ECE 445

SENIOR DESIGN LABORATORY

DESIGN DOCUMENT

Design Document for ECE 445

Team #36

CHENGHAN LI (cli104@illinois.edu)
YIPU LIAO (yipul2@illinois.edu)
YIMING LI (yiming20@illinois.edu)
HAORAN YANG
(haorany8@illinois.edu)

TA: Leixin Chang

March 24, 2024

Contents

1 Introduction **1**
- 1.1 Problem and Solution Overview . . . 1
- 1.2 Visual Aid . . . 2
- 1.3 High-level requirements list . . . 2

2 Design **4**
- 2.1 Block Diagram: . . . 4
- 2.2 Physical design . . . 5
  - 2.2.1 Subsystem rail way . . . 5
  - 2.2.2 Subsystem robotic arm . . . 6
  - 2.2.3 Parameter . . . 13
  - 2.2.4 Subsystem end-effector . . . 16
- 2.3 Audio Recognition Subsystem . . . 18
  - 2.3.1 Description . . . 18
  - 2.3.2 Requirement and Verification . . . 19
- 2.4 Visual Detection Subsystem . . . 19
  - 2.4.1 Description . . . 19
  - 2.4.2 Requirement and Verification . . . 20
- 2.5 Robot Arm Subsystem . . . 21
  - 2.5.1 Description . . . 21
  - 2.5.2 Requirement and Verification . . . 27
- 2.6 Tolerance Analysis . . . 27
  - 2.6.1 Audio Recognition . . . 27
  - 2.6.2 Visual Detection . . . 27
  - 2.6.3 Robot Arm . . . 28

3 Cost and Schedule **30**
- 3.1 Cost Analysis . . . 30
  - 3.1.1 Labor . . . 30
  - 3.1.2 Parts . . . 30
  - 3.1.3 Total Cost . . . 30
- 3.2 Schedule . . . 31
  - 3.2.1 Timeline . . . 31
  - 3.2.2 Per-person . . . 31

4 Discussion of Ethics and Safety: **33**
- 4.1 Ethics . . . 33
- 4.2 Safety . . . 33

References **35**

5 Appendix a: **36**
- 5.1 Audio Recognition Subsystem . . . 36

5.1.1 Install audio drive 36
5.1.2 CNN-LSTM: 36
5.2 Visual Subsystem 37
5.2.1 System Input and Output 37
5.2.2 Deploy YoloV5 on ROS System 37
5.2.3 Framework of YoloV5 38
5.2.4 Install the Driver of Depth Camera on ROS 40
5.2.5 Detect the Depth of the Object 41

1 Introduction

1.1 Problem and Solution Overview

China's aging population is facing a growing challenge with the increasing prevalence of geriatric diseases, notably Parkinson's disease (PD), a debilitating neurological condition predominantly affecting the elderly. The prevalence of PD among individuals over 65 years old in China is approximately 1.7%, with a projected surge to 5 million patients by 2030, representing nearly half of the global PD population. And the prevalence increases further with age, exceeding 4% in people over 80 years of age [1].This demographic shift underscores the urgent need for innovative solutions to alleviate the burden of PD and enhance the quality of life for affected individuals.

Parkinson's disease manifests through symptoms such as tremors, rigidity, bradykinesia, and postural instability, severely impeding daily activities and independence. Elderly PD patients often struggle with simple tasks due to motor impairments, necessitating accessible and intelligent assistive technologies to support their daily living needs.

In response to this pressing need, our project proposes a novel solution in the form of an intelligent robotic arm system. This system is designed to empower Parkinson's patients by providing assistance with various tasks through advanced speech and visual recognition capabilities. By leveraging cutting-edge technology, our solution aims to revolutionize the way individuals interact with their environment, enabling seamless task execution and fostering greater independence and autonomy.

The specific system can be divided into three subsystems: language model, visual recognition, and robotic arm. These systems interact with each other and cooperate with each other in the process of completing the ordered tasks. First, the language model recognizes the operator's language instructions and converts them into computer signal instructions for transmission to the visual recognition system. The visual recognition system automatically recognizes the target object according to the instruction and the visual information from the camera as input, and generates the position information of the target object as output. Finally, after the robot arm obtains the position information from the visual recognition system, the corresponding motor moves initially, reaches the predetermined position and completes the command task.

The key innovation lies in the integration of a sophisticated language model, visual recognition system, and robotic arm subsystems, which work in tandem to interpret user commands, identify target objects, and execute precise actions. Through this cohesive integration, our intelligent robotic arm offers a comprehensive solution tailored to the specific needs of Parkinson's patients, enhancing their ability to navigate daily challenges with ease and confidence.

In summary, our project represents a significant advancement in assistive technology for Parkinson's disease, offering a practical and innovative solution to address the evolving needs of China's aging population. By harnessing the power of robotics and artificial intelligence, we aim to mitigate the impact of PD and empower individuals to lead fulfilling and independent lives.

1.2 Visual Aid

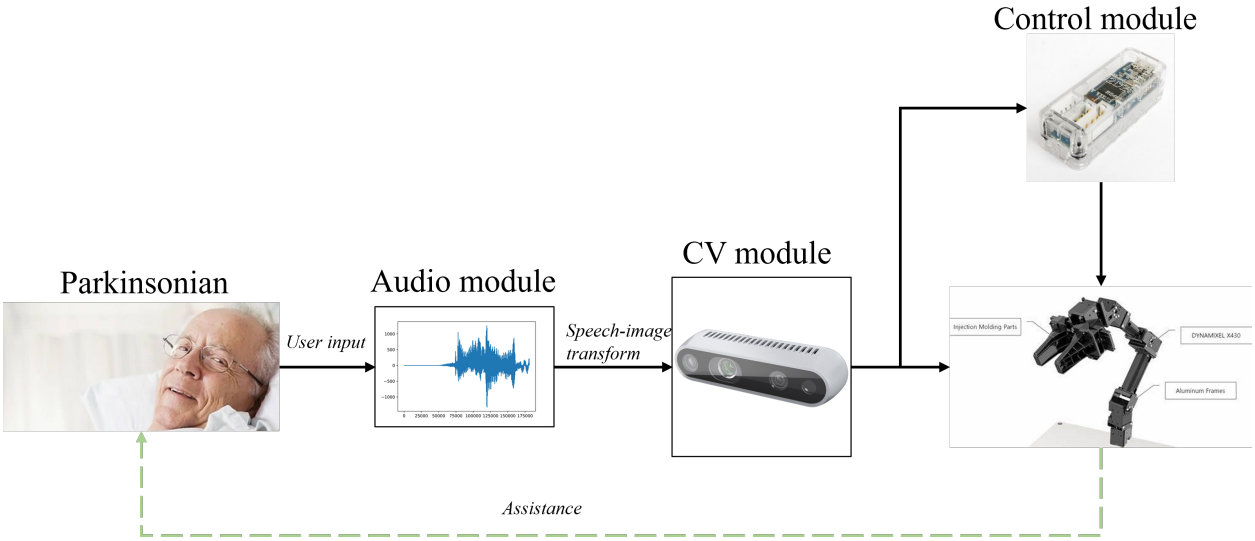


Figure 1: Visual Aid for our Project

1.3 High-level requirements list

1. The robotic arm can accurately move according to the user's instructions through inverse kinematics analysis, and complete a series of tasks such as grasping, transferring and placing, so as to realize the auxiliary living function of Parkinson's patients.

2. The language recognition model and visual recognition model can accurately identify the corresponding voice and image, accurately judge different voice commands and different types of objects in different positions, and transmit the data to the manipulator operation program.

3. Through the specific mechanical structure design, the robot arm should be able to cover a large enough range of space to ensure that the purpose of patient assistance is truly achieved.

2 Design

2.1 Block Diagram:

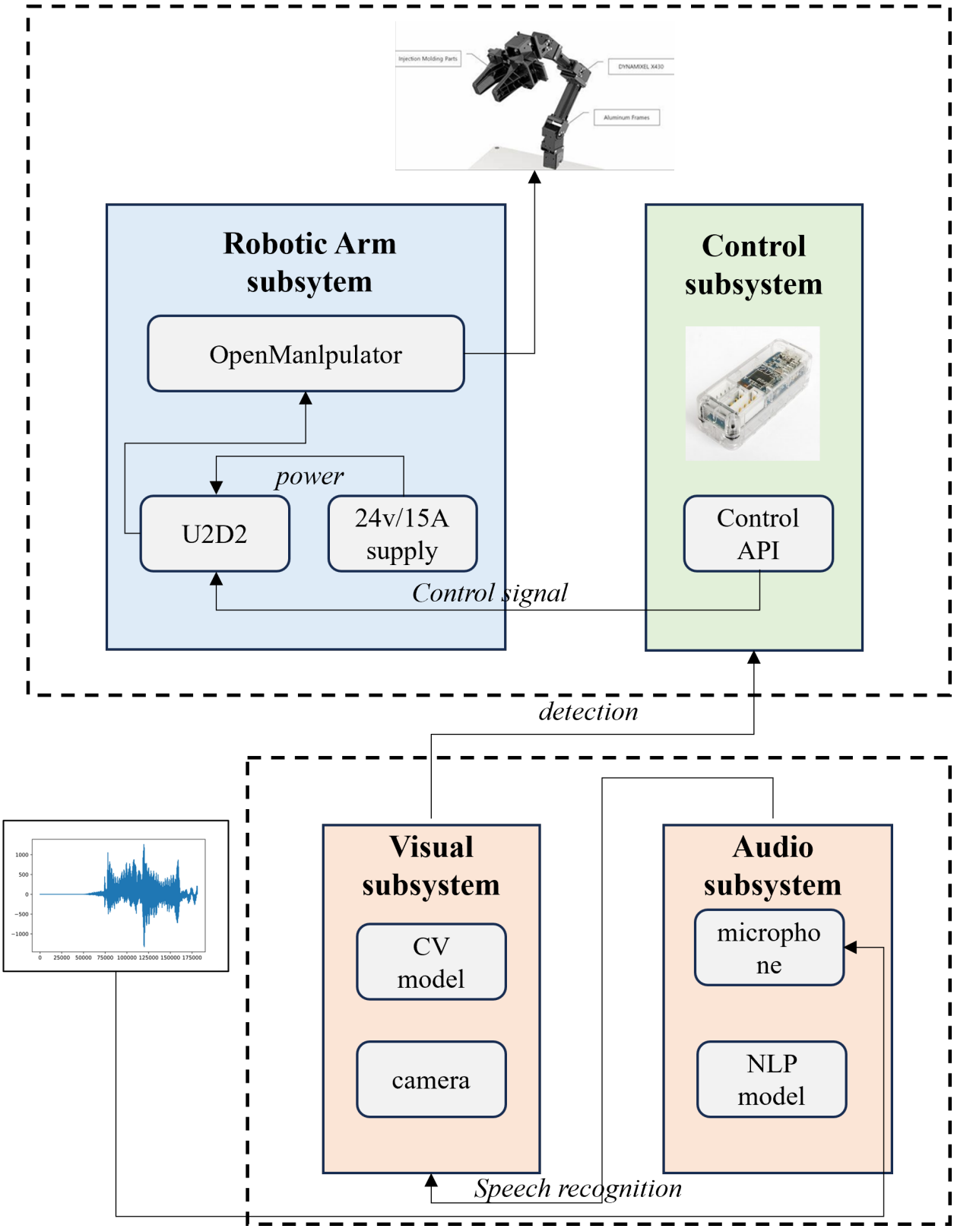


Figure 2: Block Diagram

2.2 Physical design

2.2.1 Subsystem rail way

For the mechanical part, we plan to design a rail that can allow the robotics arm to travel though the edge of the desk in order to get a wider grasping space.

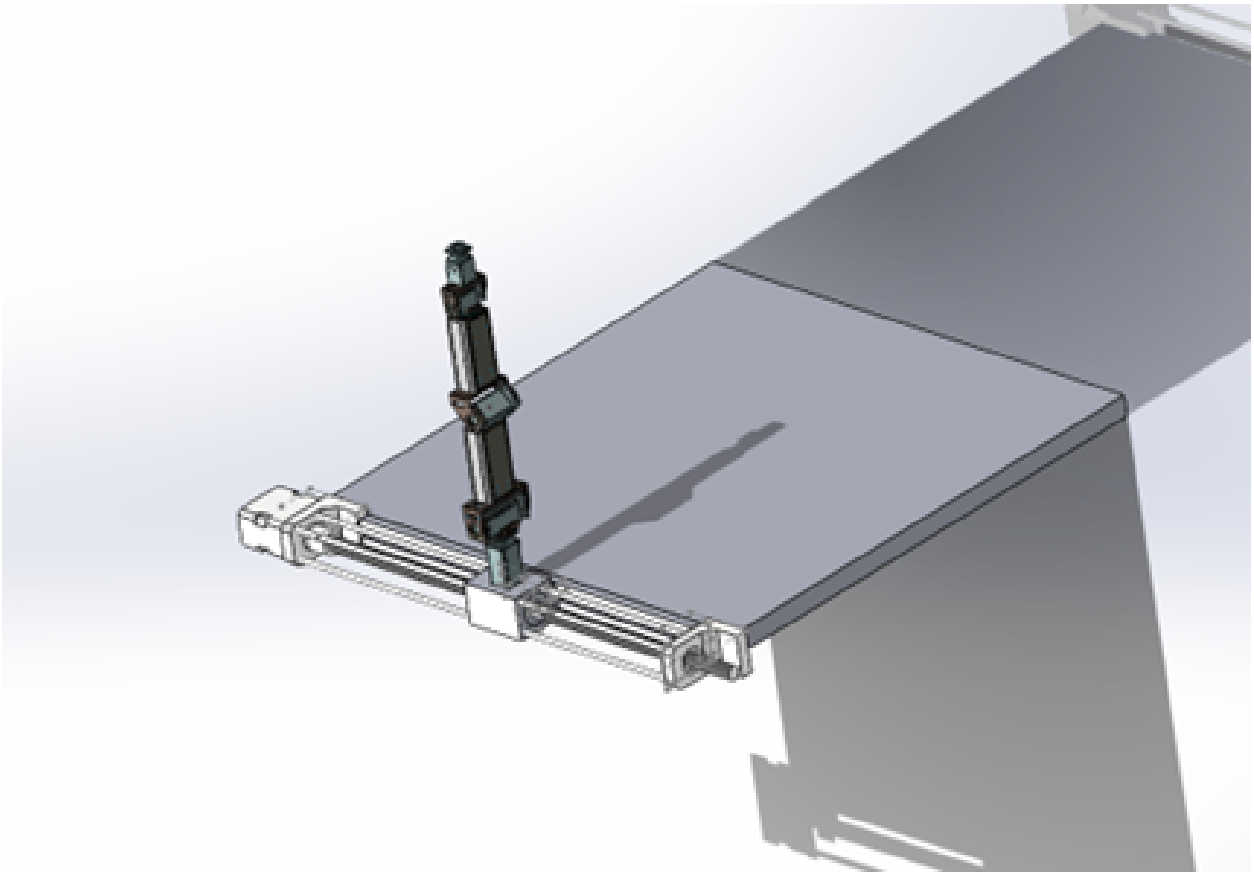


Figure 3: Physical Design Diagram 1

The total length of the lead screw is 1m with diameter of 32 mm and four small rod with the same length of lead screw and diameter of 12mm are applied to help to redistribute the weight of robotics arm. The configuration of mounting is rigid/rigid

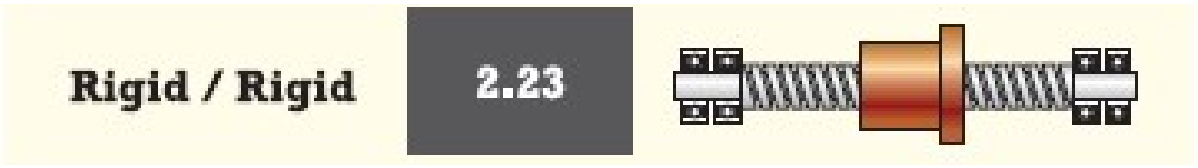


Figure 4: Physical Design Diagram 2

Rigid mounting ensures that the lead screw remains stable and aligned during operation. This stability helps maintain accurate positioning and movement of the load (such as a robotics arm) along the rail. And Rigid support at both ends helps minimize deflection or bending of the lead screw, especially under heavy loads or during high-speed operation. Reduced deflection contributes to smoother and more precise motion control. For the motor, we plan to use NEMA 17 Stepper motors to drive the robotics arm.

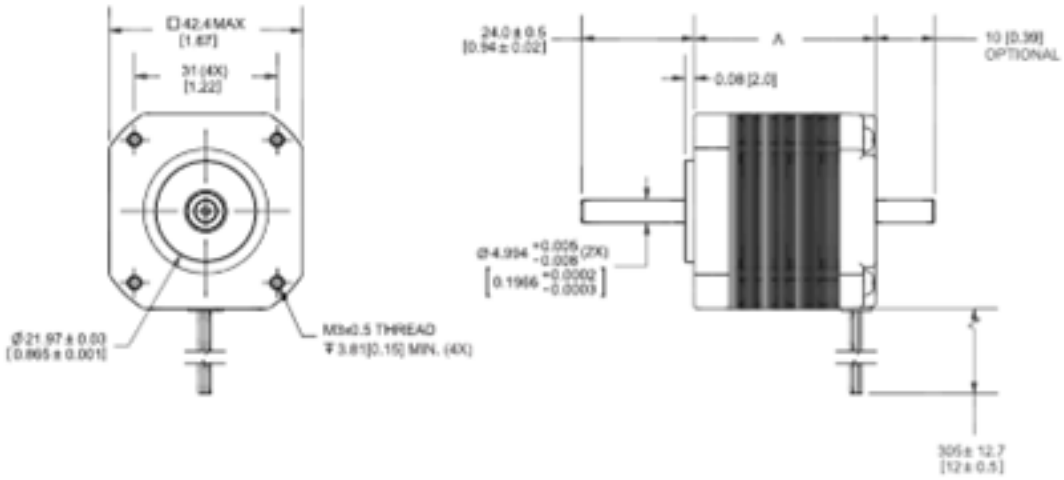
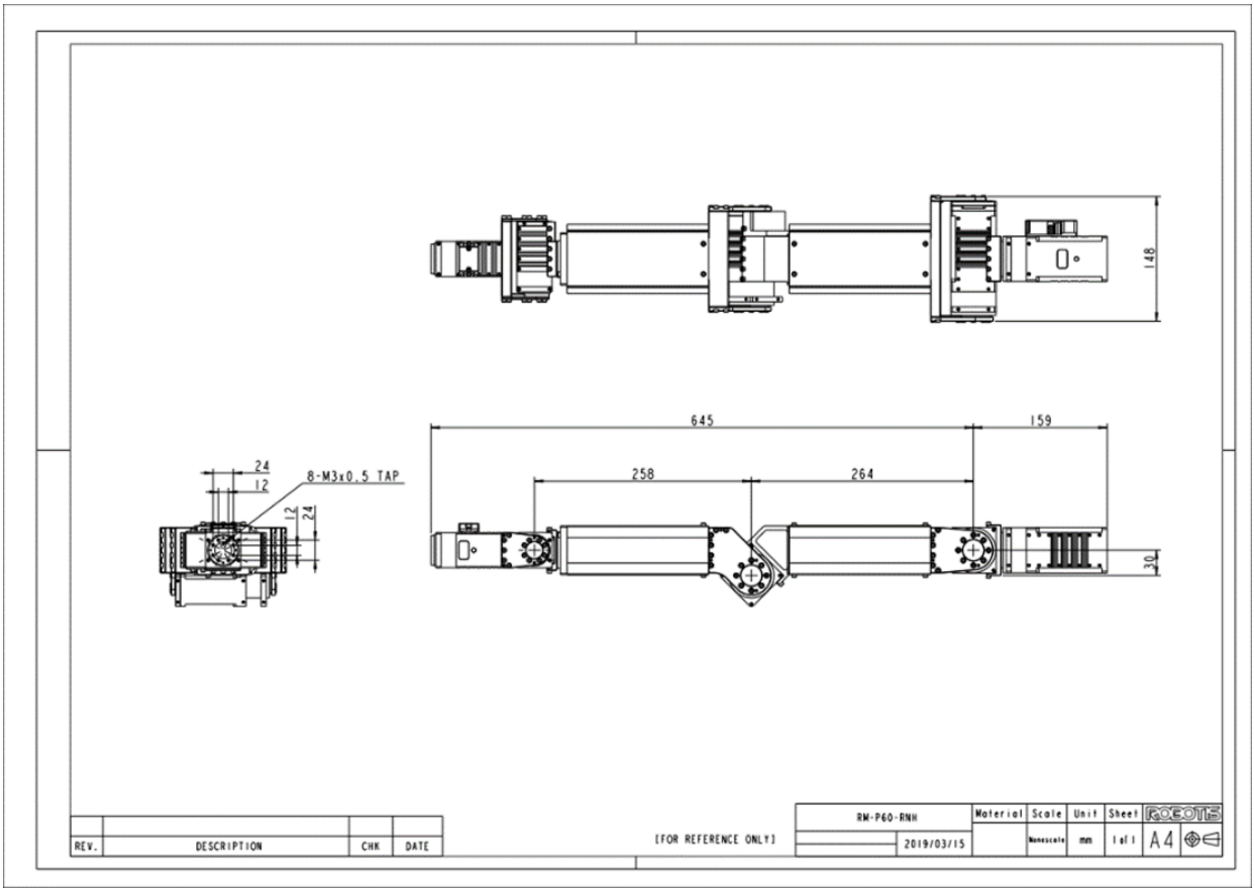


Figure 5: Physical Design Diagram 3

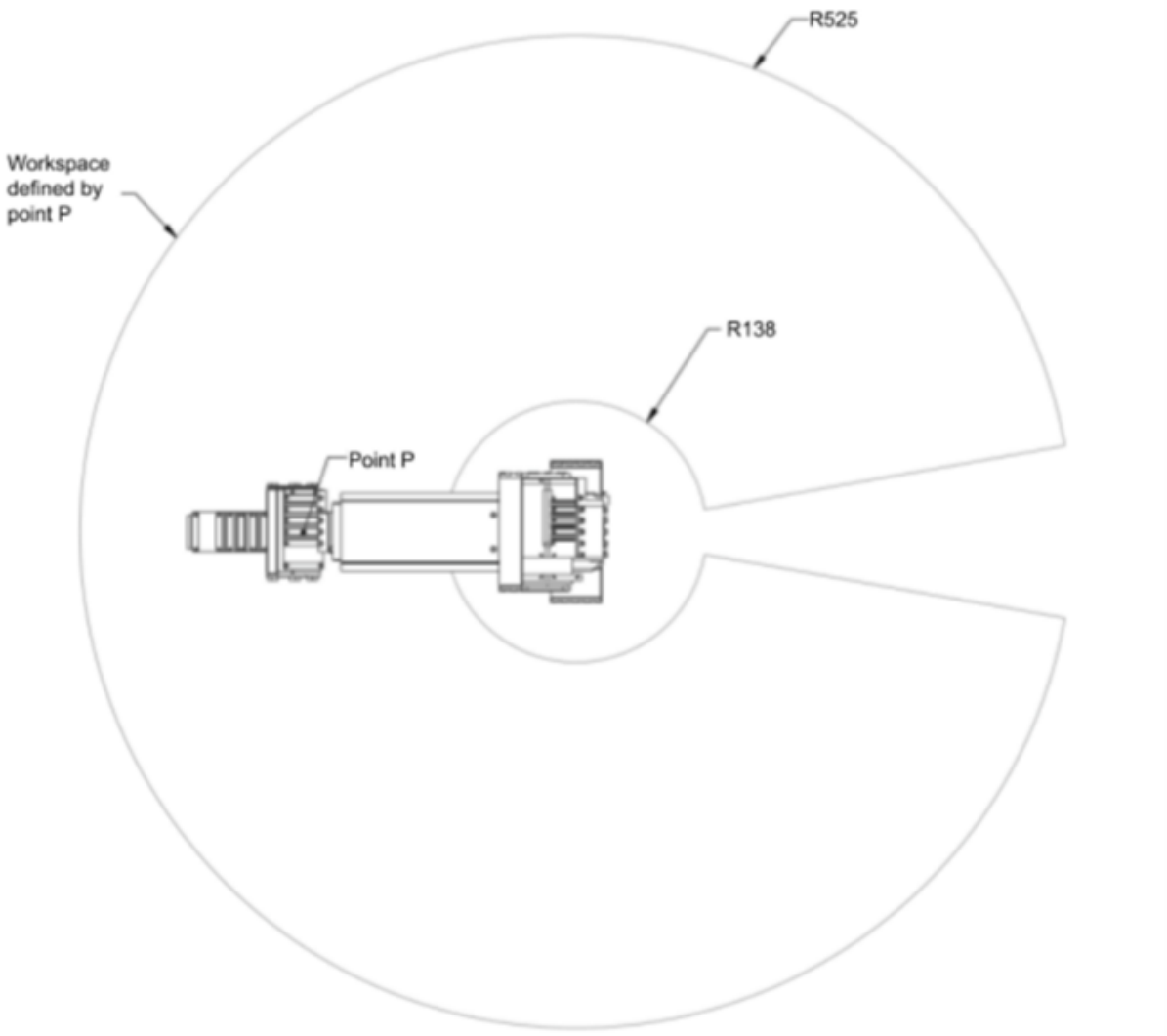
2.2.2 Subsystem robotic arm

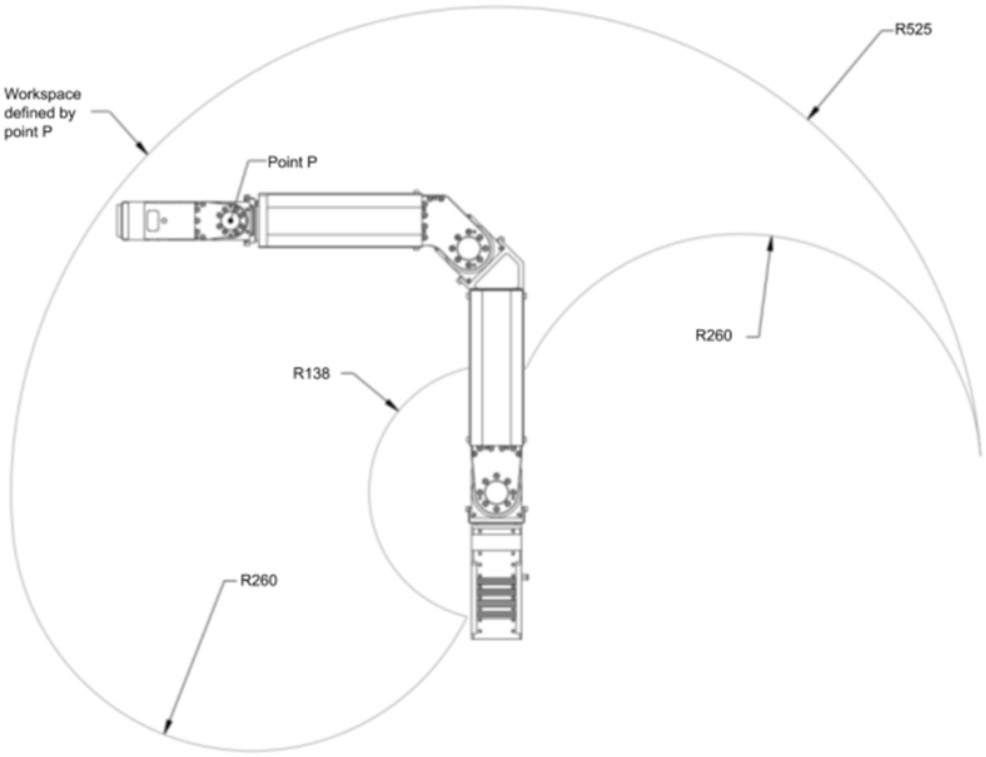
For the robotics arm, we will use OpenMANIPULATORP that has six joints. The OpenMANIPULATORP is versatile and can be used for various tasks, including pick-and-place operations, manipulation of objects, and simple assembly tasks. Its multiple degrees of freedom (DOF) and configurable end-effectors make it suitable for a wide range of applications. We will modify it so that it can achieve our goal of helping people.

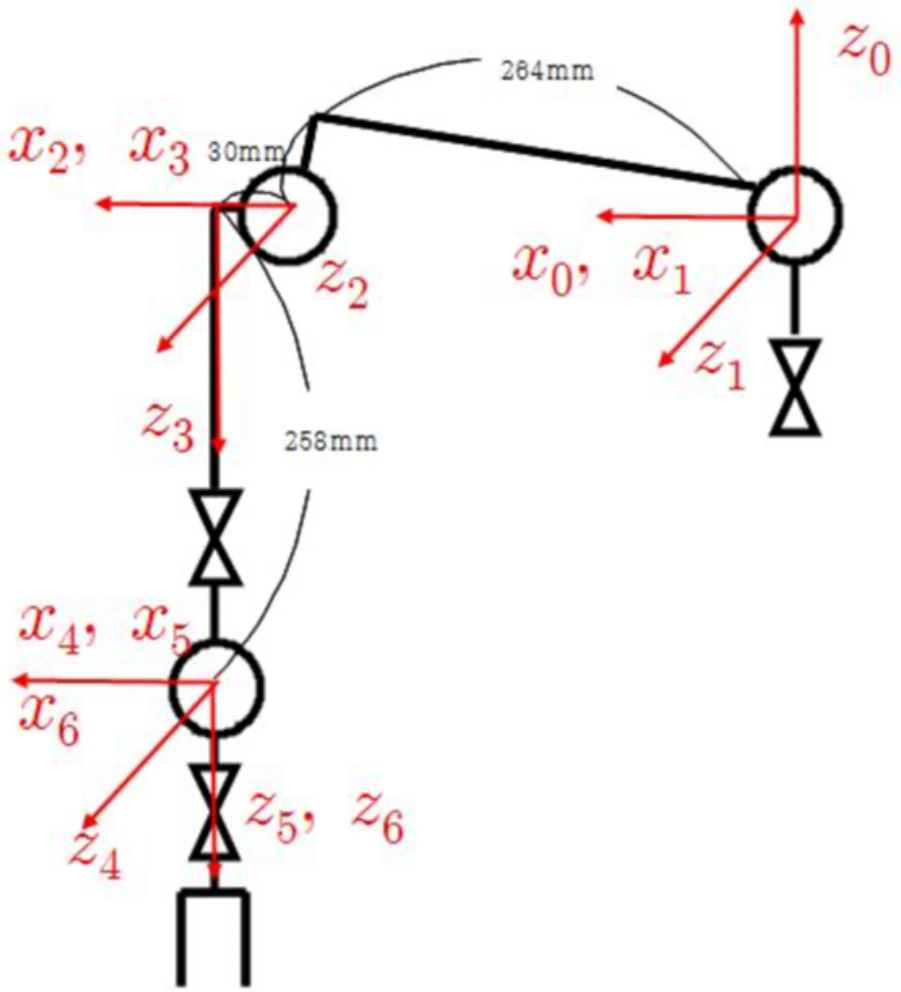
Radial Play	0.001 in @ 1 lbs
End Play	0.003 in @ 3 lbs
Shaft Run Out	0.002 TIR
Concentricity of Mounting Pilot to Shaft	0.003 TIR
Perpendicularity of Mounting Pilot to Face	0.003 TIR
Max Radial Load at Dimension "K" from mounting face	6 lbs
Dimension "K"	0.620 in
Max Axial Load	6 lbs
Maximum Case Temperature	176.00 °F maximum
Ambient Temperature	-4 °F to 122.00 °F
Storage Temperature	-4 °F to 212.00 °F
Humidity Range (%)	85% or less, non-condensing
Magnet Wire Insulation	Class B 130 deg C
Insulation Resistance	100M Ohm at 500 VDC
Dielectric Strength	500 VDC for 1 min


148
645
159
258
264
24
12
8-M3x0.5 TAP
12
24
30
REV.
DESCRIPTION
CHK
DATE
[FOR REFERENCE ONLY]
RM-P60-RNH
2019/03/15
Material
Scale
Unit
Sheet
Nonescale
mm
1 of 1
A4

Item	OpenMANIPULATOR-P (RM-P60-RNH)
DOF	6
Payload	3 kg
Reach	645 mm
Repeatability	±0.05 mm
Weight	5.76 kg
Operating voltage	24 V
Resolution	Joint 1 : -π(rad) ~ π(rad) , -501,923 ~ 501,923 (pulse) Joint 2 : -π(rad) ~ π(rad) , -501,923 ~ 501,923 (pulse) Joint 3 : -π(rad) ~ π(rad) , -501,923 ~ 501,923 (pulse) Joint 4 : -π(rad) ~ π(rad) , -501,923 ~ 501,923 (pulse) Joint 5 : -π(rad) ~ π(rad) , -303,750 ~ 303,750 (pulse) Joint 6 : -π(rad) ~ π(rad) , -303,750 ~ 303,750 (pulse)
DYNAMIXEL Pro Model Name	Joint 1, 2 : PH54-200-S500-R (200W) Joint 3, 4 : PH54-100-S500-R(100W) Joint 5, 6 : PH42-020-S300-R (20W)
Operating Range	Joint 1 : -π(rad) ~ π(rad) Joint 2 : -π/2(rad) ~ π/2(rad) Joint 3 : -π/2(rad) ~ 3π/4(rad) Joint 4 : -π(rad) ~ π(rad) Joint 5 : -π/2(rad) ~ π/2(rad) Joint 6 : -π(rad) ~ π(rad)
Default ID	Joint 1 (ID:1), Joint 2 (ID:2), Joint 3 (ID:3), Joint 4 (ID:4), Joint 5 (ID:5), Joint 6 (ID:6)
Motor type	Brushless DC Servo(H54P Series), Coreless DC Motor(H42P Series)
Position sensor type	Absolute Encoder(for Homing), Incremental Encoder(for Control)
Communications	RS485

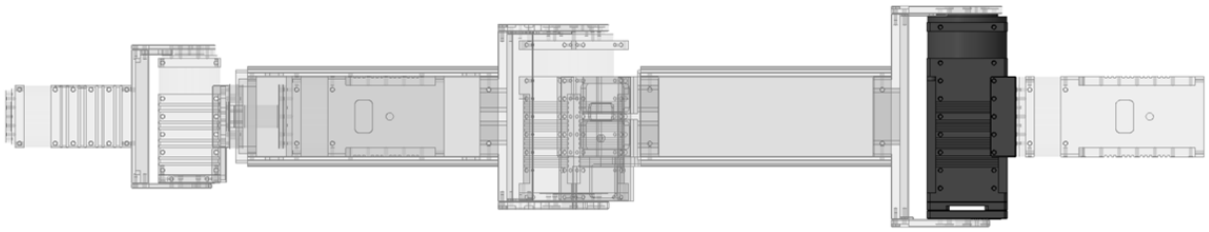

R525
Workspace
defined by
point P
R138
Point P


R525
Workspace
defined by
point P
Point P
R260
R138
R260



Link	Link Length(mm)	Link Twist(rad)	Joint Offset(mm)	Joint Angle(rad)	DXL Angle(rad)
1	0	-π/2	0	0	0
2	265.69	0	0	0	$\frac{\pi}{2}-\tan^{-1}(\frac{30}{264})$
3	30	-π/2	0	0	$\frac{\pi}{4}+\tan^{-1}(\frac{30}{264})$
4	0	-π/2	258	0	0
5	0	-π/2	0	0	0
6	0	0	0	0	0

2.2.3 Parameter

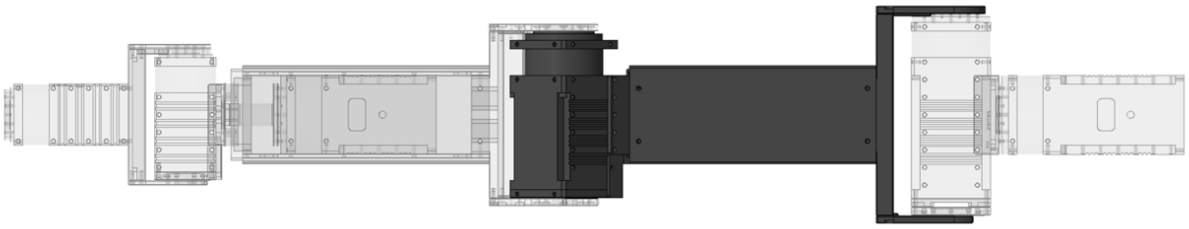


Mass [Kg] : 9.4360595e-01
Center of Gravity [m]
x : 0.0000000e+00
y : -1.7653633e-04
z : -1.0030209e-03

Inertia Tensor with respect to coordinate frame [Kg * m2]
Ixx Ixy Ixz : 1.5694005e-03 0.0000000e+00 0.0000000e+00
Iyx Iyy Iyz : 0.0000000e+00 4.5593385e-04 6.4581824e-09
Izx Izy Izz : 0.0000000e+00 6.4581824e-09 1.5561809e-03

Inertia Tensor at CENTER OF GRAVITY with respect to coordinate frame [Kg * m2]
Ixx Ixy Ixz : 1.5684218e-03 0.0000000e+00 0.0000000e+00
Iyx Iyy Iyz : 0.0000000e+00 4.5498454e-04 1.7354214e-07
Izx Izy Izz : 0.0000000e+00 1.7354214e-07 1.5561515e-03

Principal Moments of Inertia [Kg * m2]
I1 : 4.5498451e-04
I2 : 1.5561515e-03
I3 : 1.5684218e-03

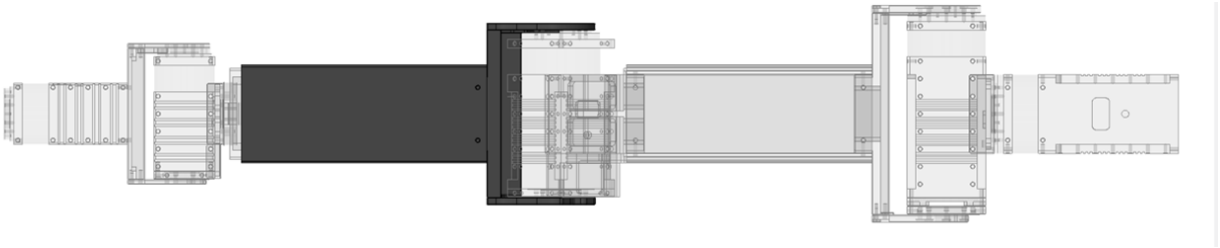


Mass [Kg] : 1.3825862e+00
Center of Gravity [m]
x : 1.5751501e-02
y : -2.2073221e-04
z : 1.9913687e-01
Inertia Tensor with respect to coordinate frame [Kg * m2]

Ixx Ixy Ixz : 6.7630430e-02 -1.9988597e-05 -5.7477571e-03
Iyx Iyy Iyz : -1.9988597e-05 6.7208001e-02 7.4823203e-05
Izx Izy Izz : -5.7477571e-03 7.4823203e-05 2.6031353e-03

Inertia Tensor at CENTER OF GRAVITY with respect to coordinate frame [Kg * m2]
Ixx Ixy Ixz : 1.2803228e-02 -2.4795661e-05 -1.4109928e-03
Iyx Iyy Iyz : -2.4795661e-05 1.2037834e-02 1.4050354e-05
Izx Izy Izz : -1.4109928e-03 1.4050354e-05 2.2600348e-03

Principal Moments of Inertia [Kg * m2]
I1 : 2.0744566e-03
I2 : 1.2037112e-02
I3 : 1.2989528e-02

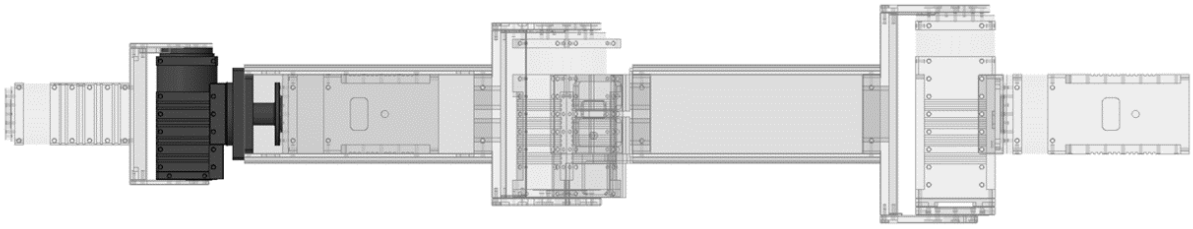


Mass [Kg] : 1.2126965e+00
Center of Gravity [m]
x : 3.0352597e-04
y : 4.1703880e-05
z : 3.8074728e-01

Inertia Tensor with respect to coordinate frame [Kg * m2]
Ixx Ixy Ixz : 1.7985424e-01 -9.9417476e-07 -8.1564441e-05
Iyx Iyy Iyz : -9.9417476e-07 1.7948679e-01 -1.7705853e-05
Izx Izy Izz : -8.1564441e-05 -1.7705853e-05 1.1422079e-03

Inertia Tensor at CENTER OF GRAVITY with respect to coordinate frame [Kg * m2]
Ixx Ixy Ixz : 4.0514666e-03 -9.7882420e-07 5.8582868e-05
Iyx Iyy Iyz : -9.7882420e-07 3.6839091e-03 1.5501165e-06
Izx Izy Izz : 5.8582868e-05 1.5501165e-06 1.1420941e-03

Principal Moments of Inertia [Kg * m2]
I1 : 1.1409140e-03
I2 : 3.6839076e-03
I3 : 4.0526481e-03

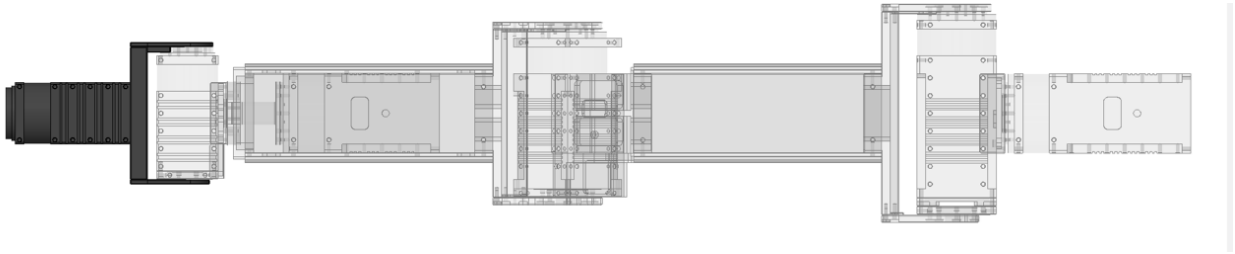


Mass [Kg] : 4.6635550e-01
Center of Gravity [m]
x : -2.1388493e-06
y : -2.2290515e-03
z : 5.1387207e-01

Inertia Tensor with respect to coordinate frame [Kg * m2]
Ixx Ixy Ixz : 1.2357713e-01 -1.9465317e-09 5.1228604e-07
Iyx Iyy Iyz : -1.9465317e-09 1.2339567e-01 5.3769657e-04
Izx Izy Izz : 5.1228604e-07 5.3769657e-04 3.3216863e-04

Inertia Tensor at CENTER OF GRAVITY with respect to coordinate frame [Kg * m2]
Ixx Ixy Ixz : 4.2687885e-04 0.0000000e+00 0.0000000e+00
Iyx Iyy Iyz : 0.0000000e+00 2.4774029e-04 3.5109263e-06
Izx Izy Izz : 0.0000000e+00 3.5109263e-06 3.2985146e-04

Principal Moments of Inertia [Kg * m2]
I1 : 2.4759044e-04
I2 : 3.3000130e-04
I3 : 4.2687885e-04



Mass [Kg] : 4.2561606e-01
Center of Gravity [m]
x : -2.1268822e-06
y : 1.8039922e-04
z : 5.9028250e-01

Inertia Tensor with respect to coordinate frame [Kg * m2]
Ixx Ixy Ixz : 1.4881319e-01 0.0000000e+00 5.3435266e-07

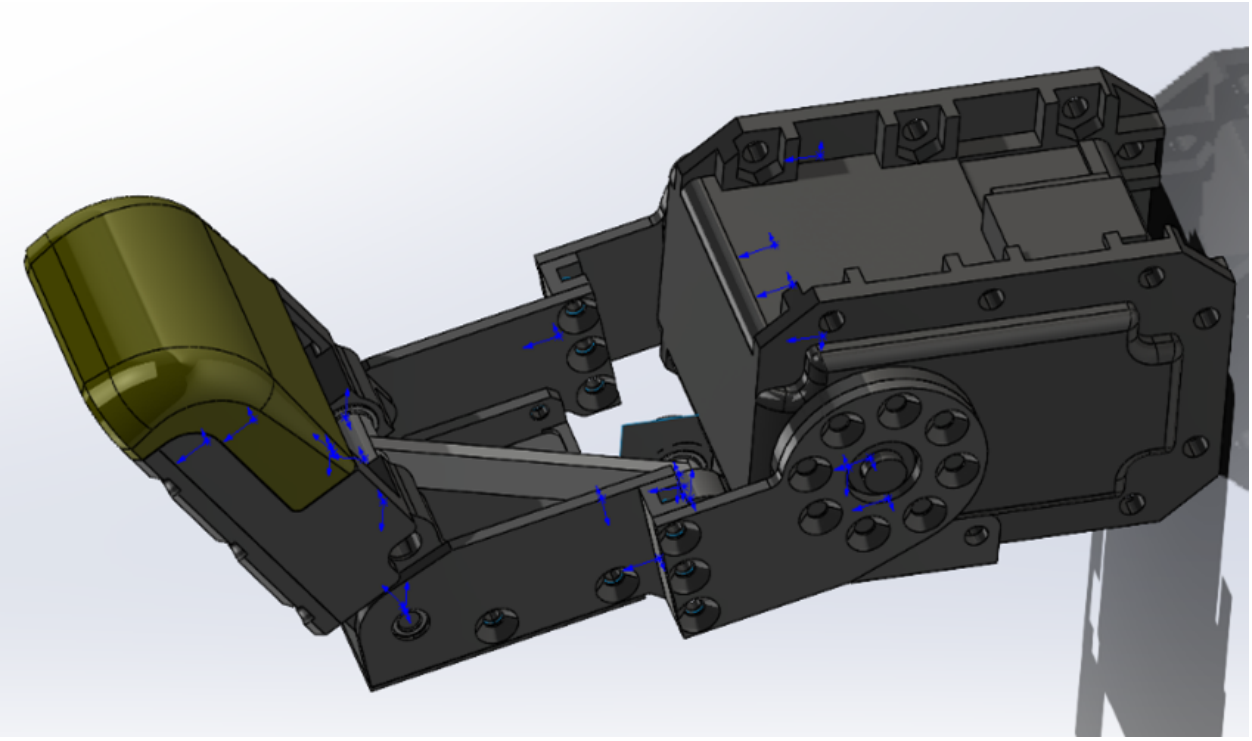
Iyx Iyy Iyz : 0.0000000e+00 1.4872708e-01 -4.3708754e-05
Izx Izy Izz : 5.3435266e-07 -4.3708754e-05 2.1040082e-04

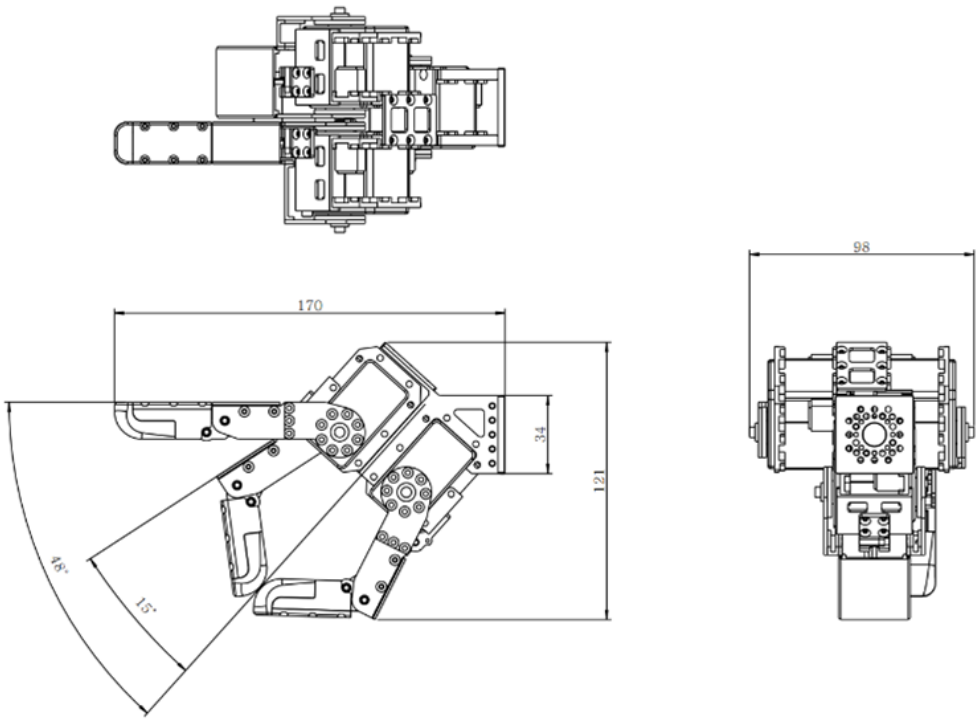
Inertia Tensor at CENTER OF GRAVITY with respect to coordinate frame [Kg * m2]
Ixx Ixy Ixz : 5.1431341e-04 0.0000000e+00 0.0000000e+00
Iyx Iyy Iyz : 0.0000000e+00 4.2820999e-04 1.6136111e-06
Izx Izy Izz : 0.0000000e+00 1.6136111e-06 2.1038697e-04

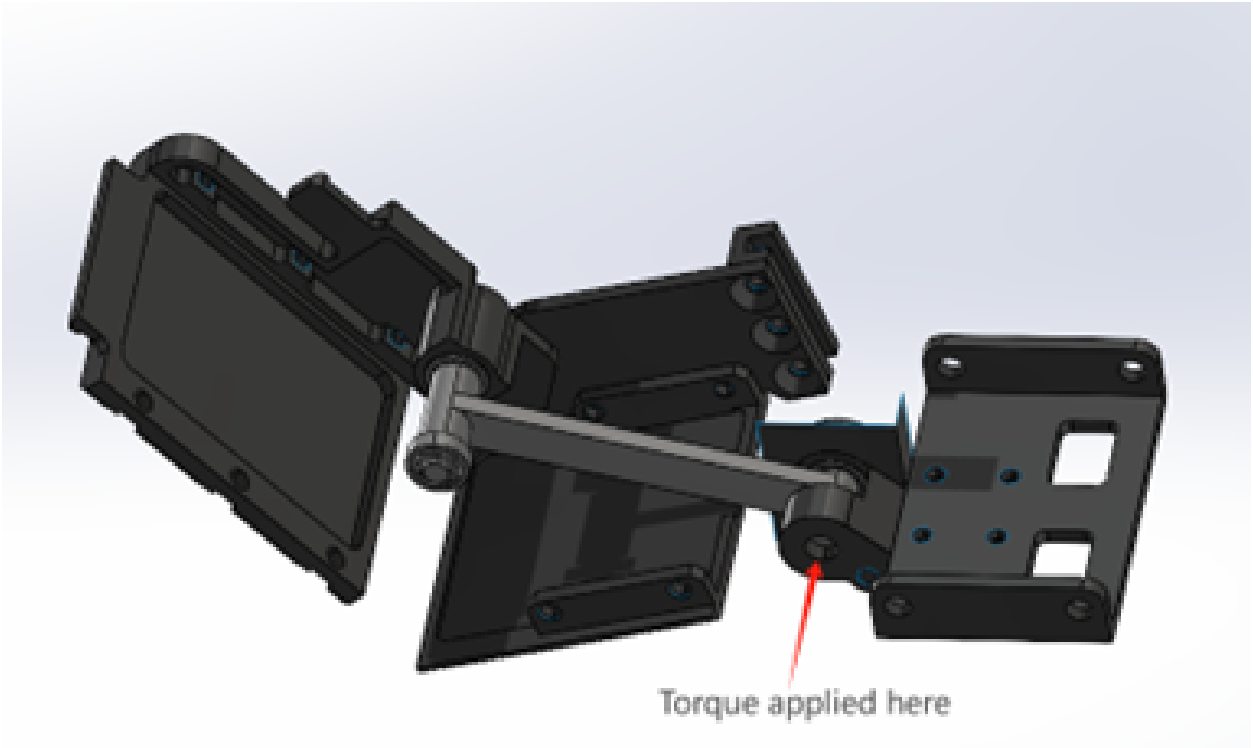
Principal Moments of Inertia [Kg * m2]
I1 : 2.1037502e-04
I2 : 4.2822194e-04
I3 : 5.1431341e-04

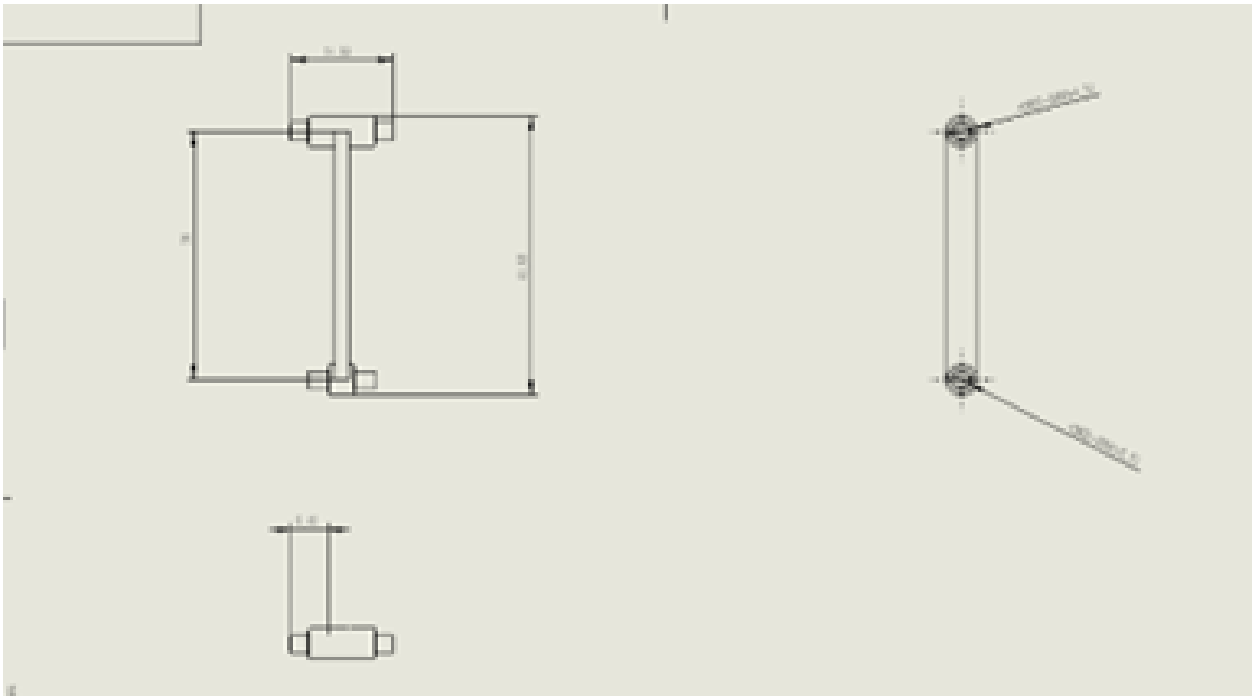
2.2.4 Subsystem end-effector

Torque will be applied so the finger can rotate, and multiple fingers will be added together to form a grasper for real-world task.




98
170
34
121
48°
15°


Torque applied here



2.3 Audio Recognition Subsystem

2.3.1 Description

The audio recognition system comprises the audio capture, audio processing, model recognition, and output modules. Initially, the audio capture module serves as the system's first step, utilizing an advanced microphone array system to capture human audio. This microphone array system, equipped with multiple microphones, accurately captures sound from various directions and distances, ensuring the acquisition of clear audio signals.
More detail about principle and design of our audio recognition subsystem can be seen at Appendix a (5.1).

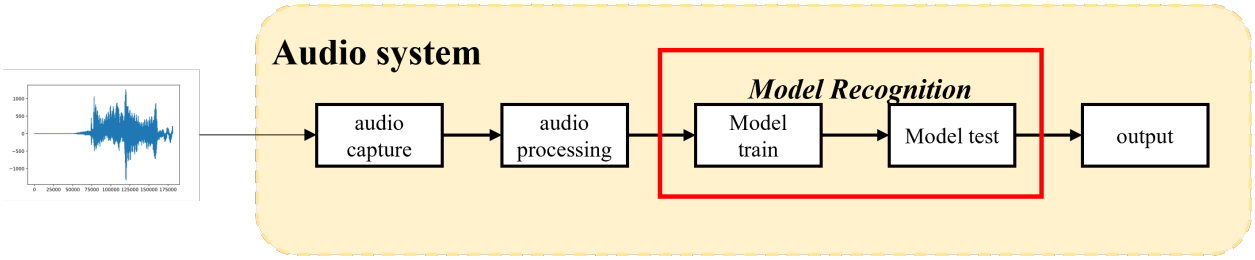


Figure 6: Audio Recognition Subsystem

2.3.2 Requirement and Verification

The R&F for Audio Recognition Subsystem as below:

Table 1: R&F for Audio Recognition Subsystem

Requirement	Verification
1. A microphone can capture audio 2. The audio recognition accuracy rate is above 90%	A. Testing AI models using publicly available speech data sets B. Find classmates to conduct speech recognition tests in real situations C. The test microphone will be connected to the computer and open the ROS voice driver to see if the microphone can work properly

2.4 Visual Detection Subsystem

2.4.1 Description

Visual Detection Subsystem is a target detection module. Its function is to detect the three-dimensional coordinates of the desired object in the real-world coordinate system and transmit these coordinates to the Control Subsystem. The Control Subsystem, equipped with ROS (Robot Operating System), then issues instructions to the robotic arm based on these coordinates. The Visual Detection Subsystem includes a depth camera for real-time environmental detection around the robotic arm, as well as a set of deep learning algorithms and networks for target detection.
More detail about principle and design of our visual detection subsystem can be seen at Appendix a (5.2).

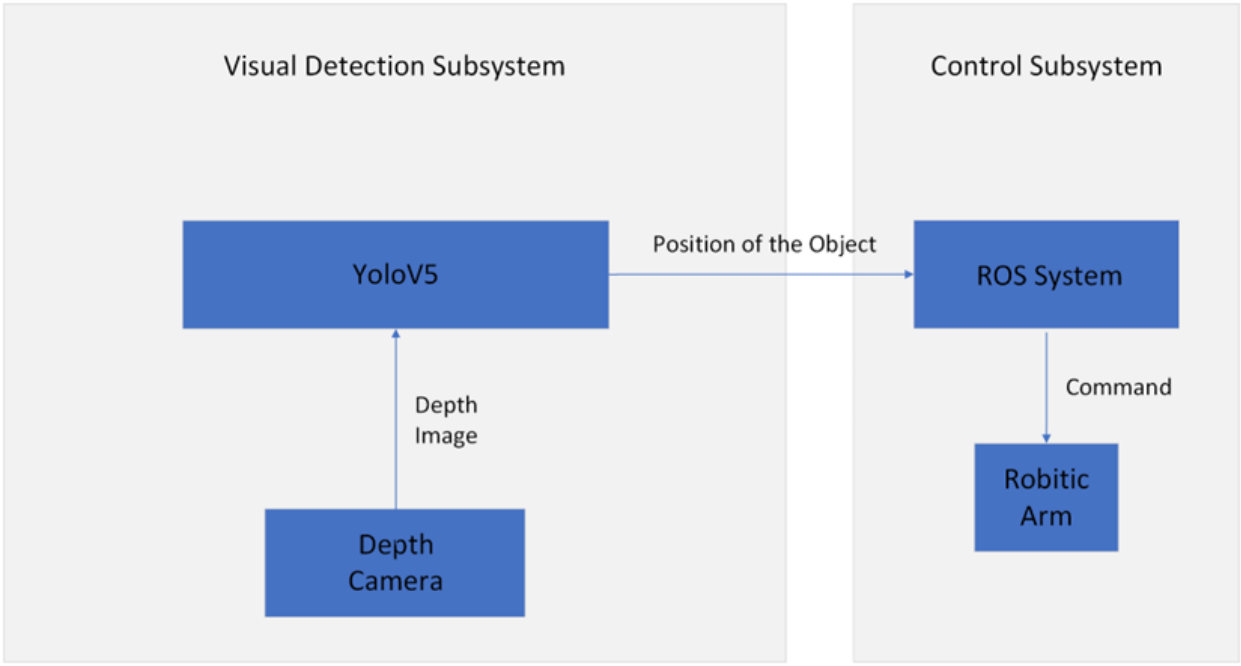


Figure 7: Visual Detection Subsystem

2.4.2 Requirement and Verification

The R&F for Visual Detection Subsystem as below:

Table 2: R&F for Visual Detection Subsystem

Requirement	Verification
1. Ensure accurate data reading from the depth camera. 2. Enable precise detection of the specified object in an image. 3. Utilize the depth camera to consistently detect the desired object in each frame, highlighting it with red bounding boxes, and facilitate access to depth information via the camera's data package. 4. Following depth data retrieval, ensure precise communication of the object's coordinates to the ROS system within the camera's coordinate framework, with seamless conversion to the robotic arm's coordinate system.	A. Verify the accuracy of data reading by analyzing sample data streams from the depth camera and confirming consistency and reliability. B. Conduct tests with various images containing the specified object, ensuring precise detection and minimal false positives. C. Evaluate the system's performance by capturing multiple frames with the desired object present, confirming the consistent detection of the object with accurately placed red bounding boxes. Additionally, validate the accessibility and correctness of depth information provided by the camera's data package. D. Test the system's ability to communicate object coordinates to the ROS system within the camera's coordinate framework by comparing the provided coordinates with ground truth values. Verify the successful conversion of coordinates to the robotic arm's coordinate system and ensure alignment with expected positions.

2.5 Robot Arm Subsystem

2.5.1 Description

ROS

One of the primary purposes of ROS is to facilitate communication between the ROS modules called nodes. These nodes represent the executable code and the code can reside entirely on one computer, or nodes can be distributed between computers or between computers and robots. The advantage of this distributed structure is that each node can control one aspect of a system. One node can capture and output camera images, and another node can control a robot's manipulator in response to the camera view.

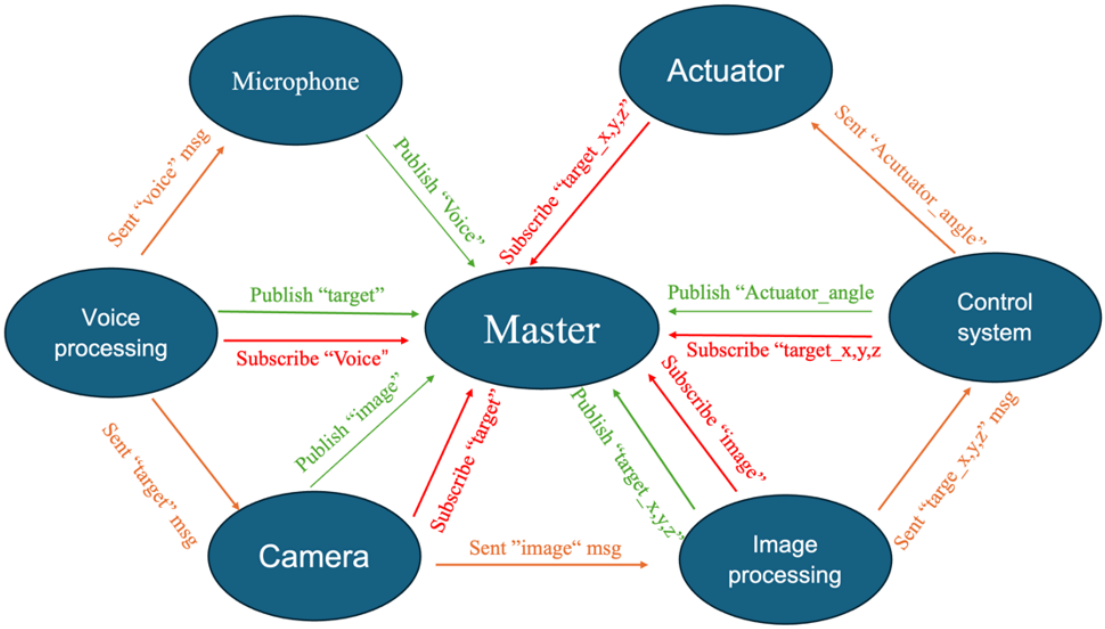
ROS Master and Node

Each ROS system has a centralized coordination mechanism called the ROS Master. Nodes register with the Master to advertise the topics they publish or subscribe to, enabling dy-

namic discovery and connection between nodes at runtime.

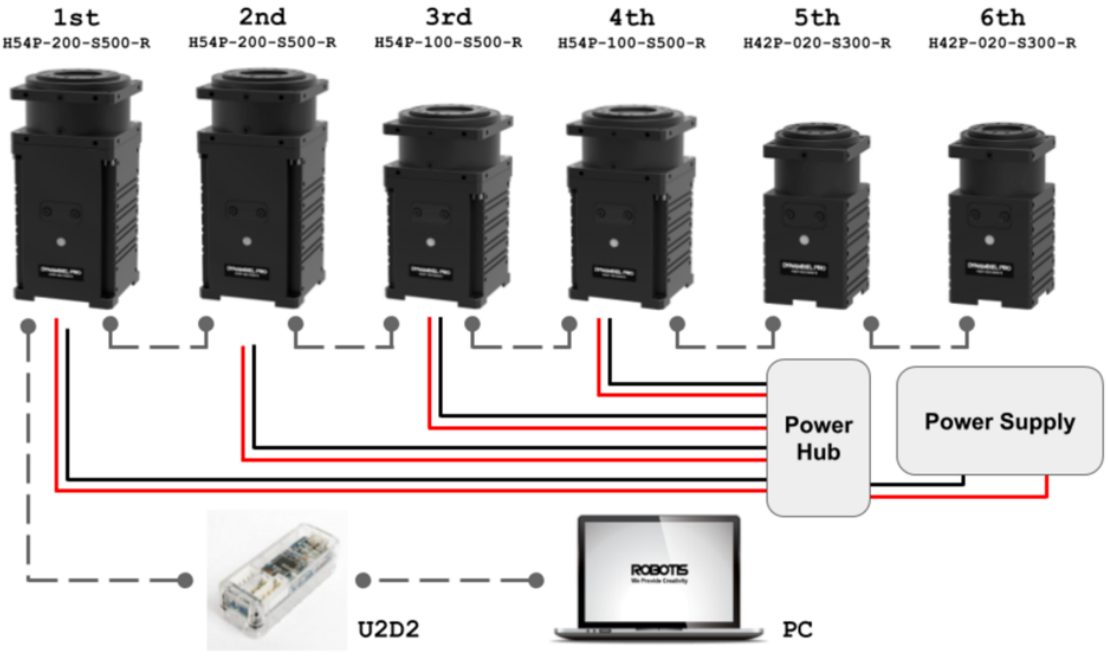
Basically, nodes are processes that perform some action. The nodes themselves are really software modules but with the capability to register with the ROS Master node and communicate with other nodes in the system. Nodes are designed to be independent units of computation. They communicate with each other via ROS topics, services, and actions, allowing for a decentralized and modular architecture. This independence enables parallelism and facilitates scalability in ROS-based systems.

Nodes in our design include microphone, voice processing, camera, image processing, control system, Actuator.



Nodes of our design

To be more specific, "voice" msg is an information in mp3 format capturing the voice of human. The "target" msg is an information in string format with context of specific target that the human point out. "image" msg is a three dimensional matrix of a picture storing the RBG value of each pixel. "target_x,y,z" msg is an information of coordinates of target relative to camera. The "Acutuator_angle" msg is an information of angles of each actuator needed to be executed for the robotics arm to get the target.



Wiring setup

All the programs will be executed in one PC and the result or msg will be sent on each equipment.

Subsystem Control

For control, we will use inverse kinematics. After the image processing get the coordinates and orientation of the target relative to the robotics $(\alpha, \beta, \gamma, x, y, z)$, the transformation will become:

$$ {}^{0}_{E}T = \begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma & x \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma & y \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma & z \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

And,

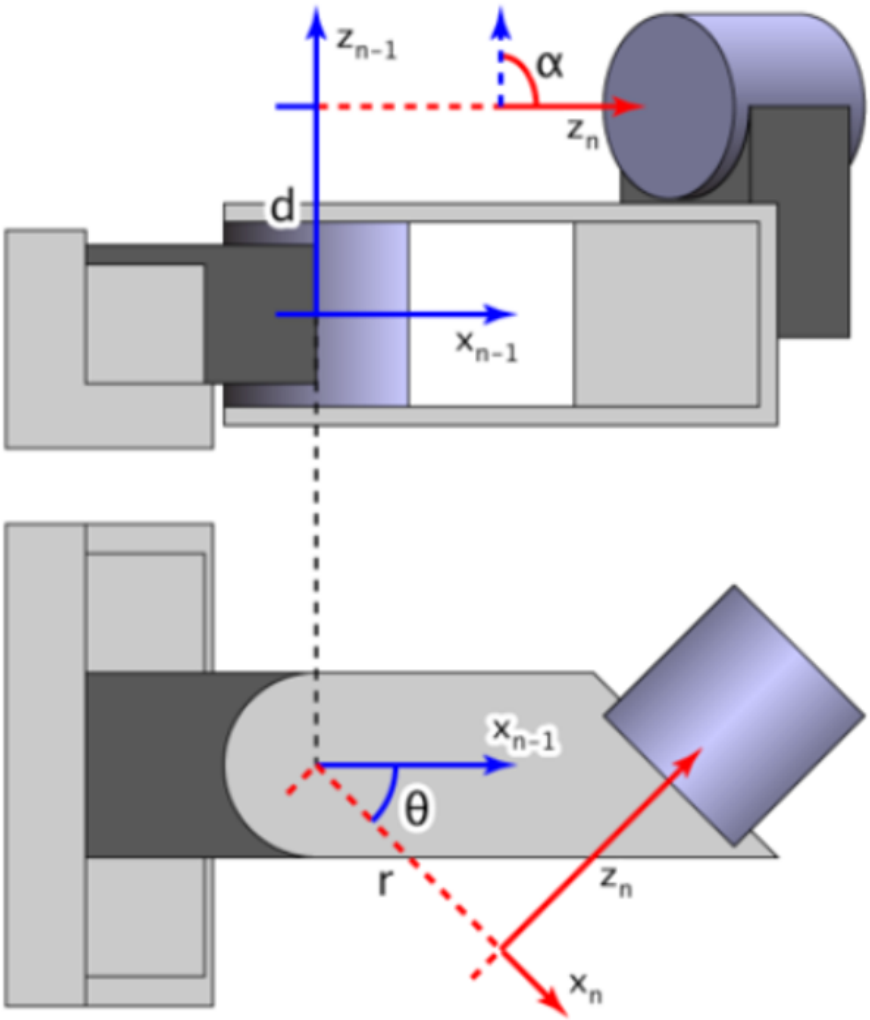
$$ {}^{0}_{E}T = {}^{0}_{1}T\, {}^{1}_{2}T \cdots $$

For transformation ${}^{i-1}_{i}T$, we use a set of D-H parameters to determine which is

$$ {}^{i-1}_{i}T = [Z_{i-1}][X_i] $$

$$ [Z_i] = \begin{bmatrix} \cos\vartheta_i & -\sin\vartheta_i & 0 & 0 \\ \sin\vartheta_i & \cos\vartheta_i & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ [X_i] = \begin{bmatrix} 1 & 0 & 0 & r_{i,i+1} \\ 0 & \cos\alpha_{i,i+1} & -\sin\alpha_{i,i+1} & 0 \\ 0 & \sin\alpha_{i,i+1} & \cos\alpha_{i,i+1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$



D-H configuration

After writing down the D−H parameter, we can identify each α, ϑ by combining the transformation matrix and comparing them to ${}^{0}_{E}T$. Therefore

Those pairs of (α, ϑ) can form a joint space q=[$\dot{q} = \begin{bmatrix} \dot{q_1} \\ \dot{q_2} \\ \vdots \end{bmatrix}$. Each q_i represents the orientation of each actuator. And those will be packed as "Acutuator_angle" msg and sent to actuator.

Furthermore, to evaluate the velocity and acceleration of bodies. The velocity of body i with respect to body j can be represented in frame k by the matrix

$$W_{i,j(k)} = \left[\begin{array}{ccc|c} 0 & -\omega_z & \omega_y & v_x \\ \omega_z & 0 & -\omega_x & v_y \\ -\omega_y & \omega_x & 0 & v_z \\ \hline 0 & 0 & 0 & 0 \end{array}\right]$$

where ω is the angular velocity of body j with respect to body i and all the components are expressed in frame k; v is the velocity of one point of body j with respect to body i (the pole). The pole is the point of j passing through the origin of frame i.
The acceleration matrix can be defined as the sum of the time derivative of the velocity plus the velocity squared

$$H_{i,j(k)} = \dot{W_{i,j(k)}} + W^2_{i,j(k)}$$

The velocity and the acceleration in frame i of a point of body j can be evaluated as

$$\dot{P} = W_{i,j}P$$
$$\ddot{P} = H_{i,j}P$$

And

$$\dot{M_{i,j}} = W_{i,j(k)}M_{i,j}$$
$$\ddot{M_{i,j}} = H_{i,j(k)}M_{i,j}$$

Velocity and acceleration matrices add up according to the following rules

$$W_{i,k} = W_{i,j} + W_{j,k}$$
$$H_{i,k} = H_{i,j} + H_{j,k} + 2W_{i,j}W_{j,k}$$

in other words the absolute velocity is the sum of the parent velocity plus the relative velocity; for the acceleration the Coriolis' term is also present.

The components of velocity and acceleration matrices are expressed in an arbitrary frame k and transform from one frame to another by the following rule

$$W_{(h)} = M_{h,k}W_{(k)}M_{k,h}$$
$$H_{(h)} = M_{h,k}H_{(k)}M_{k,h}$$

2.5.2 Requirement and Verification

The R&F for Robot Arm Subsystem as below:

Table 3: R&F for Robot Arm Subsystem

Requirement	Verification
1.The control subsystem could receive data from sensors and cameras as well as user's voice input correctly in real-time. 2. The robotic arm works normally when data is entered%	A. For safety concern, we firstly run our test cases of robotic control in software simulation. B. The function of the robot arm is tested in the actual scenario

2.6 Tolerance Analysis

2.6.1 Audio Recognition

1.Acurracy about audio acquisition Tolerance analysis for an audio recognition system involves assessing the system's ability to maintain accuracy despite variations in input conditions. The accuracy of the system can be calculated as:

$$Accuracy = N_c/N_t$$

,N_c means number of correctly recognized instances.N_t means number of incorrectly recognized instances.

2.6.2 Visual Detection

1. Inaccuracy in Feature Extraction: Feature extraction can be represented as:

$$F(image) = features$$

, where F denotes the feature extraction function, $image$ is the input image, and $features$ are the extracted features. If the extracted features are inaccurate, it can be represented as the difference between the output of the feature extraction function and the expected features: $L(features, expected\_features)$, where L is the loss function.

2. Variability in Target Transformations: Target transformations can be represented as a transformation function:

$$T(target) = transformed\_target$$

, where $target$ is the original target, and $transformed\_target$ is the transformed target. The variability in targets can be described through the parameter space of the transformation function, such as scale transformation $scale$, rotation angle $rotation$, and translation

$translation$.

3. Background Interference: Background interference can be represented as the error of the background model, i.e., the model's misclassification probability of the background. Binary cross-entropy loss function can be used to measure the error of background classification:

$$L_{background}(predicted, actual) = -(actual \cdot \log(predicted) + (1 - actual) \cdot \log(1 - predicted))$$

4. Data Imbalance: Data imbalance can be quantified through statistical measures of class distribution, such as the ratio of positive to negative samples $class\_imbalance$, defined as:

$$class\_imbalance = \frac{num\_positive\_samples}{num\_negative\_samples}$$

. The loss function can be weighted according to data imbalance, for example, with weighted cross-entropy loss function:

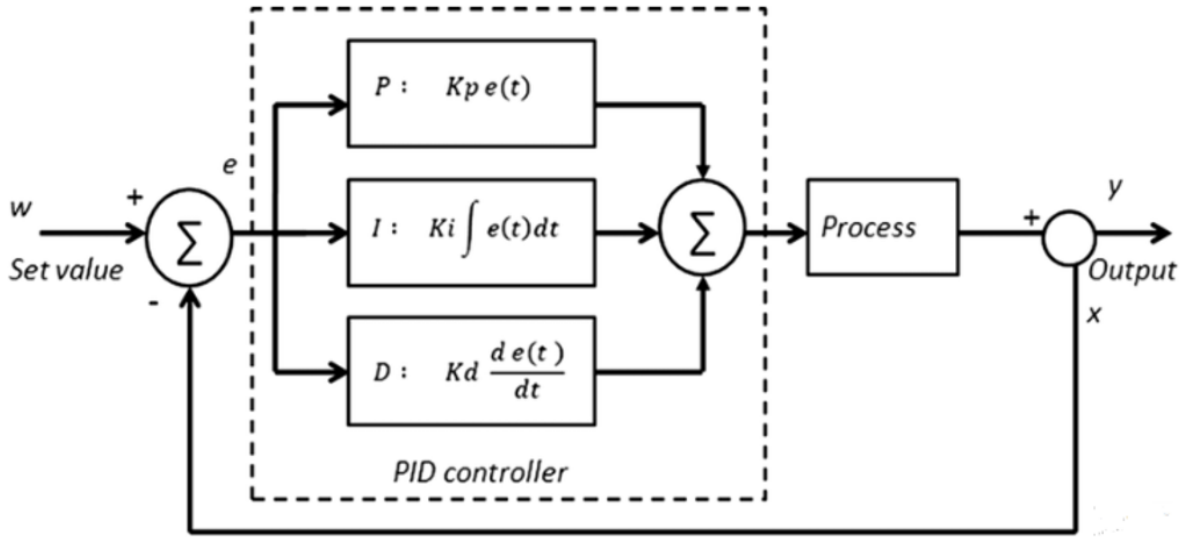
$$L_{weighted}(predicted, actual) = -(actual \cdot \log(predicted) + class\_imbalance \cdot (1 - actual) \cdot \log(1 - predicted))$$

5. Inappropriate Model Structure: The inappropriate model structure can be quantified through the complexity of the model, such as the number of parameters and layers. Model complexity can be defined as $complexity$, and compared with the model's performance. A model with higher complexity may lead to overfitting, while a model with lower complexity may lead to underfitting.

2.6.3 Robot Arm

In the real world, due to the leak capability of the motor, the robotics arm often can't reach the target point. Therefore, we will add a closed loop controller to adjust the position of the robotics arm. Using the camera, we can identify the output position of the robotics arm. This output position is a space vector and will be subtracted from the set value to form an error vector e_t. And this will be sent to a PID controller and the result will be sent to the process to produce a joint space $\dot{q} = \begin{bmatrix} \dot{q_1} \\ \dot{q_2} \\ \vdots \end{bmatrix}$. By using the Jacobians matrix of the robotics arm ${}^0J(\theta)$, we can calculate the relate joint velocities for the robotics arm to adjust its position.

$$^0v =^0 J(\theta)\dot{q}$$



Closed loop controller

3 Cost and Schedule

3.1 Cost Analysis

3.1.1 Labor

In this project, the labor cost is calculated by the total number of hours to complete the task and the hourly wage per team member. With reference to ECE industry average hourly wage, we see that the labor cost per team member per hour is determined to be 100 RMB. At the same time, considering that we have about 8 weeks to complete the project and we plan to We plan to spend 20 hours a week on the project, we can calculate:

$$20hr/week \times 8weeks \times 100RMB/hr = 16000RMB$$

Therefore, the labor cost per team member is 16000 RMB.

3.1.2 Parts

The parts list and estimated costs are shown in table 4.

Table 4: Parts List and Estimated Costs

Description	Manufacturer	Part Number	Quantity	Cost
Robotic Arm	Robotis	OpenMANIPULATOR-P	1	97000 RMB
Serial Module: Steering Engine Controller	Robotis	U2D2	1	500 RMB
3D Camera	Intel	RealSense D435i	1	2100 RMB
Notebook	Xitong	954	4	40RMB
Student power supply			1	150RMB

3.1.3 Total Cost

According to the above statistics, we know that the labor cost of our project is 16000RMB and the cost of Parts is 99790RMB. Therefore, the total cost of our project is 115,790RMB.

3.2 Schedule

3.2.1 Timeline

March:

1. Complete the configuration of the robotic arm environment and download pre-training models.
2. Accomplish basic manipulator motion control.

April:

1. Procure all sensors, devices, and control boards required for the project. Assemble and connect all components.
2. Investigate the YOLOv5 Detection model and integrate it into the system.
3. Finalize the audio model.
4. Complete the preliminary assembly of each module.

May:

1. Finalize the overall design of the robot arm function.
2. Conduct tests for assigned tasks.
3. Prepare project-related reports.

3.2.2 Per-person

Our personal weekly schedule is shown as table 5.

Table 5: Weekly Schedule

Week Number	Haoran Yang	Yipu Liao	Yiming Li	Chenghan Li
3.25-3.31	Inital set up of robotic arm	Inital set up of robotic arm	Run the yolov5 model	Run the yolov5 model
4.1-4.7	Install the manipulator motion algorithm library	Install the manipulator motion algorithm library	Install the Realsense camera driver	yolov5 was used to detect the images collected by the camera
4.7-4.14	Fix previous problems	Fix previous problems	Fix previous problems	Fix previous problems
4.14-4.21	Design and print the gripper	Design and print the gripper	Test the vision system	Implement audio system
4.21-4.28	Complete the control system and preliminary demo	Complete the control system and preliminary demo	Integrated voice system and visual system	Integrated voice system and visual system
4.28-5.5	Test mechanical reliability test	Do real safety and functionality test	Test visual-audio reliability test	Test visual-audio reliability test
5.5-5.12	Finish the whole project and prepare the demo	Finish the whole project and prepare the demo	Finish the whole project and prepare the demo	Finish the whole project and prepare the demo
5.12-5.19	Improve thr functionality and write final report	Improve the functionality and write final report	Improve the functionality and write final report	Improve the functionality and write final report

4 Discussion of Ethics and Safety:

4.1 Ethics

Developing a project involving the use of robotic arms to provide assistance to Parkinson's patients raises ethical and safety concerns. Ethical issues to consider include privacy, the right to know, and potential harm to customers.
The IEEE Code [2] of Ethics emphasizes the importance of the safety and welfare of the public. Therefore, the project should prioritize the safety and comfort of the human clients. The ACM Code of Ethics [3] emphasizes the importance of avoiding harm, and the project should strive to avoid any potential harm to the clients.To avoid ethical violations, project teams should obtain informed consent from clients before providing services to them.
In the development process of robots, the project team should pay close attention to their potential interference with patients' normal lives, especially considering the functional limitations that may arise in certain situations. This means not only ensuring that the design and functionality of the robots can integrate seamlessly into patients' daily lives but also carefully considering the potential constraints that may arise in specific circumstances. Such attention not only contributes to ensuring the effectiveness and practicality of the robots but also minimizes the disruption and distress they may cause to patients' lives.
Furthermore, we should also pay particular attention to the issues that may arise from robots handling sensitive information and personal privacy. This includes but is not limited to how robots collect, store, and use personal data, as well as ensuring that this information is not accessed or abused without authorization. In the design and development process, strict adherence to best practices in privacy protection is essential, while also considering various potential threats and risks to ensure that users' data security and privacy are adequately protected.

4.2 Safety

According to the Occupational Safety and Health Administration (OSHA) safety standards, in section 1910.212, it is stated that all machinery should meet general requirements. Specifically, it is mandated that robotic systems operate in compliance with safety regulations. In this project, the primary concern revolves around the pressure exerted by the mechanical arm and the pressure applied by the massage head on the human body. This issue persists across all stages of our project, from design and testing to the final demonstration. To elaborate, we have the following safety concerns:
1.The mechanical arm must possess sufficient precision and sensitivity to ensure that no unintended harm is inflicted upon patients during task execution. Additionally, the design of the mechanical arm must take into account the physical condition and mobility of the patients to ensure that its operation does not exert additional pressure or discomfort.
2.The safety of the mechanical arm must also consider its contact with the patients'

bodies, necessitating the implementation of appropriate protective measures to prevent injury from accidental contact. Furthermore, the control system of the mechanical arm should feature reliable safety functionalities, such as emergency stop buttons and limit protections, to address potential unforeseen circumstances and ensure patient safety under all conditions.

For addressing safety concerns regarding the mechanical arm assisting Parkinson's patients, the following solutions can be implemented:
1.Precision and Sensitivity: Ensure that the mechanical arm possesses high precision and sensitivity, which can be achieved by using high-quality sensors and precise control systems. Additionally, thorough testing and calibration should be conducted to ensure that the mechanical arm can accurately recognize and respond to the patients' movements during task execution.
2.Consideration of Patients' Physical Condition and Mobility: When designing the mechanical arm, it's essential to fully consider the patients' physical condition and mobility to ensure that its operation does not exert additional pressure or discomfort on them. This can be achieved through customized design or adjustable operating parameters.
3.Implementation of Safety Measures: The design of the mechanical arm should account for its contact with the patients' bodies and incorporate appropriate protective measures, such as soft protective covers or safety triggers, to prevent injuries from accidental contact.
4.Integration of Safety Features into the Control System: The control system of the mechanical arm should be equipped with reliable safety features, including emergency stop buttons, limit protections, and collision detection, to address potential unforeseen circumstances. Additionally, training and guidance should be provided to ensure that operators know how to respond correctly to emergency situations.

References

[1] Baidu. ""Parkinson's Disease"." (2020), [Online]. Available: https://baike.baidu.com/item/%E5%B8%95%E9%87%91%E6%A3%AE%E7%97%85/945855 (visited on 03/24/2024).

[2] IEEE. ""IEEE Code of Ethics"." (2016), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 03/24/2024).

[3] ACM. ""ACM Code of Ethics"." (2018), [Online]. Available: https://www.acm.org/code-of-ethics (visited on 03/24/2024).

5 Appendix a:

5.1 Audio Recognition Subsystem

Subsequently, the captured audio signals are transmitted to the audio processing module. At this stage, the audio sequence is segmented into different segments, each of which undergoes noise reduction to eliminate background noise interference. Additionally, the audio processing module performs format conversion, transforming audio signals into a digital format recognizable by computers for subsequent processing and analysis.
Then, the processed audio data is passed to the model recognition module. In this module, the system imports pre-trained audio recognition models, with plans to employ a CNN-LSTM model at the current stage, capable of accurately recognizing different audio signals. The model recognition module analyzes and matches the input audio signals, then outputs corresponding text or commands, thus achieving audio recognition functionality.

5.1.1 Install audio drive

:We plan to utilize the speech recognition package-pocketsphinx integrated within ROS and incorporate it into our system. Through this package, we can leverage the functionalities and nodes provided by the ROS framework to achieve speech recognition capabilities. We will connect a high-quality microphone to the system to capture user speech input. Once the speech signal is captured, it will be sent to ROS nodes for speech recognition using the pocketsphinx package. Pocketsphinx is an open-source, real-time, and accurate continuous speech recognition engine with low computational resource requirements, making it suitable for embedded systems and embedded computing devices. Once the speech is recognized and converted into text, the system will be able to execute corresponding actions based on user commands or requests.

5.1.2 CNN-LSTM:

Convolutional neural networks (CNNS) are used in CNN-LSTM to extract features from each fragment, helping capture key information in the speech signal. These features are then fed into the bidirectional Long Short-term Memory Network (BiLSTM) for further processing. BiLSTM effectively models long-term dependencies in sequence data to better understand contextual information in speech signals. Finally, we use the output of BiLSTM as a target vector for subsequent speech recognition or other related tasks. This end-to-end process makes full use of the advantages of cnn and BiLSTM to achieve efficient processing and accurate recognition of speech signals.

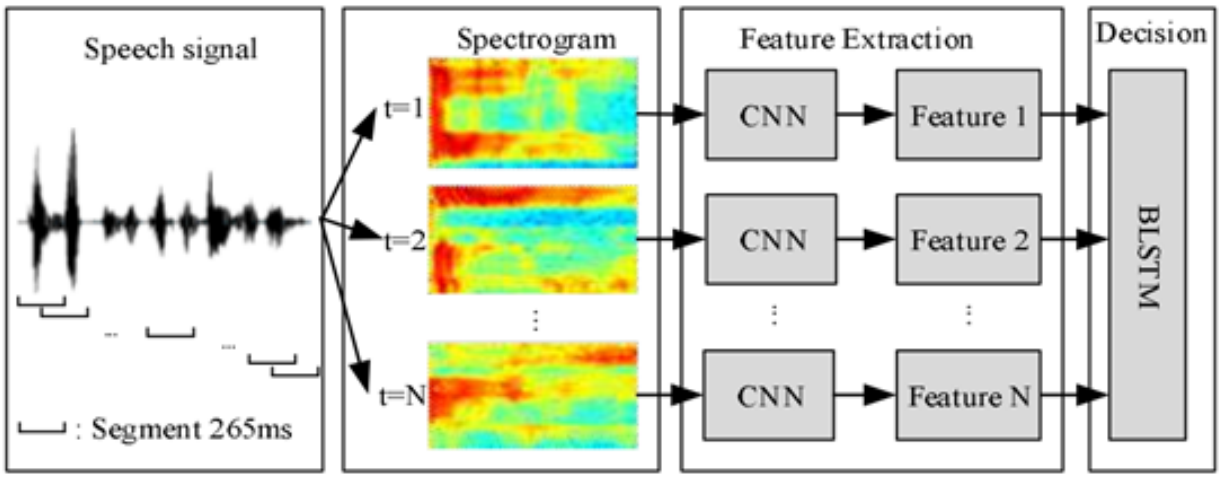


Figure 8: CNN-LSTM Algorithm Schematic Diagram

Finally, the output module represents the ultimate stage of the entire system. Here, the system integrates the audio recognition results with those of the visual model to achieve more comprehensive information output. For instance, by analyzing the recognized audio content and identified object categories, the system can generate outputs with greater informational value, such as instructing users to locate a specific object in the image or perform particular actions.

5.2 Visual Subsystem

5.2.1 System Input and Output

Input: The class of the wanted object.
Output: The position of the wanted object in the real world coordinate.

5.2.2 Deploy YoloV5 on ROS System

1. YoloV5 requirements:
$gitpython \geq 3.1.30 matplotlib \geq 3.3$
$numpy \geq 1.23.5$
$opencv - python \geq 4.1.1$
$Pillow \geq 9.4.0$
$psutil \# systemresources$
$PyYAML \geq 5.3.1$
$requests \geq 2.23.0$
$scipy \geq 1.4.1$
$thop \geq 0.1.1 \qquad \# FLOPscomputation$

$torch \geq 1.8.0 \qquad \#seehttps://pytorch.org/get-started/locally(recommended)$
$torchvision \geq 0.9.0$
$tqdm \geq 4.64.0$
$ultralytics \geq 8.0.232$
$pandas \geq 1.1.4$
$seaborn \geq 0.11.0$
$wheel \geq 0.38.0 \qquad \#notdirectlyrequired, pinnedbySnyktoavoidavulnerability$

2. Configure Python Environment:
First, download the Anaconda Linux version. Then, add the correct environment variables in the .bashrc file. After successfully adding the Python interpreter, create the necessary virtual environment and install the required libraries according to the above requirements. Finally, choose the appropriate PyTorch version based on the CUDA version. Next, download the required YOLOv5 pre-trained models.

5.2.3 Framework of YoloV5

Suitability: YOLOv5 is a deep learning-based object detection framework developed by Ultralytics. Compared to its predecessors, YOLOv5 adopts a simplified network architecture that achieves faster inference speeds while maintaining high performance. Its key features include:
•Fast and accurate object detection: YOLOv5 employs an efficient object detection algorithm that achieves rapid object detection while maintaining high accuracy.
•Model lightweighting: Compared to previous versions, YOLOv5 has a simpler model structure with fewer parameters, making it easier to deploy on mobile and embedded devices.
•Simple and user-friendly interface: YOLOv5 provides a simple and user-friendly Python interface, allowing users to easily perform object detection tasks and seamlessly integrate the model into their projects.
In summary, YOLOv5 is an efficient, accurate, and lightweight object detection framework suitable for various real-time object detection tasks.

Structure of Yolov5:

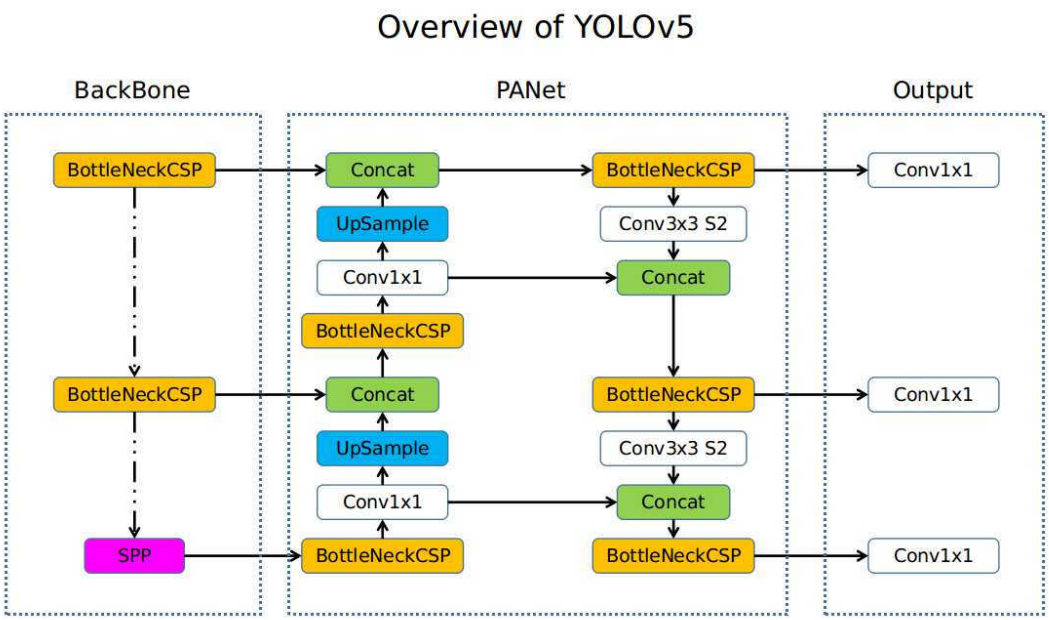


Figure 9: YOLOv5 Overview

The YOLOv5 network architecture diagram above shows four main components: Input, Backbone, Neck, and Prediction.

•Input: Includes Mosaic data augmentation, adaptive anchor calculation, and adaptive image scaling.

Yolov5 adopts the same Mosaic data augmentation approach as Yolov4 for input preprocessing. It involves random scaling, random cropping, and random arrangement to concatenate images. This method yields good detection results, especially for small objects.

In the YOLO algorithm, anchor boxes with initial predefined dimensions are set for different datasets. During network training, the network adjusts these initial anchor boxes based on the predicted boxes, compares them with the ground truth boxes, calculates the differences, and updates the network parameters through backpropagation.

In commonly used object detection algorithms, images have different dimensions, so a common approach is to resize all images to a standard size before feeding them into the detection network. However, in practical applications, many images have varying aspect ratios. Therefore, after resizing and padding, the size of the black borders on both ends differs. If too much padding is applied, it results in redundant information and affects inference speed.

Therefore, in the Yolov5 code, modifications have been made in the 'letterbox' function in 'datasets.py' to add the minimal amount of black border adaptively to the original image.

•Backbone: Incorporates the Focus structure and CSP structure.

The Focus structure, not present in YOLOv3 and YOLOv4, involves a key operation called slicing. Taking the architecture of YOLOv5s as an example, the original 608x608x3 image input is passed through the Focus structure. Through slicing operations, it is initially transformed into a 304x304x12 feature map. Then, after a convolution operation with 32 convolutional kernels, it ultimately becomes a 304x304x32 feature map.
In YOLOv5, two types of CSP structures are designed. In the YOLOv5s network, the CSP1_X structure is applied to the backbone main network, while the other CSP2_X structure is applied to the Neck.

•Neck: Utilizes the FPN+PAN structure.
In YOLOv4, the Neck structure uses regular convolution operations. However, in YOLOv5's Neck structure, the CSP2 structure inspired by CSPnet design is adopted to enhance the network's feature fusion capability.

•Prediction: Employs GIOU Loss.
In YOLOv5, the Bounding Box Loss function utilizes CIOU_Loss. During post-processing in object detection, many bounding boxes need to be filtered, typically requiring non-maximum suppression (NMS) operations. Since CIOU_Loss includes an influence factor "v" that involves ground truth information, and during testing and inference, ground truth is not available. Therefore, YOLOv4 employs DIOU_nms based on DIOU_Loss, while YOLOv5 uses a weighted NMS approach.

Dataset Used for Pretraining: We use COCO dataset as our train and test dataset. The COCO dataset is a comprehensive collection used for object detection, segmentation, and captioning tasks. It is designed for scene understanding and is extracted from diverse and complex everyday scenes. Objects in images are precisely delineated through segmentation, enabling accurate positioning within the scene. The dataset comprises 91 object categories and includes 328,000 images with 2,500,000 labeled instances. As of now, it stands as the largest dataset for semantic segmentation, featuring 80 classes and over 330,000 images, with annotations available for 200,000 of them. The dataset encompasses more than 1.5 million object instances across various categories, providing rich and extensive data for object detection challenges. With an average of 7.2 objects per image, COCO poses a significant challenge for object detection tasks and remains one of the most widely used databases in the field.

5.2.4 Install the Driver of Depth Camera on ROS

The RealSense D435 camera provides a global shutter sensor and a larger lens to achieve better low-light performance compared to the cheaper D415 camera. Additionally, the D435 features a more powerful RealSense module, the D430.

This RealSense camera series can capture distances up to 10 meters and can be used out-

doors in sunlight. They support outputting depth images at a resolution of 1280x720, and for regular video transmission, they can achieve up to 90fps.

Figure 10: D435 Camera

The front of the D435 camera has four holes. From left to right, the first and third are infrared sensors (IR Stereo Cameras); the second is an infrared laser projector (IR Projector), and the fourth is a color camera (Color Sensor).

5.2.5 Detect the Depth of the Object

The frame captured by the RealSense camera is a high-dimensional tensor containing depth information. Upon receiving this high-dimensional data, the system should first extract a low-dimensional plane image containing only depth information from this tensor. This image can be mathematically represented as a three-dimensional tensor of width × height × number of channels. This image is then inputted into YOLOv5 for prediction, allowing it to identify the positions of the desired objects. The information of these position boxes is recorded, and pixels belonging to the object are identified in the image while others are disregarded. Subsequently, the depth information of these pixels is extracted from the original data packet received initially. This process allows for the prediction and estimation of the object's depth information, resulting in obtaining the object's three-dimensional coordinates in the camera coordinate system. By performing coordinate transformations, the object's three-dimensional coordinates in the real world can be

obtained.