Design And Development Of Computer-vision Based Controlled Humanoid Robot Arm

ME 3261: Mechatronic System Design Project

Department of Mechanical Engineering,

University of Moratuwa.

Name: De silva P.K.V

Index No: 200112A

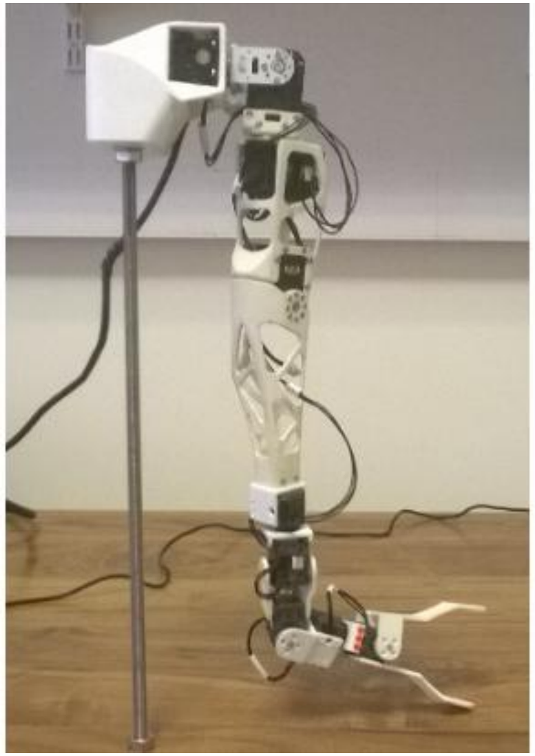
OUTLINE

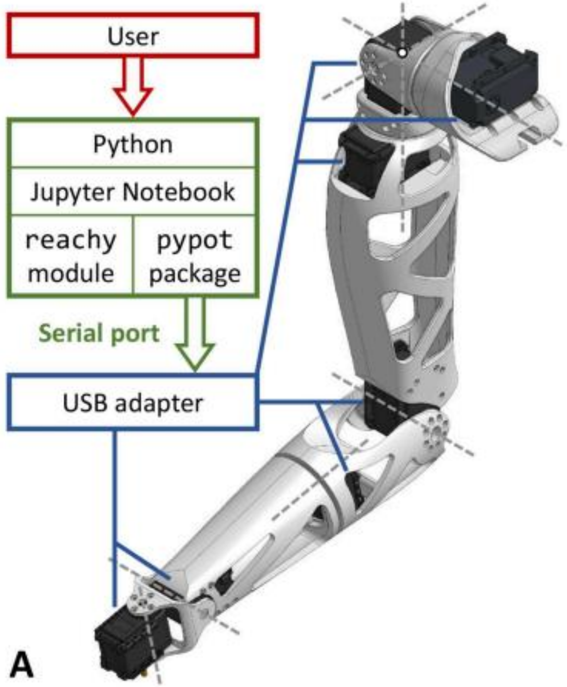
- Introduction
- Concept
- Mechanical design
- Stress analysis
- Joint torque calculations
- Kinematics
- Implementation
- Simulation
- Prototype

INTRODUCTION

- Assistive robotic systems consisting of a robot arm mounting on a moving platform become increasingly important nowadays.
- My objective is to design and develop a 3-DOF humanoid robot arm controlled using computer vision to accomplish specific tasks.

Literature survey





- Poppy robot's right arm
  - 3D printed plastic skeleton parts
  - Light weight and high payload ratio
  - 7 Degrees of Freedom

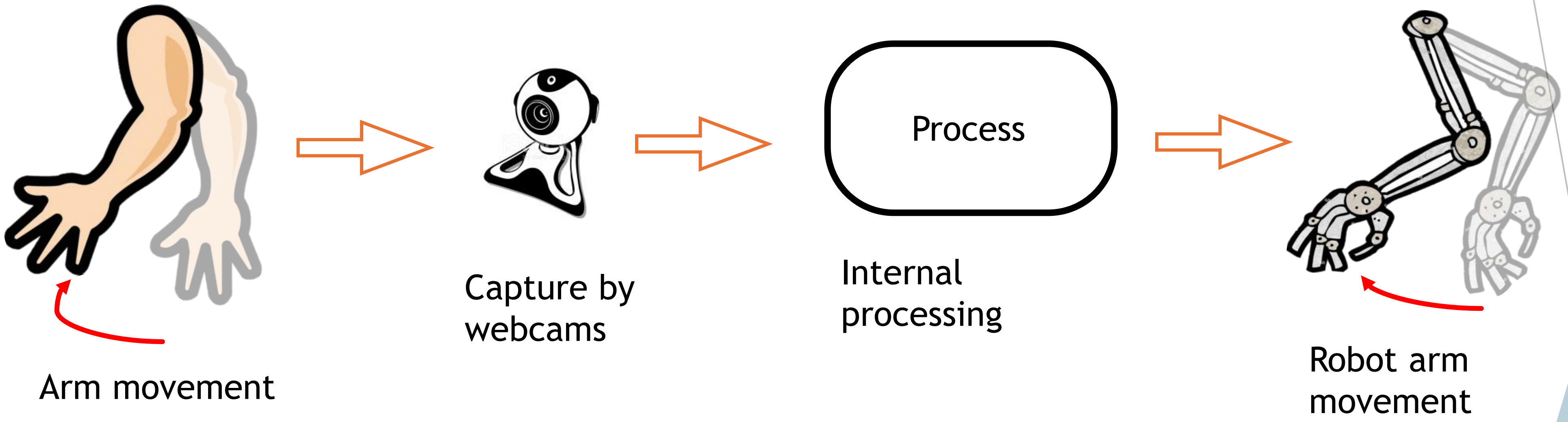
- Reachy robot's right arm
  - 3D printed plastic skeleton parts
  - Advancement of poppy robot's arm
  - 7 Degrees of Freedom

CONCEPT

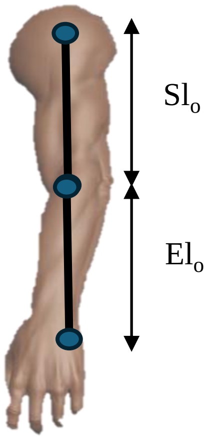
- Robotic manipulators can be controlled using various methods, each suited for specific tasks, levels of complexity, and environments.
  1. Position control (point-to-point control)
  2. Trajectory control (continuous path control)
  3. Force/Impedance control
  4. Learning based control (Reinforcement learning)
  5. BCI(Brain Computer Interface) methods
  6. Visual servoing

     The manipulator’s movements are controlled based on feedback from a vision system. The robot adjusts its position and orientation based on real-time camera data.

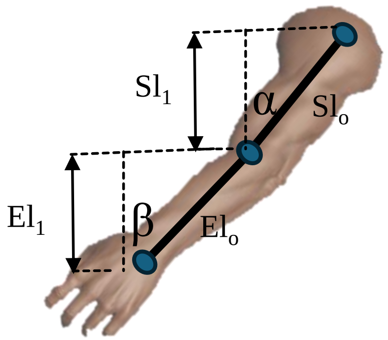
- My proposed approach involves employing visual feedback for the real-time control of a humanoid robotic arm, utilizing a visual servoing method to enhance precision and adaptability during task execution.



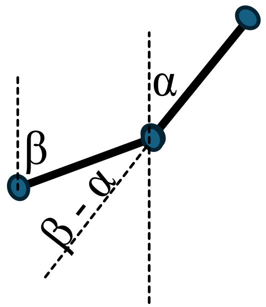
- Teleoperation is one of the main advantage of this method.
- This method enables highly precise control, making it suitable for tasks requiring considerable accuracy.



Initial lengths



if

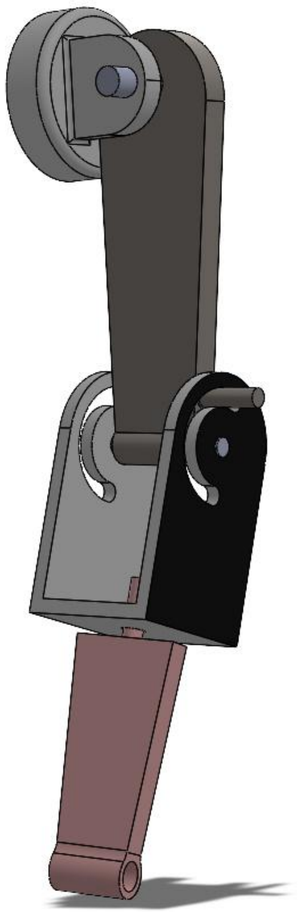


current lengths

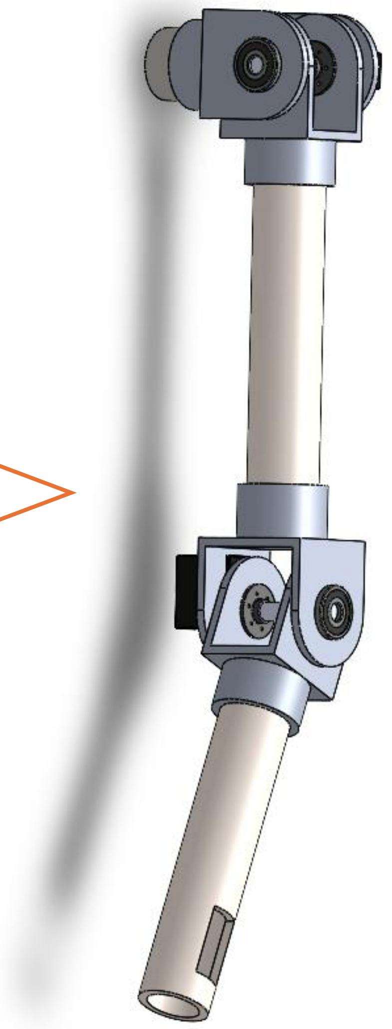
$$\alpha = \cos^{-1}\left(\frac{Sl_1}{Sl_0}\right) \qquad \beta = \cos^{-1}\left(\frac{El_1}{El_0}\right)$$

- Shoulder angle is α
- Compare α and β,
    - If, $\alpha = \beta$ → Elbow has not fold
    - If, $\beta > \alpha$ → Elbow has fold and elbow angle is β - α

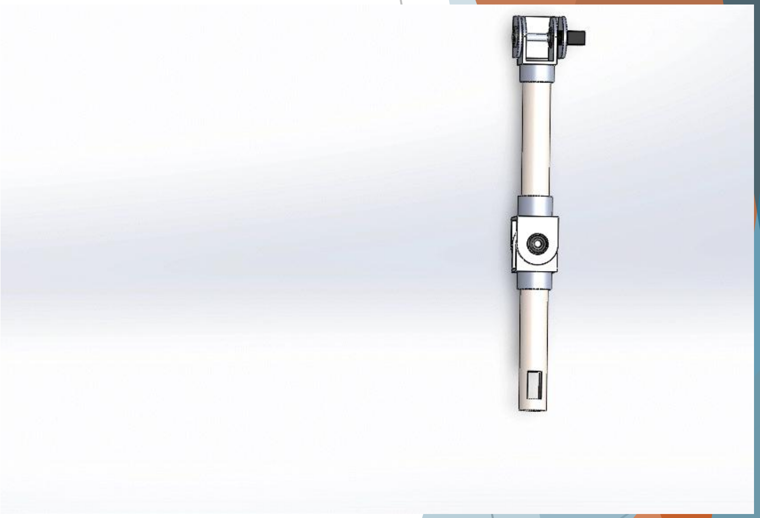
MECHANICAL DESIGN



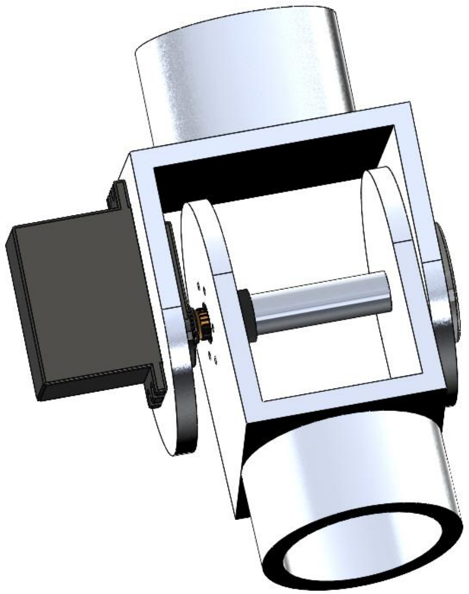
First model

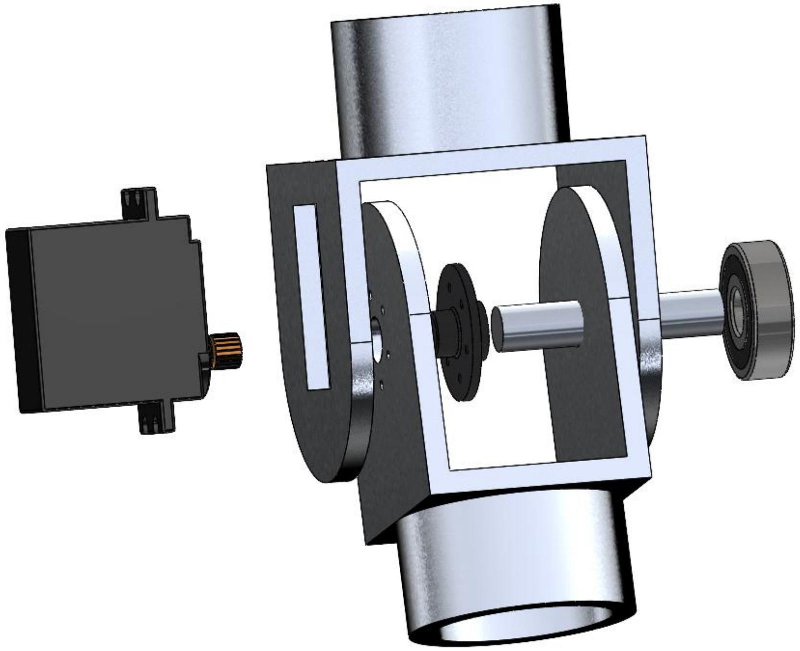


Developed model after stress analysis

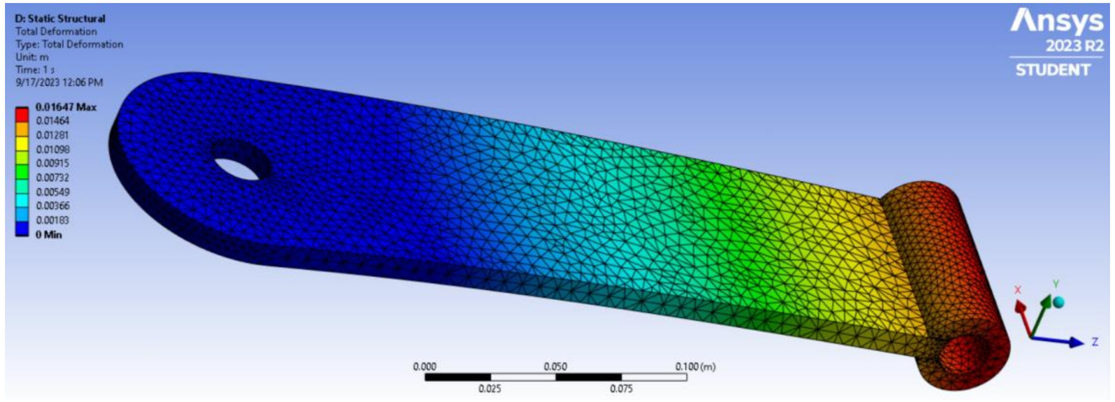


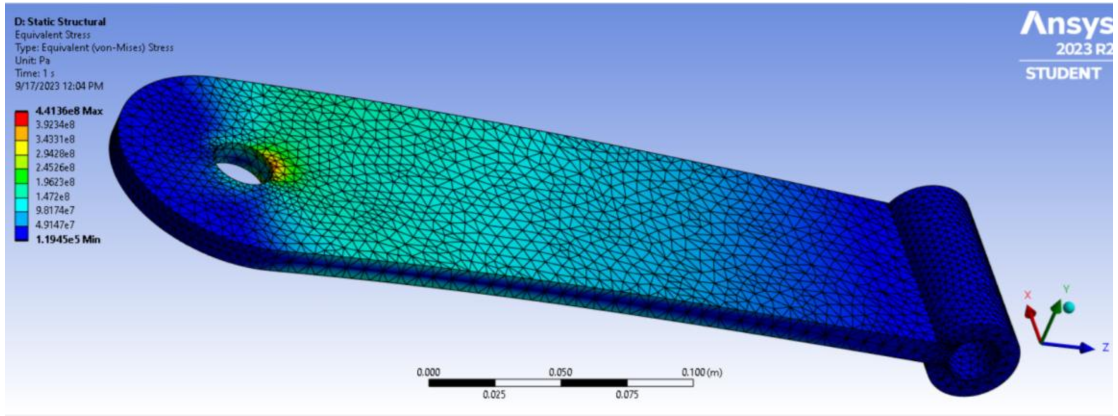
Joint design



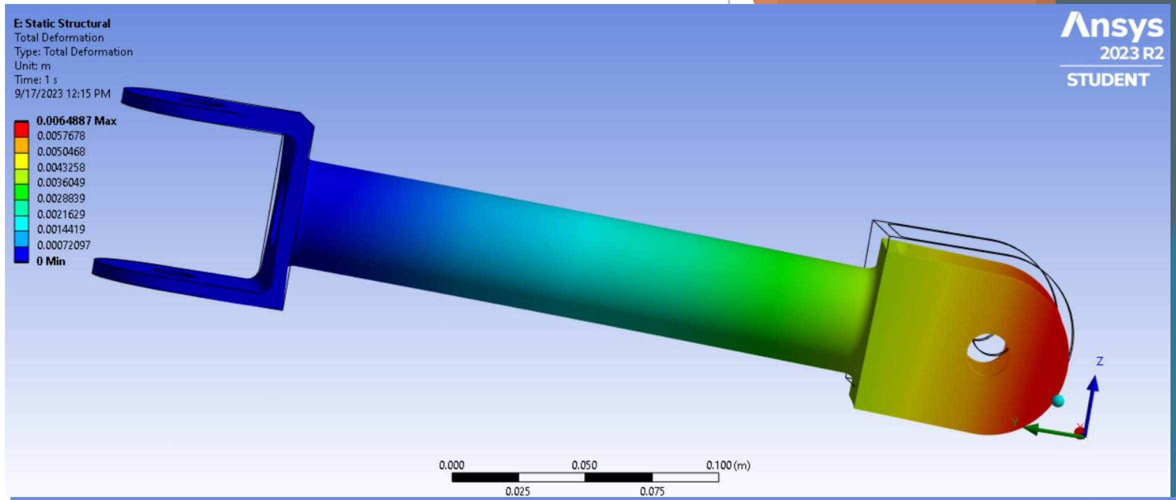


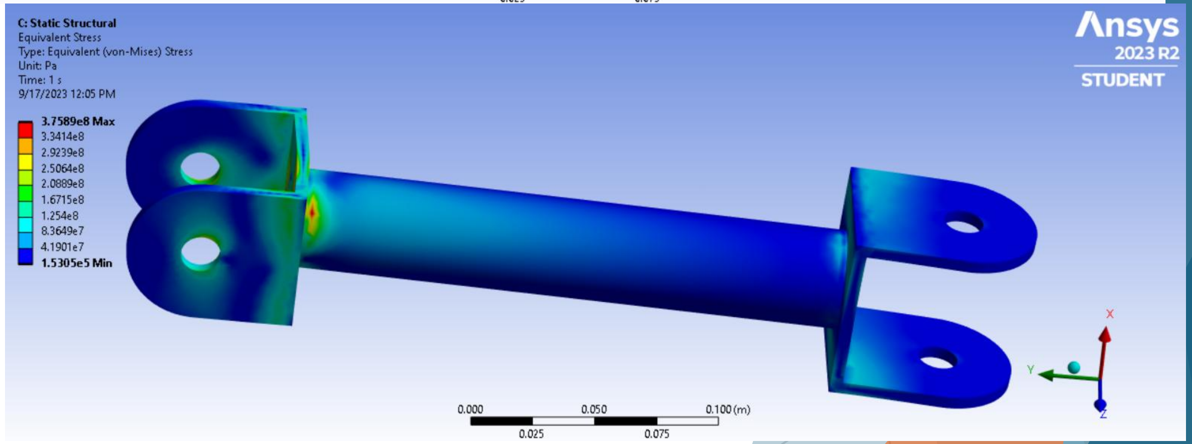
STRESS ANALYSIS





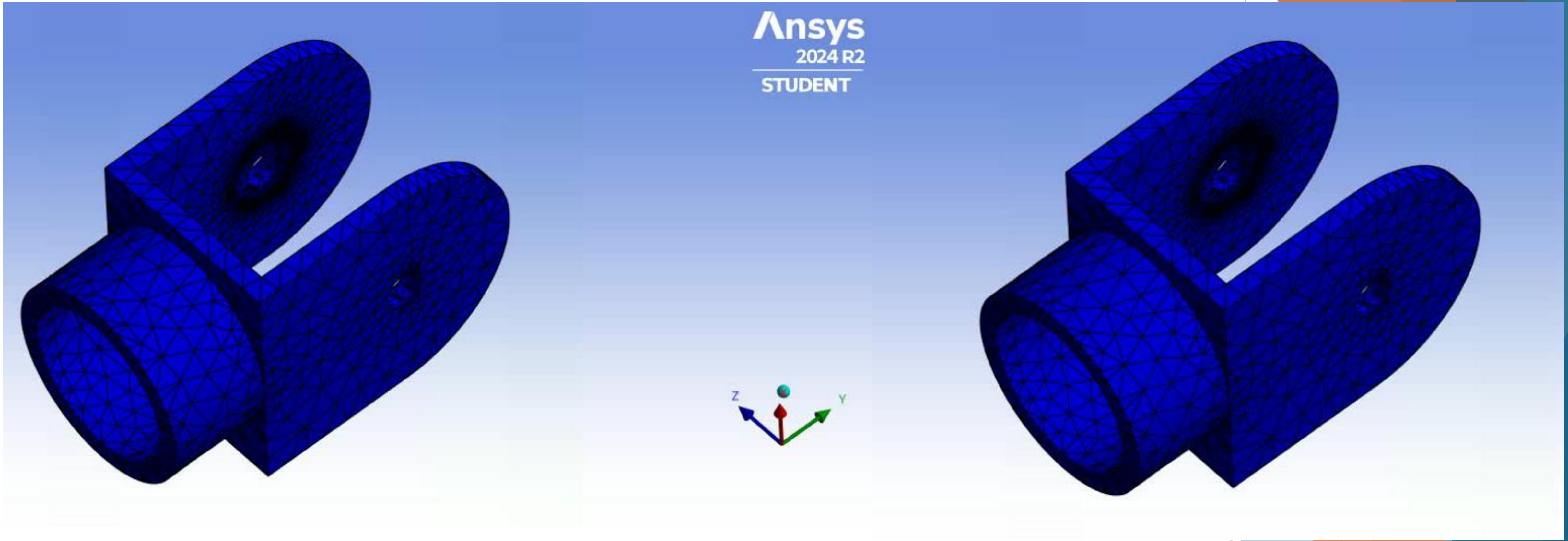
Length - 30 cm
Mass - 936.58 g
Density - 2.7 g/cm^3
Applied force - 1000N (-X & -Z directions)
Maximum Deflection - 16.47 mm
Maximum Stress - 441.36 MPa





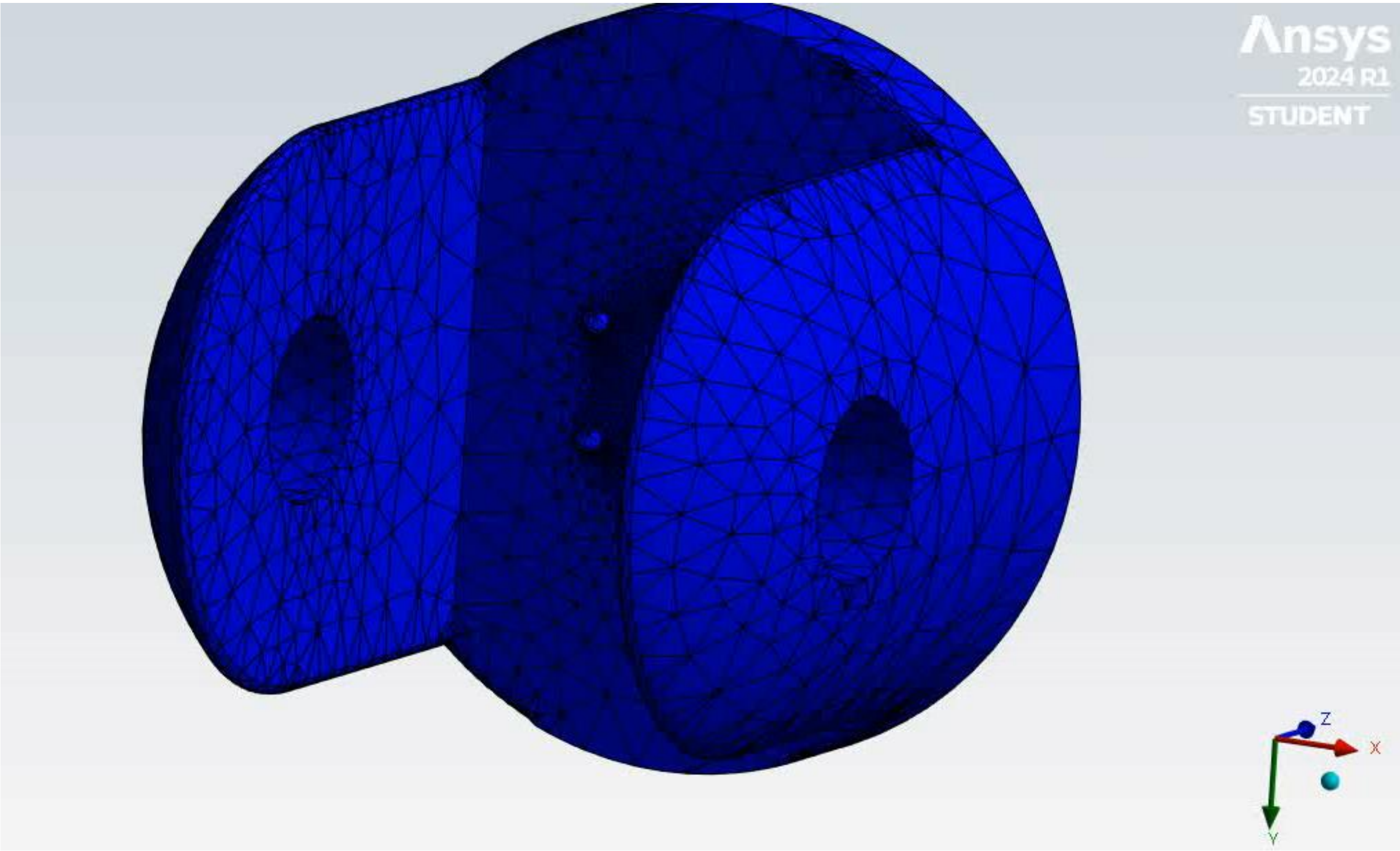
Length - 30 cm
Mass - 615.19 g
Density - 2.7 g/cm^3
Applied force - 1000N (-X & -Z directions)
Maximum Deflection - 6.48 mm
Maximum Stress - 375.89 MPa

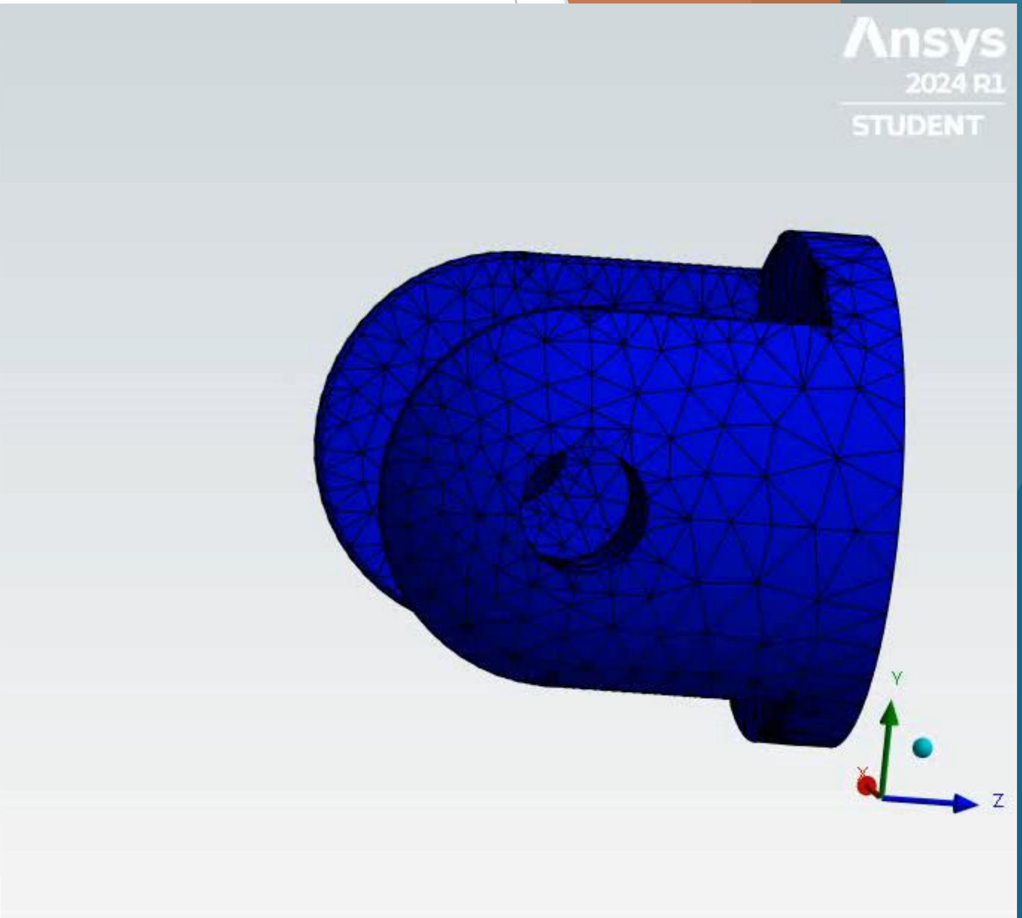
Stress and strain analysis for bracket



- Load applied towards –x direction.

Deformation of 3D printed shoulder socket



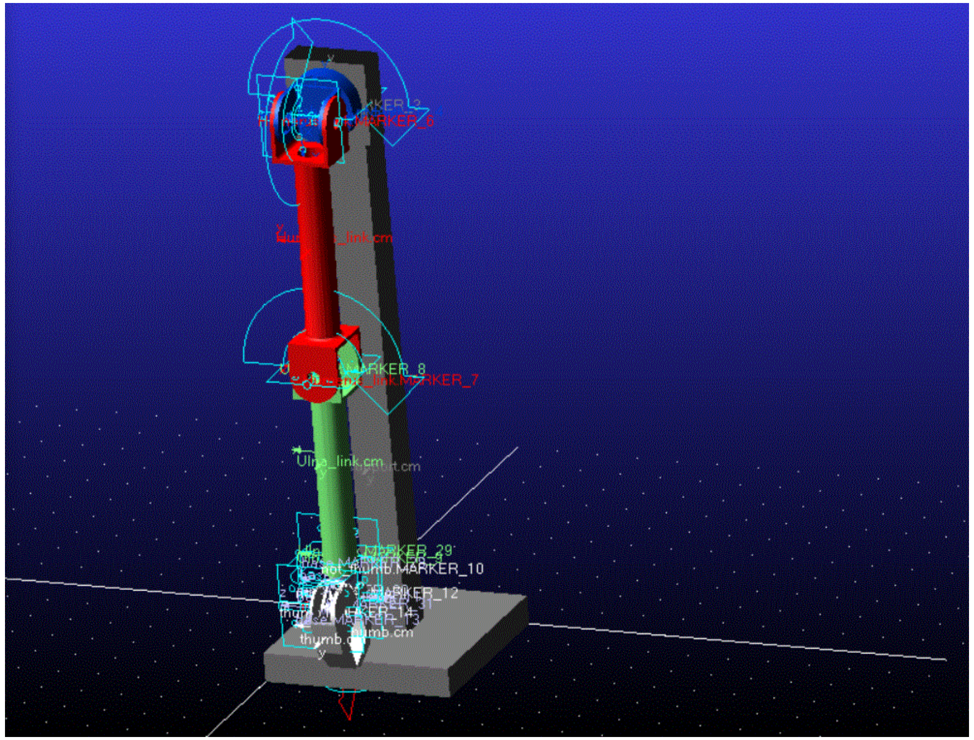


Deformation during flexion movement of the shoulder

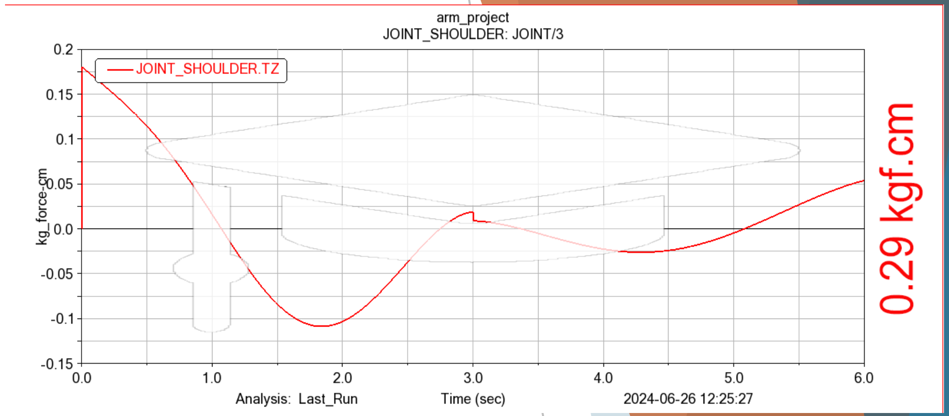
Deformation during holding weight/arm.

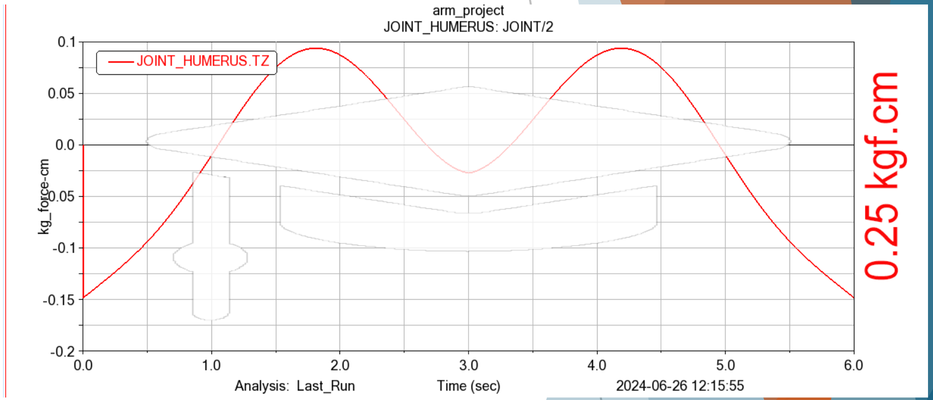
JOINT TORQUE CALCULATIONS

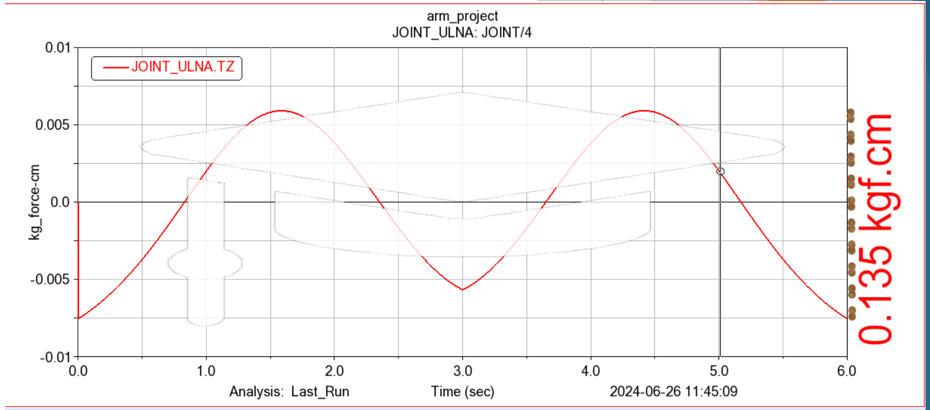
- All torques have been calculated under no load condition



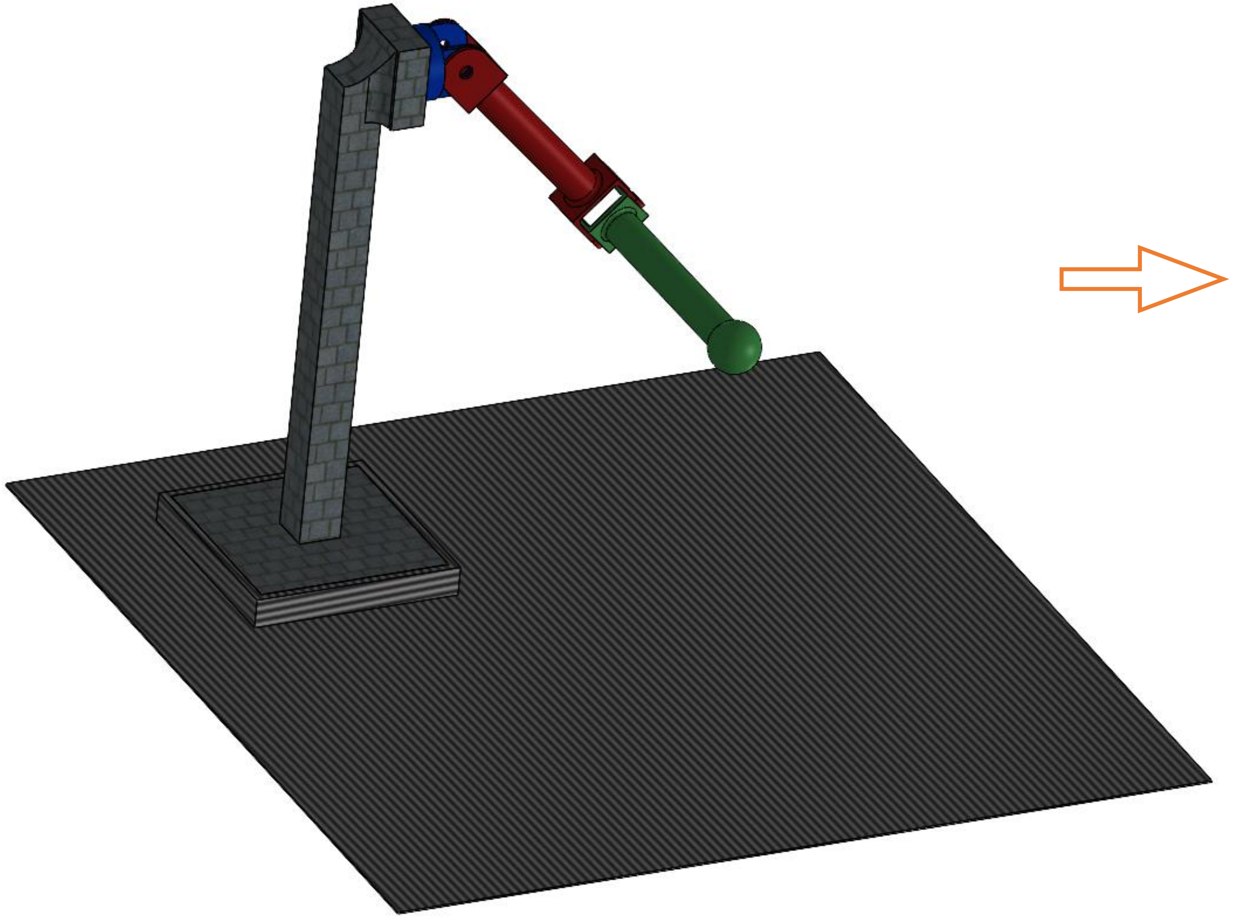
- This simulation encompasses all possible movements that the robotic arm can perform.

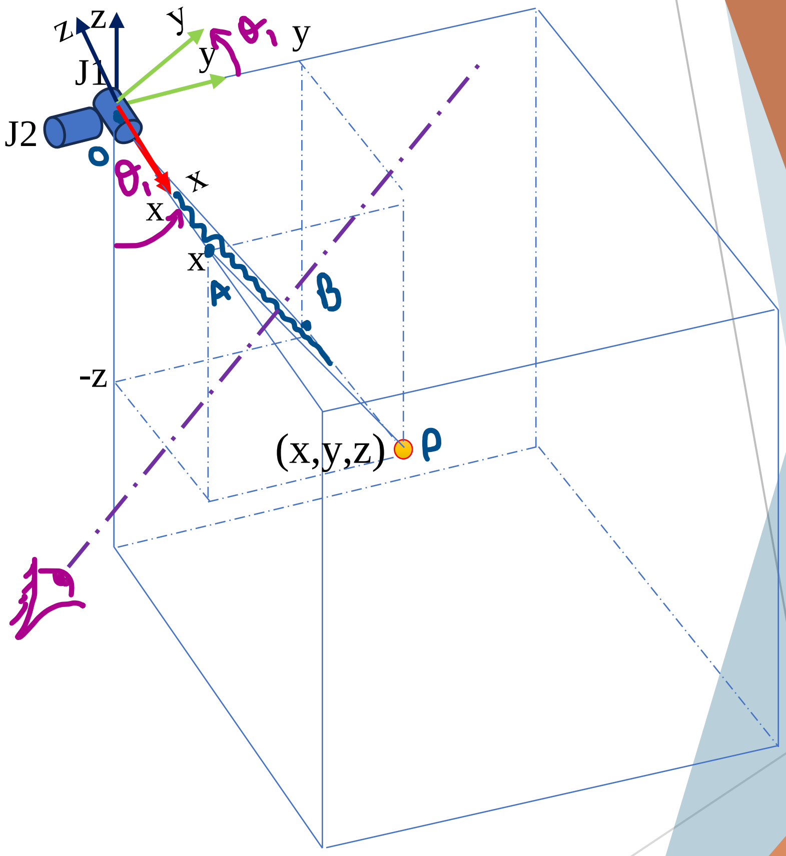






KINEMATICS

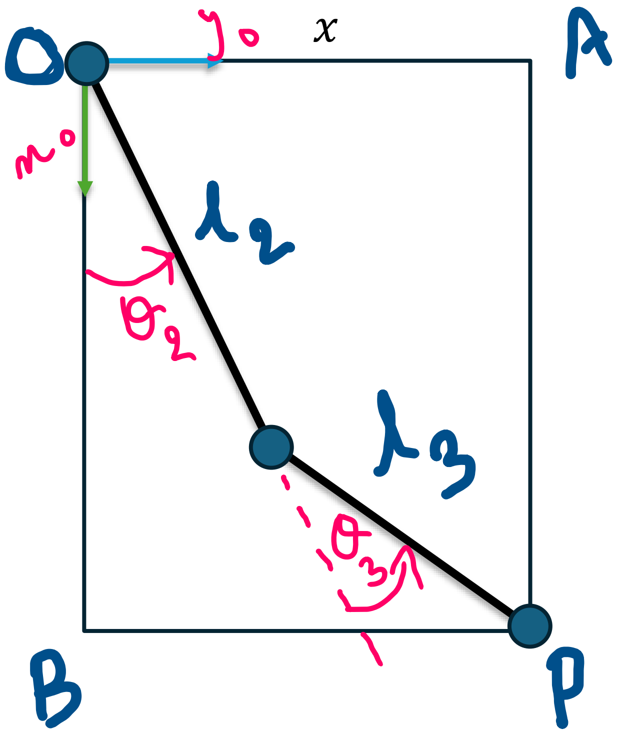




$$\tan \theta_1 = \frac{y}{z} \qquad \theta_1 = \tan^{-1}\frac{y}{z}$$

$\theta 1$ - Joint 1 angle

OAPB plan (new x_0y_0 coordinate system)



$$\boxed{c2, c3, s2, s3 \rightarrow \cos\theta_2, \cos\theta_3, \sin\theta_2, \sin\theta_3}$$

$$\boxed{c23, s23 \rightarrow \cos(\theta_2+\theta_3), \sin(\theta_2+\theta_3)}$$

l1 – humerus link length
l2 – Ulna link length

$$A_1 = \begin{matrix} c2 & -s2 & 0 & l2c2 \\ s2 & c2 & 0 & l2s2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

$$A_2 = \begin{matrix} c3 & -s3 & 0 & l3c3 \\ s3 & c3 & 0 & l3s3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

$$T_2^0 = \begin{matrix} c23 & -s23 & 0 & l2c2 + l3c23 \\ s23 & c23 & 0 & l2s2 + l3s23 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

<u>D-H table</u>

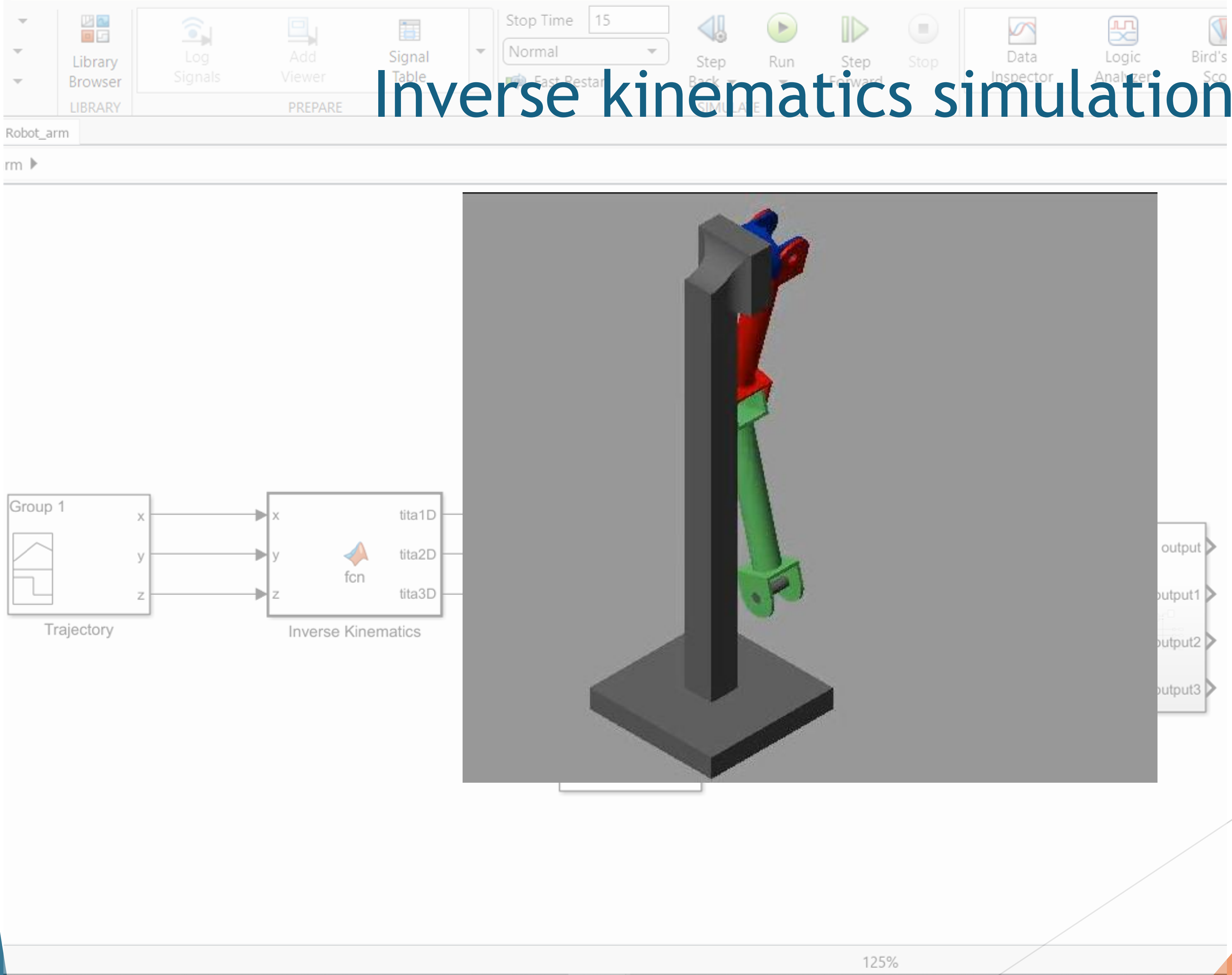
Link	a_i	α_i	d_i	θ_i
1	l_1	0	0	θ_2
2	l_2	0	0	θ_3

$$\boxed{l2.\cos(\theta 2) + l3.\cos(\theta 2+\theta 3) = z.\sec(\theta 1)}$$

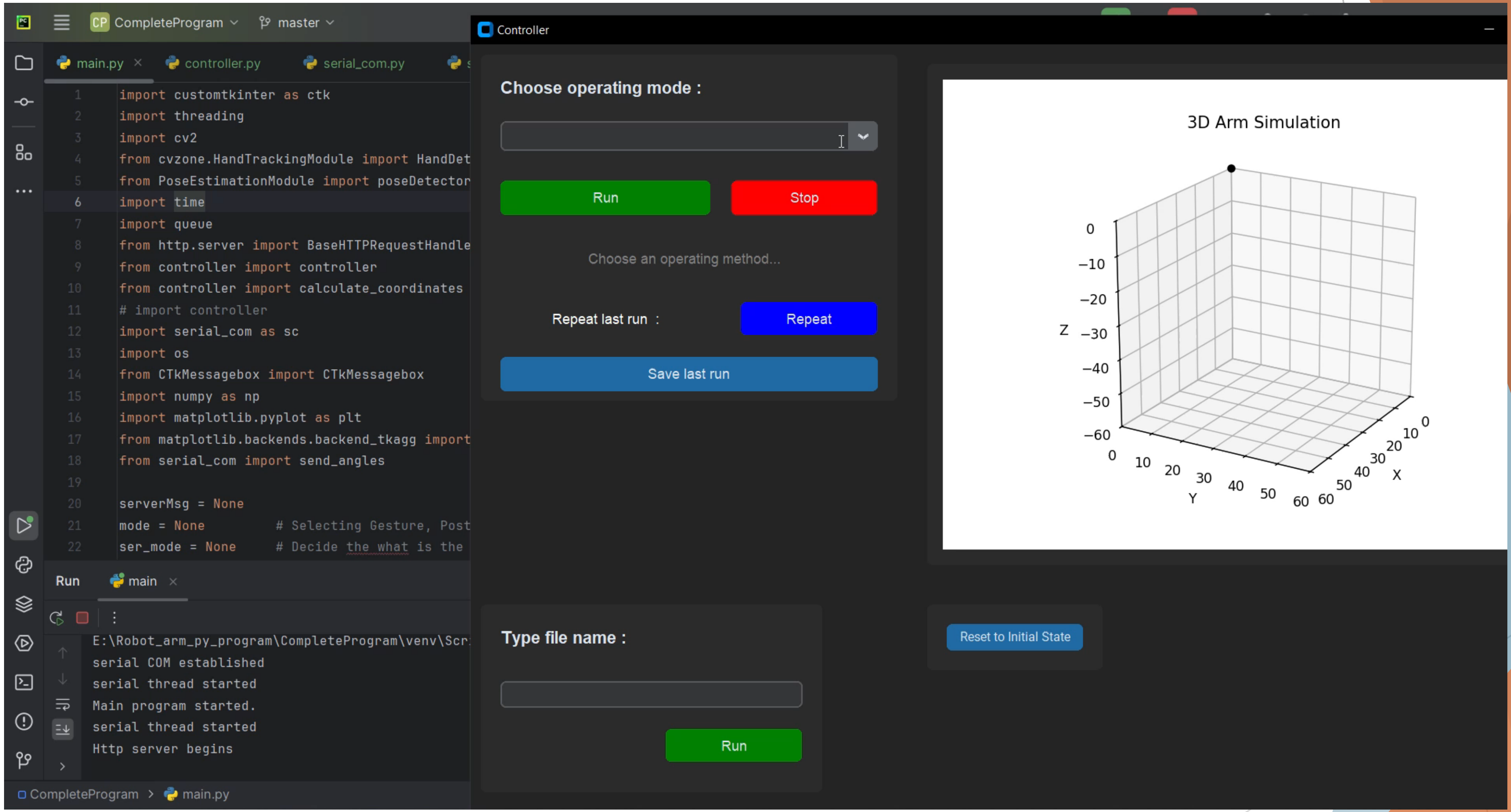
$$\boxed{l2.\sin(\theta 2) + l3.\sin(\theta 2+\theta 3) = x}$$

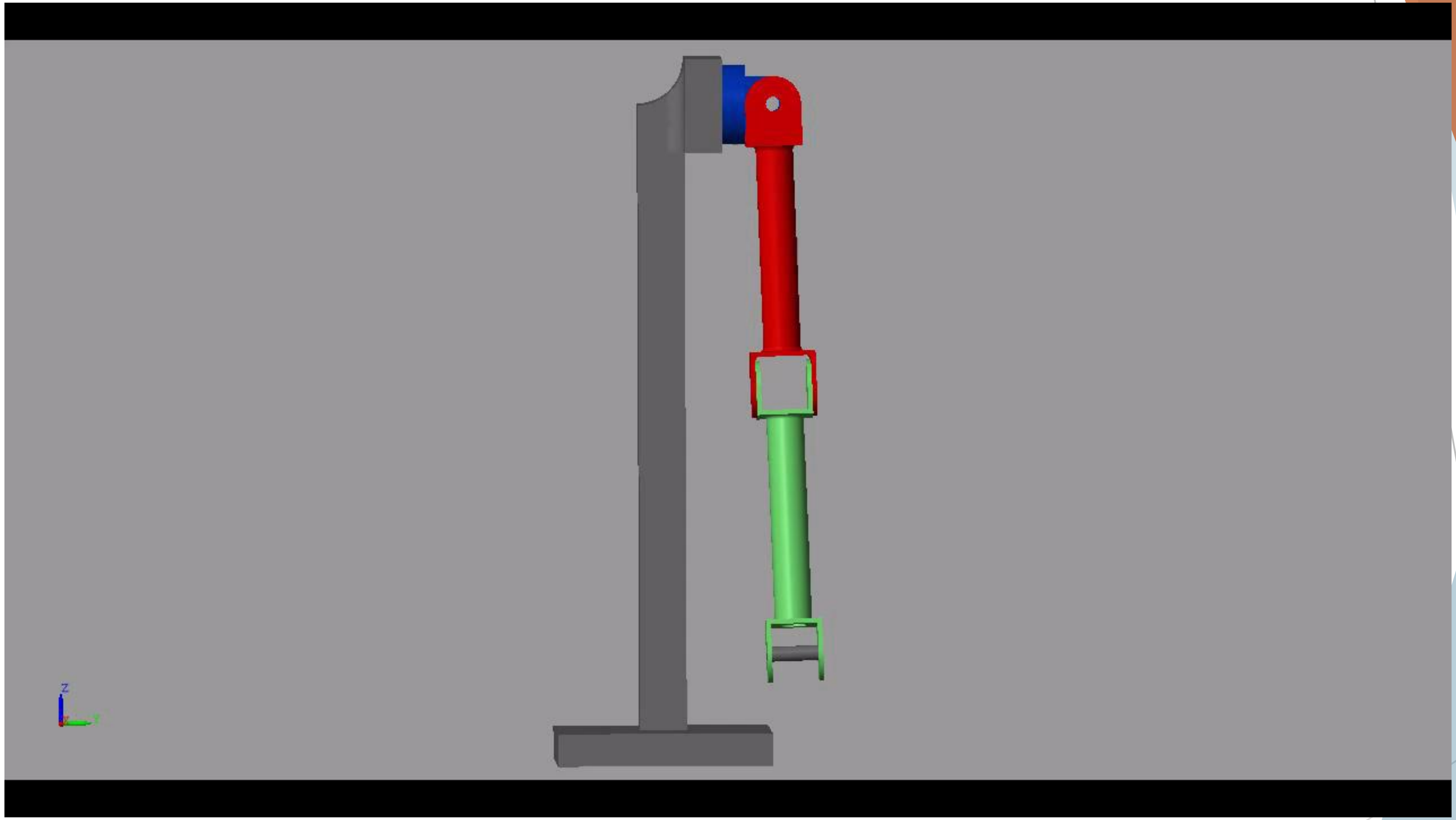
$\theta 2$ - Joint 2 angle
$\theta 3$ - Joint 3 angle

Inverse kinematics simulation



IMPLEMENTATION





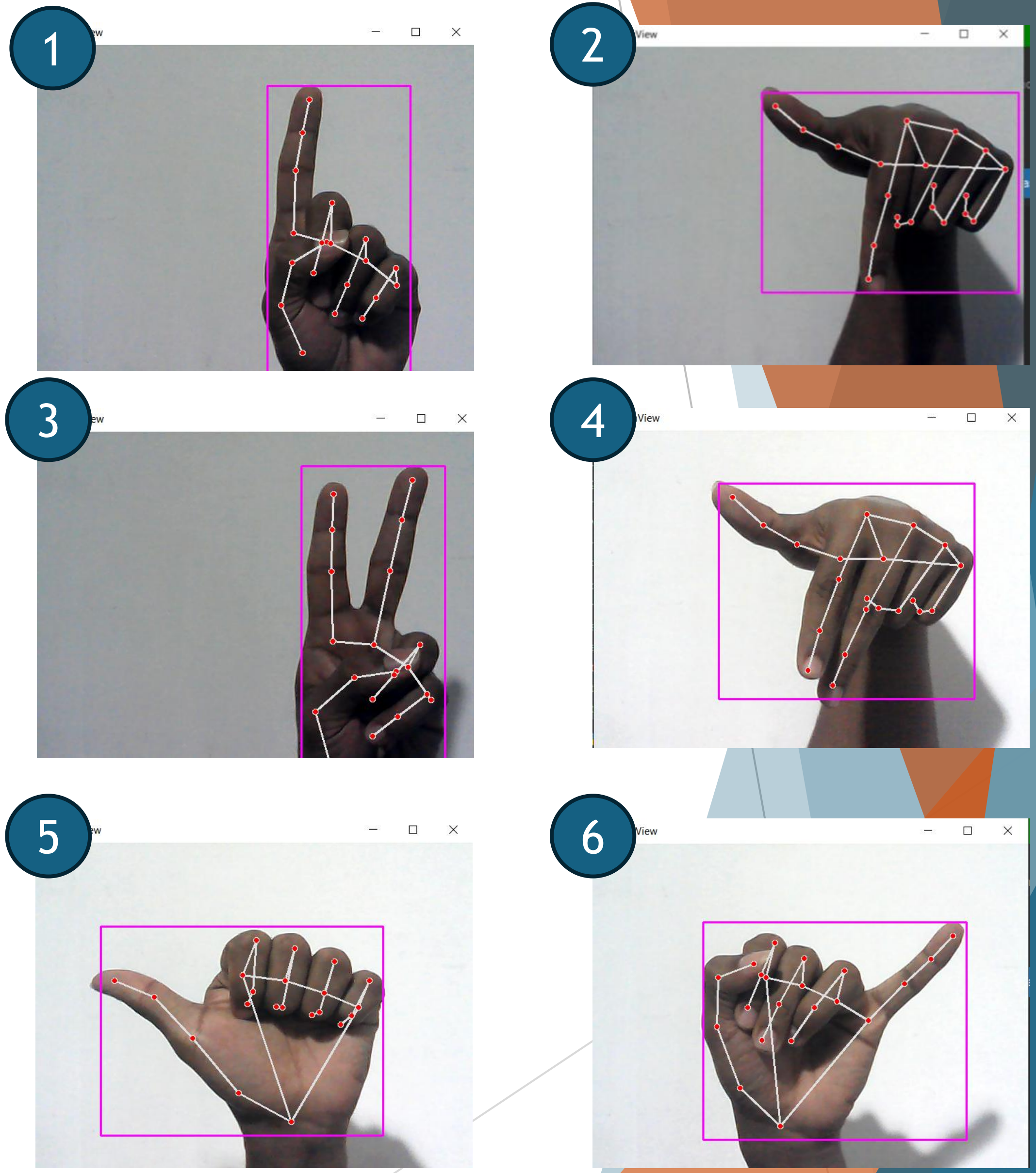
Gesture based control

Six different gestures are used to control each joint individually.

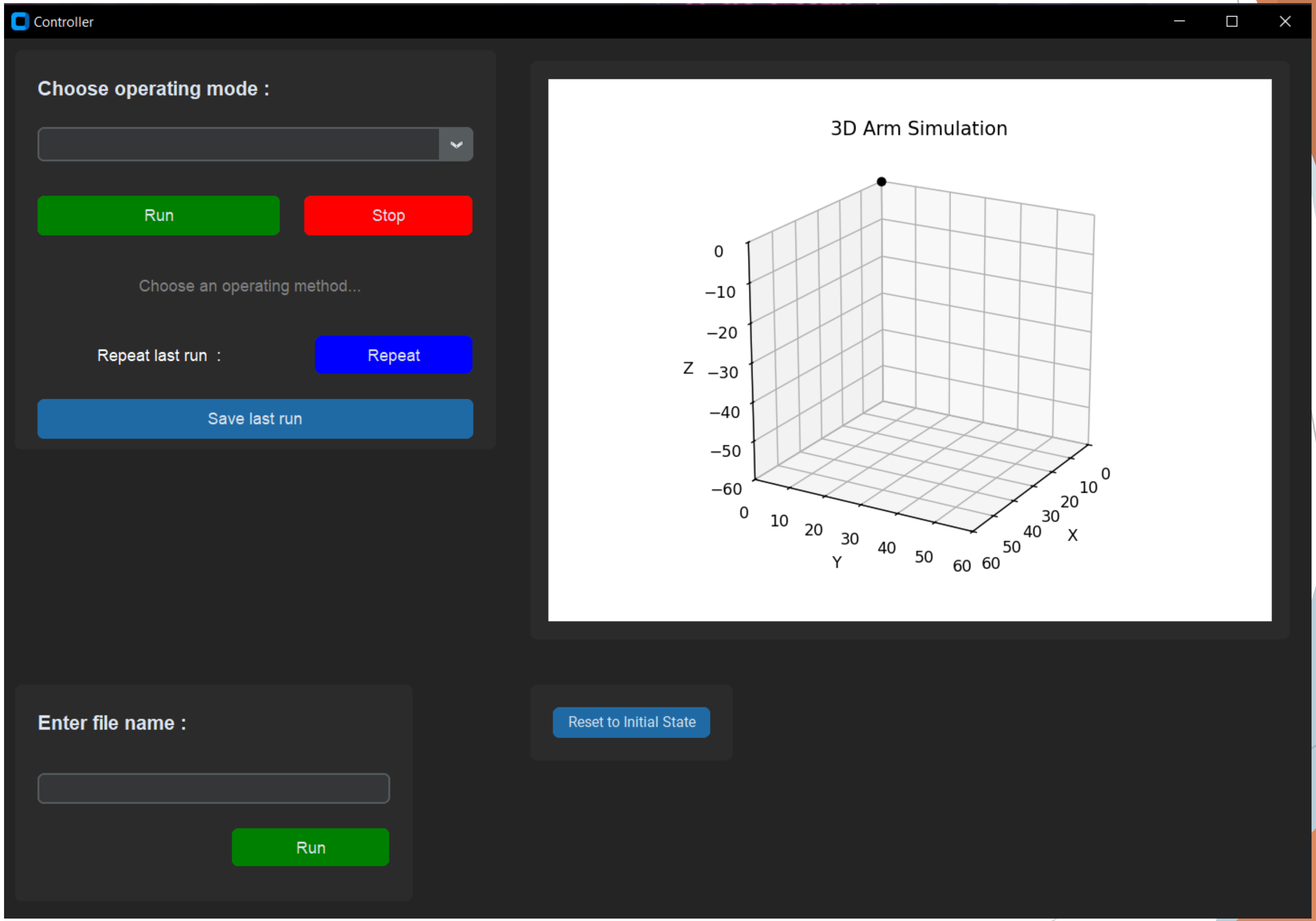
(1) & (2) Control Shoulder flexion and extension movements.

(3) & (4) Control Elbow flexion and extension movements.

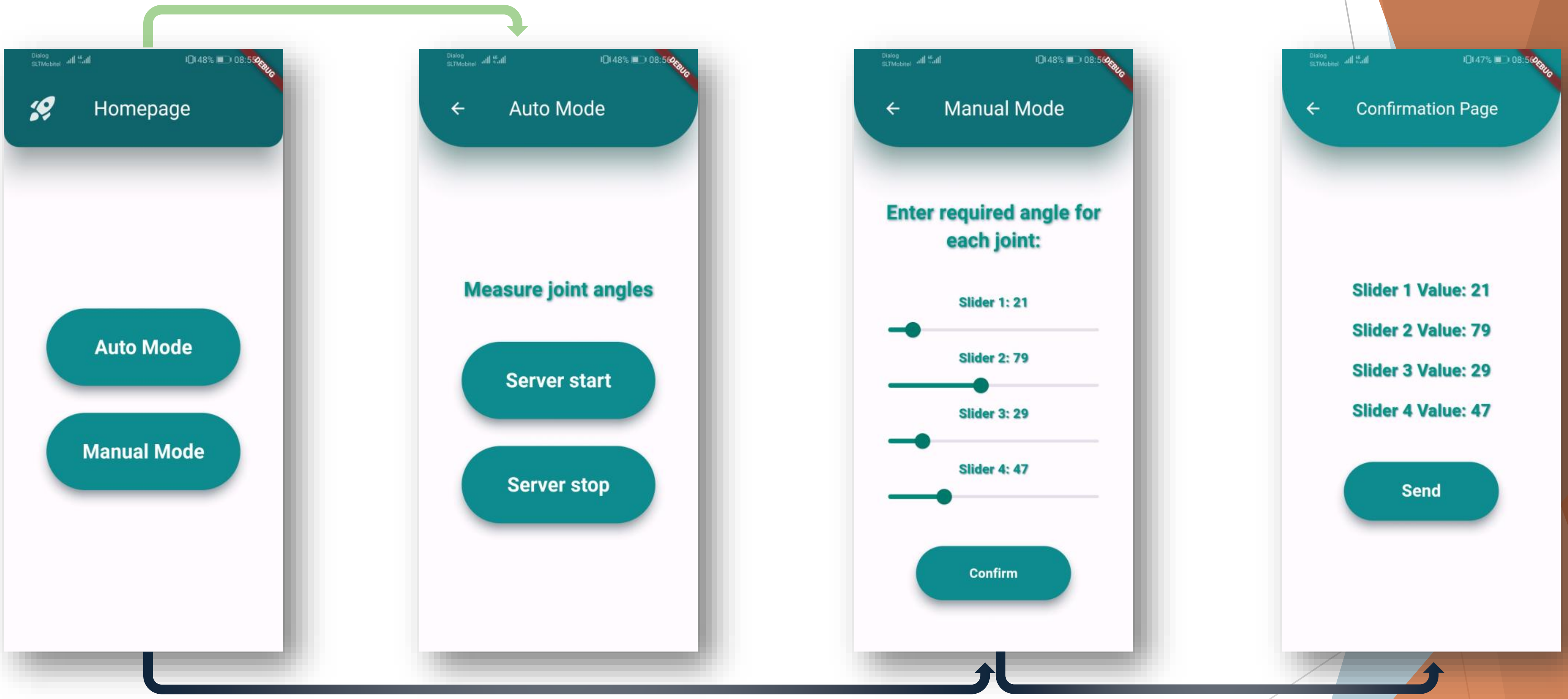
(5) & (6) Control Shoulder abduction and adduction movements.



GUI


Controller
Choose operating mode :
Run
Stop
Choose an operating method...
Repeat last run :
Repeat
Save last run
3D Arm Simulation
Z
Y
X
Enter file name :
Run
Reset to Initial State

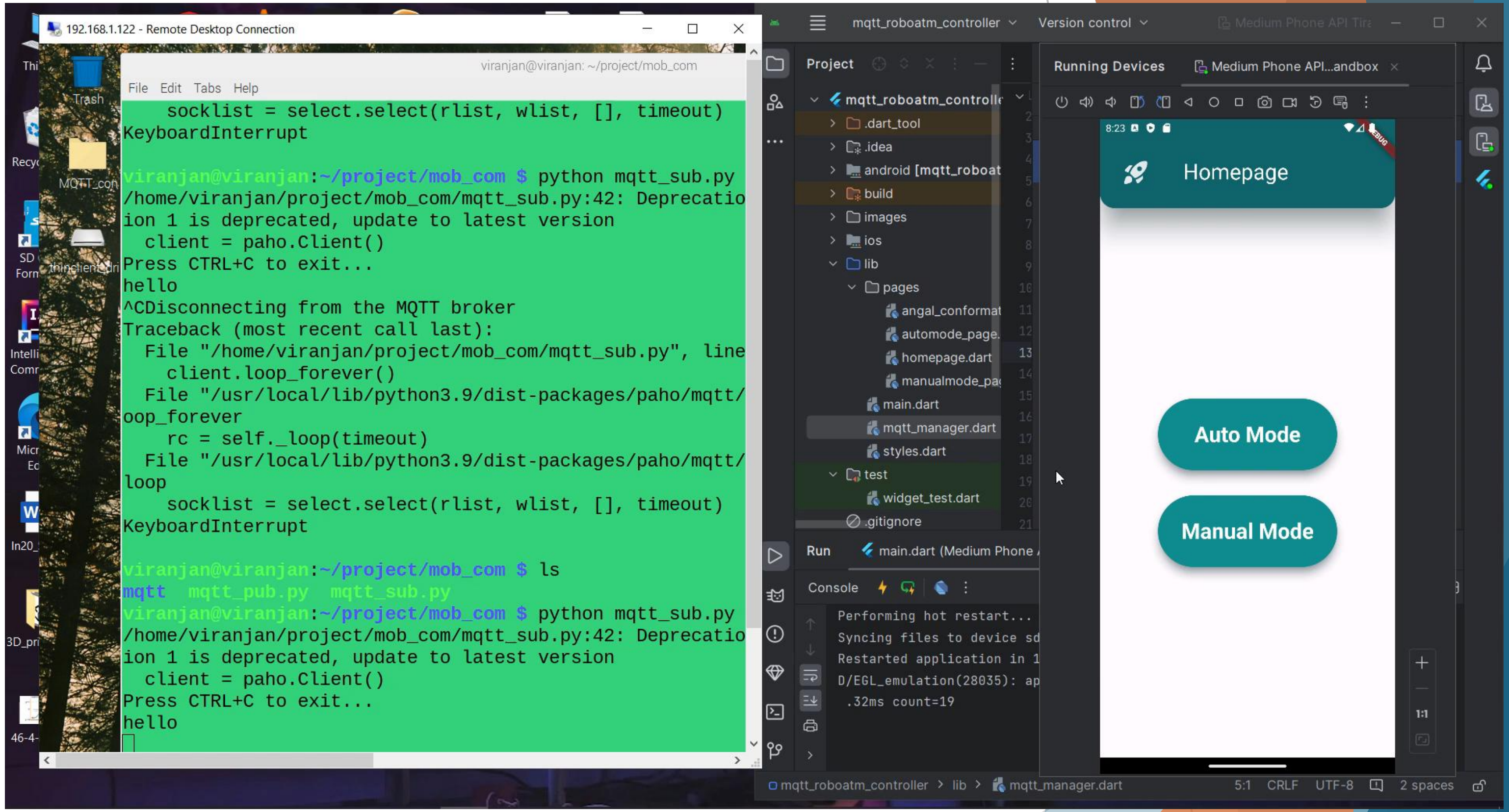
Mobile app


Homepage
Auto Mode
Manual Mode
Auto Mode
Measure joint angles
Server start
Server stop
Manual Mode
Enter required angle for each joint:
Slider 1: 21
Slider 2: 79
Slider 3: 29
Slider 4: 47
Confirm
Confirmation Page
Slider 1 Value: 21
Slider 2 Value: 79
Slider 3 Value: 29
Slider 4 Value: 47
Send

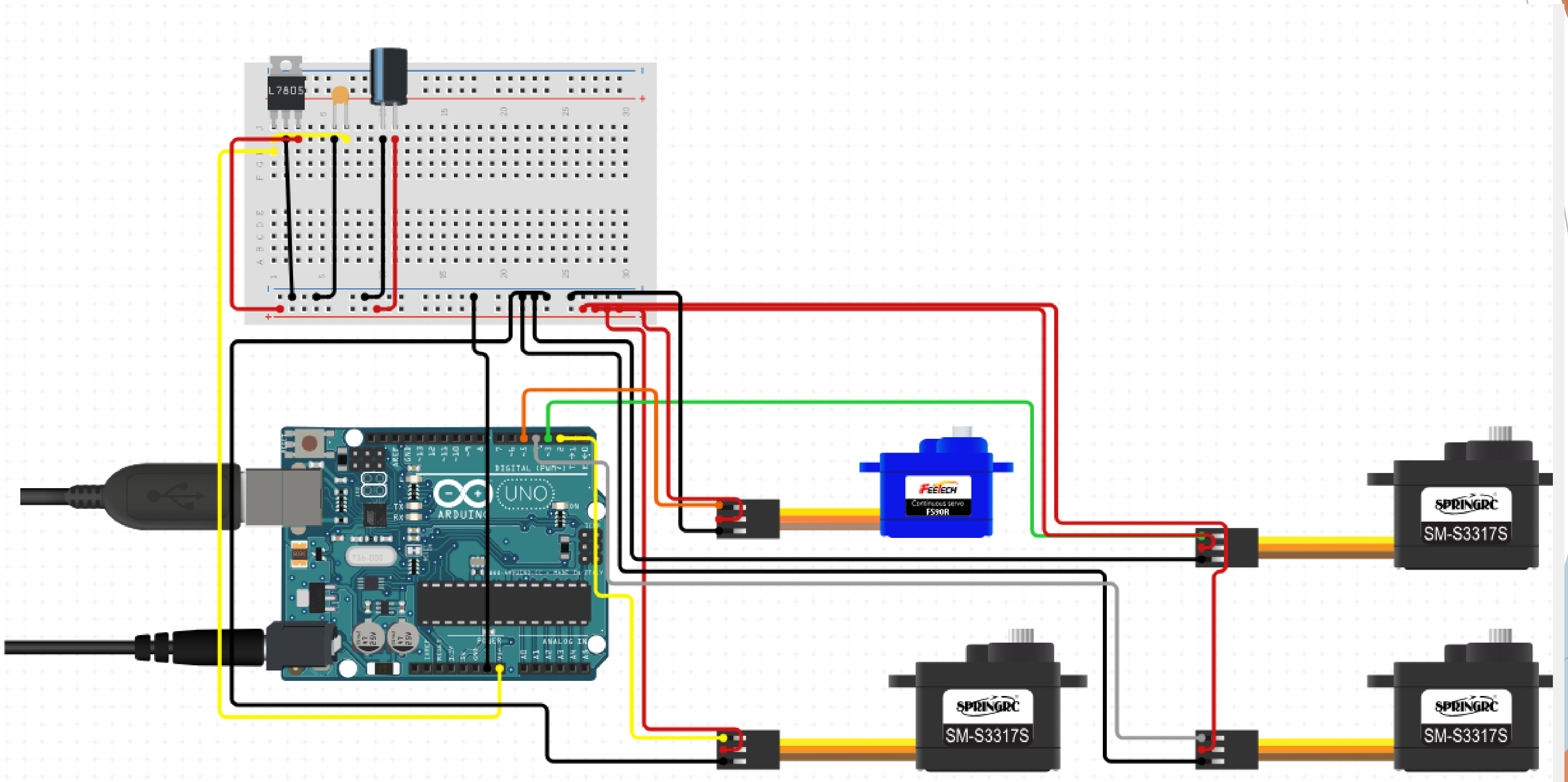
Mobile app communication

Here a representation of MQTT communication from mobile app to raspberry pi.

Mobile app is publisher, Raspberry pi is subscriber,
Topic : robot_arm/angles



Circuit diagram

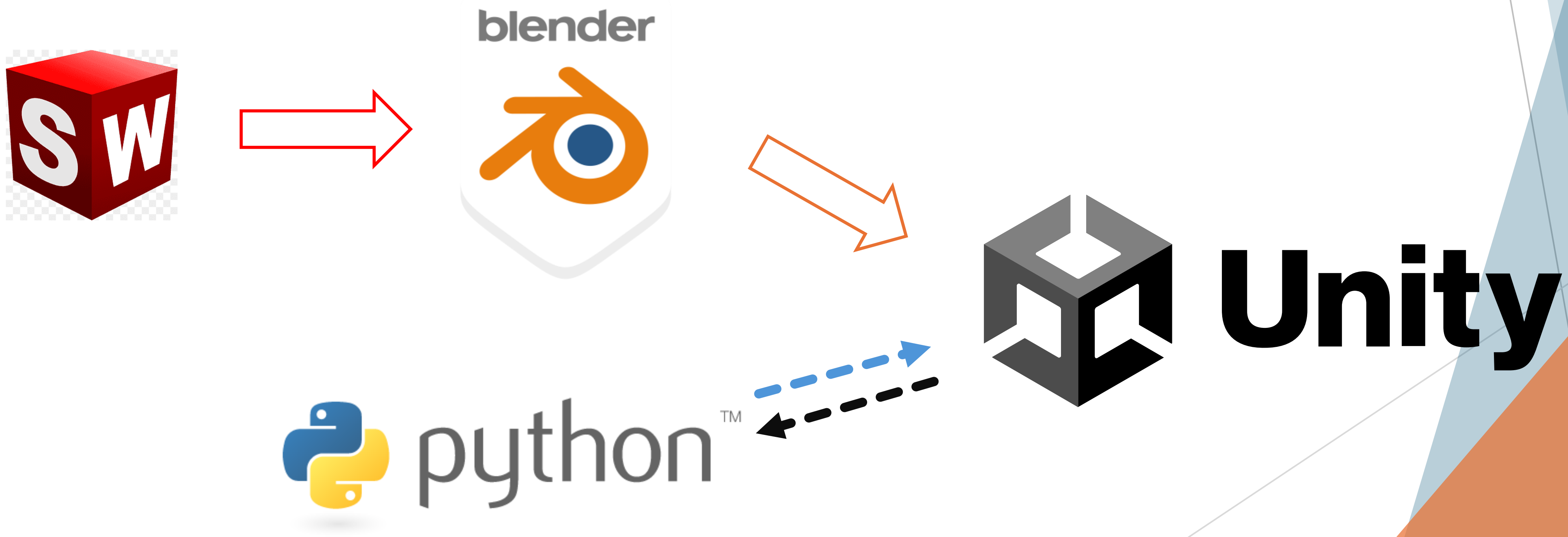


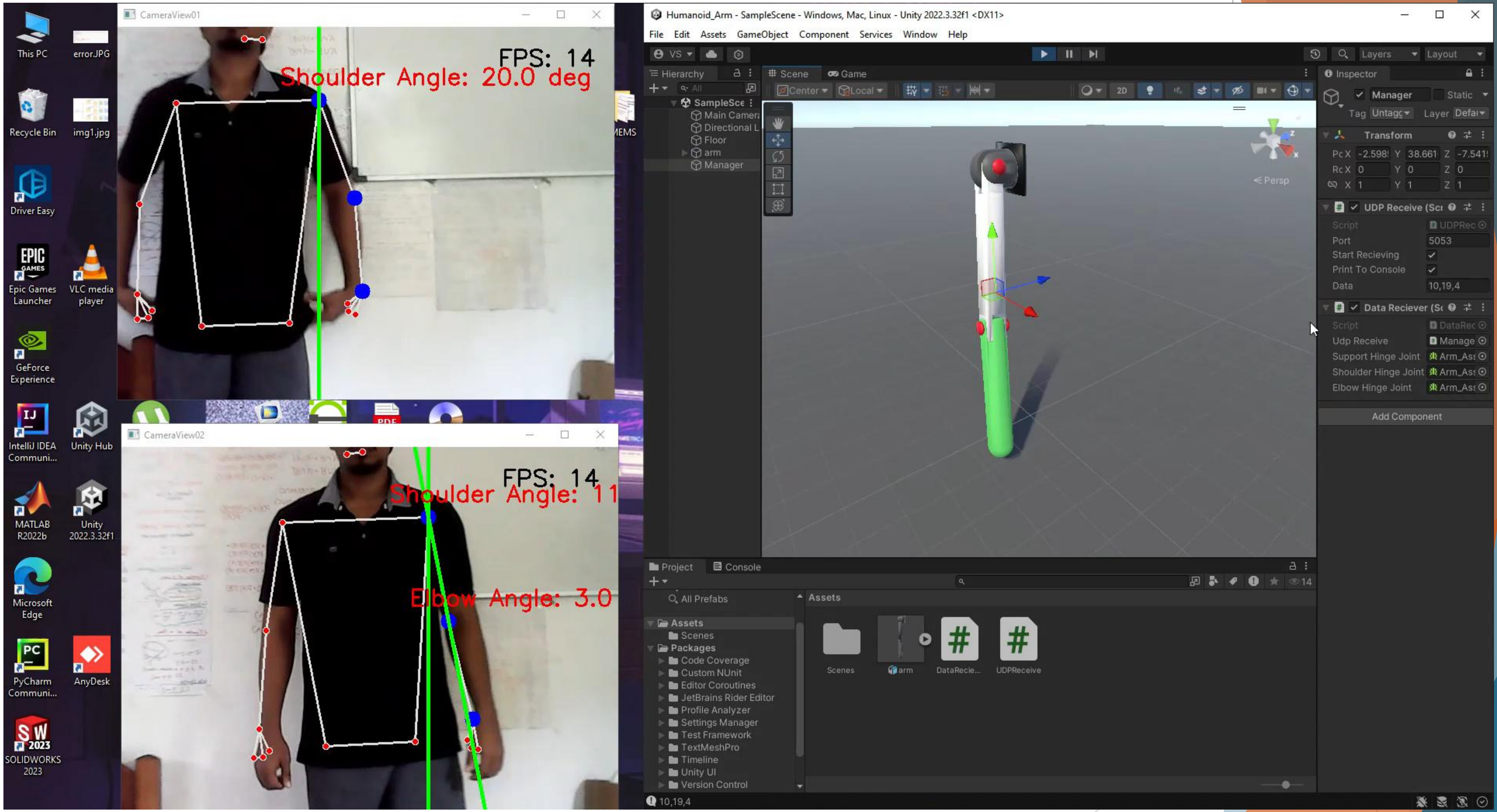
SIMULATION

Real-time camera feedback is simulated in unity environment. Unity is a cross-platform game engine developed by Unity Technologies.

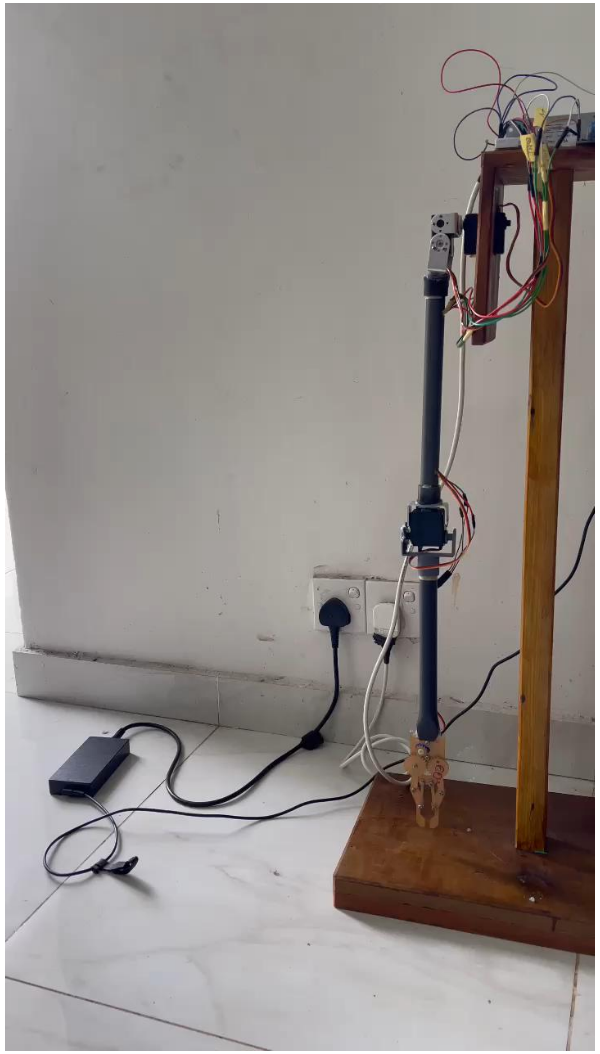
This environment can also be used to simulate scenarios that require real-time physics.

Unity has capability to communicate with python.




CameraView01
FPS: 14
Shoulder Angle: 20.0 deg
CameraView02
FPS: 14
Shoulder Angle: 11
Elbow Angle: 3.0
Humanoid_Arm - SampleScene - Windows, Mac, Linux - Unity 2022.3.32f1 <DX11>
File Edit Assets GameObject Component Services Window Help
Hierarchy
SampleScene
Main Camera
Directional L
Floor
arm
Manager
Scene
Game
Inspector
Manager
Transform
UDP Receive
Port 5053
Start Recieving
Print To Console
Data 10,19,4
Data Reciever
Udp Receive
Support Hinge Joint
Shoulder Hinge Joint
Elbow Hinge Joint
Add Component
Project
Console
Assets
Scenes
Packages
10,19,4

PROTOTYPE



Operating with “Arm posture control”

THANK YOU