A6= [  cos(theta),   -sin(theta)*cos(alpha),   sin(theta)*sin(alpha),   length_a*cos(theta);
               sin(theta),   cos(theta)*cos(alpha),   -cos(theta)*sin(alpha),   length_a*sin(theta);
               0,           sin(alpha),              cos(alpha),              length_d;
               0,           0,                       0,                       1];
    end

end

T06 = [ nx, ox, ax, px;
        ny, oy, ay, py;
        nz, oz, az, pz;
         0,  0, 0,  1];

T06_A0 = A1 * A2 * A3 * A4 * A5 * A6;
T06_A1 = A2 * A3 * A4 * A5 * A6;
T06_A2 = A3 * A4 * A5 * A6;
T06_A3 = A4 * A5 * A6;
T06_A4 = A5 * A6;
T06_A5 = A6;

t60 = simplify(T06_A0);
t61 = simplify(T06_A1);
t62 = simplify(T06_A2);
t63 = simplify(T06_A3);
t64 = simplify(T06_A4);
t65 = simplify(T06_A5);

iAT6_1 = inv(A1)*T06;
iAT6_2 = inv(A1*A2)*T06;
iAT6_3 = inv(A1*A2*A3)*T06;
iAT6_4 = inv(A1*A2*A3*A4)*T06;
iAT6_5 = inv(A1*A2*A3*A4*A5)*T06;

i61 = simplify(iAT6_1);
i62 = simplify(iAT6_2);
i63 = simplify(iAT6_3);
i64 = simplify(iAT6_4);
i65 = simplify(iAT6_5);

TA1 = A1;
TA2 = A1*A2;
TA3 = A1*A2*A3;
TA4 = A1*A2*A3*A4;
```

TA5 = A1*A2*A3*A4*A5;
TA6 = A1*A2*A3*A4*A5*A6;

APPENDIX-4: MATLAB CODES FOR INVERSE KINEMATICS

```
% MATLAB 2021a, Version 9.10.0.1602886

function [t1_deg, t2_deg, t3_deg, t4_deg, t5_deg ,t6_deg]= MCT_Inverse_Kinematics_Function(x1, y1, z1, orientation, x2, y2, z2)
        a1 = 0;
        a2 = 300;
        d1 = 299;
        d4 = 309;
        d6 = 35.5;
        tooltip = 55.5;

        t1 = 0;
        t2 = 0;
        t3 = 0;
        t4 = 0;
        t5 = 0;
        t6 = 0;

        t1_deg = 0;
        t2_deg = 0;
        t3_deg = 0;
        t4_deg = 0;
        t5_deg = 0;
        t6_deg = 0;

        px = x1;
        py = y1;
        pz = z1;

        ph = sqrt( (px)^2 + (py)^2 );

        if orientation == 1 % Down
           pz = pz + d6 + tooltip;

           nx = 1;
           ox = 0;
           ax = 0;

           ny = 0;
           oy = -1;
           ay = 0;

           nz = 0;
           oz = 0;
           az = -1;
```

```
elseif orientation == 2 % Up
    pz = pz - d6 - tooltip;

    nx = 1;
    ox = 0;
    ax = 0 ;

    ny = 0;
    oy = 1;
    ay = 0;

    nz = 0;
    oz = 0;
    az = 1;

elseif orientation == 3 % Normal to YZ
    ph = ph - d6 - tooltip;

    nx = 0;
    ox = 0;
    ax = 1;

    ny = 0;
    oy = -1;
    ay = 0;

    nz = 1;
    oz = 0;
    az = 0;
elseif orientation == 4 %Normal to XZ
    ph = ph - d6 - tooltip;

    nx = 1;
    ox = 0;
    ax = 0 ;

    ny = 0;
    oy = 0;
    ay = -1;

    nz = 0;
    oz = 1;
    az = 0;
else
    pz = pz - d6 - tooltip;
```

```
    nx = 1;
    ox = 0;
    ax = 0 ;

    ny = 0;
    oy = 1;
    ay = 0;

    nz = 0;
    oz = 0;
    az = -1;
end

if pz >= d1

    pv = pz - d1;
    R = sqrt(ph^2 + pv^2);
    P1 = atan(pv / ph);
    P2 = acos((d4^2 - a2^2 - R^2) / (-2 * a2 * R));
    P3 = acos((R^2 - d4^2 - a2^2) / (-2 * d4 * a2));

    t1 = atan2(py , px);
    t1_deg = t1 * 180 / pi;

    t2 = P1 + P2;
    t2_deg = t2 * 180 / pi;

    t3 = pi + P3;
    t3_deg = t3 * 180 / pi;

elseif pz < d1

    pv = d1 - pz;
    R = sqrt(ph^2 + pv^2);
    P1 = atan(pv / ph);
    P2 = acos((d4^2 - a2^2 - R^2) / (-2 * a2 * R));
    P3 = acos((R^2 - d4^2 - a2^2) / (-2 * d4 * a2));

    t1 = atan2(py , px);
    t1_deg = t1 * 180 / pi;

    t2 = P2 - P1;
    t2_deg = t2 * 180 / pi;

    t3 = pi + P3;
    t3_deg = t3 * 180 / pi;
end
```

```
    t4_o1 = ax*sin(t1) - ay*cos(t1);
    t4_o2 = az*cos(t2)*cos(t3) - az*sin(t2)*sin(t3) - ax*cos(t1)*cos(t2)*sin(t3) -
    ax*cos(t1)*cos(t3)*sin(t2) - ay*cos(t2)*sin(t1)*sin(t3) - ay*cos(t3)*sin(t1)*sin(t2);
    t4 = atan2(t4_o1 , t4_o2);
    t4_deg = t4 * 180 / pi;

    t5_o11 = ax*sin(t1)*sin(t4) - ay*cos(t1)*sin(t4) + az*cos(t2)*cos(t3)*cos(t4) -
    az*cos(t4)*sin(t2)*sin(t3) - ax*cos(t1)*cos(t2)*cos(t4)*sin(t3) -
    ax*cos(t1)*cos(t3)*cos(t4)*sin(t2) - ay*cos(t2)*cos(t4)*sin(t1)*sin(t3) -
    ay*cos(t3)*cos(t4)*sin(t1)*sin(t2);
    t5_o22 = ax*cos(t1)*sin(t2)*sin(t3) - az*cos(t3)*sin(t2) - ax*cos(t1)*cos(t2)*cos(t3) -
    ay*cos(t2)*cos(t3)*sin(t1) - az*cos(t2)*sin(t3) + ay*sin(t1)*sin(t2)*sin(t3);
    t5 = atan2(t5_o11 , -t5_o22);
    t5_deg = t5 * 180 / pi;

    t6_o1 = nx*cos(t4)*sin(t1) - ny*cos(t1)*cos(t4) - nz*cos(t2)*cos(t3)*sin(t4) +
    nz*sin(t2)*sin(t3)*sin(t4) + nx*cos(t1)*cos(t2)*sin(t3)*sin(t4) +
    nx*cos(t1)*cos(t3)*sin(t2)*sin(t4) + ny*cos(t2)*sin(t1)*sin(t3)*sin(t4) +
    ny*cos(t3)*sin(t1)*sin(t2)*sin(t4);
    t6_o2 = ox*cos(t4)*sin(t1) - oy*cos(t1)*cos(t4) - oz*cos(t2)*cos(t3)*sin(t4) +
    oz*sin(t2)*sin(t3)*sin(t4) + ox*cos(t1)*cos(t2)*sin(t3)*sin(t4) +
    ox*cos(t1)*cos(t3)*sin(t2)*sin(t4) + oy*cos(t2)*sin(t1)*sin(t3)*sin(t4) +
    oy*cos(t3)*sin(t1)*sin(t2)*sin(t4);
    t6 = atan2(t6_o1 , t6_o2);
    t6_deg = t6 * 180 / pi;
end
```

APPENDIX-5: MATLAB CODES FOR USER INTERFACE

```
% MATLAB 2021a, Version 9.10.0.1602886

properties (Access = public)
    workmode = "1"
    orientation1
    orientation2
    t1_series
    t2_series
    t3_series
    t4_series
    t5_series
    t6_series
    res
end
function startupFcn(app)
    movegui(app.UIFigure,'center');
    app.Px1EditField.Value = str2double("400");
    app.Py1EditField.Value = str2double("0");
    app.Pz1EditField.Value = str2double("599");
    app.Px2EditField.Value = str2double("400");
    app.Py2EditField.Value = str2double("0");
    app.Pz2EditField.Value = str2double("599");
    app.OrientationDropDown.Value = "3";
    app.orientation1 = 3;
    app.workmode = "Single";
    sim("Assembly_010_GUI.slx");
end
function UIFigureCloseRequest(app, event)
    set_param([bdroot],'SimulationCommand','stop')
    delete(app)
end
function StartStopSimulationSwitchValueChanged(app, event)
    value = app.StartStopSimulationSwitch.Value;
    if value == "On"
        sim("Assembly_010_GUI.slx");
    elseif value == "Off"
        set_param([bdroot],'SimulationCommand','stop')
    end
end
function WorkModeButtonGroupSelectionChanged(app, event)
    selectedButton = app.WorkModeButtonGroup.SelectedObject;
    if selectedButton == app.SinglePointButton
        app.workmode = "Single";
    elseif selectedButton == app.PointAtoBButton
        app.workmode = "AtoB";
```

```
        end
    end
    function OrientationDropDownValueChanged(app, event)
        value = app.OrientationDropDown.Value;
        if value == "1"
            app.orientation1 = 1;
        elseif value == "2"
            app.orientation1 = 2;
        elseif value == "3"
            app.orientation1 = 3;
        elseif value == "4"
            app.orientation1 = 4;
        elseif value == "5"
            app.orientation1 = 5;
        end
    end
    function OrientationDropDown_2ValueChanged(app, event)
        value = app.OrientationDropDown_2.Value;
        if value == "1"
            app.orientation2 = 1;
        elseif value == "2"
            app.orientation2 = 2;
        elseif value == "3"
            app.orientation2 = 3;
        elseif value == "4"
            app.orientation2 = 4;
        elseif value == "5"
            app.orientation2 = 5;
        end
    end
    function CalculateValuesButtonPushed(app, event)
        a1 = 0;
        a2 = 300;
        d1 = 299;
        d4 = 309;
        d6 = 35.5;
        tooltip = 55.5;
        x1 = app.Px1EditField.Value;
        x2 = app.Px2EditField.Value;
        y1 = app.Py1EditField.Value;
        y2 = app.Py2EditField.Value;
        z1 = app.Pz1EditField.Value;
        z2 = app.Pz2EditField.Value;
        if app.workmode == "Single"
            px = x1;
            py = y1;
            pz = z1;
```

```
ph = sqrt( (px)^2 + (py)^2 );

if app.orientation1 == 1 % Down
pz = pz + d6 + tooltip;
nx = 1;
ox = 0;
ax = 0;
ny = 0;
oy = 1;
ay = 0;
nz = 0;
oz = 0;
az = -1;
elseif app.orientation1 == 2 % Up
pz = pz - d6 - tooltip;
nx = 1;
ox = 0;
ax = 0 ;
ny = 0;
oy = 1;
ay = 0;
nz = 0;
oz = 0;
az = 1;
elseif app.orientation1 == 3 % Normal
ph = ph - d6 - tooltip;
nx = 0;
ox = 0;
ax = -1;
ny = 0;
oy = 1;
ay = 0;
nz = -1;
oz = 0;
az = 0;
else
pz = pz - d6 - tooltip;
nx = 1;
ox = 0;
ax = 0 ;
ny = 0;
oy = 1;
ay = 0;
nz = 0;
oz = 0;
az = -1;
end
```

```
if pz >= d1
pv = pz - d1;
R = sqrt(ph^2 + pv^2);
P1 = atan(pv / ph);
P2 = acos((d4^2 - a2^2 - R^2) / (-2 * a2 * R));
P3 = acos((R^2 - d4^2 - a2^2) / (-2 * d4 * a2));
t1 = atan2(py , px);
t1_deg = t1 * 180 / pi;
t2 = P1 + P2;
t2_deg = t2 * 180 / pi;
t3 = pi + P3;
t3_deg = t3 * 180 / pi;
elseif pz < d1
pv = d1 - pz;
R = sqrt(ph^2 + pv^2);
P1 = atan(pv / ph);
P2 = acos((d4^2 - a2^2 - R^2) / (-2 * a2 * R));
P3 = acos((R^2 - d4^2 - a2^2) / (-2 * d4 * a2));
t1 = atan2(py , px);
t1_deg = t1 * 180 / pi;
t2 = P2 - P1;
t2_deg = t2 * 180 / pi;
t3 = pi + P3;
t3_deg = t3 * 180 / pi;
end
t4_o1 = ax*sin(t1) - ay*cos(t1);
t4_o2 = az*cos(t2)*cos(t3) - az*sin(t2)*sin(t3) - ax*cos(t1)*cos(t2)*sin(t3) - ax*cos(t1)*cos(t3)*sin(t2) - ay*cos(t2)*sin(t1)*sin(t3) - ay*cos(t3)*sin(t1)*sin(t2);
t4 = atan2(t4_o1 , t4_o2);
t4_deg = t4 * 180 / pi;
t5_o11 = ax*sin(t1)*sin(t4) - ay*cos(t1)*sin(t4) + az*cos(t2)*cos(t3)*cos(t4) - az*cos(t4)*sin(t2)*sin(t3) - ax*cos(t1)*cos(t2)*cos(t4)*sin(t3) - ax*cos(t1)*cos(t3)*cos(t4)*sin(t2) - ay*cos(t2)*cos(t4)*sin(t1)*sin(t3) - ay*cos(t3)*cos(t4)*sin(t1)*sin(t2);
t5_o22 = ax*cos(t1)*sin(t2)*sin(t3) - az*cos(t3)*sin(t2) - ax*cos(t1)*cos(t2)*cos(t3) - ay*cos(t2)*cos(t3)*sin(t1) - az*cos(t2)*sin(t3) + ay*sin(t1)*sin(t2)*sin(t3);
t5 = atan2(t5_o11 , -t5_o22);
t5_deg = t5 * 180 / pi;
t6_o1 = nx*cos(t4)*sin(t1) - ny*cos(t1)*cos(t4) - nz*cos(t2)*cos(t3)*sin(t4) + nz*sin(t2)*sin(t3)*sin(t4) + nx*cos(t1)*cos(t2)*sin(t3)*sin(t4) + nx*cos(t1)*cos(t3)*sin(t2)*sin(t4) + ny*cos(t2)*sin(t1)*sin(t3)*sin(t4) + ny*cos(t3)*sin(t1)*sin(t2)*sin(t4);
t6_o2 = ox*cos(t4)*sin(t1) - oy*cos(t1)*cos(t4) - oz*cos(t2)*cos(t3)*sin(t4) + oz*sin(t2)*sin(t3)*sin(t4) + ox*cos(t1)*cos(t2)*sin(t3)*sin(t4) + ox*cos(t1)*cos(t3)*sin(t2)*sin(t4) + oy*cos(t2)*sin(t1)*sin(t3)*sin(t4) + oy*cos(t3)*sin(t1)*sin(t2)*sin(t4);
t6 = atan2(t6_o1 , t6_o2);
```

```
        t6_deg = t6 * 180 / pi;
        % app.UITable.Data.Clear();
        app.UITable.Data(1,1)="1";
        app.UITable.Data(1,2)=px;
        app.UITable.Data(1,3)=py;
        app.UITable.Data(1,4)=pz;
        app.UITable.Data(1,5)=app.orientation1;
        app.UITable.Data(1,6)=t1_deg;
        app.UITable.Data(1,7)=t2_deg;
        app.UITable.Data(1,8)=t3_deg;
        app.UITable.Data(1,9)=t4_deg;
        app.UITable.Data(1,10)=t5_deg;
        app.UITable.Data(1,11)=t6_deg;
    elseif app.workmode == "AtoB"
        app.res = 100;
        px_calc = linspace (x1,x2,app.res);
        py_calc = linspace (y1,y2,app.res);
        pz_calc = linspace (z1,z2,app.res);
        for i=1:app.res
        px = px_calc(i);
        py = py_calc(i);
        pz = pz_calc(i);
        ph = sqrt( (px)^2 + (py)^2 );

        if app.orientation1 == 1 % Down
        pz = pz + d6 + tooltip;
        nx = 1;
        ox = 0;
        ax = 0;
        ny = 0;
        oy = 1;
        ay = 0;
        nz = 0;
        oz = 0;
        az = -1;
        elseif app.orientation1 == 2 % Up
        pz = pz - d6 - tooltip;
        nx = 1;
        ox = 0;
        ax = 0 ;
        ny = 0;
        oy = 1;
        ay = 0;
        nz = 0;
        oz = 0;
        az = 1;
        elseif app.orientation1 == 3 % Normal to YZ
```

```
ph = ph - d6 - tooltip;
nx = 0;
ox = 0;
ax = -1;
ny = 0;
oy = 1;
ay = 0;
nz = -1;
oz = 0;
az = 0;
elseif app.orientation1 == 4 %Normal to XZ
    ph = ph - d6 - tooltip;

    nx = 1;
    ox = 0;
    ax = 0 ;

    ny = 0;
    oy = 0;
    ay = -1;

    nz = 0;
    oz = 1;
    az = 0;

else
pz = pz - d6 - tooltip;
nx = 1;
ox = 0;
ax = 0 ;
ny = 0;
oy = 1;
ay = 0;
nz = 0;
oz = 0;
az = -1;
end
if pz >= d1
pv = pz - d1;
R = sqrt(ph^2 + pv^2);
P1 = atan(pv / ph);
P2 = acos((d4^2 - a2^2 - R^2) / (-2 * a2 * R));
P3 = acos((R^2 - d4^2 - a2^2) / (-2 * d4 * a2));
t1 = atan2(py , px);
t1_deg = t1 * 180 / pi;
t2 = P1 + P2;
t2_deg = t2 * 180 / pi;
```

```
t3 = pi + P3;
t3_deg = t3 * 180 / pi;
elseif pz < d1
pv = d1 - pz;
R = sqrt(ph^2 + pv^2);
P1 = atan(pv / ph);
P2 = acos((d4^2 - a2^2 - R^2) / (-2 * a2 * R));
P3 = acos((R^2 - d4^2 - a2^2) / (-2 * d4 * a2));
t1 = atan2(py , px);
t1_deg = t1 * 180 / pi;
t2 = P2 - P1;
t2_deg = t2 * 180 / pi;
t3 = pi + P3;
t3_deg = t3 * 180 / pi;
end
t4_o1 = ax*sin(t1) - ay*cos(t1);
t4_o2 = az*cos(t2)*cos(t3) - az*sin(t2)*sin(t3) - ax*cos(t1)*cos(t2)*sin(t3) - ax*cos(t1)*cos(t3)*sin(t2) - ay*cos(t2)*sin(t1)*sin(t3) - ay*cos(t3)*sin(t1)*sin(t2);
t4 = atan2(t4_o1 , t4_o2);
t4_deg = t4 * 180 / pi;
t5_o11 = ax*sin(t1)*sin(t4) - ay*cos(t1)*sin(t4) + az*cos(t2)*cos(t3)*cos(t4) - az*cos(t4)*sin(t2)*sin(t3) - ax*cos(t1)*cos(t2)*cos(t4)*sin(t3) - ax*cos(t1)*cos(t3)*cos(t4)*sin(t2) - ay*cos(t2)*cos(t4)*sin(t1)*sin(t3) - ay*cos(t3)*cos(t4)*sin(t1)*sin(t2);
t5_o22 = ax*cos(t1)*sin(t2)*sin(t3) - az*cos(t3)*sin(t2) - ax*cos(t1)*cos(t2)*cos(t3) - ay*cos(t2)*cos(t3)*sin(t1) - az*cos(t2)*sin(t3) + ay*sin(t1)*sin(t2)*sin(t3);
t5 = atan2(t5_o11 , -t5_o22);
t5_deg = t5 * 180 / pi;
t6_o1 = nx*cos(t4)*sin(t1) - ny*cos(t1)*cos(t4) - nz*cos(t2)*cos(t3)*sin(t4) + nz*sin(t2)*sin(t3)*sin(t4) + nx*cos(t1)*cos(t2)*sin(t3)*sin(t4) + nx*cos(t1)*cos(t3)*sin(t2)*sin(t4) + ny*cos(t2)*sin(t1)*sin(t3)*sin(t4) + ny*cos(t3)*sin(t1)*sin(t2)*sin(t4);
t6_o2 = ox*cos(t4)*sin(t1) - oy*cos(t1)*cos(t4) - oz*cos(t2)*cos(t3)*sin(t4) + oz*sin(t2)*sin(t3)*sin(t4) + ox*cos(t1)*cos(t2)*sin(t3)*sin(t4) + ox*cos(t1)*cos(t3)*sin(t2)*sin(t4) + oy*cos(t2)*sin(t1)*sin(t3)*sin(t4) + oy*cos(t3)*sin(t1)*sin(t2)*sin(t4);
t6 = atan2(t6_o1 , t6_o2);
t6_deg = t6 * 180 / pi;
app.UITable.Data(i,1)=i;
app.UITable.Data(i,2)=px;
app.UITable.Data(i,3)=py;
app.UITable.Data(i,4)=pz;
app.UITable.Data(i,5)=app.orientation1;
app.UITable.Data(i,6)=t1_deg;
app.UITable.Data(i,7)=t2_deg;
app.UITable.Data(i,8)=t3_deg;
app.UITable.Data(i,9)=t4_deg;
```

```
                app.UITable.Data(i,10)=t5_deg;
                app.UITable.Data(i,11)=t6_deg;
                app.t1_series(i) = t1_deg;
                app.t2_series(i) = t2_deg;
                app.t3_series(i) = t3_deg;
                app.t4_series(i) = t4_deg;
                app.t5_series(i) = t5_deg;
                app.t6_series(i) = t6_deg;
                end
        end
end
function RunRobotButtonPushed(app, event)
        % % % % % % % % TO SEND RAW INPUT DATA INTO THE SIMULINK MODEL
        if app.workmode == "Single"
                set_param([bdroot,'/px1'],'Value', num2str(app.Px1EditField.Value));
                set_param([bdroot,'/py1'],'Value', num2str(app.Py1EditField.Value));
                set_param([bdroot,'/pz1'],'Value', num2str(app.Pz1EditField.Value));
                set_param([bdroot,'/orientation1'],'Value', num2str(app.orientation1));
        elseif app.workmode == "AtoB"
                set_param([bdroot,'/px1'],'Value', num2str(app.Px1EditField.Value));
                set_param([bdroot,'/py1'],'Value', num2str(app.Py1EditField.Value));
                set_param([bdroot,'/pz1'],'Value', num2str(app.Pz1EditField.Value));
                set_param([bdroot,'/orientation1'],'Value', num2str(app.orientation1));
                set_param([bdroot,'/px2'],'Value', num2str(app.Px2EditField.Value));
                set_param([bdroot,'/py2'],'Value', num2str(app.Py2EditField.Value));
                set_param([bdroot,'/pz2'],'Value', num2str(app.Pz2EditField.Value));
        end
        % % % % % % % %
        % % % % % % % % TO SEND CALCULATED ANGLES DIRECTLY INTO THE
SIMULINK MODEL
        % if app.workmode == "Single"
                % set_param([bdroot,'/t1_deg'],'Value', num2str(app.UITable.Data(1,6)));
                % set_param([bdroot,'/t2_deg'],'Value', num2str(app.UITable.Data(1,7)));
                % set_param([bdroot,'/t3_deg'],'Value', num2str(app.UITable.Data(1,8)));
                % set_param([bdroot,'/t4_deg'],'Value', num2str(app.UITable.Data(1,9)));
                % set_param([bdroot,'/t5_deg'],'Value', num2str(app.UITable.Data(1,10)));
                % set_param([bdroot,'/t6_deg'],'Value', num2str(app.UITable.Data(1,11)));
        % elseif app.workmode == "AtoB"
                % k=1;
                % set_param([bdroot,'/t1_deg'],'Value', num2str(app.t1_series(k)));
                % set_param([bdroot,'/t2_deg'],'Value', num2str(app.t2_series(k)));
                % set_param([bdroot,'/t3_deg'],'Value', num2str(app.t3_series(k)));
                % set_param([bdroot,'/t4_deg'],'Value', num2str(app.t4_series(k)));
                % set_param([bdroot,'/t5_deg'],'Value', num2str(app.t5_series(k)));
                % set_param([bdroot,'/t6_deg'],'Value', num2str(app.t6_series(k)));
                % while k <= app.res-1
                        % set_param([bdroot,'/t1_deg'],'Value', num2str(app.t1_series(k)));
```

```
                    % set_param([bdroot,'/t2_deg'],'Value', num2str(app.t2_series(k)));
                    % set_param([bdroot,'/t3_deg'],'Value', num2str(app.t3_series(k)));
                    % set_param([bdroot,'/t4_deg'],'Value', num2str(app.t4_series(k)));
                    % set_param([bdroot,'/t5_deg'],'Value', num2str(app.t5_series(k)));
                    % set_param([bdroot,'/t6_deg'],'Value', num2str(app.t6_series(k)));
                    % k = k+1;
                % end
            % end
            % % % % % % % %
        end
```

APPENDIX-6: MATLAB CODES FOR WORKSPACE ESTIMATION

```
% MATLAB 2021a, Version 9.10.0.1602886

% % % WORKSPACE ESTIMATION BY THE MONTE CARLO METHOD
% % % Robot Dimensions
a2 = 300;
d1 = 299;
d4 = 309;
tooltip = 55.5;
d6 = 35.5 + tooltip;
% % %

% % % Axis Limitations
t1_min = -170;
t1_max = +170;
t2_min = -60;
t2_max = +120;
t3_min = -120;
t3_max = +30;
t4_min = -355;
t4_max = +355;
t5_min = -130;
t5_max = +130;
t6_min = -355;
t6_max = +355;
% % %

steps = 10000; % Number of samples for the Monte Carlo method

for i = 1 : steps
% % % Formulation for the Monte Carlo Method
   t1 = (t1_max-t1_min)*rand() + t1_min;
   t2 = (t2_max-t2_min)*rand() + t2_min;
   t3 = (t3_max-t3_min)*rand() + t3_min;
   t4 = (t4_max-t4_min)*rand() + t4_min;
   t5 = (t5_max-t5_min)*rand() + t5_min;
   t6 = (t6_max-t6_min)*rand() + t6_min;
% % %

% % % X, Y and Z positions of the TCP
   px = d4*cosd(t2 + t3)*cosd(t1) + a2*cosd(t1)*cosd(t2) + d6*cosd(t2 + t3)*cosd(t1)*cosd(t5) +
d6*sind(t1)*sind(t4)*sind(t5) - d6*cosd(t1)*cosd(t2)*cosd(t4)*sind(t3)*sind(t5) -
d6*cosd(t1)*cosd(t3)*cosd(t4)*sind(t2)*sind(t5);
   py = d4*(cosd(t2)*sind(t1)*sind(t3 + pi/2) + cosd(t3 + pi/2)*sind(t1)*sind(t2)) -
d6*(sind(t5)*(cosd(t1)*sind(t4) - cosd(t4)*(cosd(t2)*cosd(t3 + pi/2)*sind(t1) - sind(t1)*sind(t2)*sind(t3 +
```

```
pi/2))) - cosd(t5)*(cosd(t2)*sind(t1)*sind(t3 + pi/2) + cosd(t3 + pi/2)*sind(t1)*sind(t2))) +
a2*cosd(t2)*sind(t1);
   pz = d1 + a2*sind(t2) + d4*cosd(t2)*sind(t3) + d4*cosd(t3)*sind(t2) + d6*cosd(t2)*cosd(t5)*sind(t3) +
d6*cosd(t3)*cosd(t5)*sind(t2) + d6*cosd(t2)*cosd(t3)*cosd(t4)*sind(t5) -
d6*cosd(t4)*sind(t2)*sind(t3)*sind(t5);
%%%

%%% Adding the angles and positions for the current sample to an array
   arr_x(i) = px;
   arr_y(i) = py;
   arr_z(i) = pz;
   arr_t1(i) = t1;
   arr_t2(i) = t2;
   arr_t3(i) = t3;
   arr_t4(i) = t4;
   arr_t5(i) = t5;
   arr_t6(i) = t6;
%%%
end

%%% Plotting the of results in 3D
axis equal
grid on
hold on
plot3(arr_x, arr_y, arr_z, 'r.')
view(3)
xlabel('X (mm)', 'FontSize', 20);
ylabel('Y (mm)', 'FontSize', 20);
zlabel('Z (mm)', 'FontSize', 20);
%%%
```

Appendix-7: MATLAB Codes to Simulate Continuous Motion and to Generate Graphs via Timeseries

```
% MATLAB 2021a, Version 9.10.0.1602886

a1 = 0;
a2 = 300;
d1 = 299;
d4 = 309;
d6 = 35.5;
tooltip = 55.5;

res = 100;

orientation = 3;

% % % FOR LINEAR MOTION
x1 = 400;
x2 = 400;
y1 = 0;
y2 = 0;
z1 = 550;
z2 = -100;
px_calc = linspace (x1,x2,res);
py_calc = linspace (y1,y2,res);
pz_calc = linspace (z1,z2,res);
step = linspace(1,res);
% % % %

% % % FOR CIRCULAR MOTION
% x1 = 250;
% y1 = 250;
% z1 = 650;
% r1 = 150;
% theta_circle = linspace (0, 2*pi, res);
%
% px_calc = -r1 * cos(theta_circle) + x1;
% py_calc = r1 * sin(theta_circle) + y1;
% pz_calc = linspace (z1,z1,res);
%
% % px_calc = linspace (x1,x1,res);
% % py_calc = r1 * cos(theta_circle) + y1;
% % pz_calc = r1 * sin(theta_circle) + z1;
%
% % py_calc = r1 - r1 * cos(theta_circle);
% % pz_calc = r1 - r1 * sin(theta_circle);
%
```

```
% step = linspace(1,res);
% % %

% % % FOR SINUSOIDAL MOTION
% x1 = -200;
% x2 = 200;
% y1 = -400;
% z1 = 400;
% r1 = 100;
% theta_circle = linspace (0, 4*pi, res);
%
% px_calc = linspace (x1,x2,res);
% py_calc = linspace (y1,y1,res);
% pz_calc = r1 * sin(theta_circle) + z1;
%
% step = linspace(1,res);
% % %

t1_array = zeros(res,1);
t2_array = zeros(res,1);
t3_array = zeros(res,1);
t4_array = zeros(res,1);
t5_array = zeros(res,1);
t6_array = zeros(res,1);

for i=1:res
    px = px_calc(i);
    py = py_calc(i);
    pz = pz_calc(i);

    ph = sqrt( (px)^2 + (py)^2 );

    if orientation == 1 % Down
        pz = pz + d6 + tooltip;

        nx = 1;
        ox = 0;
        ax = 0;

        ny = 0;
        oy = -1;
        ay = 0;

        nz = 0;
        oz = 0;
        az = -1;
```

```
elseif orientation == 2 % Up
    pz = pz - d6 - tooltip;

    nx = 1;
    ox = 0;
    ax = 0 ;

    ny = 0;
    oy = 1;
    ay = 0;

    nz = 0;
    oz = 0;
    az = 1;

elseif orientation == 3 % Normal
    ph = ph - d6 - tooltip;

    nx = 0;
    ox = 0;
    ax = 1;

    ny = 0;
    oy = -1;
    ay = 0;

    nz = 1;
    oz = 0;
    az = 0;
elseif orientation == 4 %Normal to XZ
    ph = ph - d6 - tooltip;

    nx = 1;
    ox = 0;
    ax = 0 ;

    ny = 0;
    oy = 0;
    ay = -1;

    nz = 0;
    oz = 1;
    az = 0;
else
    nx = 1;
    ox = 0;
    ax = 0 ;
```

```
    ny = 0;
    oy = 0;
    ay = -1;

    nz = 0;
    oz = 1;
    az = 0;
end

if pz >= d1

    pv = pz - d1;
    R = sqrt(ph^2 + pv^2);
    P1 = atan(pv / ph);
    P2 = acos((d4^2 - a2^2 - R^2) / (-2 * a2 * R));
    P3 = acos((R^2 - d4^2 - a2^2) / (-2 * d4 * a2));

    t1 = atan2(py , px);
    t1_deg = t1 * 180 / pi;

    t2 = P1 + P2;
    t2_deg = t2 * 180 / pi;

    t3 = pi + P3;
    t3_deg = t3 * 180 / pi;

elseif pz < d1

    pv = d1 - pz;
    R = sqrt(ph^2 + pv^2);
    P1 = atan(pv / ph);
    P2 = acos((d4^2 - a2^2 - R^2) / (-2 * a2 * R));
    P3 = acos((R^2 - d4^2 - a2^2) / (-2 * d4 * a2));

    t1 = atan2(py , px);
    t1_deg = t1 * 180 / pi;

    t2 = P2 - P1;
    t2_deg = t2 * 180 / pi;

    t3 = pi + P3;
    t3_deg = t3 * 180 / pi;
end

t4_o1 = ax*sin(t1) - ay*cos(t1);
```

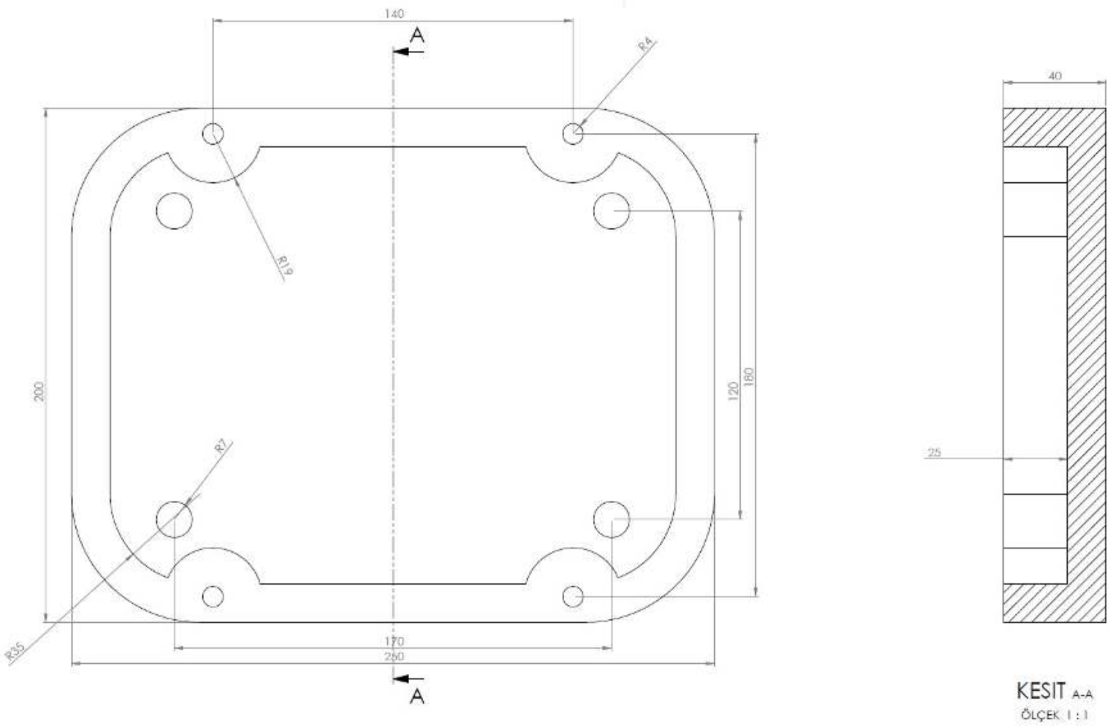
```
    t4_o2 = az*cos(t2)*cos(t3) - az*sin(t2)*sin(t3) - ax*cos(t1)*cos(t2)*sin(t3) - ax*cos(t1)*cos(t3)*sin(t2) -
ay*cos(t2)*sin(t1)*sin(t3) - ay*cos(t3)*sin(t1)*sin(t2);
    t4 = atan2(t4_o1 , t4_o2);
    t4_deg = t4 * 180 / pi;

    t5_o11 = ax*sin(t1)*sin(t4) - ay*cos(t1)*sin(t4) + az*cos(t2)*cos(t3)*cos(t4) - az*cos(t4)*sin(t2)*sin(t3)
- ax*cos(t1)*cos(t2)*cos(t4)*sin(t3) - ax*cos(t1)*cos(t3)*cos(t4)*sin(t2) -
ay*cos(t2)*cos(t4)*sin(t1)*sin(t3) - ay*cos(t3)*cos(t4)*sin(t1)*sin(t2);
    t5_o22 = ax*cos(t1)*sin(t2)*sin(t3) - az*cos(t3)*sin(t2) - ax*cos(t1)*cos(t2)*cos(t3) -
ay*cos(t2)*cos(t3)*sin(t1) - az*cos(t2)*sin(t3) + ay*sin(t1)*sin(t2)*sin(t3);
    t5 = atan2(t5_o11 , -t5_o22);
    t5_deg = t5 * 180 / pi;

    t6_o1 = nx*cos(t4)*sin(t1) - ny*cos(t1)*cos(t4) - nz*cos(t2)*cos(t3)*sin(t4) + nz*sin(t2)*sin(t3)*sin(t4)
+ nx*cos(t1)*cos(t2)*sin(t3)*sin(t4) + nx*cos(t1)*cos(t3)*sin(t2)*sin(t4) +
ny*cos(t2)*sin(t1)*sin(t3)*sin(t4) + ny*cos(t3)*sin(t1)*sin(t2)*sin(t4);
    t6_o2 = ox*cos(t4)*sin(t1) - oy*cos(t1)*cos(t4) - oz*cos(t2)*cos(t3)*sin(t4) + oz*sin(t2)*sin(t3)*sin(t4)
+ ox*cos(t1)*cos(t2)*sin(t3)*sin(t4) + ox*cos(t1)*cos(t3)*sin(t2)*sin(t4) +
oy*cos(t2)*sin(t1)*sin(t3)*sin(t4) + oy*cos(t3)*sin(t1)*sin(t2)*sin(t4);
    t6 = atan2(t6_o1 , t6_o2);
    t6_deg = t6 * 180 / pi;

    t1_array(i) = t1_deg;
    t2_array(i) = t2_deg;
    t3_array(i) = t3_deg;
    t4_array(i) = t4_deg;
    t5_array(i) = t5_deg;
    t6_array(i) = t6_deg;

end
time = (0 : res/(res-1) : res);
t1_series = timeseries(t1_array,time);
t2_series = timeseries(t2_array,time);
t3_series = timeseries(t3_array,time);
t4_series = timeseries(t4_array,time);
t5_series = timeseries(t5_array,time);
t6_series = timeseries(t6_array,time);


figure
subplot(3,3,1)
plot(py_calc , pz_calc , 'k.')
xlabel('Position on the Y Axis')
ylabel('Position on the Z Axis')
title('Overall Path')
axis equal
grid on
```

```
% % % To Draw a 3D Graph for Interplanar Motion
% subplot(3,3,1)
% plot3(px_calc , py_calc , pz_calc , 'k.');
% xlabel('X (mm)', 'FontSize', 14);
% ylabel('Y (mm)', 'FontSize', 14);
% zlabel('Z (mm)', 'FontSize', 14);
% title('Overall Path')
% axis equal
% grid on
% % %

subplot(3,3,2)
plot(step , py_calc , 'k.')
xlabel('Waypoint Along the Path')
ylabel('Position on the Y Axis (mm)')
title('Position Change Along the Y Axis')
grid on

subplot(3,3,3)
plot(step , pz_calc , 'k.')
xlabel('Waypoint Along the Path')
ylabel('Position on the Z Axis (mm)')
title('Position Change Along the Z Axis')
grid on

subplot(3,3,4)
plot(step , t1_array)
xlabel('Waypoint Along the Path')
ylabel('Angle Theta 1 (deg)')
title('Angular Position for Axis 1')
grid on

subplot(3,3,5)
plot(step , t2_array)
xlabel('Waypoint Along the Path')
ylabel('Angle Theta 2 (deg)')
title('Angular Position for Axis 2')
grid on

subplot(3,3,6)
plot(step , t3_array)
xlabel('Waypoint Along the Path')
ylabel('Angle Theta 3 (deg)')
title('Angular Position for Axis 3')
grid on
```

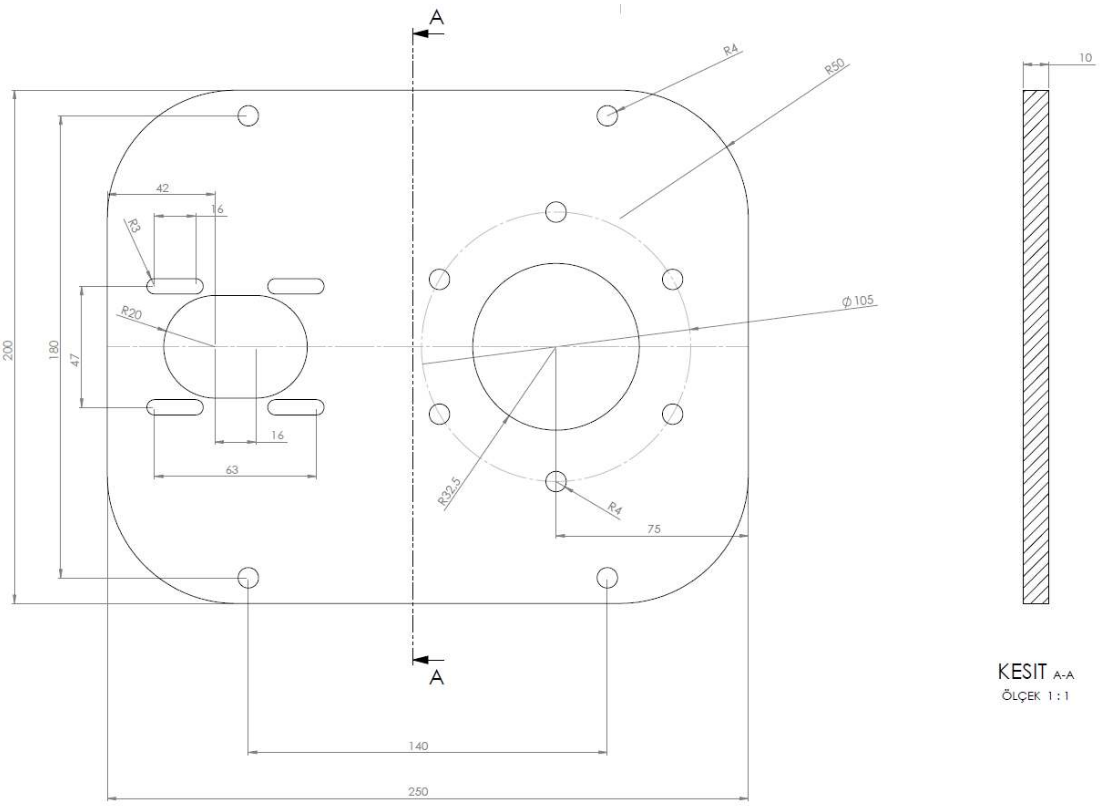
```
subplot(3,3,7)
plot(step , t4_array)
xlabel('Waypoint Along the Path')
ylabel('Angle Theta 4 (deg)')
title('Angular Position for Axis 4')
grid on

subplot(3,3,8)
plot(step , t5_array)
xlabel('Waypoint Along the Path')
ylabel('Angle Theta 5 (deg)')
title('Angular Position for Axis 5')
grid on

subplot(3,3,9)
plot(step , t6_array)
xlabel('Waypoint Along the Path')
ylabel('Angle Theta 6 (deg)')
title('Angular Position for Axis 6')
grid on
```

APPENDIX-8: TECHNICAL DRAWINGS OF THE DESIGNED ROBOT

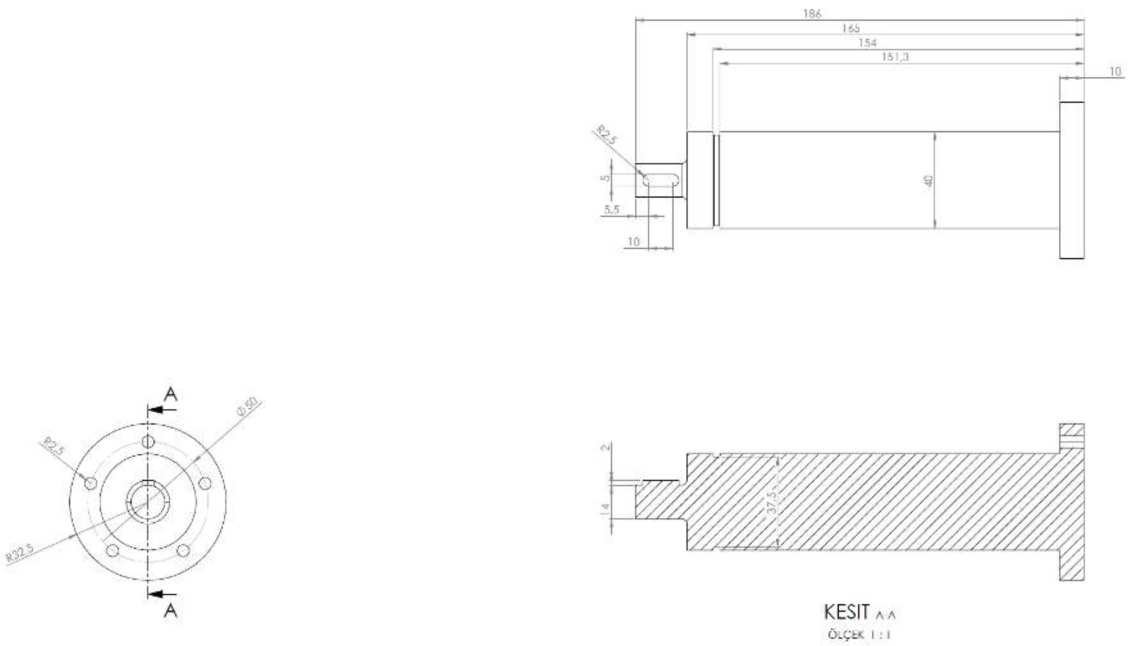
Part_010_Pedestal



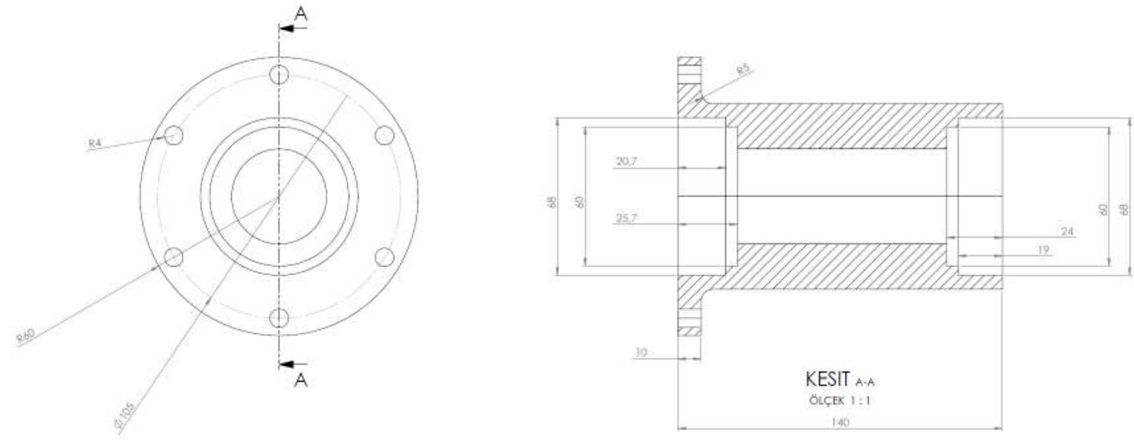
Part_020_Pedestal_Top_Plate



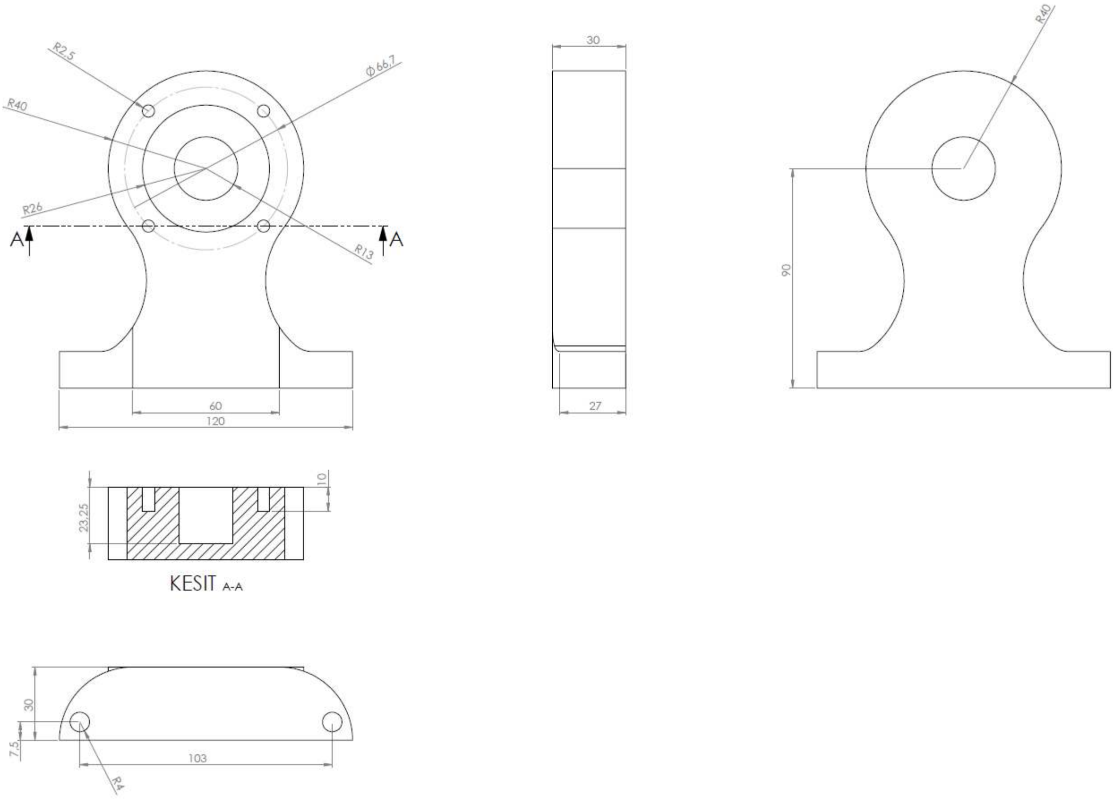
Part_030_A1_Turret



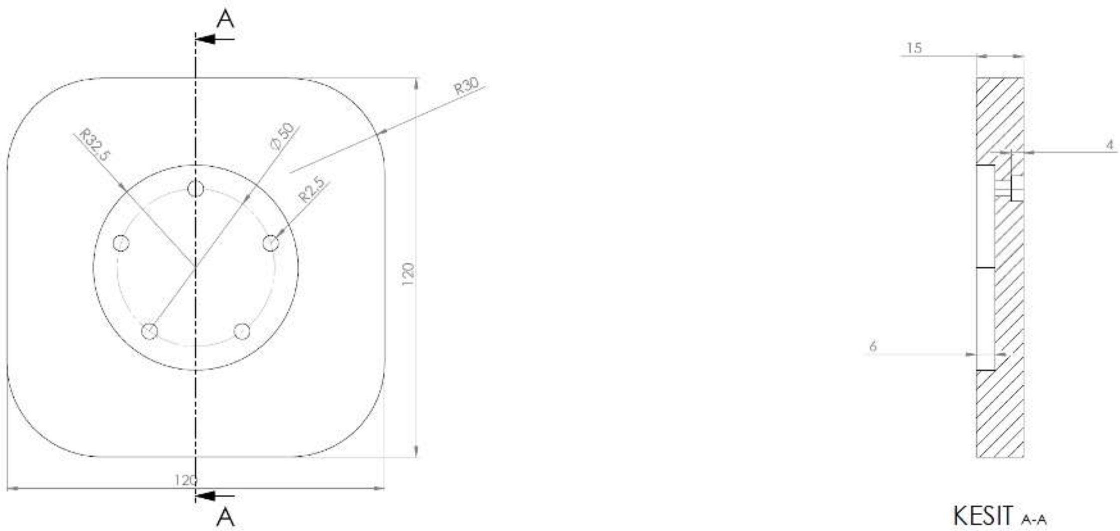
Part_040_A1_Spindle



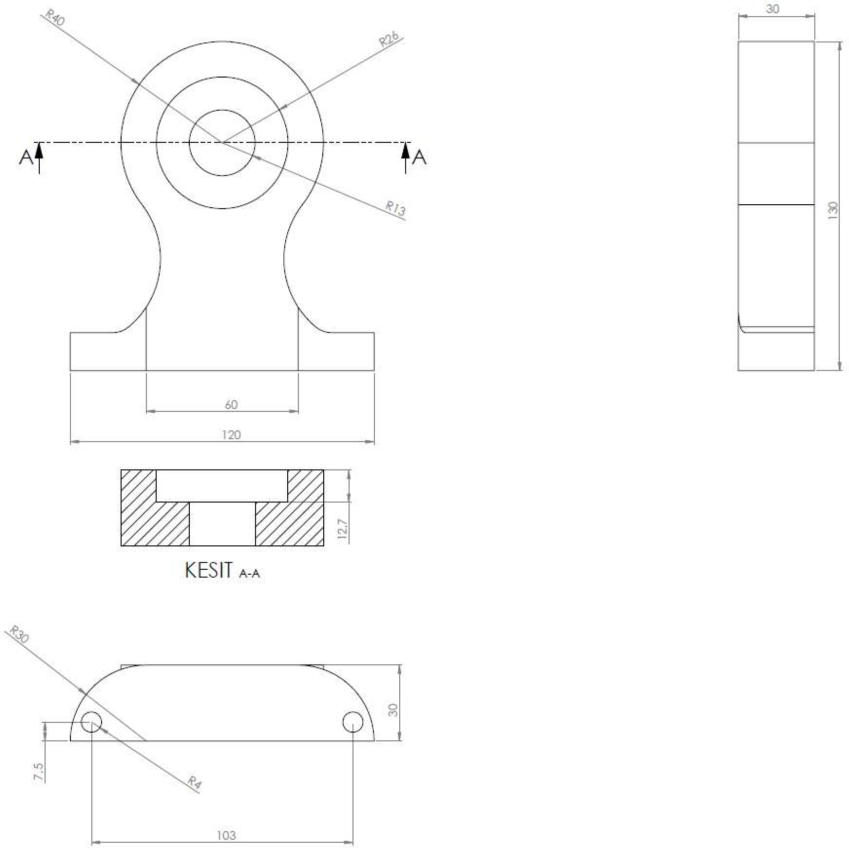
Part_050_A1_Turret_Platform



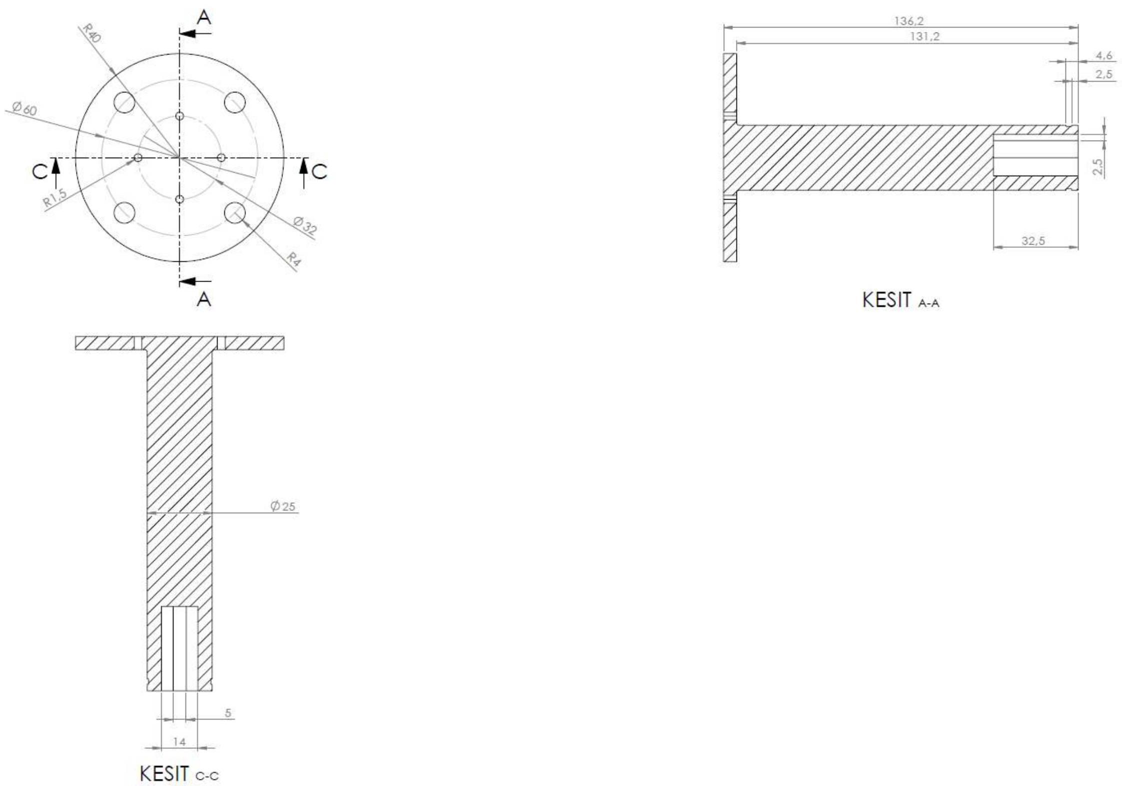
Part_060_A2_Support_Left



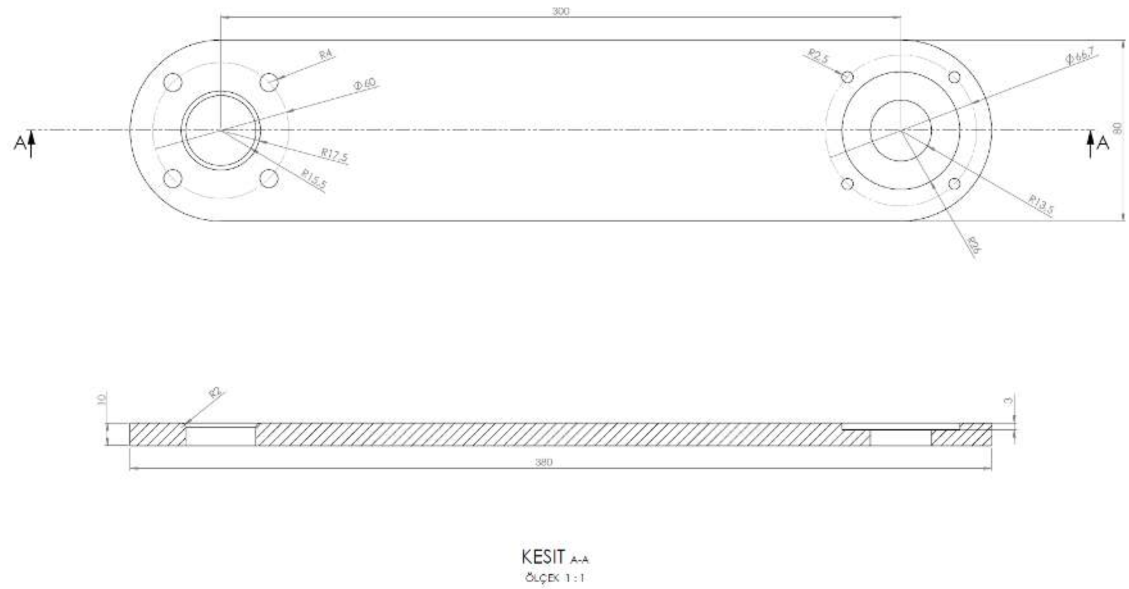
Part_070_A2_Support_Right



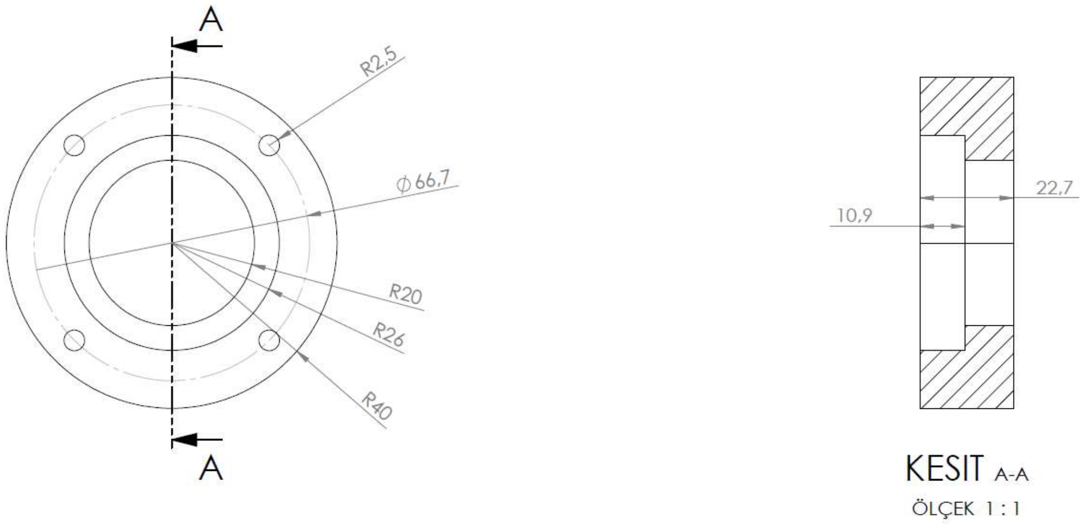
Part_080_A2_Spindle



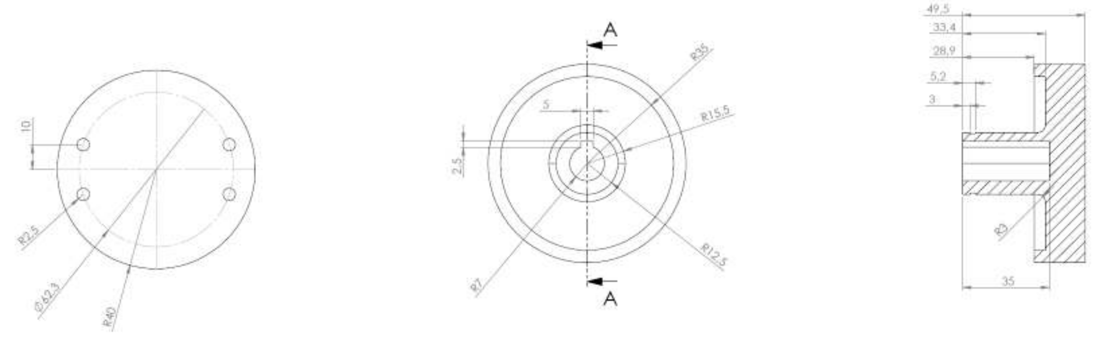
Part_090_Link_2_Body



Part_100_A3_Elbow_Support

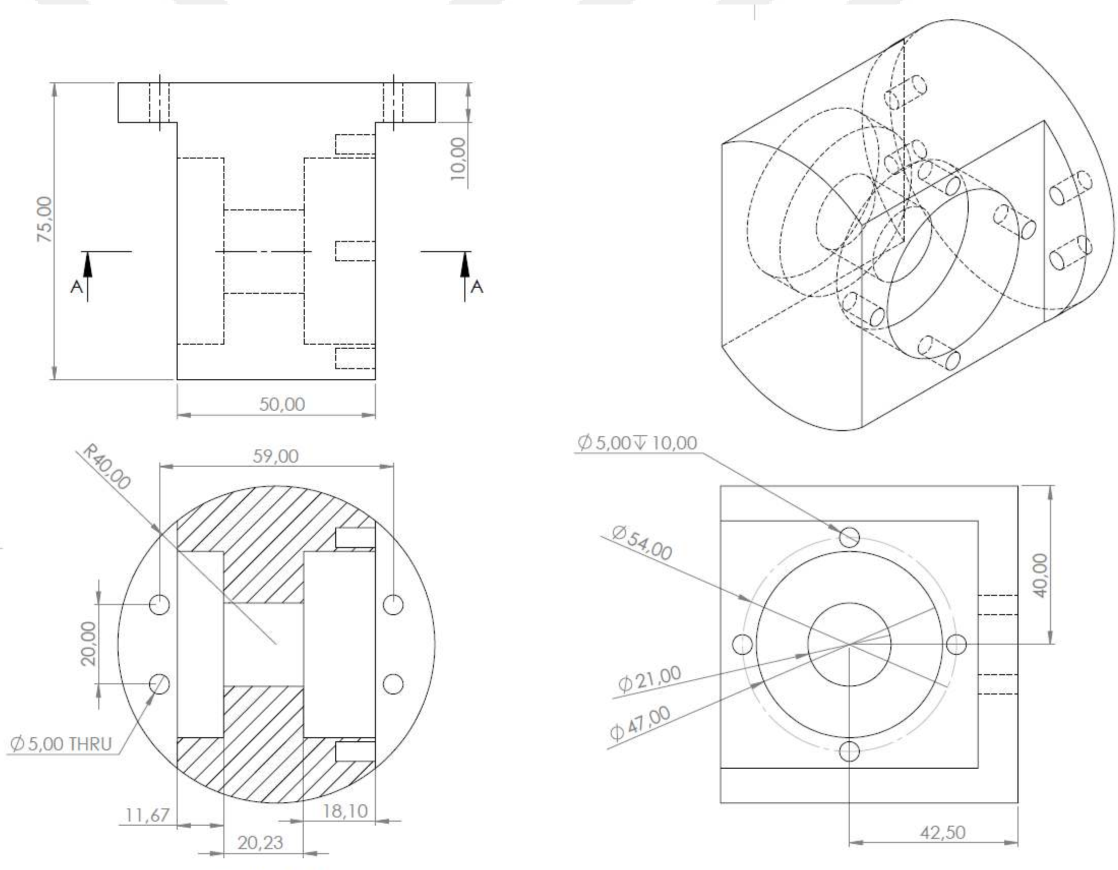


Part_110_A3_Spindle



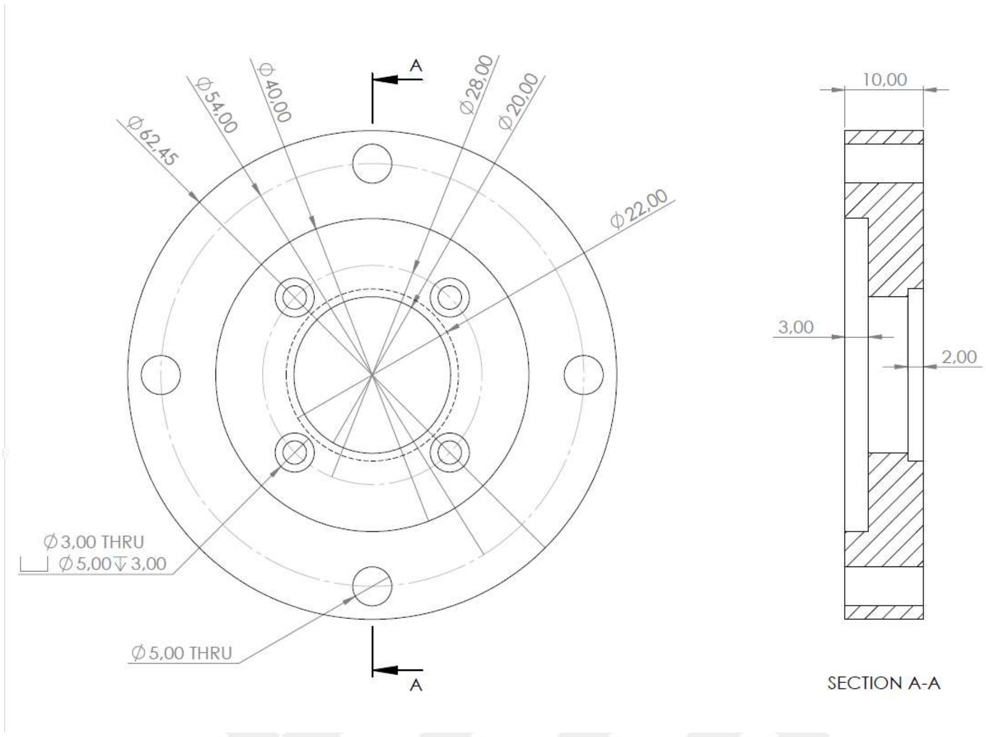
KESIT A-A

Part_120_A4_Housing

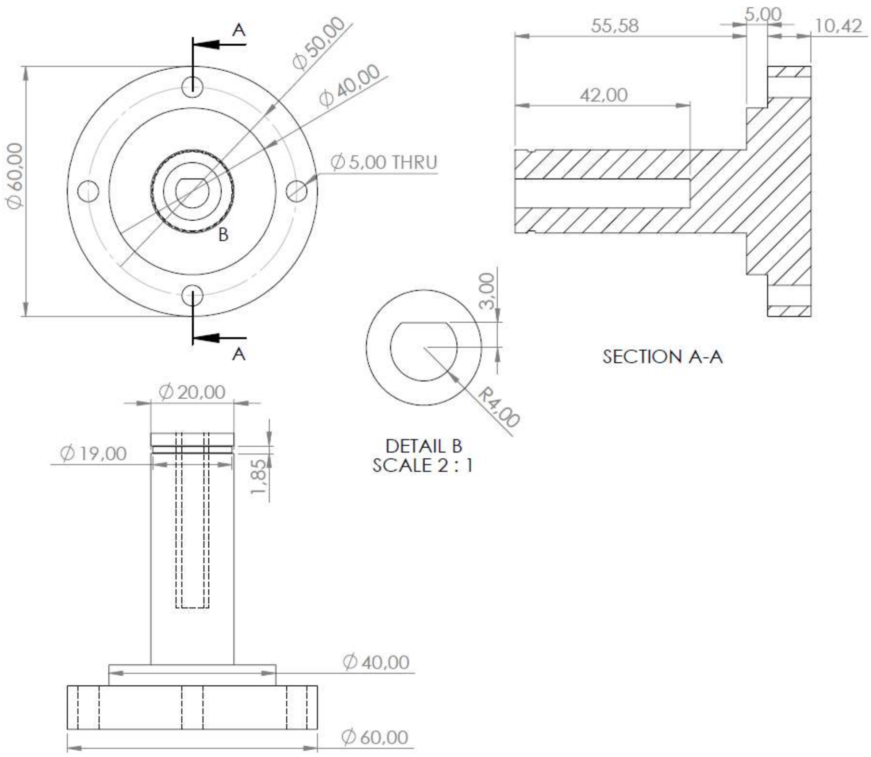


SECTION A-A

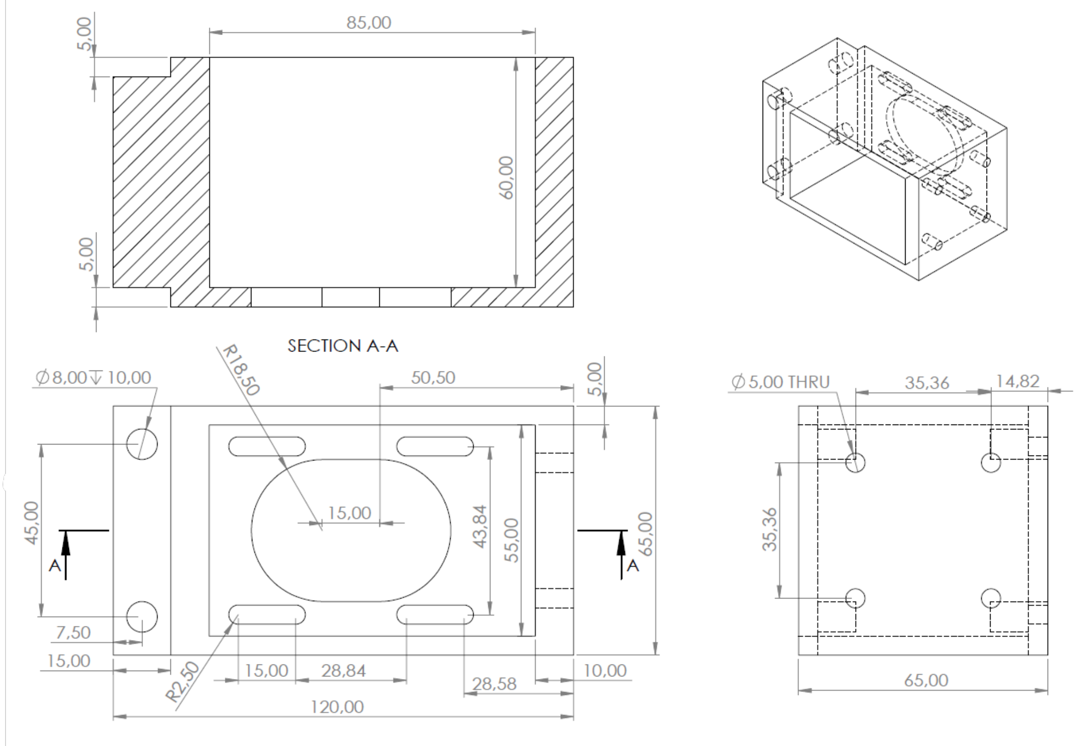
Part_130_A4_Support



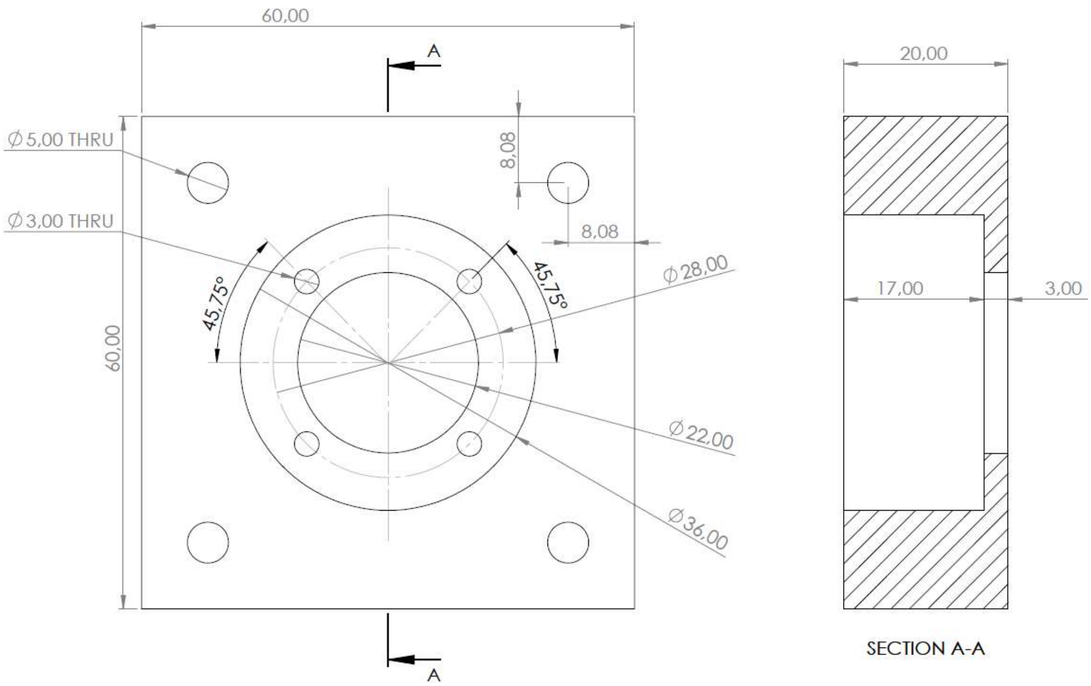
Part_140_A4_Spindle



Part_150_A5_Housing

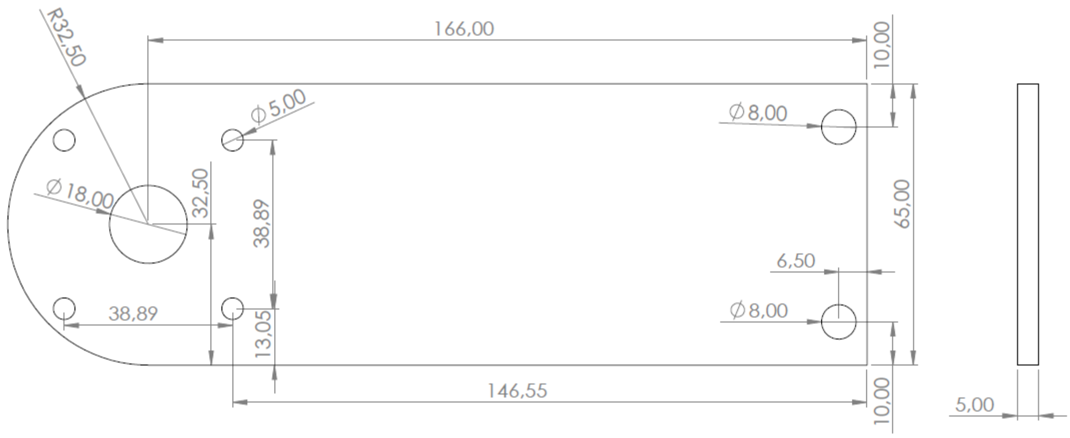


Part_160_A5_Mounting_Plate

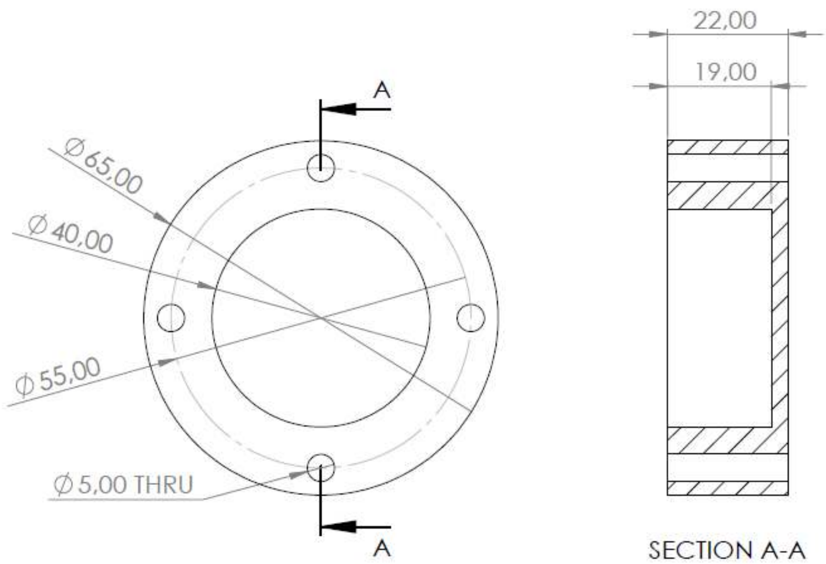


Part_170_A5_Arm_Left

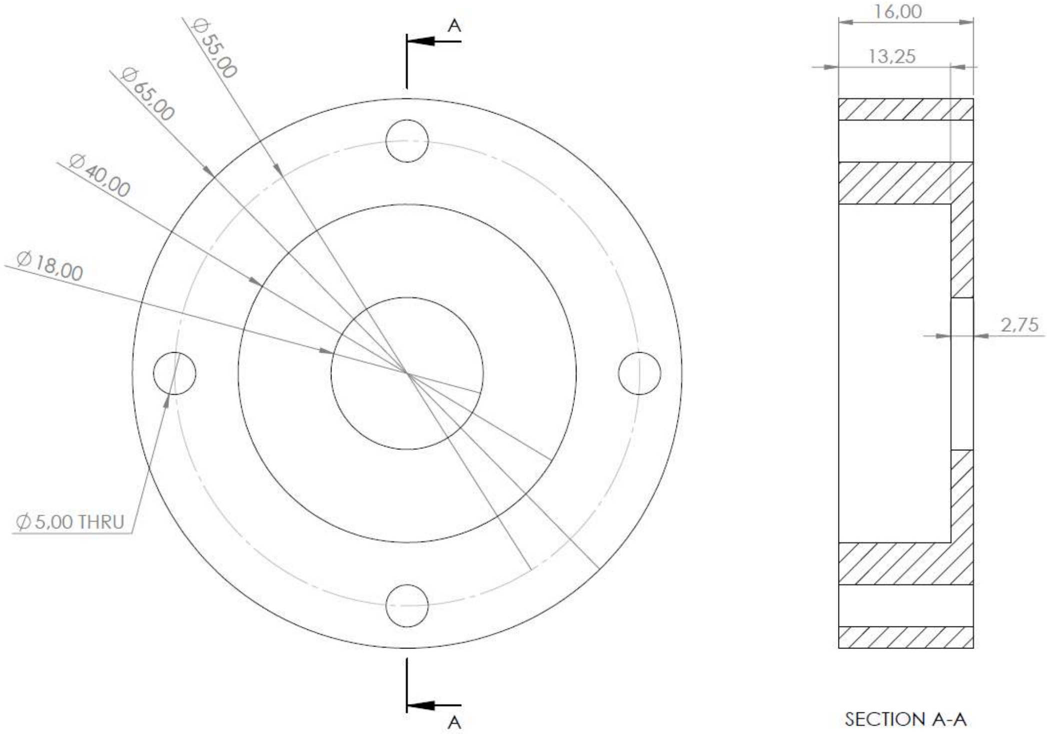
Part_180_A5_Arm_Right



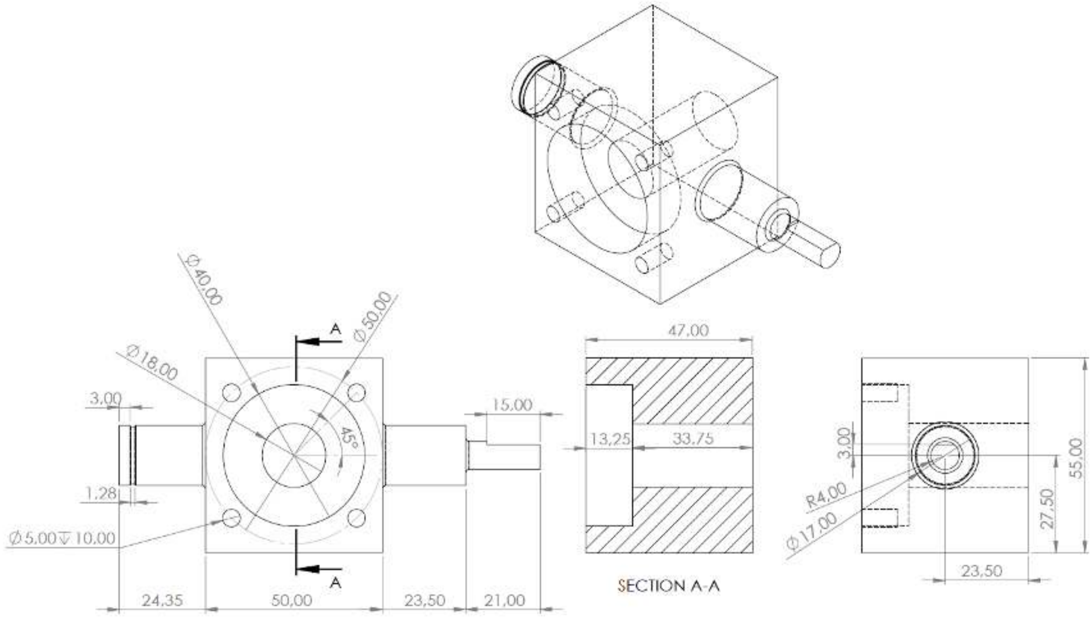
Part_190_A5_Support_Left



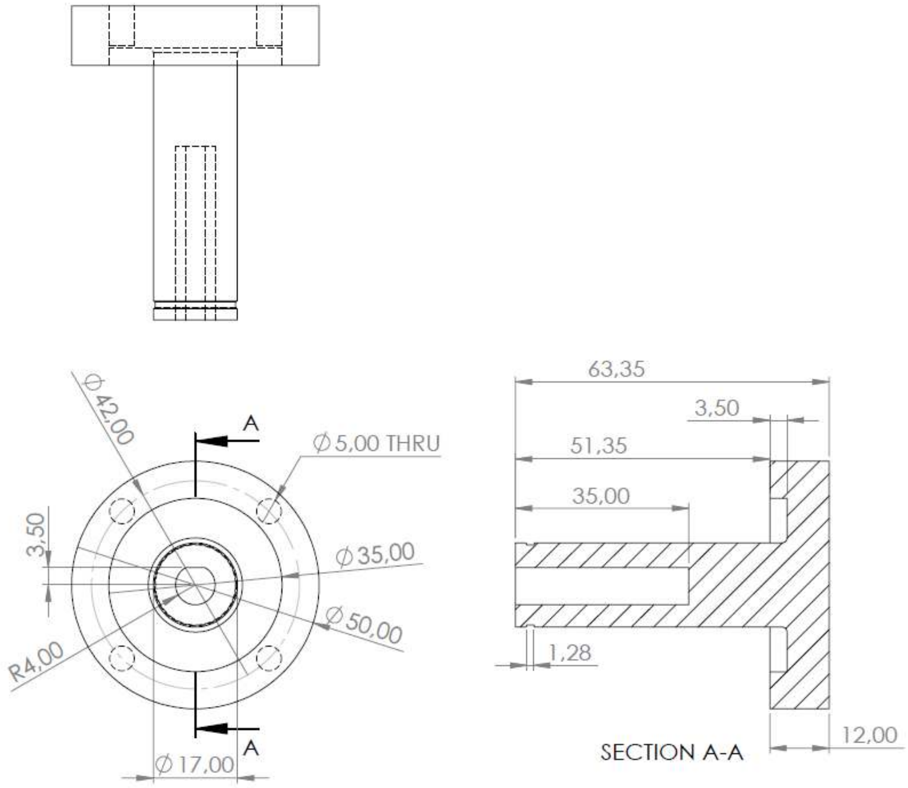
Part_200_A5_Support_Right



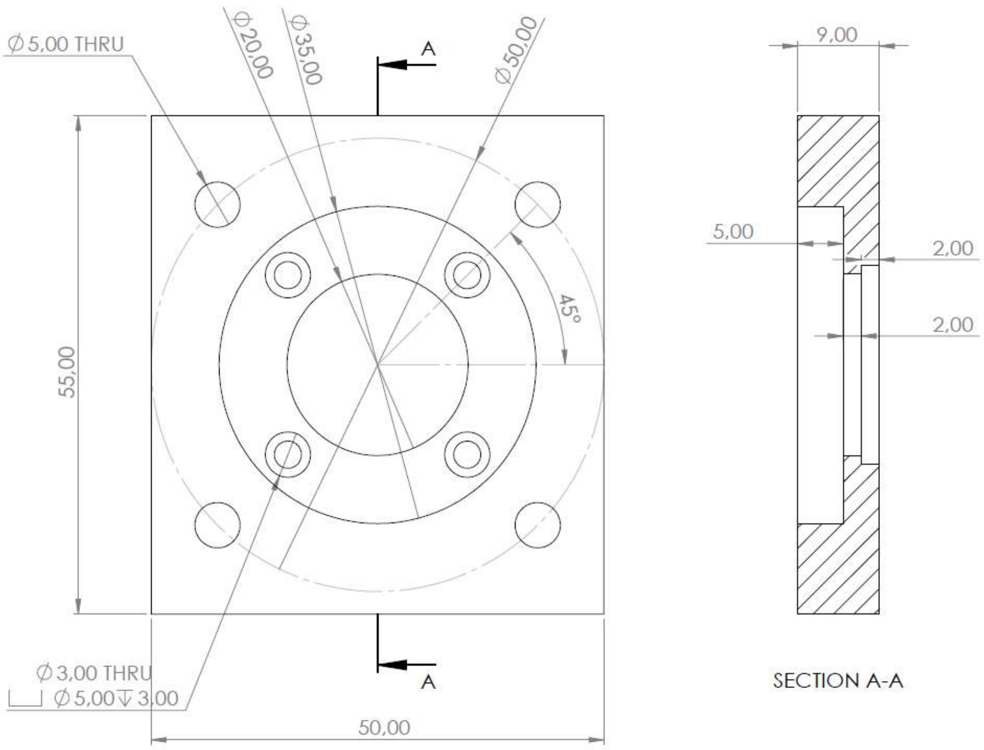
Part_210_A6_Housing



Part_220_A6_Spindle



Part_230_A6_Support



CURRICULUM VITAE

Mehmet Can TÜRK

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

- [redacted]
- [redacted]

[redacted]

[redacted]

[redacted]

[redacted]