

PICK AND PLACE ROBOTIC ARM FOR THE ELDERLY & DISABLED

*Report submitted to the SASTRA Deemed to be University
as the requirement of the course*

MEC300/MCT300 MINI PROJECT

Submitted by

JAMALAPURAM PRAHARSHITH KASHYAP

(Reg No: 124009054)

ANNADANAM SATVIK

(Reg No: 124009118)

RUCHI GUPTA

(Reg No: 124012036)

May 2023



SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 of the UGC Act, 1956)



THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

**SCHOOL OF MECHANICAL ENGINEERING
THANJAVUR - 613401**

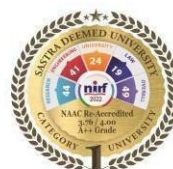


SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 of the UGC Act, 1956)



THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

SCHOOL OF MECHANICAL ENGINEERING THANJAVUR - 613401

Bonafide Certificate

This is to certify that the report titled “**Pick & Place Robotic Arm For The Elderly & Disabled**” submitted as a requirement for the course, **MEC300/MCT300 Mini-Project** for B.Tech. Mechanical Engineering & B.Tech. Mechatronics programme, is a bonafide certificate of the work done by **Mr. Jamalapuram Praharshith Kashyap (Reg No: 124009054)**, **Mr. Annadanam Satvik (Reg no: 124009118)** & **Ms. Ruchi Gupta (Reg No: 124012036)** during the Academic Year 2022-23, in the School of Mechanical Engineering, under my supervision.

Signature of the Project Supervisor :

Name with Affiliation

: Dr. C. Ramprasadh, Professor

Date

: 16-05-2023

Mini-Project Viva-voce conducted on :

Examiner 1

Examiner 2



SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 of the UGC Act, 1956)



THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

SCHOOL OF MECHANICAL ENGINEERING THANJAVUR - 613401

Declaration

We declare that the report “**Robotic Arm For The Elderly & Disabled**” submitted by us is an original work done by us under the guidance of **Dr. C. Ramprasadh, Senior Professor, School of Mechanical Engineering, SASTRA Deemed to be University** during the even semester of the Academic Year 2022-23, in the **School of Mechanical Engineering**. The work is original and wherever we have used materials from other sources, we have given due credit and cited them in the text of the report. This report has not formed the basis for the award of any degree, diploma, associateship, fellowship or other similar title to any candidate of any University.

Signature of the Candidates:

Name of the Candidates:

Jamalapuram Praharshith Kashyap (Reg No: 124009054)

Annadanam Satvik (Reg No: 124009118)

Ruchi Gupta (Reg No: 124012036)

Date: 17-05-2022

Acknowledgements

First and foremost, we are thankful to God Almighty for giving us the potential and strength to work on this project fruitfully. We express our gratitude to **Dr. S. VAIDHYASUBRAMANIAM**, Vice- Chancellor, SASTRA Deemed to be University for providing us with an encouraging platform for the course of study. We also thank **Dr. R. CHANDRAMOULI**, Registrar, SASTRA Deemed to be University for providing us the opportunity to work in the SOME, SASTRA Deemed to be University for the course of the work. We render our solicit thanks to **Dr. S. PUGAZHENTHI**, Dean, SOME, SASTRA Deemed to be University for introducing this Project Based paper in our curriculum.

We owe a debt of earnest gratitude towards our guide & mentor **Dr. C. RAMPRASADH** for his continued support and guidance throughout the duration of our project on the topic “**Robotic Arm For The Elderly & Disabled**”. I take this opportunity to thank all our lecturers who have directly or indirectly helped our project. I also extend our sincere thanks to our parents for their continuous moral support and friends without whose thought-provoking suggestion and encouragement, along with our guide, this work would not have been what it is today.

List of Figures

Figure No.	Name	Page No.
3.0	Methodology Flowchart	13
3.1.2.1	Mathematical Model of a manipulator	14
3.1.2.2	Representation of Links & Joints of 6 DoF	17
3.1.2.3	Frames of 6 DoF	17
3.1.2.4	D-H Parameters of 6 DoF	18
3.2.1	4 DoF CAD Design	18
3.2.2	6 DoF CAD Design	19
3.3.2	3D Printed Parts	20-24
3.4.1	SG90 Micro Servo Motor 9G	25
3.5	Gazebo & ROS Codes	26-28
3.6.1	Code running behind Alexa for Voice Recognition	29
3.6.2	Voice Interaction Models with Alexa	30
4.4.1 to 4.4.6	4 DoF Simulation	31-33

List of Tables

Table No.	Name	Page No.
<i>3.1.2.4</i>	<i>D-H Parameters of 6 DoF</i>	18

Nomenclature

Abbreviation	Full Form
DoF	Degrees of Freedom
IIoT	Industrial Internet of Things
MND	Motor Neuron Disease
PLA	Polylactic Acid
ROS	Robotic Operating System
Rviz	Robotic Visualization
D-H Parameters	Denavit-Hartenberg Parameters

Special Notations (Symbols)	Meaning
a_i	Link length [Link Parameters]
α_i	Link twist [Link Parameters]
d_i	Joint displacement [Joint Parameters]
θ_i	Joint angle [Joint Parameters]

Abstract

Robotics is currently a growing trend in digital media as well as in technology applications across a number of important industries. Robotic applications and other robotic features have grown since the Industrial Revolution 3.0. Our industrial efficiency and the caliber of our products have grown enormously since the introduction of manipulator kinds of robots. A manipulator is, by definition, a device used in robotics to move objects without the user making any direct physical contact. Robotic arms with a certain degree of freedom are used to do this; each arm is unique and created for a particular application or function. Designing a Pick & Place Robotic Manipulator that can be operated by voice recognition is the major goal of this project work. It is proposed that the manipulator be designed using ROS, Gazebo, and Robotics Theory. The project's initial strategy is to simulate the manipulator in Gazebo before subsequently designing and implementing it physically.

Specific Contributions:

- **124009054:**
 - Numerical analysis of 4 Degrees of Freedom using Inverse & Forward Kinematics.
 - Mini Project Report making.
- **124009118:**
 - CAD Designing of the 6 DoF Manipulator.
 - Voice Recognition feature addition.
- **124012036:**
 - Numerical analysis of 6 Degree of Freedom using Inverse & Forward Kinematics.
 - Literature Research.

Specific Learnings:

- **124009054:**
 - Inverse & Forward Kinematics.
 - Literature review & summarizing for report.
- **124009118:**
 - ROS, Gazebo & Alexa for voice recognition.
 - Analytical Analysis of Forward & Inverse Kinematics using VSCode.
- **124012036:**
 - Inverse & Forward Kinematics.
 - ROS and Gazebo functioning.

Table of Contents

Title	Page No
<i>Bonafide Certificate</i>	<i>1</i>
<i>Declaration</i>	<i>2</i>
<i>Acknowledgements</i>	<i>3</i>
<i>List of Figures</i>	<i>4</i>
<i>List of Tables</i>	<i>5</i>
<i>Nomenclature</i>	<i>6</i>
<i>Abstract</i>	<i>7</i>
1. <u>Introduction</u>	9
1.1. Robots	
1.2. Motor Neuron Issues	
1.3. IoT & Voice Assistants	
1.4. Motivation	
2. <u>Objectives</u>	11
3. <u>Methodology</u>	<u>12</u>
3.1. Kinematics	
3.1.1. Forward Kinematics	
3.1.2. Denavit-Hartenberg Notation	
3.2. Design	
3.3. 3D Printing	
3.3.1. Materials	
3.3.2. Parts	
3.4. Actuators	
3.5. Gazebo & ROS	
3.6. Voice Recognition using Alexa	
4. <u>Results & Discussion</u>	<u>31</u>
4.1. Working Designs	
5. <u>Conclusions & Further work</u>	<u>34</u>
6. <u>References</u>	<u>35</u>
7. <u>Appendix</u>	<u>36</u>
7.1. Similarity Check Report	

CHAPTER 1

INTRODUCTION

1.1 ROBOTS

Robots have been a crucial part of our day to day lives in the current present. A robot by definition is:

- *“Robot, any automatically operated machine that replaces human effort, though it may not resemble human beings in appearance or perform functions in a humanlike manner”* - Britannica

Or

- *“A robot is a machine—especially one programmable by a computer—capable of carrying out a complex series of actions automatically.”* - Wikipedia

Robots have always been a big part of human history, be it in respect to Mythologies or Science Fiction or even Science. The idea of a machine that could think like a human and reduce the strain on its creators, the humans, has been a route to shape what Robots are in the present. Many cultures around the world had legends that gave rise to the concept of automata. Ancient engineers and creators from cultures like Ancient China, Ancient Greece, and Ptolemaic Egypt strove to build self-contained devices, some of which resembled humans and animals. Early examples of automata include Archytas' artificial doves, Mozi and Lu Ban's artificial birds, Hero of Alexandria's "speaking" automaton, Philo of Byzantium's washstand automaton, and Lie Zi's tale of a human automaton.

Especially with the rapid development in the fields of IIoT & Industry 4.0, Robots and Robotics have taken up a notch in their respective domains as well. There still are ongoing attempts to integrate certain aspects of technological ease from software to hardware.

Robots have also helped humans, not just in the industrial sector, but medical & domestic aspects as well. It is safe to predict cybernetic enhancements of the human body using advanced robotic parts might not be just a fantasy anymore.

1.2 MOTOR NEURON ISSUES

Neurodegeneration is premature damage and dying of motor neurons. This is a rare disease that leads to a loss of a lot of motor activities in our body. Usually it starts off with a weakened grip or weakened breathing patterns. Which is usually an early sign of a worsening condition inside the body.

Motor Neuron Issues are not bound by any age, and that's what makes it all the more deadlier to deal with than any other neurological diseases. This detrimentation can happen regardless of what age a person might be in.

Motor Neurons usually carry the following duties:

- Gripping
- Walking
- Speaking
- Swallowing
- Breathing

Thus the premature ending of these neurons can be lethal to a person's stability both physically and mentally. 20% of MND cases are genetic causes, this is one of the known causes of the disease. People with a family history of MND will make about half of all genetic cases. Those with no family history will account for the other half of genetic cases.

The likelihood of inheriting MND can also rise if frontotemporal dementia runs in the family. The gene's error has an impact on the cells' capacity for survival and normal function. There is a possibility that you could put your child at risk for having MND if you have a hereditary type of the disease. Numerous variables, including age, can influence the likelihood that they will develop the illness.

There is no cure to this condition as of now yet, just specialized treatment to slow down the progression of the disease in the body.

1.3 IoT & VOICE ASSISTANTS

Voice Assistants are the softwares that carry out tasks which are given out through voice commands. One such voice assistant is Alexa which is a multipurpose voice assistant that serves from daily simple tasks to direct a complex robotic arm using voice commands and voice recognition. Alexa is also connected through IoT inside a workspace which makes it all the more easier to communicate with and connect with, from anywhere in the world.

1.4 MOTIVATION

The motivation to choose this as our mini project after observing elderly and disabled people, specifically the ones with motor neuron issues who struggle to have a strong grip over physical objects. It is aimed to design a redundant manipulator bio-inspired by an elephant's trunk with more than or equal to 6 degrees of freedom to freely move and pick items flexibly.

CHAPTER 2

OBJECTIVES

- ➔ To design a 6 degree of freedom end manipulator robotic arm using forward kinematics
- ➔ To design and fabricate a 4 degree of freedom end manipulator robotic arm using 3D printed parts integrated with servo motors.
- ➔ To integrate Alexa's voice recognition into the 4 dof system

CHAPTER 3

METHODOLOGY

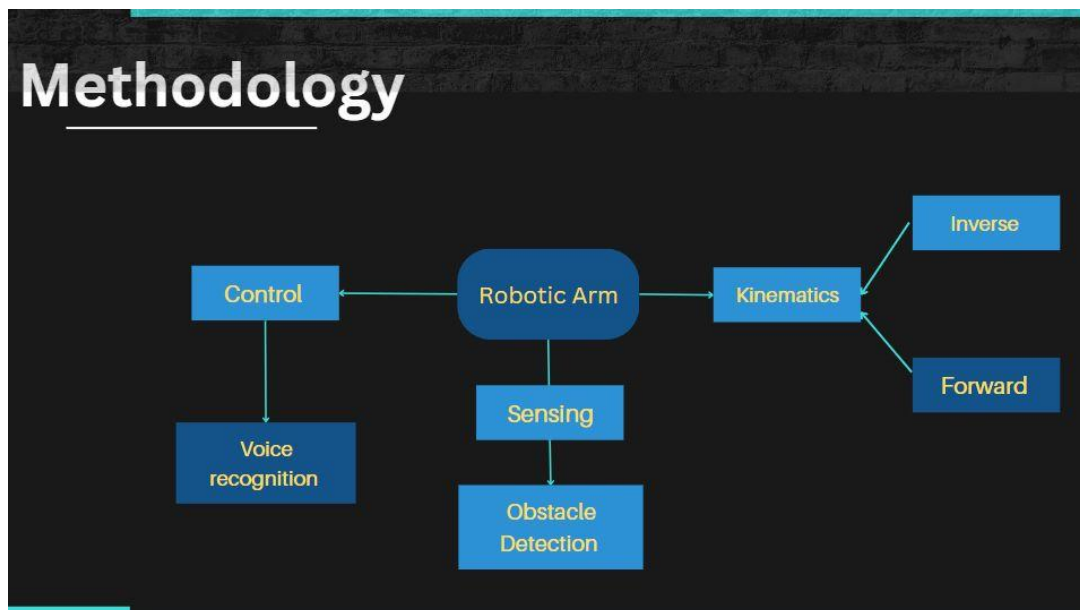


Fig.3.0 Methodology Flowchart

3.1 KINEMATICS

Coming up next we're put into thought in the structure procedure. Electrical actuators DC servo are picked rather than pressure driven and pneumatic actuators due to the little power necessity and its light weight which is reasonable for this plan. Materials utilized for the creation were privately sourced from accessible materials. The materials which will be utilized for the structure will be light in weight to decrease the weight focus on the base and the shoulder. A continuous path controller was chosen (Arduino was used). The torque is completely adjusted by the inertia of the electric motors.

3.1.1 FORWARD KINEMATICS

- Forward kinematics is a term used in robot kinematics to describe the process of using a robot's kinematic equations to determine the position of the end-effector given provided values for the joint parameters. Robotics, video games, and animation all employ the robot's kinematics equations. Inverse kinematics is the procedure that computes the joint parameters necessary to move an end-effector to a specific position.

3.1.2 DENAVIT-HARTENBERG(D-H) NOTATION

The definition of a controller with four joint-interface parameters for each connection and a precise methodology for appointing right-gave orthonormal arrange outlines, one to each connection in an open kinematic chain, was proposed by Denavit &Hartenberg ,so is known as Denavit - Hartenberg (D-H) notation.

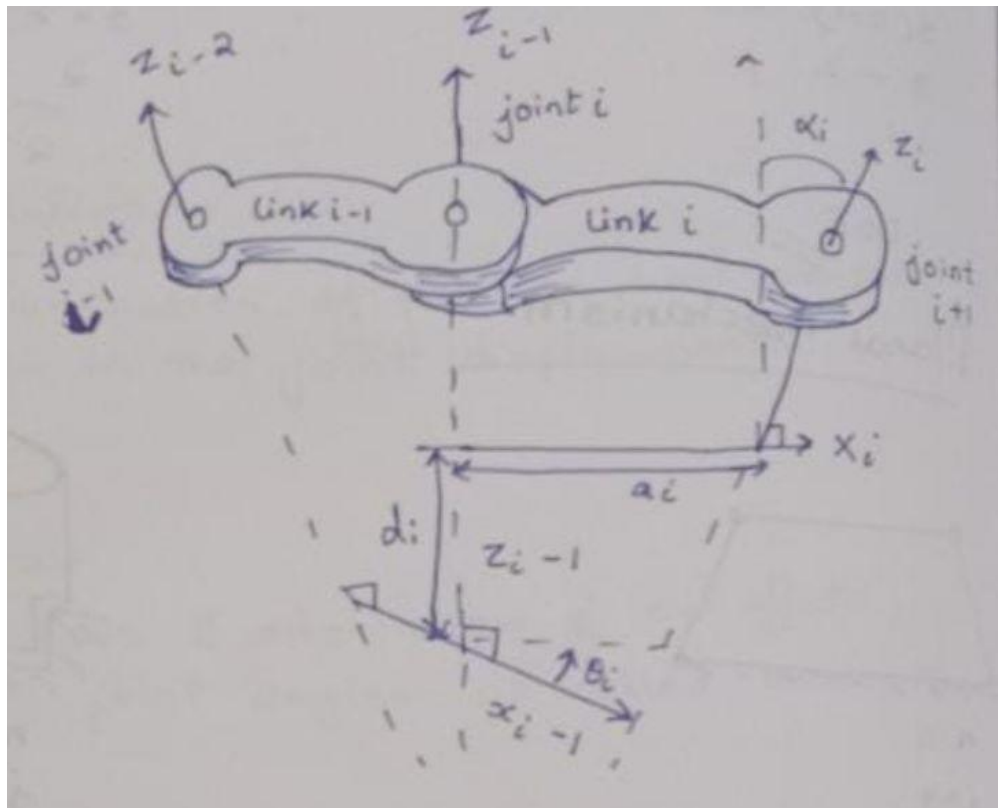
A frame $\{i\}$ is assigned to link i as follows:

- i. The Z_{i-1} lies along the axis of motion of the i th joint.
- ii. The X_i pivot is normal to the Z_{i-1} axis, and pointing far from it.
- iii. The Y_i axis completes the right – handed coordinate system as required.

The DH representation of a rigid link depends on four geometric parameters associated with each links. These four parameters completely describe any revolute or prismatic joint as follows:

- I. Connection length (a_i) – separate estimated along x_i hub from the purpose of convergence of x_i pivot with z_{i-1} hub to the cause of edge $\{i\}$.
- Ii. Link twist (α_i) – angle between z_{i-1} and z axes measured about x_i -axis in the right hand sense.
- Iii. Joint distance(d_i) - distance measured along z_{i-1} axis from the origin of frame $\{i-1\}$ to intersection of x_i axis with the z_{i-1} axis.
- Iv. Joint angle (θ_i) – angle between x_{i-1} and x_i axes measured about the z_{i-1} axis in the right hand sense.

Fig.3.1.2.1 Mathematical model of a manipulator



The transformation matrix for a link i is described as follows:

$$A_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & \alpha_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & \alpha_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

A_1 is the transformation matrix T_1^0 :

$$T_1^0 = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 \cos \alpha_1 & \sin \theta_1 \sin \alpha_1 & \alpha_1 \cos \theta_1 \\ \sin \theta_1 & \cos \theta_1 \cos \alpha_1 & -\cos \theta_1 \sin \alpha_1 & \alpha_1 \sin \theta_1 \\ 0 & \sin \alpha_1 & \cos \alpha_1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and $\alpha = 90$, so T_1^0 will be as follows:

$$T_1^0 = \begin{bmatrix} \cos \theta_1 & 0 & \sin \theta_1 & \alpha_1 \cos \theta_1 \\ \sin \theta_1 & 0 & -\cos \theta_1 & \alpha_1 \sin \theta_1 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

For A2:

$$T_2^1 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & \alpha_2 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & \alpha_2 \sin \theta_2 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For A3:

$$T_3^2 = \begin{bmatrix} \cos \theta_3 & 0 & \sin \theta_3 & \alpha_3 \cos \theta_3 \\ \sin \theta_3 & 0 & -\cos \theta_3 & \alpha_3 \sin \theta_3 \\ 0 & 1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For A4:

$$T_4^3 = \begin{bmatrix} \cos \theta_4 & 0 & -\sin \theta_4 & \alpha_4 \cos \theta_4 \\ \sin \theta_4 & 0 & \cos \theta_4 & \alpha_4 \sin \theta_4 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For A5:

$$T_5^4 = \begin{bmatrix} \cos \theta_5 & 0 & \sin \theta_5 & \alpha_5 & \cos \theta_5 \\ \sin \theta_5 & 0 & -\cos \theta_5 & \alpha_5 & \sin \theta_5 \\ 0 & 1 & 0 & d_5 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

For A6:

$$T_6^5 = \begin{bmatrix} \cos \theta_6 & -\sin \theta_6 & 0 & \alpha_6 & \cos \theta_6 \\ \sin \theta_6 & \cos \theta_6 & 0 & \alpha_6 & \sin \theta_6 \\ 0 & 0 & 1 & d_6 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_6^0 = T_1^0 \times T_2^1 \times T_3^2 \times T_4^3 \times T_5^4 \times T_6^5$$

T_6^0 gives the Forward Kinematic equation.

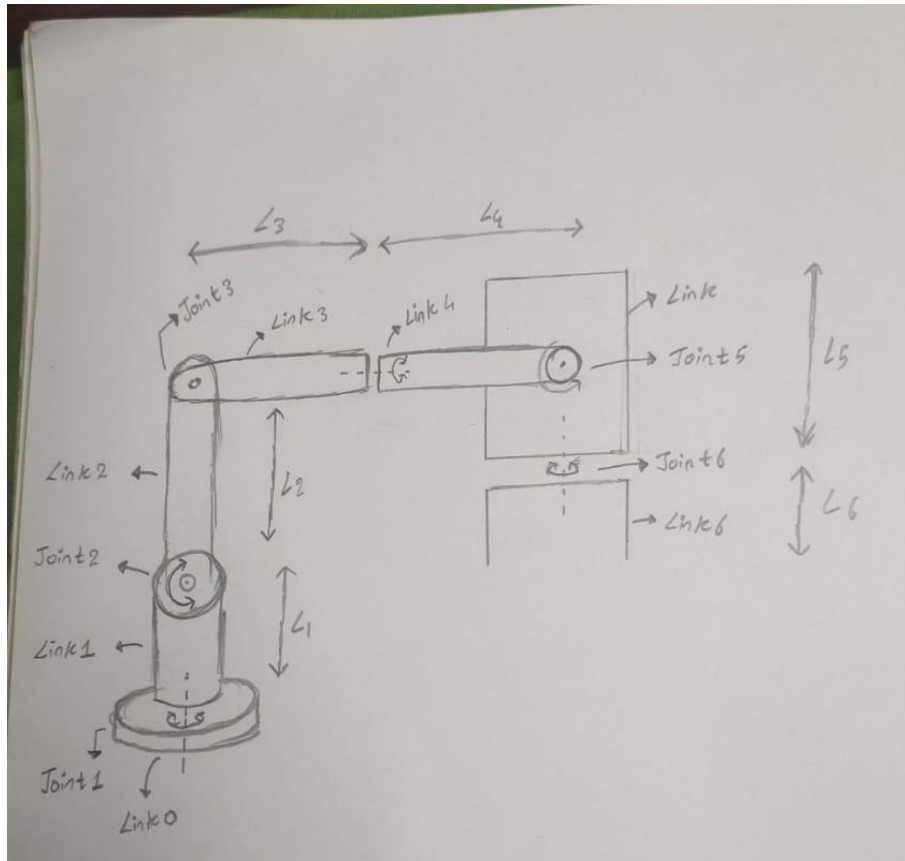


Fig.3.1.2.2. Representation of Links & Joints of 6 DoF

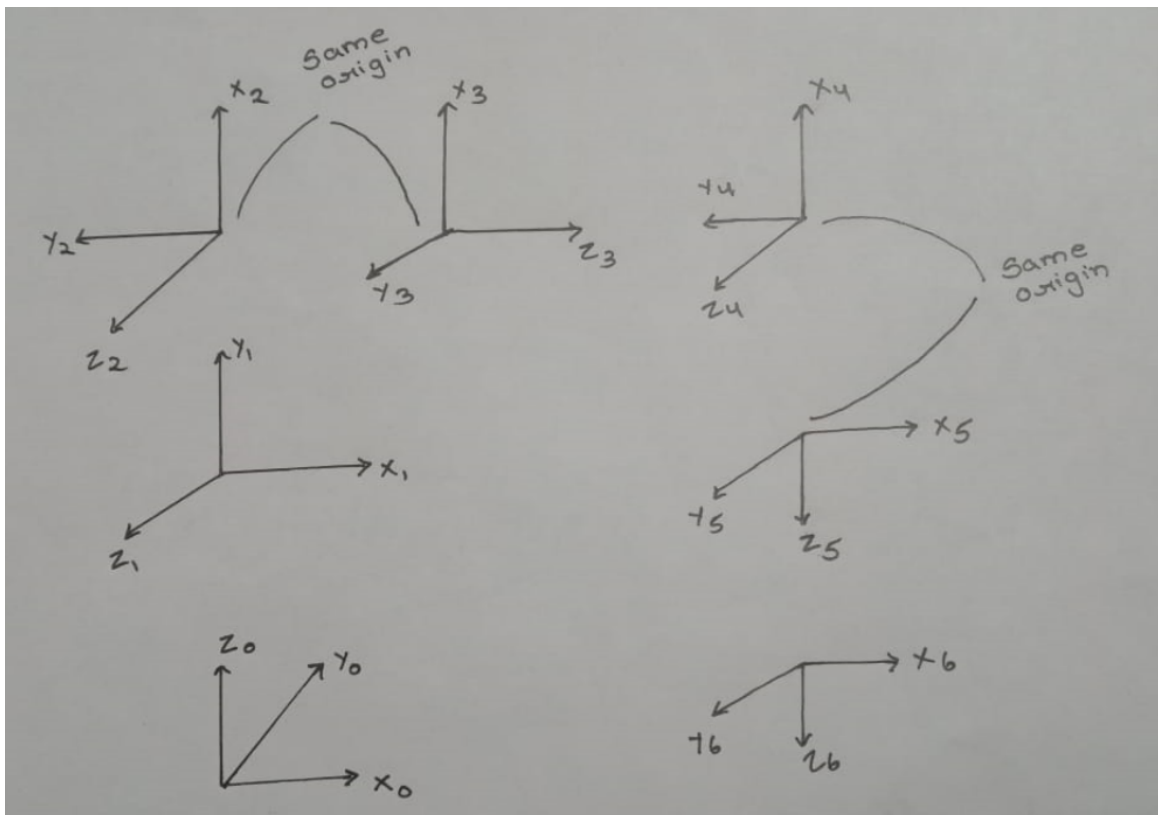


Fig.3.1.2.3 - Frames of 6 DoF

D-H Parameters

Link i	a_i	α_i	d_i	θ_i	Joint variable
1	0	90	L_1	θ_1	θ_1
2	L_2	0	0	$90 + \theta_2$	θ_2
3	0	90	0	θ_3	θ_3
4	0	-90	$L_3 + L_4$	θ_4	θ_4
5	0	90	0	$90 + \theta_5$	θ_5
6	0	0	$L_5 + L_6$	θ_6	θ_6

Table 3.1.2.4. D-H Parameters of 6 DoF

3.2 Design of the Manipulators

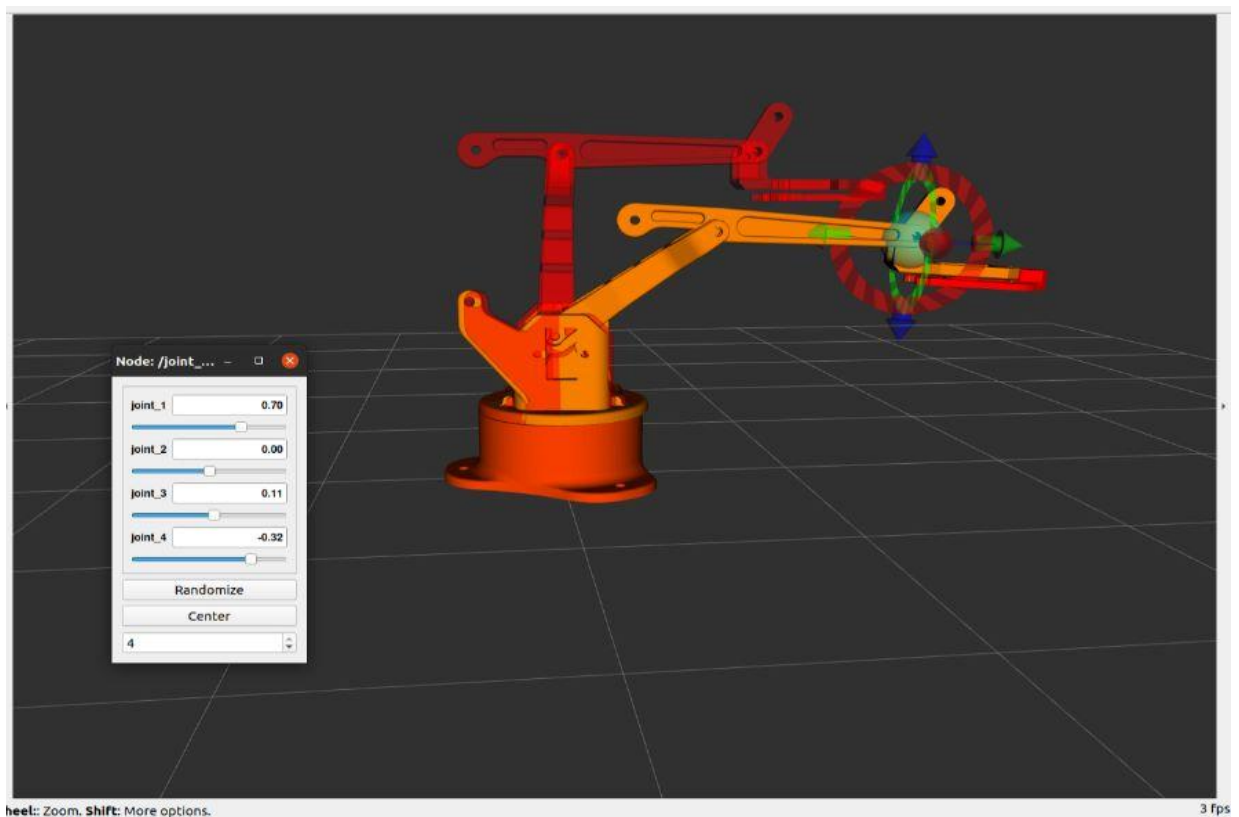


Fig.3.2.1. 4 DoF CAD Design.

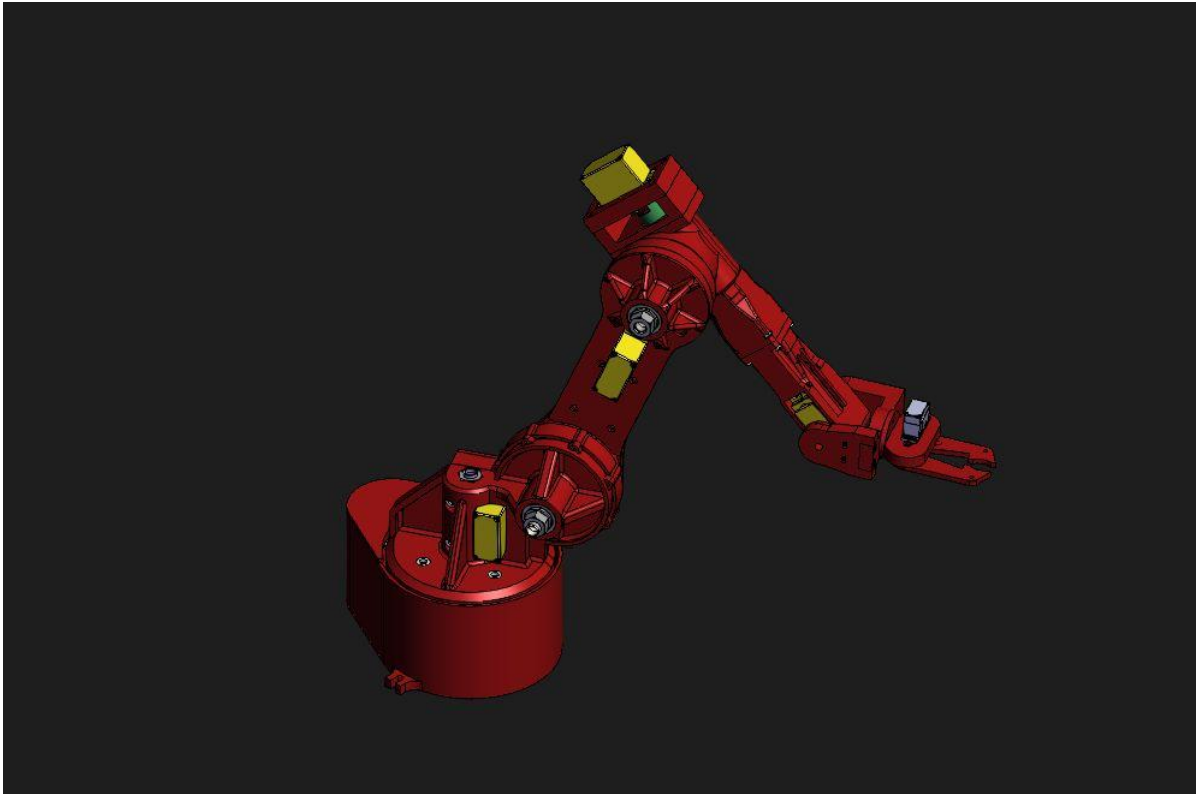


Fig.3.2.2. 6 DoF CAD Design.

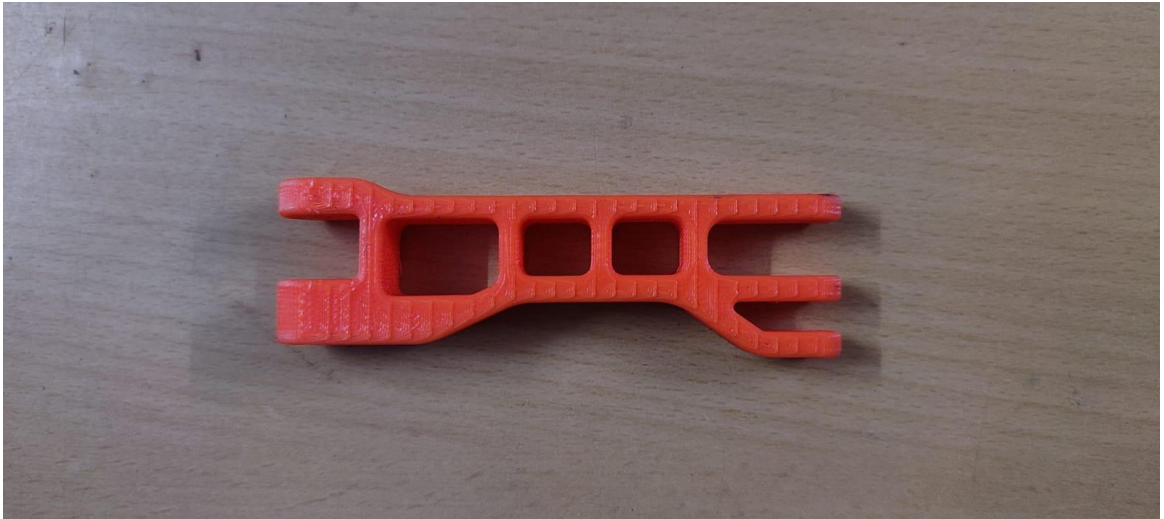
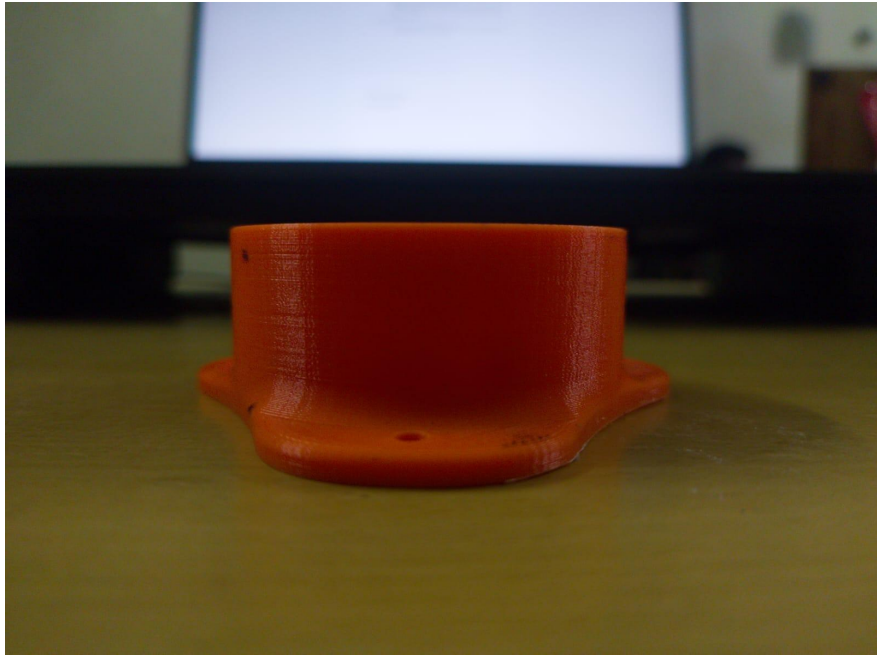
3.3 3D Printing

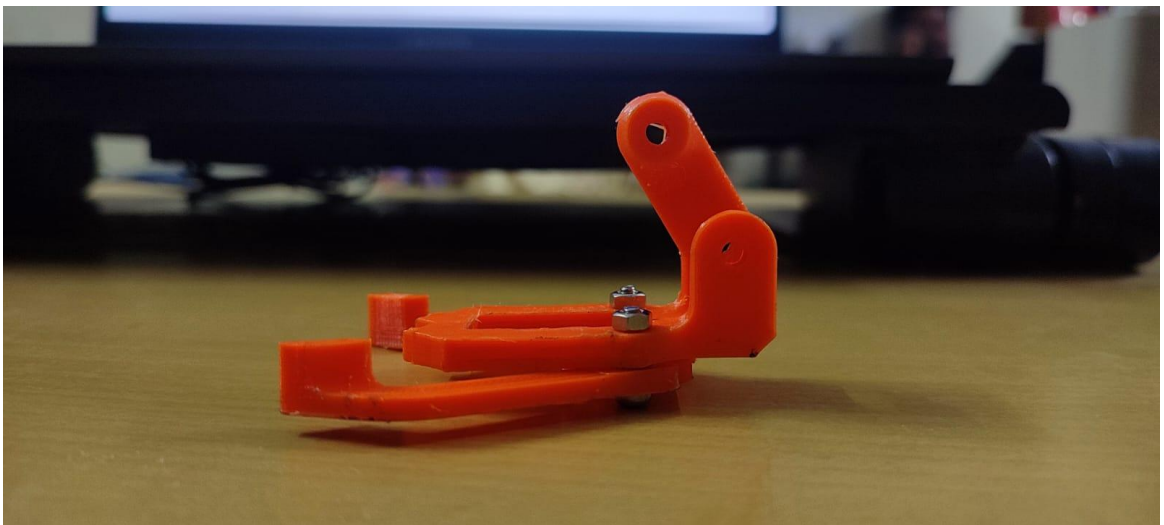
3.3.1 Materials

- For the 3D printing of the 4 DoF end manipulator, it was proposed to use Polylactic Acid polymer for the economic value and strength of the model that carries a minimum weight of 2.5 kgs.
- Polylactic Acid or PLA is a dehydrated polymer with Lactic Acid as its base: $[-C(CH_3)HC(=O)O-]_n$. It is also an organic material that could be a healthier alternative.

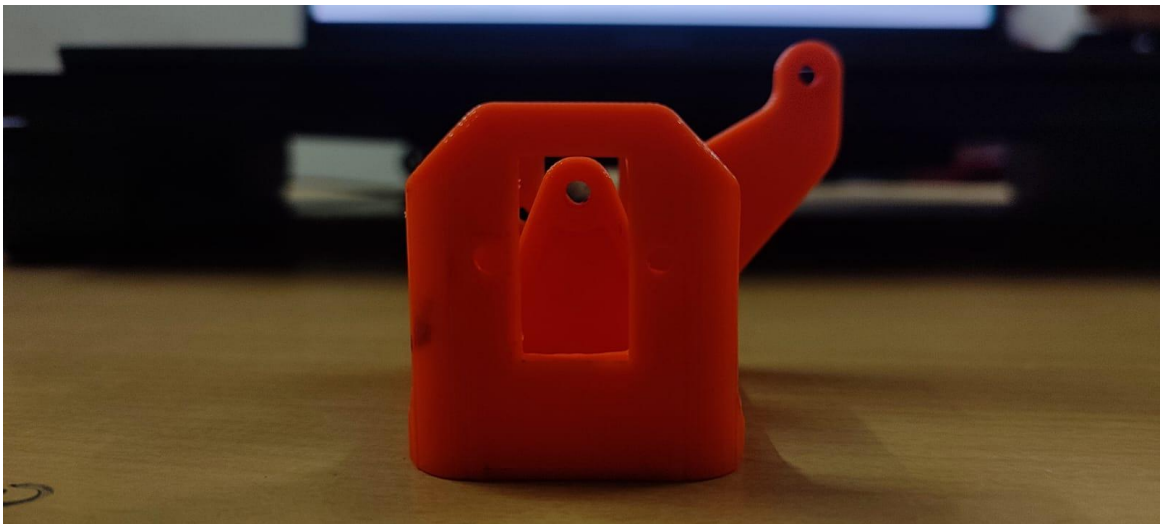
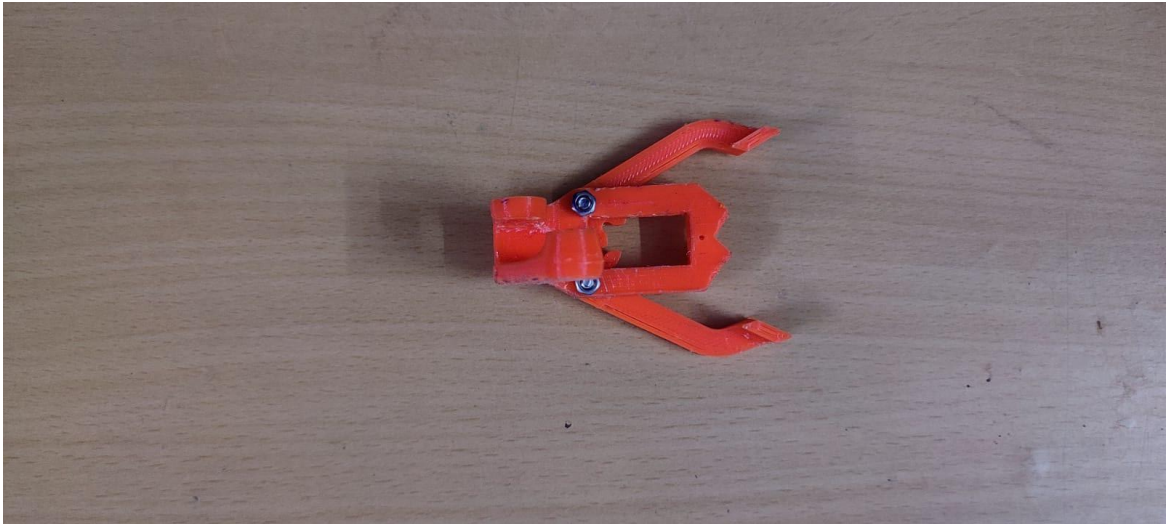
3.3.2 Parts

- The following figures are the pictures taken of the 3D printed parts that were fabricated using PLA at ASK II:











3.4 Actuators

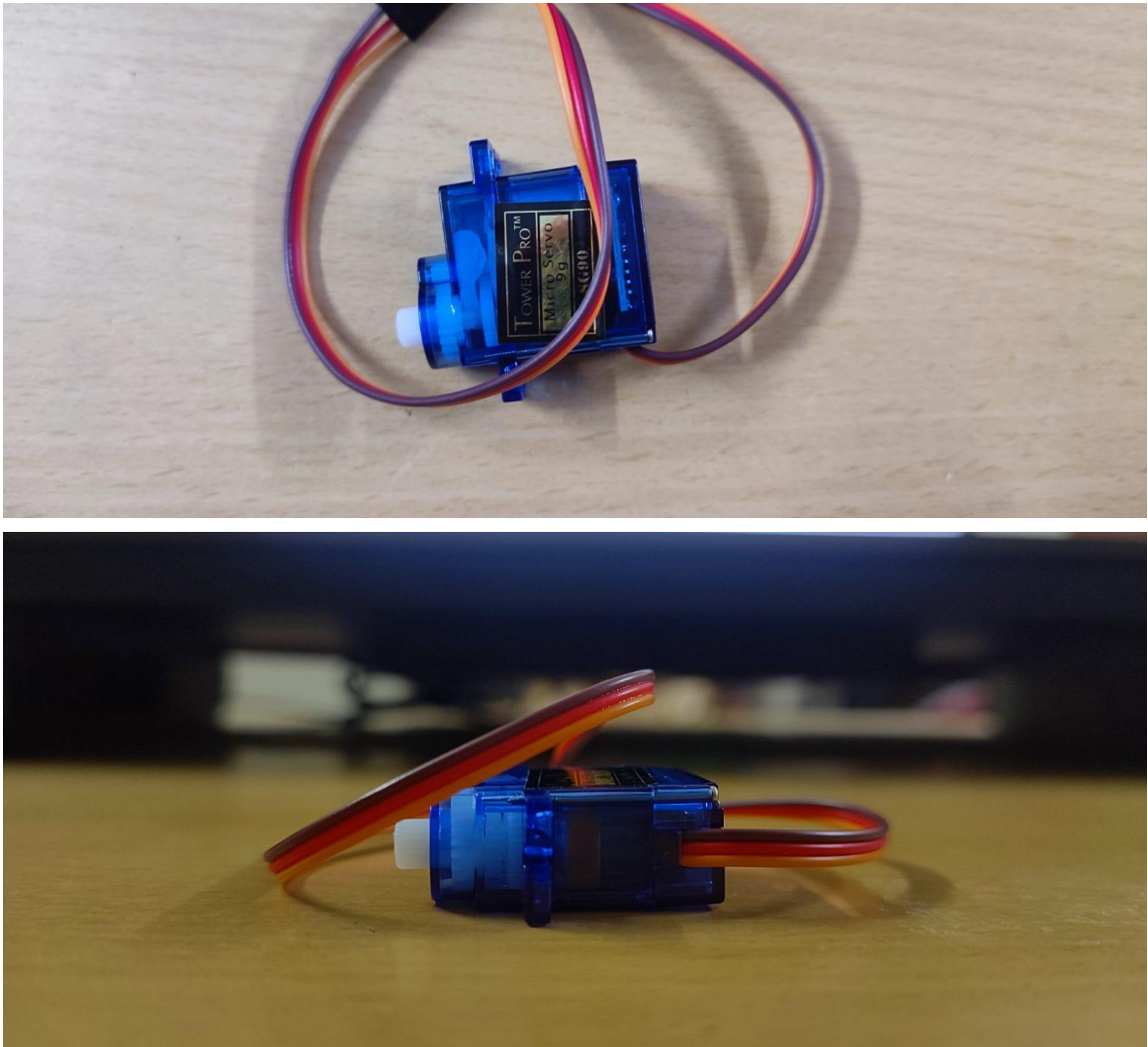


Fig.3.4.1. SG90 Micro Servo Motor 9G

3.5 Gazebo & ROS

```
arduinobot.urdf.xacro - arduinobot_ws - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
ARDUINOBOT_WS
  .vscode
  build
  devel
  src
  arduinobot_control...
    config
    joint_state_cont...
    trajectory_contr...
  include
  launch
  src
  CMakeLists.txt
  package.xml
  arduinobot_descrp...
    launch
    mesh
    rviz
    urdf
  arduinobot.urdf.xacro
  M CMakeLists.txt
  package.xml
  arduinobot_moveit...
  arduinobot_test
  arduinobot_moveit...
  config
  E arduinobot.urdf
  f cartesian_limit...
  f chomp_planning...
  f fake_controllers...
  f gazebo_control...
  f joint_limits.yaml
  f kinematics.yaml
  move_group.launch
  ros_controllers.yaml
arduinobot.urdf.xacro x
joint_state_controller.yaml
arduinobot_interface.h
arduinobot_interface.cpp
AnglesConverter.srv
sim
src > arduinobot_description > urdf > arduinobot.urdf.xacro > robot > @ joint
1 <!-- Links -->
2
3 <!-- Base -->
4 <link name="world"/>
5
6 <link name="base link">
7   <xacro:default_inertial mass="1"/>
8   <visual>
9     <origin rpy="0 0 0" xyz="-0.5 -0.5 0"/>
10    <geometry>
11      <mesh filename="package://arduinobot_description/mesh/basement.STL" scale="0.01 0.01 0.01"/>
12    </geometry>
13  </visual>
14  <collision>
15    <origin rpy="0 0 0" xyz="-0.5 -0.5 0"/>
16    <geometry>
17      <mesh filename="package://arduinobot_description/mesh/basement.STL" scale="0.01 0.01 0.01"/>
18    </geometry>
19  </collision>
20 </link>
21
22 <link name="base plate">
23   <xacro:default_inertial mass="0.1"/>
24   <visual>
25     <origin rpy="0 0 0" xyz="-0.39 -0.39 -0.56"/>
26     <geometry>
27       <mesh filename="package://arduinobot_description/mesh/base_plate.STL" scale="0.01 0.01 0.01"/>
28     </geometry>
29   </visual>
30   <collision>
31     <origin rpy="0 0 0" xyz="-0.39 -0.39 -0.56"/>
32     <geometry>
33       <mesh filename="package://arduinobot_description/mesh/base_plate.STL" scale="0.01 0.01 0.01"/>
34     </geometry>
35   </collision>
36 </link>
37
38 <!-- Useful Variables (Properties) -->
39 <xacro:property name="PI" value="3.14159265359" />
40 <xacro:property name="effort" value="30" />
41 <xacro:property name="velocity" value="10" />
42
43 <!-- Useful Macros -->
44 <xacro:macro name="default_inertial" params="mass">
45   <inertial>
46     <mass value="${mass}" />
47     <inertia ixx="1.0" ixy="0.0" ixz="0.0"
48              iyy="1.0" iyz="0.0"
49              izz="1.0" />
50   </inertial>
51 </xacro:macro>
52
53 <xacro:macro name="default_transmission" params="number">
54   <transmission name="transmission_${number}">
55     <type>transmission_interface/SimpleTransmission</type>
56     <joint name="joint_${number}">
57       <hardwareInterface>hardware_interface/PositionJointInterface</hardwareInterface>
58     </joint>
59     <actuator name="motor_${number}">
60       <hardwareInterface>hardware_interface/PositionJointInterface</hardwareInterface>
61       <mechanicalReduction>1</mechanicalReduction>
62     </actuator>
63   </transmission>
64 </xacro:macro>
65
66 <!-- Forward Drive Arm -->
67 <link name="forward drive arm">
68   <xacro:default_inertial mass="0.1"/>
69   <visual>
70     <origin rpy="0 0 0" xyz="0.19 0.06 -0.08"/>
71     <geometry>
72       <mesh filename="package://arduinobot_description/mesh/forward_drive_arm.STL" scale="0.01 0.01 0.01"/>
73     </geometry>
74   </visual>
75   <collision>
76     <origin rpy="0 0 0" xyz="0.19 0.06 -0.08"/>
77     <geometry>
78       <mesh filename="package://arduinobot_description/mesh/forward_drive_arm.STL" scale="0.01 0.01 0.01"/>
79     </geometry>
80   </collision>
81 </link>
82
83 <!-- Horizontal Arm -->
84 <link name="horizontal arm">
85   <xacro:default_inertial mass="0.1"/>
86   <visual>
87     <origin rpy="${PI/2} 0 ${PI/2}" xyz="0.03 -0.4 -0.06"/>
88     <geometry>
89       <mesh filename="package://arduinobot_description/mesh/horizontal_arm.STL" scale="0.01 0.01 0.01"/>
90     </geometry>
91   </visual>
92   <collision>
93     <origin rpy="${PI/2} 0 ${PI/2}" xyz="0.03 -0.4 -0.06"/>
94     <geometry>
95       <mesh filename="package://arduinobot_description/mesh/horizontal_arm.STL" scale="0.01 0.01 0.01"/>
96     </geometry>
97   </collision>
98 </link>
99
100 <!-- Gripper Left -->
101 <link name="gripper left">
102   <xacro:default_inertial mass="0.01"/>
103   <visual>
104     <origin rpy="0 0 0" xyz="0.04 0.50 -0.1"/>
105     <geometry>
106       <mesh filename="package://arduinobot_description/mesh/left_finger.STL" scale="0.01 0.01 0.01"/>
107     </geometry>
108   </visual>
109   <collision>
110     <origin rpy="0 0 0" xyz="0.04 0.50 -0.1"/>
111     <geometry>
112       <mesh filename="package://arduinobot_description/mesh/left_finger.STL" scale="0.01 0.01 0.01"/>
113     </geometry>
114   </collision>
115 </link>
116
117 <!-- Tool Link -->
118 <link name="tool link"></link>
119
```

```
File Edit Selection View Go Run Terminal Help
arduinobot.urdf.xacro - arduinobot_ws - Visual Studio Code

EXPLORER
ARDUINOBOT_WS
  src
    arduinobot_description
      urdf
        arduinobot.urdf.xacro
        robot
          joint
            child
              kinematics.yaml
            move_group.launch
            ros_controllers.yaml
            arduinobot.urdf.xacro
            joint_state_controller.yaml
            arduinobot_interface.h
            arduinobot_interface.cpp
            AnglesConverter.srv
            sim
            ...
          build
          devel
          src
          launch
          srv
          M MakeList.txt
          package.xml
          arduinobot_descrp...
          launch
          mesh
          Reiz
          urdf
            arduinobot.urdf...
            M MakeList.txt
            package.xml
            arduinobot_mov...
            arduinobot_test
            arduinobot_mov...
            config
            arduinobot.urdf
            cartesian_bvies...

151 <!-- joints -->
152
153 <!-- base joint link 0 -->
154 <!-- joint name="virtual joint" type="fixed" -->
155 <parent link="world"/>
156 <child link="base link"/>
157 <origin xyz="0 0 0" rpy="0 0 0"/>
158 </joint>
159
160 <!-- 1st 180 revolute joint link 1 -->
161 <!-- joint name="joint 1" type="revolute" -->
162 <parent link="base link"/>
163 <child link="base plate"/>
164 <origin xyz="0 0 0.307"/>
165 <axis xyz="0 0 1"/>
166 <limit lower="-(PI/2)" upper="(PI/2)" effort="(effort)" velocity="(velocity)"/>
167 </joint>
168
169 <!-- 2nd 180 revolute joint link 2 -->
170 <!-- joint name="joint 2" type="revolute" -->
171 <parent link="base plate"/>
172 <child link="forward drive arm"/>
173 <origin xyz="0.02 0 0.33"/>
174 <axis xyz="1 0 0"/>
175 <limit lower="-(PI/2)" upper="(PI/2)" effort="(effort)" velocity="(velocity)"/>
176 </joint>
```

```
arduinobot.urdf.xacro - arduinobot_ws - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER
ARDUINOBOT_WS
  src
    arduinobot_description
      urdf
        arduinobot.urdf.xacro
        robot
          joint
            child
              kinematics.yaml
            move_group.launch
            ros_controllers.yaml
            arduinobot.urdf.xacro
            joint_state_controller.yaml
            arduinobot_interface.h
            arduinobot_interface.cpp
            AnglesConverter.srv
            sim
            ...
          build
          devel
          src
          launch
          srv
          M MakeList.txt
          package.xml
          arduinobot_descrp...
          launch
          mesh
          Reiz
          urdf
            arduinobot.urdf...
            M MakeList.txt
            package.xml
            arduinobot_mov...
            arduinobot_test
            arduinobot_mov...
            config
            arduinobot.urdf
            cartesian_bvies...

100 <link name="claw support">
101 <acro:default_inertial mass="0.05"/>
102 <visual>
103 <origin rpy="0 0 (PI/2)" xyz="0 -0.85 -0.15"/>
104 <geometry>
105 <mesh filename="package://arduinobot_description/mesh/claw_support.STL" scale="0.01 0.01 0.01"/>
106 </geometry>
107 </visual>
108 <collision>
109 <origin rpy="0 0 (PI/2)" xyz="0 -0.85 -0.15"/>
110 <geometry>
111 <mesh filename="package://arduinobot_description/mesh/claw_support.STL" scale="0.01 0.01 0.01"/>
112 </geometry>
113 </collision>
114 </link>
115
116 <link name="gripper_right">
117 <acro:default_inertial mass="0.01"/>
118 <visual>
119 <origin rpy="0 0 -(PI/2)" xyz="0.1 0.50 -0.1"/>
120 <geometry>
121 <mesh filename="package://arduinobot_description/mesh/right_finger.STL" scale="0.01 0.01 0.01"/>
122 </geometry>
123 </visual>
124 <collision>
125 <origin rpy="0 0 -(PI/2)" xyz="0.1 0.50 -0.1"/>
126 <geometry>
127 <mesh filename="package://arduinobot_description/mesh/right_finger.STL" scale="0.01 0.01 0.01"/>
128 </geometry>
129 </collision>
130 </link>
```



```

File Edit Selection View Go Run Terminal Help
explorer | kinematics.yaml | move_group.launch | ros_controllers.yaml | arduinobot.urdf.xacro | joint_state_controller.yaml | arduinobot_interface.h | arduinobot_interface.cpp | AnglesConverter.srv | sim | ...
ARDUINOBOT_WS
  > src
    > arduinobot_description
      > urdf
        > arduinobot.urdf.xacro
          > robot
            > joint
              > child
                > kinematics.yaml
                  176 <!-- 3rd 180 revolute joint link 3-->
                  177
                  178 <joint name="joint_3" type="revolute">
                  179   <parent link="forward_drive_arm"/>
                  180   <child link="horizontal_arm"/>
                  181   <origin xyz="0 0 0.8"/>
                  182   <axis xyz="1 0 0"/>
                  183   <limit lower="-${PI/2}" upper="${PI/2}" effort="${effort}" velocity="${velocity}"/>
                  184 </joint>
                  185
                  186 <!-- 4th 180 revolute joint link 4-->
                  187
                  188 <joint name="joint_4" type="revolute">
                  189   <parent link="horizontal_arm"/>
                  190   <child link="claw_support"/>
                  191   <origin xyz="0 0 0.8"/>
                  192   <axis xyz="0 0 1"/>
                  193   <limit lower="-${PI/2}" upper="${PI/2}" effort="${effort}" velocity="${velocity}"/>
                  194 </joint>
                  195
                  196 <joint name="gripper_1" type="revolute">
                  197   <parent link="claw_support"/>
                  198   <child link="gripper_right"/>
                  199   <origin xyz="0.04 0.13 -0.1"/>
                  200   <axis xyz="0 0 1"/>
                  201   <limit lower="-${PI/2}" upper="0" effort="${effort}" velocity="${velocity}"/>
                  202 </joint>
                  203
                  204 <joint name="gripper_2" type="revolute">
                  205   <parent link="claw_support"/>
                  206   <child link="gripper_left"/>
                  207   <origin xyz="0.22 0.13 -0.1"/>
                  208   <axis xyz="0 0 1"/>
                  209   <limit lower="0" upper="${PI/2}" effort="${effort}" velocity="${velocity}"/>
                  210   <mimic joint="gripper_1" multipliers="1" offset="0"/>
                  211 </joint>
                  212
                  213 <joint name="gripper_right_to_tool" type="fixed">
                  214   <parent link="gripper_right"/>
                  215   <child link="tool link"/>
                  216   <origin xyz="0 0 0"/>
                  217 </joint>

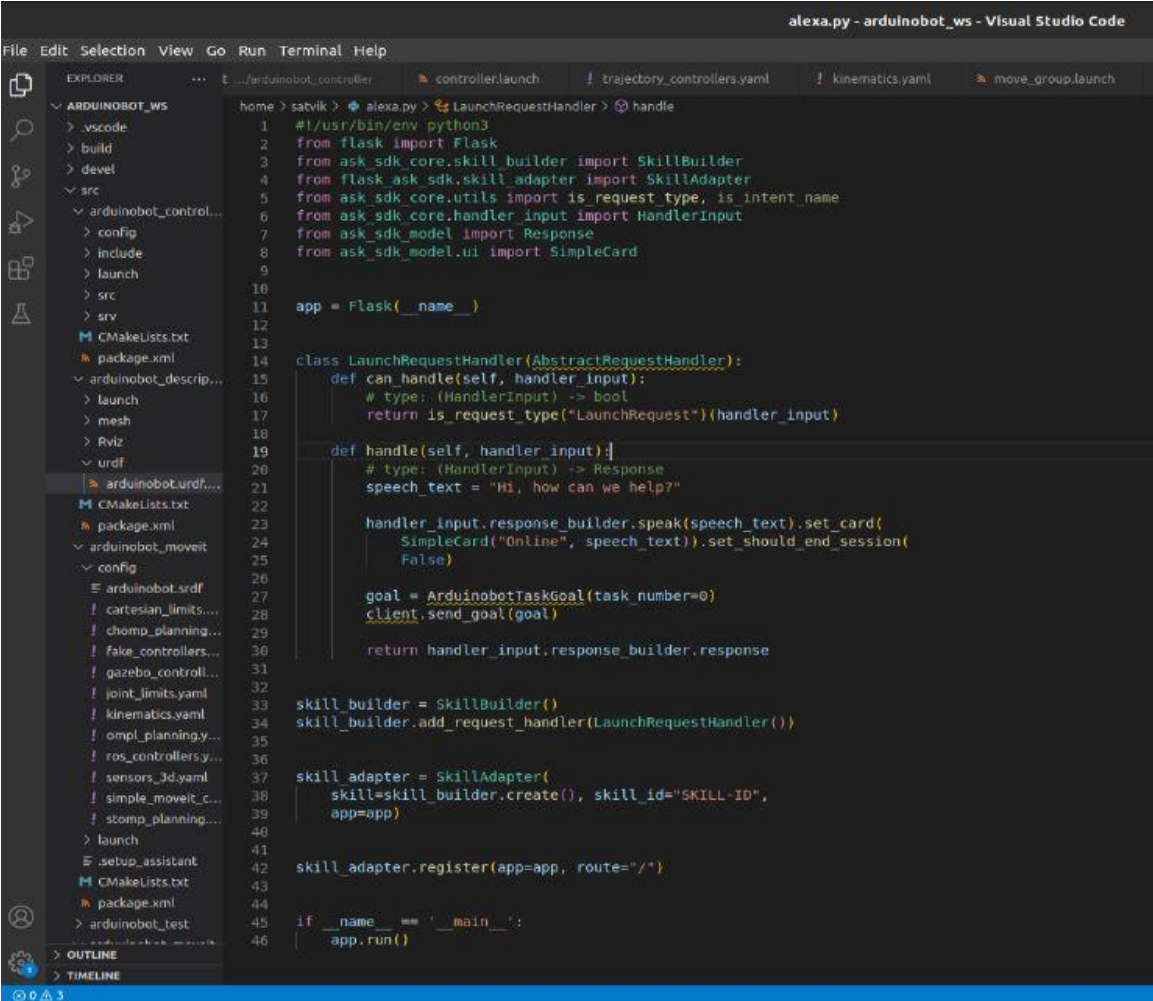
```

```

File Edit Selection View Go Run Terminal Help
explorer | kinematics.yaml | move_group.launch | ros_controllers.yaml | arduinobot.urdf.xacro | joint_state_controller.yaml | arduinobot_interface.h | arduinobot_interface.cpp | AnglesConverter.srv | sim | ...
ARDUINOBOT_WS
  > src
    > arduinobot_description
      > urdf
        > arduinobot.urdf.xacro
          > robot
            > joint
              > child
                > kinematics.yaml
                  217
                  218 <!-- Transmissions-->
                  219
                  220 <xacro:default transmission number="1"/>
                  221 <xacro:default transmission number="2"/>
                  222 <xacro:default transmission number="3"/>
                  223 <xacro:default transmission number="4"/>
                  224
                  225 <!-- gazebo ros control plugin -->
                  226
                  227 <gazebo>
                  228   <plugin name="gazebo_ros_control" filename="libgazebo_ros_control.so">
                  229     <robotNamespace>arduinobot</robotNamespace>
                  230     <robotSimType>gazebo_ros_control/DefaultRobotHWSim</robotSimType>
                  231     <legacyModel>true</legacyModel>
                  232   </plugin>
                  233
                  234   <plugin name="joint_5_mimic_joint_4" filename="libroboticsgroup_opatras_gazebo_mimic_joint_plugin.so">
                  235     <joint>joint_4</joint>
                  236     <mimicJoint>joint_5</mimicJoint>
                  237     <multiplier>-1.0</multiplier>
                  238     <offset>0</offset>
                  239     <maxEffort>${effort}</maxEffort>
                  240     <robotNamespace>arduinobot</robotNamespace>
                  241   </plugin>
                  242 </gazebo>
                  243
                  244 </robot>

```

3.6 Voice Recognition using Alexa



```
alexapy - arduinobot_ws - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
ARDUINOBOT_WS
  .vscode
  build
  devel
  src
    arduinobot_control...
      config
      include
      launch
      src
      srv
    CMakeLists.txt
    package.xml
    arduinobot_descrip...
      launch
      mesh
      Rviz
      urdf
    arduinobot.urdf...
    CMakeLists.txt
    package.xml
    arduinobot_moveit...
      config
        arduinobot.srdf
        cartesian_limits...
        chomp_planning...
        fake_controllers...
        gazebo_controll...
        joint_limits.yaml
        kinematics.yaml
        ompl_planning.y...
        ros_controllers.y...
        sensors_3d.yaml
        simple_moveit_c...
        stomp_planning...
      launch
        setup_assistant
    CMakeLists.txt
    package.xml
    arduinobot_test
  OUTLINE
  TIMELINE
home > satvik > alexapy > LaunchRequestHandler > handle
1  #!/usr/bin/env python3
2  from flask import Flask
3  from ask_sdk_core.skill_builder import SkillBuilder
4  from flask_ask_sdk.skill_adapter import SkillAdapter
5  from ask_sdk_core.utils import is_request_type, is_intent_name
6  from ask_sdk_core.handler_input import HandlerInput
7  from ask_sdk_model import Response
8  from ask_sdk_model.ui import SimpleCard
9
10
11 app = Flask(__name__)
12
13
14 class LaunchRequestHandler(AbstractRequestHandler):
15     def can_handle(self, handler_input):
16         # type: (HandlerInput) -> bool
17         return is_request_type("LaunchRequest")(handler_input)
18
19     def handle(self, handler_input):
20         # type: (HandlerInput) -> Response
21         speech_text = "Hi, how can we help?"
22
23         handler_input.response_builder.speak(speech_text).set_card(
24             SimpleCard("Online", speech_text)).set_should_end_session(
25                 False)
26
27         goal = ArduinobotTaskGoal(task_number=0)
28         client.send_goal(goal)
29
30         return handler_input.response_builder.response
31
32
33 skill_builder = SkillBuilder()
34 skill_builder.add_request_handler(LaunchRequestHandler())
35
36
37 skill_adapter = SkillAdapter(
38     skill=skill_builder.create(), skill_id="SKILL-ID",
39     app=app)
40
41
42 skill_adapter.register(app=app, route="/")
43
44
45 if __name__ == '__main__':
46     app.run()
```

Fig.3.6.1. Code running behind Alexa for Voice Recognition.

```

File Edit Selection View Go Run Terminal Help
EXPLORER t .../arduinobot_controller controller.launch trajectory_controllers.yaml kinematics.yaml
ARDUINOBOT_WS
  .vscode
  build
  devel
  src
  arduinobot_control...
    config
    include
    launch
    src
    srv
    CMakeLists.txt
    package.xml
  arduinobot_descrip...
    launch
    mesh
    Rviz
    urdf
    arduinobot.urdf...
    CMakeLists.txt
    package.xml
  arduinobot_moveit
    config
    arduinobot.srdf
    cartesian_limits...
    chomp_planning...
    fake_controllers...
    gazebo_controlli...
    joint_limits.yaml
    kinematics.yaml
    ompl_planning.y...
    ros_controllers.y...
    sensors_3d.yaml
    simple_moveit_c...
    stomp_planning...
    launch
    .setup_assistant
    CMakeLists.txt
    package.xml
  arduinobot_test
    OUTLINE
    TIMELINE
  0 1

home > satvik > alexapy > ...
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128

def handle(self, handler_input):
    # type: (HandlerInput) -> Response
    speech_text = "Hi, I am ready"

    handler_input.response_builder.speak(speech_text).set_card(
        SimpleCard("Wake", speech_text)).set_should_end_session(
            True)

    goal = ArduinobotTaskGoal(task_number=0)
    client.send_goal(goal)

    return handler_input.response_builder.response

class AllExceptionHandler(AbstractExceptionHandler):

    def can_handle(self, handler_input, exception):
        # type: (HandlerInput, Exception) -> bool
        return True

    def handle(self, handler_input, exception):
        # type: (HandlerInput, Exception) -> Response

        speech = "Hmm, I don't know that. Can you please say it again?"
        handler_input.response_builder.speak(speech).ask(speech)
        return handler_input.response_builder.response

skill_builder = SkillBuilder()
skill_builder.add_request_handler(LaunchRequestHandler())
skill_builder.add_request_handler(PickIntentHandler())
skill_builder.add_request_handler(SleepIntentHandler())
skill_builder.add_request_handler(WakeIntentHandler())
skill_builder.add_exception_handler(AllExceptionHandler())

skill_adapter = SkillAdapter(
    skill=skill_builder.create(), skill_id="SKILL-ID",
    app=app)

skill_adapter.register(app=app, route="/")

if __name__ == '__main__':
    app.run()

```

Fig.3.6.2. Voice Interaction Model with Alexa.

CHAPTER 4

RESULTS & DISCUSSIONS

4.4 Working Designs

The following screenshots show the working designs of the 4 DoF End Manipulator on simulation:

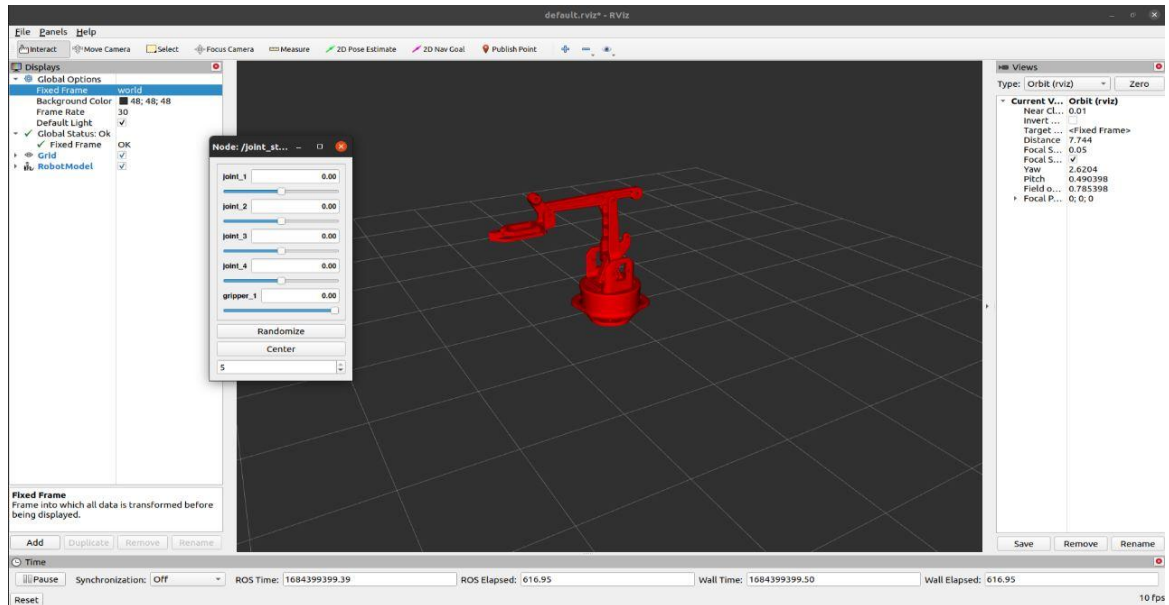


Fig.4.4.1. 4 DoF Home Position

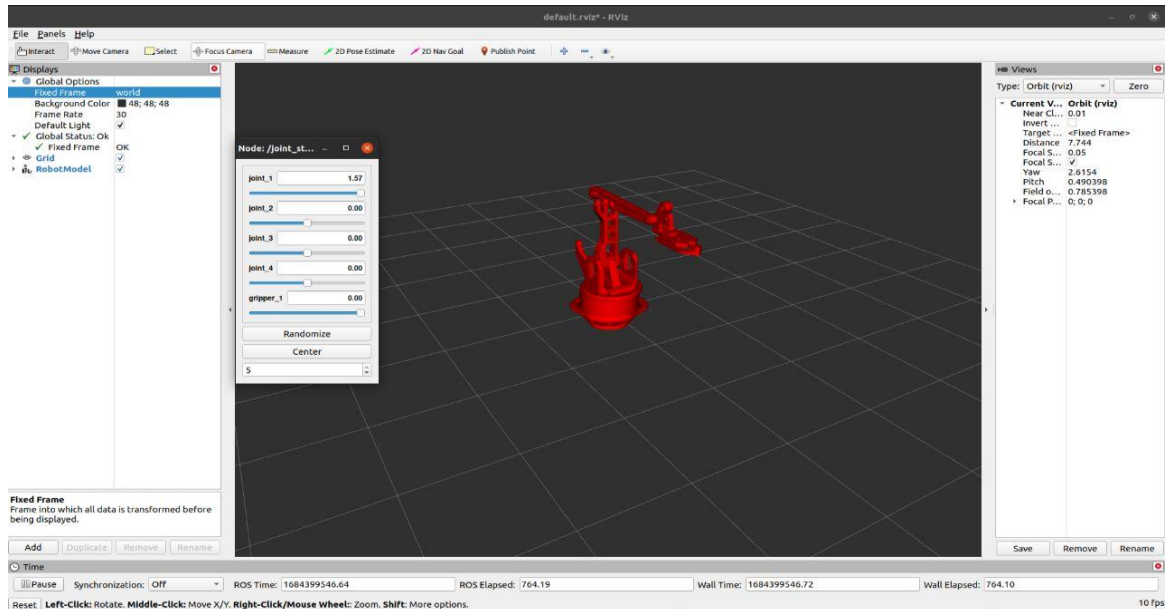


Fig.4.4.2. 4 DoF - Joint 1 180 degree movement

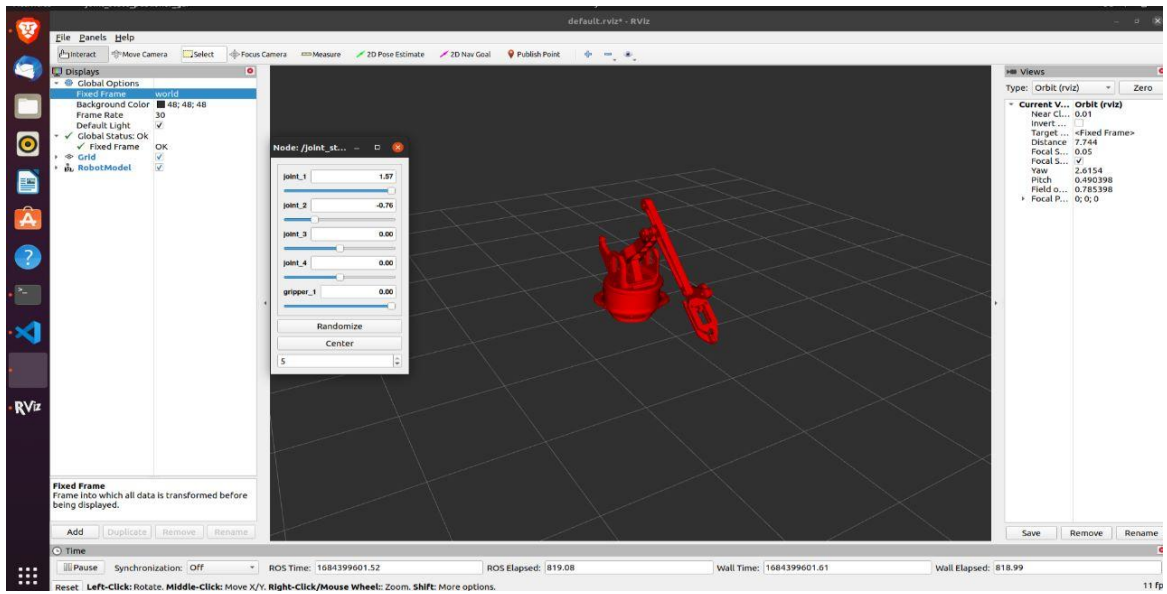


Fig.4.4.3. 4 DoF - Joint 2 - bending to pick up objects

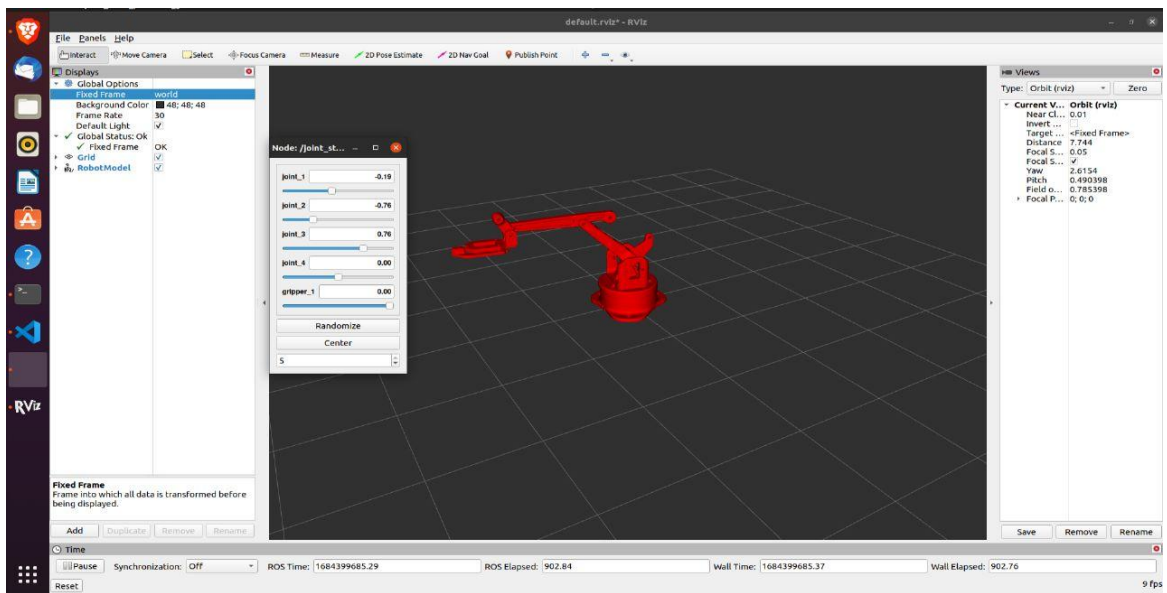


Fig. 4.4.4. 4 DoF - Joint 3 - Placing Objects

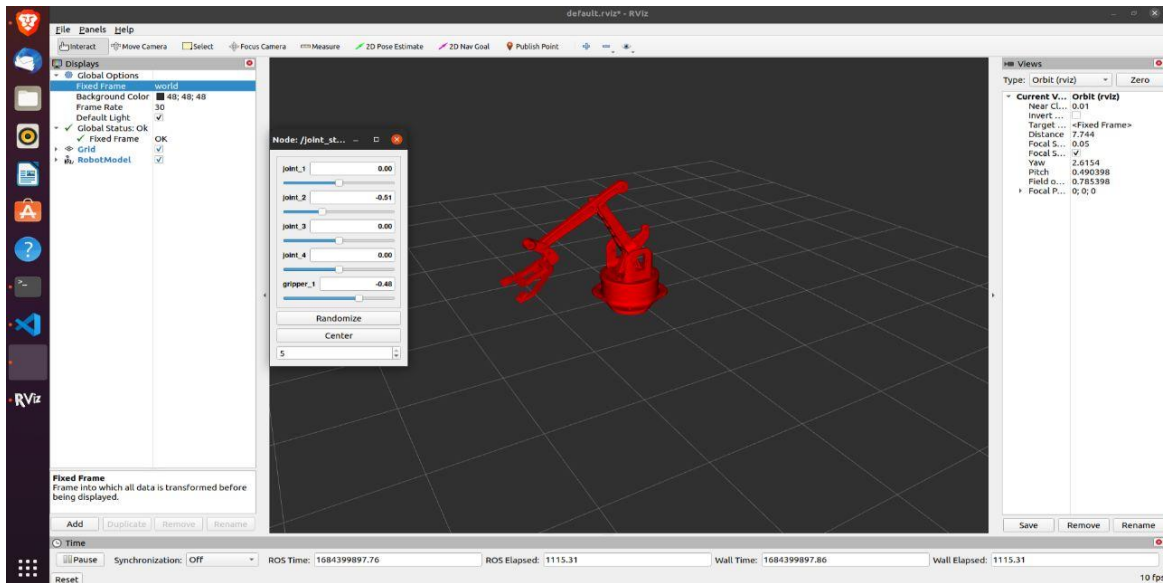


Fig.4.4.5. 4 DoF - Joint 4 - Movement

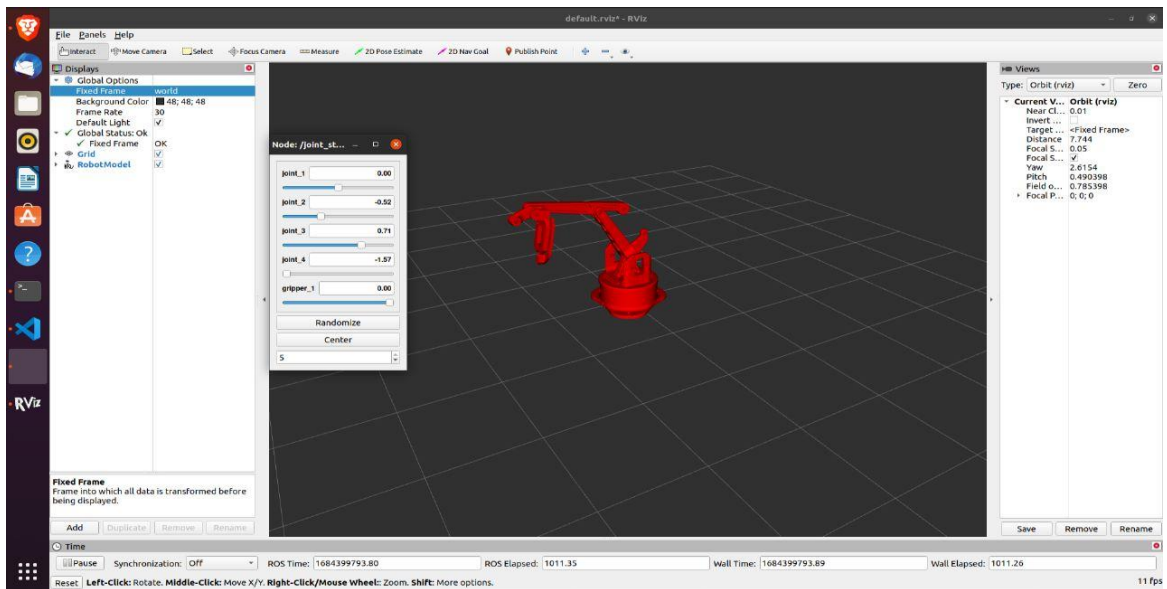


Fig.4.4.6. 4 DoF - Joint 4 - Dropping object.

CHAPTER 5

CONCLUSIONS & FURTHER WORK

5.1 CONCLUSION

The robotic arm's design and construction have been successfully finished. Solidworks is used to generate the robotic arm's line diagram. The manipulator's design has been identified. For various loads, the torque acting at each joint is determined. On the basis of the torque, the power is predicted. Depending on the required torque, the servo is chosen for each joint. Microsoft's C++ programming language is being used to construct the robotic arm's servo controller and control software. The pick-and-place robot was used in the studies, and the outcomes were quite pleasing. The joints can be powered by motors with higher torque ratings to keep the robotic arm in place even when there is no electric supply. The robotic arm can implement object recognition and collision avoidance by adding proximity sensors.

5.2 FURTHER WORK

For further work, We have looked into possibly applying the design in industrial applications and reduce the costs to build and maintain an end effector manipulator. This could delve into slight economical debate on how good 3D printed materials are at holding and working with heavier materials.

The robot so programmed for pick and place operation can be made versatile and more efficient by providing the feedback and making it work on its own than any human interventions. It can be made possible by an image processing tool interfaced with this Arduino. The features that can be added on to improve its efficiency, make it operate on its own thought without any human intervention are line follower, wall hugger, obstacle avoider, metal detector, bomb diffuser etc.

CHAPTER 6

REFERENCES

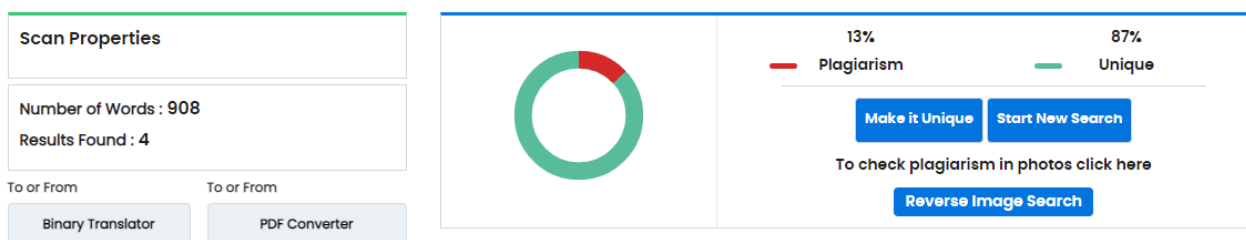
- [1] **Priyambada Mishra, Riki Patel, Trushit Upadhyaya, Arpan Desai** “*Review of Development Of Robotic Arm Using Arduino UNO*”, International Journal on Recent Researches in Science, Engineering and Technology, ISSN: 2348-3105 Volume 5, Issue 5, May 2017.
- [2] **Puran Singh, Anil Kumar, Mahesh Vashishth** “*Design of Robotic Arm with Gripper and End effector for spot welding*”, Universal Journal of Mechanical Engineering 1(3); 92-97, 2013, DOI: 10.13189/ujme,2013.010303.
- [3] **Areepen Sengsalonga , Nuryono Satya Widodo** “*ObjectMoving Robot Arm based on Color*”, Signal and Image Processing Letters, Vol.1., No.3, November 2019, pp. 13-19 ISSN 2714-6677.
- [4] **V. K. Banga, Jasjit Kaur, R. Kumar, Y. Singh** “*Modeling and Simulation of Robotic Arm Movement using Soft Computing*”, International Journal of Mechanical And Mechatronics Engineering, Vol:5, No:3,2011.
- [5] **Kemal Oltun Evliyaoğlu1, Meltem Elitaş** “*Design and Development of a Self-Adaptive,Reconfigurable and LowCost Robotic Arm*” Department of Mechatronics Engineering, Sabanci University. Üniversite St. No:27, 34956 Tuzla/İstanbul Turkey1
- [6] **Anughna N, Ranjitha V, Tanuja G** “*Design and Implementation of Wireless Robotic Arm Model using Flex and Gyro Sensor*”, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-5, January 2020.
- [7] **Kurt E. Clothier and Ying Shang** “*A Geometric Approach for Robotic Arm Kinematics with Hardware Design, Electrical Design, and Implementation*” Hindawi Publishing CorporationJournal of Robotics Volume 2010, Article ID 984823, 10 pages doi:10.1155/2010/984823
- [8] **Gurudu Rishank Reddy and Venkata Krishna Prashanth Eranki** “*Design and Structural Analysis of a Robotic Arm*” Blekinge Institute of Technology, Karlskrona, Sweden, 2016, ISRN: BTH-AMT-EX--2016/D06—SE.
- [9] **Dr. Bindu A Thomas, Stafford Michahial, Shreeraksha.P, Vijayashri B Nagvi, Suresh M** “*Industry Based Automatic Robotic Arm*”, International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 11, May 2013, ISSN: 2277-3754 ISO 9001:2008 Certified

[10] **A Hui Wei and B Yang Chen**, “*Robotic object recognition and grasping with a natural background*”, International Journal of Advanced Robotic Systems, March-April 2020: 1–17, DOI: 10.1177/1729881420921102.

CHAPTER 7

APPENDIX

7.1 DUPLICHECKER PLAGIARISM TEST REPORT



PICK AND PLACE ROBOTIC ARM FOR THE ELDERLY & DISABLED

ReportsubmittedtotheSASTRADeemedtobeUniversity
astherequirementofthecourse

MEC300/MCT300MINIPROJECT

Submittedby

JAMALAPURAMPRAHARSHITHKASHYAP
(RegNo:124009054)
ANNADANAMSATVIK
(RegNo:124009118)
RUCHIGUPTA
(RegNo:124012036)

Similarity 25%

[Division of MedicaidDivision of Medicaid](#)

DOM requires that an agency seeking approval as a Medicaid Home and Community-Based Waiver provider for the Elderly & Disabled Waiver be established and in ...Jul 1, 2009 — DOM requires that an agency seeking approval as a Medicaid Home and Community-Based Waiver provider for the Elderly & Disabled Waiver be ...

<https://sos.ms.gov/ACProposed/00015239b.pdf>

Similarity 13%

[S. Jeyaraj](#)

Assistant Professor, School of Mechanical Engineering, SASTRA University – Cited by 5
5 – Electrodeposition – Composite coatings

<https://scholar.google.com/citations?user=60YiFcwAAAAJ>

Similarity 9%

["Charge breeding of radioactive isotopes at the CARIBU ...](#)

by RC Vondrasek · 2018 — We express our gratitude to Dr. S. Kondrashev, Dr. J. Alessi, Dr. E. Beebe, and Dr. A. Pikin for their significant contributions during the design, ...

<https://aip.scitation.org/doi/abs/10.1063/1.5063798>

Similarity 4%

[D-Glucose- 13 C 6 ,1,2,3,4,5,6,6-d 7D -Glucose-13C6,1,2,3,4,5,6,6-d7 13C: 99](#)

PICK AND PLACE ROBOTIC ARM FOR THE ELDERLY & DISABLED

ReportsubmittedtotheSASTRADeemedtobeUniversity
astherequirementofthecourse

MEC300/MCT300MINIPROJECT

Submittedby

JAMALAPURAMPRAHARSHITHKASHYAP
(RegNo:124009054)
ANNADANAMSATVIK
(RegNo:124009118)
RUCHIGUPTA
(RegNo:124012036)

Similarity 13%

[S. Jeyaraj](#)

Assistant Professor, School of Mechanical Engineering, SASTRA University – Cited by 5
5 – Electrodeposition – Composite coatings

<https://scholar.google.com/citations?user=60YiFcwAAAAJ>

Similarity 9%

["Charge breeding of radioactive isotopes at the CARIBU ...](#)

by RC Vondrasek · 2018 — We express our gratitude to Dr. S. Kondrashev, Dr. J. Alessi, Dr. E. Beebe, and Dr. A. Pikin for their significant contributions during the design, ...

<https://aip.scitation.org/doi/abs/10.1063/1.5063798>

Similarity 4%

[D-Glucose- 13 C 6 ,1,2,3,4,5,6,6-d 7D -Glucose-13C6,1,2,3,4,5,6,6-d7 13C: 99 atom , ...](#)

D -Glucose- 13 C 6 ,1,2,3,4,5,6,6-d 7 . undefined. Synonyms: D -Glucose- 13 C 6 ,C-d 7 , Dextrose- 13 C 6 ,C-d 7 . CAS 201417-01-8. Molecular Weight 193.15.D-Glucose-13C6,1,2,3,4,5,6,6-d ; Molecular Weight: 193.15 ; MDL number: MFCD00274634 ; PubChem Substance ID: 329758091 ; NACRES: NA.12 ; Hazard Statements. H413 ...

<https://www.sigmaaldrich.com/US/en/substance/dglucose13c61234566d719315201417018>