Learning-based Model Predictive Controller for Drink Dispensing Robotic Arm Relying on Multimodal Inputs

Team Members

Joseph Thapa Magar (THA076BEI011)

Ritu Ram Ojha (THA076BEI024)

Rupak Mani Sharma (THA076BEI027)

Sujan Prasad Bhattarai (THA076BEI037)

Project Supervisor:

Er. Dinesh Baniya Kshatri

Lecturer

Department of Electronics and Computer Engineering
Institute of Engineering, Thapathali Campus

March 6, 2024

Presentation Outline

- Motivation
- Introduction
- Problem Statement & Objectives
- Scope of Project
- Project Applications
- Methodology
- Results
- Analysis of Results
- Future Enhancements
- Conclusion
- References

Motivation

Monotonous and Tired Bartender

Command Operated Drink Serving Arm

Introduction

- Replaces manual drink serving with 4 DOF robotic arm
- Integrates speech recognition and computer vision
- Implements model predictive controller for precise robotic movements
- Addresses environmental anomalies with adaptable vision-based planning
- Simulates and physically instantiates the robotic arm

Problem Statement & Objectives

- Problem Statement
  - Minimal adoption of robotic arms in the general environments populated by humans along with rich user interface and multimodal control features

- Objectives
  - To model and simulate a drink dispensing robotic arm

  - To instantiate the robotic arm and perform performance comparison between simulation and reality

Scope of Project

- Project Capabilities:
  - Voice-based human-machine interaction
  - Proper detection of glass, dispenser, nozzle and obstacle
  - Precise and responsive control of robotic arm

- Project Limitations :
  - Language understanding limited to English language
  - Challenge in object recognition due to poor visibility
  - Robotic arm movement constrained to 4 degrees of freedom

Project Applications

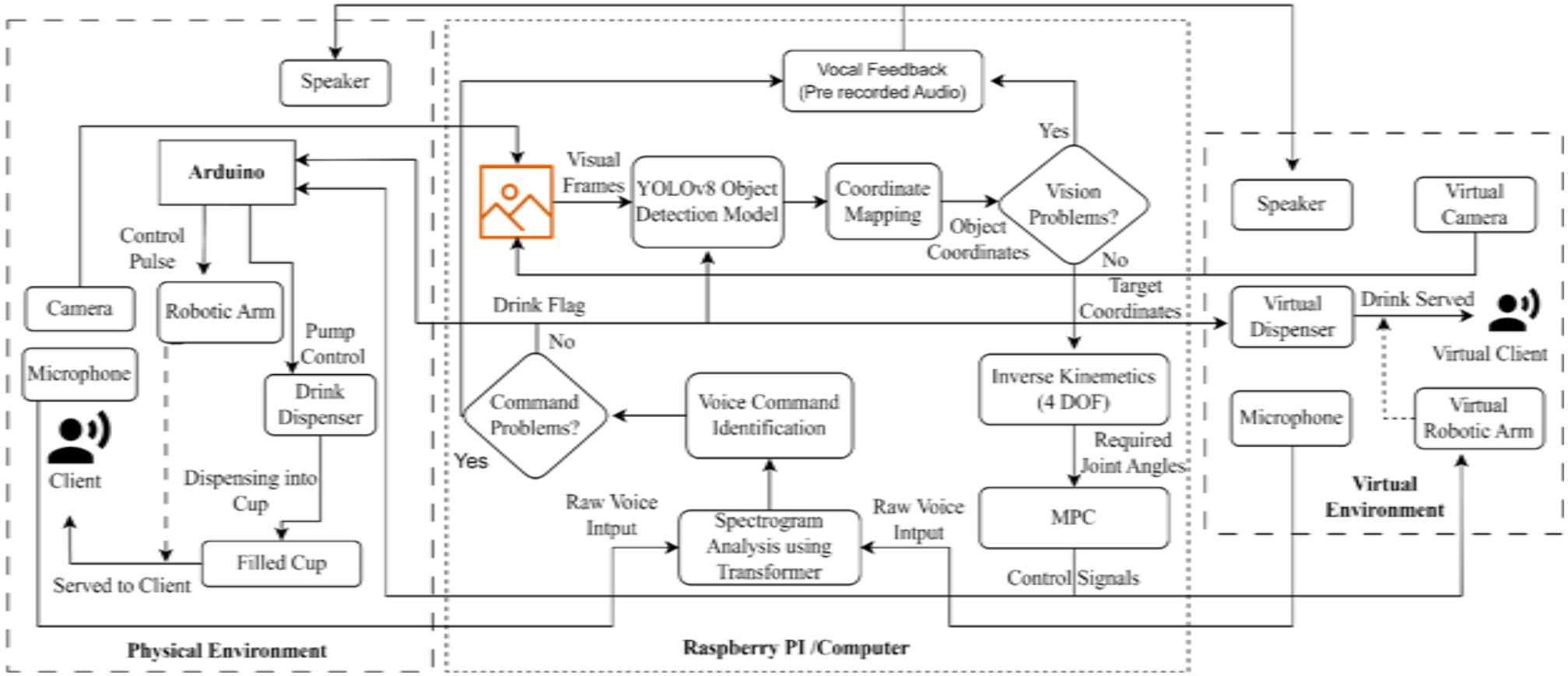
- Medical and Healthcare
  - Surgical procedures, rehabilitation, diagnostics, prosthetics
- Aerospace and Defense
  - Aircraft assembly, space exploration, defense applications
- Manufacturing and Industrial Automation
  - Assembly, welding, pick and place operations, packaging
- Hazardous Environments
  - Nuclear power plants, deep-sea exploration, mining, disaster response

Methodology-[1]
(Hardware/Software Needs)

Software Needs		
Name	**Version**	**Purpose**
Pytorch	2.1.0+cu121	Creating and Training Speech Recognition and vision model
Python	3.10	Writing code for various purposes, including integration with other software modules.
Arduino IDE	2.3.2	Developing and writing code for Arduino microcontrollers, including robotic control.
Audacity	3.4.2	Manipulation of audio Datasets
CoppeliaSim	4.5.1	Simulating the robotic arm
Fusion 360	16.5.0	Designing the robotic arm
Cura	5.6	Slicing the mesh file
Play.ht	1.0	Create voice clone

Hardware Needs				
Name	**Part Number**	**Characteristics**	**Purpose**	**Quantity**
Ball Bearings	6810	2zz double sided steel seal 50x65x7mm	Gearbox	8
Ball Bearings	6804	2zz double sided steel seal 20x32x7mm	Gearbox	16
Motor Driver	TB6600	4A current 32 micro stepping	Stepper	4
Stepper Motor	Nema 17	45Ncm, 60 Ncm 1.7x1.7-inch cross section	Joints	4
Servo	MG90	Metal Geared 180 degrees of rotation 20Ncm torque	Gripper	1
Power Supply	S-200-6	12V 20A	Power	1
Webcam	Odroid	720P HD cam	Vison	1
Microphone	Fantech MCX01 Leviosa	Condenser Microphone	Voice Dataset Collection	1
3D printer	Ender 3	220mmx220mmx 250mm ABS support	Gearbox	1
Google Collab GPU	T4	2,560 cuda cores 16GB of GDDR6 memory	Training vision and speech model	1

Methodology-[2]
(System Block Diagram)



Methodology-[3]
(Working Principle)

- Microphone takes raw voice input from users
- Squeezeformer takes raw voice and output text
- If error in recognizing command, provides vocal feedback
- Camera captures the current frame of environment
- The frames are then pass into object detection model
- Obtained coordinates are mapped into world coordinates
- Any problems related to vision are given as vocal feedback
- The world coordinates are used to set target position

Methodology-[4]
(Working Principle)

- Inverse kinematics provides joint angles based on target position
- MPC provides the control signal for the target joint angles
- Virtual Environment
  - Microphone and speaker of computer is utilized
  - Virtual camera captures frame of the environment
  - Virtual arm serves drink to virtual client
- Physical Environment
  - External microphone, camera, speaker are utilized
  - Dispenser pumps the drink according to the ordered drink
  - Robotic arm serves the drink to the client

Methodology-[5]
(3D Design of Joint/Cycloidal Drive)

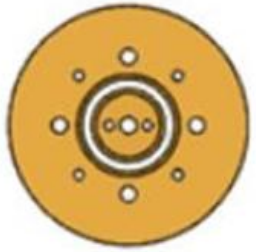
Housing Lid

Eccentric Cam

Main Housing

Cycloidal Disk 1

Cycloidal Disk 2

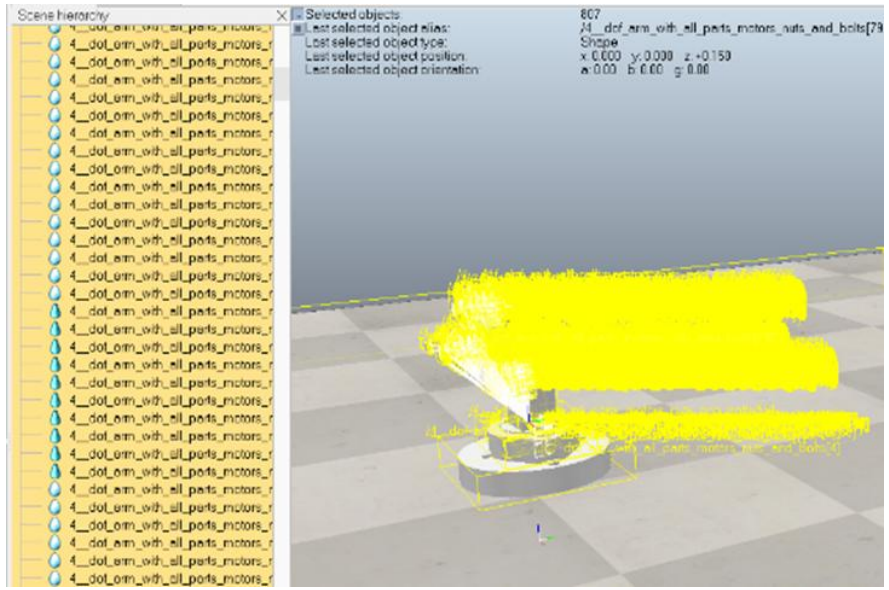


Input/Output
Coupler

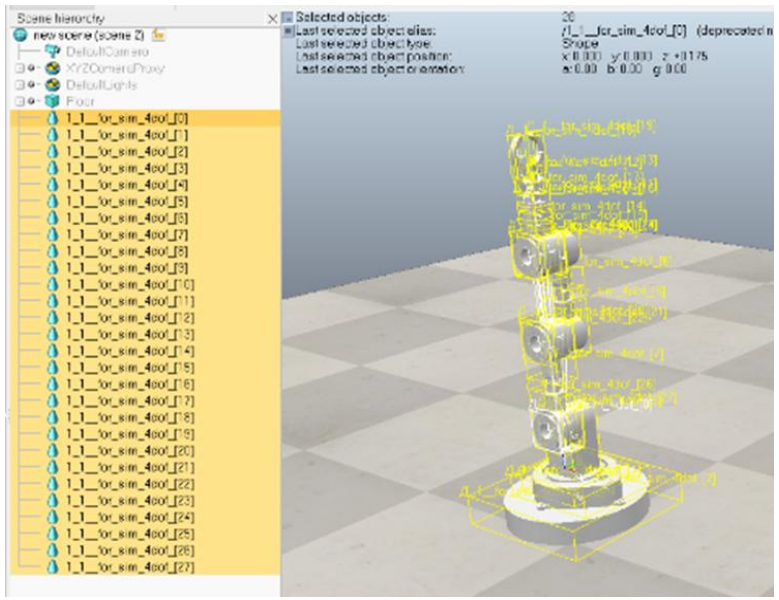
Stepper Mount

Input/Output
Coupler

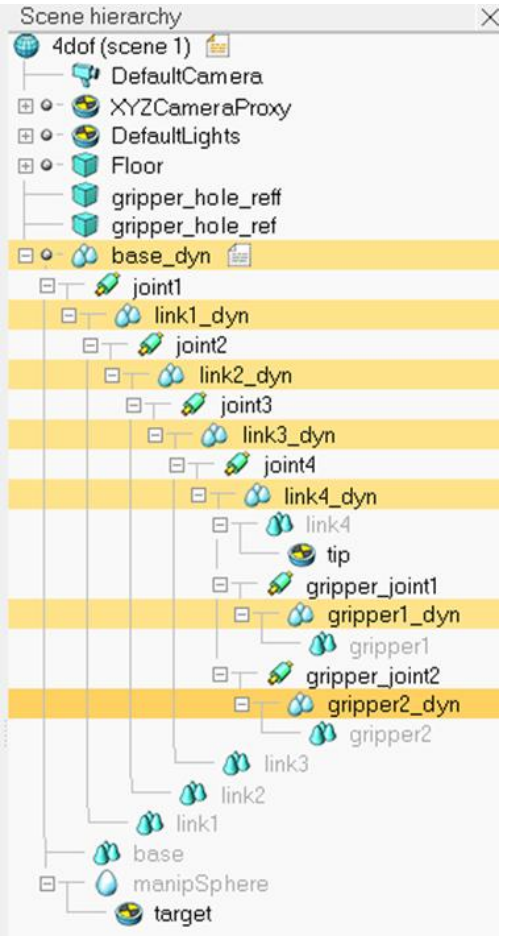
Methodology-[6]
(Fusion Mesh to Coppelia Sim)



Large Number of Triangle Counts

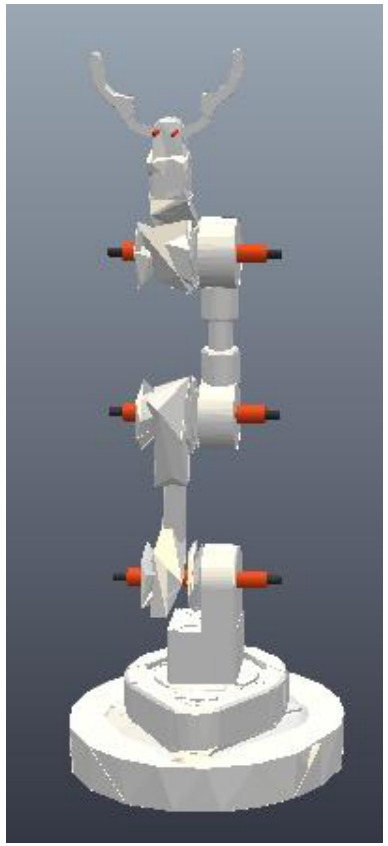


After Reduction of Triangle Counts



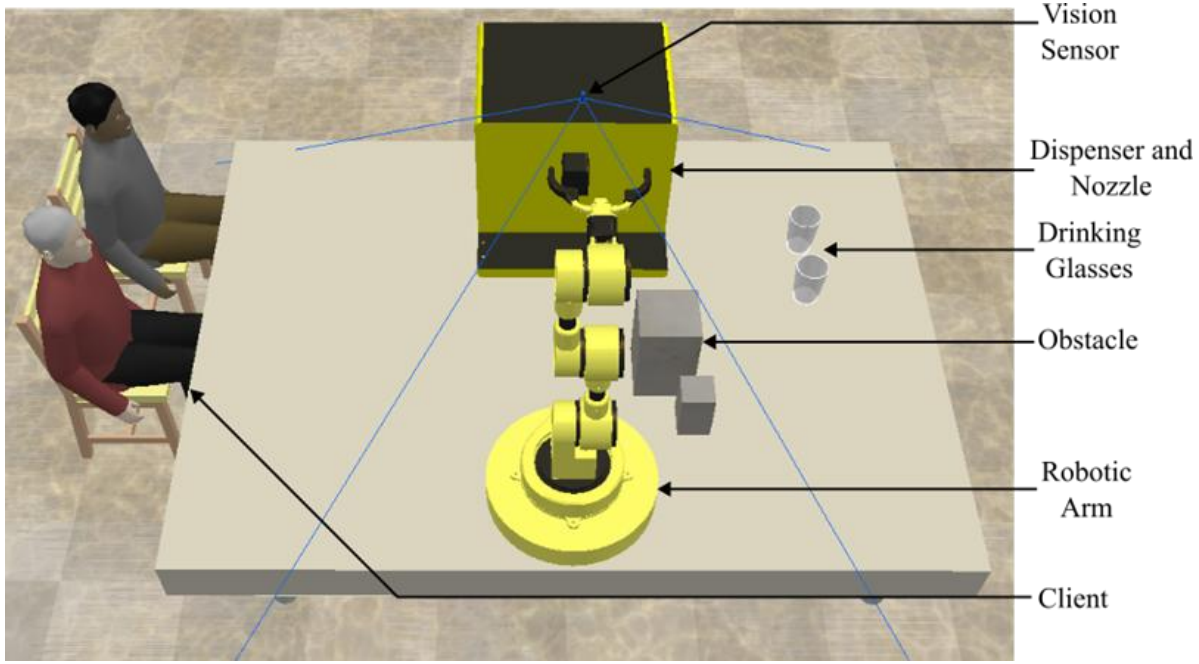
Grouping and Hierarchy for Joints and Links

Methodology-[7]
(Fusion Mesh to Coppelia Sim)



Dynamic and Respondable Objects

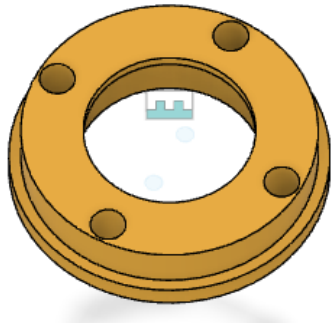
Static and Non-Respondable Objects

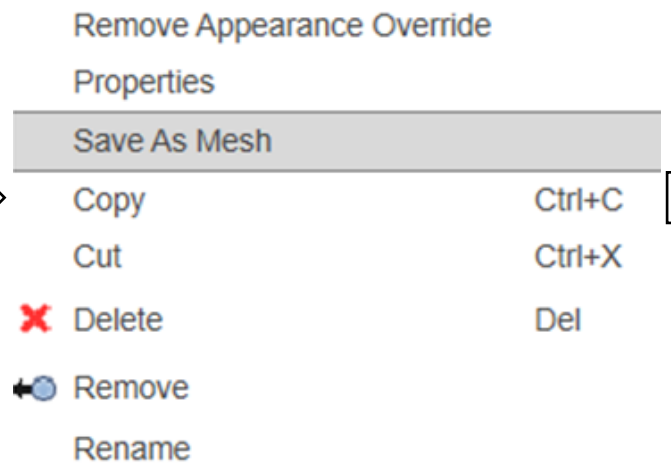


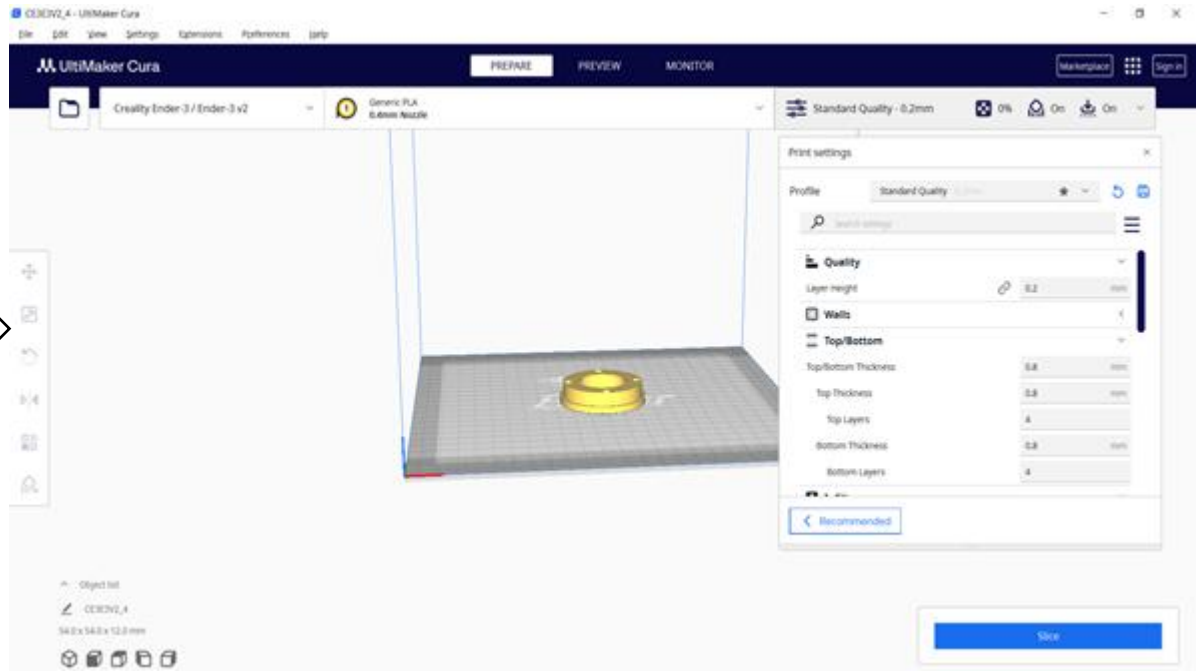
Complete Setup in Simulation Environment

Methodology-[8]
(3D Printing Procedure)

- Fusion 360 to slicing software Cura







Fusion 360 Design

Exporting Design as Mesh

Setting Infill and Layer Height in Cura

Methodology-[9]
(3D Printing Procedure)

- Levelling the bed and start printing

- 0.4mm Nozzle
- 3D printer bed
- Bed level knobs
- Filament
  Material Polylactic Acid(PLA)

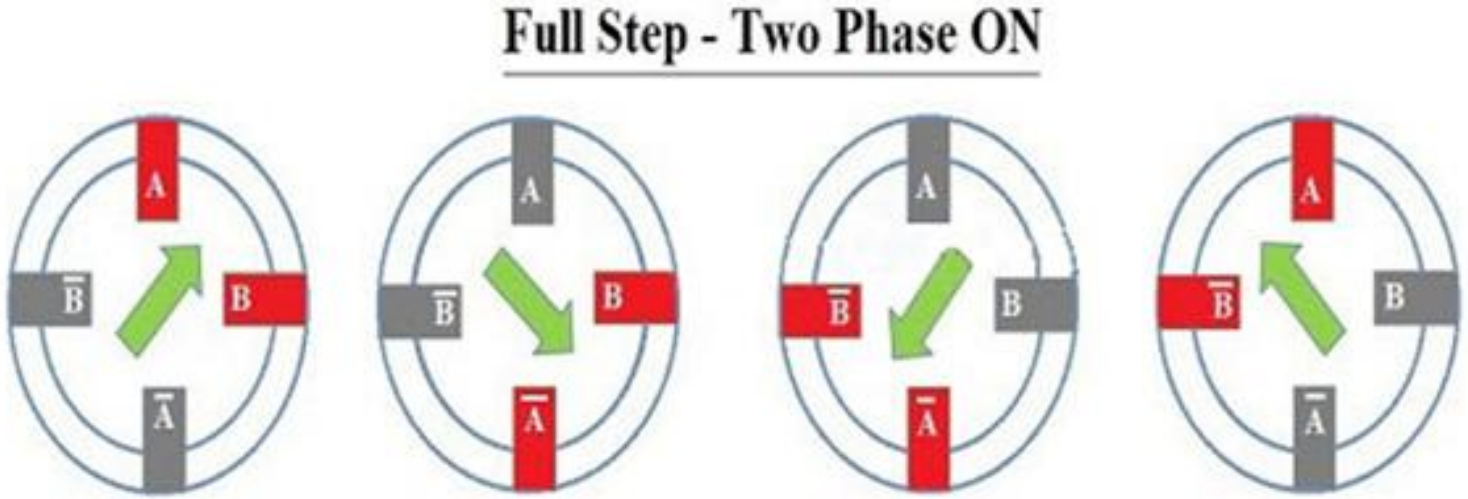
Methodology-[10]
(Stepper Motor Driver Settings)

Micro Step	Pulse/rev	S1	S2	S3
NC	NC	ON	ON	ON
1	200	ON	ON	OFF
2/A	400	ON	OFF	ON
2/B	400	OFF	ON	ON
4	800	ON	OFF	OFF
8	1600	OFF	ON	OFF
16	3200	OFF	OFF	ON
32	6400	OFF	OFF	OFF

Current(A)	PK Current	S4	S5	S6
0.5	0.7	ON	ON	ON
1.0	1.2	ON	OFF	ON
1.5	1.7	ON	ON	OFF
2.5	2.2	ON	OFF	OFF
2.5	2.7	OFF	ON	ON
2.8	2.9	OFF	OFF	ON
3.0	3.2	OFF	ON	OFF
3.5	4.0	OFF	OFF	OFF

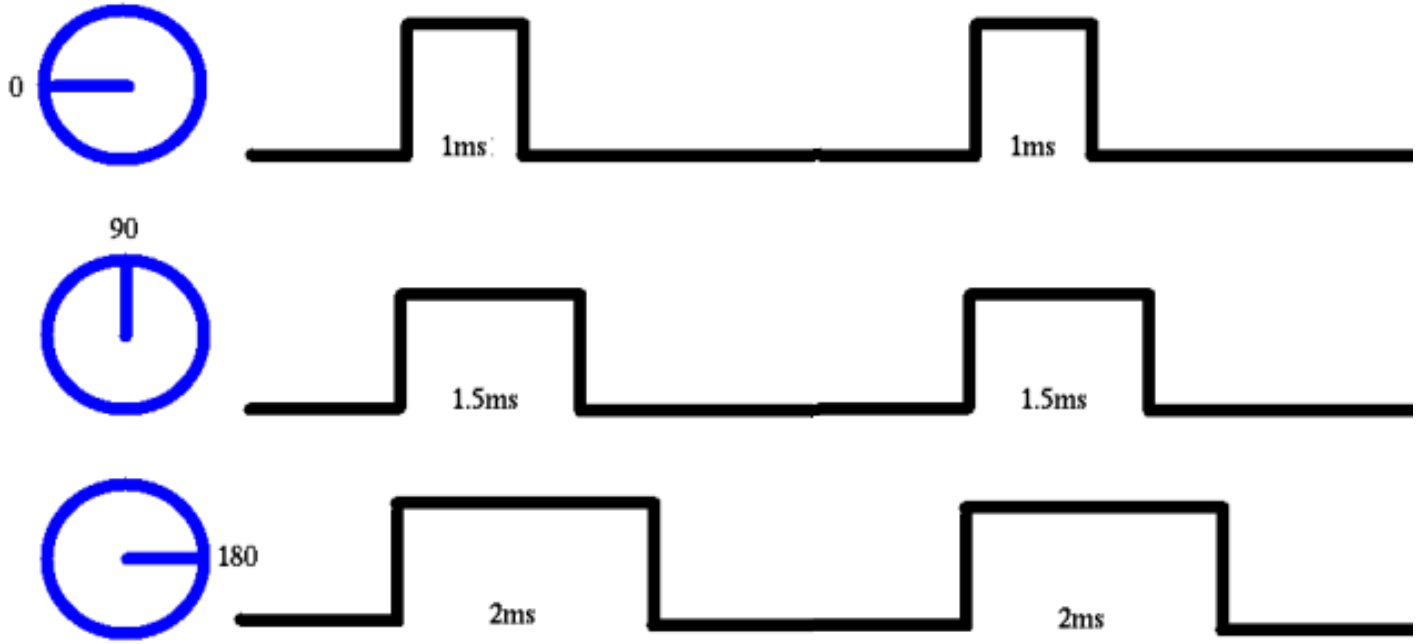
DIP Switch

Methodology-[11]
(Driving the Actuators)



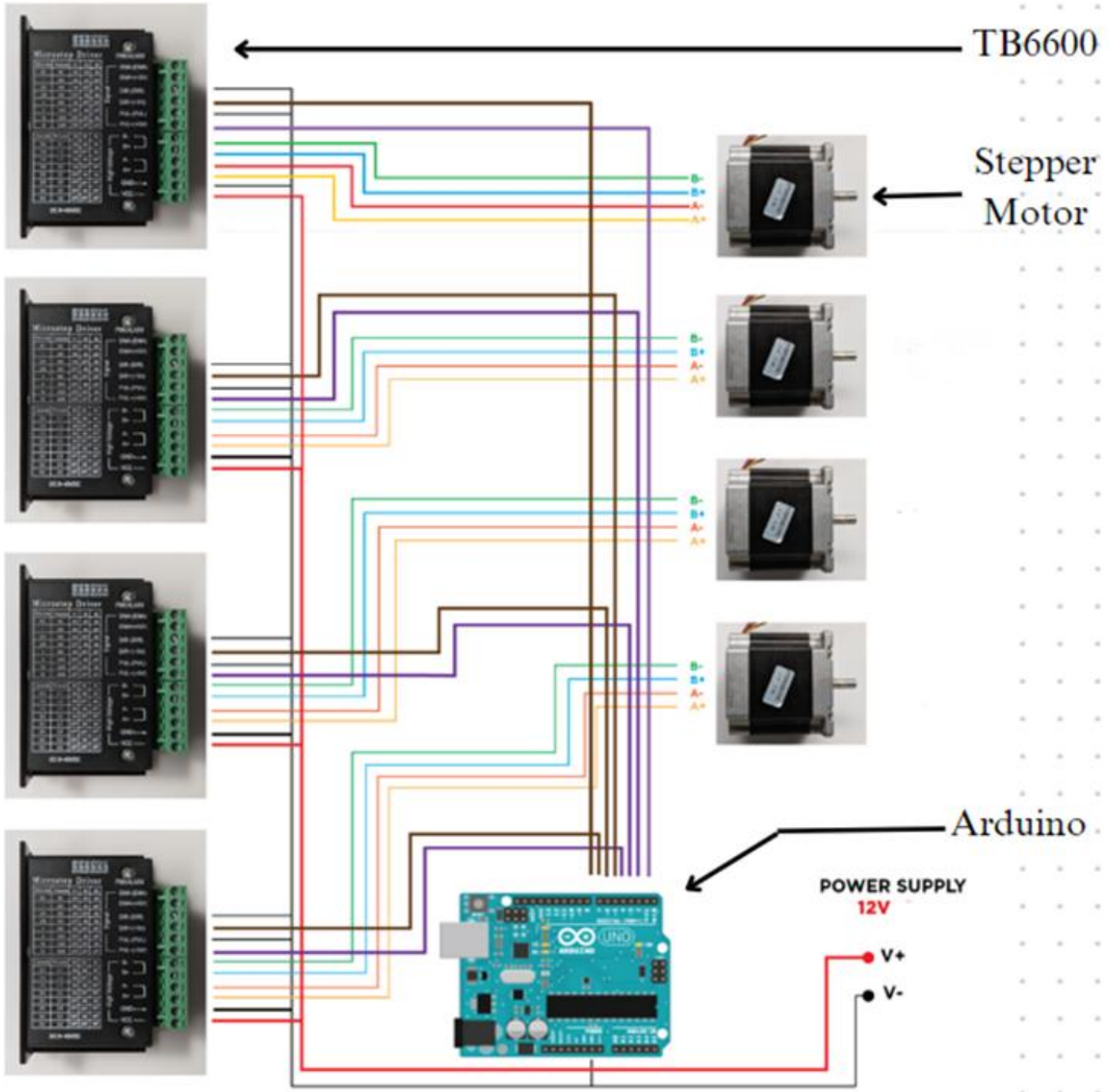
Step	Phase			
	A	B	$\bar{A}$	$\bar{B}$
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

Excitation Method for Stepper Motor

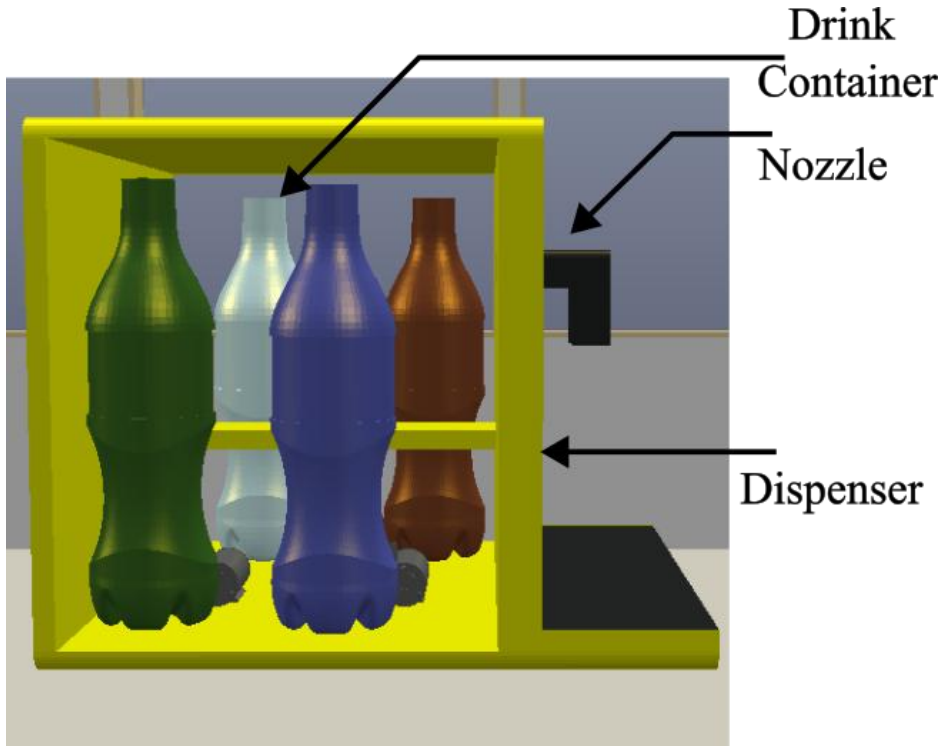


PWM Signal to Servo Motor

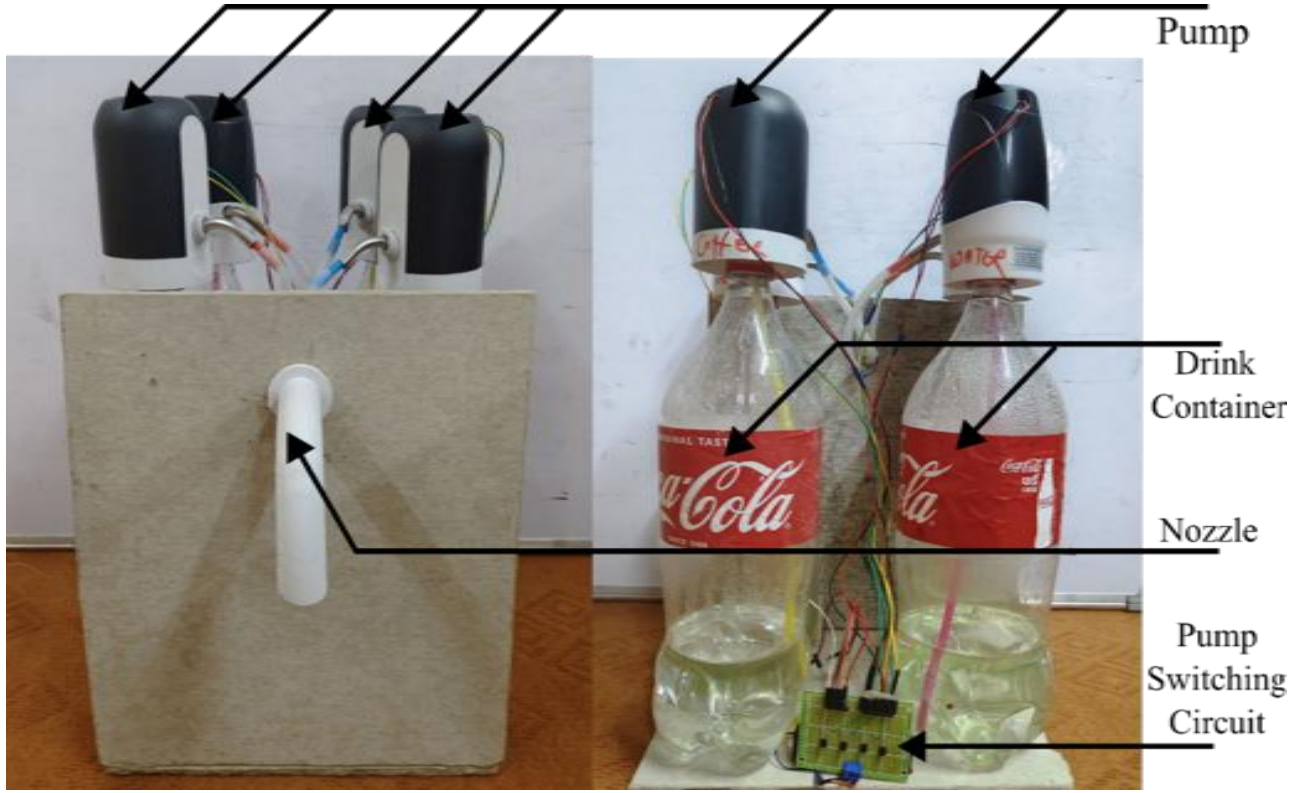
Methodology-[12]
(Driving Stepper Motor)



Methodology-[13]
(Dispenser Design)

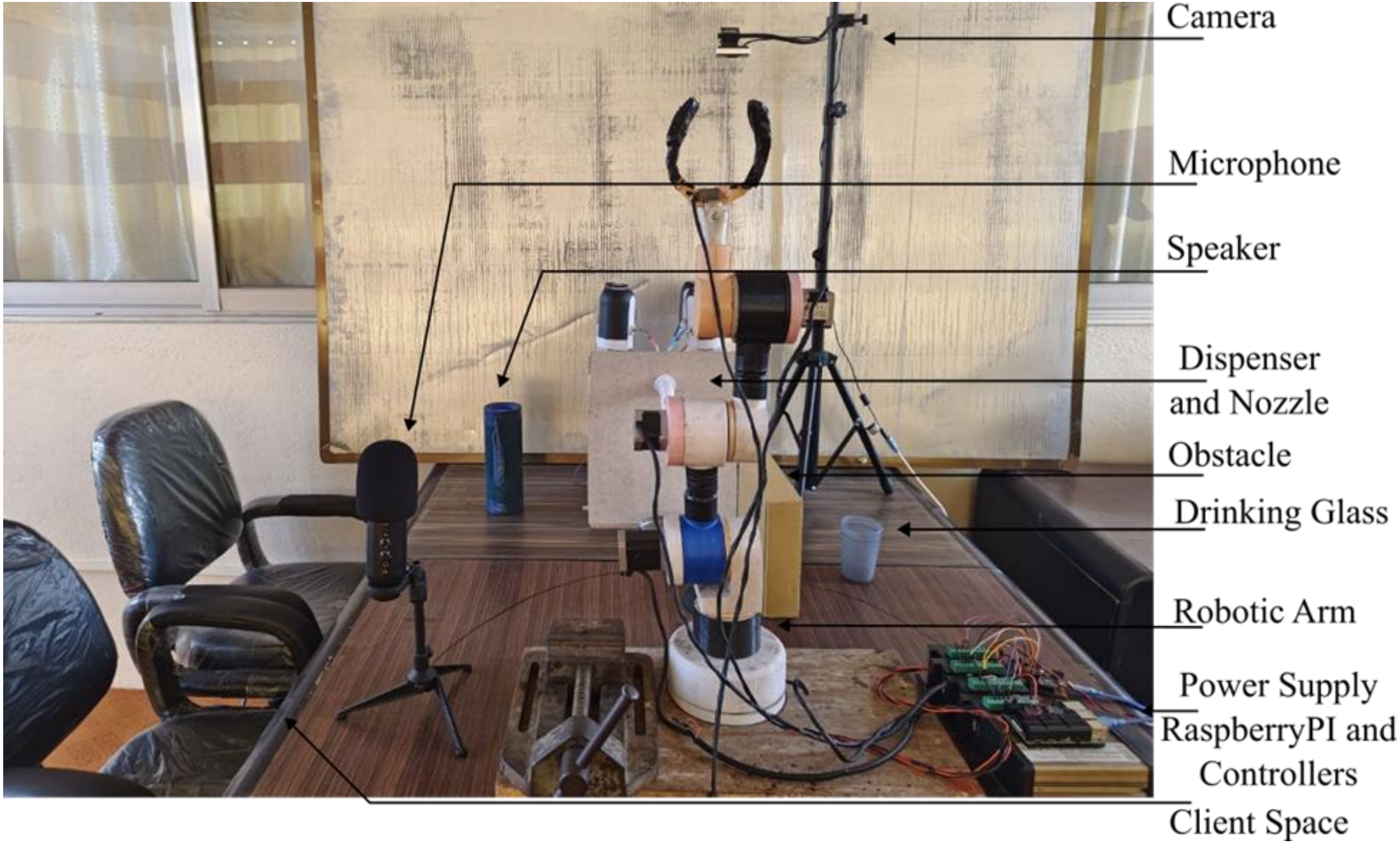


Simulation

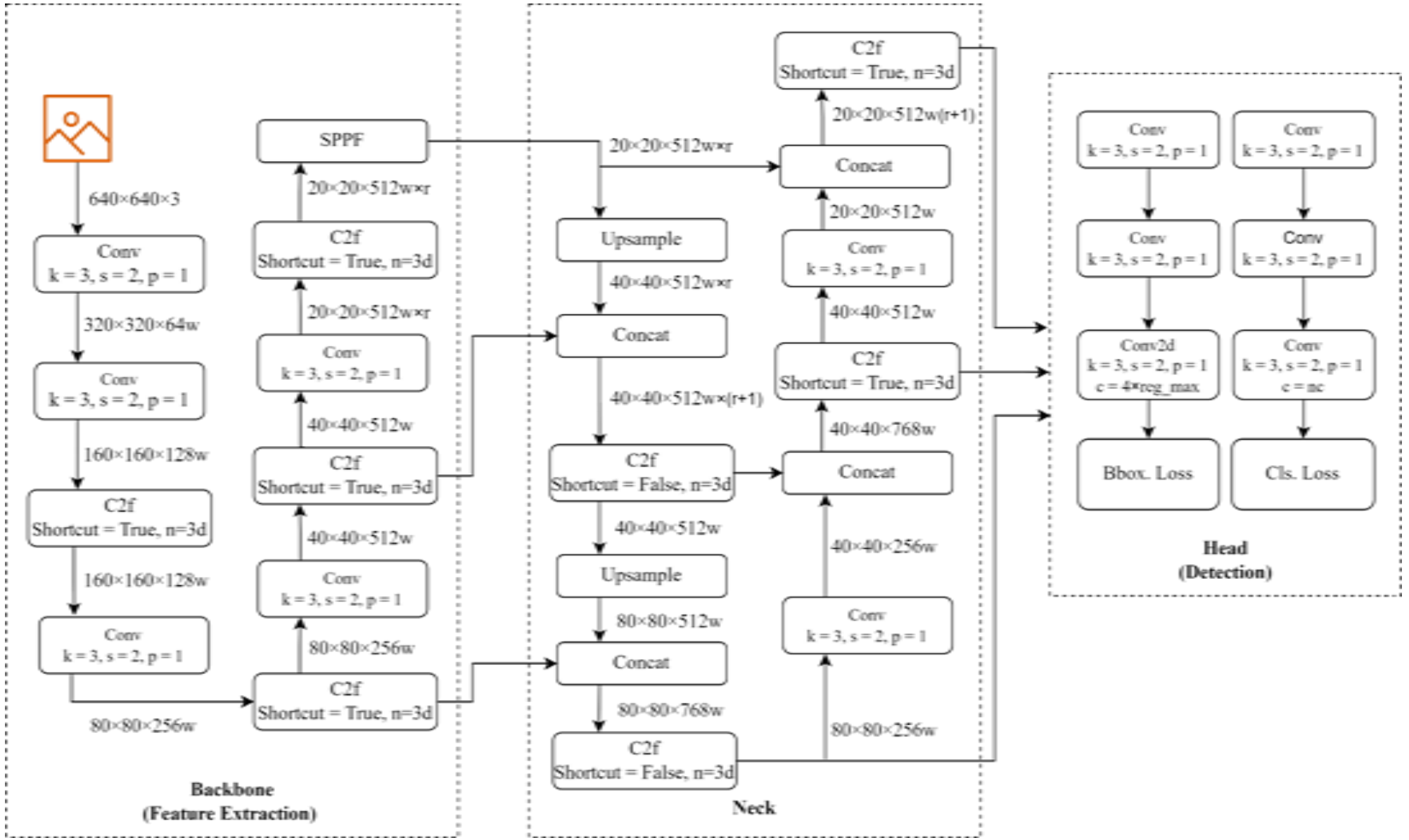


Reality

Methodology-[14] (Complete Physical Setup)



Methodology-[15]
(Object Detection Model)



Methodology-[16]
(Object Detection Model)

- Backbone
  - Receives image of dimension 640*640
  - Extracts the feature from image
- Neck
  - The neck refines backbone features
  - Utilizes series of upscaling, downscaling and concatenation
- Head
  - Head block detects the object form provided images
  - Provides bounding box, label classes and accuracy scores

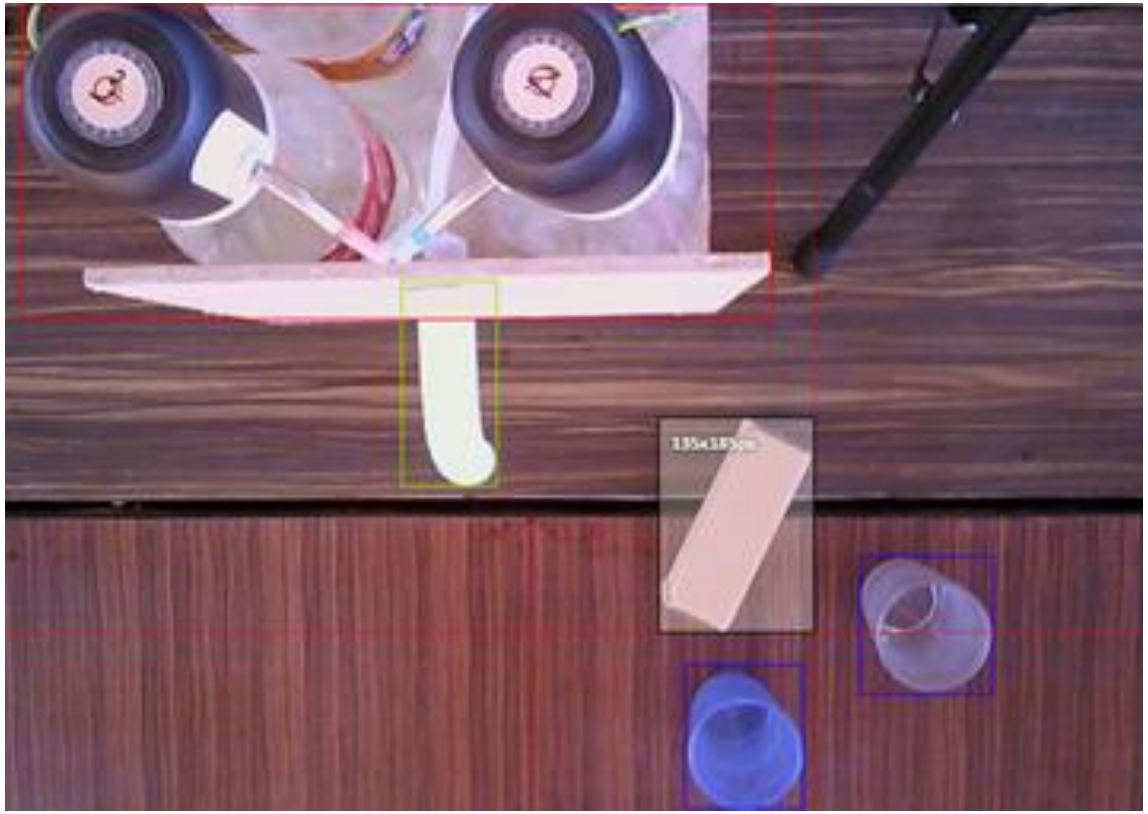
Methodology-[17]
(Image Dataset Collection Procedure)

- Diverse object pictures taken within CoppeliaSim for simulation
- Varied lighting condition for adaptability
- Real-world objects on tabletop environment taken for dataset
- Smartphone and webcam used for image capturing

Parameters	Value(Simulation)	Value(Reality)
Field of View	75°	60°
Resolution	356 ×356	1280 ×720

Camera Parameters

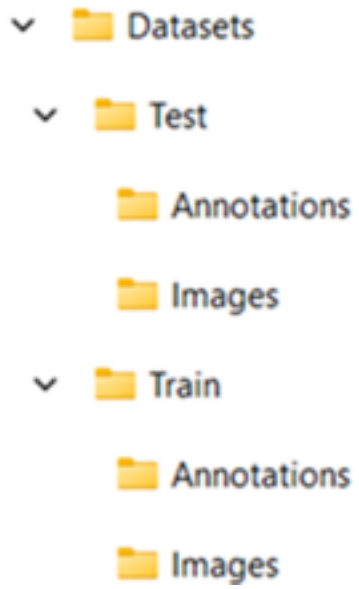
Methodology-[18]
(Image Dataset Annotation)



Annotating Image in CVAT

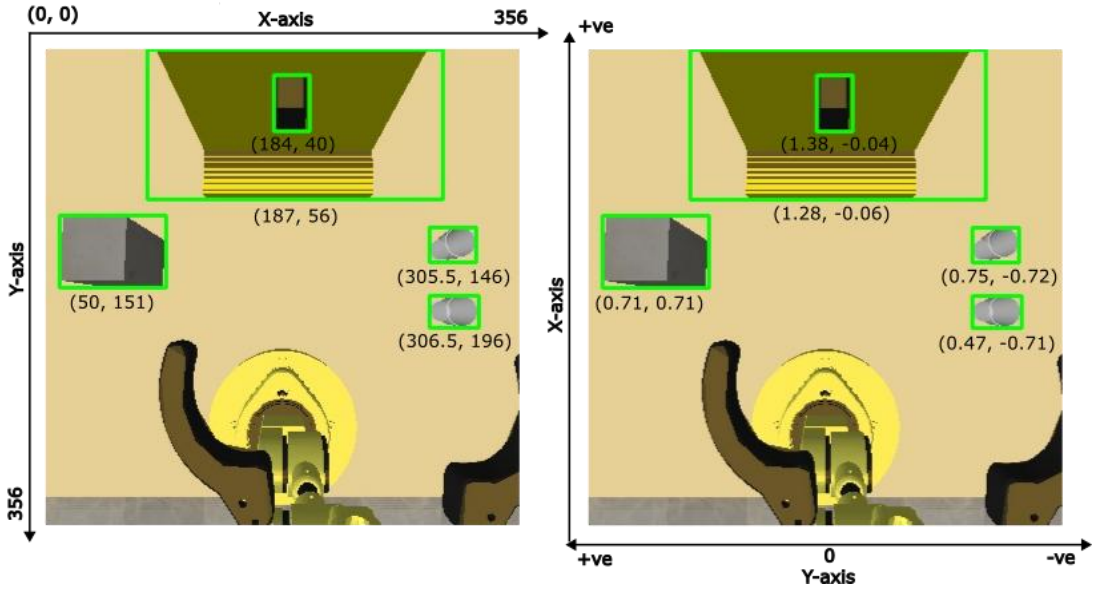
Class Index	Class Name	Center Coordinates		Width	Height
		x	y		
1	Glass	0.39	0.18	0.50	0.37
1	Glass	0.75	0.74	0.09	0.16
2	Dispenser	0.63	0.87	0.08	0.17
3	Nozzle	0.62	0.61	0.10	0.25
4	Obstacle	0.43	0.45	0.06	0.24

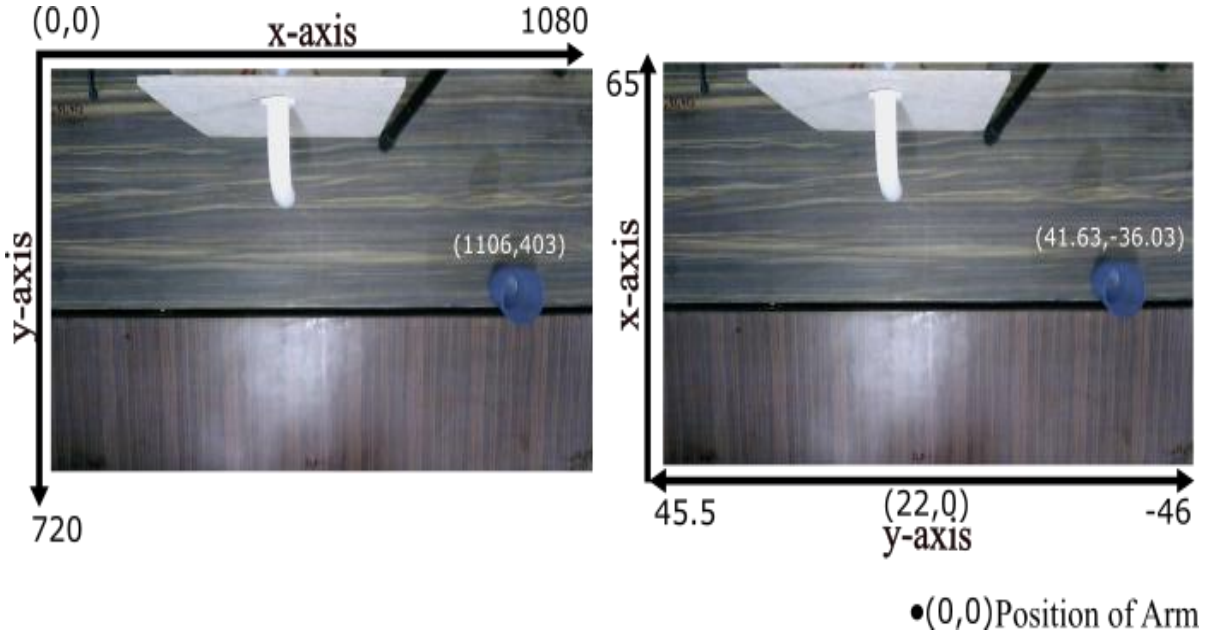
Bounding Box Parameters



Darknet Format

Methodology-[19] (Coordinate Mapping)





Simulation Coordinate System

Reality Coordinate System

$$\begin{bmatrix} -4.10e-04 & -6.12e-02 & 6.44e+01 \\ -7.12e-02 & 1.34e-03 & 4.42e+01 \\ 1.9e-08 & -1.38e-04 & 1.00 \end{bmatrix}$$

Simulation Transformation Matrix

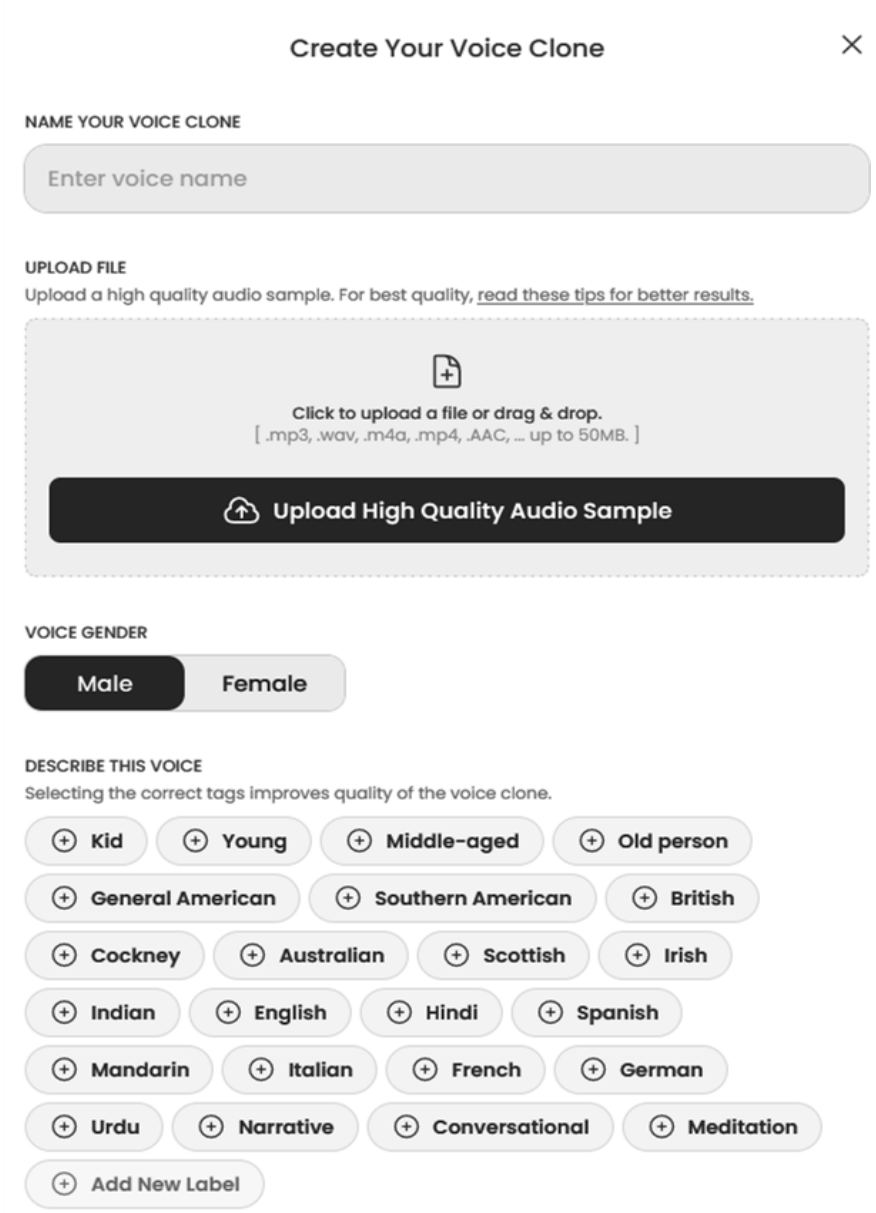
$$\begin{bmatrix} 9.69e-06 & -5.75e-03 & 1.63 \\ -5.97e-03 & -3.37e-08 & 1.06 \\ -6.63e-07 & 4.47e-04 & 1.00 \end{bmatrix}$$

Reality Transformation Matrix

Methodology-[20] (Anomalies and Feedbacks)

S.N.	Problem	Description	Voice-Based Feedback
1.	Path Obstruction	Objects obstruct the robotic arm's movement trajectory	There’s an obstacle in the way. please clear the path.
2.	Glass Unavailable	Empty glasses are not available	I’m sorry but we are currently out of that drink, would you like something else.
3.	Drinking Glass Unreachable	The drinking glass is out of reach of the robotic arm	The glasses are out of reach, please place it within the reach of robotic arm.
4.	Vision Sensor Blockage	The vision sensor is obstructed, affecting accuracy.	The vision is obstructed, please check the camera.
5.	Incorrect Command	The drink command doesn’t contain available drink name	I’m sorry, I didn’t understand your command, please give me a drink related command.

Methodology-[21] (AI Voice Generation)

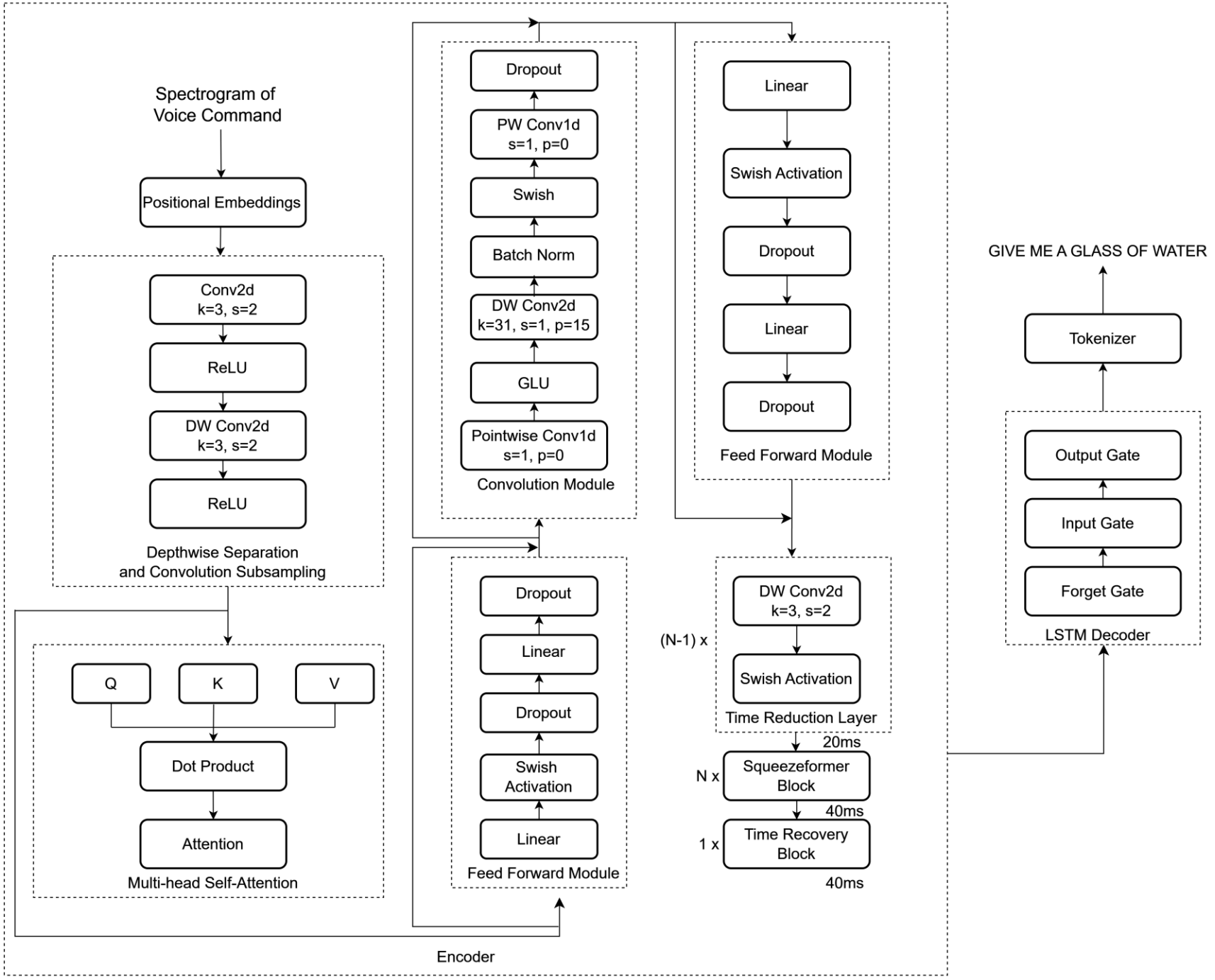


These snapshots depict the process of giving text input to create AI voice clone for feedbacks on anomalies.

Methodology-[22]
(Voice Data Augmentation Techniques)

Augmentation	Formula
Gaussian Noise	$\left[f(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} exp\left[\frac{-1}{2}\left(\frac{x-\mu}{\sigma}\right)\right)^2\right]$
Time-Shifting	$y(t) = x(t + \Delta t)$ $y(t) = x(t - \Delta t)$
Pitch Shifting	$y(t) = x(t/\alpha)$
Random Gain	$Y(t) = X(t) * G$
Time Masking	$Y(t) = X(t) * M(t)$
Frequency Masking	$Y(f) = X(f) * M(f)$

Methodology-[23]
(Speech Recognition Model Architecture)



Methodology-[24]
(Speech Recognition Model Architecture)

- Spectrogram and positional embedding pass-through Conv layers
- Next, it undergoes MHSA (Multi-Head Self-Attention) layer
- The output further passes through a Feed Forward layer
- Subsequently, it is processed by a Convolution layer
- Followed by another Feed Forward operation and Time Reduction
- Then, time recovery is applied
- Full transcription is achieved through an LSTM decoder

Methodology-[25]
(Voice Commands)

Drink	Variation 1	Variation 2	Variation 3	Variation 4	Variation 5
Water	Can I have a cup of water	Hey arm can I would like a cup of water	Can you get me a cup of cold water	Fill up a cup with water please	Serve me a cup of fresh water
Sprite	Pour me a cup of sprite	Hey arm can I have a cup of chilled sprite	Fill up a cup with sprite	I would like a cup of sprite	Serve me a cup of sprite
Orange Juice	Pour me a cup of orange juice	Fill up a cup with orange juice	Hey arm I would like a cup of orange Juice	Provide me with a cup of orange juice	Pour me a cup of delicious orange juice
Coffee	Provide me with a cup of coffee	Fill up a cup with coffee	I would like a cup of coffee	Can you get me a cup of coffee	Hey Arm Make me a cup of delicious coffee

Methodology-[26]
(Training and Finetuning Datasets for Speech Model)

Subset	Use	Hours	Minutes per Speaker	Female Speakers	Male Speakers	Total Speakers
train-clean-100	Pretraining	100.6	25	125	126	251
train-clean-360	Pretraining	363.6	25	439	482	921
train-other-500	Pretraining	496.7	30	564	602	1166
Thapathali Campus	Finetuning	8.2	1.2	48	57	105
Nepal Mega School	Finetuning	2.6	1.2	16	17	33
Texas International School	Finetuning	5.44	1.2	37	32	69

Methodology-[27]
(Dataset Collection Snapshots)

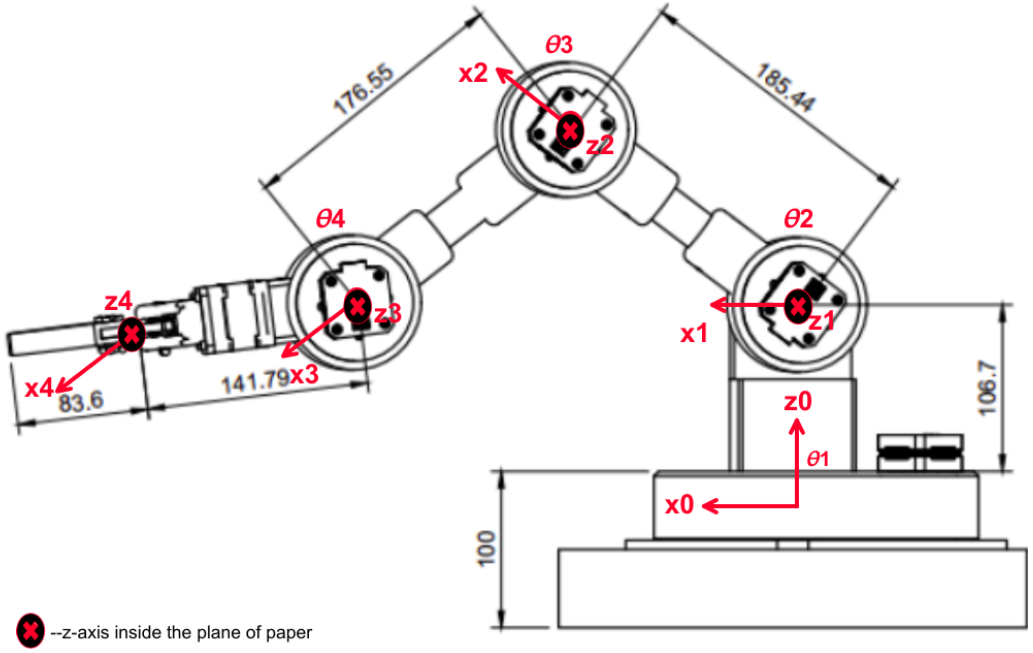
Texas International School

Nepal Mega School

Thapathali Campus

Methodology-[28]
(Kinematics of Robotic Arm)

- Deals with the motion of robotic arm
- Doesn’t consider forces and torques associated with it
- Provides the homogeneous transformation matrix
- Denavit-Hartenberg (D-H) parameters are used to find the transformation matrix



Methodology-[29]
(D-H Parameters and Homogeneous Transformation Matrix)

$$H_4^0 = \begin{bmatrix} c_1c_{234} & -c_1s_{234} & s_1 & c_1(a_4c_{234} + a_2c_2 + a_3c_{23}) \\ c_{234}s_1 & -s_1s_{234} & -c_1 & s_1(a_4c_{234} + a_2c_2 + a_3c_{23}) \\ s_{234} & c_{234} & 0 & a_4s_{234} + a_2c_2 + a_3s_{23} + a_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Homogeneous Transformation Matrix

n (link)	Parameters			
	θ	α	r	d
1	θ_1	90°	0	a_1
2	θ_2	0	a_2	0
3	θ_3	0	a_3	0
4	θ_4	0	a_4	0

- D-H parameters can be used to derive the homogeneous transformation matrix

Methodology-[30]
(Homogeneous Transformation Matrix)

$$H_4^0 = \begin{bmatrix} c_1c_{234} & -c_1s_{234} & s_1 & c_1(a_4c_{234} + a_2c_2 + a_3c_{23}) \\ c_{234}s_1 & -s_1s_{234} & -c_1 & s_1(a_4c_{234} + a_2c_2 + a_3c_{23}) \\ s_{234} & c_{234} & 0 & a_4s_{234} + a_2c_2 + a_3s_{23} + a_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Provides rotation matrix as well as displacement vector of end effector w.r.t base frame
- The first three elements of the third column provide equations for forward kinematics

Methodology-[31]
(Forward Kinematics and Inverse Kinematics)

- Forward Kinematics provides position and orientation of an end effector based on known joint angles
- Inverse Kinematics aims to find the joint angles necessary to reach a particular position and orientation in the workspace.

Desired End Effector Position in Task Space (m)		
X	Y	Z
0.587	-0.764	0.05

Desired Joint Angles(deg)			
θ_1	θ_2	θ_3	θ_4
-52.2	-72	-76.2	141.2

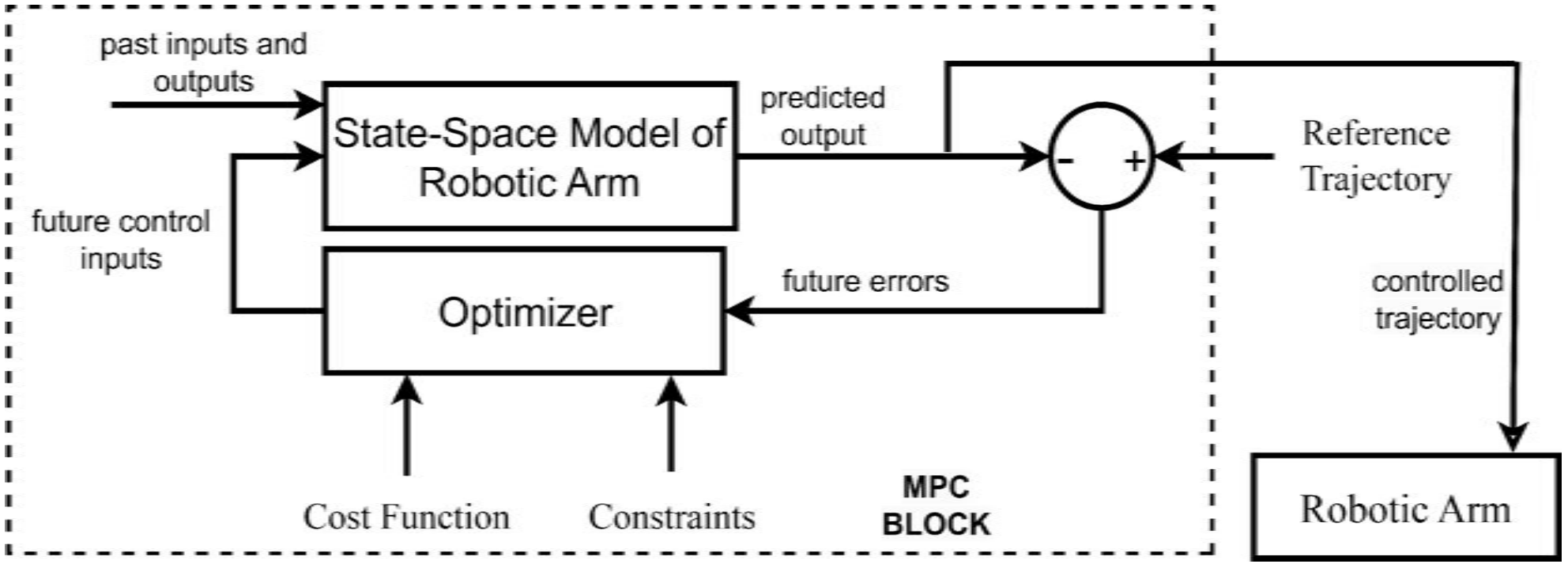
Methodology-[32]
(Inverse Kinematics Via Gradient Descent)

Algorithm:

- Start with H_d and H_k at step k.
- Evaluate: $\Delta x = \begin{bmatrix} e_p \\ e_o \end{bmatrix}$
- x=$FK(\theta_i)$
  - $\Delta x = J\Delta\theta$
  - $\Delta\theta = J^{+}\Delta x$
- Incrementing θ_k to converge end effector pose on the desired pose:
  - $\theta_{k+1} = \theta_k + \alpha J^{+}\Delta\theta_k$
- Repeat the process until$||\theta_k|| \approx 0$.

Symbol	Remarks
H_k	Current homogeneous transformation matrix of end effector w.r.t base
α	Step size
H_d	Desired homogeneous transformation matrix of end effector w.r.t base
Δx	Vector: difference in desired and actual pose of the end effector
e_p, e_o	Vectors: position and orientation error between current and desired pose
θ_k	Vector: joint angles at time instant k
$FK(\theta_i)$	Forward Kinematics method; provides end effector x, y, z position according to given joint angles
$\Delta\theta$	Vector: Change in joint angles
J^{+}	Pseudo-Inverse of Jacobian matrix

Methodology-[33] Model Predictive Control



Reference Trajectory	Trajectory of 4 joint angles
Predicted Output	4 joint angles at each time step
Cost Function	Reference Tracking and Input Changes
Future Inputs	Control Input to change the state of the model

Methodology-[34] Model Predictive Control

- Dynamic Equation

$$\ddot{\theta} = -D^{-1}C\dot{\theta} - D^{-1}g + D^{-1}\tau$$

- State-Space Model

$$x_{k+1} = Ax_k + Bu_k$$
$$z_k = Cx_k$$

- Cost Function

$$\min_u (J_z + J_u)$$

Where,

$$J_z = (z^d - z)^T W_4 (z^d - z)$$

$$J_u = (W_1 u)^T W_2 (W_1 u)$$

Symbol	Remarks
x_k	State
u_k	Control Input
z_k	Output to Control
J_z	Cost function corresponding to tracking error
J_u	Cost function corresponding to change in inputs
D	Manipulator inertia matrix
g	Acceleration due to gravity
τ	Vector of actuator torques
$\dot{\theta}$	Angular velocity of joints
$\ddot{\theta}$	Acceleration of joints

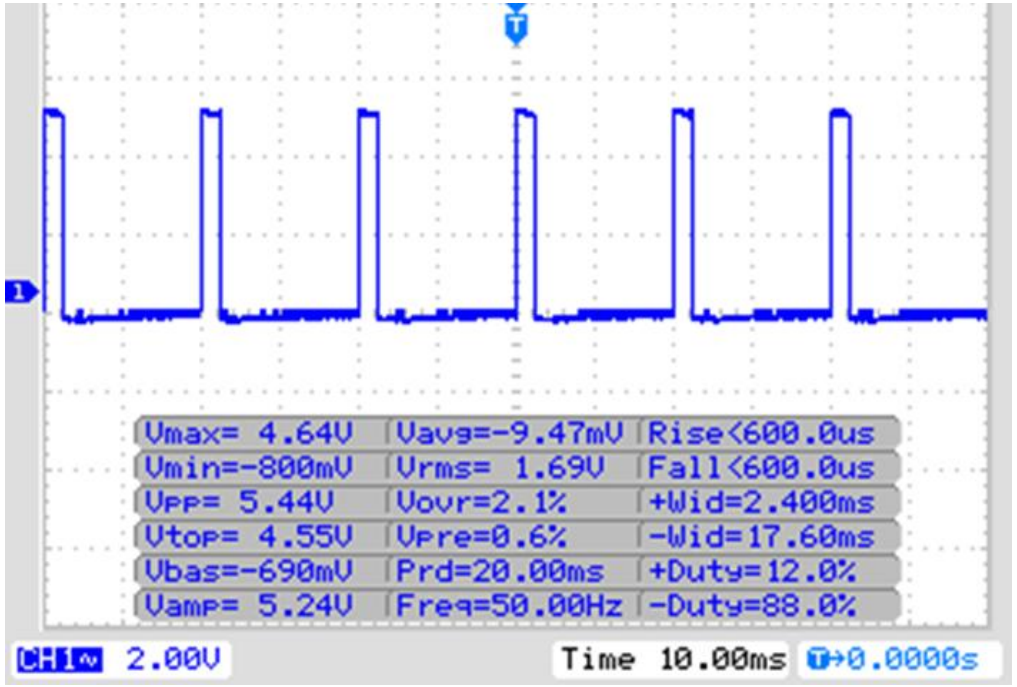
- Optimizer

$$\frac{d(J_z + J_u)}{du} = 0$$

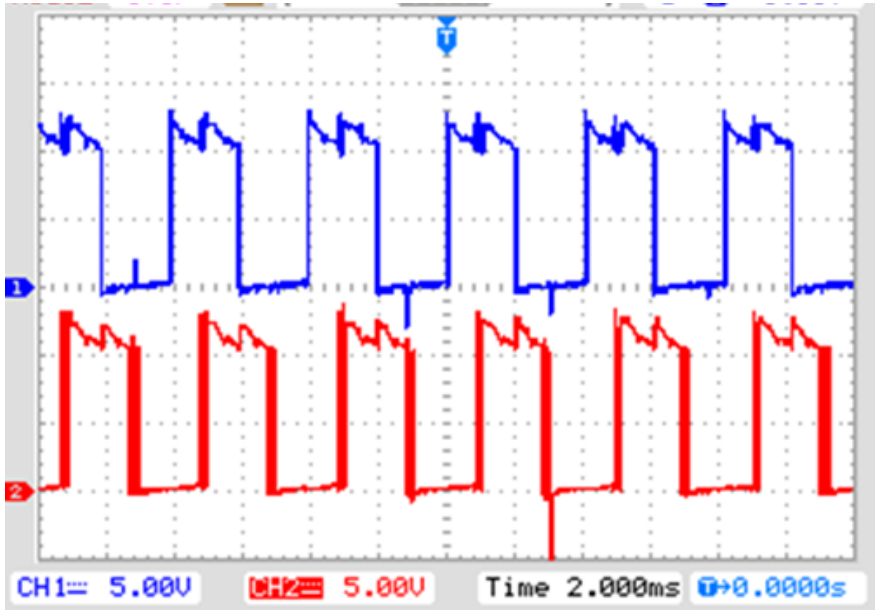
$$\hat{u} = (M^T W_4 M + W_3)^{-1} M^T W_4 s$$

$$\hat{u} = \begin{bmatrix} \hat{u}_k \\ \hat{u}_{k+1} \\ \dots \\ \hat{u}_{k+v-1} \end{bmatrix}$$

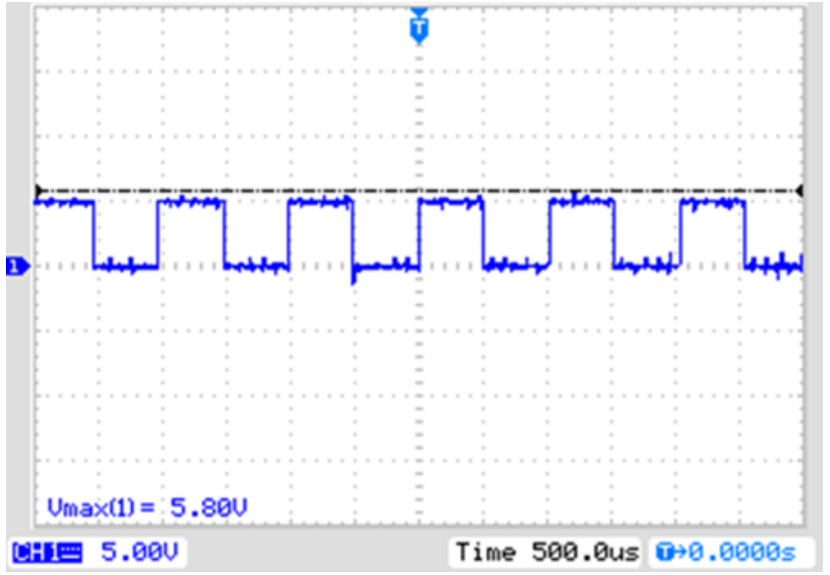
Results-[1]
(Driving Actuators)



Input to servo motor(180 degree)



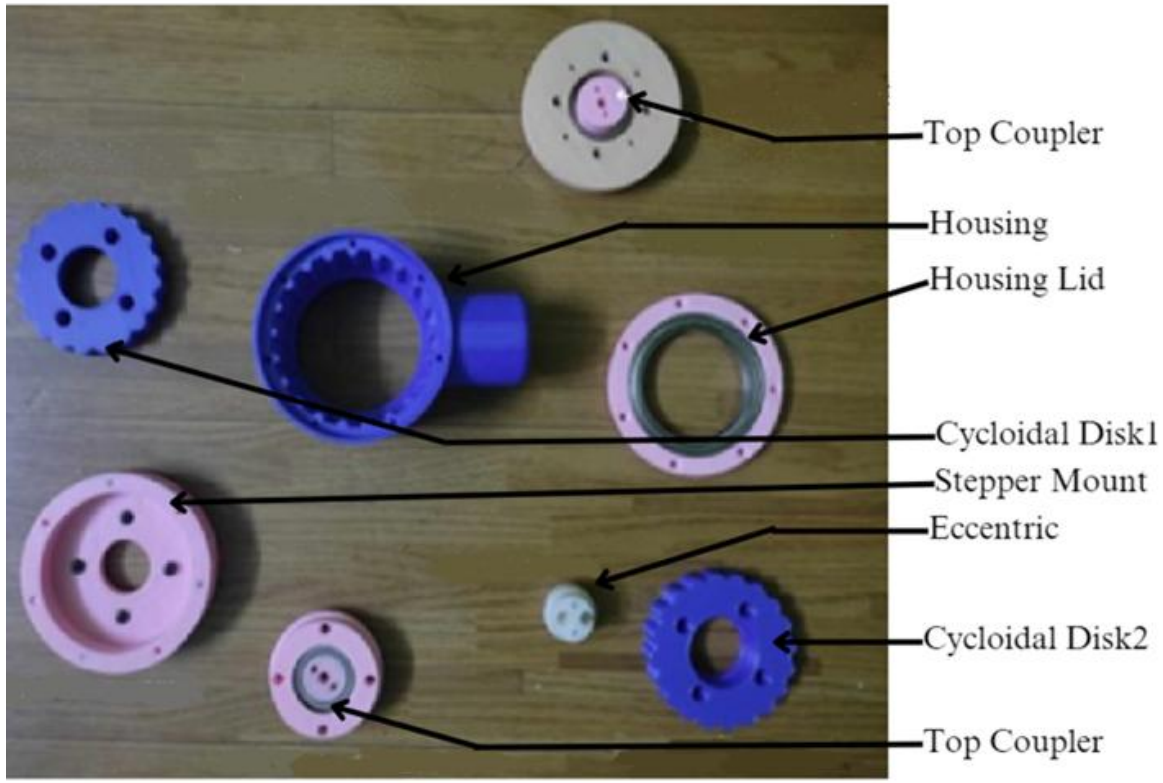
Input to Stepper Motor



Input to Stepper Driver

Results-[2]
(3D Printed Parts)

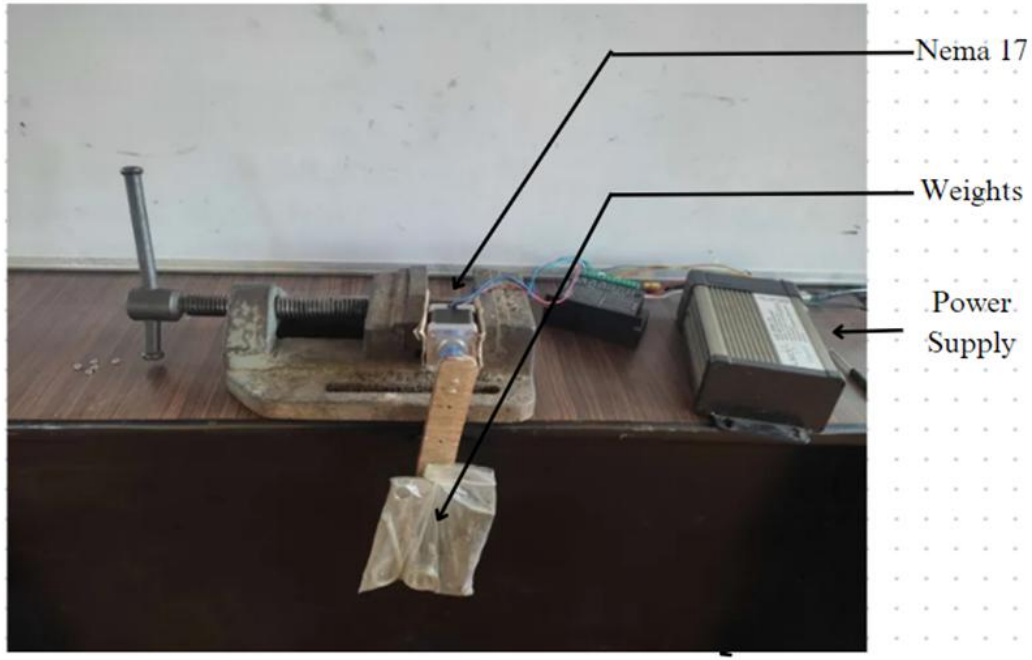
Printed Parts

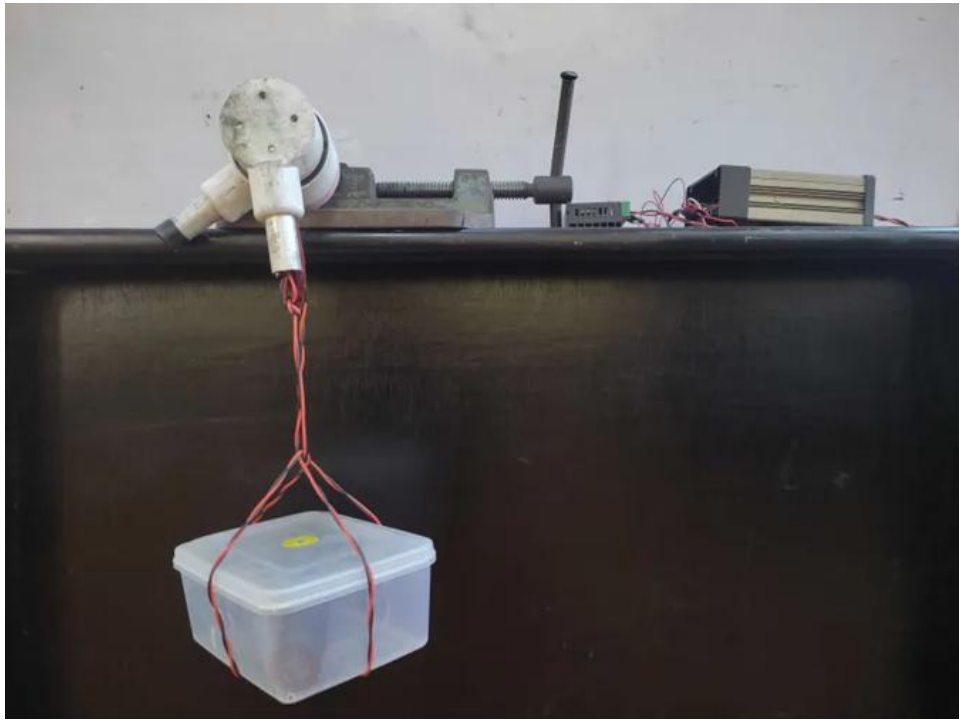


Lathe Turned Shaft

Lidless Assembled Joint

Results-[3]
(Torque Test of Stepper Motor)





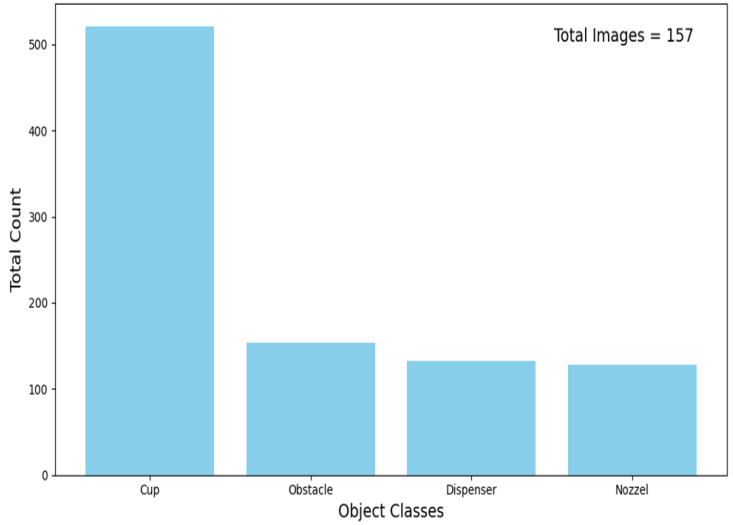
Nema 17 Torque (25.32 Ncm)

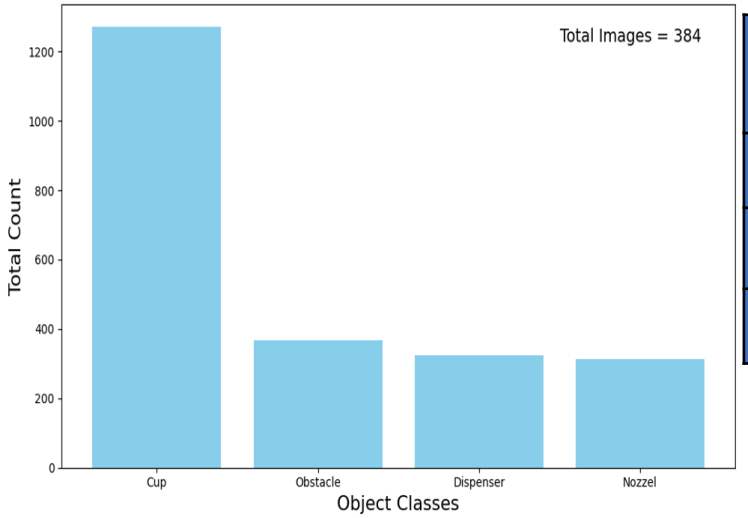
Gearbox Torque Test

Length(cm)	Mass(kg)	Torque (Nm)	Remark
10	0.5	0.5	Stable
10	1	1	Stable
10	1.5	1.5	Stable
10	2	2	Stable
15	1.74	2.61	Stable
15	2.5	3.75	Unstable

Results-[4]
(Datasets Discussion for Object Detection Model)

Simulation

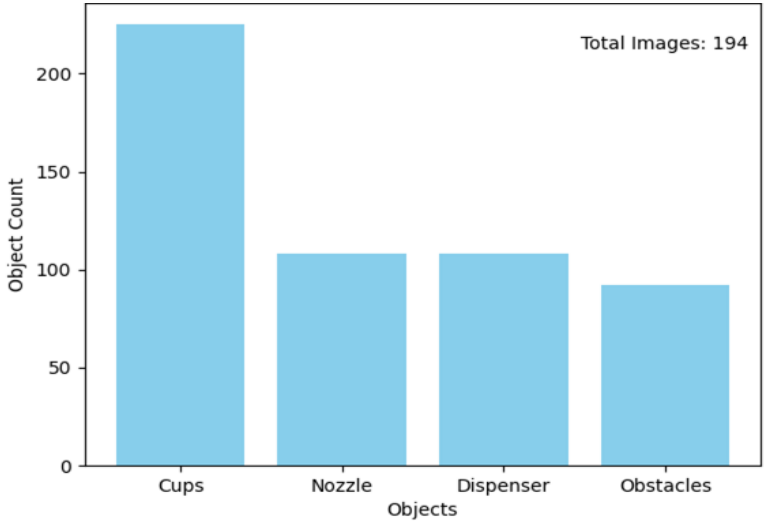


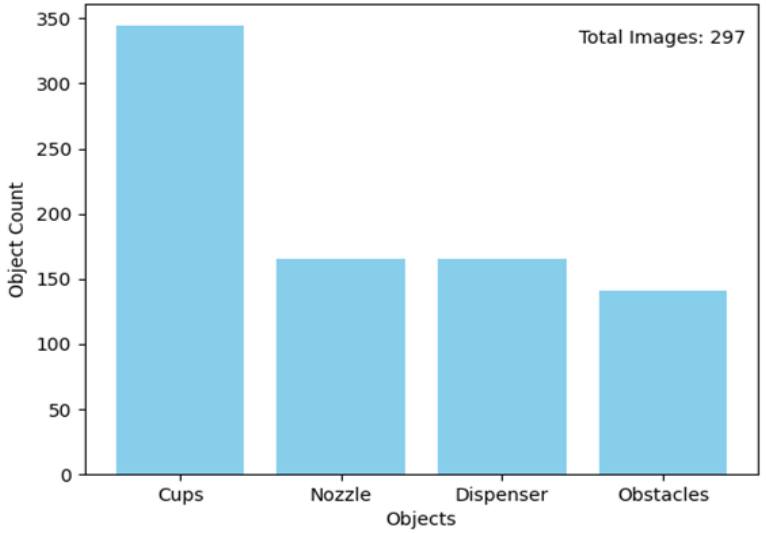


Subset	Object Count	Image Count
Train2017	860,001	118,287
Val2017	36,553	5,000
Test2017	147905	20,288

Pre-trained Dataset

Reality





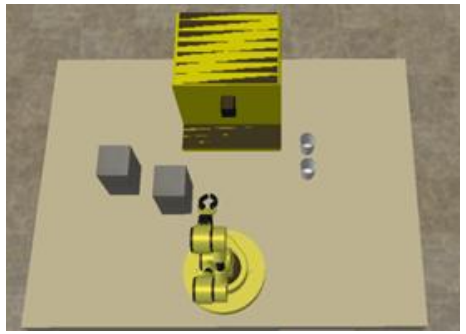
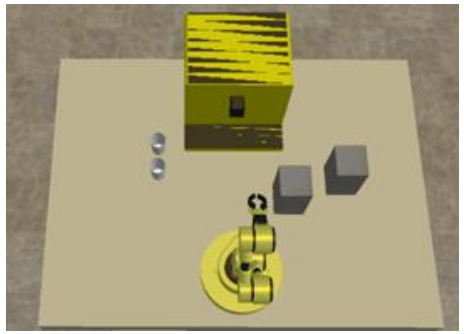
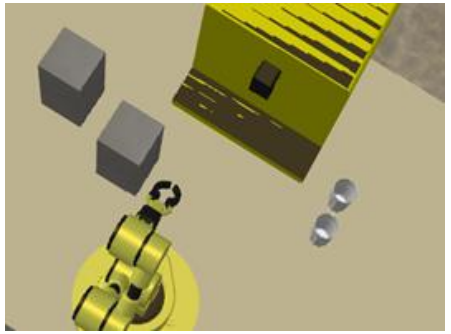
- COCO-2017 dataset is used for pretraining
- Contains 20000 annotated images
- It encompasses 80 distinct object categories

Before Augmentation

After Augmentation

Results-[5]
(Data Augmentation for Object Detection Model)

Simulation

Reality

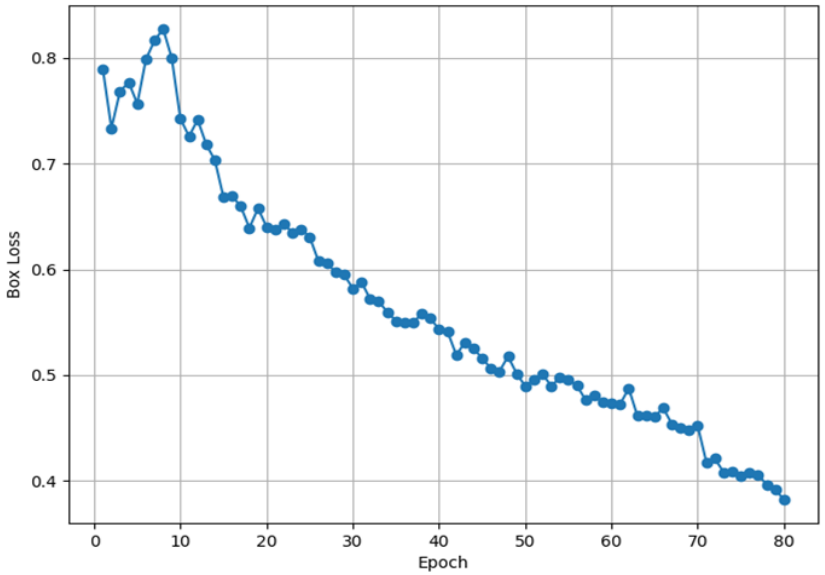


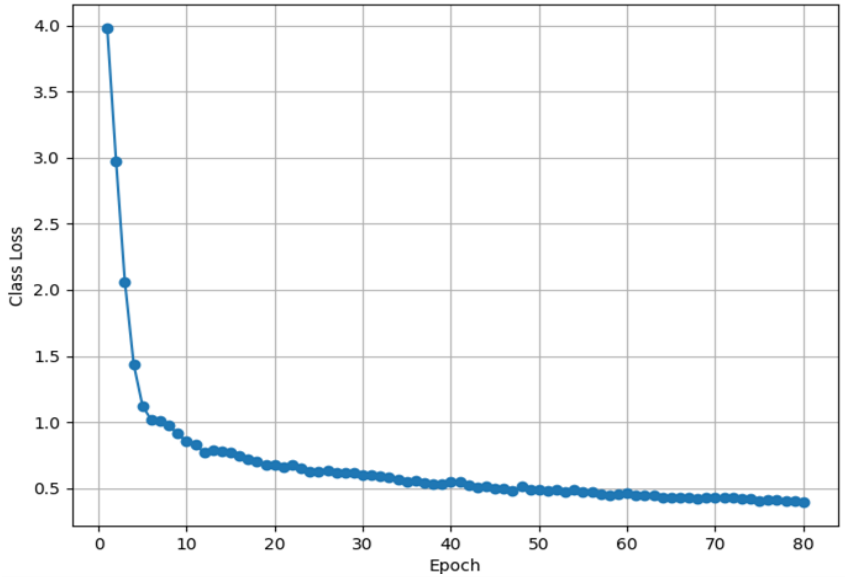
Original Image | Flipped Image | 30-degree Rotation

Augmentation	Transformation Matrix
Flip	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
30-degree CW Rotation	$\begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

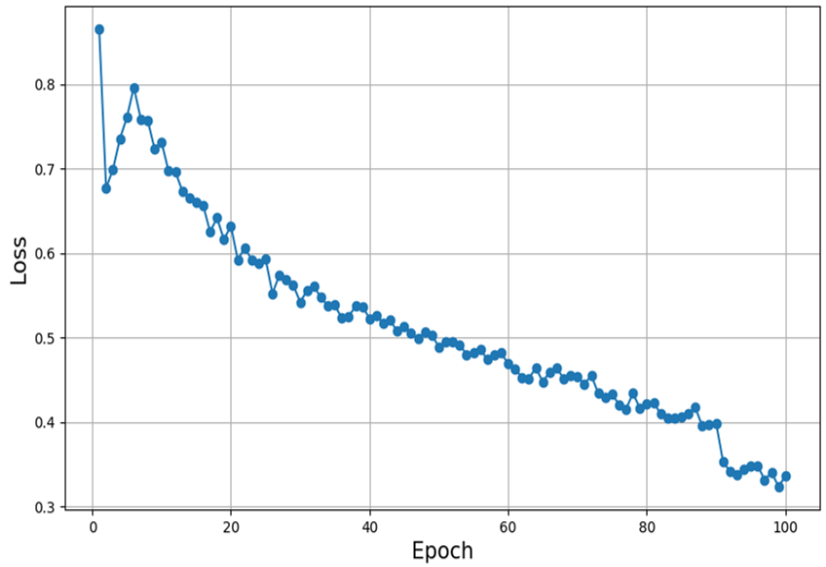
Results-[6] (Object Detection Model (Losses))

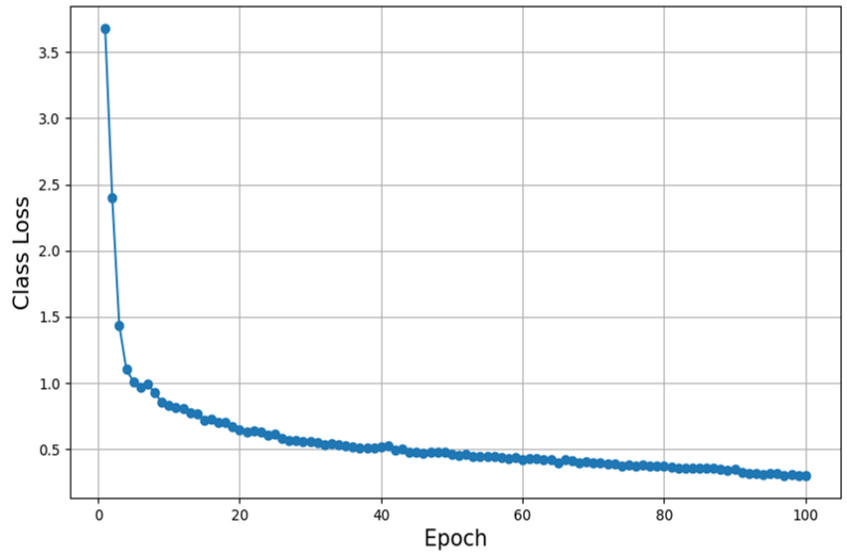
Simulation





Reality



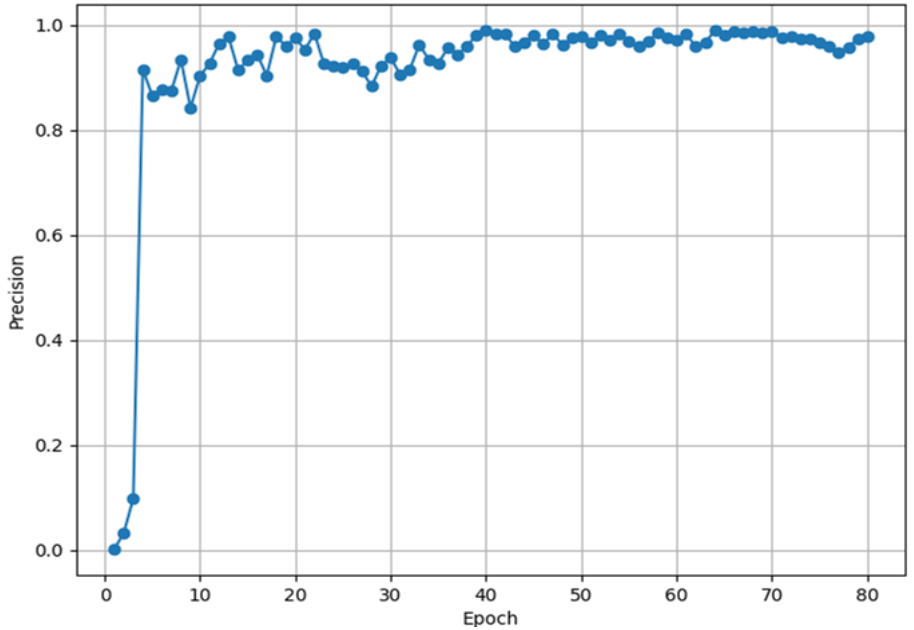


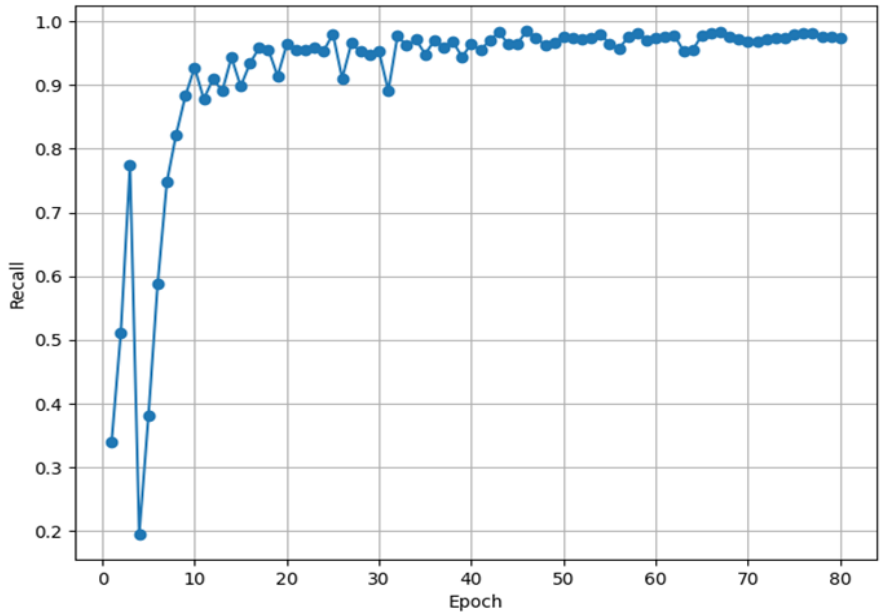
Box Loss vs Epoch

Class Loss vs Epoch

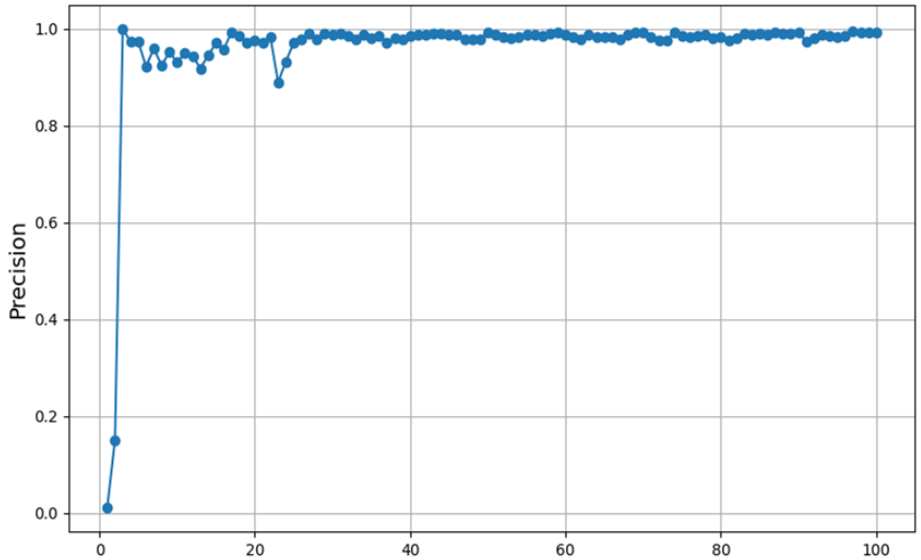
Results-[7]
(Object Detection Metrics)

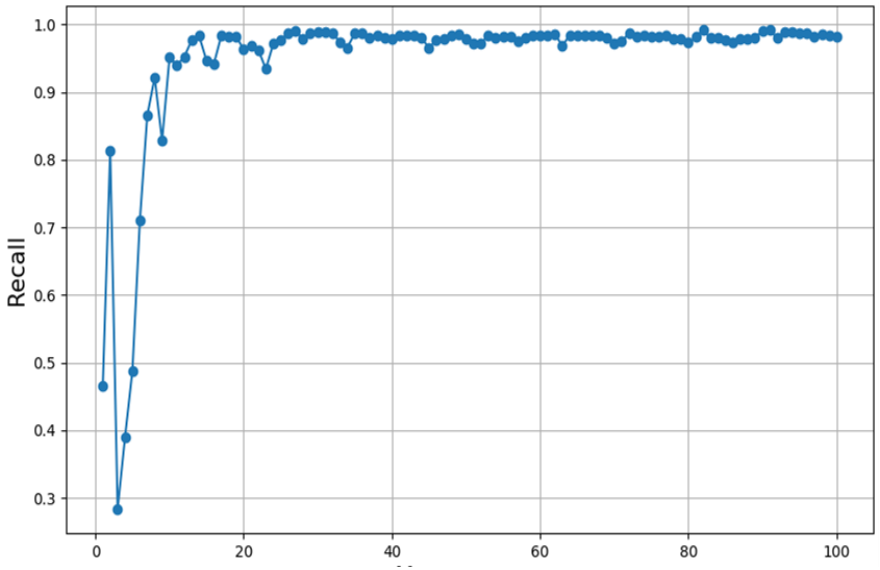
Simulation





Reality





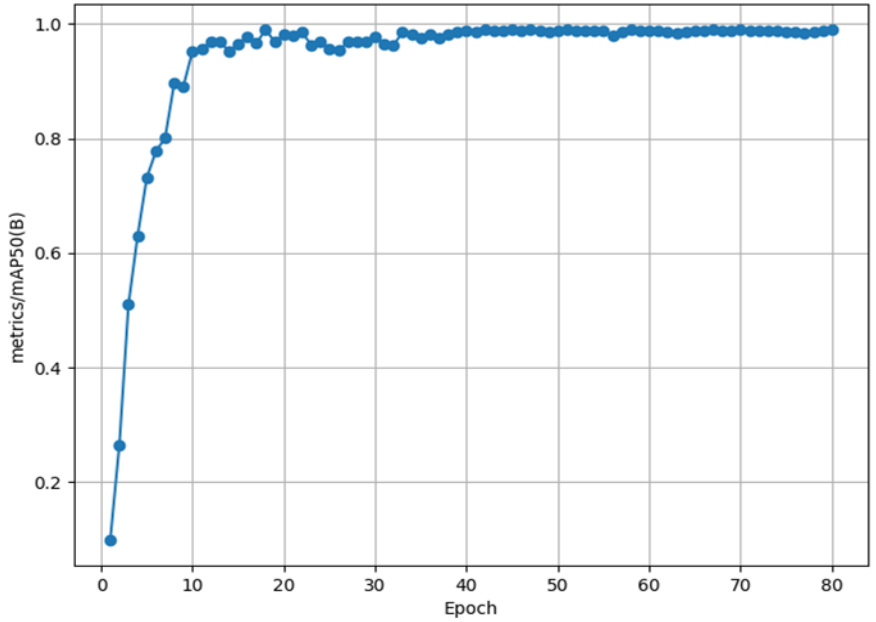
Precision vs Epoch

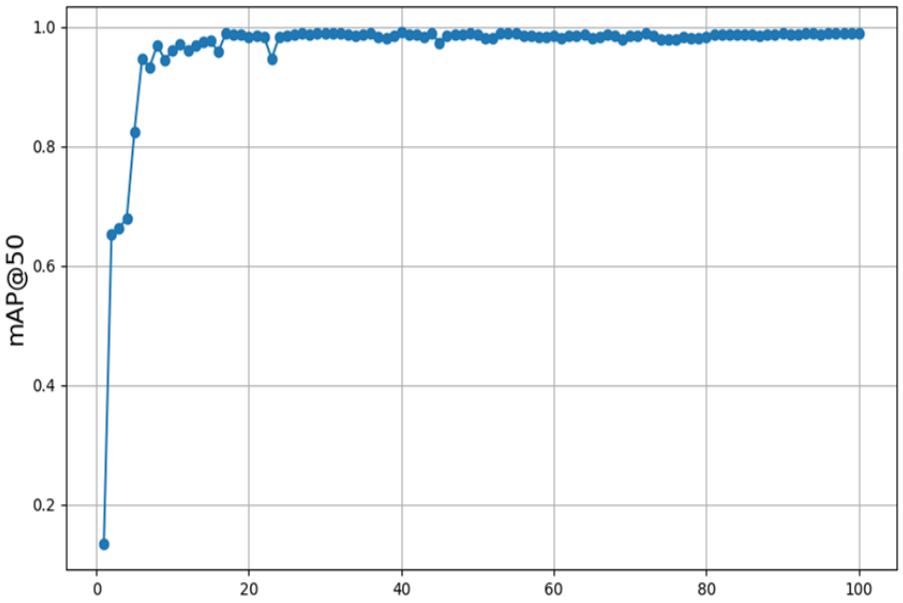
Recall vs Epoch

Results-[8] (Object Detection Metrics)

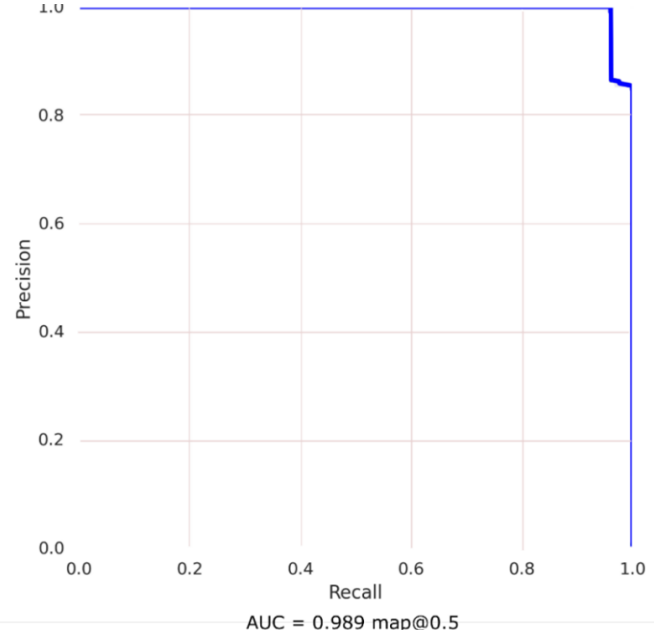
Simulation

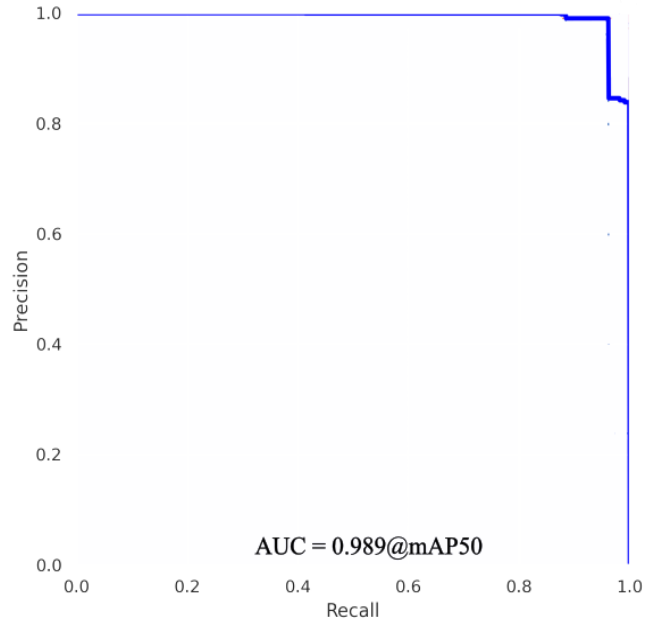
Reality





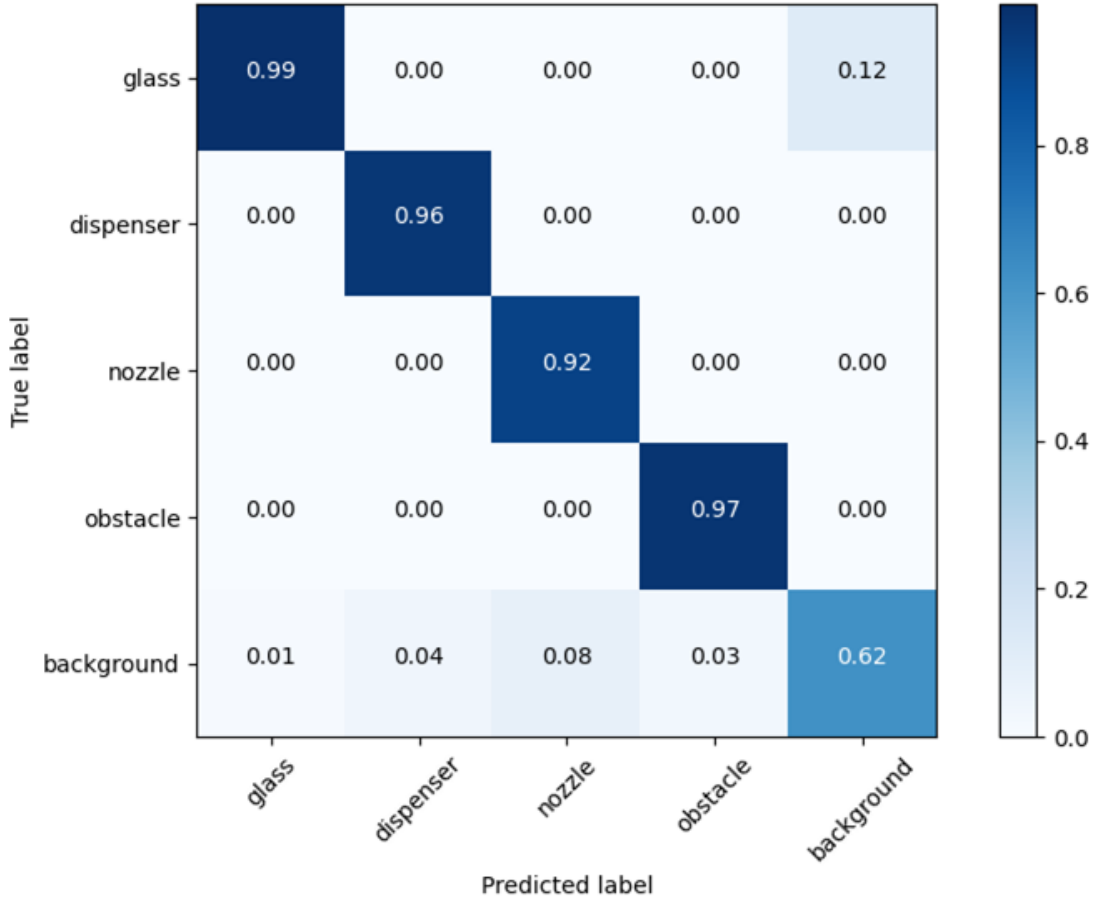
mAP50 vs Epoch



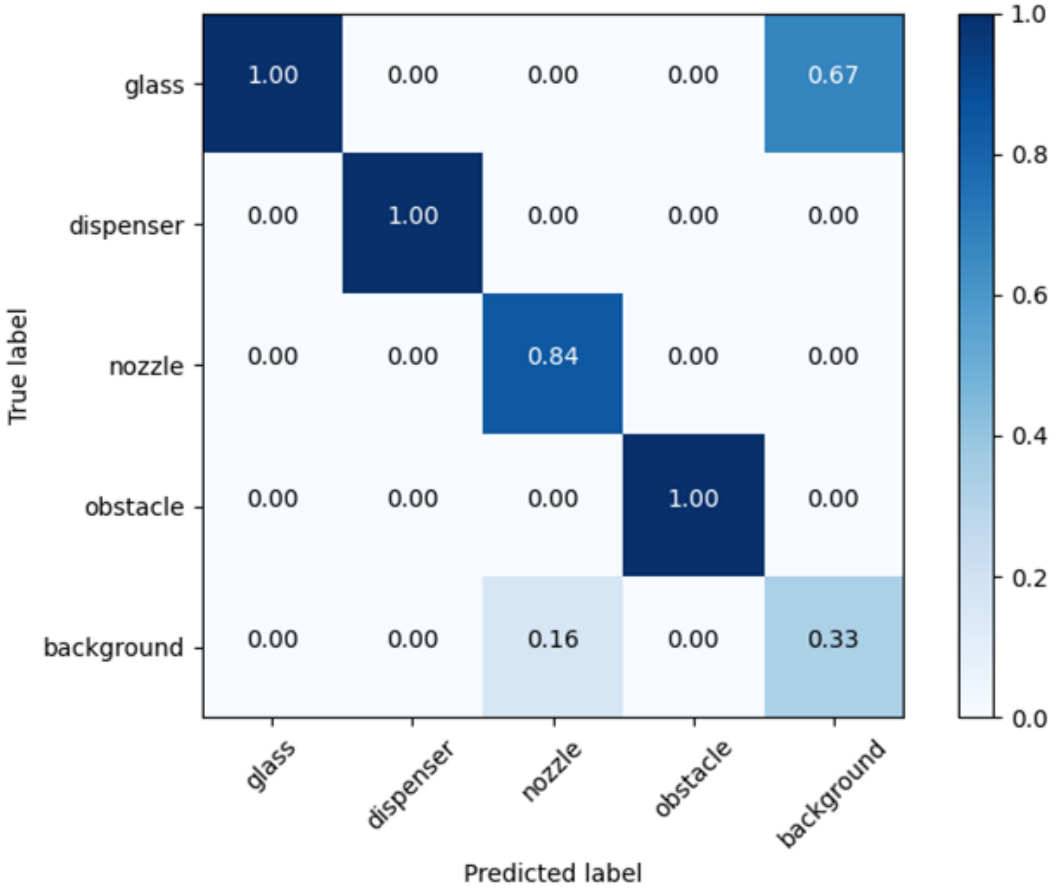


Precision Recall Curve

Results-[9]
(Object Detection Model (Confusion Matrix))

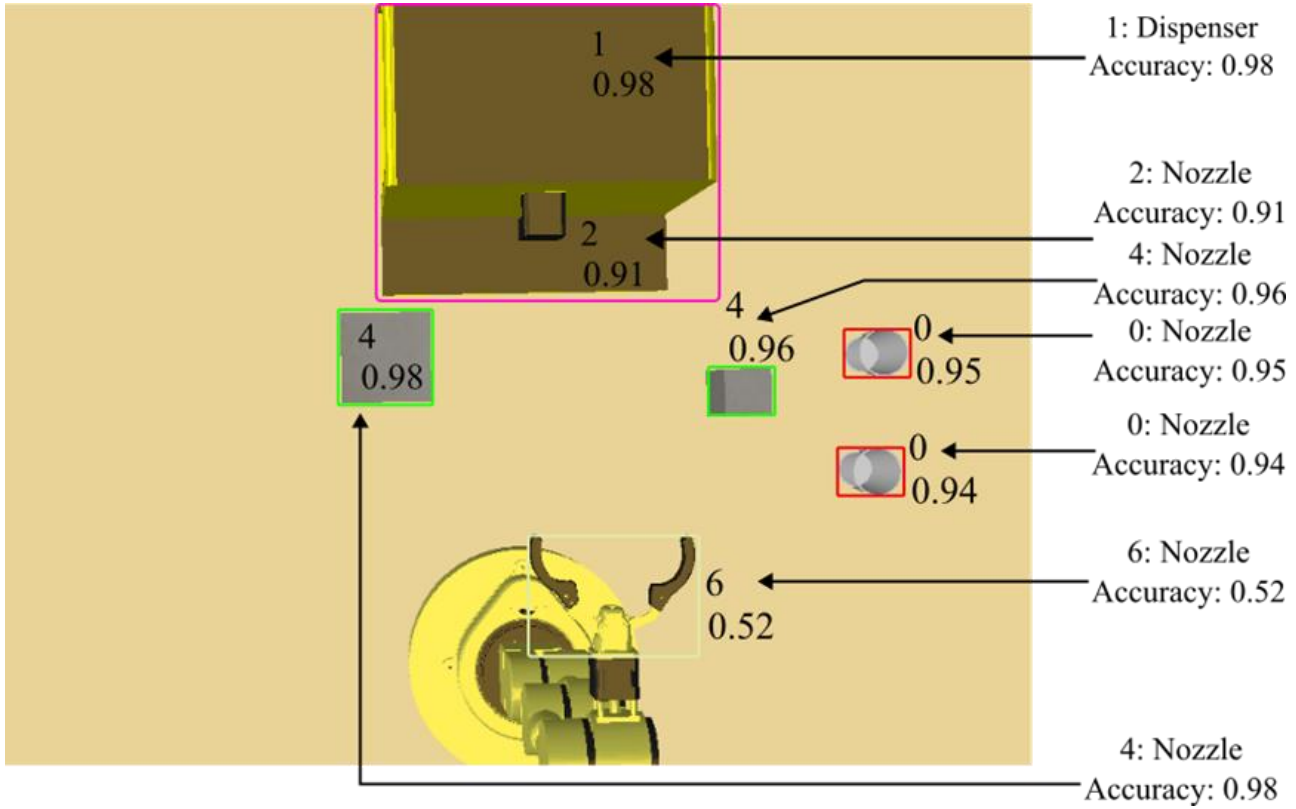


Simulation

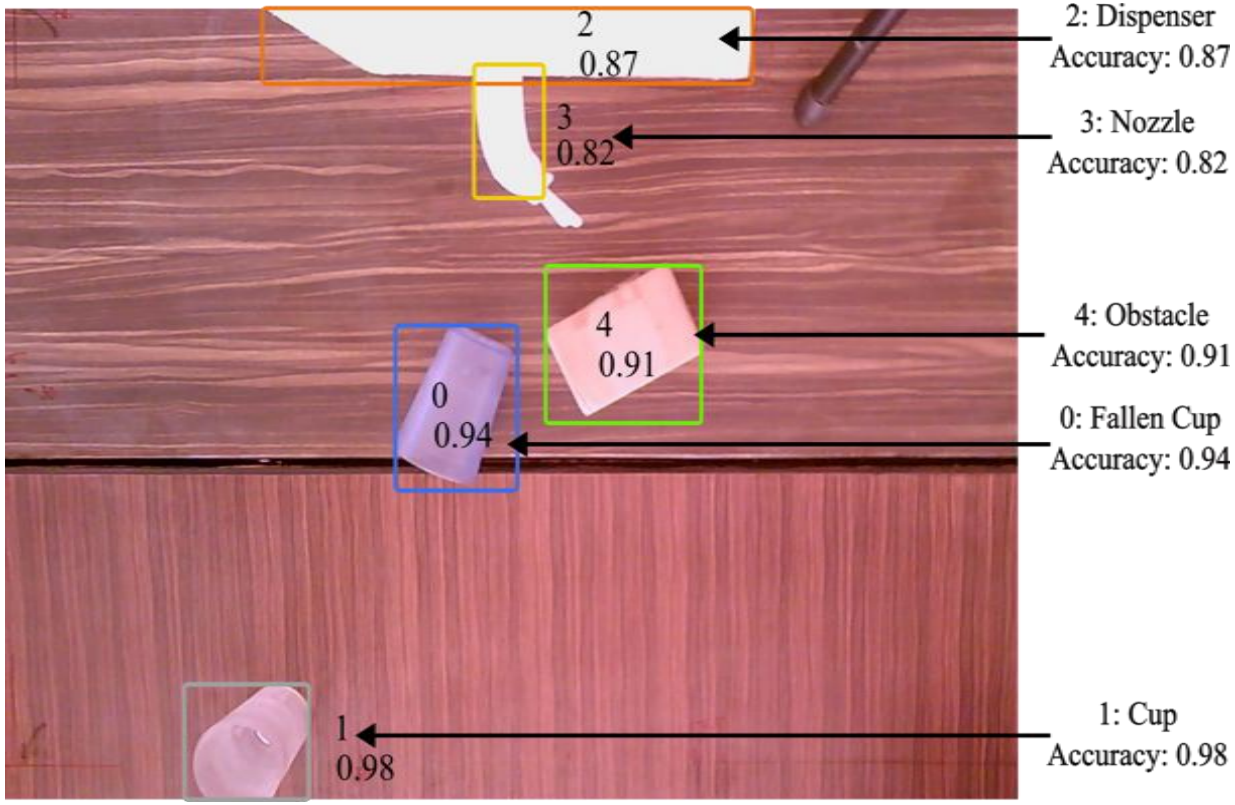


Reality

Results-[10]
(Object Detection Model (Detections))

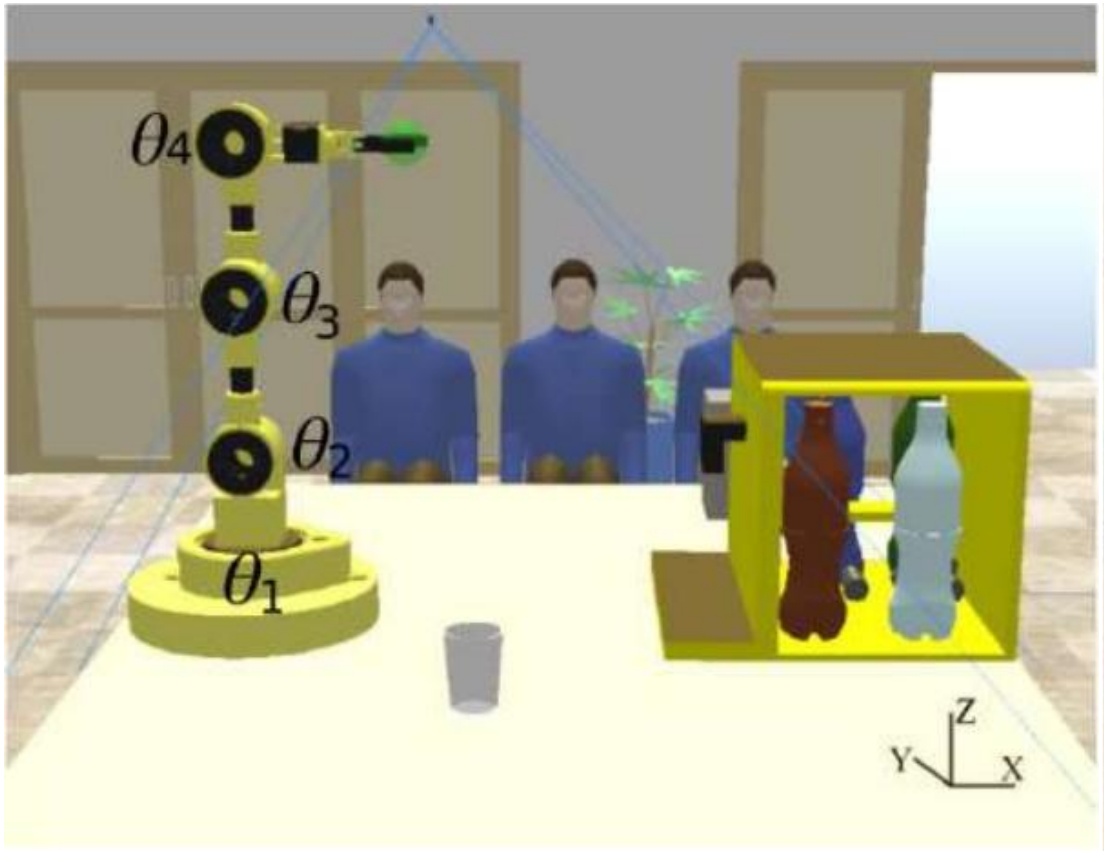


Simulation

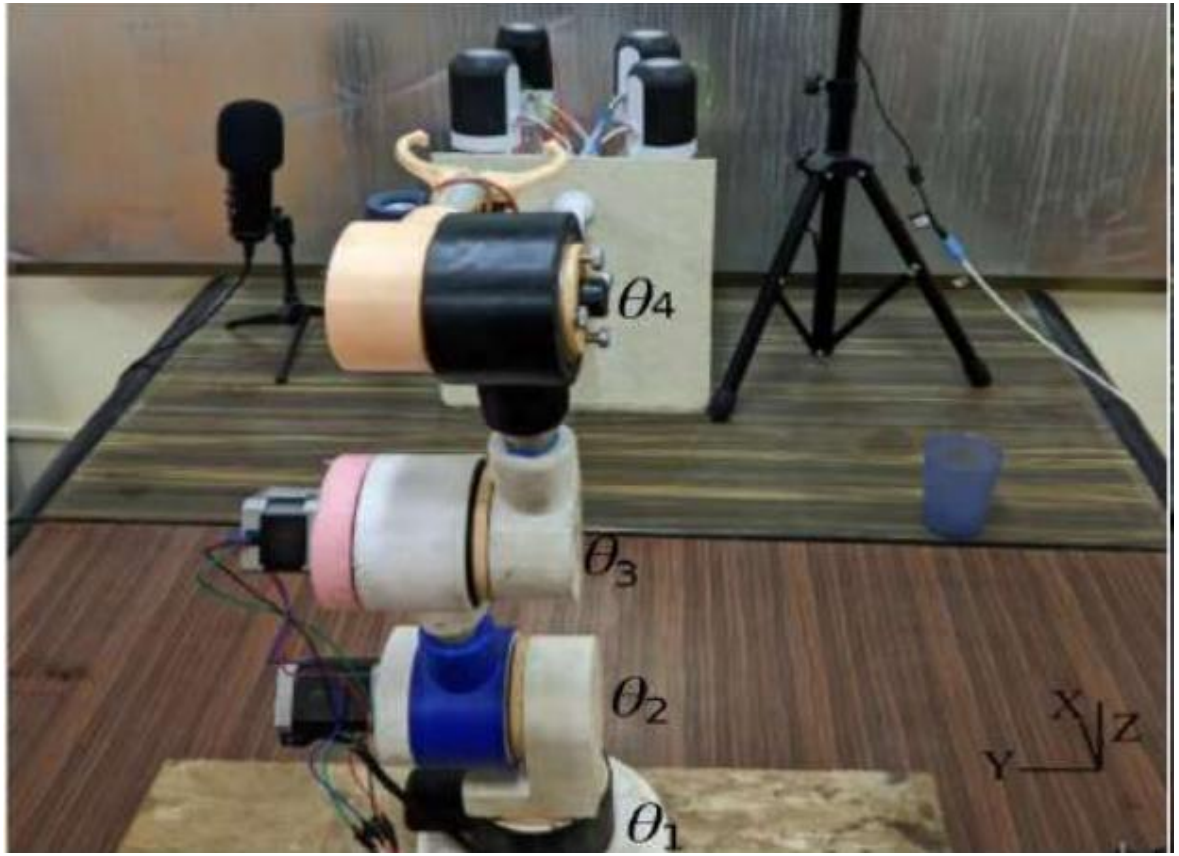


Physical

Results-[11] (Robot Actions)



Simulation

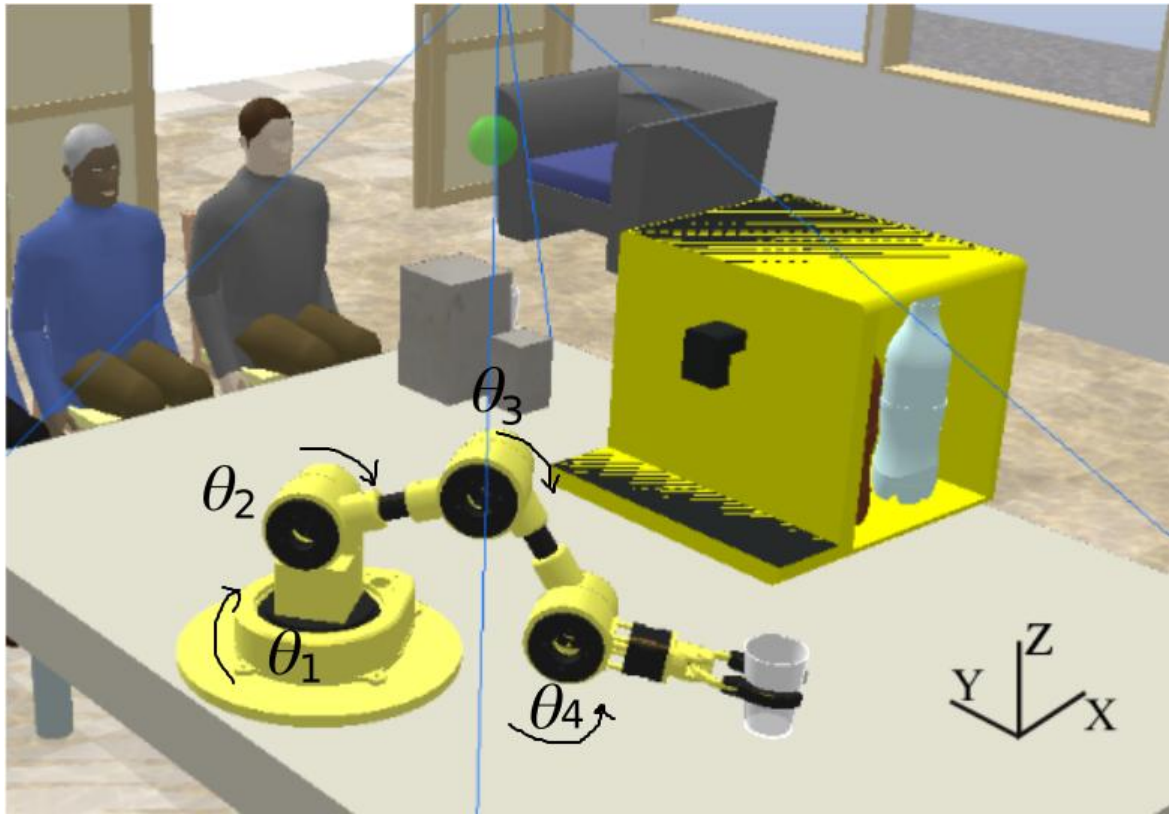


Reality

End Effector Position in Task Space (m)		
X	Y	Z
0.33	0	1.012

End Effector Position in Task Space (m)			Joint Angles(degree)			
X	Y	Z	θ_1	θ_2	θ_3	θ_4
0.20	0	0.57	0	0	0	0

Results-[12] (Robot Actions)



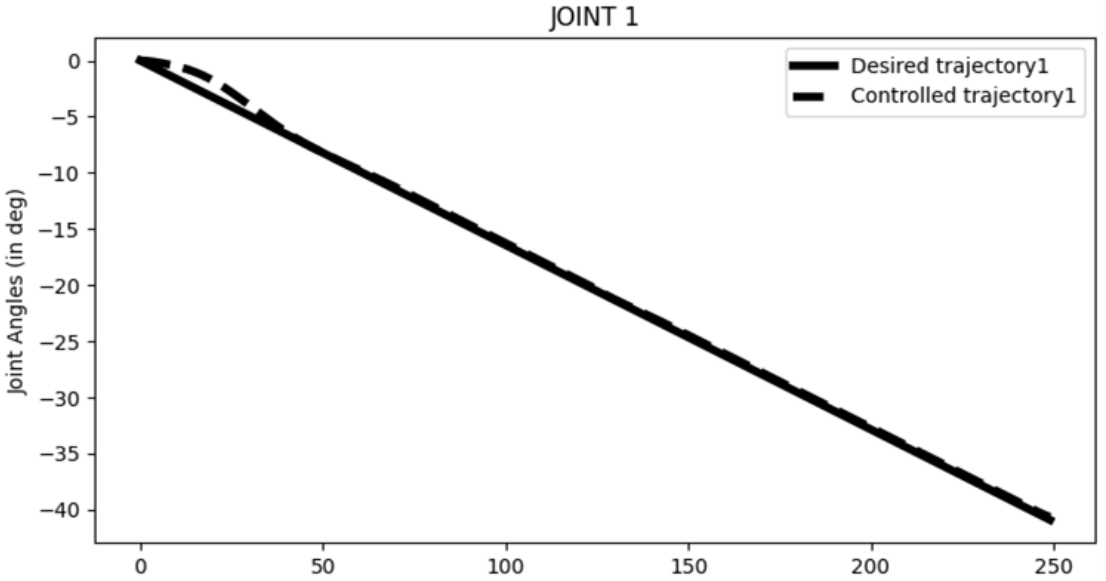
Simulation

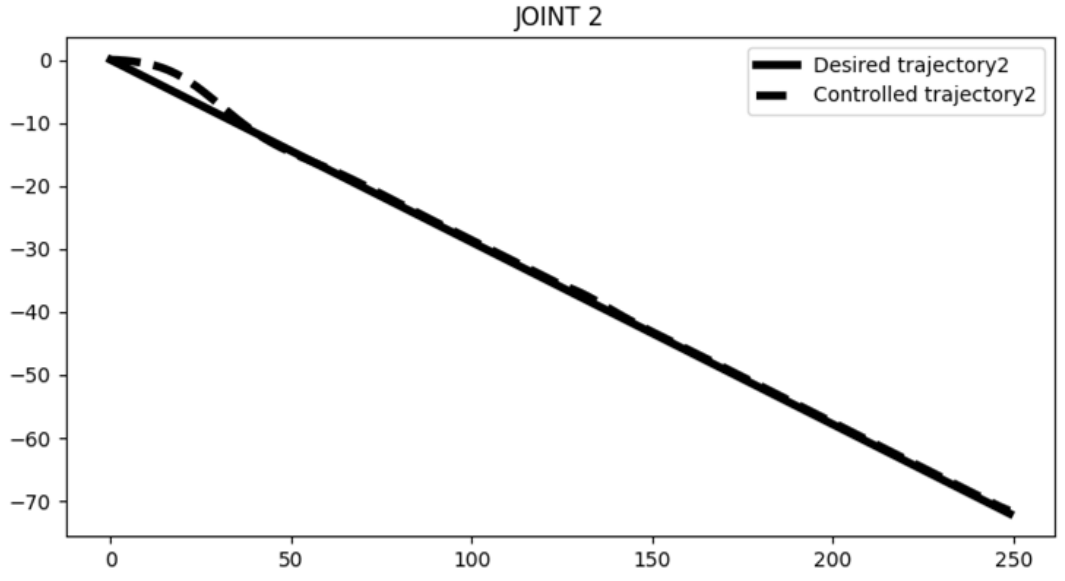
Change in Angles ($\Delta\theta$)			
$\Delta\theta_1$ (CW)	$\Delta\theta_2$ (CW)	$\Delta\theta_3$ (CW)	$\Delta\theta_4$ (CCW)
= -52.5 -0 = -52.5	= -72 - 0 = -72	= -66.0 - 0 = -66.0	= 131.4 – 0 = 131.4

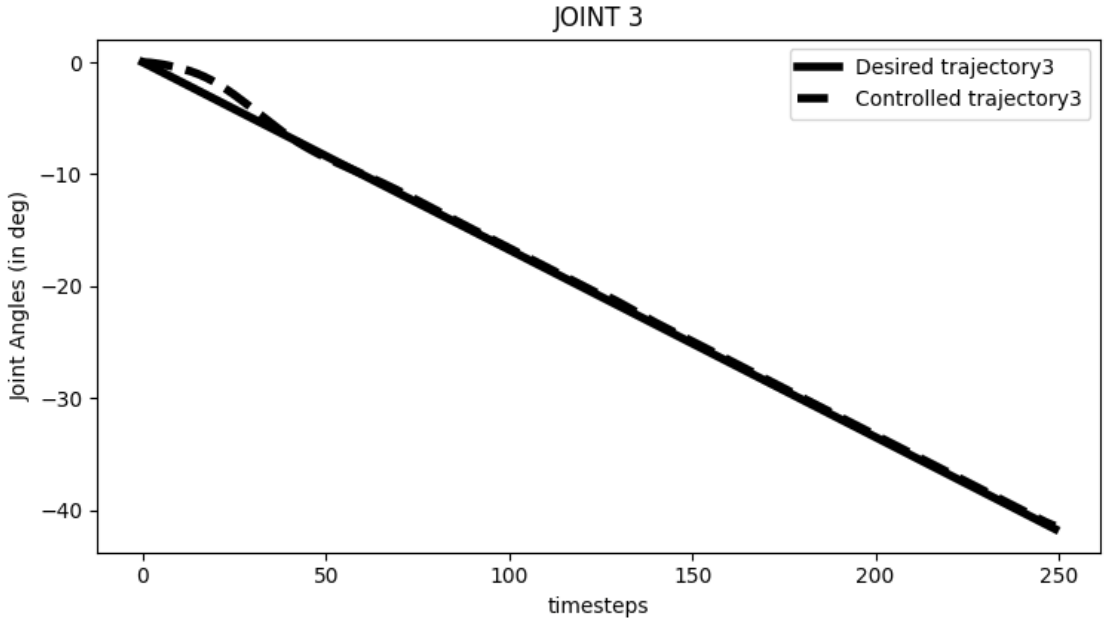
Reality

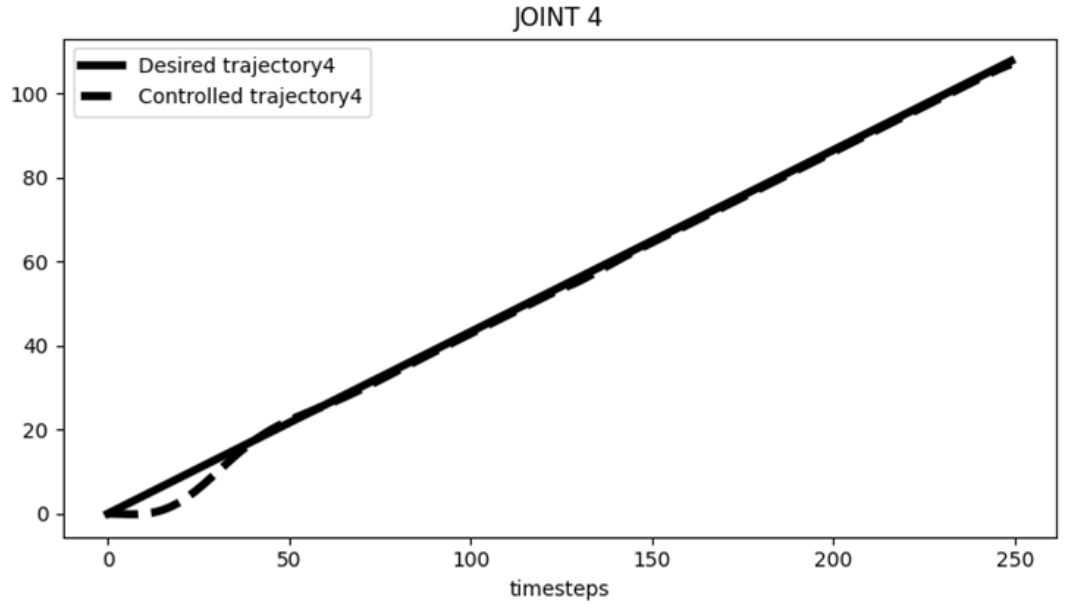
Change in Angles ($\Delta\theta$)			
$\Delta\theta_1$ (CW)	$\Delta\theta_2$ (CW)	$\Delta\theta_3$ (CW)	$\Delta\theta_4$ (CCW)
= -40.9 – 0 = -40.9	= -72 - 0 = -72	= -41.7 - 0 = -41.7	= 107.7 – 0 = 107.7

Results-[13]
(Model Predictive Controller Results)

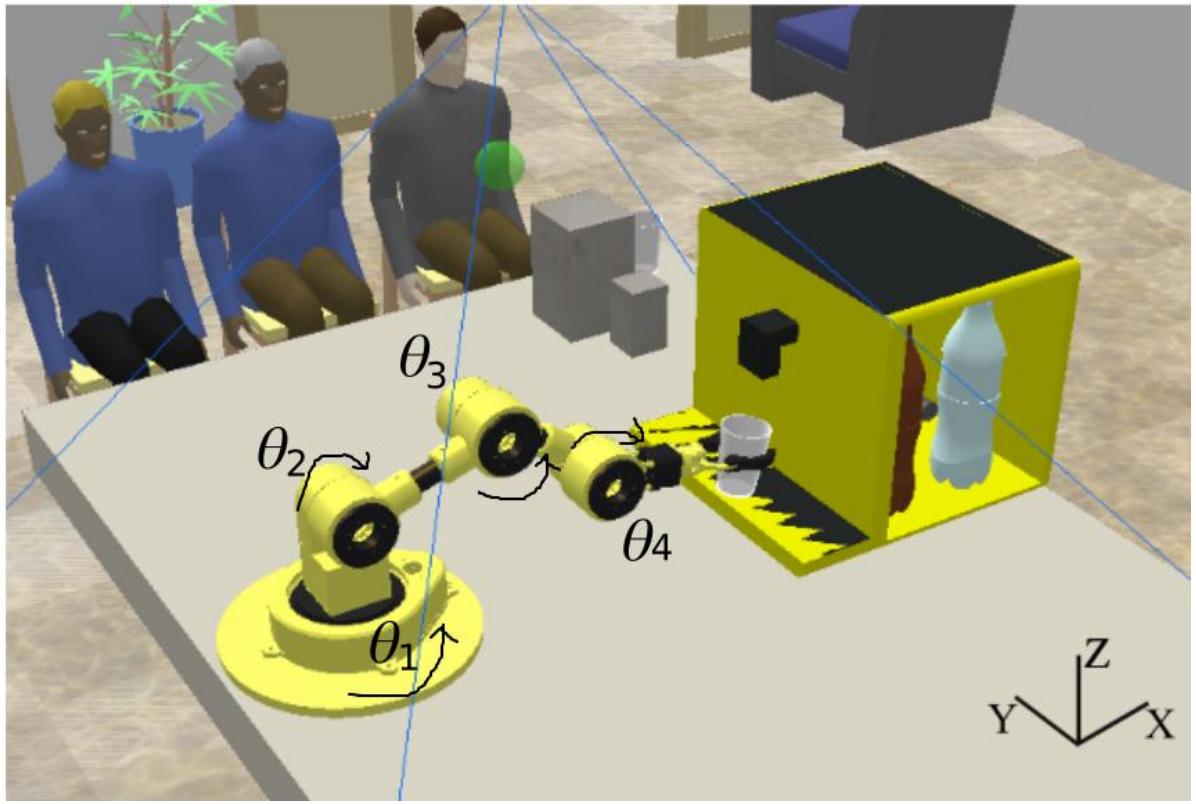




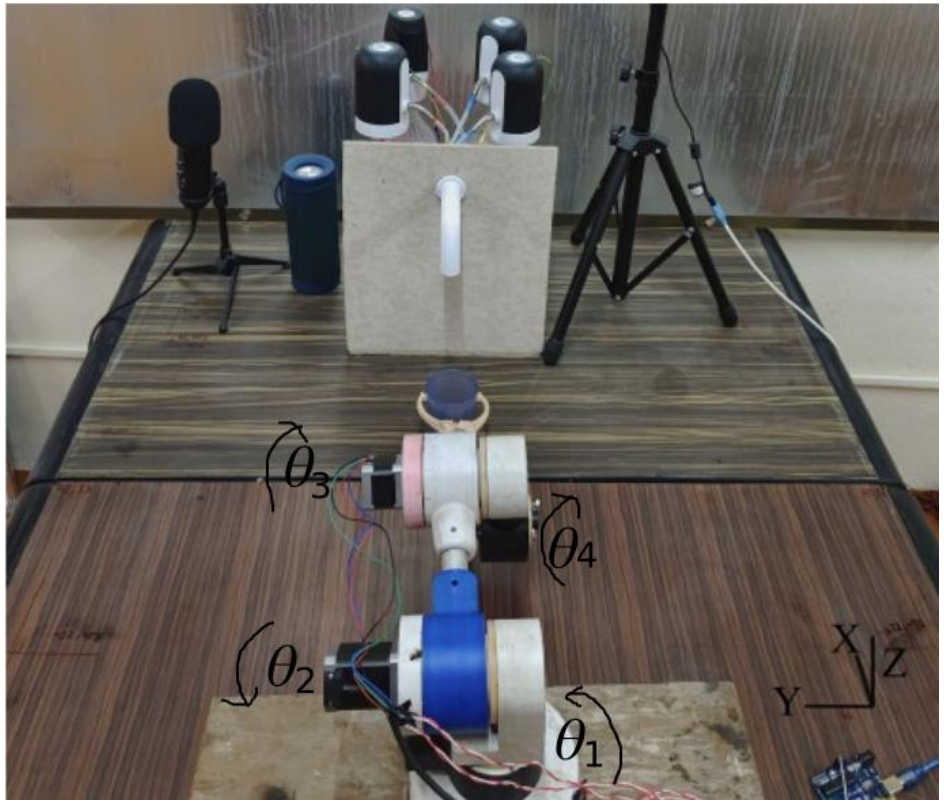




Results-[14] (Robot Actions)



Simulation



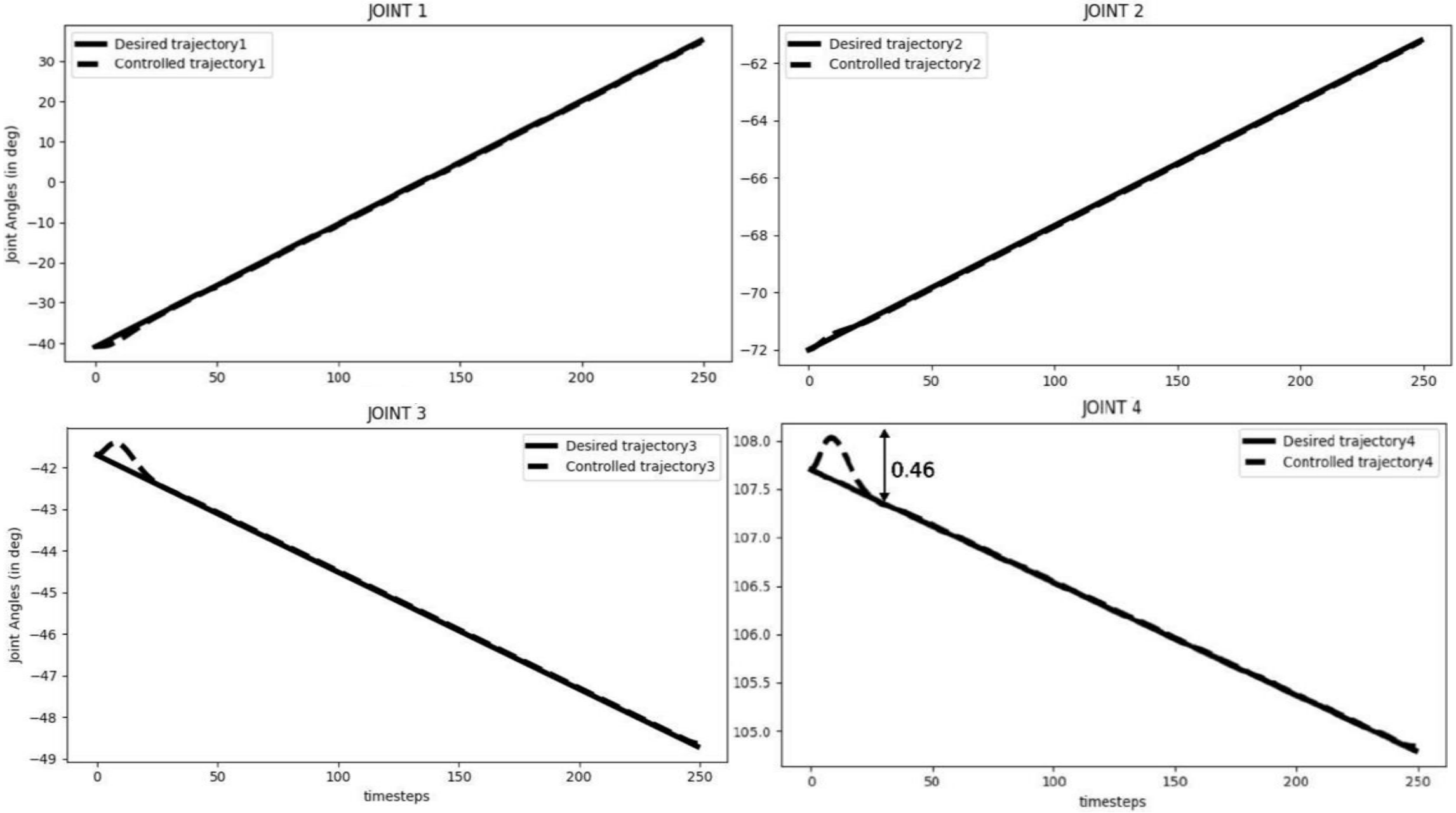
Reality

Change in Angles ($\Delta\theta$)			
$\Delta\theta_1$ (CCW)	$\Delta\theta_2$ (CW)	$\Delta\theta_3$ (CCW)	$\Delta\theta_4$ (CW)
= 0.4 - (-52.5) = 52.9	= -72 – (-72) = 0	= -50.1 - (-66.0) = 15.9	= 115.8 - 131.4 = -15.6

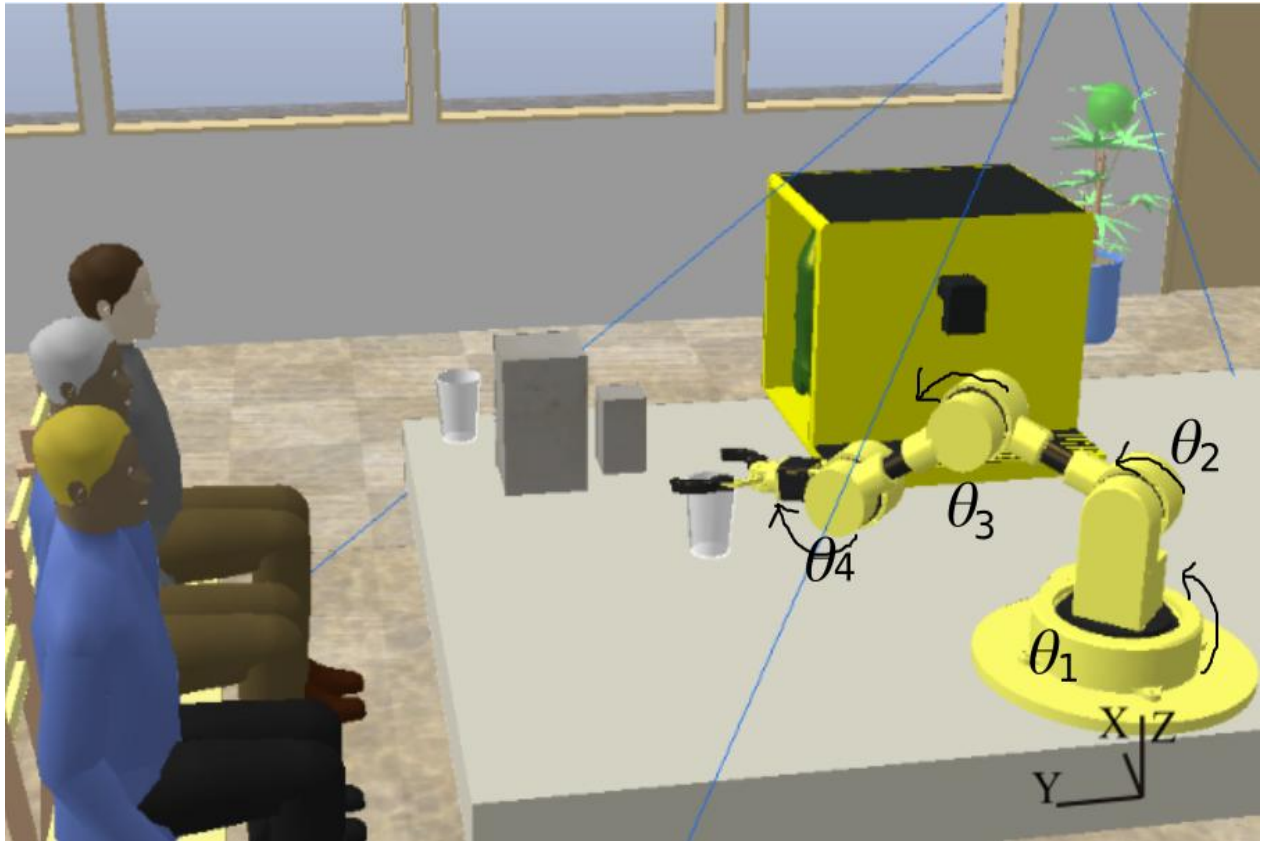
Change in Angles ($\Delta\theta$)			
$\Delta\theta_1$ (CCW)	$\Delta\theta_2$ (CCW)	$\Delta\theta_3$(CW)	$\Delta\theta_4$ (CW)
= 3.6 - (-40.9) = 44.5	= -60.7 - (-72) = 11.3	= -48.5 - (-41.7) = -6.8	= 104.4 - 107.7 = -3.3

Results-[15]

(Model Predictive Controller Results)



Results-[16] (Robot Actions)



Simulation

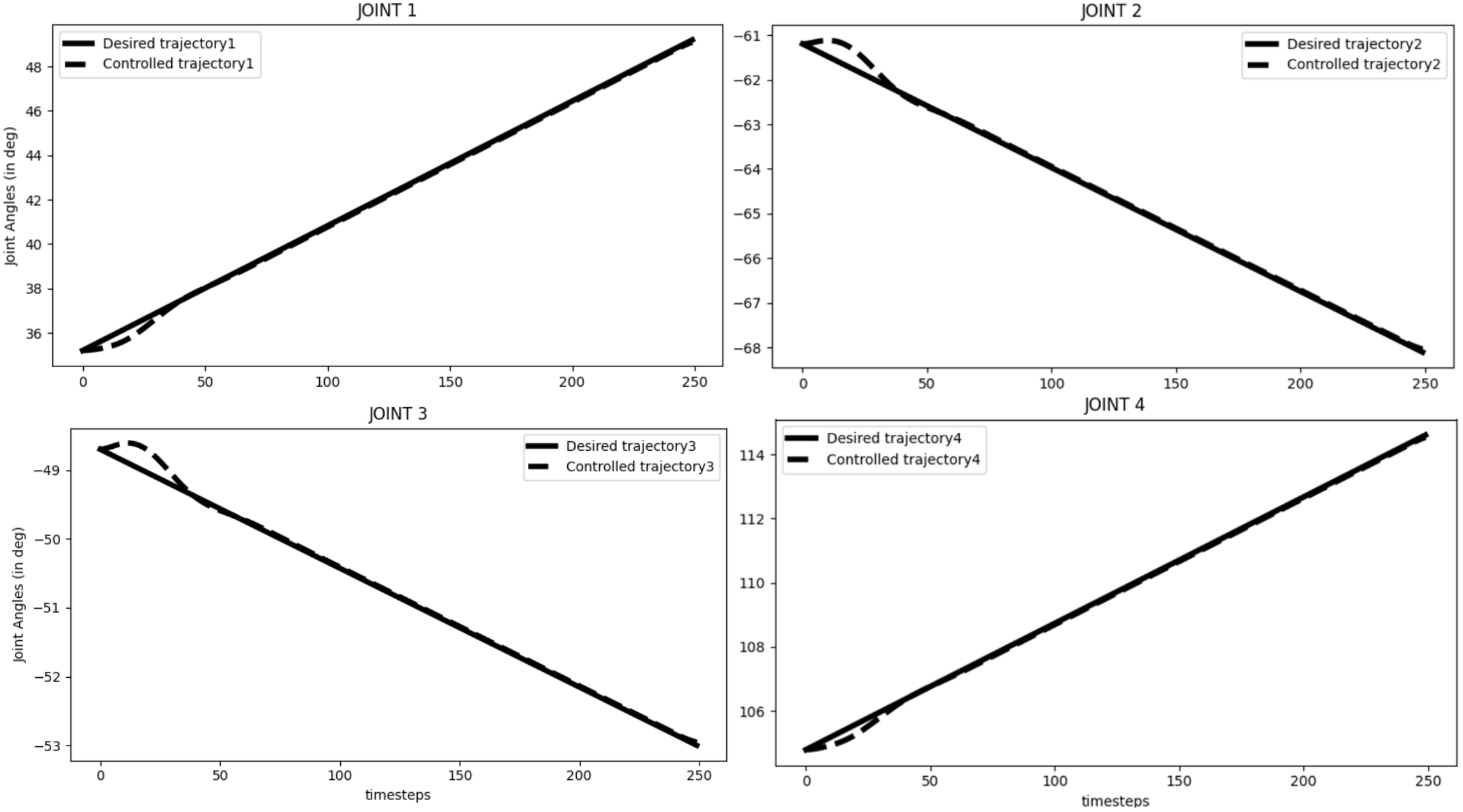
Change in Angles ($\Delta\theta$)			
$\Delta\theta_1$ (CCW)	$\Delta\theta_2$ (CW)	$\Delta\theta_3$ (CW)	$\Delta\theta_4$ (CCW)
= 47.9 - 0.4 = 47.5	= -72 – (-72) = 0	= -54.4 - (-50.1) = -4.3	= 119 - 115.8 = 3.2

Reality

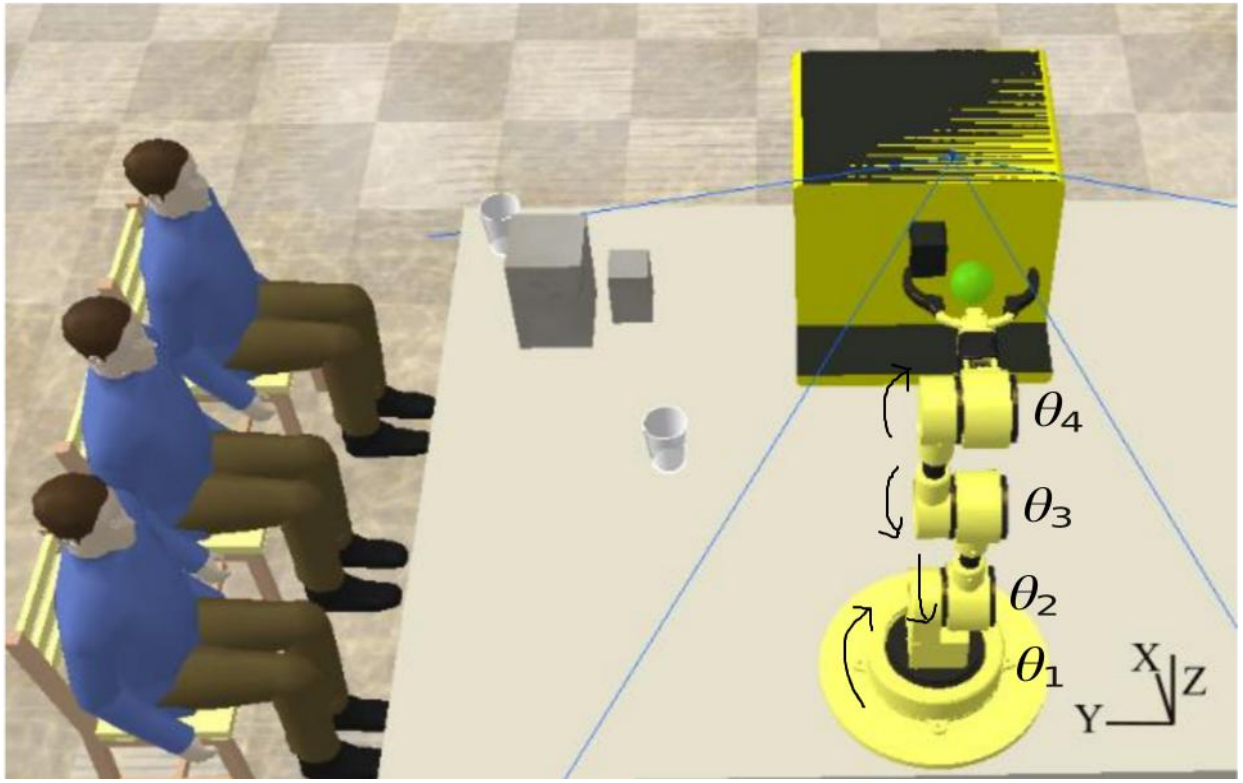
Change in Angles ($\Delta\theta$)			
$\Delta\theta_1$ (CCW)	$\Delta\theta_2$ (CW)	$\Delta\theta_3$ (CW)	$\Delta\theta_4$ (CCW)
= 49.3 - 3.6 = 45.7	= -71 - (-60.7) = -10.3	= -54.7 - (-48.5) = -6.2	= 119.4 - 104.4 = 15

Results-[17]

(Model Predictive Controller Results)



Results-[18] (Robot Actions)



Simulation

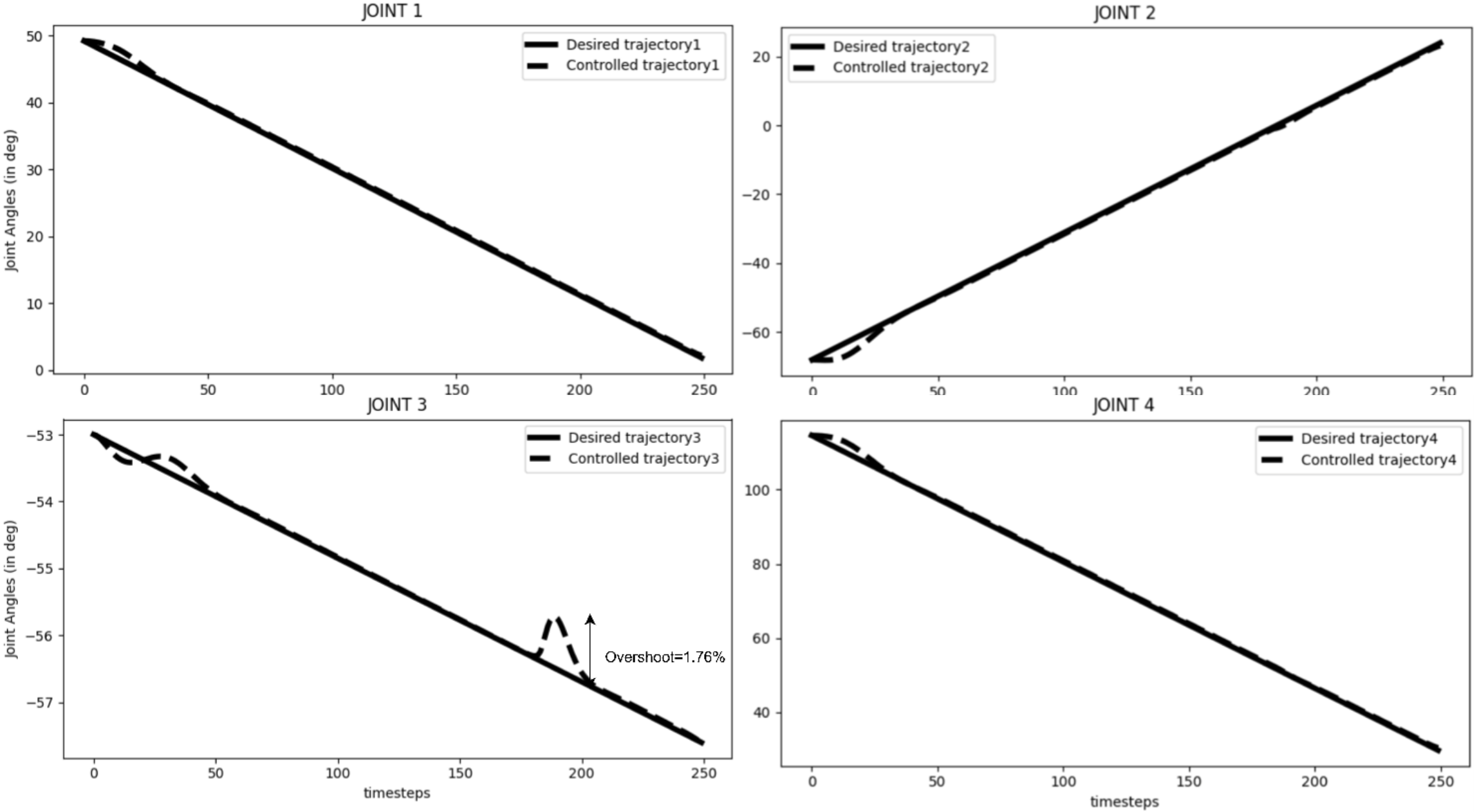
Reality

Change in Angles ($\Delta\theta$)			
$\Delta\theta_1$ (CW)	$\Delta\theta_2$ (CCW)	$\Delta\theta_3$ (CCW)	$\Delta\theta_4$ (CW)
= 0 - 47.9 = -47.9	= 0 - (-72) = 72	= 0 - (-54.4) = 54.4	= 0 - 119 = -119

Change in Angles ($\Delta\theta$)			
$\Delta\theta_1$ (CW)	$\Delta\theta_2$ (CCW)	$\Delta\theta_3$ (CCW)	$\Delta\theta_4$ (CW)
= 0 - 49.3 = -49.3	= 0 - (-71) = 71	= 0 - (-54.7) = 54.7	= 0 - 119.4 = -119.4

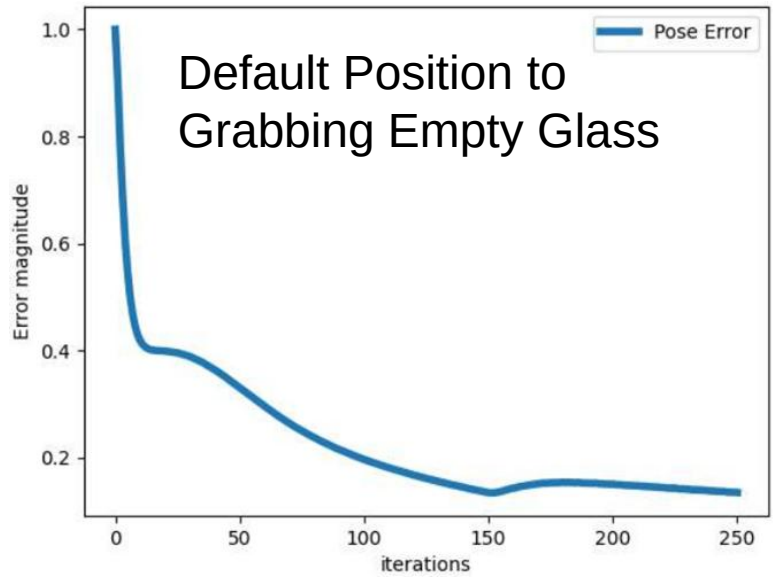
Results-[19]

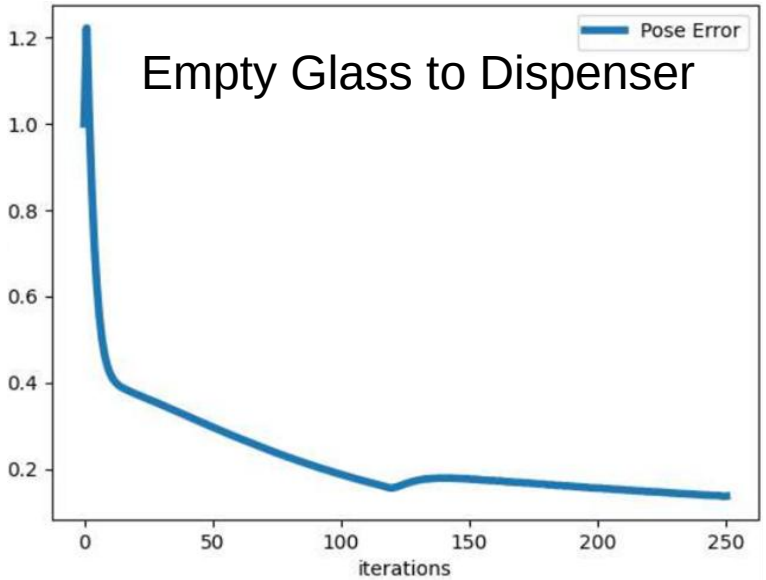
(Model Predictive Controller Results)

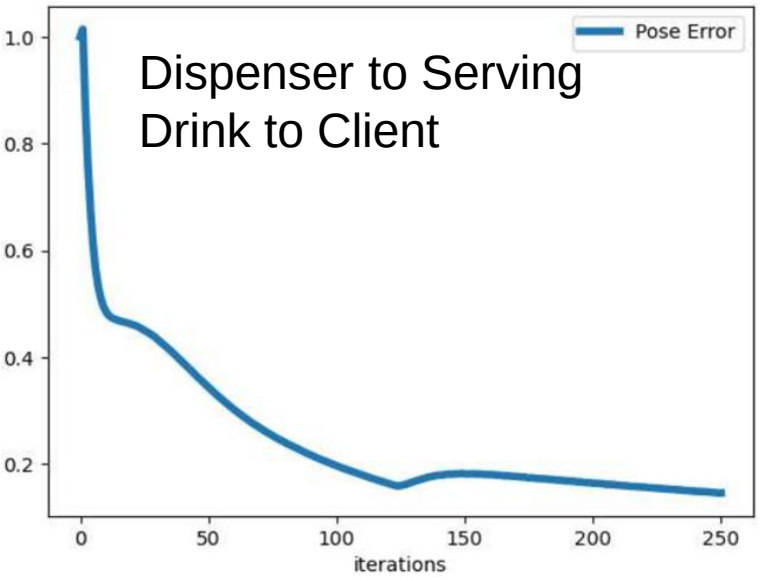


Results-[20]
(Pose Error on All Actions)

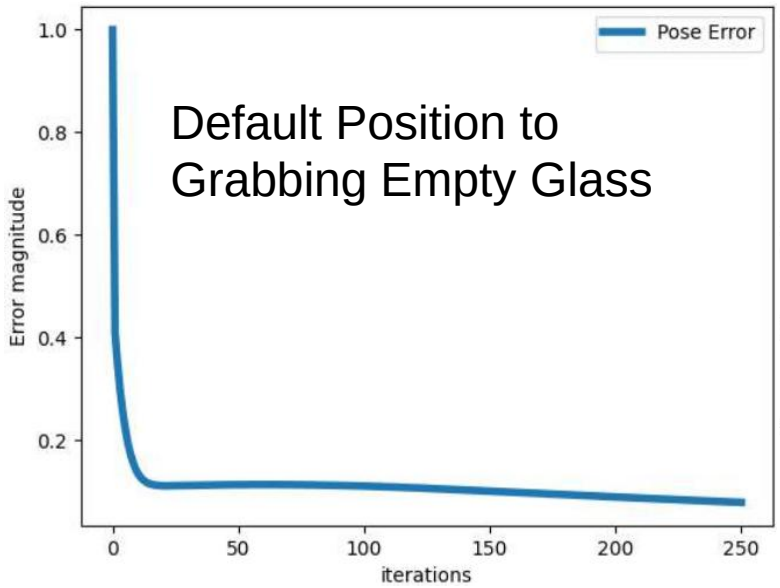
Simulation Plots

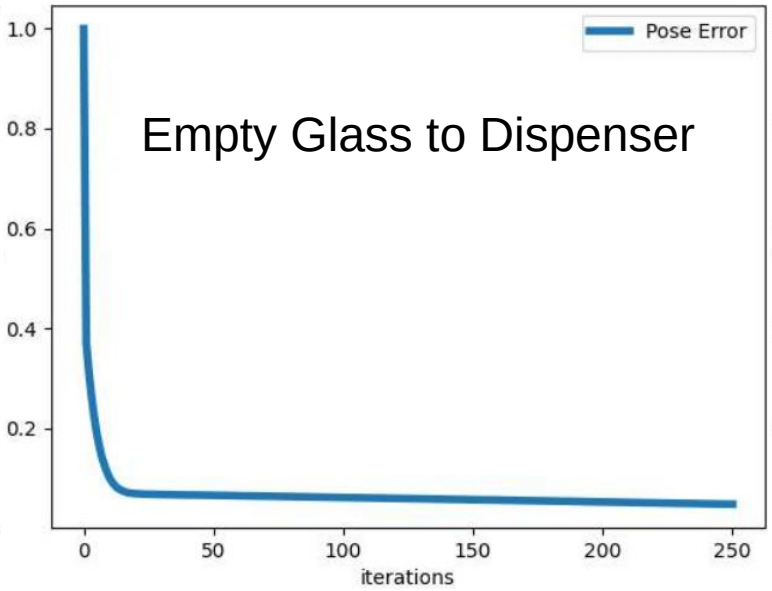


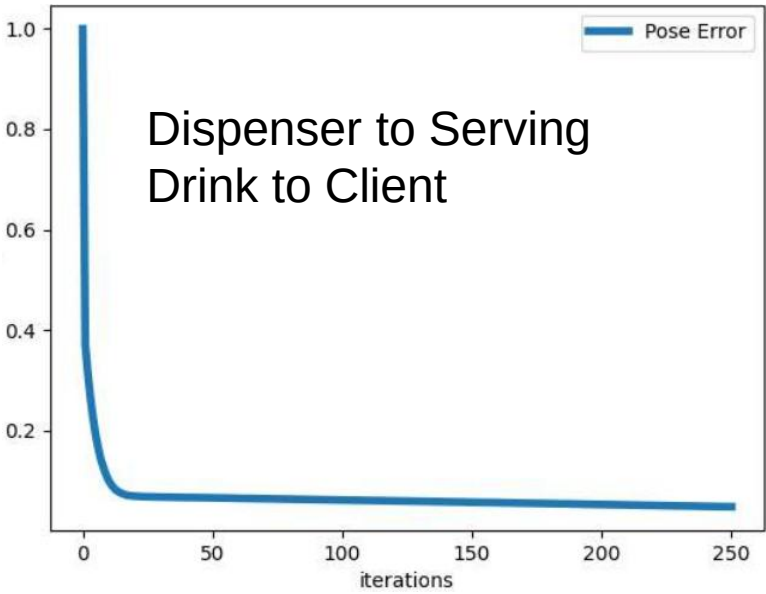




Reality Plots



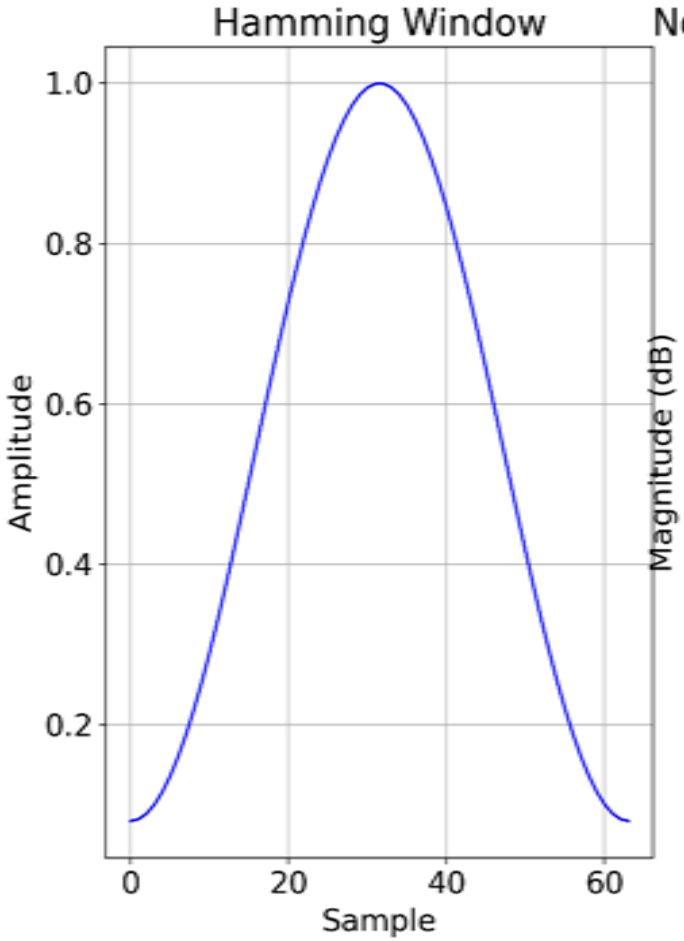


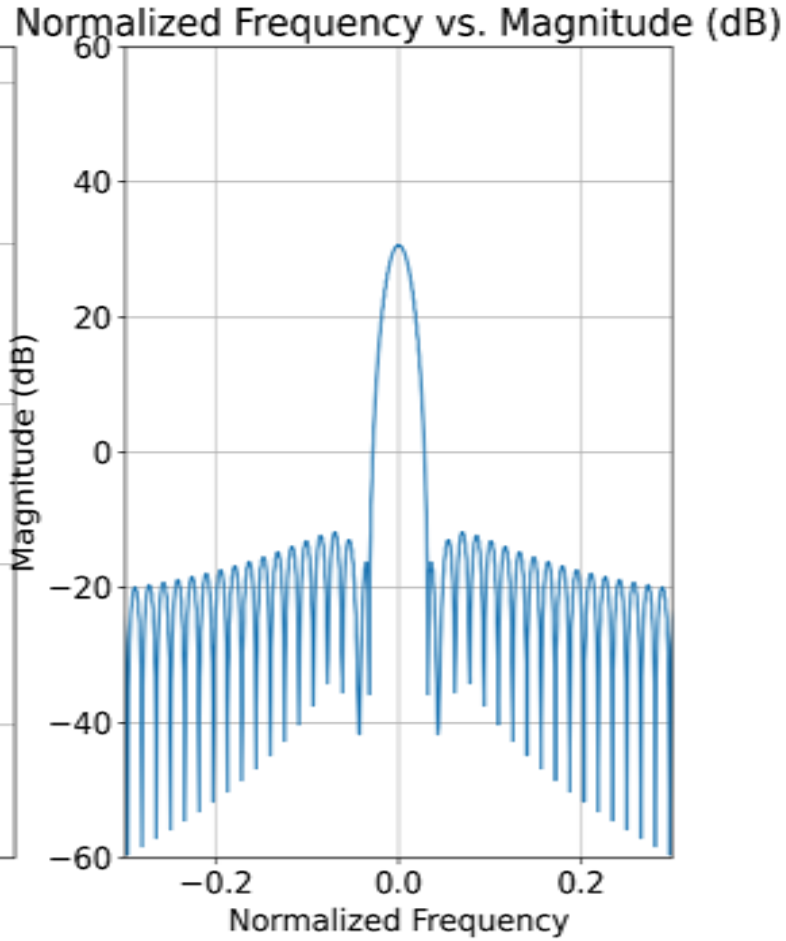


Results-[21]
(Windowing for STFT)

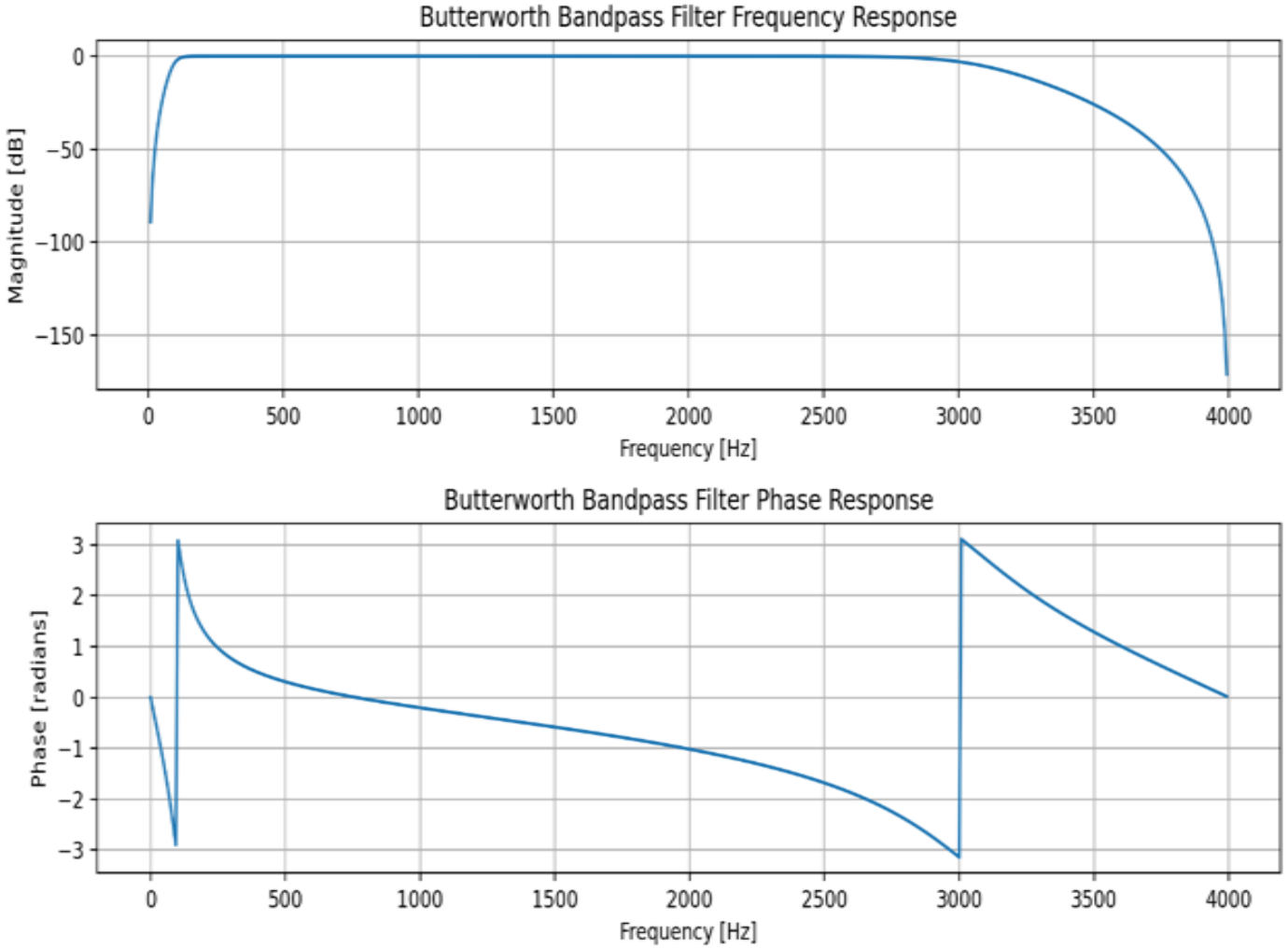
Parameters	Values	No of Samples
Sample rate	16000 Hz	1600
Frame length	20 ms	320
Frame shift	10 ms	160

Parameters	Theoretical	Observed
Peak to peak side lobe amplitude(dB, Relative)	-41	-40.32
Width of main lobe (Normalized Frequency)	8π/M	0.049
Roll off factor(dB/octave)	-18	-24.55



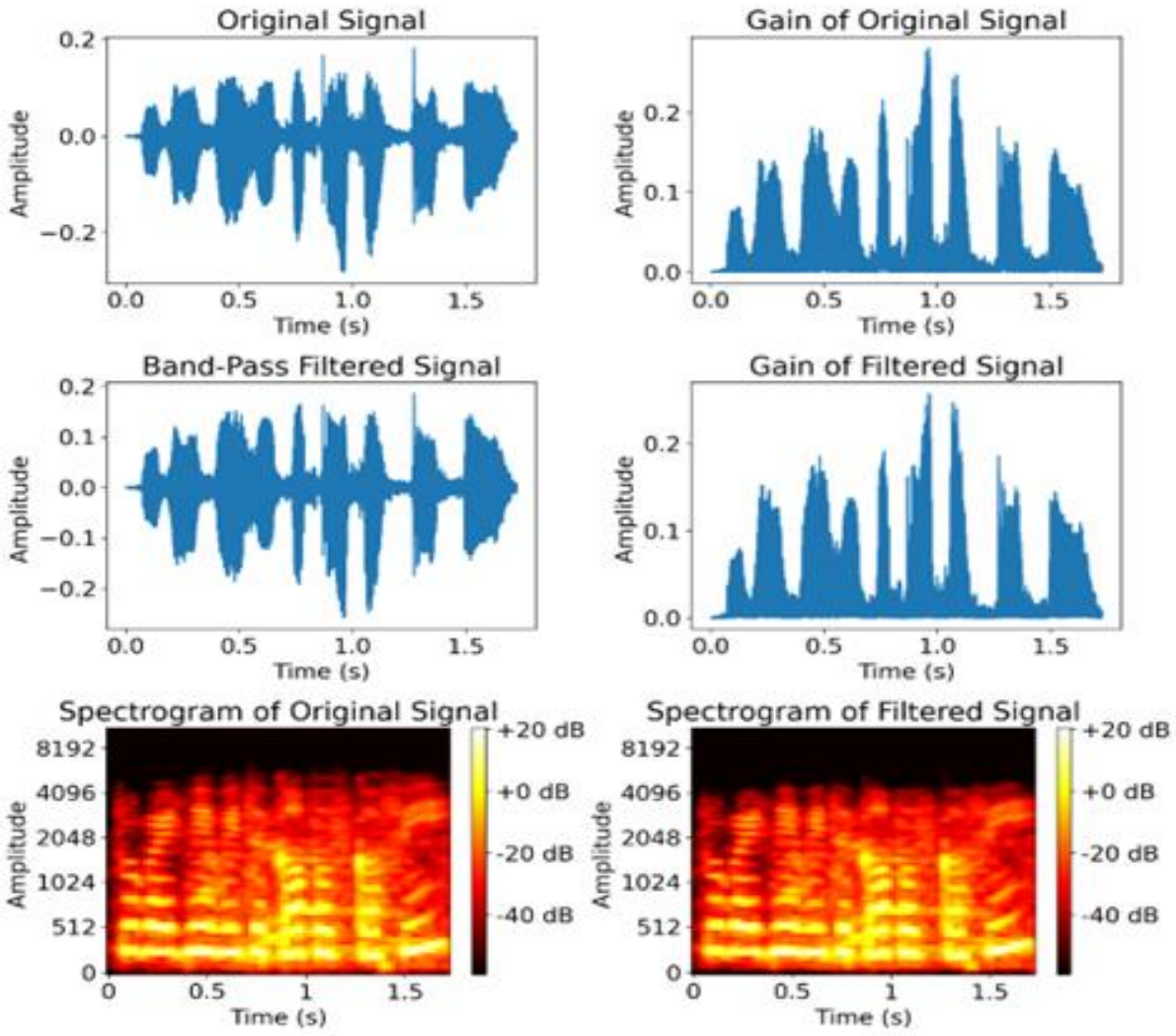


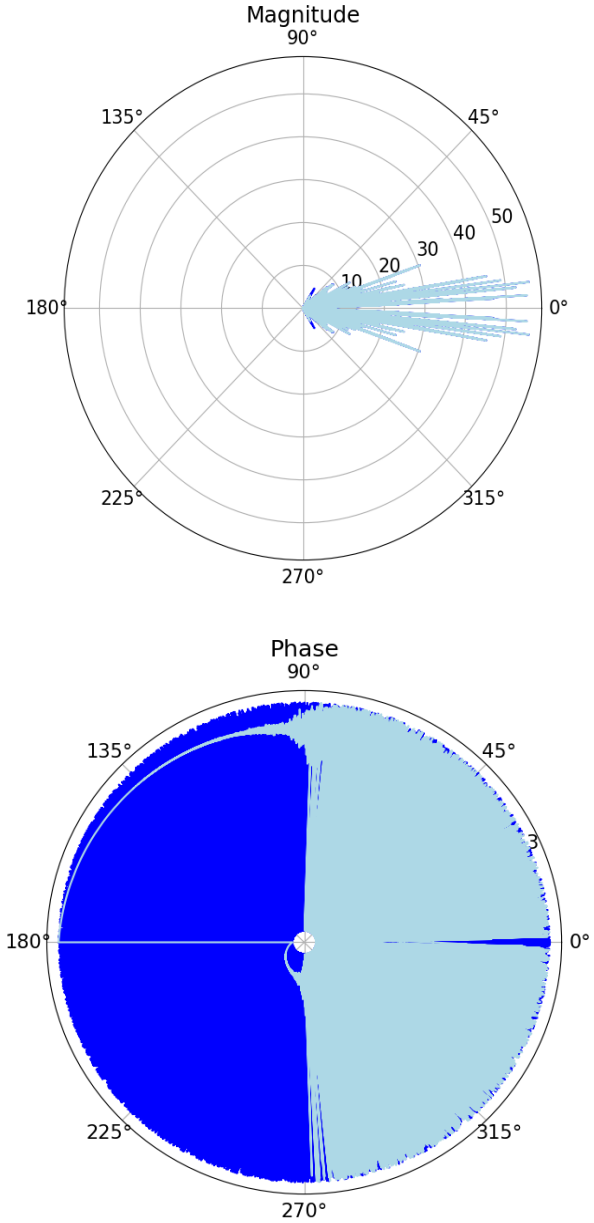
Results-[22]
(Background Noise Removal Using Band Pass Filter)



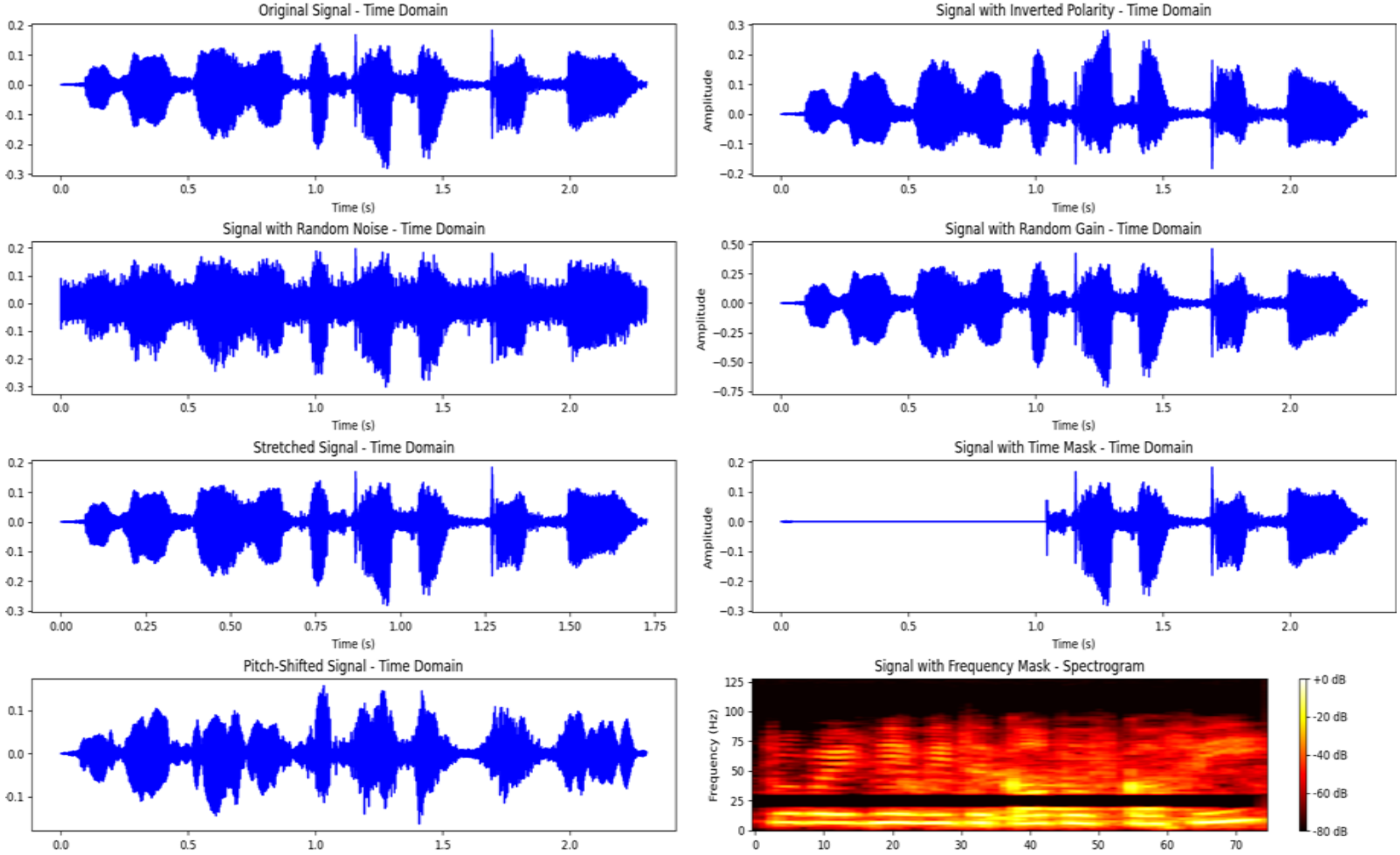
A 4th-order Butterworth filter was applied with a bandpass frequency range of 100 Hz to 3000 Hz. The 3 dB cutoff frequency was determined to be 98 Hz to 2999.0 Hz.

Results-[23] (Noise Removal Using BPF)



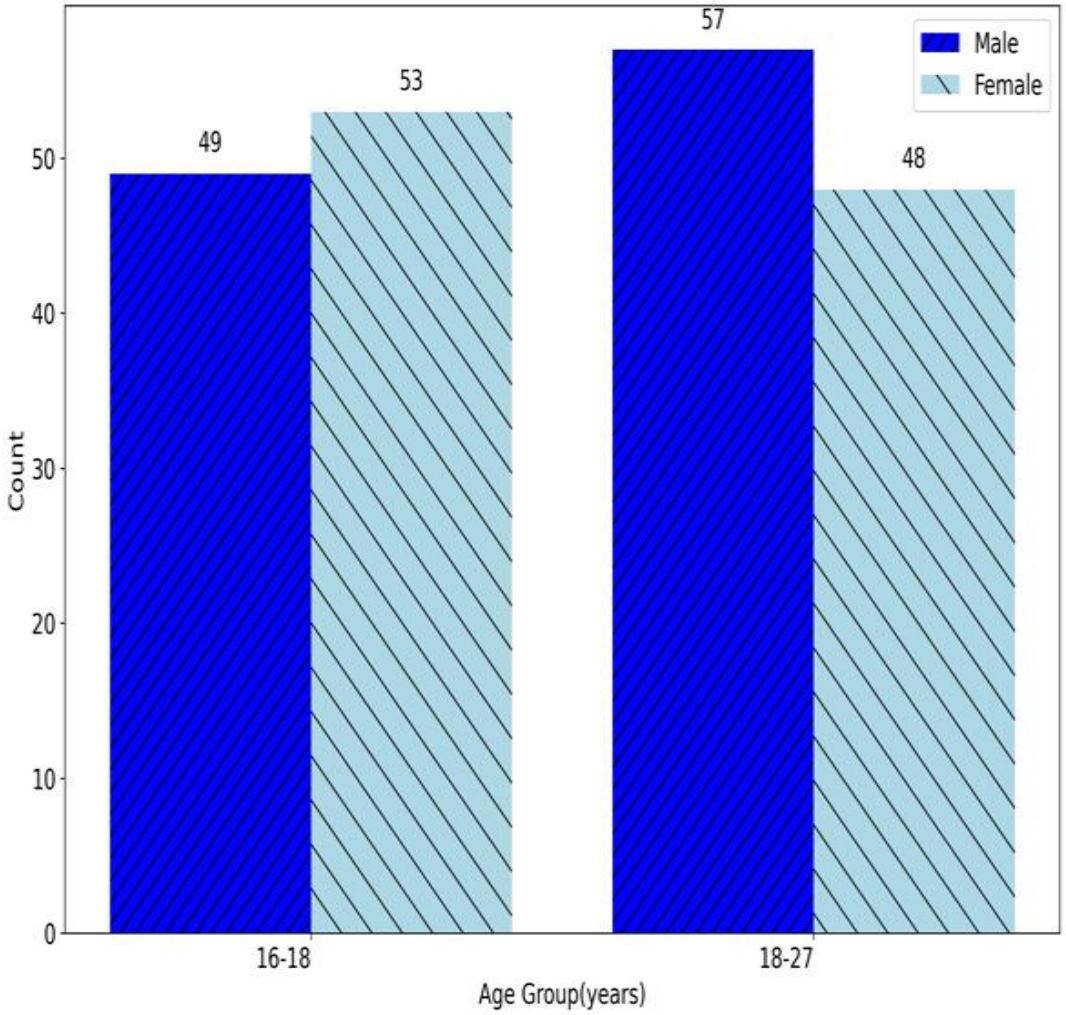


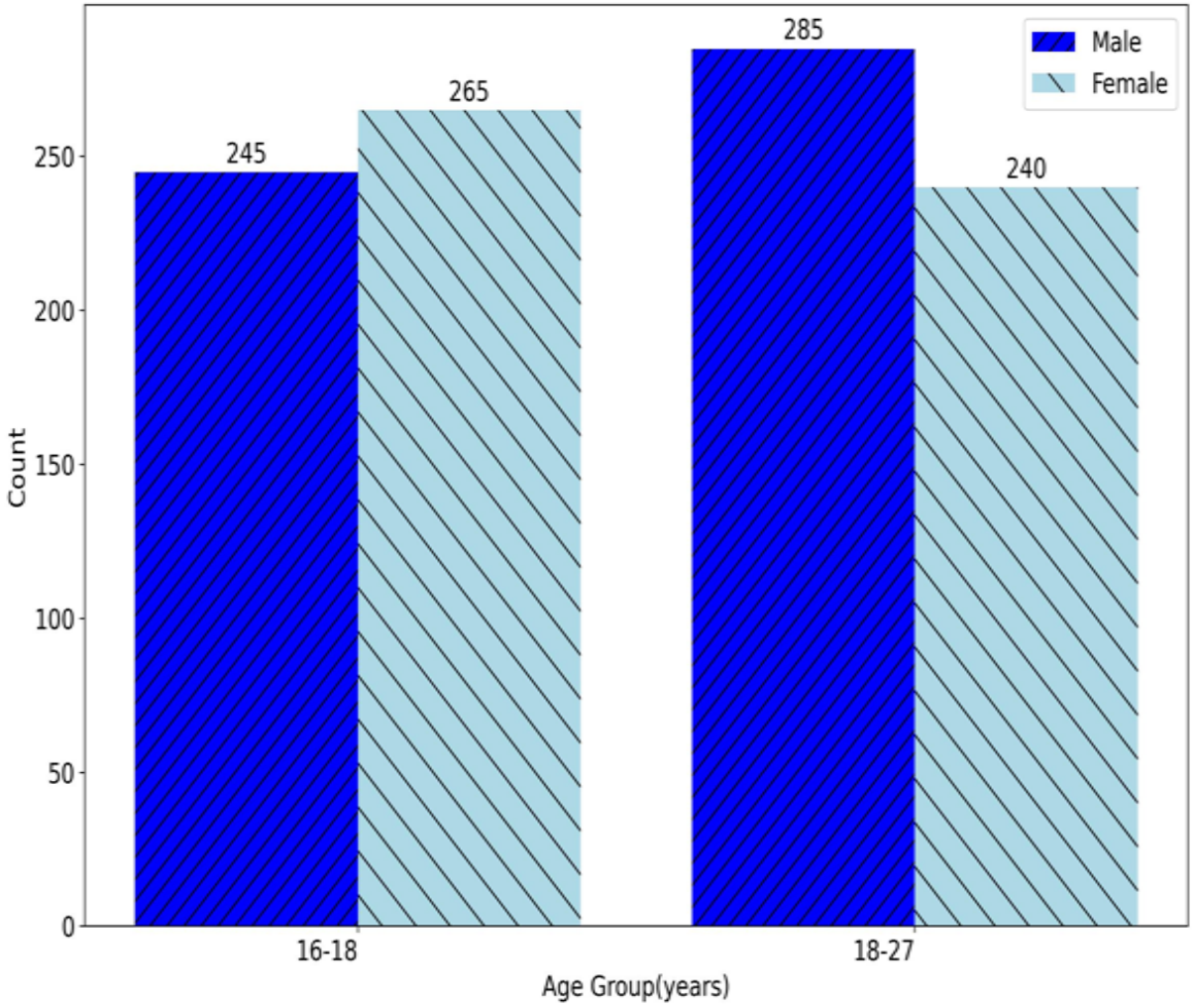
Results-[24] (Female Voice Augmentations)



Plots of female voice for “**Provide me with a cup of coffee**”

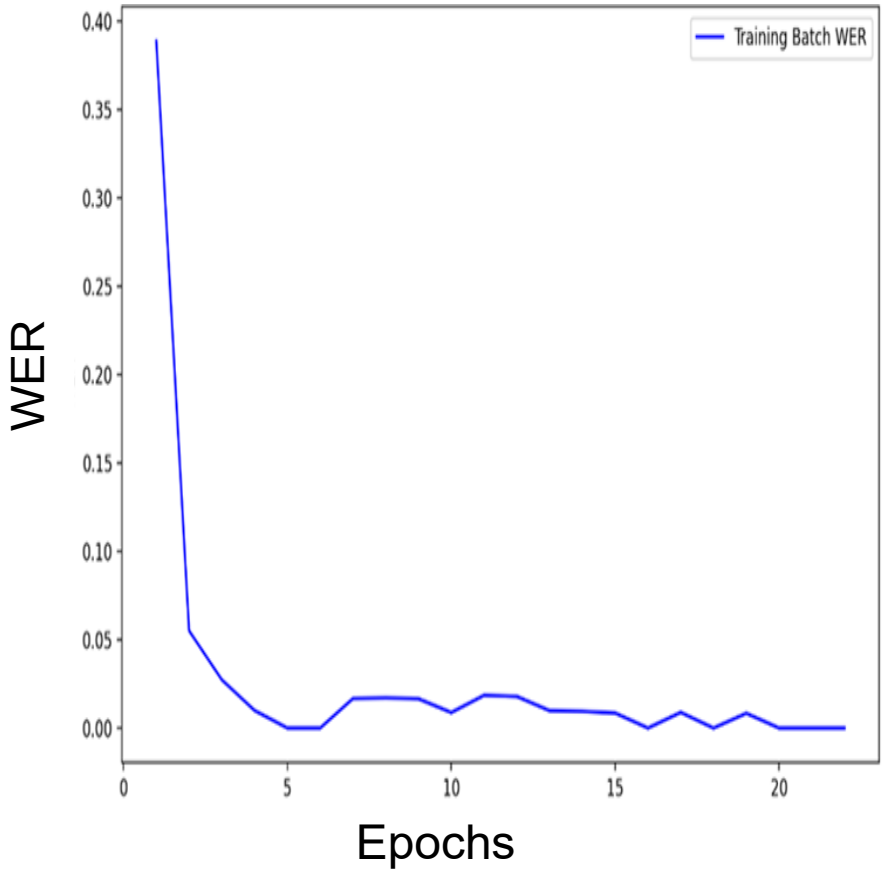
Results-[25]
(Audio Sample Counts Before and After Augmentation)



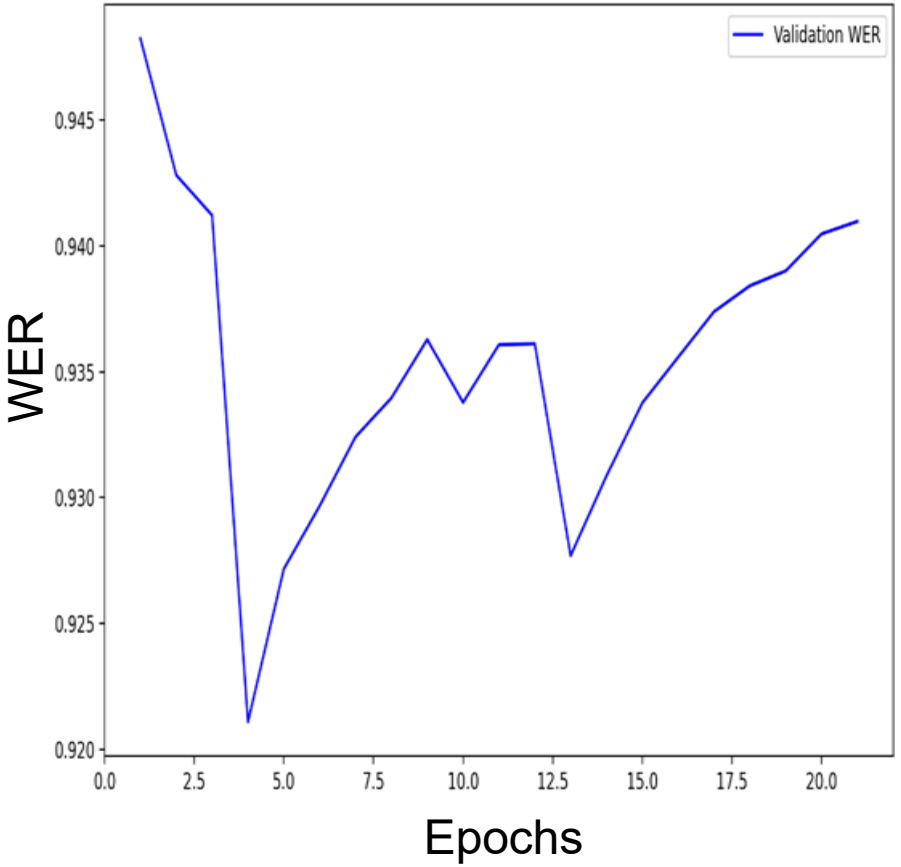


Results-[26]
(Training and Validation WER and CER Curves for Speech Recognition Model)

Training batch WER



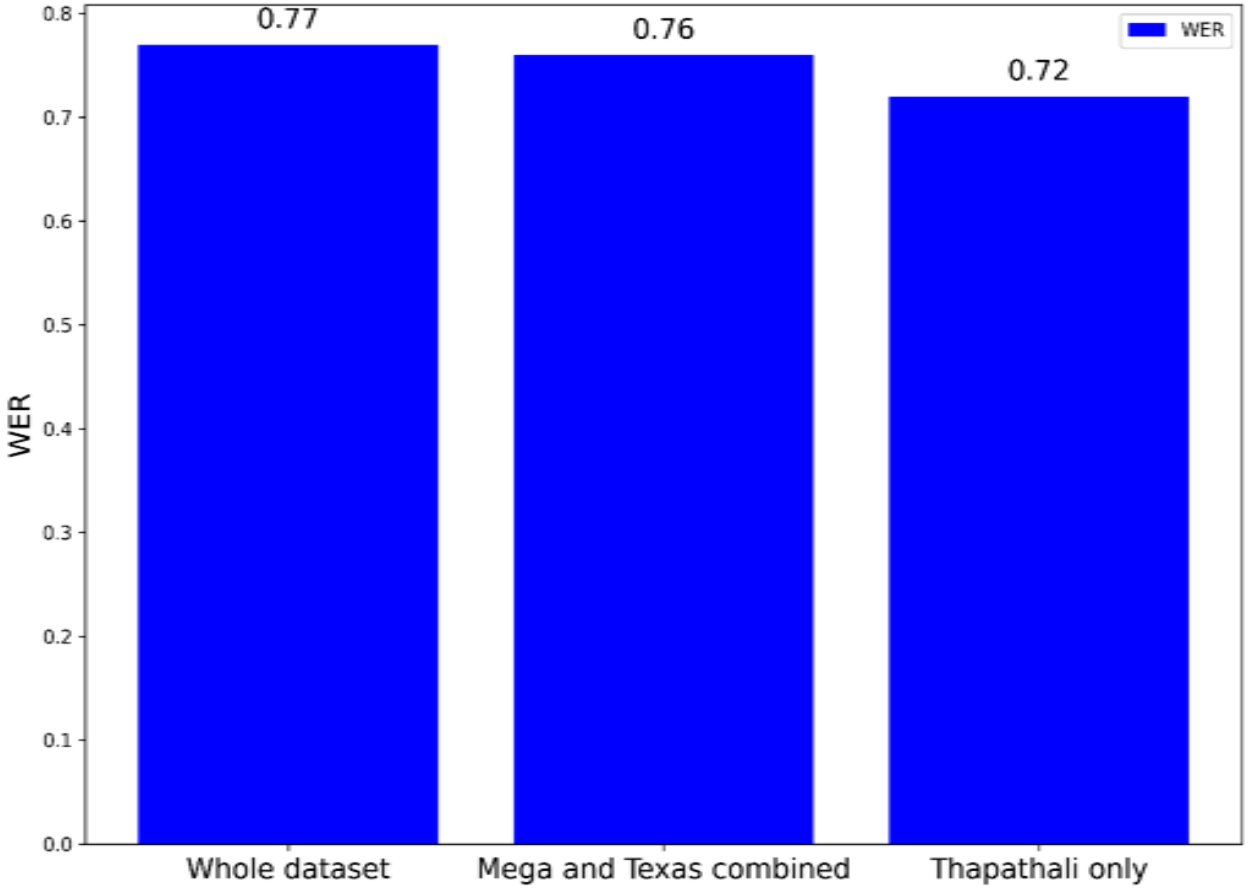
Validation WER

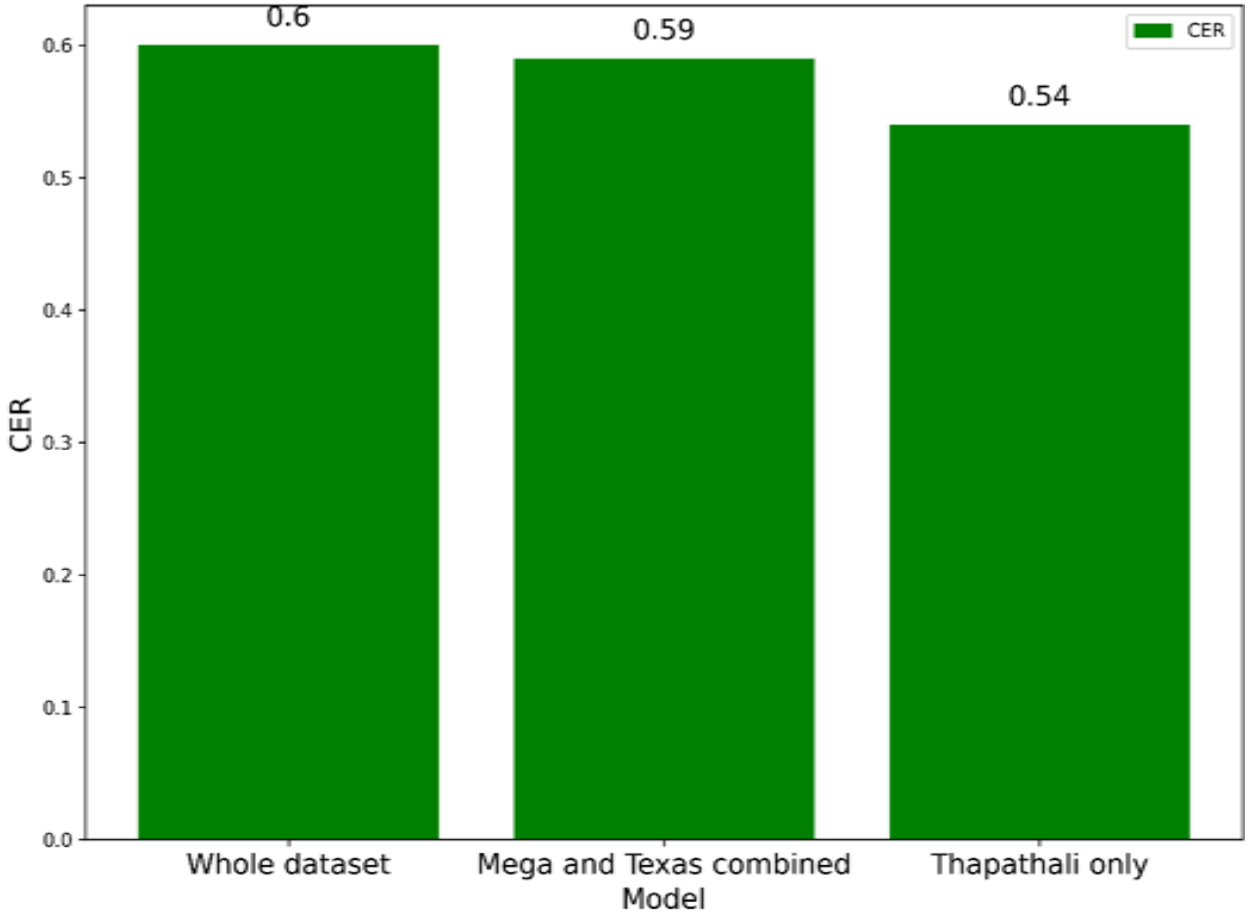


Hyperparameter	Value
Batch Size	16
Epochs	22
Learning Rate	0.001
Dropout	0.1

- Epoch 4 was used for inferencing

Results-[27]
(Speech Model Test Error for Different Datasets)





Results-[28]
(Sample Calculation of WER and CER)

Transcription: " fill up a cup with water"
Prediction": "fill up a cup with water please"

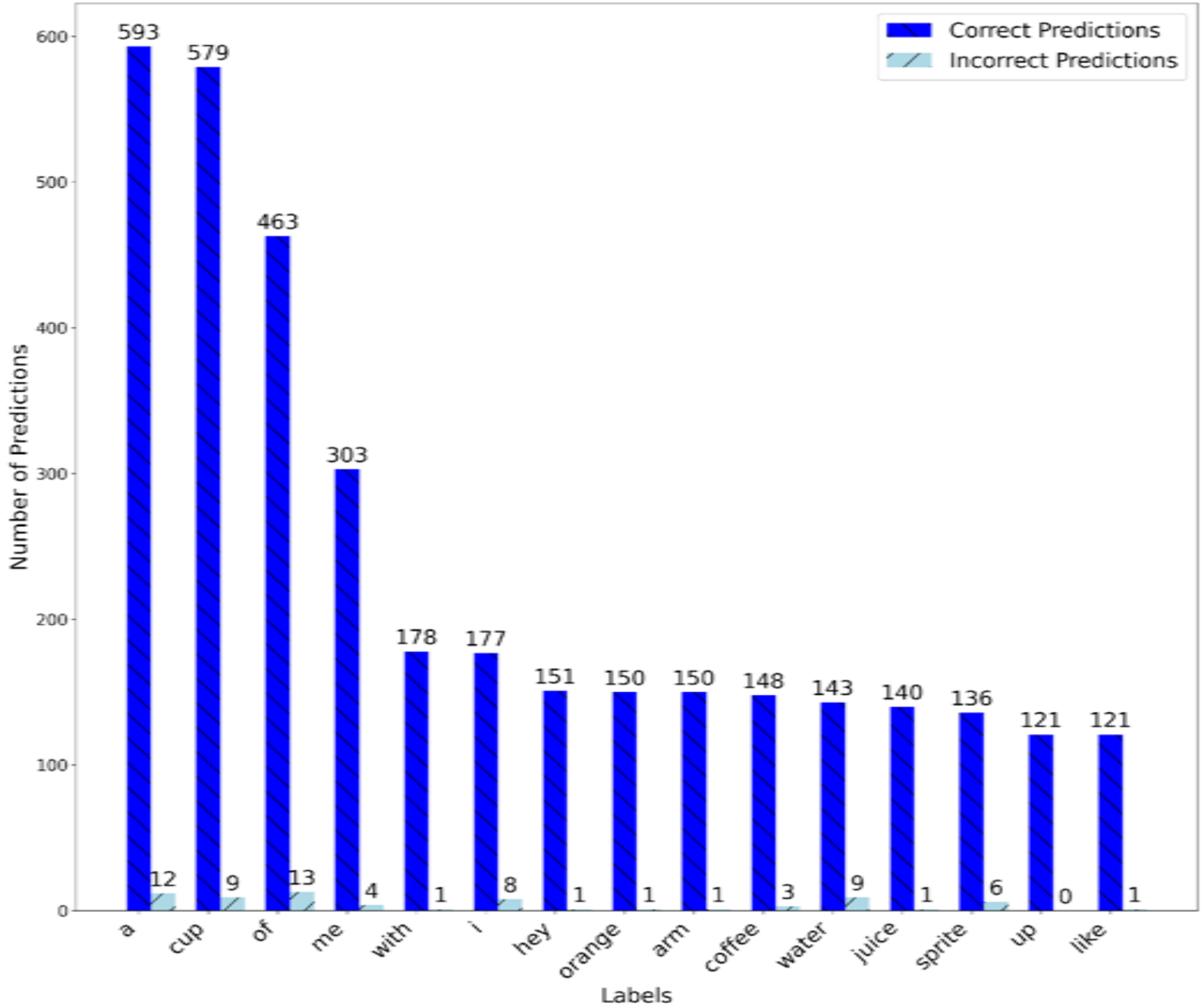
Number of substitutions (S) = 0
Number of deletions (D) = 0
Number of insertions (I) = 1 Number of words in the reference (N) = 6

WER = (S + D + I) / N) * 100%
WER = (1 / 6) * 100% = 16.67%

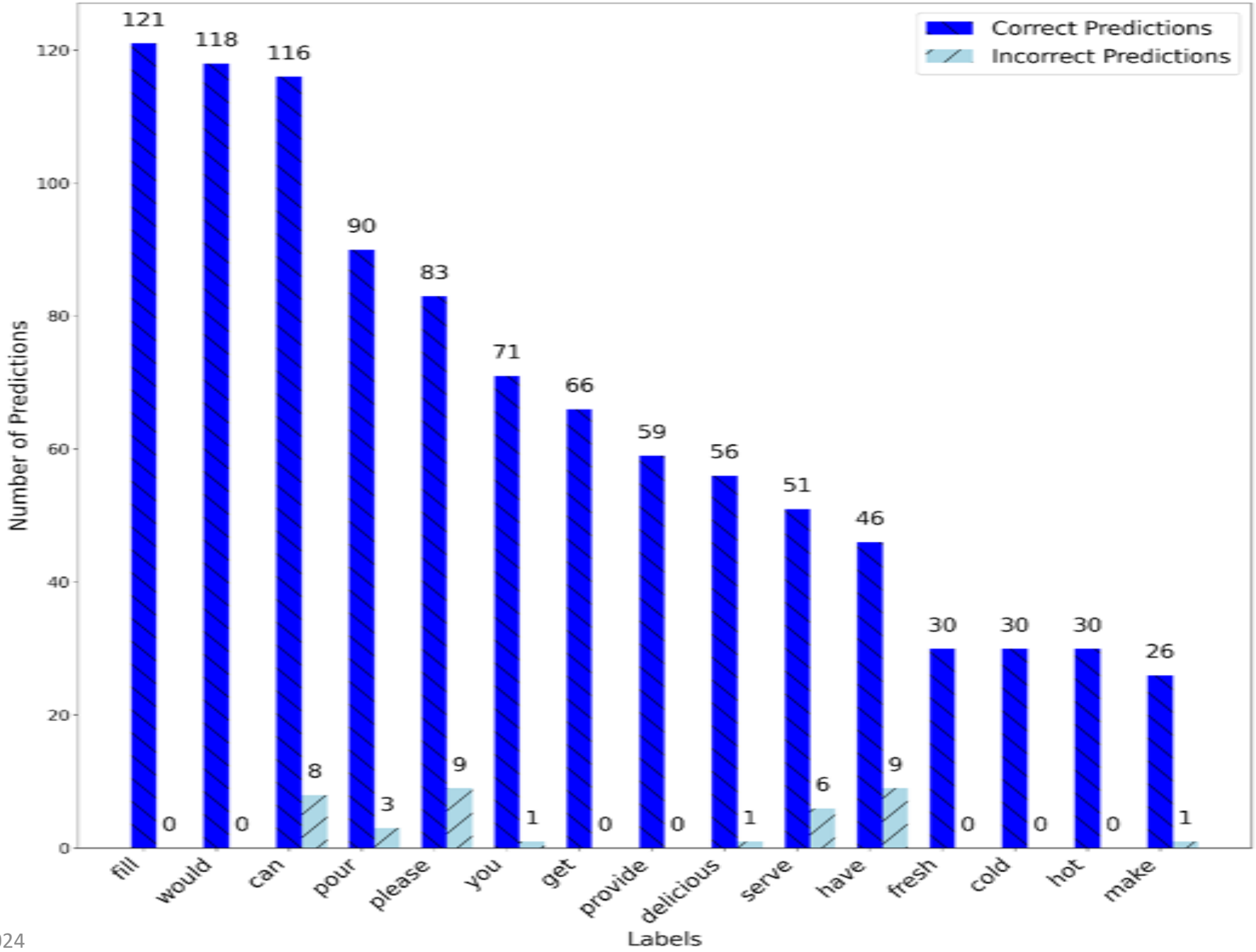
Number of substitutions (S) = 0
Number of deletions (D) = 0
Number of insertions (I) = 7
Number of characters in the reference (N) = 24 (including spaces)

CER = (S + D + I) / N) * 100%
CER = (7 / 24) * 100% = 26.09%

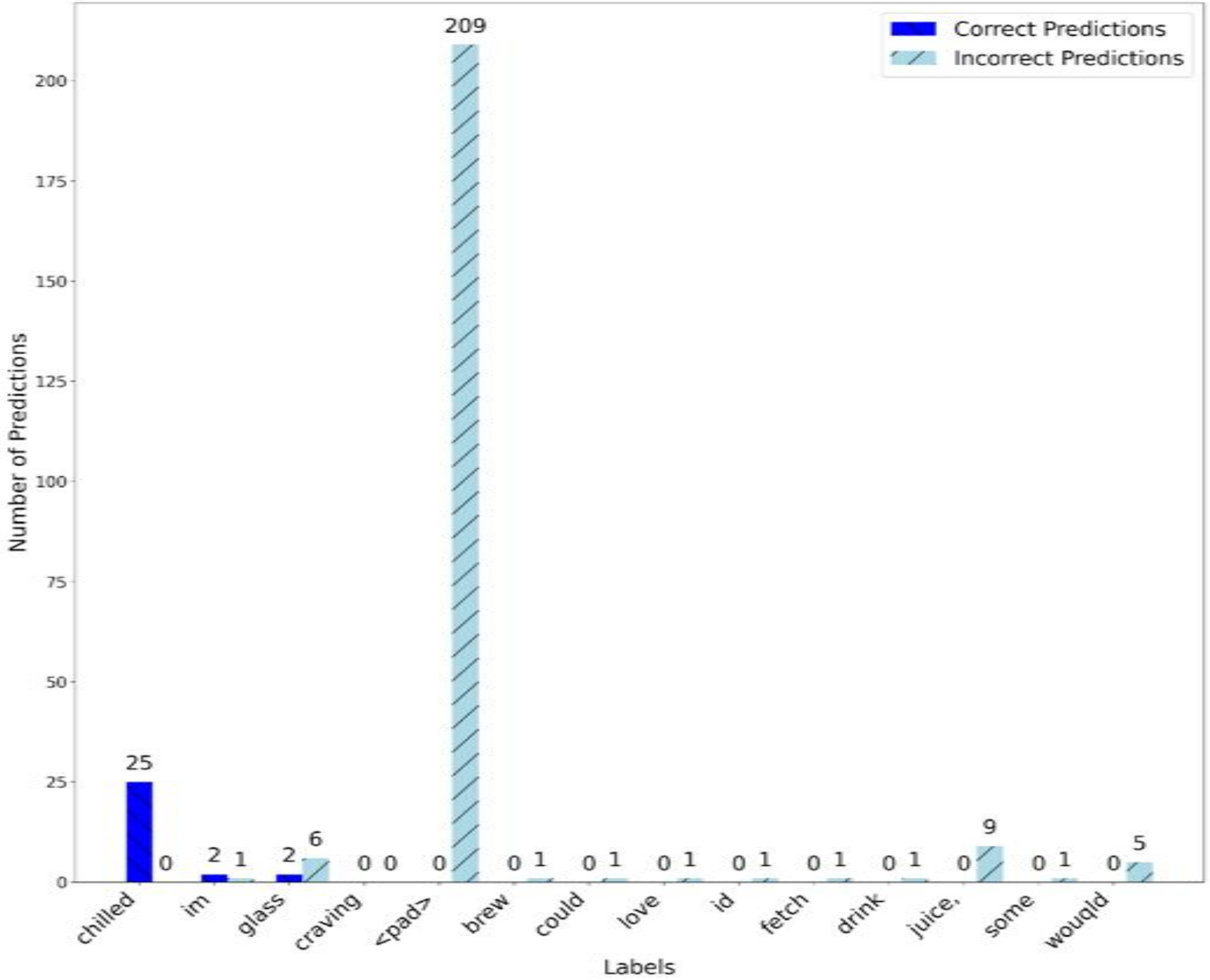
Results -[29] (Confusion Histogram for Labels in Vocabulary)



Results-[30] (Confusion Histogram for Labels in Vocabulary)



Results-[31] (Confusion Histogram for Labels in Vocabulary)



Analysis of Results-[1]
Performance Comparision (Vision System)

- Smooth operation due to static camera in simulation
- Disruptions due to camera movement in reality
- Requires frequent coordinate mapping in reality

Evaluation Metrics	Simulation	Physical
Box Loss	0.38233	0.3361
Class Loss	0.39422	0.30173
Precision Metrics	0.97882	0.99319
Recall Metrics	0.97295	0.98161
mAP@50	0.98941	0.98869

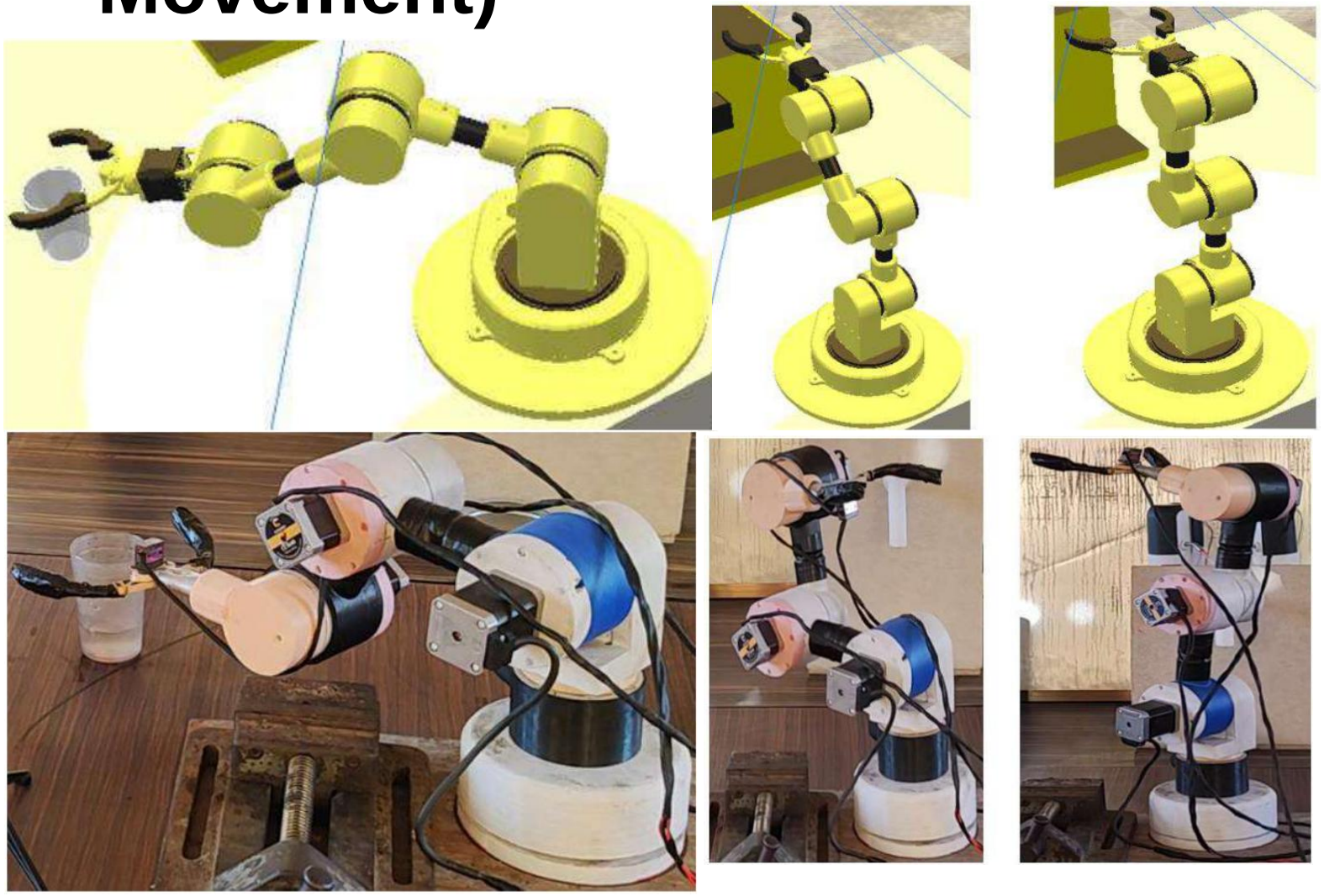
Analysis of Results-[2]
Performance Comparision (Speech Model)

- Lesser error in noiseless environment due to more clarity in sound.
- Lack of clarity in sound due to background noise in reality
- Requires noise removal in reality

Evaluation Metrics	Simulation	Reality
WER	7.08%	12.3%
CER	6.4%	10.4%

Analysis of Results-[3] Performance Comparision (Robotic Arm Movement)

- Simulation
  - Obtain home position after serving drink to client without any torque issues and slippage
- Reality
  - First rotate fourth joint in anticlockwise direction followed by third joint and second joint
  - Obtain home position



Future Enhancements

- Robotic Arm could be enhanced to make it a 6-DOF
- CNC-machined aluminum components could be used
- BLDC motors could be used in joints for more precise control
- Adaptive MPC algorithms could be implemented
- Speech Recognition Model could be trained to be multilingual
- Vision system could be updated by adding depth cameras

Conclusion

- 4DOF robotic arm has been created successfully
- Complete functionality has been realized in simulation and reality
- Performance comparison has been done between simulation and reality
- Contribution has been made in the field of intersection between robotics and machine learning

Budget Analysis (Real Time Solutions)

S.N.	Items	Description	Quantity	Unit Price (NPR)	Total (NPR)	Source
1.	6810	Ball bearings	8	1200	9600	Real Time Solutions
2.	6804	Ball bearings	16	400	6400	Real Time Solutions
3.	Nema 17	45 Ncm	3	1600	4800	Real Time Solutions
4.	Nema 17	60 Ncm	1	3320	3320	Real Time Solutions
5.	M3 screw	Screws	20	5	100	Real Time Solutions
6.	Camera	Vision	1	1700	1700	Real Time Solutions
7.	A4988	Stepper driver	4	210	840	Real Time Solutions
8.	TB6600	Stepper Driver	3	1200	3600	Real Time Solutions
				Grand Total	**30,360**	

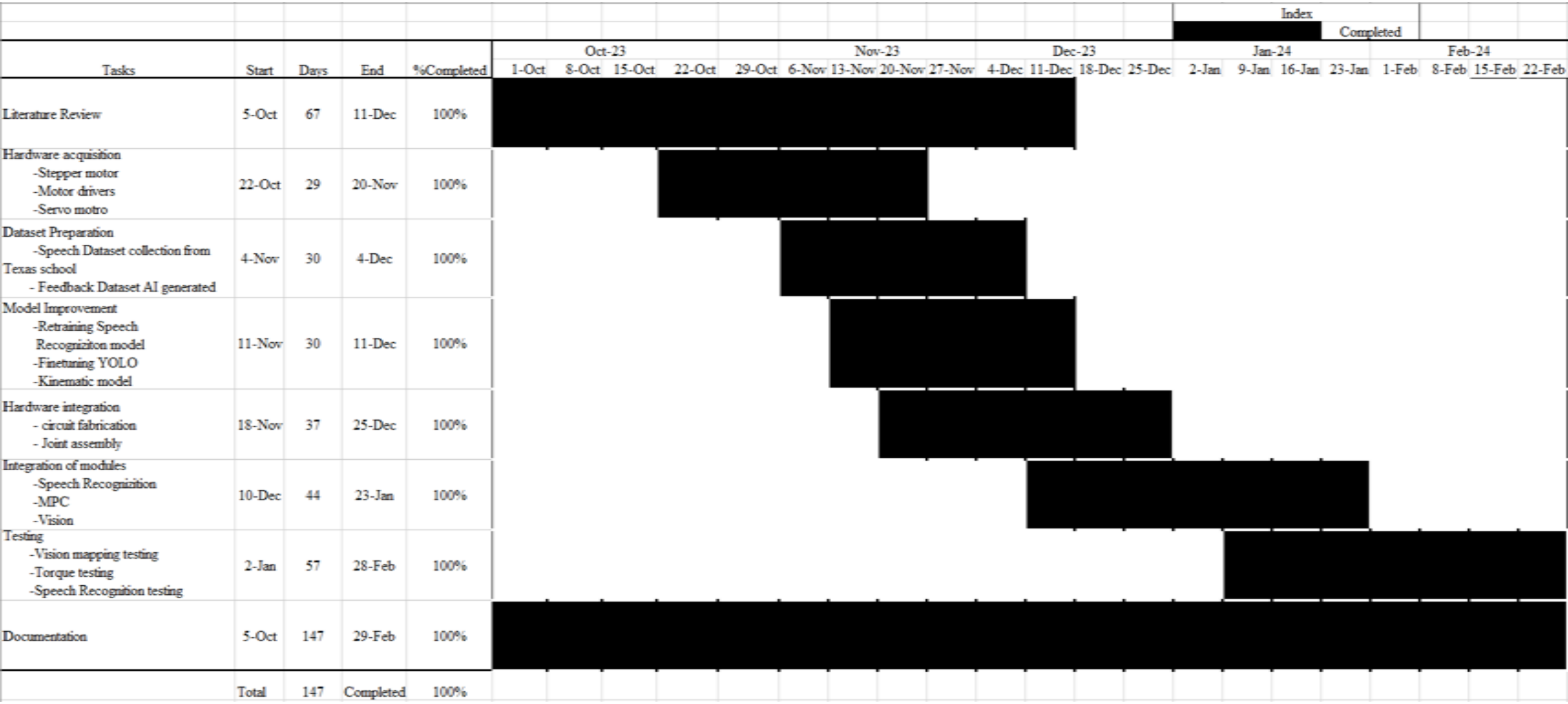
Budget Analysis (Self-Purchased)

S.N.	Items	Description	Quantity	Unit Price (NPR)	Total (NPR)	Source
8.	6mm Bolts	Nut bolts	16	10	160	Self-Purchased
9.	Lathe Turning	For output shafts	16	100	1600	Self-Purchased
10.	DRV8825	Stepper driver	1	500	500	Self-Purchased
11.	3D printing	Gearbox print	4	1470	5880	Self-Purchased
12.	Grease	Lubricant	1	150	150	Self-Purchased
13.	Arduino	Stepper controller	1	1200	1200	Self-Purchased
14.	Dc Pump	Dispenser Pump	4	400	1600	Self-Purchased
15.	Raspberrypi	Main controller	1	29500	29500	Self-Purchased
16.	TB6600	Stepper Driver	1	1200	1200	Self-Purchased
				Grand Total	**41,790**	

Project Schedule (Part-A)

Tasks	Start	Days	End	%Complete
Literature Review	5-Jun	12	21-Jun	100%
Model creation -Speech recogniition model -YOLO model -3D CAD model -MPC model	21-Jun	14	11-Jul	100%
Dataset collection -Speech data from Mega school -Speech data from thapathli campus -Vision data	5-Jul	14	25-Jul	50%
Training of models -Speech recogniition model -YOLO model	25-Jul	10	8-Aug	100%
Integration of modules on copellia -Speech Recogniition -MPC -YOLO	8-Aug	12	24-Aug	100%
Finetuning of models	24-Aug	13	12-Sep	50%
Simulation and testing of system	12-Sep	14	2-Oct	100%
Documentation	5-Jun	89	2-Sep	100%
	Total	89	Completed	100%

Project Schedule (Part-B)



Tasks	Start	Days	End	%Completed
Literature Review	5-Oct	67	11-Dec	100%
Hardware acquisition -Stepper motor -Motor drivers -Servo motro	22-Oct	29	20-Nov	100%
Dataset Preparation -Speech Dataset collection from Texas school - Feedback Dataset AI generated	4-Nov	30	4-Dec	100%
Model Improvement -Retraining Speech Recogniziton model -Finetuning YOLO -Kinematic model	11-Nov	30	11-Dec	100%
Hardware integration - circuit fabrication - Joint assembly	18-Nov	37	25-Dec	100%
Integration of modules -Speech Recognition -MPC -Vision	10-Dec	44	23-Jan	100%
Testing -Vision mapping testing -Torque testing -Speech Recognition testing	2-Jan	57	28-Feb	100%
Documentation	5-Oct	147	29-Feb	100%
	Total	147	Completed	100%

References-[1]

[1] Zhou, Z., Zhang, Y. and Li, Y. (2023) ‘Model predictive control design of a 3-dof robot arm based on recognition of spatial coordinates’, 2023
9th International Conference on Mechatronics and Robotics Engineering (ICMRE), doi:10.1109/icmre56789.2023.10106581.

[2] Voice Transformer Network: Sequence-to-Sequence Voice . https://www.isca-speech.org/archive_v0/Interspeech_2020/pdfs/1066.pdf

[3] L. Dong, S. Xu and B. Xu, "Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition," 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 2018, pp. 5884-5888, doi: 10.1109/ICASSP.2018.8462506.

[4] S. Revay and M. Teschke, “Multiclass Language Identification using Deep Learning on Spectral Images of Audio Signals,” May 2019, [Online]. Available: http://arxiv.org/abs/1905.04348

[5] L. Rafael Stefanel Gris and A. Candido Junior, “Automatic Spoken Language Identification using Convolutional Neural Networks.” [Online]. Available: http://www.freesound.org

[6] P. Kaur, Q. Wang, and W. Shi, “Fall Detection from Audios with Audio Transformers,” Aug. 2022, [Online]. Available: http://arxiv.org/abs/2208.10659

References-[2]

[7] A. A. Q. Mohammed, J. Lv, and M. D. S. Islam, “A deep learning-based end-to-end composite system for hand detection and gesture recognition,” *Sensors (Switzerland)*, vol. 19, no. 23, Dec. 2019, doi: 10.3390/s19235282.

[8] M. Musaev, I. Khujayorov, and M. Ochilov, “Image Approach to Speech Recognition on CNN,” in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Sep. 2019. doi: 10.1145/3386164.3389100.

[9] Y. Gong, Y.-A. Chung, and J. Glass, “AST: Audio Spectrogram Transformer,” Apr. 2021, [Online]. Available: http://arxiv.org/abs/2104.01778

[10] R. P. A. Petrick and M. E. Foster, “Planning for Social Interaction in a Robot Bartender Domain.” [Online]. Available: www.aaai.org

[11] A. V. Oppenheim and A. S. Willsky, Signals and Systems, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 1996. [Online]. Available:https://www.academia.edu/37486178/Signals_and_Systems_2nd_Edition_by_Oppenheim_

[12] Ultralytics, "Ultralytics/ultralytics: Open-source deep learning inference & training on YOLOv3/YOLOv4/PyTorch," GitHub. [Online]. Available: https://github.com/ultralytics/ultralytics