TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS

B-01-BME-2019-2024

DESIGN, FABRICATION AND AUTOMATION OF INDUSTRIAL 6-DOF ROBOTIC ARM WITH ON-DEMAND MANUFACTURING CAPABILITIES FOR MASS/BATCH PRODUCTION INDUSTRIES

by

Abishek Chaudhari (076BME001)
Gangotri Sah (076BME012)
Sameer Timsina (076BME037)
Sneha Poudel (076BME042)

A FINAL DEFENSE REPORT
SUBMITTED TO THE DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR IN MECHANICAL ENGINEERING

DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING
LALITPUR, NEPAL

April 2024

COPYRIGHT

TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS
DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING

The undersigned certify that they have read, and recommended to the Institute of Engineering for acceptance, a project report entitled **"Design, Fabrication and Automation of Industrial 6 dof Robotic Arm with On-Demand Manufacturing Capabilities for Mass/Batch Production Industries "** submitted by **Abishek Chaudhari, Gangotri Sah, Sameer Timsina, Sneha Poudel** in partial fulfilment of the requirements for the Bachelor's Degree in Mechanical Engineering.

Supervisor:**Dr. Sanjeev Maharjan**
Assistant Professor
Department of Mechanical and Aerospace Engineering
Institute of Engineering, Pulchowk Campus

External Examiner:**Manisha Maharjan**
Senior Divisional Engineer
Ministry of Education, Science and Technology
Singhadurbar

Tribhuvan University
Institute of Engineering
Department of Mechanical and Aerospace Engineering
Pulchowk Campus

Committee Chairperson: **Asst Prof. Sudip Bhattarai (PhD)**
Head of Department
Department of Mechanical and Aerospace Engineering
Institute of Engineering, Pulchowk Campus

Date of Approval: 09/04/2024

ABSTRACT

Industry 4.0, the fourth industrial revolution, combines digital technologies and automation in manufacturing. This project aims to design and control a six-degree - of freedom robotic arm with counterbalance mechanism and it also incorporates the concept of on-demand manufacturing and uses the model robotic arm to integrate it. The project seeks to enhance manufacturing operations, improve efficiency, and increase productivity. The robotic arm will be driven by servo and stepper motors controlled by Raspberry Pi B+ microcontroller. Real-time communication and control will optimize the manufacturing process, leading to streamlined operations and increased responsiveness. Additionally, weight reduction and counterbalancing mechanism has been implemented to ensure use of cost effective actuators. This project has the potential to transform the manufacturing industry by enabling direct communication between clients and manufacturers, eliminating intermediaries and revolutionizing manufacturing processes.

Keywords: Industrial Internet of Things, Digital Supply Chain, Smart Manufacturing, Cyber-Physical Systems, 6-DOF Robotic Arm

ACKNOWLEDGEMENT

We wish to extend our deepest gratitude to those whose contributions were pivotal in bringing our final report to fruition. Our heartfelt thanks are extended to the Head of the Department, Dr. Sudip Bhattarai, for providing unwavering guidance and steadfast support. We also express our gratitude to Deputy Head of the Department, Assistant Prof. Laxman Motra, whose insightful perspectives played a crucial role in refining our report.

A profound acknowledgment is owed to our supervisor, Assistant Prof. Dr. Sanjeev Maharjan, whose continuous support, expert guidance, and constructive feedback have been instrumental throughout the project. Their mentorship has been a cornerstone in our academic and professional development. A special note of appreciation goes to the director of Mantra Incorporation, Mr.Aditya Neupane for his invaluable mentorship and guidance, significantly shaping the trajectory of our project.

In closing, we sincerely appreciate the entire Department of Mechanical and Aerospace Engineering and its dedicated faculty members. Their commitment to nurturing an environment of academic excellence has been inspiring.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

WIP	Work in Progress
3D	Three DImensional
DOF	Degree of Freedom
IoT	Internet of Things
JIT	just-in-time
PWM	Pulse Width Modulation
IIoT	Industrial Internet of Things
CPS	Cyber-Physical Systems
CSS	Cascading Style Sheet
AI	Artificial Intelligence
GPIO	General Purpose Input/Output
CNC	Computerised Numerical Control
CAD	Computer Aided Designing
SaaS	Software-as-a-Service
PaaS	Platform-as-a-Service
JS	Javascript

HTML Hyper Text Markup Language

CBM Counterbalance Mechanism

CG Center of Gravity

PLA Polylactic Acid

IK Inverse Kinematics

FK Forward Kinematics

DH Denavit-Hartenberg

CNC Computerised Numerical Control

ORM Object-Relational Mapping

DIY Do It Yourself

CPU Central Processing Unit

GT2 Gates Tooth 2mm

SQL Structured Query Language

SLERP Spherical Linear Interpolation

LIST OF SYMBOLS

$\sum$ Summation

θ Theta Angle

F_t Forecast of tth day

w_t Weight of tth day

s_{12} $\sin(\theta_1 + \theta_2)$

s_{123} $sin(\theta_1 + \theta_2 + \theta_3)$

c_{12} $cos(\theta_1 + \theta_2)$

c_{123} $cos(\theta_1 + \theta_2 + \theta_3)$

CHAPTER 1: INTRODUCTION

1.1 Background

The concept of on-demand manufacturing is not entirely new; its roots can be traced back to the principles of lean manufacturing and just-in-time (JIT) production pioneered by companies such as Toyota in the 20th century. However, recent advancements in technology, coupled with shifting consumer preferences and market dynamics, have catalyzed a renewed interest in this approach across diverse industries ranging from apparel and consumer electronics to automotive and aerospace.

The on-demand manufacturing concept is an innovative approach which represents a departure from conventional mass production models, offering a dynamic and responsive framework tailored to meet the evolving needs and preferences of consumers.

At its core, on-demand manufacturing embodies the principle of producing goods in direct response to customer demand, thereby eliminating the need for large-scale production runs and excessive inventory stockpiling. Instead of adhering to rigid production schedules and predetermined quantities, manufacturers leverage real-time data insights to optimize production processes, minimize lead times, and deliver customized products with greater efficiency and precision.

In the world of robots, the improvement of robotic arms has been all about becoming more versatile and precise. A big step forward in this progress is the creation of 6 Degree of Freedom (6-DOF) robotic arms. These are like mechanical arms that can move in six different ways. This new invention has made a big impact, allowing robots to do more things in different industries. 6-DOF robotic arms represent a milestone in robotics, offering unparalleled flexibility and accuracy in performing intricate tasks. With six degrees of freedom, these robotic arms can navigate through three-dimensional space, making them ideal for applications demanding complex and multi-directional movements.

At present,there are numerous industries manufacturing robotic arm for industrial purposes.At the same time, the demand for a small size cheap robotic arm capable of performing simple indoor tasks in home, offices and small industries is also increasing.This project is about designing and building a cheap and reliable robotic arm yet

strong enough to perform simple works like pick and place in a precise manner.The robotic arm will utilize widely available manufacturing methods like 3D-Printing and utilize cheap actuators such as stepper motors.

The job of Robotic arm becomes more interesting if it is controlled through wireless communication network.This project will add on this feature of long distance communication with the robot through the use of Raspberry Pi and internet protocols.

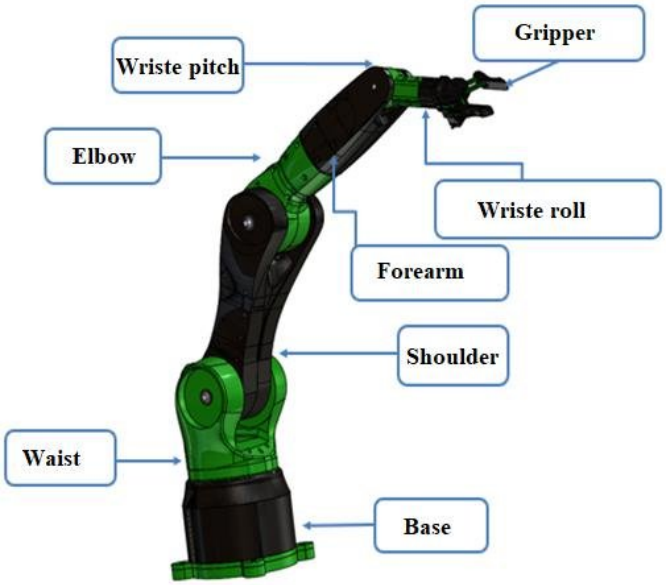


Figure 1.1: Structure of a 6 DOF Robotic arm [1]

1.2 Problem Statement

While the concept of on-demand manufacturing isn't entirely novel, its adoption remains limited within the manufacturing industry. The project's driving force is to overcome this discrepancy by accomplishing the mentioned objective.The project thus aims to accomplish the implementation of on-demand manufacturing by presenting data from any point in the supply chain and uses robotic arm to integrate its concept.This concept of on demand manufacturing allows the end-user to observe their order status,and also allows for customization and flexibility by producing items as needed.

Similarly,robotic arms maneuvers are the offering of the combination of actuators movements.These actuators need tremendous power and reduction system to handle self weight and weight of payload. High power actuators are not economical. This projects

also intends to study, design and fabricate a spring counterbalance system inside a 6 DOF robotic arm to reduce the power requirement of actuators and hence support to build a cost effective robotic arm.

1.3 Objectives

1.3.1 Main Objectives

Design, Fabrication and Automation of 6-DOF robotic arm with on-demand manufacturing capabilites for mass/batch production industries.

1.3.2 Specific Objectives

i. Streamline the order process by developing a client-side website that enables real-time order placement, provides comprehensive product information, and offers a login system for order tracking.

ii. Enhance order management and real-time tracking capabilities by creating a software program that integrates with user and order database.

iii. Enhance inventory control by implementing software to track and monitor each material in storage, facilitating proactive inventory management.

iv. Design and Fabrication of 6-DOF Robotic Arm with spring counter-balancing mechanism within the body.

v. Simulation of kinematics of the robotic arm within the MATLAB environment.

1.4 Scope & Limitations

1.4.1 Scope

This project is intended to build a industrial management software aiding customers in order placement and employees in job assignment to the machines in the industry. Also the research of 6DOF robotic arm is done by incorporating spring counterbalancing mechanism. The concept of counterbalancing is projected to reduce the cost of actuators required, hence making it more economical.

1.4.2 Limitations

- This project doesn't intend to build a industry-ready system. It deals with the development of prototypes with control features that reflects the enterprise systems.
- The robot prototype is an open-ended system and thus provides no feedback on the status of the system.
- This project is focused towards incorporating CBM within a 6 DOF robotic arm and its technicalities and doesn't cover the ergonomics and aesthetics part of the built prototype.

1.5 Features

The design has undergone through multiple iterations to accommodate the counterbalance mechanism along with six actuators and in addition, make it manufacturable.The design has the following features:

1. The gravitational torque due to self weight of the arms is counterbalanced by the spring counterbalance mechanism. Due to this, torque requirement for actuator is significantly reduced. Therefore, a reduction ratio of only 1:6 is designed in the shoulder.The actuator has a holding torque of 10.1 kgfcm making the capacity of the transmission system to be 40.4 kgfcm.

2. The configuration of the arm is RPPPYR .(R – Roll, P – Pitch , Y – Yaw). We can formulate analytical solutions for such arm configuration and need not go towards numerical approach(methods).

3. The actuators for joint 2, 3 and 4 are all mounted on the base. This design approach reduces the gravitational torque in joint 2 and 3 and thus reduces the required spring stiffness to counterbalance the self weight. This also reduces the rotational inertia acting on joint 1.

4. To withstand the overall weight and rotate smoothly, a thrust bearing and two radial ball bearings are used in joint 1.

1.6 System Requirements

1.6.1 Hardware Requirements

- MG996R Servo Motor
- SG90 Micro Servo Motor
- 5V 2A DC Power Supply
- Raspberry Pi
- Jumper Wires
- NEMA 23 and NEMA 17 Stepper Motor

1.6.2 Software Requirements

- Python with modules - streamlit, sqlalchemy, RPi.GPIO
- Raspberry Pi OS

CHAPTER 2: LITERATURE REVIEW

2.1 Industrial Evolution of 6-DOF Robotic Arm

The first ”position controlling apparatus” was patented by Willard Pollard presenting a spray fishing robotic arm with five degrees of freedom. However, he never built his arm but inspired by his design and interest in automated robotic arm ,Harold A. Roselund , working for De Vilbiss, successfully developed another sophisticated sprayer based on Pollard's concept.Though both arms were advanced for their time, the electronic controller systems lacked the fidelity required for broad utilization.

The modern era of robotics took its initial steps with these little-known arms developed in the late 1930s. Unimate introduced its first robotic arm in 1962, invented by George Devol and marketed by Joseph Engelberger.It was mainly used for material handling in manufacturing. The first industrial arm found its place at the General Motors plant in Ternstedt, New Jersey, for automated diecasting, marking the transition of industrial robots from the laboratory to the factory. Interestingly, during this transition, nautical terms like pitch, yaw, and roll were incorporated to describe the movements and degrees-of-freedom of robotic arms.In 1966, Kawasaki acquired the license to manufacture industrial robot arms from Unimation. Swift competition arose, with Cincinnati-based Milacron and AMF Hermatool introducing commercially available robots. In 1963, AMF Hermatool launched the Verstran industrial robot, imported by Japan in 1967. Academic interest in microelectronics and the potential applications of robotic arms surged, leading to the development of more sophisticated designs.

Victor Scheinman, an investigator at Stanford Research Institute, pioneered electrically powered articulated arms that could move through six axes, known as the Stanford arm. This innovation allowed for more complex tasks to be assigned to robotic arms. Marvin Minsky, from MIT, constructed a robotic arm for the Office of Naval Research, designed for possible underwater exploration, utilizing twelve single-degree-of-freedom joints for electro-hydraulic high-dexterity. Scheinman continued his work on robotic arms, and with backing from General Motors, Unimation developed Schein-

man's technology into a Programmable Universal Machine for Assembly (PUMA).

As industrial robots became more sophisticated, the number of degrees of freedom increased. By the 1970s, 6-DOF robotic arms were becoming more common in industrial settings, enabling a greater range of motions and tasks.In the 1980s, advancements in computer technology and control systems contributed to increased precision and flexibility. By the 1990s, 6-DOF robotic arms expanded beyond traditional manufacturing, with advancements in sensors and programming making them suitable for various tasks. The 2000s saw miniaturization, leading to smaller and more precise arms used in medical surgery and laboratory automation. In the 2010s, collaborative robots with 6 degrees of freedom, or cobots, emerged, designed to work alongside humans in various industries. Today, these arms continue to be widely used, with ongoing integration of artificial intelligence and machine learning enhancing their adaptability and intelligence.

2.2 Linux Machines - Raspberry Pi

Linux machines are found to have low energy consumption. The operating system also termed Linux distributions has different configurations and control settings based on the need and environment the machine is running in. Similar to every Linux machine, Raspberry Pi, a chipboard running with Raspberry Pi OS is so far commonly used in many IoT-based projects and is popular among students and hobbyists.
Raspberry Pi Model B+ is an upgraded version of the original Model B, released in July 2014. In Raspberry Pi Model B+, the GPIO header now has grown to 40 pins, with the first 26 pins maintaining the same pin out as the previous Model A and B. Additionally, there are now four USB 2.0 ports, an improvement over the two ports found on the Model B, offering enhanced hot plug and over current behavior. Another notable upgrade is the replacement of the old friction-fit SD card socket with a more convenient push-push micro SD version. The board continues to support 100 Base Ethernet, just like the original Model B. Power consumption has been significantly reduced by replacing linear regulators with more efficient switching ones, resulting in a decrease of approximately 0.5W to 1W. Furthermore, the audio circuit features a dedicated low-noise power supply, ensuring improved audio performance. In terms of aesthetics, the

USB connectors have been aligned with the board edge, the composite video has been moved to the 3.5mm jack, and four well-positioned mounting holes have been added, providing a tidier overall form factor.[2]

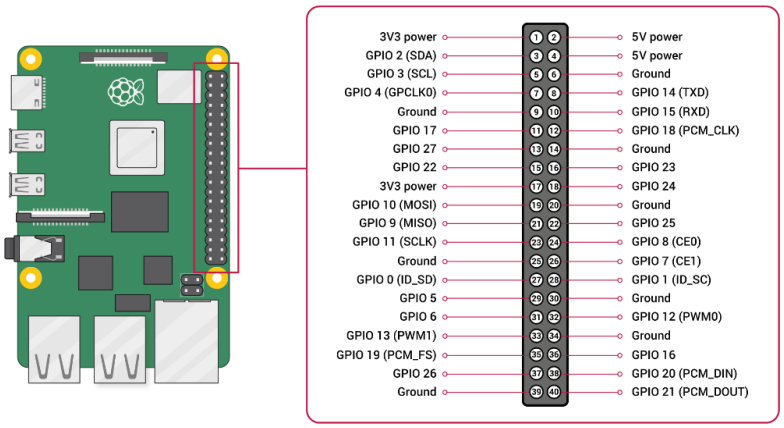


Figure 2.1: Circuit diagram of Raspberry Pi B+[2]

The Raspberry Pi Model B+ is equipped with GPIO (General Purpose Input/Output) pins, which serve as a fundamental interface for physical computing and electronics projects. Introduced as an enhanced version of its predecessor, the B+ features 40 GPIO pins, expanding the capabilities for hardware interfacing and offering increased flexibility for DIY enthusiasts, educators, and developers.

In the context of the Raspberry Pi B+, these GPIO pins play a crucial role in enabling the board to communicate with and control external devices. Whether configuring a pin as an input to receive data from sensors or setting it as an output to control LEDs, motors, or other components, the GPIO pins facilitate the interaction between the Raspberry Pi and the physical world. The specific layout and numbering of these pins on the B+ model can be referenced from the official documentation provided by the Raspberry Pi Foundation.

Programming the GPIO pins on the Raspberry Pi B+ is typically done through programming languages like Python or C. The RPi.GPIO library in Python, for instance, simplifies the process of manipulating these pins in software. Given that GPIO pins operate at 3.3 volts on the Raspberry Pi B+, users must be mindful of voltage levels to prevent damage to the board. These pins are physically located along the edge of the B+ board, allowing users to connect jumper wires or other electronic components directly,

or by using a GPIO header for more organized and secure connections.

2.3 Actuators

For the PLA-based robotic arm prototype, the following motors were used:

1. MG996R Servo Motor (3x)
2. SG90 Micro Servo Motor (3x)

MG996R Servo Motor

The MG996R servo motor is a high-torque, metal-geared servo motor often used in robotics, remote-controlled vehicles, airplanes, and other applications requiring precise control of angular motion.
Motor Specifications:

- Weight and Dimensions: 40.7mm x 19.7mm x 42.9mm, 50g
- Torque: 11kg/cm at 4.8V and 13kg/cm at 6V
- Power Supply: 4.8-6V
- PWM Control Frequency: 50 Hz
- Pulse width(angle) : 1000μs(0 degree) to 2500μs (180 degrees)
- Gear Train: Metallic

The MG996R is compatible with most microcontroller platforms such as Arduino, Raspberry Pi, and other development boards. It can be easily controlled using PWM signals generated by these platforms, making it versatile and widely used in hobbyist and educational projects.

SG90 Micro Servo Motor

The SG90 micro servo motor is a small, lightweight servo motor commonly used in smaller-scale projects where space and weight constraints are significant factors. Here's an specification of the motor:

- Weight and Dimensions: 23mm x 12mm x 29mm, 9g
- Torque and speed: 1.8kg/cm and 0.12 sec/60° at 4.8V
- Power Supply: 4.8-6V
- PWM Control Frequency: 50 Hz
- Pulse width(angle) : 1000μs(0 degree) to 2500μs (180 degrees)
- Gear Train: Plastic

In later prototype, stepper motors are chosen as actuator. Stepper motor is an electric motor that converts electrical pulses into precise and controlled mechanical rotation. It consists of a rotor and a stator with coils. By energizing and de-energizing the coils in a specific sequence, the motor moves in small increments or steps, allowing for accurate positioning and speed control. Stepper motors offer advantages such as high torque at low speeds, excellent positioning accuracy, and the ability to maintain position without external feedback. They are widely used in robotics, automation, 3D printers, CNC machines, and other applications where precise motion control is essential. Stepper motor used are JK57HS51-2804 NEMA 23 stepper motor,17PM-J349-P1VS NEMA 17 and NEMA 17 pancake stepper motor.

NEMA 23 Stepper motor

The NEMA 23 10.1 kg-cm Hybrid Stepper Motor stands out as an ideal choice for heavy-duty CNC machines,due to its superior operational advantages, notably its high torque at elevated speeds. With an impressive torque output of 10.1 kg-cm at 2.8A current per phase, this stepper motor is a powerhouse, particularly suited for larger machines that demand extra torque. Renowned for its precisely repeatable steps, it excels in applications requiring accurate position control, making it the preferred motor for CNC milling and CNC router setups that necessitate more torque. Being a brushless DC motor, its longevity is linked to the life of the bearings, and it operates seamlessly with a simple Open Loop control mechanism, eliminating the need for intricate electronic control circuitry. The motor's machined shaft ensures a reliable grip with pulleys or drive gears, minimizing the risk of stall or slip during operation.[3]

NEMA 17 Stepper motor

The NEMA 17 stepper motor, model 17PM-J349-P1VS, boasts a 1.8-degree step angle in a unipolar drive configuration, featuring a rated current of 1 Amperes and a resistance of 3.6 Ohms. With a substantial holding torque of 200 mNm, this motor ensures robust performance in applications demanding precise control over rotational movements. Its inductance of 4.7 millihenrys contributes to efficient current flow, while a rotor inertia of 35 gram-square centimeters enhances its responsiveness. The detent torque, measuring at 7.9 mNm, minimizes static friction during stationary periods. Weighing in at 200 grams, this NEMA 17 stepper motor is a compact yet powerful solution, well-suited for various applications where accurate and controlled motion is essential.[4]

2.4 3D Printer

The body of the motor will be 3D printed with the help of a 3D printer. A 3D printer is a revolutionary device that enables the creation of three-dimensional objects by sequentially depositing layers of material based on a digital model or design. It works by utilizing additive manufacturing technology, where the object is built layer by layer from the bottom up. The process begins with a digital 3D model that is sliced into thin cross-sectional layers. The 3D printer then follows these layers' instructions, precisely depositing or curing materials such as plastics, metals, ceramics, or even biological substances. This layer-by-layer approach allows for the fabrication of intricate and complex objects with high precision and accuracy. 3D printers have gained immense popularity in various industries, including manufacturing, engineering, healthcare, education, and design, as they offer immense flexibility, rapid prototyping capabilities, and the ability to produce custom-made parts and products on demand.

2.5 SolidWorks

Based on the design parameters, the robotic arm will be developed in 3D CAD Software such as SolidWorks. SolidWorks is a widely used computer-aided design (CAD) software that allows users to create 3D models, perform simulations, and generate technical drawings. It provides a user-friendly interface and powerful tools for designing and analyzing products. SolidWorks supports parametric modeling, and simulations, and offers

extensive compatibility for collaboration with other CAD software. It is employed in various industries for product design and engineering tasks.

2.6 Counterbalance mechanism for robotic arm

Counterbalancing of a robotic arm refers to the use of various mechanisms or systems to offset the weight of the arm itself and the payload it carries. A Counterbalance Mechanism (CBM) is a mechanical device that cancels or reduces the gravitational torque applied to a linkage joint with compensation torque generated by a counterweight or springs. Several techniques are employed for counterbalancing robotic arms, including spring counterbalancing, weight counterbalancing, pneumatic counterbalancing, hydraulic counterbalancing, and tension-based systems. The choice depends on factors like size, specific application, available resources, cost-effectiveness, desired accuracy, precision, and various other considerations. The most commonly used method is spring counterbalancing which utilizes spring to offset the gravitational forces acting on arm and payload. It is simple, reliable effective in most robotic applications so we have also used spring counterbalancing in combination with wires and pulleys along with the use of a double parallelogram mechanism in our robotic arm.

Typically, traditional robot arms utilize costly gear reducers and high-performance actuators to generate substantial torques, enabling the arm to endure its weight and movement. In most cases, inexpensive or low-capacity components cannot produce the necessary torques for supporting the robot arm. This is a key factor contributing to robots' elevated cost and potential collision risks. Therefore, there is a need for innovative strategies aimed at diminishing the required torques for the efficient operation of robot arms.

Addressing these challenges, counterbalance mechanisms have been developed to efficiently offset the torques necessary for the operation of a robot arm. Nevertheless, earlier counterbalance mechanisms posed usability challenges for diverse robot arms, such as limitations on the number of applicable joints, restricted operational range, and the drawbacks of being heavy and bulky.[5]

Kim et al.[5] (2014) employed counterbalancing mechanism that uses a combination of spring wires and pulleys, coupled with a specially designed double parallelogram mechanism. The double parallelogram mechanism used in the four-bar mechanism is

modified by using springs, wires, and pulleys to build the CBM mechanism at each joint of the arm. This unique double parallelogram mechanism is instrumental in preserving reference axes for each joint. By integrating this counterbalancing mechanism (CBM), we successfully offset gravitational torques across nearly all robot configurations using compression springs. This implementation has enabled us to achieve a dedicated CBM mechanism for each link.

Kim et al.[5] (2014) mentioned different ideas on how the springs can be used or oriented to get the proper counterbalance.

2.6.1 1 DOF counterbalance mechanism

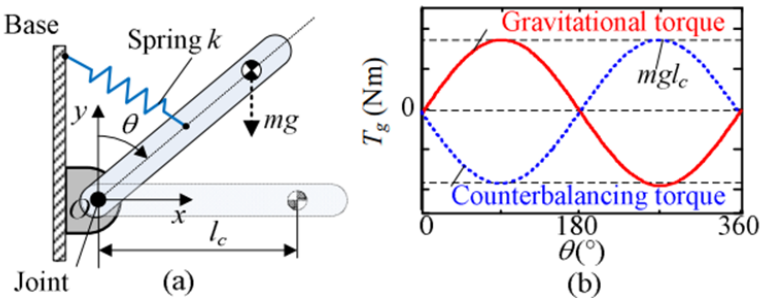


Figure 2.2: Counterbalance for 1 DOF link[5]

When the link is rotated in Clockwise direction as shown in Figure 2.1, the gravitational torque,

$T_g = mgl_c sin\theta$ .....(2.1)

Where,

m = mass of link

l_c = distance from the joint axis to link's center of mass

θ= angular displacement of a link from y-axis

is applied at the joint. This gravitational torque can be compensated by installing a spring of suitable stiffness k as shown in Fig above. To incorporate this system into a robotic arm and achieve a more compact design, the following design approach has been employed by Kim et al.[5](2014) as shown in Fig below which can provide counterbalancing torques for all arm positions.

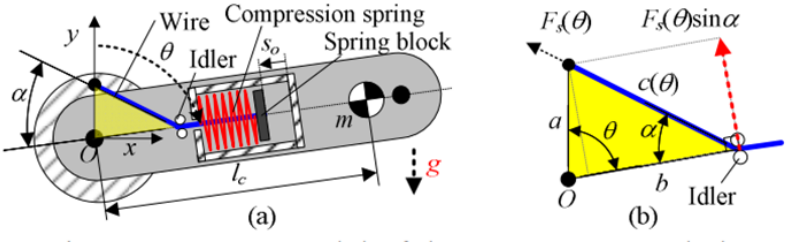


Figure 2.3: Concept model of 1DOF CBM[5]

Counterbalancing torque due to spring force in the idler is given by:

$T_c = b * k * [s_o + c(\theta) - (b-a)] sin(\alpha)$.....(2.2)

Where, k= stiffness of spring

s_o = Initial compression of spring

c(θ) = distance between upper fixed pulley and idler

a = distance between two fixed pulleys

b = distance between lower fixed pulley and idler

For complete counterbalancing, the gravitational torque must be equal to the counterbalancing torque, i.e. $T_g = T_c$

$mgl_c \sin\theta = b \cdot k\,[s_o + c(\theta) - (b-a)]\sin(\alpha)$ .....(2.3)

From Fig 2.2(b), using sine law:

$$\sin(\alpha) = \frac{a\sin(\theta)}{c(\theta)} \text{ .....(2.4)}$$

From equations (2.3) and (2.4):

$$mgl_c \sin\theta = b \cdot k \cdot [s_o + c(\theta) - (b-a)] \cdot a \cdot \frac{\sin(\theta)}{c(\theta)} \text{.....(2.5)}$$

If s_o is set to (b-a), then this configuration will provide proper counterbalancing torques for all positions of the arm irrespective of the values of θ and equation (2.5) becomes

$k = \frac{(m \cdot g \cdot l_c)}{(a \cdot b)}$ .....(2.6)

In equation (2.6), all the values on the right-hand side are known. The value of k can be determined and can be adjusted to the required value by changing the values of a and b.

2.6.2 Multi – DOF counterbalance mechanism

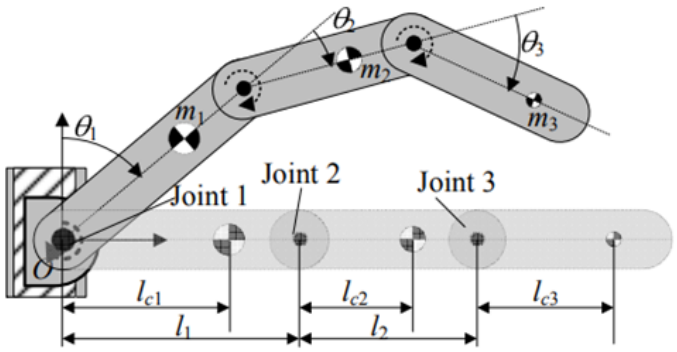


Figure 2.4: Gravitational torque at Multi DOF robotic arm[5]

Consider a 3-DOF robotic arm as shown in Fig 2.3. the gravitational torque acting on each link is given by the following equations:

$$T_{g1} = (m_1 g l_{c1} + m_2 g l_1 + m_3 g l_1) s_1 + (m_2 g l_{c2} + m_3 g l_2) s_{12} + m_3 g l_{c3} s_{123} \quad (2.7)$$

$$T_{g2} = (m_2 g l_{c2} + m_3 g l_2) s_{12} + m_3 g l_{c3} s_{123} \quad (2.8)$$

$$T_{g3} = m_3 g l_{c3} s_{123} \quad (2.9)$$

Here, s_1, s_{12}, s_{123} are $\sin(\theta_1), \sin(\theta_1+\theta_2), \sin(\theta_1+\theta_2+\theta_3)$ respectively. From equations (2.7), (2.8), and (2.9), we can see that the gravitational torques at each joint are dependent on the other joint's motion. In such cases, proper counterbalancing torques cannot be generated by the proposed mechanism for each joint for compensation of gravitational torques. The counterbalancing mechanism is not affected by the rotation of other joints even though each joint is equipped with a separate CBM.

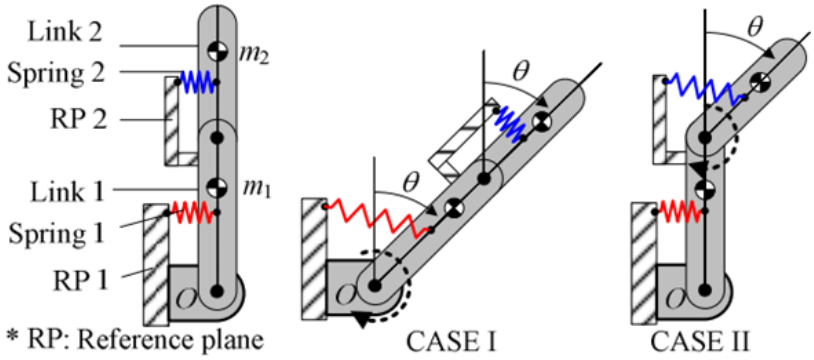


Figure 2.5: Failure of Multi DOF CBM[5]

To illustrate the failure of multi-DOF CBM, Kim et al.(2014) considered a 2-DOF robotic arm with two single-DOF counterbalance mechanisms as shown in Figure 2.4.

In case I, when link 1 and link 2 rotate together at an angle θ. The reference plane of link 2 is attached to link 1.This will help in rotating link 2 with rotation of link 1.Therefore, spring 2 will not be able to generate the required counterbalance torque at joint 2. In case II, only link 2 is rotated at the same angle θ but link 1 is stationary. The movement of link 2 will produce gravitational torque at joint 1 which cannot be compensated by spring 1. This is because link 1 is stationary. It is clear that this mechanism is not useful for multi–DOF robotic arms. Although identical gravitational torque is applied at joint 2 in both configurations, spring 2 cannot generate counterbalancing torque in case I but it can do so in case II. Hence, each link should have its own fixed reference plane for counterbalancing and the movement of the reference plane at each joint should be independent to the movement of other joints. To create this fixed reference plane, the option for a double parallelogram mechanism based on pulleys and wires is taken.[5]

2.6.3 Implementation of double parallelogram mechanism:

To address issues related to the failure of multi-degree-of-freedom links, Kim et al.[5] (2014) utilized a double parallelogram mechanism. The double parallelogram mechanism is usually constructed of a 4-bar linkage. In this mechanism, the driven link is in the same direction irrespective of the movement of the driving link as shown in Fig. 2.5(a). The links can be replaced by wires and pulleys as shown in Fig 2.5(b).

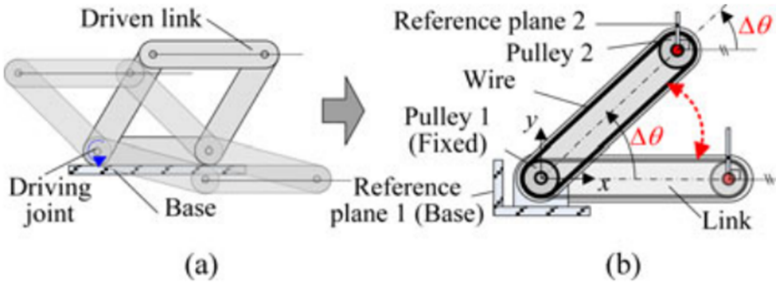


Figure 2.6: Implementation of double parallelogram mechanism[5]

The parallelogram mechanism keeps a consistent angle at pulley 2's fixed reference plane, typically aligned with gravity. This setup enables the generation of counterbalancing torque at joint 2 relative to reference plane 2. Instead of rotating the link, the wire connects pulley 2 to the fixed pulley 1. The double parallelogram technique establishes a reference plane for each of the three pitch motions of the robotic arm through pulleys

and wires. A spring is inserted between the reference plane and its corresponding link to produce the counterbalancing torque.

2.6.4 Working of double parallelogram mechanism

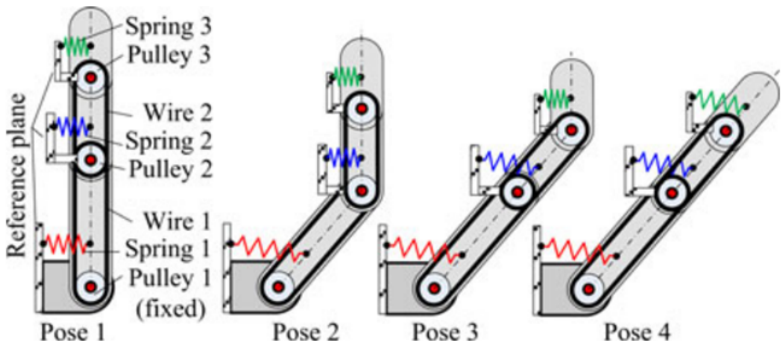


Figure 2.7: Working of double parallelogram mechanism[5]

In Figure 2.6, a counterbalancing mechanism with a reference plane at each joint for 3-dof robotic arm is shown. We designed a similar counterbalancing mechanism for three continuous pitch joints: the lower two joints and the wrist joint. Each of the three joints is provided with fixed reference planes. When the first link rotates, the gravitational torque that is generated at joint 1 is due to the gravitational torque of all the links 1, 2 and 3 as can be seen from equation (2.8) if the reference plane is not created. But when the reference planes are created at joint 2 and 3, the links 2 and 3 do not rotate with link 1 as in previous case and hence the gravitational torque due to link 2 and 3 are not induced at joint 1. Therefore, CBM 1 has to counterbalance the gravitational torque of link 1 only. Similarly, when link 2 rotates, CBM 2 has to compensate the gravitational torque induced due to link 2 only. The gravitational torque of link 3 is counterbalanced by CBM 3 at joint 3. In this way, each counterbalance mechanism has to balance the gravitational torque of their corresponding links. In the case of a 3-degree-of-freedom (DOF) robotic arm, the gravitational torque at joint 1 is composed of three terms: $(m_1 g l_{c1} + m_2 g l_1 + m_3 g l_1) s_1$, $(m_2 g l_{c2} + m_3 g l_2) s_{12}$, $m_3 g l_{c3} s_{123}$ (from eq. 2.8) which represents the gravitational torques due to links 1, 2, and 3, respectively. With the implementation of the double parallelogram mechanism, Counterbalance Mechanism 1 (CBM 1) only needs to compensate for the gravitational torque due to link 1. The next two terms are counterbalanced by CBM 2 at joint 2 and CBM 3 at joint 3.

Similarly, the gravitational torque at joint 2 comprises two terms: $(m_2 g l_{c2} + m_3 g l_2) s_{12}$, and $m_3 g l_3 s_{123}$ (from eq. 2.9). In this case, the second term is balanced by CBM 3 installed at joint 3. This approach ensures proper counterbalancing in all joints through the use of the parallelogram mechanism.[5]

2.6.5 Determination of value of k (spring stiffness) for multi- DOF robotic arm

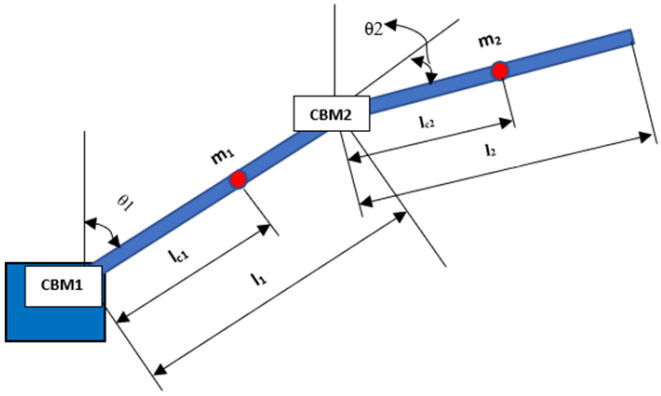


Figure 2.8: Two DOF arm with CBM at Joints

Now, consider 2 -DOF robotic arm as shown in figure. The gravitational torques acting at each joint are given by equations:

$T_{g1} = (m_1 g l_{c1} + m_2 g l_1) s_1 + m_2 g l_{c2} * s_{12}$ .....(2.10)

Similarly,

$T_{g2} = m_2 g l_{c2} * s_{12}$ .....(2.11)

By the implementation of double parallelogram mechanism, we have seen that among the two terms in equation (2.8), spring1 has to counterbalance only the first term. The second term is balanced by spring 2 in CBM 2.

Now, for link 1 to obtain proper counterbalancing the spring force must be equal to the first term of equation (2.8)

$$(m_1 g l_{c1} + m_2 g l_1) s_1 = b * k_1 * [s_o + c(\theta) - (b-a)] * a * \frac{s_1}{c(\theta)} \quad \text{.....(2.12)}$$

Again, if s_o is set to (a-b) then the above equation becomes:

$$k_1 = \frac{(m_1 g l_{c1} + m_2 g l_1)}{ab} \quad \text{.....(2.13)}$$

Similarly, for link 2, the value of k will be:

$$k_2 = \frac{(m_2 g l_{c2})}{ab} \quad \text{.....(2.14)}$$

where, k_1=spring constant for CBM 1 spring

k_2=spring constant for CBM 2 spring

In this way, the value of spring constants for each spring in CBM can be determined.

Following thorough research and analysis, our ultimate decision is to implement a Counterbalancing Mechanism (CBM) utilizing a spring and pulley system, complemented by a specially designed double parallelogram mechanism. To achieve a lightweight design, we've incorporated a spring and pulley system, and the double parallelogram mechanism is employed to establish reference axes for each joint. Lee's modifications to Kim's work, as presented in Lee et al.[6](2016) and Lee et al.[7](2017), introduce a multi-winding system to reduce spring stiffness, input torque, and wire tension which is also implemented in our design. Lee's research demonstrates that the input tension exhibits an inverse relationship with the number of windings in the pulley (n). Additionally, the spring stiffness is inversely proportional to the square of n.[7]

2.7 Kinematics

Our 6 DOF robot consists of 6 revolute joints viz. base, shoulder, elbow, wrist pitch, wrist yaw, and wrist roll. The primary challenge in the automatic control of robot manipulators revolves around kinematics. Robot kinematics, a focal point in this context, involves the examination of the motion of robotic systems, specifically focusing on position and velocity. It deals with the geometric and dynamic aspects of the robot's motion without considering the forces or torques that cause the motion. It forms the basis for trajectory planning, motion control, and overall system optimization. It involves forward kinematics and inverse kinematics problems.

Forward kinematics deals with the determination of the end effector's position and orientation based on joint angles or joint displacements. The forward kinematics equations describe the relationship between the joint variables and the position and orientation of the end-effector. These equations are often represented by homogeneous transformation matrices (${}^0T_T$).

Inverse Kinematics deals with the determination of joint variables needed to achieve the desired position and orientation of the end effector. In simple terms, it is the opposite of forward kinematics. Both forward and inverse kinematics require a relation between the world frame and the tool frame of reference to map the coordinates from one frame to another. This relation is given by the Homogeneous Transformation Matrix. The

closed-form solution for the inverse kinematics problem has been successfully attained through the application of methods elucidated by Hayawi, M. J.[11] (2012)[8]. This approach enables the determination of joint variables corresponding to a desired end-effector position and orientation.

Homogeneous Transformation Matrix is a 4*4 matrix that combines rotation, translation and scaling transformation in a single matrix. It is used to transform coordinates from one coordinate system to another.

Symbolically,

$$^{0}\mathrm{T}_{\mathrm{T}} = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix}$$

where,

$^{0}T_{T}$ is a homogeneous transformation matrix of frame T relative to frame 0,

R= 3X3 Rotation matrix which represents the orientation of the end effector,

P= 3X1 Translation vector which represents the position of the tool tip.

This transformation matrix for a robotic manipulator is developed using the Denavit-Hartenberg(D-H) theorem.

The methodology adopted to address the kinematics problems draws inspiration from the works of Hayawi, M. J.[11] (2012)[8].

For the forward kinematics problem, the Denavit-Hartenberg (DH) algorithm has been employed to derive a comprehensive solution. This algorithm facilitates the determination of the end-effector pose based on the joint parameters, contributing to the effective modeling of the robotic system.

The Denavit-Hartenberg (D-H) theorem is employed to model the kinematics of each joint, facilitating a systematic approach to solving both forward and inverse kinematics problems. The Denavit-Hartenberg parameters provide a standardized method for describing the geometry and kinematics of robotic systems. The D-H parameters for each joint are determined by the help of the D-H algorithm. The parameters include

link length (a), link twist (α), joint distance (d), and joint angle (θ).The representation of D-H parameters is shown in the figure below: -

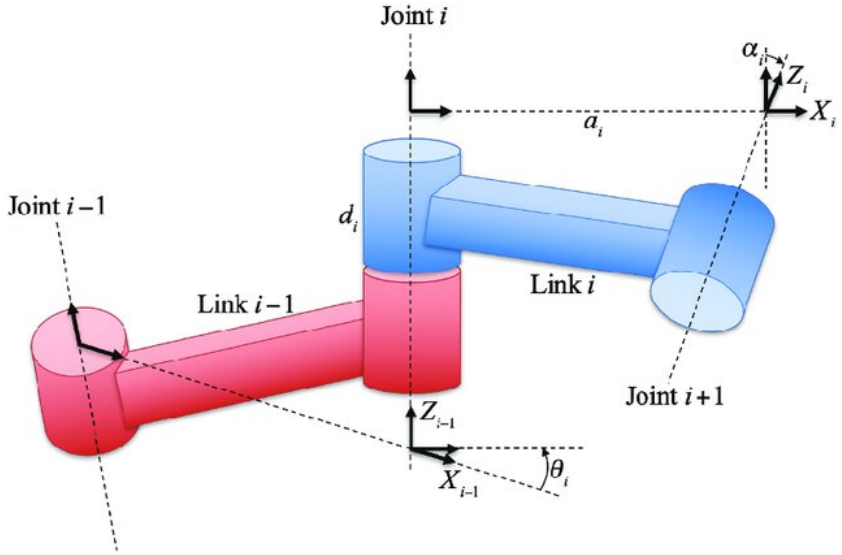


Figure 2.9: Representation of D-H parameter[9]

Here, Joint angle (θ_i) and joint distance (d_i) are known as joint parameters.

- Joint angle (θ_i) is the rotation about (z^{i-1}) needed to make axis (x^{i-1}) parallel with axis (x^i).
- Joint distance (d_i) is the translation along z^{i-1} needed to make axis x^{i-1} intersect with axis x^i.

Link length (a) and Link twist (α) are known as link parameters.

- Link length (a_i) is the translation along x^i needed to make axis z^{i-1} intersect z^i.
- Link twist (α_i) is the rotation about x^i needed to make z^{i-1} parallel with axis z^i.

If the values of joint variables $q_1, q_2, q_3, \ldots\ q_n$ are known (using D-H Algorithm), then the end effector's location (position and orientation) with respect to base frame can be found by:

$${}^0T_T = {}^0T_1 * {}^1T_2 * {}^2T_3 \ldots\ *{}^{n-1}T_n \quad \ldots\ldots (2.15)$$

Where,

${}^0T_1,\ {}^1T_2,\ {}^2T_3,\ \ldots,\ {}^{n-1}T_n$ are individual transformation matrices of the adjacent links which is given by:

${}^{n-1}T_n$ = Rot(θ_i , Z) * Trans(d_i , Z) * Trans(a_i , X) * Rot(α_i , X)

Here, Rot(θ_i , Z) represents the fundamental rotation matrix obtained from the rotation of mobile frame by an angle θ_i about the Z axis of reference frame.

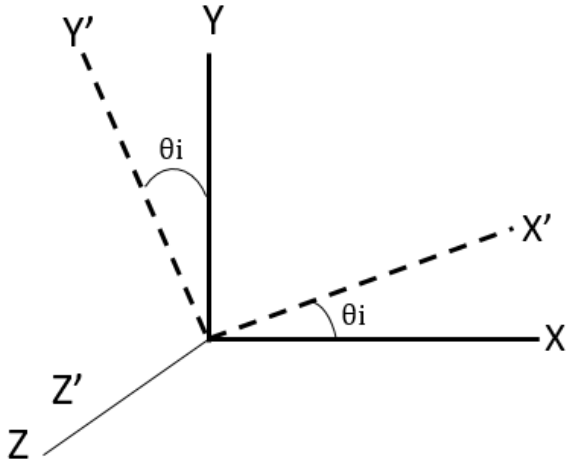


Figure 2.10: Rotation of mobile frame by an angle θ_i about the Z axis

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

As shown in figure, if the mobile axes is rotated by an angle θ_i about fixed reference frame Z axis, then the above transformation matrix helps to transform the coordinates w.r.t. the mobile frame into the coordinates w.r.t. the fixed reference frame. Similarly, other expressions in the equation below have the similar corresponding meaning.

$^{n-1}T_n$ = Rot(θ_i , Z) * Trans(d_i , Z) * Trans(a_i , X) * Rot(α_i , X)

Thus, by multiplication, we get the following matrix:

$$^{n-1}T_n = \begin{bmatrix} \cos(\theta i) & -\cos(\alpha i) * \sin(\theta i) & sin(\alpha i) * \sin(\theta i) & a_i\cos(\theta i) \\ \sin(\theta i) & \cos(\alpha i) * \cos(\theta i) & -\sin(\alpha i) * \cos(\theta i) & a_i\sin(\theta i) \\ 0 & \sin(\alpha i) & \cos(\alpha i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{.....(ii)}$$

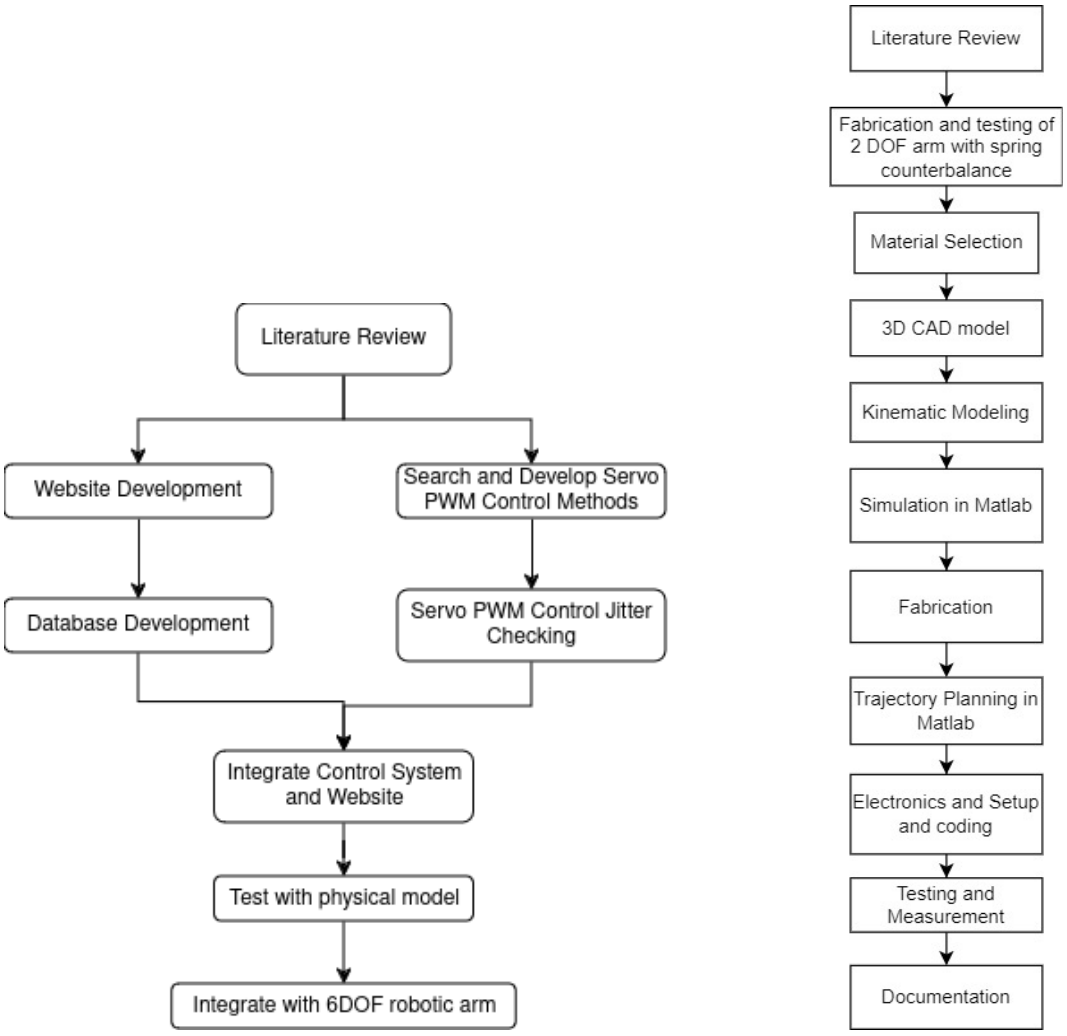
For 6 - DOF robotic arm, n=6 and hence, the transformation matrix is given by:

$^0T_6 = {}^0T_1 * {}^1T_2 * {}^2T_3 * {}^3T_4 * {}^4T_5 * {}^5T_6$ .....(2.16)

This $^0T_6$ represents the homogeneous transformation matrix . Using the matrix, we develop the forward and inverse kinematic model of our robotic manipulator.

CHAPTER 3: METHODOLOGY

Our objective of building a six DOF robotic arm follows the general outline as follows:



(a) (b)

Figure 3.1: FlowChart representing Project Outline

It is divided into two major segments. One is to test the counterbalancing mechanism for robotic arm and another is to prepare the control system for the robotic arm.

3.1 Fabrication and Testing a 2-DOF Experimental arm for Counterbalancing

A 2-degree-of-freedom (2-DOF) robotic arm was initially designed in SolidWorks and subsequently fabricated to assess the functionality of the proposed Counterbalancing Mechanism (CBM).

Figure 3.2: 2 DOF CBM Test Robot

All the essential data required for calculating the torque provided by the Counterbalancing Mechanism (CBM) spring were collected through a variety of methods. Required data included: Link lengths, the center of gravity (CG) of each link, values of a and b (as explained above), weight of each link, payload, and stiffness of spring (k). At first, we determined the gravitational torques acting at each of the joints. To counterbalance these gravitational torques, a spring of proper stiffness k has to be chosen.

The exact values of spring stiffness to be used in CBM 1 and CBM2 for proper counterbalancing were calculated. But, in real scenario, it is difficult to find spring with the exact stiffness as calculated. The springs that we have used in our test robot has values:

k_1 (used) = 581.1818799 N/m

k_2 (used)= 225.2337424 N/m

These values represent the average spring stiffness calculated from the conducted experiments. Using these values of k_1 and k_2, the counterbalancing torques at the respective links were found. Since, the counterbalancing torque found were greater than the gravitational torque due to the higher values of k_1 and k_2, so in order to make the two forces

equal an extra weight has to be added at link 1 and link 2 to compensate for the higher spring force.

Analysis

Upon analyzing the calculations, it is evident that the gravitational torques at the joints have been effectively diminished. Without the implementation of the Counterbalancing Mechanism (CBM), gravitational torques of 1.811907 Nm and 0.426735 Nm would act at joints 1 and 2, respectively. These torques would necessitate the use of high-power, expensive motors or intricate reduction mechanisms, thereby elevating the overall cost of robotic arms and escalating energy consumption in large industries.
However, with the application of the CBM mechanism, the torques at joints 1 and 2 are significantly reduced to 0.635 Nm and 0.258 Nm, respectively. This reduction not only alleviates the load on motors but also enhances the cost-effectiveness of the robotic arm. By mitigating the need for high-power motors and complex reduction mechanisms, the CBM contributes to both cost savings and increased energy efficiency in industrial applications.

3.2 Control of Servo motors using Raspberry Pi

Rotation Angle of Servo Motors is dependent on the duty cycle of PWM signal on its control pin. By varying the length of HIGH pulse (in seconds) the angle of the motor can be controlled. Raspberry Pi can be utilized to produce variable time-length pulses and transmitted via one of its GPIO pins.

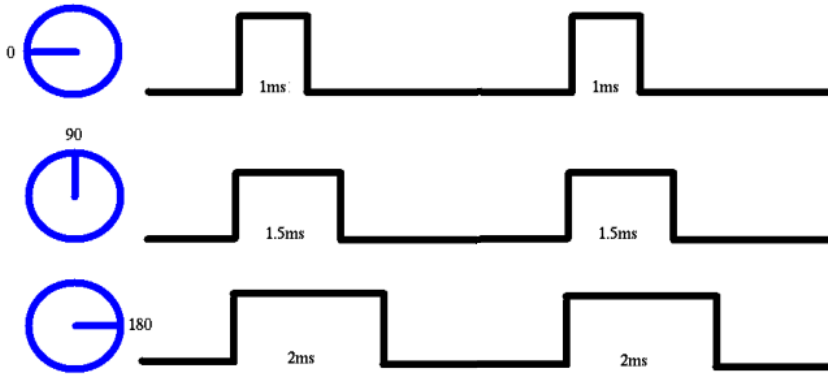


Figure 3.3: PWM duty length and Rotation Angle in servo

The use of Arduino and similar microcontrollers was discontinued because it only supported a single task thread. The limitation of using single thread is that the controller can only control single motor at a time. Thus,the use of parallel threading concept was incorporated. Parallel Work Threads utilize multiple CPU cores with each assigned its dedicated task i.e. rotating motors at each joint. Now the motors move in unison and we can hereby utilize the concepts of forward and inverse kinematics fluently. The total execution time for single work thread is the sum of time duration of each task but the execution time for parallel thread is the time duration of the longest event.

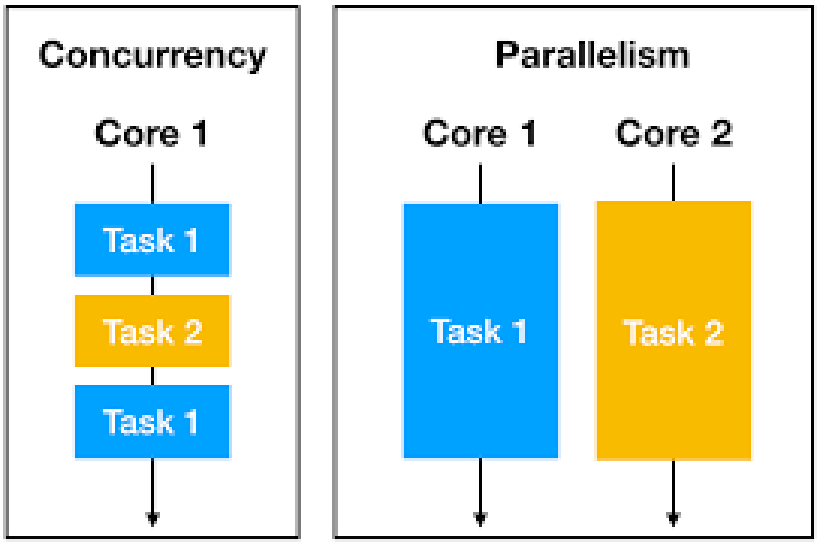


Figure 3.4: Single Thread vs Parallel Threads [3]

We use python language with additional module such as pigpio, RPi.GPIO to generate pulse, threading to generate pulse parallely in each pins. For six different motors we require 7 parallel threads (with main thread).

3.3 Website Development

Usually HTML, CSS and Javascript are used to build a website. But since the communication between the javascript and python program requires heavy coding and error detection, we use abstracted modules such as Bokeh, Django, and streamlit. For simplicity we use Streamlit module to prepare a website with login systems and servo control mechanism.

3.4 Database Management

We can use flat files like csv,txt to serve as database for our software. In flat files the tasks of deleting and updating entries gets difficult as the size of database progresses. We utilize a commonly used database software, sqlite3. To use sqlite3 we use the module sqlalchemy.

3.5 Control and Verification

As the website and database are ready to be used, we connect it to a DIY model of robotic arm for verification and bug-detection. For this, 6-DOF robotic arm from previous projects was burrowed to test our software system. Website mentioned above consist of a page for the remote control of the arm.The page allows the end user to provide the angle of rotation of each joint and also can allow them to provide the lists of rotation angles.After they press the run motors option the robotic arm will rotate through those angles automatically. The motion of the robotic arm initially was very quick which is not wanted.In order to solve this, the smoothing function was developed:

y= A*sin(x)+B*cos(x)+C

where,

y gives the delay between the pulses

x is the angle in radians

A,B,C are constants

It might be confusing but the value of x does not correspond to the angle of the servo motor. The parameter input to the function is the no of divisions the robotic arm is

subjected to. The range of old angle to new angle is divided uniformly by 4 units or 12 units. The list count is sent to the smoothing function. Smoothing function now divides the time delay value in same no of list count. It is controlled by the value of C. The trigonometric part of the function introduces variation in delay which helps to control the speed of motor based on the values of A and B.

This smoothing function helps in slow and smooth motion of the arm.If the single value of rotations are provided, then the arm executes the motion to reach those angle.And if the list of values are given then it will start its automation feature. With python it is easier to work with the data as list or matrix and thus enables the automated control of robotic arm.

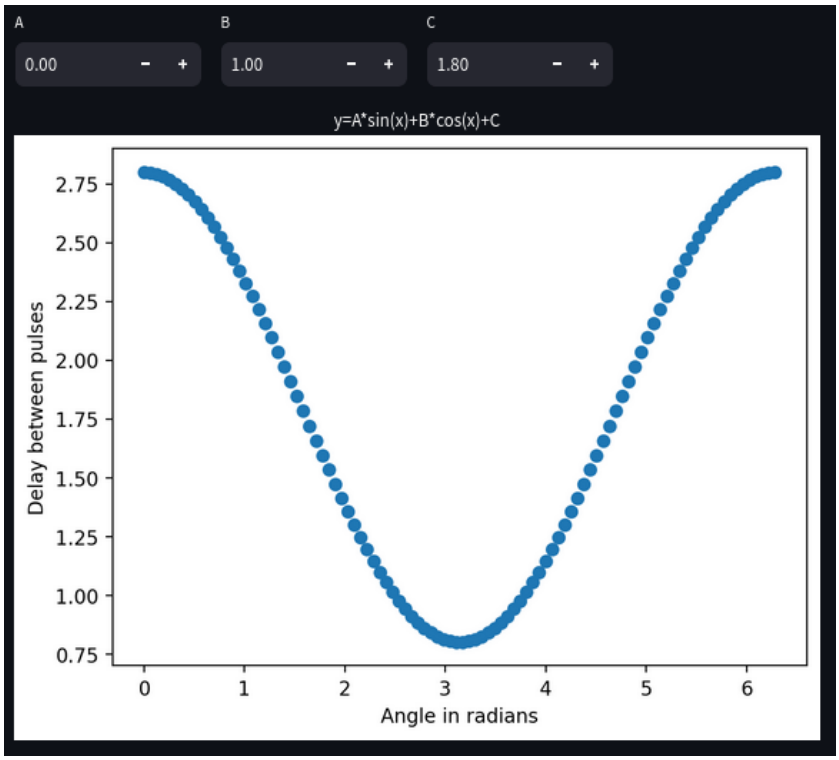


Figure 3.5: Delay between pulses vs Angle in radian

For testing and verification, a DIY 6-DOF robotic arm commonly seen in various students projects were used.

3.6 Order Placement and Tracking

Among the list of products, the user is allowed to choose the product to order and for this buy button is provided.After pressing the buy button, the order is placed and in order page, the order information for the user is seen.The order information consists of order

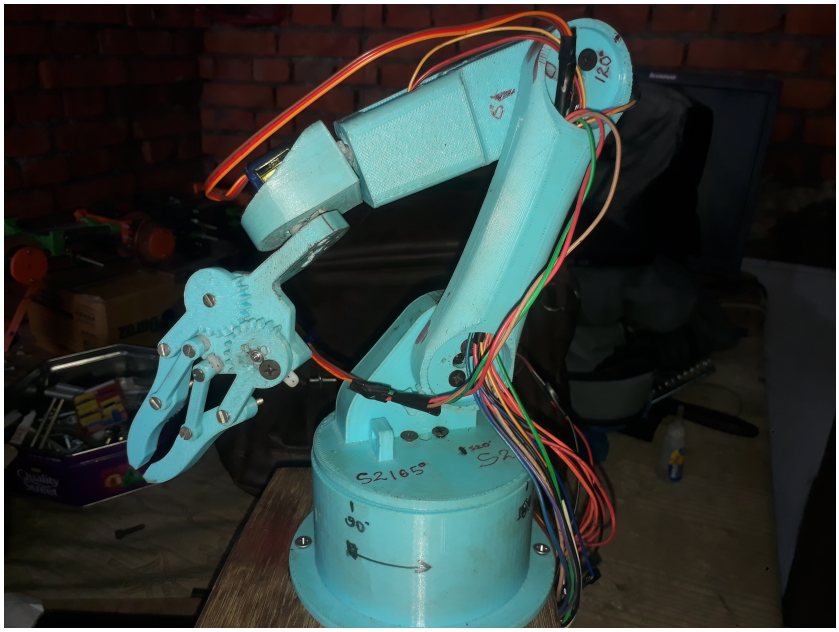


Figure 3.6: DIY 6-DOF Robotic arm for control verification

id, product ,order status.The order status allows the user to track their order.Initially, the order status shows order placed comment.After each component is produced completely,the number of product produced is updated and when the number of product produced is equal to the number of product ordered , then the status changes to order completed and when the order is set for delivery, it shows order set to delivery.This allows for the user to track their orders from the stage of order placing to order delivery state.

3.7 3D CAD model of 6-DOF Robotic Arm

While designing the 6-DOF robotic arm, the existing arm designs were studied. Afterwards, a 3D model of the robotic arm was prepared using SolidWorks. The construction of robotic arm parts started with the first axis and gradually went to higher levels. This sequential approach helped us to analyze each assembly directly after its construction and through different iterations we connected and aligned each part. The physical design has to be adjusted to facilitate weight reduction and counter-balancing. Therefore, the design underwent multiple iterations, each carefully calibrated to align with the manufacturing capabilities of the machinery within our reach. The final CAD design of our robotic arm is shown in figure 3.3:

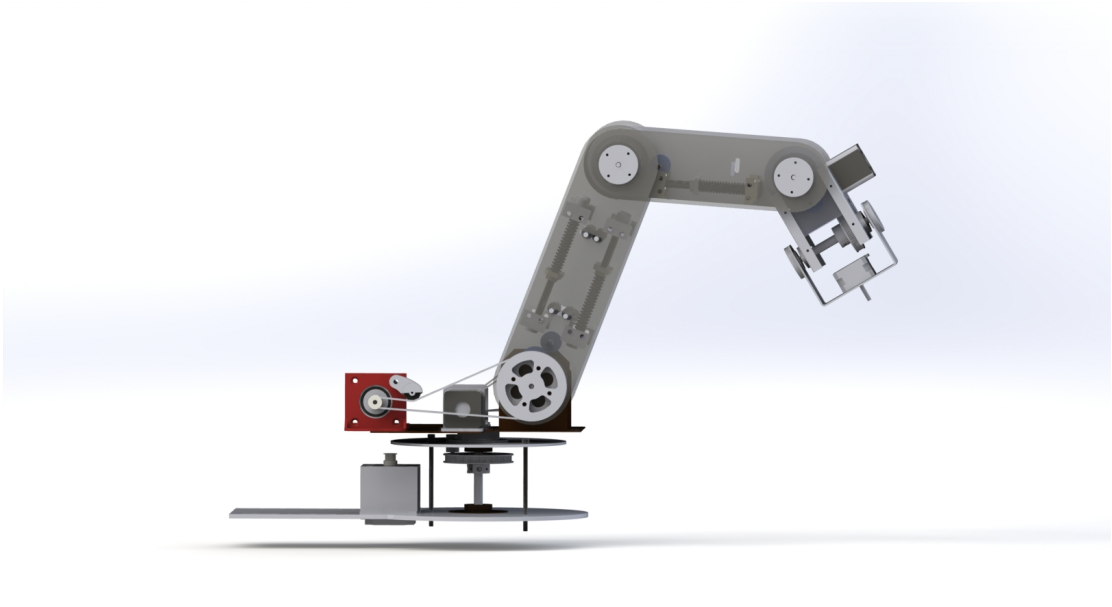


Figure 3.7: 3D CAD model of 6-DOF Arm

3.7.1 The Base

Designing the base proved to be the most formidable challenge. The imperative to diminish gravitational torques necessitated positioning the three motors to drive the shoulder, elbow, and wrist at the base, ensuring the stability of our robotic arm. After numerous iterations on the base design, we ultimately devised a compact circular base with base motor and three motors for shoulder, elbow and wrist with belt and pulley as transmission system as shown in figure 3.4, meticulously placing all components. However, during the base design process, space constraints prevented us from integrating idlers for belt tensioning.

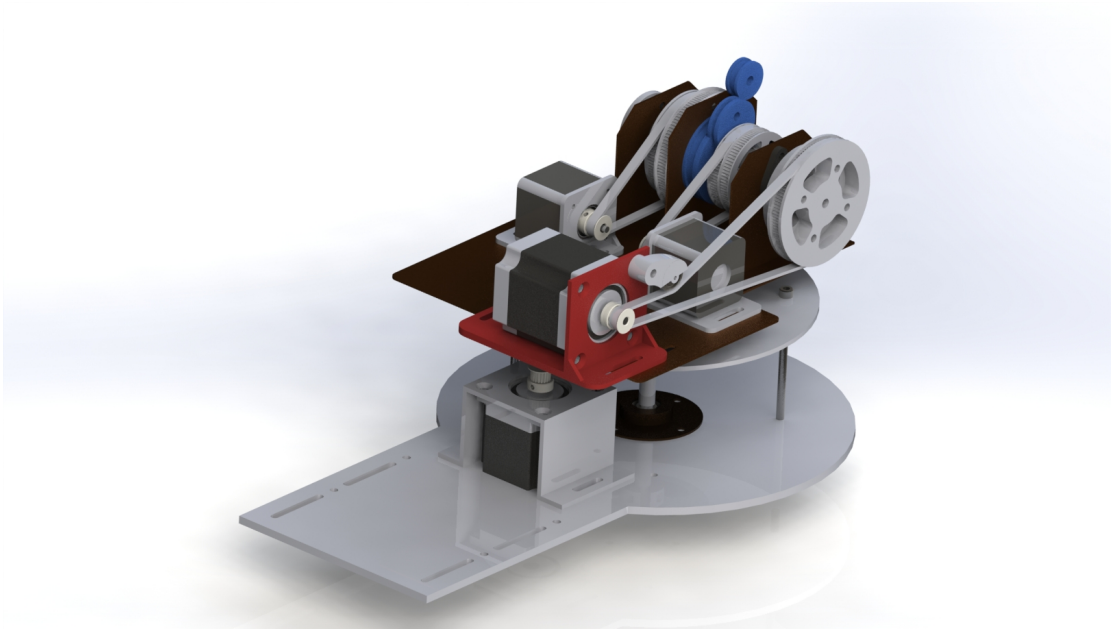


Figure 3.8: Base

3.7.2 The Shoulder

The shoulder assembly incorporates a spring counterbalance mechanism for both the shoulder and elbow, employing a double parallelogram mechanism with ropes and pulleys. The counterbalance mechanism comprises springs, reference pulleys, sliders, belts, and idlers. To maintain the reference plane, a reference pulley is strategically placed to ensure the consistency of each link's reference regardless of the position of other links. The finalized CAD model of the shoulder assembly is illustrated in the figure 3.5.

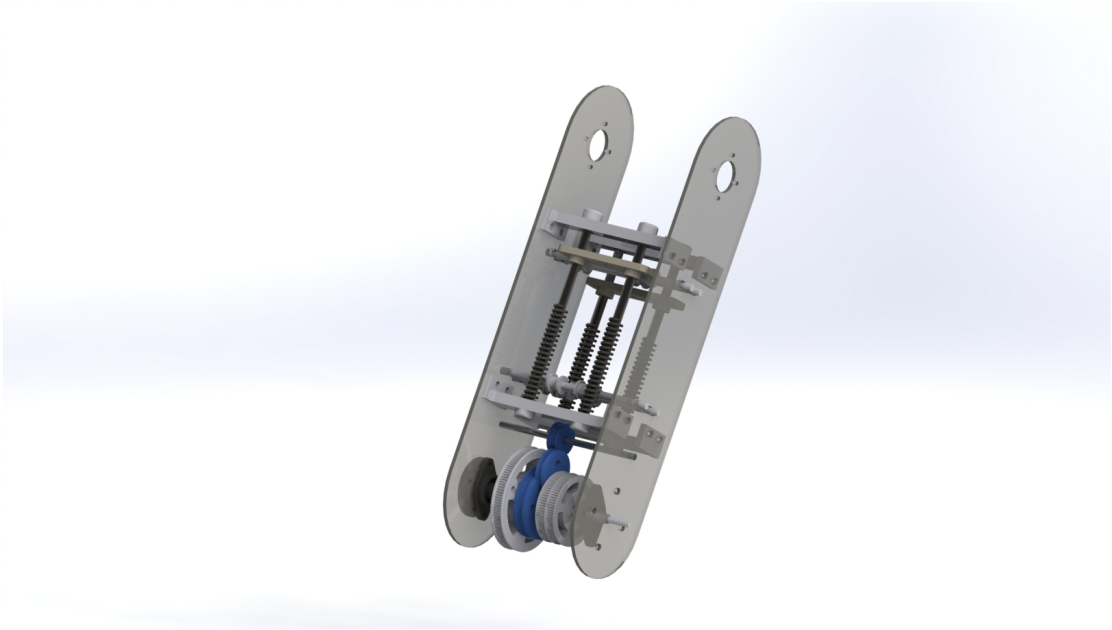


Figure 3.9: Shoulder

3.7.3 The elbow

Similarly to the shoulder, the elbow is equipped with a spring counterbalancing mechanism for the wrist, employing a double parallelogram mechanism with springs, ropes, pulleys, and idlers. The comprehensive CAD model of the elbow assembly is showcased in Figure 3.6.

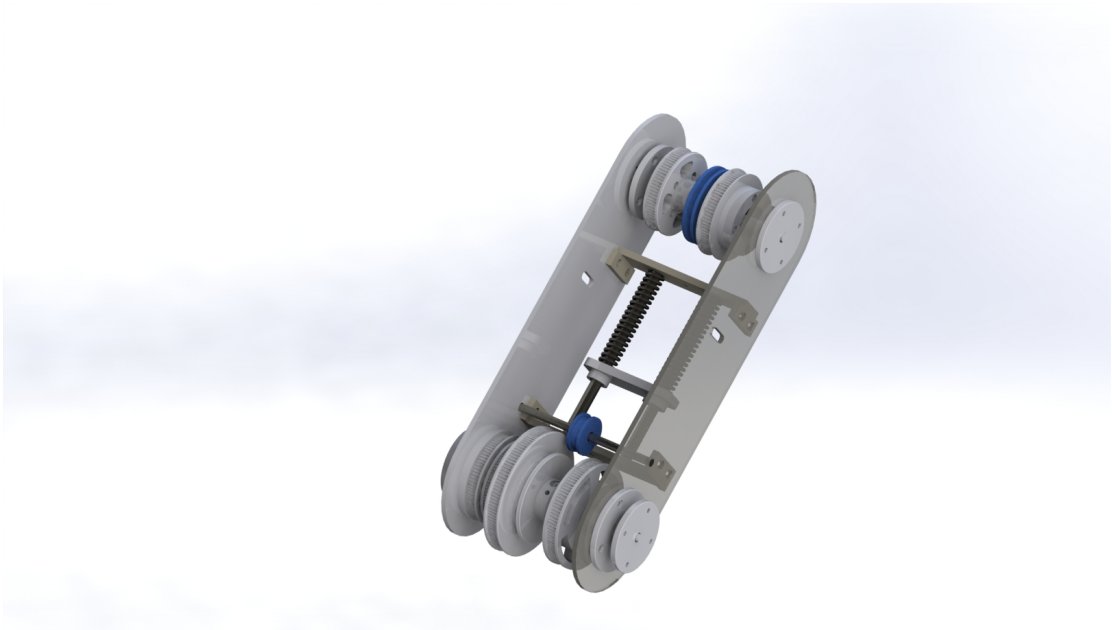


Figure 3.10: Elbow

3.7.4 The wrist assembly

Figure 3.7 depicts the CAD model of the wrist assembly. This assembly facilitates the mechanism for yaw, pitch, and roll motions of the robot, each governed by dedicated actuators. As previously mentioned, the actuator responsible for pitch motion is positioned at the base to mitigate gravitational torques. The remaining two actuators, responsible for yaw and roll motions, are situated at the wrist, as illustrated in Figure 3.7. Additionally, the wrist is outfitted with an end effector, which can be customized to meet the user's requirements for various tasks.

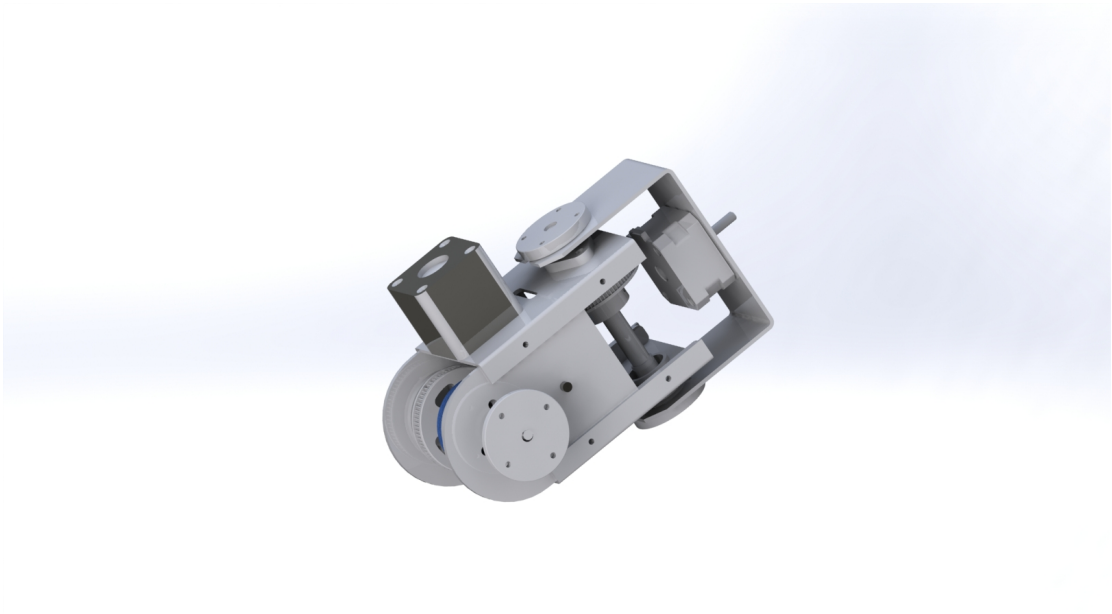


Figure 3.11: Wrist Assembly

3.8 Kinematic Modeling

To construct the forward and inverse relationship between the joint variables and Cartesian coordinates, it is necessary to develop a kinematic model based on the geometry and configuration of the arm design.The kinematic model of the arm was established utilizing Denavit-Hartenberg parameters, and mathematical solutions for both forward and inverse kinematics were derived.

3.9 Simulation in MATLAB

The developed kinematics model underwent thorough testing within MATLAB. Verification of the forward and inverse kinematics solutions was conducted using Peter Corke's toolbox. Additionally, the robot's workspace was meticulously determined and analyzed.

3.10 Fabrication

Fabrication is the process of creating products,structures,or components by manipulating raw materials through various technique and processes.Key aspects of fabrication include material selection,preparation,shaping,assembly and finishing. For the material,acrylic is chosen as the arm plates whereas mild steel is chosen as the shaft.According to the 3D cad model,the acrylic was cut using CNC laser cutter and for the shaft, it was machined in the lathe machine to make it stepped as required by our design.The remaining components, including pulleys, motor mounts, and idlers, were 3D printed for their convenience, cost-effectiveness, and ease of assembly

3.10.1 Material Selection

Material selection is a crucial aspect in the design and engineering of various products, structures, and systems. The choice of materials can significantly impact the performance, durability, cost, and environmental sustainability of the final product.

Components/Materials to be used:

- Acrylic (5 mm) – For side plates

- MS (3 mm) – For rotating base and base of the robot.
- Aluminium (3mm) - For End Effector
- PLA - For 3D Printed Parts (Pulleys, Sleeves, Mounts)

** May need to add additional weights to lower the arm's C.G.

For the frame/body of the robotic arm, three materials were considered namely PLA, acrylic, and aluminum. However, in the end, acrylic is selected because of its following advantages:

- Acrylic is easy to cut, shape, and fabricate. It can be laser-cut or CNC-machined with ease, allowing for intricate designs and custom shapes.
- Acrylic is relatively lightweight.
- Acrylic is generally less expensive and easily available than aluminum, making it a cost-effective option.
- Acrylic is transparent, which makes robots aesthetically beautiful.

For shafts, Mild steel is selected because of the several properties that make it suitable for shafts in various contexts such as cost-effectiveness, machinability, ductility, moderate strength, and availability.

3.10.2 Bearing selection

The bearings used in the arm will have to deal with radial forces and not allow misalignments. Deep groove ball bearings emerge as the most fitting choice for the task, effectively restricting the necessary degrees of freedom, offering straightforward installation, being economically priced, and readily accessible. 607, 608 and MR695 deep groove ball bearings are used. 607 bearing has the bore diameter of 7mm, outside diameter of 19mm with 6mm width. 608 bearing has the bore diameter of 8mm, outside diameter of 22mm with 7mm width whereas MR695 bearing has a bore diameter of 5mm, outside diameter of 13mm and width of 4mm.

The thrust bearing(51105) is used in the lower base where the axial load is dealt with. The reason for choosing the thrust bearing is that it is readily available and is less

expensive. 51105 dimensions: 25 mm ID and 42 mm OD, 11 mm width.

For the fabrication process, initially parts for the base were gathered and assembled according to our design.Following base,shoulder,elbow,wrist all were assembled separately and finally these sub-assemblies were assembled using fastening methods such as screws,bolts and more advanced techniques such as welding.The fabrication process also includes the setup for our counterbalance mechanism involving spring-wire system as well as power transmission system. In our design, a combination of belts and 3D-printed timing pulleys are used for the power transmission. For the base, the motor is placed at the bottom which drives the smaller pulley, belt, and then the larger pulley giving the transmission ratio of 1/6, thereby increasing the torque of the output shaft by 6 times and decreasing the speed by 6 times. The motors for driving the axis 2,3 and 4 are mounted on the rotating base and a belt-pulley system is used for the power transmission to those axes. The reason for mounting the motors on the base is to reduce the weight thus decreasing the gravitational torque.

3.10.3 Transmission Components

GT2 belt

GT2 stands for ”Gates Tooth 2mm” and is a specific type of timing belt with a 2mm pitch between the teeth. It is known for its precision and is commonly used in applications where accurate positioning and timing are crucial. The GT2 timing belt system often includes a belt, pulleys with GT2 profiles, and sometimes tensioners. [12] Generally, it is available in two different widths:
6 mm: It is most commonly used in 3D printers. Pulleys are also designed to work with these belts.
10 mm: It is less common and used when machines are typically larger and when there is more force dealing with in the moving axis.
A 6 mm GT2 belt is used in our design which drives all the arms of our robotic arm.

3D printed pulleys

For our design, 3D printed pulleys are used because they allow for easy customization of pulley designs,provide a quick and cost-effective way to produce prototypes,and 3D

printing allows the use of lightweight materials decreasing the overall weight of the body.

Actuators

Six stepper motors are used in total as the actuator for the robotic arm. The first, second axis uses the JK57HS51-2804 NEMA 23 stepper motor which is driven by a TB6600 motor driver. It has a holding torque of 10.1 kgfcm. The second and third axis is driven by Nema 17 motor with a holding torque of 4kgfcm. It is driven by A4988 stepper motor driver. The fifth axis uses 17PM-J349-P1VS NEMA 17 stepper motor which is driven by the A4988 motor driver and the 6th axis uses NEMA 17 pancake stepper motor.

3.11 Electronics setup and coding

For electronics setup, one stepper motor(NEMA23) is clamped at the lower part of base along with TB6600 motor driver.The power box and raspberry pi is also placed at the bottom part

3.12 Testing and Measurement

Different tests were performed to see whether the operation of robotic arm could be obtained as desired or not. Initially, we conducted tests to assess the functionality of the spring balance counterbalance mechanism by designing and fabricating a 2 DOF manipulator. Subsequently, we observed a significant reduction in gravitational torques at all three joints.Finally, incorporating this system in the final 6 DOF model, torque test were done.

CHAPTER 4: RESULT AND DISCUSSION

4.1 Website and Order Database

The website was built using the streamlit module which is the popular open-source Python library used for building interactive web applications. For database, sqlalchemy module is used which uses ORM(Object-Relational Mapping) which abstracts SQL language and thus enables use of python code to access the sql database.

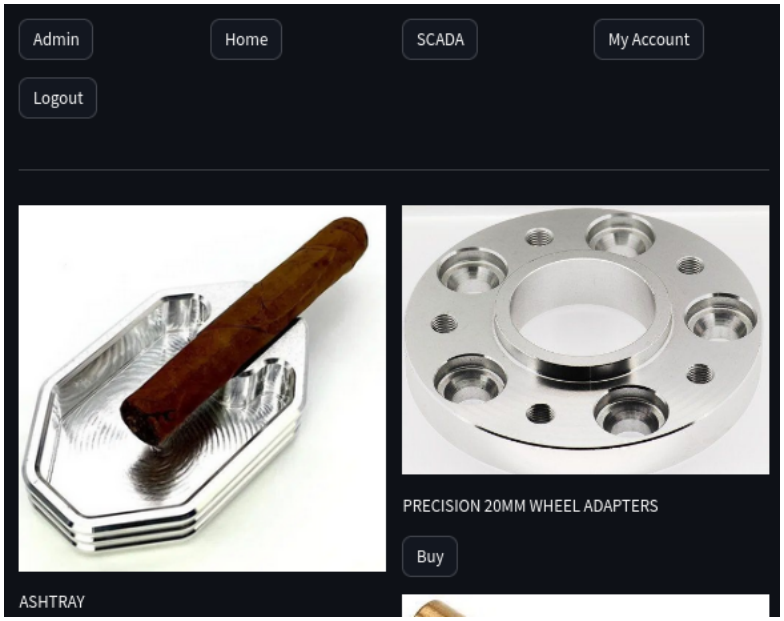


Figure 4.1: Items listed in website

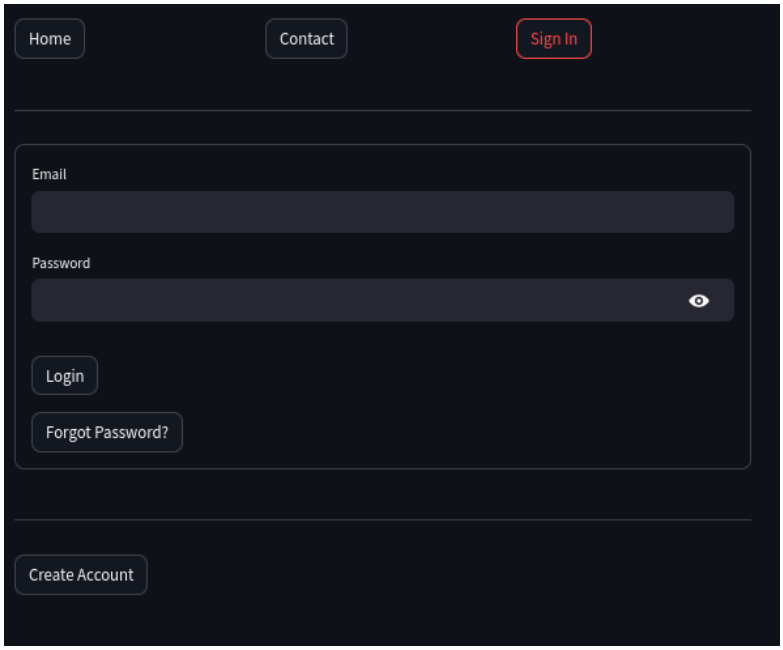


Figure 4.2: Login required to access product information

In addition, if the account is admin, he/she will be introduced to a different interface that involves page for robot control debugging.

After the user logs in the system, the user will be shown the variety of products in the homepage as shown below:

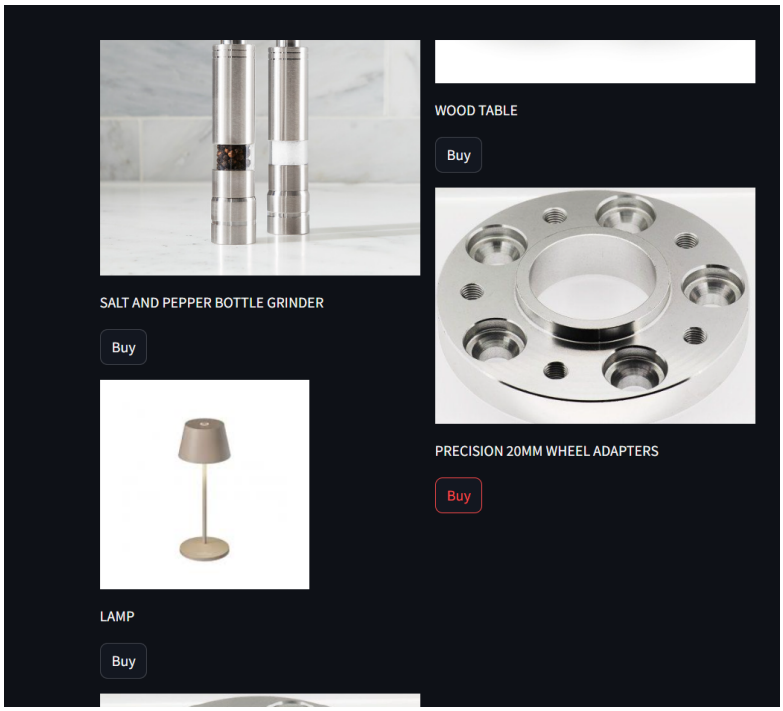
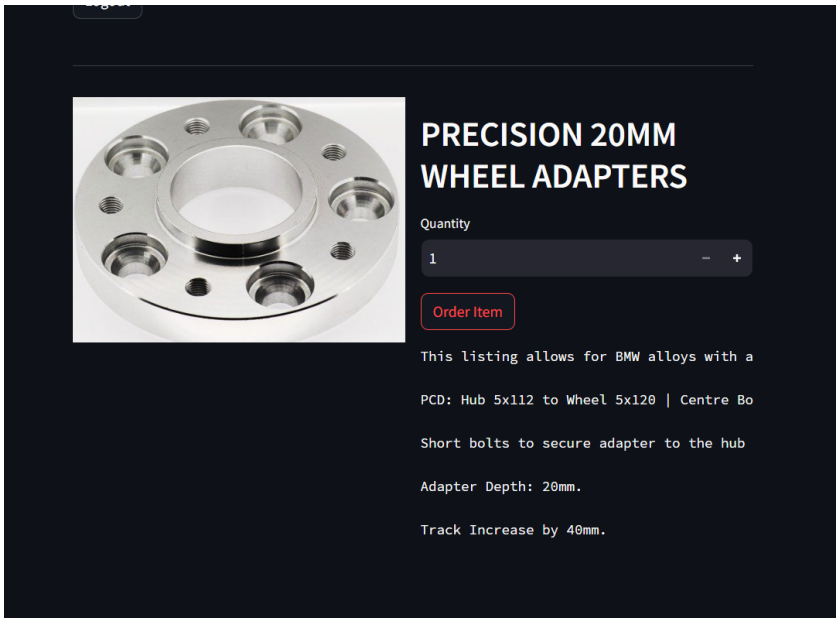


Figure 4.3: Product Display in website

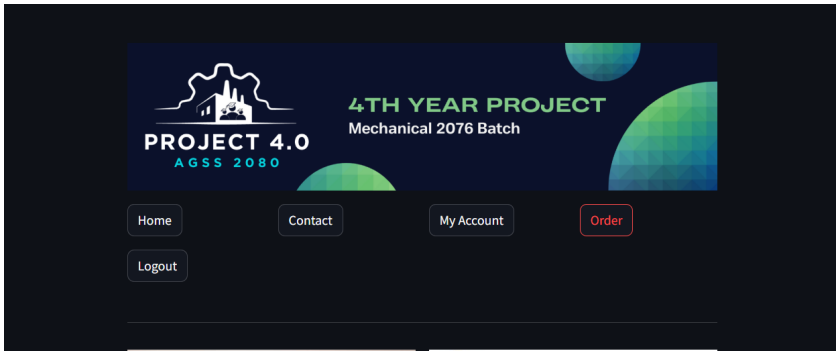
And when the user presses the buy button (the red in buy button shows that the user

pressed the buy button) the user will be directed to the product description page where the user can provide the quantity to order which is shown below:



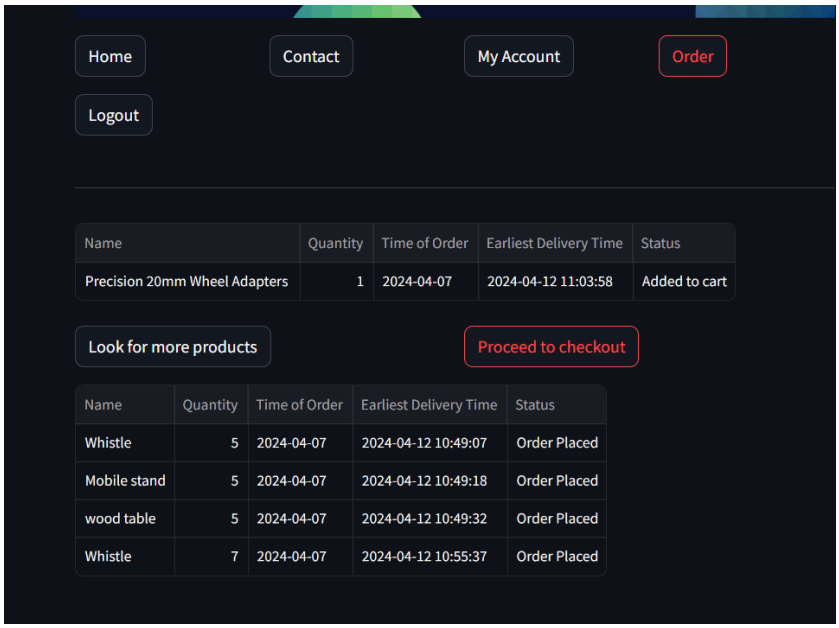
Product Description and Order

After specifying the quantity and pressing the buy button, the order will be placed and will be directed to the home page.



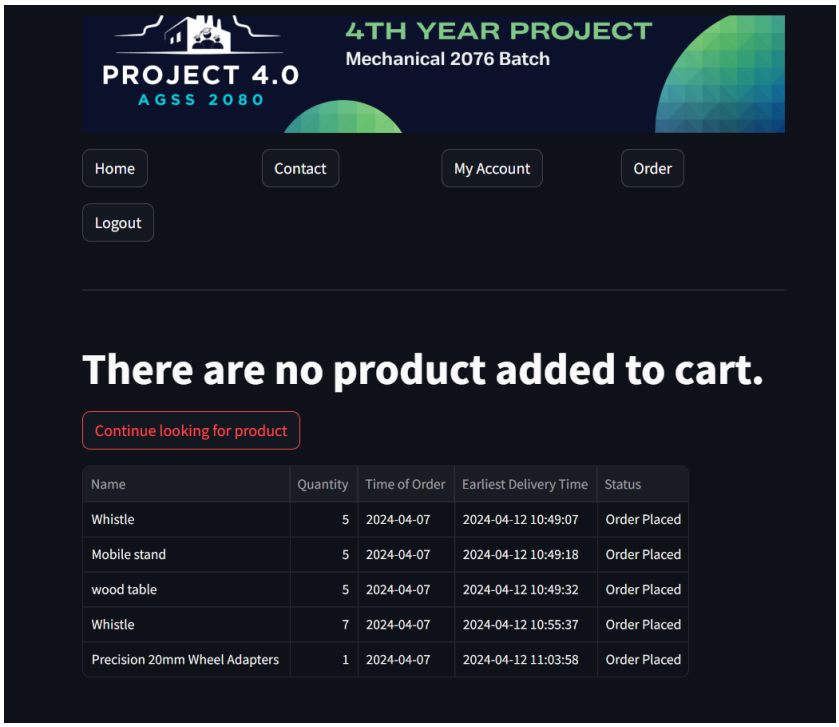
Home page

In order to view the orders and their description, users are allowed to access the order page.The order page allows the user to track their order.



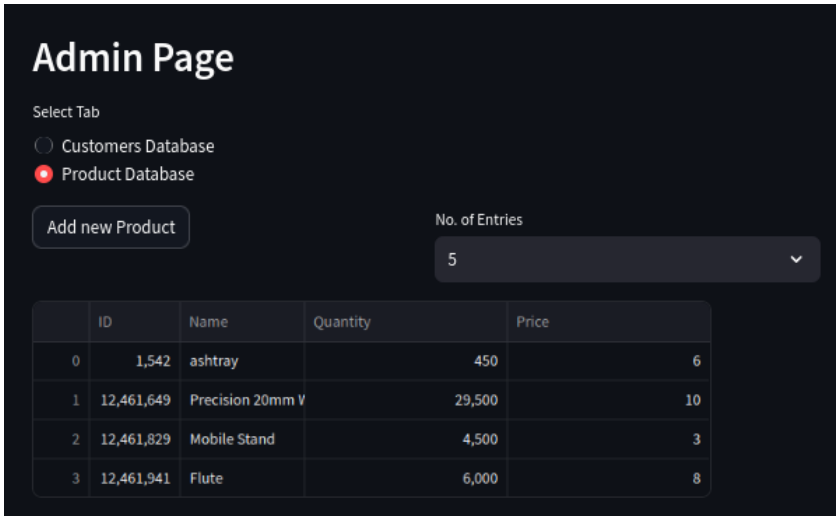
Order page with the descriptions

After pressing the proceed to checkout button, the order will be shifted from added to cart to order placed status and will allow the user to check for more products if desired.



Shifting of order from added to cart status to order placed

This can further be checked and each product be assessed via admin page. It offers control over customer and order database.



4.2 Robotic Arm Control

For arm control, the program initiates 6 parallel thread each waiting from main thread for task. The task sent is usually the value of rotation. Here from the main website, the control of each robot can be performed by varying values using the slider.
This can only be accessed by admin only that too one at a time. It is utilized for debugging rather than in operation.
It offers the following functions.

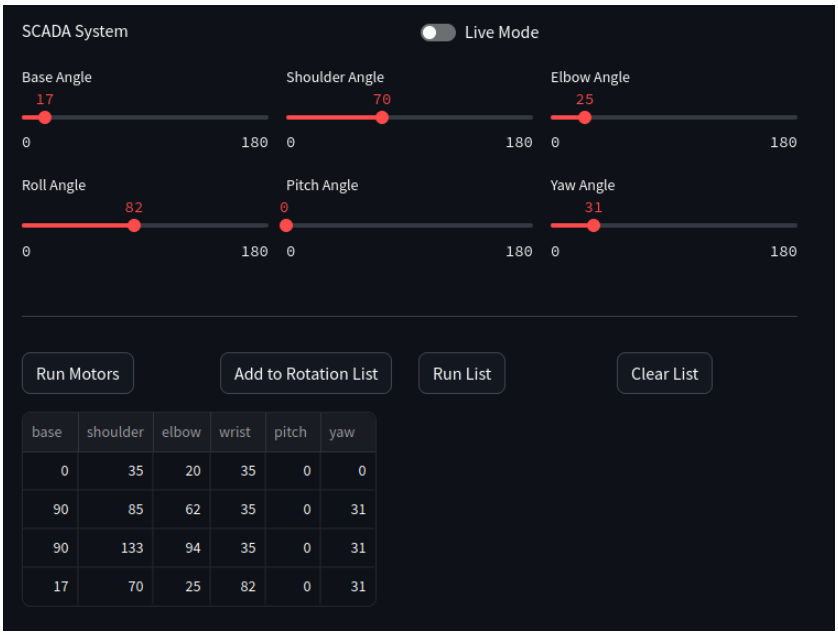
- Normal Mode: In normal mode, the signal will be sent only when the ”Run Motor” is pressed. If the angle is changed via a slider, the arm will move to the new position as specified.

- Live Mode: In live mode the angle set in slider will be directly sent to the servo motor without having to hit the ”Run Motor” button.

- List Mode: In list mode, we give it a set of coordinates it have to pass through. This can be given by using Normal or Live mode to update the list with new angles.

  By using ”Run List” to run the motor as directed and to clear the list ”Clear List”, the robotic arm can be controlled to follow a designated path.

With this, by giving angular coordinates to each parallel thread, the robotic arm was moved in a specified order. The angular coordinates can either be input manually or provided as a list by any inverse kinematic algorithms.



4.3 Output (Related discussion)

The fabrication of 6 DOF robotic manipulator brings up major challenges in torque handling, actuator selection and power supply.To hold the own weight and weight of payload,actuator of high torque capacity is essential.In addition, to accelerate the arm further adds to the torque capacity required in our actuator. Thus, an actuator with such a large torque is very much expensive. To encounter this problem, we went through intensive research on how to reduce the size of actuator required. Fortunately,a principle named 'Spring Counterbalance Mechanism'could do this job. This principle based on double parallelogram principle is able to counterbalance the gravitational torque of the robotic arm at any angle to the horizontal as described in the report. The weight of arm and payload is balanced by the spring while the actuator only requires to accelerate the arm. This significantly reduces the required torque capacity of the actuator.

Figure 4.4: Prototype of 6 DOF robotic arm

Figure 4.5: Spring CBM for links 2 and 3

4.3.1 2 DOF manipulator CBM

The 2 DOF robotic manipulator has two links.Each link has its own CBM with a single spring mounted on the respective arm.This allows the manipulator to be light weight and simple.On experiment, following results were obtained. The total weight of the 50 cm long arm was 0.748 kg (link1 = 0.458 kg and link2 = 0.29 kg). The torque required

to hold the arm in a horizontal fully stretched position without counterbalance in the joint 1 is 18.7 kgfcm and in joint 2 is 4.35 kgfcm.

With counterbalance, the following results were obtained: Torque required to move the link1 = 6.475 kgfcm

Torque required to move the link2 = 2.625 kgfcm

With a heavy 65.37 % reduction of the required torque in joint1, the test of our counterbalance mechanism is successful. According to the research article, this required actuator torque can be reduced by 80% provided the accurate selection of spring. Thus, the torques required to operate the robot arm were greatly reduced since the gravitational torques were compensated by the counterbalancing torques from the counterbalance mechanisms.

4.3.2 6 DOF manipulator CBM

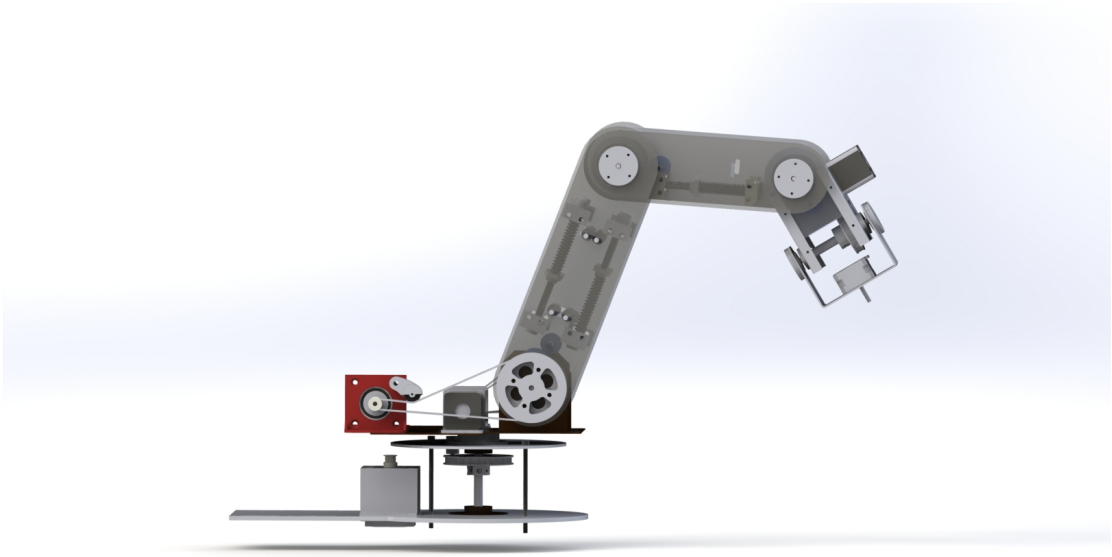


Figure 4.6: Modeled 6 DOF robotic arm

The design of the proposed 6 DOF manipulator is shown in the figure 4.1.This manipulator has six links including the base link.Link 2, 3 and 4 has pitch motions and these are the links where the actuator should work against the gravitational torque. Link 2, link 3 and link 4 are equipped with spring CBM to counter balance the gravitational torque. The CBM for link 3 is connected with link 2 and the CBM for link 4 is con-

nected with link 3. This is done due to geometrical constraints and to shift the center of mass nearer to the base. The wire is connected to the end of a pulley and the block.The wire compresses the spring when the link moves down. The actuators are on the base link to decrease the robot mass and are connected with each link to operate the robot with pulleys via the timing belt. Pulleys are installed in each joint as shown in Figure 4.2.

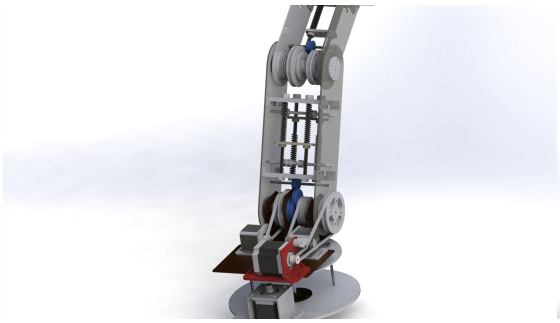


Figure 4.7: Joint Pulley Configuration

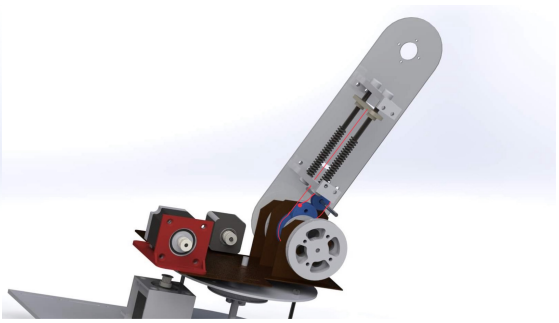


Figure 4.8: CBM in shoulder

The reference plane is maintained in the same direction to extend the multi-DOF gravity compensation. If the reference plane is not maintained in the same direction in each joint, the proper compensation can not be generated. A parallelogram four-bar linkage system is often used to maintain the direction, which is replaced by the wire and pulley. The timing belt and pulley are used to maintain the reference plane, as shown in Fig. Timing pulleys are installed in each joint, and they were connected by timing belts. Therefore, when link 1 rotates in any direction, the connected pulleys are maintained in the same direction, and each joint is able to compensate for gravity.

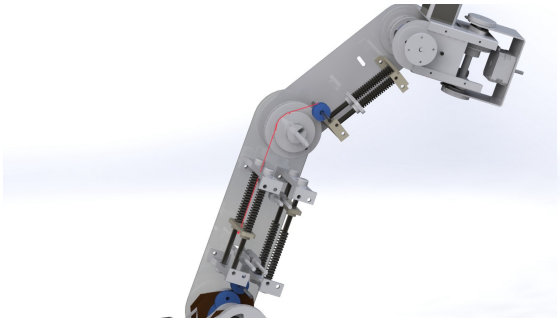


Figure 4.9: CBM in elbow

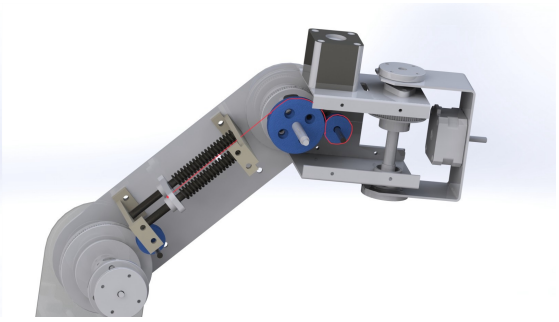


Figure 4.10: CBM in wrist

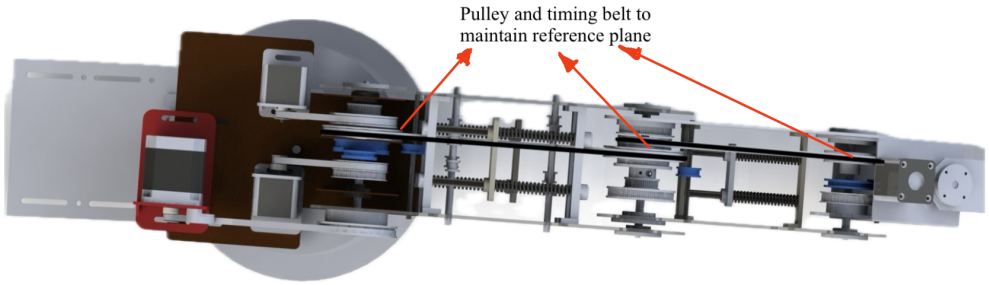


Figure 4.11: Pulley and timing belt to maintain reference plane

4.3.3 Design Parameters

The following are the design parameters for the six DOF robotic arm spring CBM:

	Design Parameters		
Joints	Joint_2	Joint_3	Joint_4
Spring	847 N/m	320 N/m	178.5 N/m
a	38 mm	30 mm	13 mm
b	60 mm	45 mm	35 mm
Mass	0.7 Kg	0.314 Kg	0.7 Kg
Length	270 mm	200 mm	160 mm
Winding	1	1	1

Figure 4.12: Design Parameters

4.3.4 Experimental Results

To verify the spring counterbalance mechanism, experiments were carried out with the prototype. The torque required for the movement of each link is determined by placing known weights on each of the link at known distance.Another method adopted was to use spring balance to determine weight needed to cause movement in the arm. Then the torque is determined by multiplying the weight by distance. The output is as shown in the figure.

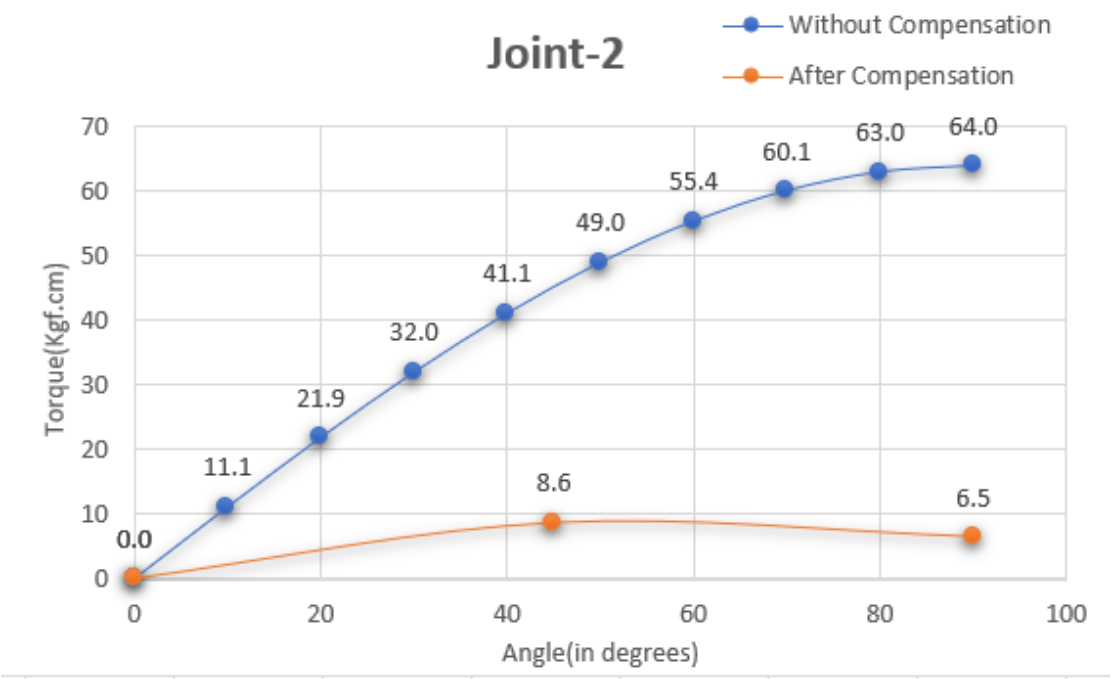


Figure 4.13: Experimental results of gravity compensation in joint-2

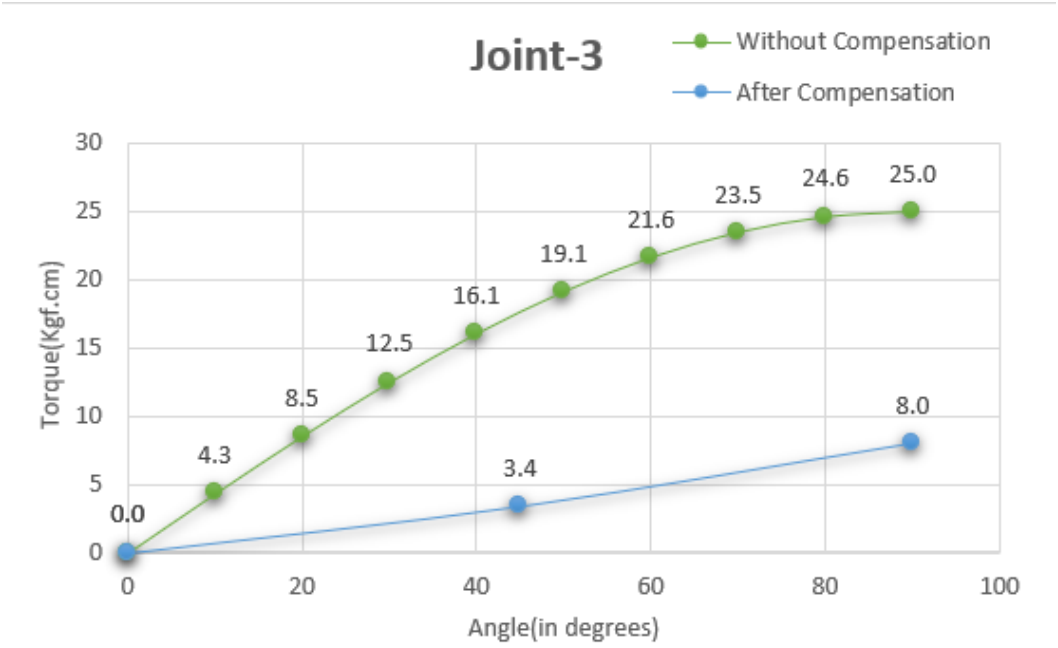


Figure 4.14: Experimental results of gravity compensation in joint-3

We see that a maximum gravitational torque of 64 kgf.cm is required at joint-2 and 25 kgf.cm is required at joint-3. After the use of spring counterbalance mechanism, the maximum required torque at joint-2 is reduced by 86.56% to only 8.6 Kgf.cm and at joint-3,it is reduced by 68% to only 8 kgf.cm.Hence, the use of CBM has significantly

reduced the torque requirement.

4.3.5 Kinematic Solution for 6 DOF Robotic Manipulator

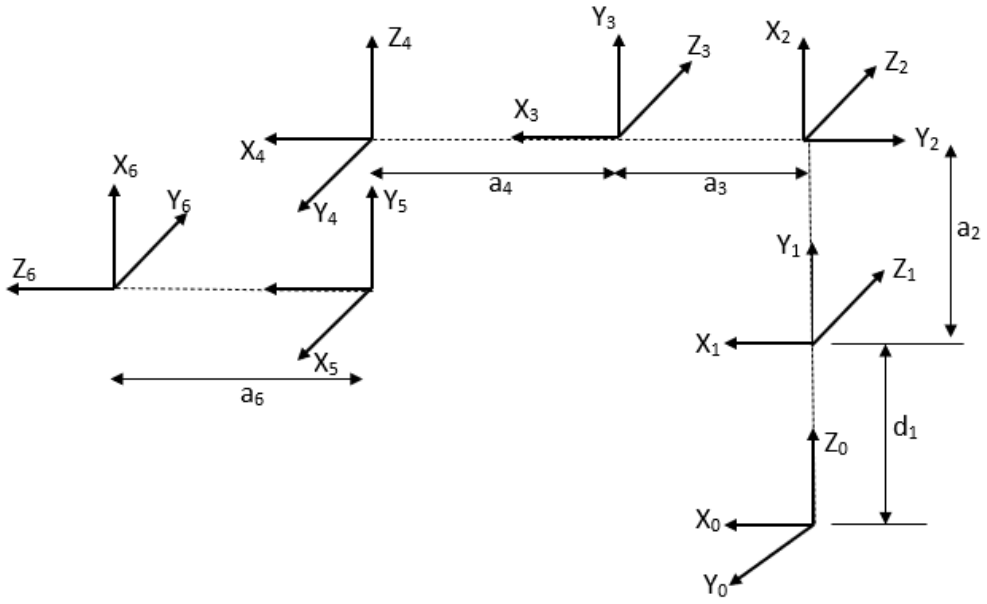


Figure 4.15: Link Frame Assignment

Table 4.1: DH-Table

Joint	θ_i	a_i	α_i	d_i
1	θ_1	0	90	d1 =100
2	θ_2	a2 = 270	0	0
3	θ_3	a3 = 200	0	0
4	θ_4	a4 = 85	-90	0
5	θ_5	0	90	0
6	θ_6	0	0	a6 = 112

Using equation (ii) in section 2.5, we obtain the values of each of the transformation matrices in equation 2.16 and multiplying them all, transformation matrix ${}^0T_6$ is obtained. This matrix can be represented as:

$$^{0}\mathrm{T}_{6} = \begin{bmatrix} nx & sx & ax & Px \\ ny & sy & ay & Py \\ nz & sz & az & Pz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where,

$$n_x = C_6 \cdot S_1 \cdot S_5 + C_6 \cdot C_5 \cdot C_1 \cdot C_{234} + S_6 \cdot C_1 \cdot S_{234} \ldots\ldots\ldots\ldots\ldots\ldots (4.1)$$

$$n_y = S_6 \cdot S_1 \cdot S_{234} - C_6 \cdot C_1 \cdot S_5 + C_5 \cdot S_1 \cdot C_{234} \cdot C_6 \ldots\ldots\ldots\ldots\ldots\ldots (4.2)$$

$$n_z = C_5 \cdot C_6 \cdot S_{234} - S_6 \cdot C_{234} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.3)$$

$$S_x = C_6 \cdot C_1 \cdot S_{234} - S_6 \cdot S_1 \cdot S_5 - S_6 \cdot C_5 \cdot C_1 \cdot C_{234} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.4)$$

$$S_y = S_6 \cdot C_1 \cdot S_5 - S_6 \cdot C_5 \cdot S_1 \cdot C_{234} + C_6 \cdot S_1 \cdot S_{234} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.5)$$

$$S_z = -C_6 \cdot C_{234} - C_5 \cdot S_6 \cdot S_{234} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.6)$$

$$a_x = -C_5 \cdot S_1 + S_5 \cdot C_1 \cdot C_{234} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.7)$$

$$a_y = C_1 \cdot C_5 + S_5 \cdot S_1 \cdot C_{234} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.8)$$

$$a_z = S_5 \cdot S_{234} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.9)$$

$$P_x = C_1 \cdot (a_2 \cdot C_2 + a_6 \cdot S_5 \cdot C_{234} + a_4 \cdot C_{234} + a_3 \cdot C_{23}) - a_6 \cdot C_5 \cdot S_1 \ldots\ldots\ldots (4.10)$$

$$P_y = a_6 \cdot C_1 \cdot C_5 + S_1 \cdot (a_6 \cdot S_5 \cdot C_{234} + a_4 \cdot C_{234} + a_2 \cdot C_2 + a3 \cdot C_{23}) \ldots\ldots\ldots (4.11)$$

$$P_z = d_1 + a_2 \cdot S_2 + a_4 \cdot S_{234} + a_3 \cdot S_{23} + a_6 \cdot S_5 \cdot S_{234} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.12)$$

4.3.6 Forward Kinematics

Given the joint variables $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$,computation of position of end effector in cartesian plane is Forward Kinematics. Equations (4.1) to (4.9) gives the orientation of the end effector and equations (4.10), (4.11) and (4.12) gives the position of the end effector in cartesian coordinate frame. By substituting the respective values of DH parameters from DH table in the right - hand side of the above equations the location (position and orientation) of the end effector can be determined.

4.3.7 Inverse Kinematics

Inverse kinematics involves the determination of joint variables q given the position and orientation of end effector. Solving inverse kinematics problems can be complex,

as multiple solutions may exist, and some configurations may be physically unreachable for the robot. The solution of inverse kinematics problem is possible if the following necessary conditions are satisfied:
Tool point must be within the workspace Tool orientation must be such that none of the joint limitations are violated. n must be greater than or equal to 6 to have any arbitrary orientation of tool.
Inverse Kinematics is the just opposite of forward kinematics. If we have the required position P_x, P_y and P_z and the orientation R matrix , then solving the 9 available equations, we can determine the joint variables.

4.3.8 Testing the Forward and Inverse Kinematic Model In MATLAB

From the DH table, the transformation matrix (Used to transform joint angles to position in coordinate space and vice versa) is determined. This matrix is also called 'The Robot Matrix' consisting of information of the position and orientation of the end effector with respect to the base frame. Therefore, it is used with MATLAB to compute the forward and Inverse Kinematics of the 6 DOF Robotic arm.
The home position of the robot is as shown in the figure below:

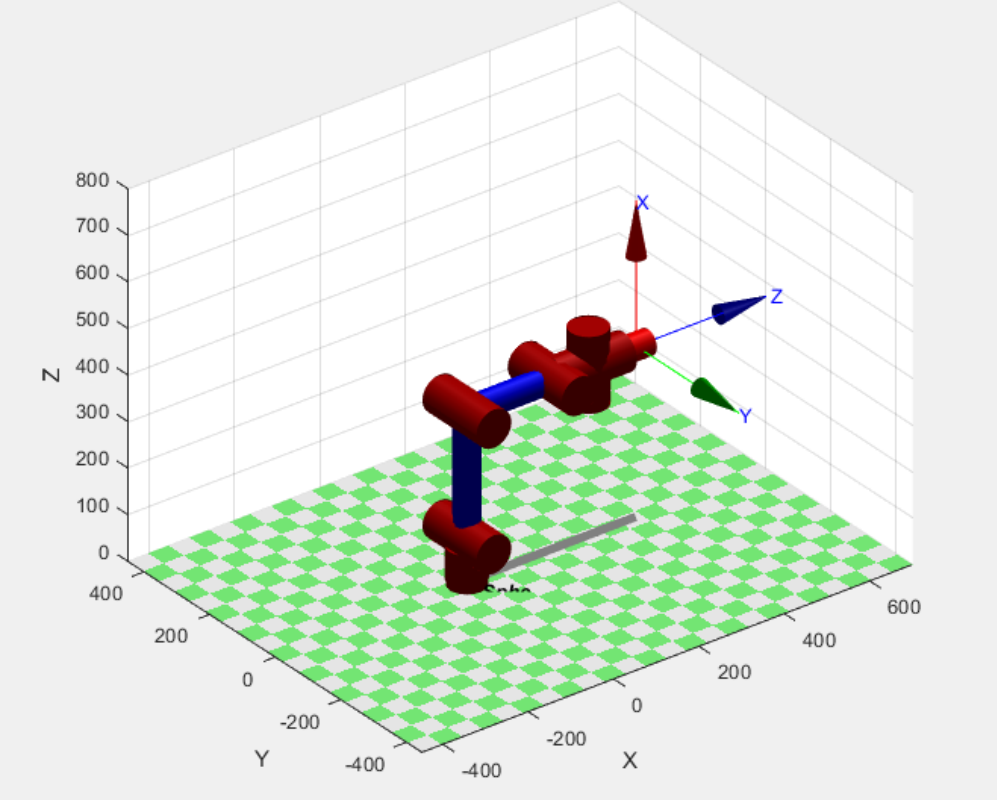


Figure 4.16: Home Position of our model

This position corresponds to joint angles (0 0 0 0 0 0) and position :
(397 mm, 0 mm , 370 mm)

Forward Kinematics

The joint angles with respect to the home position are given the following values:
$(Angth_1\ Angth_2\ \ldots\ldots Angth_6)$ = (0 30 -30 -60 0 0)

The DH parameter θ is obtained from the kinematic diagram by some manoeuvres as shown below:

$\theta_1 = Angth_1;$

$\theta_2 = Angth_2 + 90;$

$\theta_3 = Angth_3 - 90;$

$\theta_4 = Angth_4;$

$\theta_5 = \pi/2 - deg2rad(Angth_5);$

$\theta_6 = \pi/2 - deg2rad(Angth_6);$

As obtained from our Forward Kinematic Model, we have the following position vec-

tors:

$P_x = C_1 \cdot (a_2 \cdot C_2 + a_6 \cdot S_5 \cdot C_{234} + a_4 \cdot C_{234} + a_3 \cdot C_{23}) - a_6 \cdot C_5 \cdot S_1$

$P_y = a_6 \cdot C_1 \cdot C_5 + S_1 \cdot (a_6 \cdot S_5 \cdot C_{234} + a_4 \cdot C_{234} + a_2 \cdot C_2 + a_3 \cdot C_{23})$

$P_z = d_1 + a_2 \cdot S_2 + a_4 \cdot S_{234} + a_3 \cdot S_{23} + a_6 \cdot S_5 \cdot S_{234}$

Substituting the values of $\theta_1, \theta_2 \ldots, \theta_6$ in above equations, we get the following position in cartesian space:

$P_x = 163.5000$

$P_y = 0.0000$

$P_z = 163.2199$

Using the Peter Corke's Robotic Toolbox in MATLAB, the position of the end effector for the given joint configuration is found to be :

(P_x, P_y, P_z) = (163.5 , 0 , 163.22) This verifies the authenticity of the forward kinematic solution presented for our robot.

The configuration of the robotic manipulator for given joint angles is as shown:

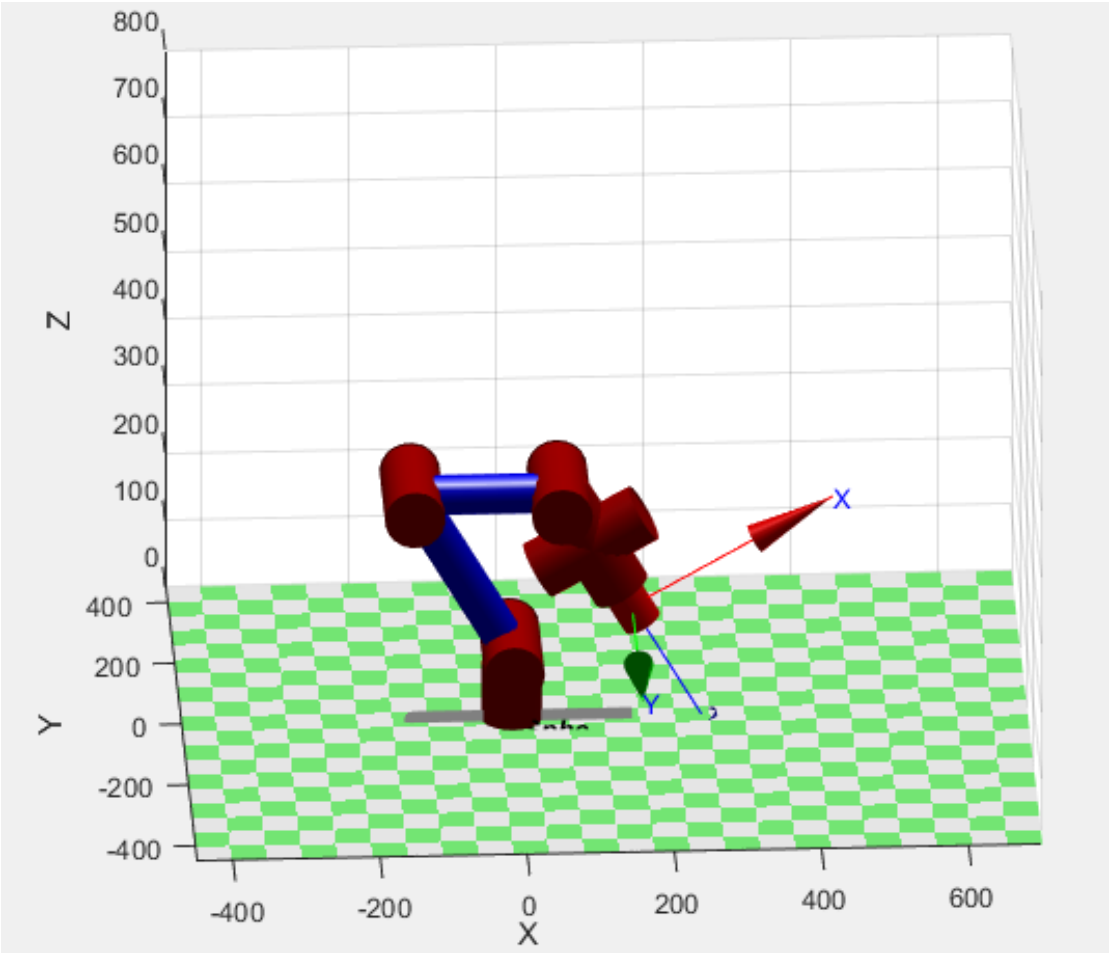


Figure 4.17: Forward Kinematics solution generated by MATLAB

Inverse Kinematics

Given the required position of the end effector in cartesian space, the process of finding the joint angles is Inverse Kinematics. Using the required position and orientation, the developed inverse kinematic model can be utilized to compute the joint angles. Below given is an example.

Given Position:$(P_x\ ,\ P_y\ ,\ P_z\ ) = (\ 200\ ,\ 200\ ,\ 200\ )$

Orientation:

$n_x\ =\ 0,\ n_y\ =\ 0,\ n_z\ =\ -1$

$s_x\ =\ 0,\ s_y\ =\ 1,\ s_z\ =\ 0$

$a_x\ =\ 1,\ a_y\ =\ 0,\ a_z\ =\ 0$

From the inverse kinematic model of our robot, the following inverse kinematic solution is obtained:

$Angth_1\ =\ 66$

$Angth_2 = -6$

$Angth_3 = -52$

$Angth_4 = 58$

$Angth_5 = -66$

$Angth_6 = 0$

(Note: Angles are in degrees. These angles are with respect to the home position of the robot. Clockwise rotation is assumed -ve)

The robot configuration as per the solution is as shown below:

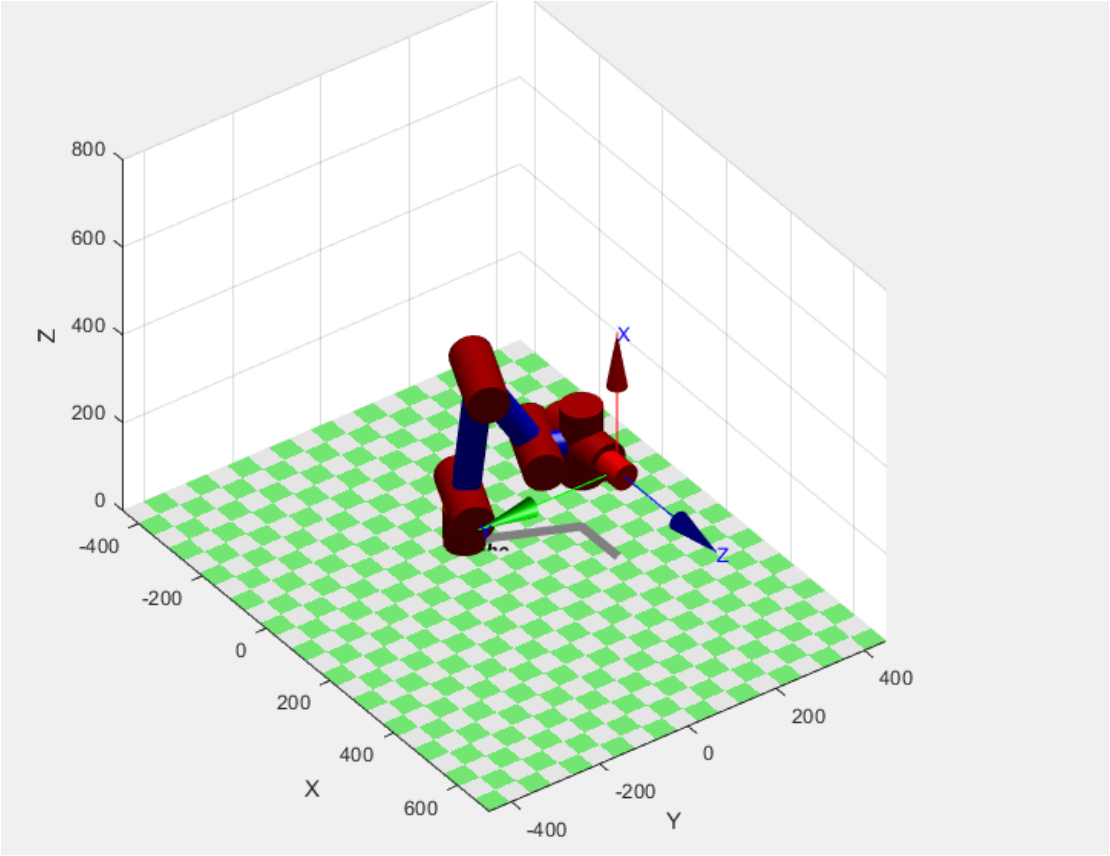


Figure 4.18: Inverse Kinematics solution of our model

Joint angles calculated by Peter Corke's Robotic Toolbox in MATLAB are as follows:

Joint Angles =(21.68 $\quad -6.34 \quad -30 \quad 0.4 \quad 90 \quad 32.3$) (in degrees)

The orientation of the robot is as shown:

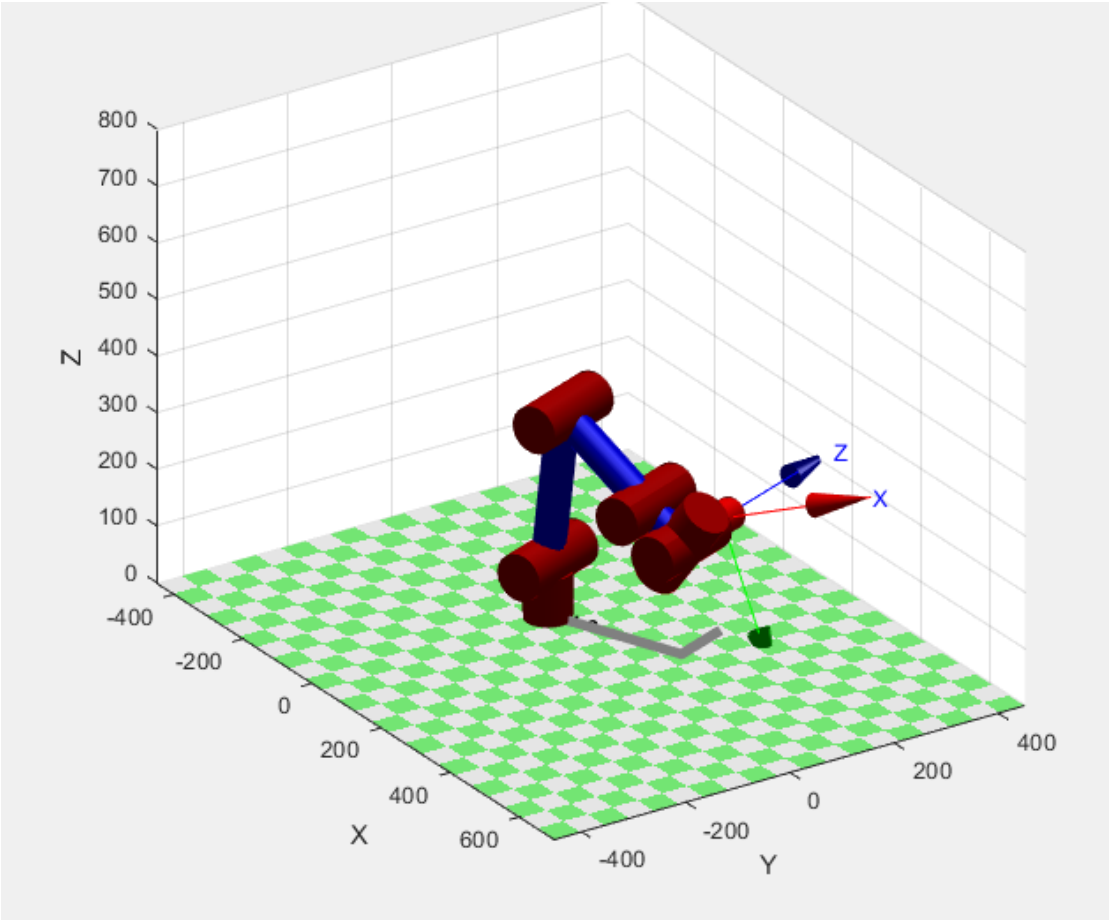


Figure 4.19: Inverse Kinematics solution generated by MATLAB

From the obtained two different solutions, we can conclude that inverse kinematics can have multiple solutions for the same position in cartesian space. This is due to multiple reasons. In our case, the reason is the different orientation of the end effector. In the first solution, the end effector's orientation is the same as that of the end effector at home position while in the second solution, the orientation is different automatically computed by MATLAB. Multiple solutions are also the result of the non-linearity of equations and non-uniqueness, singularities, geometric constraints, and joint limits.

Robot's Workspace

Robot Workspace refers to the spatial volume or region in which a robotic system, specifically the end-effector (tool or hand) of the robot, can operate. It defines the set of all possible positions and orientations that the end-effector can reach while the robot's joints stay within their specified limits. It represents the spatial envelope within which

the robot can perform tasks and is crucial for designing effective robotic systems. The workspace is influenced by the kinematic structure of the robot, which includes the number and types of joints, link lengths, and joint limits. It is a critical aspect of robotic system analysis and design, as understanding the workspace is essential for tasks such as motion planning, obstacle avoidance, and determining the reachability of specific points in the environment. Using MATLAB, 10000 samples of random joint angles within the joint limits are generated. Each sample is a (6*1) matrix that consists of six joint angles. The corresponding position of end effector for these joint angles are computed by the '.fkine ' function(denoting forward kinematics) and these positions are represented in scatter plot . Below are the plots of these positions that show the potential workspace of our robotic manipulator.

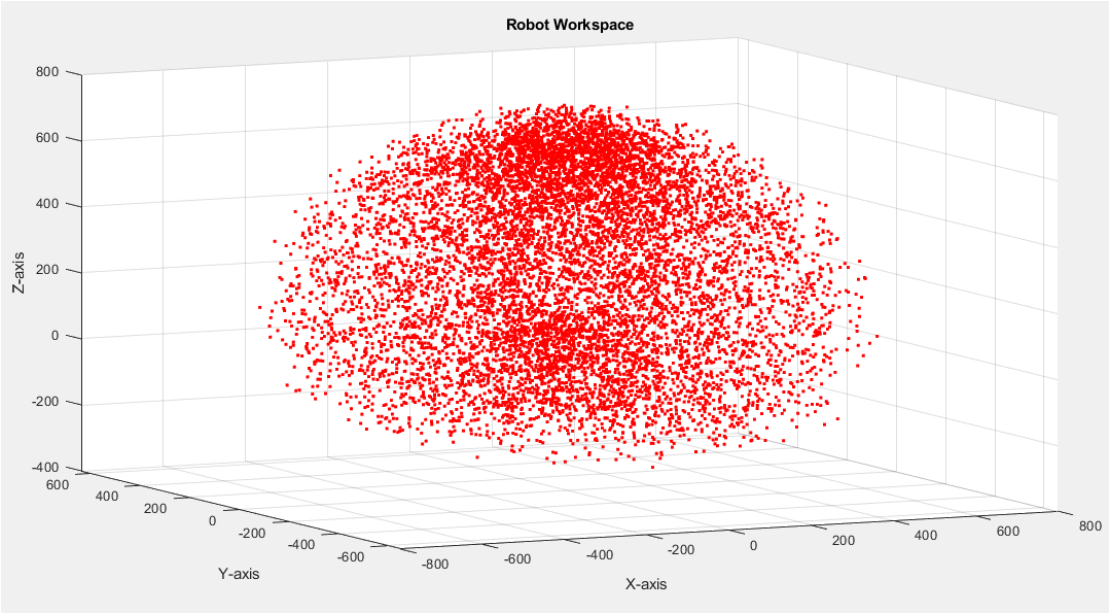


Figure 4.20: : Scatter Plot of 10000 reachable positions depicting the workspace of robot in 3D Space.

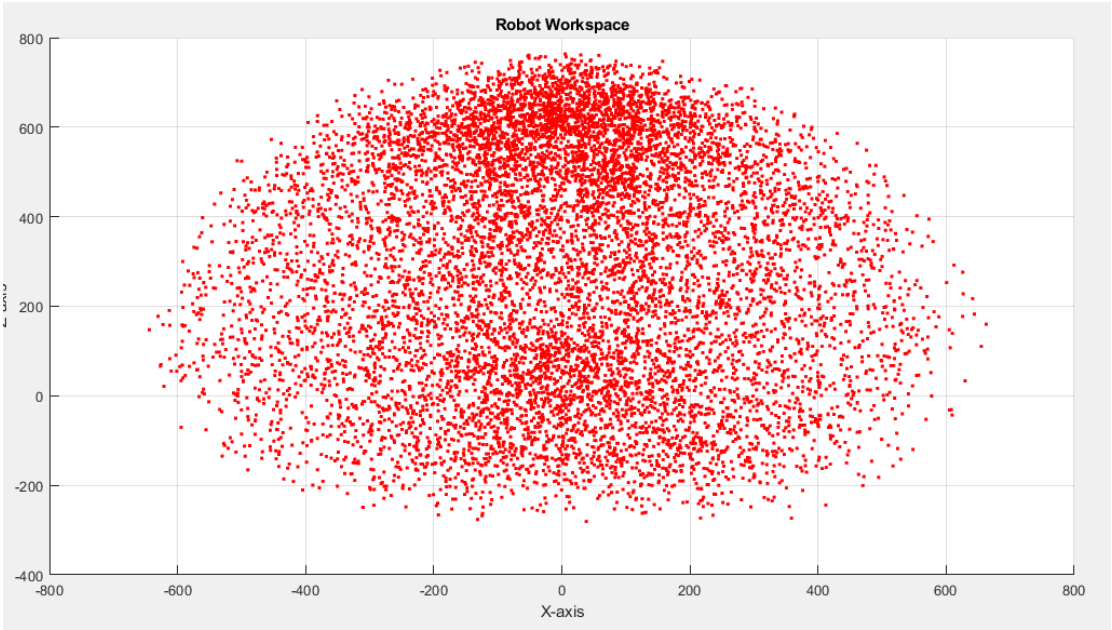


Figure 4.21: : Scatter Plot of 10000 reachable positions depicting the workspace of robot in X-Z Plane.

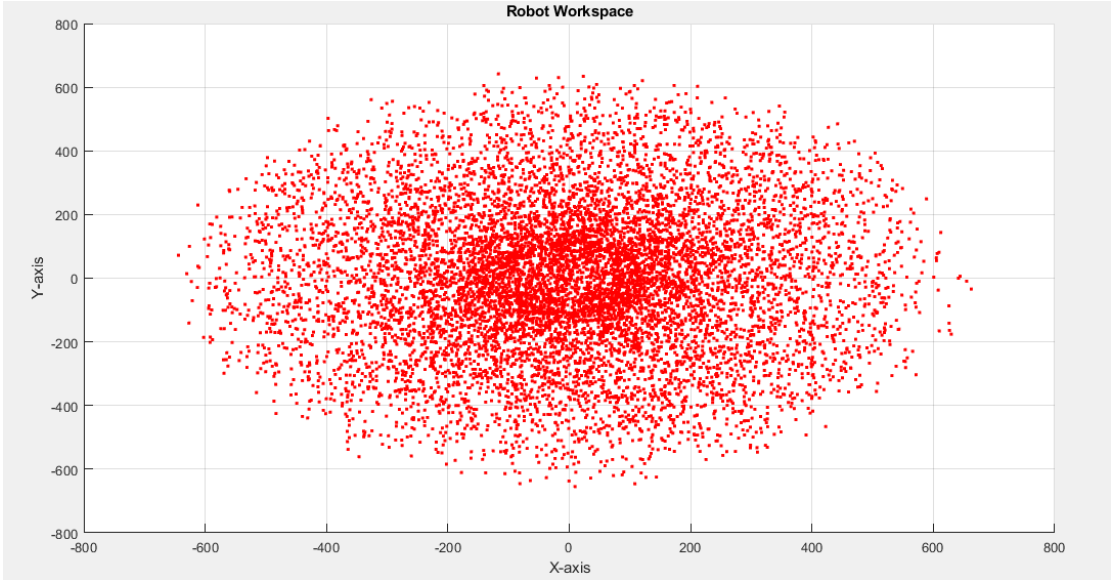


Figure 4.22: : Scatter Plot of 10000 reachable positions depicting the workspace of robot in X-Y Plane.

4.4 Path Planning

Path planning, also known as motion planning, is a fundamental problem in robotics and computer science, where the goal is to find a feasible path for a robot or vehicle to navigate from a start point to a goal point while avoiding obstacles and adhering to constraints. In our case,path planning for the arm was done in MATLAB.The SLERP interpolation was incorporated for smoothly transitioning between two orientations, such as rotating an object from one orientation to another.

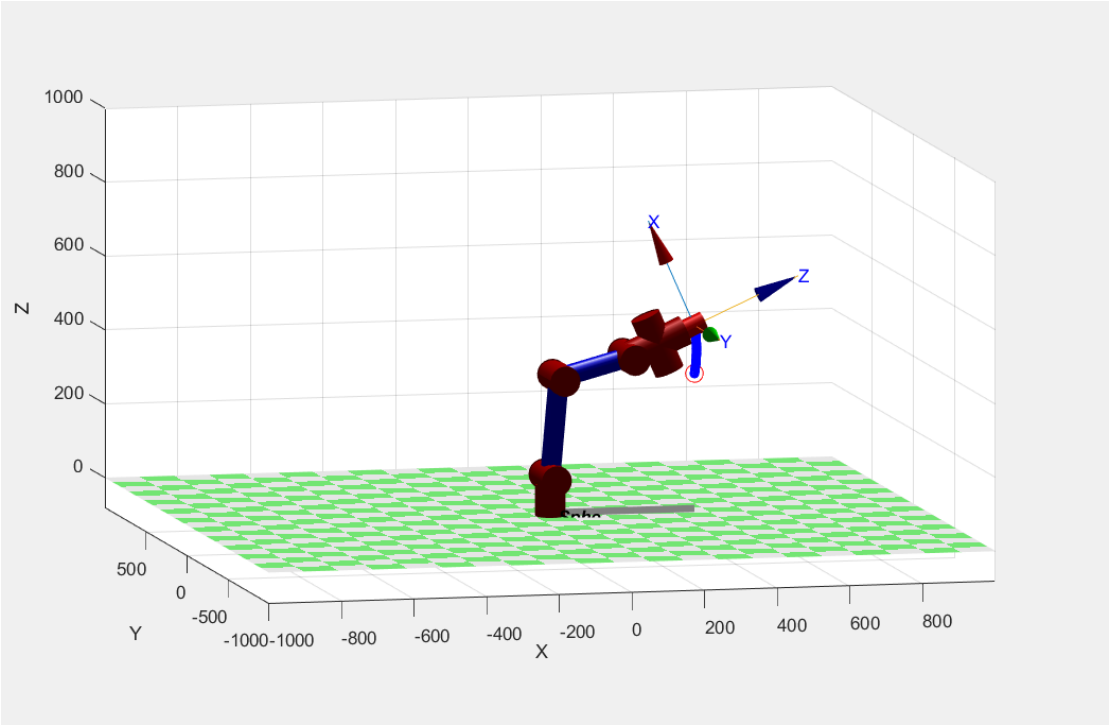


Figure 4.23: :Path planning from one co-ordinate to another.

4.5 Limitations

The design works done in this project are directed towards building a robotic arm with stepper motors having similar functionality as that of commercial arms. To achieve this, all foundational works of design for torque,kinematic modelling and simulation are done. However, there are some limitations of the model which are listed below:

1. The design consists of many belt and pulley transmissions. They increase the number of pulleys utilized which in turn make the whole robot bulky.

2. As a consequence of a design flaw, wherein the driven pulley of the shoulder is

situated on only one side of shoulder, there is an uneven distribution of motor driving force during the movement of link 1 up and down. This led to the arm twisting about an axis perpendicular to the main axis of driven pulley, ultimately increasing the motor load.

3. Due to high reduction ratio and no design consideration for tensioner, there is slippage of belt. This has prevented us to control the spring counterbalanced robotic arm through actuators in the limited amount of time available for this project.

4.6 Future Scope

With enhanced design and integration of electronics for control, a resilient robotic arm can be constructed using compact and cost-effective actuators. Thanks to its spring counterbalancing mechanism, this robotic arm can effortlessly manage loads. Consequently, it can be deployed across a spectrum of industrial applications such as pick and place, material handling, assembly, drilling, and beyond.

Moreover, the project's on-demand manufacturing capability will bolster and streamline manufacturing processes within industries. This advancement promises to enhance efficiency and intelligence within our industrial sectors, paving the way for smarter, more efficient operations.

4.7 Problems Faced

During the fabrication and testing of 2DOF counterbalance mechanism ,design of 6 DOF robotic arm ,following problems were encountered:

1. The counterbalance mechanism faced a challenge due to the unavailability of the required stiffness spring.

2. Belt slippage ensued due to a substantial reduction ratio and challenges in properly tensioning the belt, compounded by the absence of space to integrate an idler due to the compact design of the robotic arm.

3. We faced challenge due to unavailability of wire possessing the desired combination of strength and flexibility. Steel wire while robust proved difficult to ma-

nipulate, while nylon wire, although easier to handle, succumbed to spring force and broke. As a last resort, we attempted to enhance the strength of nylon wire by doubling it. However, this solution failed to adequately address the underlying issue but we were able to perform testing.

4. Lack of information about the designing and fabrication of the process caused several iterations of designing components.

5. We encountered challenges in assembling various components due to a lack of comprehensive knowledge and understanding of the design for assembly process.

4.8 Budget Analysis

Table 4.2: Budget Analysis

S.N.		Components	Qty	Market price per piece (NRs.)	Total
1		**Electric components**			
	a.	Stepper Motor (4kgcm)	2	1018.4	2036.8
	b.	Stepper Motor (10.1 kgcm)	2	2400	4800
	c.	Nema 17 Pancake	1	1600	1600
	d.	Nema 17 Stepper Motor Small	1	1600	1600
2		**Electronic Components**			
	a.	TB6600 Driver	3	1481.6	4444.8
	b.	A4988 Driver	6	240	1440
	c.	Jumper wire	120	5	600
	e.	Multimeter	1	800	800
	h.	Arduino Mega	1	2200	2200
	i.	Miscellaneous		2900	2900
		Mechanical Components			
	a.	Spring	8	150	1200
	b.	Dendrite	1	450	450
	c.	Nut and Bolts + Zip Tie		530	530
	d.	Reduction system	1	1000	1000
	e.	2GT 6mm belt	3	150	450
	f.	Spring(loose)	4	112.5	450
	g.	Nylon Rope	1	100	100
	h.	LN Bolt + Nut + 25 mm Bolt	40	22	880
	i.	Bolt 100mm	4	25	100
	j.	Wire Steel	1	40	40
	k.	Clamper	3	43.3	129.9
	l.	Screw		40	40
	m.	Acrylic Plate 3mm		1000	1000
	n.	2GT- 20 Teeth Pulley (5mm Bore)	2	75	150
	o.	Pillow Block Bearing	4	240	960
	p.	2GT 6mm Belt	1	200	200
	q.	Filament	1	2000	2000
	r.	Spring and Clamper		700	700
	s.	Spring, Al Pipe, 150mm Bolt, Rope		700	700
	t.	Spray Paint		300	300
	u.	20 Teeth pulley		175	175
	v.	RS 607 Bearing	1	320	320
	w.	Bearing (5mm bore dia)	6	40	240
	x.	608 RS Bearing	10	35	350
	y.	Thrust Bearing and Bearing 6000		1400	1400
4		**Cutting and Printing**			
	a.	Laser cutting		500	500
	b.	3D Printing		7340	7340
Total					**44126.5**

CHAPTER 5: CONCLUSION AND RECOMMENDATIONS

The six DOF robotic arm with spring CBM was designed, modeled and fabricated.The use of this mechanism decreased the required torque in joint 2 by 86.5% and in joint 3 by 68%. 3D model of the arm was designed to accomodate the CBM system. Actuators for joint 2, 3 and 4 were designed to be placed on the base so that the weight of the actuators doesn't increase the load in lower joints. This system also reduces the rotational inertia. Through this design and fabrication, the accomodation of a successful CBM mechanism with 6 dof robotic arm body has been achieved. Due to design flaws in belt size and tensioner,there was seen slippage of belt over the pulley which prevented us to control the fabricated robotic arm within the allocated time frame. However, with proper design changes, the arm can be easily controlled with the actuators. Kinematic model has been developed to build forward and inverse kinematics for the arm. Thus developed kinematic model is validated through MATLAB's simulator.Integrating this kinematics with proper control through Raspberry Pi,movements of the arm can be planned.

Also, the website was developed allowing the user to remotely place the order for the production of their product as well as allow them to track the order information at real time. This enhances the connectivity between the end user and manufacturing enterprise. In order to validate the working of website developed, the robotic arm was incorporated to integrate on-demand manufacturing.For this, the angle was provided by the end-users remotely through the website which in turn controlled the actuators in the robotic arm automatically.Database management and the inventory control was also integrated in the project.

Following recommendations can be handy on future endeavors in this field of study:

i. The power transmission by belt and pulley system imposes major complications on adjusting the tension of the belt. Proper design consideration for use of idler or tensioner is necessary.

ii. Due to use of a rope-spring counterbalance system, proper management of rope and its locking system should be taken into account during design process.

iii. Driven pulley on both sides of the shoulder is recommended.This balances the

weight and distributes the motor force evenly.

iv. Aluminium Plates on the body is recommended because it is rigid than acrylic. Acrylic plates bend in the transverse direction.

v. Spring of required stiffness should be used.This ensures the success of spring CBM.

vi. Design process should include 'design for assembly' considerations. This will make the assembly process easy.

REFERENCES

[1] Lab, L. Dudás. ”Simulation of Industrial Robots’ Six Axes Manipulator Arms - A Case Study.” *Academic Journal of Manufacturing Engineering*, vol. 19, no. 1, pp. 89-97, March 2021.

[2] Retrieved from Raspberry Pi website: https://www.raspberrypi.com/products/raspberrypi-1-model-b-plus

[3] Jkong Motor. ”Stepper Motor NEMA23 Round Shaft (Jk57hs51-2804).” Made-in-China.com, n.d.
Available: https://jkongmotor.en.made-in-china.com/product/xKtEMSJHHyhq/China-Stepper-Motor-NEMA23-Round-Shaft-Jk57hs51-2804.html.

[4] NMB. ”17PM-J349U/1 Datasheet.” DatasheetsPDF.com.
Available: https://datasheetspdf.com/pdf-file/1086665/NMB/17PM-J349U/1. Accessed: December 19, 2023.

[5] H. Kim and J. Song, ”Multi-DOF Counterbalance Mechanism for a Service Robot Arm,” IEEE/ASME Transactions on Mechatronics, vol. 19, no. 6, Dec. 2014.

[6] Lee, D.-G., Park, J.-W., Seo, T.-W. (2016, August). Low-inertia serial manipulator with counterbalance mechanism. In 2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI) (pp. 827-832).

[7] D. Lee and T. Seo, ”Lightweight Multi-DOF Manipulator with Wire-Driven Gravity Compensation Mechanism,” IEEE/ASME Transactions on Mechatronics [2017].

[8] Hayawi, M. J. (2017). The closed form solution of the inverse kinematics of a 6-DOF robot. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 1234-1239.

[9] Author(s), ”Parametrically Modeled DH Table for Soft Robot Kinematics: Case Study for A Soft Gripper,” in Advances in Mechanism and Machine Science, ed(s). Editor(s), Publisher, Jun. 2019.

[10] MachineMetrics. Industry 4.0 Technologies.
Retrieved from https://www.machinemetrics.com/blog/industry-4-0-technologies

[11] M. G. Sawczuk, "Design and control of a 3D printed, 6DoF robot arm," Degree project, Mechanical Engineering, First Cycle, Stockholm, Sweden, 2021

[12] Ferro tmt rolling mill, Nepal. (2018, August 13). [Video file]. Retrieved from https://www.youtube.com/watch?v=-YutixderJs

[13] H. -S. Kim, J. -K. Min and J. -B. Song, "Multi-DOF counterbalance mechanism for low-cost, safe and easy-usable robot arm," 2014 11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Kuala Lumpur, Malaysia, 2014.

[14] D. G. Chung, M. Hwang, J. Won and D. -S. Kwon, "Gravity compensation mechanism for roll-pitch rotation of a robotic arm," 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea (South), 2016, pp. 338-343, doi: 10.1109/IROS.2016.7759076.

[15] Y. Kim, "Design of Low Inertia Manipulator with High Stiffness and Strength Using Tension Amplifying Mechanisms," in Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Congress Center Hamburg, Hamburg, Germany, Sept 28 - Oct 2, 2015.

[16] S. Mahalingam and A. Sharan, "The optimal balancing of the robotic manipulators," Proceedings. 1986 IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, 1986, pp. 828-835, doi: 10.1109/ROBOT.1986.1087570.

[17] T. Schurink, "An Enquiry into the Relation between Components and Their Composition on Desired Performance and Cost of Pick & Place Robotic Arms," Bachelors Assignment, Industrial Design Engineering, Eindhoven University of Technology, Eindhoven, Netherlands, Oct. 2016. Advisory: T.H.J. Vaneker. Examiner: M.C. van der Voort.

Accessed: December 19, 2023.

[18] Aus3D, "GT2 Timing Belt," Aus3D - 3D Printing Supplies. [Online]. Available: https://aus3d.com.au/products/gt2-timing-belt. [Accessed:25 Dec,2023].

[19] ”Gravity compensated manipulator,” YouTube. [Online].
Available: https://youtu.be/oivGZMIMZl8?si=BR7VAtdeRHDdkReW. [Accessed: 17 July,2023].

[20] ”Using [peter corke] robotics toolbox with Matlab GUI - Forward and Inverse kinematics.,” YouTube. [Online].
Available: https://youtu.be/50EfgtsD-24?si=6CMewCzaQWrrcsu4 [Accessed: 28 Nov,2023]

[21] ”Introduction to Robotics,” YouTube. [Online].
Available:https://youtu.be/a6$_f$gnuuYfE?si = 3fYlUe9lOxoTAEbH[Accessed : June, 2023]

APPENDIX A

Exploded Views

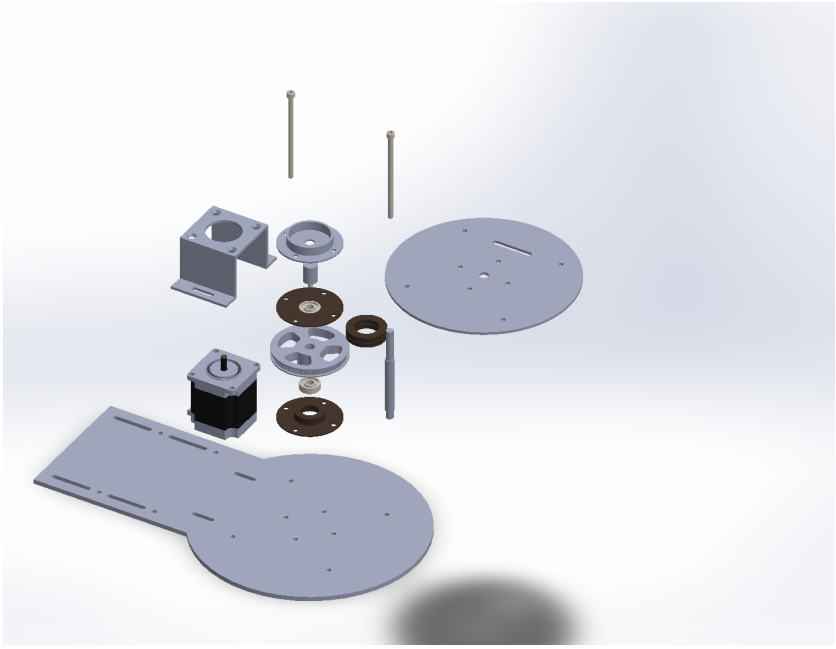


Figure A.1: Exploded View of Base Assembly

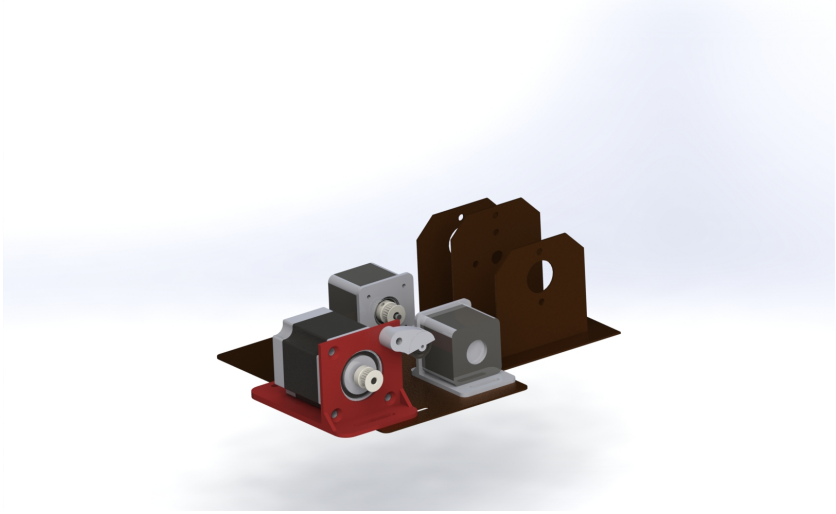


Figure A.2: Exploded View of Waist Assembly

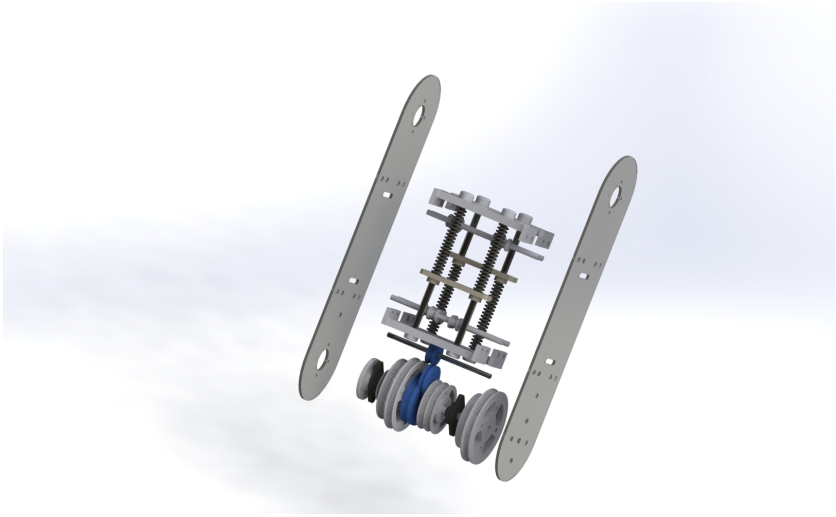


Figure A.3: Exploded View of Shoulder Assembly

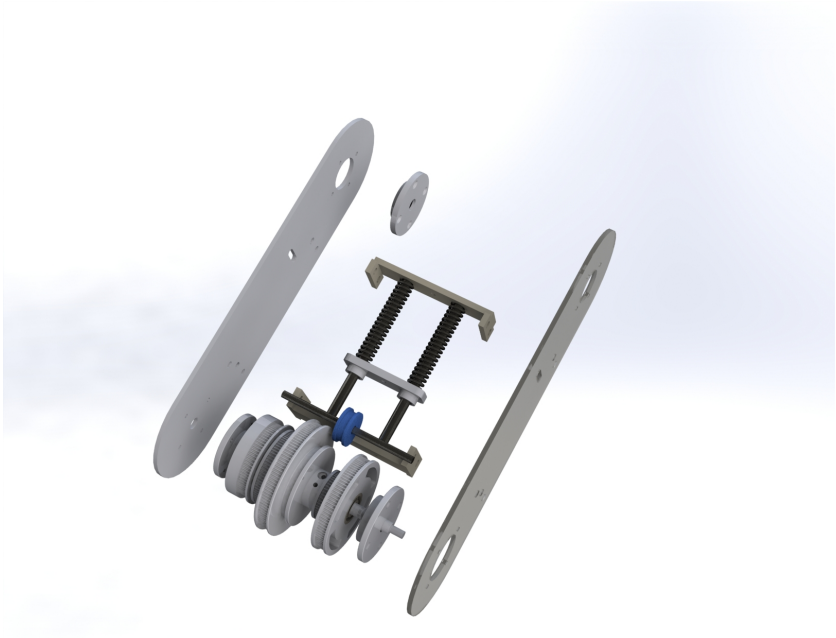


Figure A.4: Exploded View of Elbow Assembly

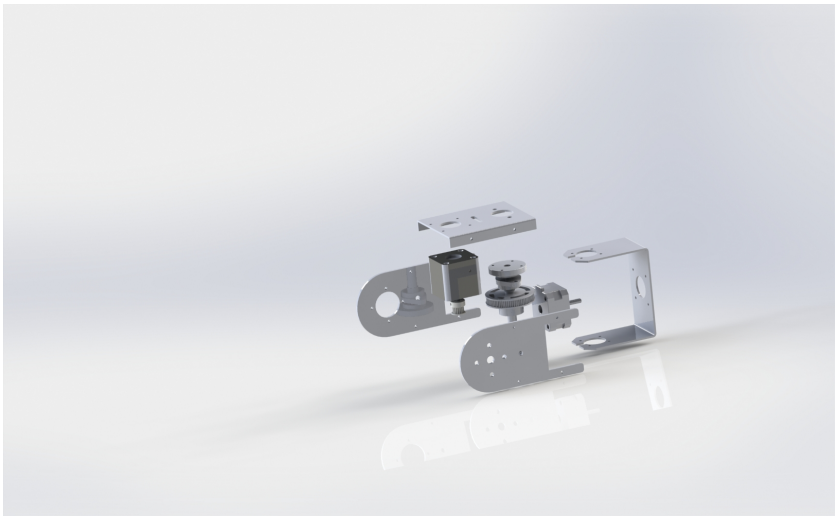


Figure A.5: Exploded View of Wrist Assembly

APPENDIX B

Fabricated Prototype

Figure B.1: Complete Assembly of Prototype

Figure B.2: Top View of Complete Asembly

Figure B.3: Motor Placement in base

Figure B.4: CBM for links 2 and 3 in Shoulder (Top View)

Figure B.5: CBM for links 2 and 3 (side View)

Figure B.6: CBM for links 4 in Elbow

APPENDIX C

Matlab Codes

Matlab Code - Kinematic Modelling

```
%Forward Transformation Matrix
%Forward Kinematics.....................................
%clear
%clc
%Link Length
a2 = 270;
a3 = 200;
a4 = 85;
%Joint Variables
d1 = 100;
a6 = 112;
%{
Angth_1 = input('Theta1(-90<x<90) = ');
Angth_2 = input('\nTheta2(-50<x<75) = ');
Angth_3 = input('\nTheta3(0<x<80) = ');
Angth_4  = input('\nTheta4(-90<x<90) = ');
Angth_5 = input('\nTheta5(-90<x<90) = ');
Angth_6 = input('\nTheta6(-180<x<180) = ');
%}
Angth_1=0;
Angth_2 =30;
Angth_3 =-30;
Angth_4 =-60;
Angth_5 =0;
Angth_6 =0;
Theta1 = deg2rad(Angth_1);
Theta2 = deg2rad(Angth_2+90);
Theta3 = deg2rad(Angth_3-90);
Theta4 = deg2rad(Angth_4);
 if (Angth_5>=0)
```

```
    %Theta5 = deg2rad(Angth_5) + pi/2;
    Theta5 =  pi/2 - deg2rad(Angth_5);

else
    Theta5 =  pi/2-deg2rad(Angth_5);
end
Theta6 = pi/2 - deg2rad(Angth_6);
rad2deg(Theta1);
rad2deg(Theta2);
rad2deg(Theta3);
rad2deg(Theta4);
rad2deg(Theta5);
rad2deg(Theta6);
%Orientation Vectors
ax = cos(Theta1)*cos(Theta2+Theta3+Theta4)*sin(Theta5)-
            sin(Theta1)*cos(Theta5);
ay = sin(Theta1)*cos(Theta2+Theta3+Theta4)*sin(Theta5) +
            cos(Theta1)*cos(Theta5);
az = sin(Theta2+Theta3+Theta4)*sin(Theta5);
sx = cos(Theta1)*(sin(Theta2+Theta3+Theta4)*cos(Theta6)-
            cos(Theta2+Theta3+Theta4)*cos(Theta5)*sin(Theta6))-
            sin(Theta1)*sin(Theta5)*sin(Theta6);
sy = sin(Theta1)*(sin(Theta2+Theta3+Theta4)*cos(Theta6)-
            cos(Theta2+Theta3+Theta4)*cos(Theta5)*sin(Theta6))
            +cos(Theta1)*sin(Theta5)*sin(Theta6);
sz= -sin(Theta2+Theta3+Theta4)*cos(Theta5)*sin(Theta6)-
            cos(Theta2+Theta3+Theta4)*cos(Theta6);

nx = cos(Theta1)*(cos(Theta2+Theta3+Theta4)*cos(Theta5)
            *cos(Theta6)+sin(Theta2+Theta3+Theta4)
            *sin(Theta6))+sin(Theta1)*sin(Theta5)*cos(Theta6);
ny = sin(Theta1)*(cos(Theta2+Theta3+Theta4)*cos(Theta5)
```

```
            *cos(Theta6)+sin(Theta2+Theta3+Theta4)*sin(Theta6))-
            cos(Theta1)*sin(Theta5)*cos(Theta6);
nz = sin(Theta2+Theta3+Theta4)*cos(Theta5)*cos(Theta6)
            -cos(Theta2+Theta3+Theta4)*sin(Theta6);
%Position vectors of end position w.r.t_base position.
%Forward Kinematics
Px = cos(Theta1)*(a6*sin(Theta5)*cos(Theta2 + Theta3+Theta4)+
            a4*cos(Theta2 + Theta3+Theta4)+a2*cos(Theta2)+a3
            *cos(Theta2 + Theta3))-sin(Theta1)*cos(Theta5)*a6;
Py = sin(Theta1)*(a6*sin(Theta5)*cos(Theta2 + Theta3+Theta4)+
            a4*cos(Theta2 + Theta3+Theta4)+a2*cos(Theta2)+
            a3*cos(Theta2 +Theta3))+cos(Theta1)*cos(Theta5)*a6;
Pz = d1+a6*sin(Theta2+Theta3+Theta4)*sin(Theta5)+
            a3*sin(Theta2+Theta3)+a2*sin(Theta2)+a4
            *sin(Theta2+Theta3+Theta4);
Pos = [Px;Py;Pz]
OR = [nx,sx,ax;ny,sy,ay;nz,sz,az];
%Inverse Kinematics.....................................
%angle theta1
theta1 = atan2((Py-a6*ay),(Px-a6*ax));
fprintf('theta1 = %.4f \n',rad2deg(theta1));

%angle theta5
x5 = cos(theta1)*ay-sin(theta1)*ax;
y5 = sqrt(1-x5^2);

theta5 = atan2(y5,x5);
if (theta5>=pi/2)
    Angth5 = -theta5+ pi/2;
else
    Angth5 = pi/2-theta5;
end
```

```
%fprintf('theta5 = %.4f\n ',rad2deg(theta5));

%angle theta234
ang = atan2(az,cos(theta1)*ax+sin(theta1)*ay);
if(theta5 > 0)
    theta234 = ang ;  %for theta5>0
elseif(theta5<0)
    theta234 = ang + pi;
end
%fprintf('theta234 = %.4f\n ',rad2deg(theta234));

%angle theta2
P1 = cos(theta1)*Px+sin(theta1)*Py-a6*sin(theta5)*cos(theta234)
                -a4*cos(theta234);
P2 = Pz-d1-a6*sin(theta5)*sin(theta234)-a4*sin(theta234);
N = (P1^2+P2^2+a2^2-a3^2)/(2*a2);
if ((P1^2+P2^2-N^2)<0)
    x5 = cos(theta1)*ay-sin(theta1)*ax;
    y5 = -sqrt(1-x5^2);
    theta5 = atan2(y5,x5);
    if (theta5>=pi/2)
    Angth5 = -theta5 + pi/2;
    else
    Angth5 =  pi/2-theta5;
    end
    %fprintf('theta5 = %.4f\n ',rad2deg(theta5));
    ang = atan2(az,cos(theta1)*ax+sin(theta1)*ay);
    if(theta5 > 0)
    theta234 = ang ;  %for theta5>0
    elseif(theta5<0)
```

```
    theta234 = ang + pi;
    end
    P1 = cos(theta1)*Px+sin(theta1)*Py-a6*sin(theta5)
                *cos(theta234)-a4*cos(theta234);
    P2 = Pz-d1-a6*sin(theta5)*sin(theta234)-a4*sin(theta234);
    N = (P1^2+P2^2+a2^2-a3^2)/(2*a2);
end
 %fprintf('theta5 = %.4f\n ',rad2deg(theta5));

 theta2 = atan2(P2,P1)+atan2(sqrt(P1^2+P2^2-N^2),N);
 Angth2 = theta2 - pi/2;

fprintf('theta2 = %.f\n ',rad2deg(theta2));
%angle theta23
theta23 = atan2(P2-a2*sin(theta2),P1-a2*cos(theta2));
thet23 = rad2deg(theta23);
%angle theta3
 theta3 = (theta23 - theta2);
 Angth3 = theta3 + pi/2;


fprintf('theta3 = %.4f\n ',rad2deg(theta3));
%angle theta4
if(theta5>=0)
    theta4 = theta234 - theta23;
else
    theta4 = theta234 - theta23-2*pi;
end
fprintf('theta4 = %.4f\n ',rad2deg(theta4));
fprintf('theta5 = %.4f\n ',rad2deg(theta5));

%angle theta6.....................................
```

```
val = atan2(sy*cos(theta1)-sx*sin(theta1),
                (-ny*cos(theta1)+nx*sin(theta1)));
if (theta5>0)
    theta6 = val;
elseif (theta5<0)
    theta6 = val +pi;
end
Angth6 = pi/2-theta6;
fprintf('theta6 = %.4f\n\n\n',rad2deg(theta6));

fprintf('In simulated environment\n');

fprintf('theta1 = %.f\n ',rad2deg(theta1));
fprintf('theta2 = %.f\n ',rad2deg(Angth2));
fprintf('theta3 = %.f\n ',rad2deg(Angth3));
fprintf('theta4 = %.f\n ',rad2deg(theta4));
fprintf('theta5 = %.f\n ',rad2deg(Angth5));
fprintf('theta6 = %.4f\n\n\n',rad2deg(Angth6));
```

Matlab Code - Robot Workspace

```
clc;
clear
disp('Spherical Manipulator');
syms a2 a3 a4 a6 d1
%Joint angles
Angth_1=deg2rad(45);
Angth_2 =deg2rad(30);
Angth_3 = deg2rad(0);
Angth_4 = deg2rad(-30);
Angth_5 = deg2rad(-45);
Angth_6 = deg2rad(0);

%Link Length
a2 = 270;
a3 = 200;
a4 = 85;
%Joint Variables
d1 = 100;
a6 = 112;
%D-H Parameter[theta,d,a,alpha,offset]
%if [Prismatic joint: theta = theta,d = 0,
        offset = 1,after offset put the
%value of d
%if revolute joint: theta = 0,offset = 0,
        after offset put the value of
%theta
%Links
H0_1 = Link([0,d1,0,pi/2,0,0]);
H0_1.qlim = [-pi pi];
```

```
H1_2 = Link([0,0,a2,0,0,pi/2]);
H1_2.qlim = [-pi/2 pi/3];

H2_3 = Link([0,0,a3,0,0,-pi/2]);
%H2_3.qlim = [-pi/6 pi/2];
H2_3.qlim = [-pi/3 pi/2];


H3_4 = Link([0,0,a4,-pi/2,0,0]);
H3_4.qlim = [-pi/2 pi/2];

H4_5 = Link([0,0,0,pi/2,0,pi/2]);
H4_5.qlim = [-pi/2 pi/2];

H5_6 = Link([0,a6,0,0,0,pi/2]);
H5_6.qlim = [-pi pi];

Sphe = SerialLink([H0_1 H1_2 H2_3 H3_4 H4_5 H5_6],'name','Sphe');
Sphe.plot([0 0 0 0 0 0],'workspace',[-450 700 -450 450 0 800]);
Sphe.teach;

% Define a range of joint configurations
q_min = (Sphe.qlim(:, 1));  % Minimum joint positions
q_max = (Sphe.qlim(:, 2)) ; % Maximum joint positions

% Generate random joint configurations within the joint limits
num_samples = 10000;
random_joint_configs = repmat(q_min, 1, num_samples)
        + rand(size(q_min, 1), num_samples) .
        * repmat(q_max - q_min, 1, num_samples);

% Compute the end-effector positions for the random joint configurat
```

```
end_effector_positions = zeros(3, num_samples);
for i = 1:num_samples
    end_effector_positions(:, i) = Sphe.fkine(random_joint_configs(:
end

% Plot the robot workspace
scatter3(end_effector_positions(1, :), end_effector_positions(2, :),
            end_effector_positions(3, :), 'r.');
% Plot the robot workspace envelope
%plot3(end_effector_positions(1, :), end_effector_positions(2, :),
            end_effector_positions(3, :), 'r-');

%{
[q1] = Sphe.ikine(T,q0);
[q2] = Sphe.ikcon(T,q0);
 disp('by Ikine');


disp(rad2deg(q1));


 disp('by Ikcon');
disp(rad2deg(q2));
%disp(solInfo);
%}
xlabel('X-axis');
ylabel('Y-axis');
zlabel('Z-axis');
title('Robot Workspace');
grid on;
```

APPENDIX D

Part Drawings

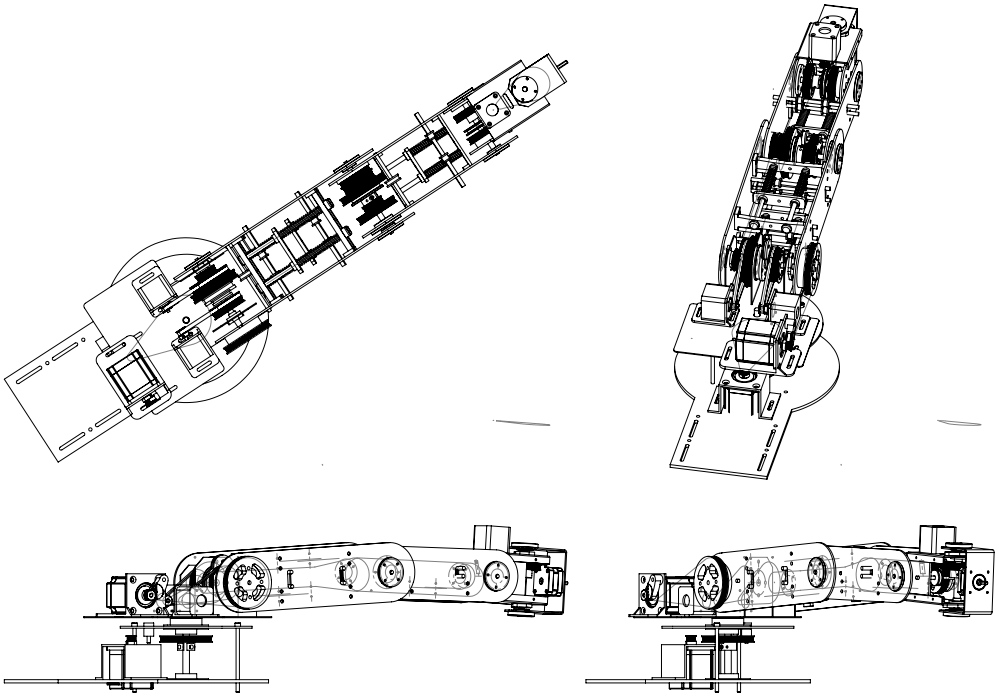


Figure C.1: Complete Assembly Drawing

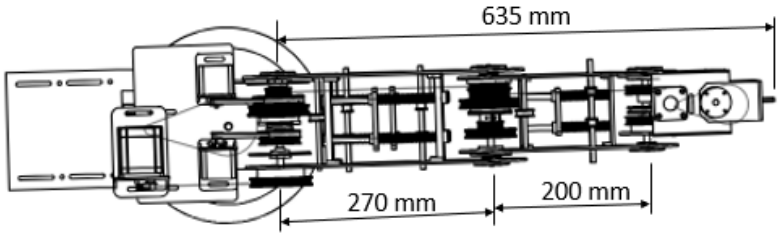


Figure C.2: Top View of Complete Assembly

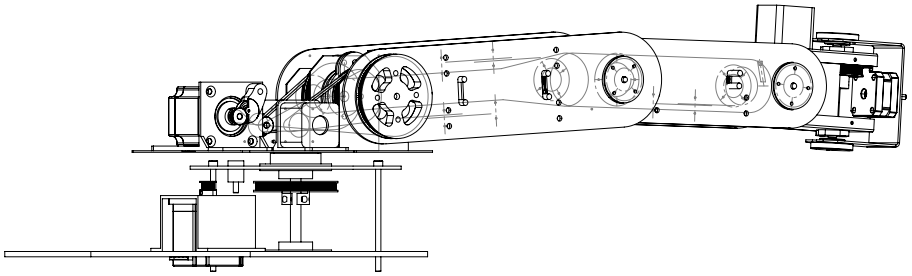


Figure C.3: Side View of Complete Assembly

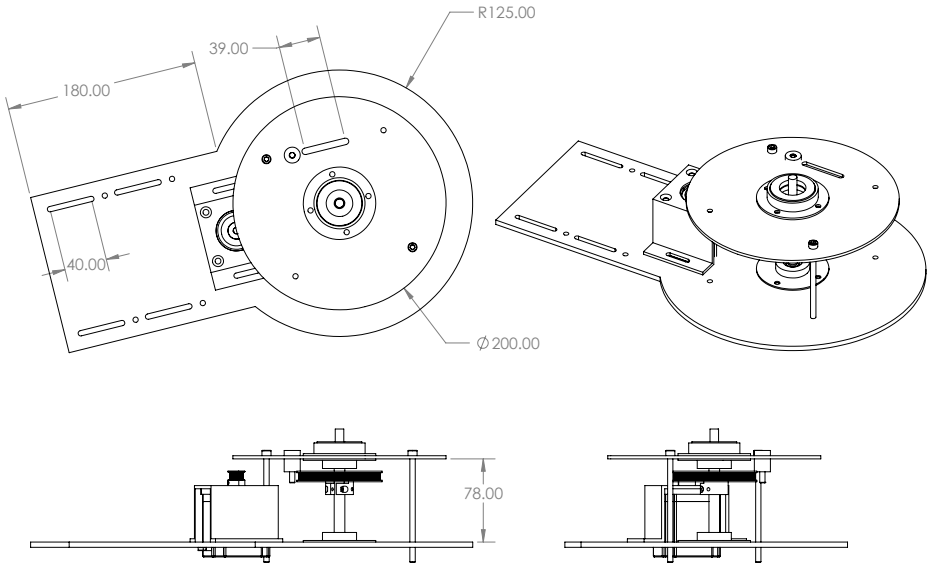


Figure C.4: Base Drawing

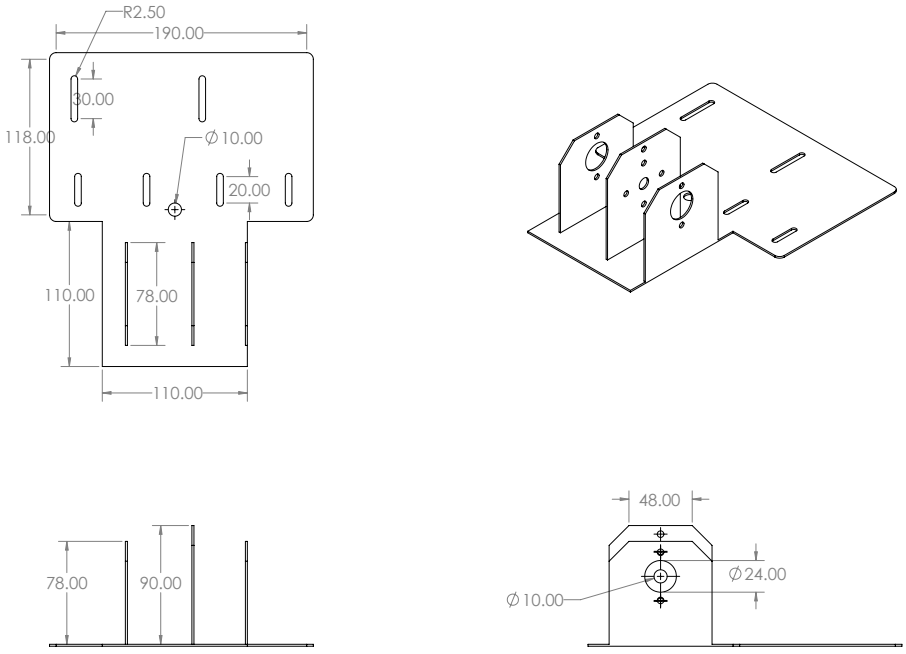


Figure C.5: Waist Drawing

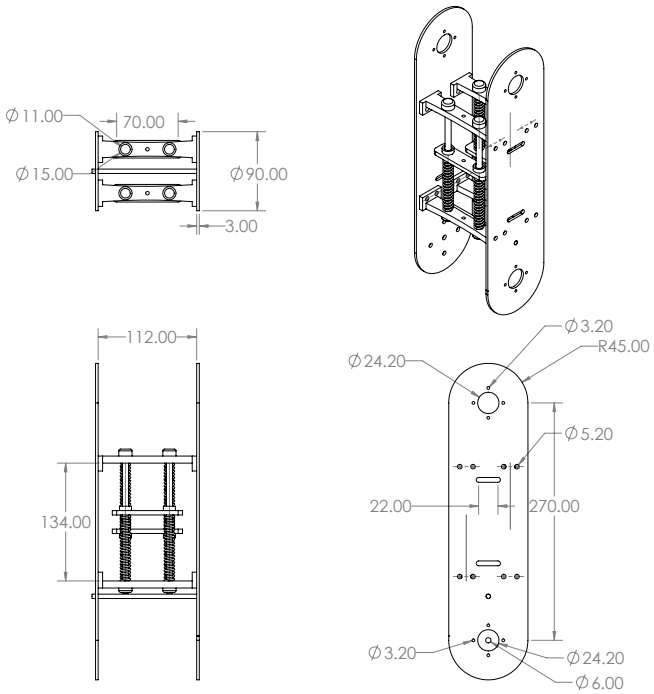


Figure C.6: Shoulder Assembly Drawing

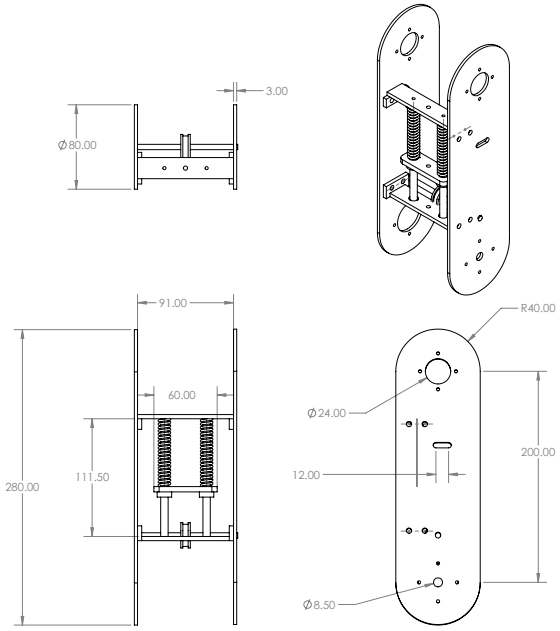


Figure C.7: Elbow Assembly Drawing

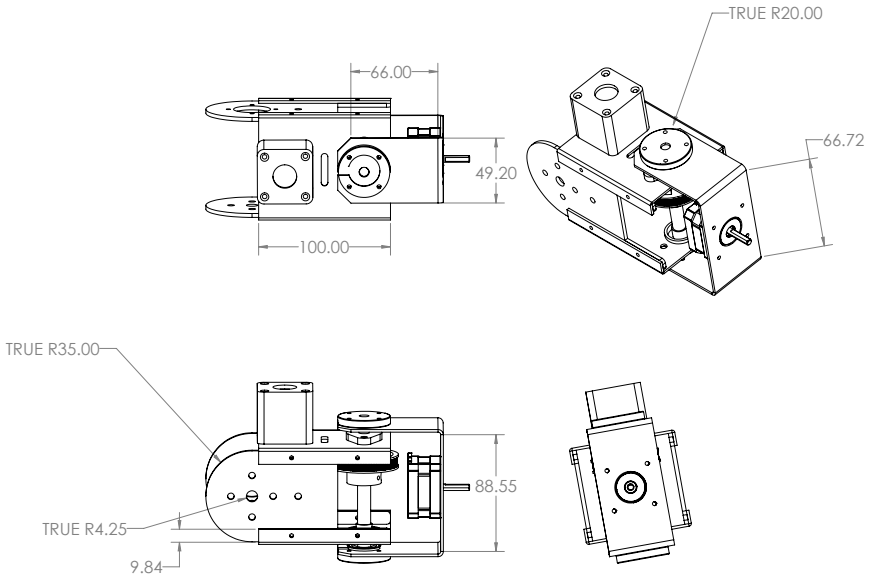


Figure C.8: Wrist Assembly Drawing

DESIGN, FABRICATION AND AUTOMATION OF INDUSTRIAL 6-DOF ROBOTIC ARM WITH ON-DEMAND MANUFACTURING CAPABILITIES FOR MASS/BATCH PRODUCTION INDUSTRIES

ORIGINALITY REPORT

15%
SIMILARITY INDEX

PRIMARY SOURCES

1	Hwi-Su Kim, Jae-Bok Song. "Multi-DOF Counterbalance Mechanism for a Service Robot Arm", IEEE/ASME Transactions on Mechatronics, 2014 Crossref	199 words — 1%
2	DongGyu Lee, TaeWon Seo. "Lightweight Multi-DOF Manipulator With Wire-Driven Gravity Compensation Mechanism", IEEE/ASME Transactions on Mechatronics, 2017 Crossref	182 words — 1%
3	www.ncbi.nlm.nih.gov Internet	156 words — 1%
4	Kim, Hwi-Su, Jae-Kyung Min, and Jae-Bok Song. "Multi-DOF counterbalance mechanism for low-cost, safe and easy-usable robot arm", 2014 11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), 2014. Crossref	121 words — 1%
5	"Advances in Mechanism and Machine Science", Springer Science and Business Media LLC, 2019 Crossref	62 words — < 1%
6	open-robotics.com	

Internet

61 words — < 1%

7 Jun Zhang, Qing-xue Huang, Wenjun Meng, Xiangdong Yu, Guoying Meng, Lifeng Ma. "Kinematics uncertainty analysis of mine bolter manipulator based on Chebyshev interval algorithms", Advances in Mechanical Engineering, 2019
Crossref

55 words — < 1%

8 Moura, Tiago Gomes. "Desenvolvimento de um Sistema Robótico de Dois Braços Para Imitação Gestual", Universidade de Aveiro (Portugal), 2023
ProQuest

47 words — < 1%

9 Hwi-Su Kim, Jae-Bok Song. "Low-cost robot arm with 3-DOF counterbalance mechanism", 2013 IEEE International Conference on Robotics and Automation, 2013
Crossref

43 words — < 1%

10 M. Kemal Ozgoren. "Kinematics of General Spatial Mechanical Systems", Wiley, 2020
Crossref

43 words — < 1%

11 Al Vito Anasrulloh, Andreas Widjaya, Mochamad Farhan Ali Irfani, Wildan Iswahyudi et al. "Implementation of Inverse Kinematics to Perform Walking Movements on Raspiberry-Pi Humanoid Robot", 2023 8th International Conference on Electrical, Electronics and Information Engineering (ICEEIE), 2023
Crossref

38 words — < 1%

12 arrow.tudublin.ie
Internet

38 words — < 1%