Machine Vision Robot Control

Robot frame control through human positioning.

Quinn Leydon
RIT Electrical Engineering Department
RIT
Rochester, USA
Qcl4604@g.rit.edu

***Abstract*— Google's MediaPipe library was used to determine the x, y, and z components of key features of a human hand. These features were used to define the position and orientation of a reference frame using a series of mathematical equations. This frame was implemented by Baxter, a 7 degree of freedom (DOF) robot developed by rethink robotics. The joint angles where set using an inverse kinematics function.**

I. Goal

There is currently a major push for humanoid robots to assist with disaster relief situations. These robots have many moving components and often require a skilled worker for operation. The intend of this project is to use human positioning to control a humanoid robot. The position and orientation of the end effector on a humanoid robot will mirror an operator's hand movements. Machine vision is used to determine a relative position and orientation of a human hand and convert that to position and Euler angles. These are converted to an end effector position and orientation of the Baxter robot.

II. Related Works

Object detection is a cutting-edge industry within electrical engineering. In [1] object detection is used to recognize components within an automotive electrical box. An SVM algorithm is implemented to detect and classify different electrical components within the box using a 14-million-pixel GigE interface HD IP Camera. The system eliminates the influence of human factors using machine vision [1]. [3] takes machine vision and applies it to robotic systems. A Kinetic V2 distance sensor is used to detect objects and react based on the depth to the object. In this application, if a Lenna photograph is placed in front of the 6 degree of freedom robot, it will react by grabbing the photograph [3].

Machine Vision is also used in higher risk situations. [2] uses machine vision to detect and react to a real time physiological situation, or when a pilot has a negative reaction to oxygen content, G-forces, or other conditions during intense flight maneuvers. This study used FiO2, SpO2, and arterial CO2 state, heart rate, breath rate, cabin pressurization, breathing gas pressure, and acceleration forces as well as IDS NXT Rio camera to detect the operator's status [2].

A common Machine vision software is OpenCV MediaPipe by Google. This is a software which detects key body joint position or facial landmarks in real time. [4] used OpenCV MediaPipe for hand tracking in an Indonesian national defense museum. To make the museum more interactive the movement through the virtual museum was controlled through hand gestures. This brought in additional attention to the museum. It was found to be effective, however the notebook laptops common in the country did not have enough resources to use the hand control software [4].

Sign interpretation is major topic for machine vision around the world. [5] Used media pipe software to detect and interpret sign language. The x, y, and z positions of the key hand features where recorded. The software interprets the z position from the x and y through a pre trained machine learning algorithm. An RNN algorithm was run on the results to determine the sign position in Tai sign language (TSL) [5]. [8] Takes this a step farther and uses sign language to control a humanoid robot. This robot uses both the hand and body detection of Media pipe. The robot detects a person and moves its head towards them. After interpreting the Argentinian sign language or body gesture, it replies with a preset reaction [8]. This is the beginning of a robotic translator.

Humanoid robots are a growing field thanks to the DARPA Robot Challenge or DRC which focuses on robotic disaster relief, such as the Fukushima reactor failure [8]. The CENTURO robot designed by Istituto Italiano di Tecnologia (IIT) was developed to further research into disaster management. This four legged, 39 DOF "Centaur" humanoid robot can move 120 kg over a surface with a friction coefficient of 0.5 [7].

Humanoid robots are also being developed to work in conditions that are dirty dangerous, and damaging situation for humans [6]. The HRP-5P robot is a humanoid robot capable of working potentially damaging construction jobs. The robot can use machine learning to locate drywall gypsum boards, pick up the board, transfer it to the required location, and install the board [6]. This is an early step to a fully automated construction worker.

Research into an upper limb prosthesis is currently in research. The ideal prosthesis modular, human looking, lightweight silent, and dexterous [9]. Myocontrol is currently being studied to control these upper limb prostheses. Myocontrol swiftly, naturally, and reliably converts bio signals in a anon invasive manner from the upper-limb disable subject. This requires a machine learning algorithm to be trained for each movement per user. The user must perform enough tests for each movement to train the model. This process was conducted using a Myo bracelet consisting of eight uniformly spaced sensors, able to detect electromyographic signals generated by muscle activities. In the test phase intact subjects used the wristband to guide a virtual hand through 36 tasks [9].

3d body measurements is important and has application in medicine, surveying, fashion, fitness, and entertainment [10]. 3d surfaces scanning technologies are faster and more convenient that traditional measuring methods. Passive stereo is a measuring technique using multiple camera views. Stereo reconstruction uses two horizontally or vertically aligned RGB cameras, where triangulation is used to find depth. Active stereo is achieved applying structured light, or a light pattern to the subject. This allows the projector to scan the scene. The resulting figure can be identified through key points or silhouette [10].

III. Implementation

The Baxter was developed by Rethink Robotics. Baxter is a humanoid robot with two 7 Degree of Freedom arms and a 2 DOF head. Baxter is 3’1” and 165 lbs. without its pedestal. Baxter has a predeveloped inverse kinematics library which positions the end effector based on position and quaternion data. Baxter in the Gazebo simulation is shown in figure 1. Gazebo is a simulation software used in development. Baxter is a collaborative robot, meaning there are torque sensors in each of the arms. If the robot was to hit something or someone the joint would be shutdown. This allows humans to be within the workspace without risk of significant bodily harm.

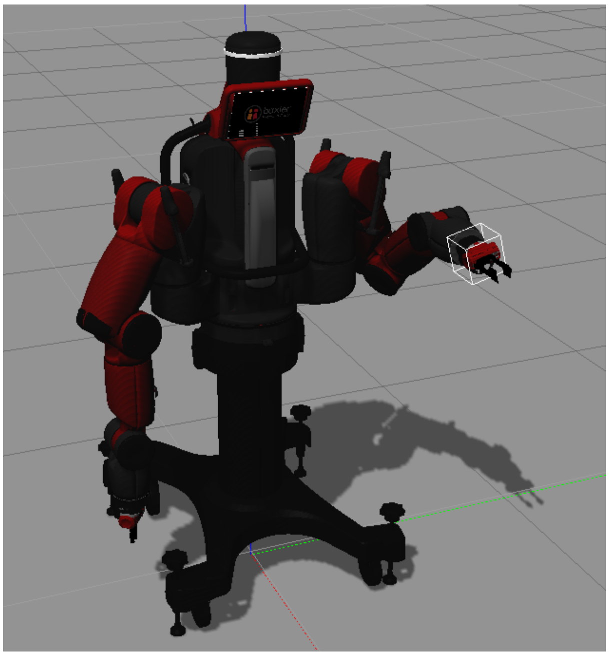


Figure 1: Baxter Robot

MediaPipe was used to detect key features in the hand. MediaPipe is an OpenCV library created by google which is a pretrained machine learning model. Holistic control combines the use of hands pose and Face mesh, however this uses more computer resources, so the project used exclusively hand features. Media pipe Hands is a high-fidelity hand and finger tracking solution which employs machine learning to infer 21 3D landmarks, so this library was used. MediaPipe is superior to other joint inference engines due to its ability to approximate a z location based on previous measurements of the hand. The joint positions are shown in figure 2.

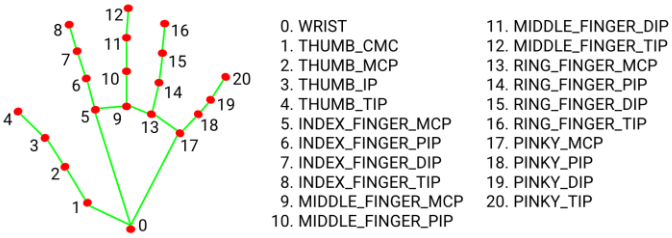


Figure 2: Key Joint Positions.

The program developed can use a tradition camera such as a web camera, or the RealSense depth camera by Intel. The RealSense is a camera which uses a distance sensor to assign a depth measurement to each pixel of the camera. During experimentation with the intel depth camera the depth was inconsistent.

Finding edge detection of a hand is difficult due to the lack of contrast and occlusion depending on the orientation of the hand. The lower quality of the RGB camera on the real sense resulted in poor edge detection by MediaPipe, thus the webcam was used during experimentation.

Once the key components of the hand are determined by MediaPipe, a mathematical model must be created to represent the position and orientation of the hand. Each joint on a robot has its position and orientation defined by a reference frame. Figure 3 shows a MATLAB script which applies a 90-degree roll to a frame.

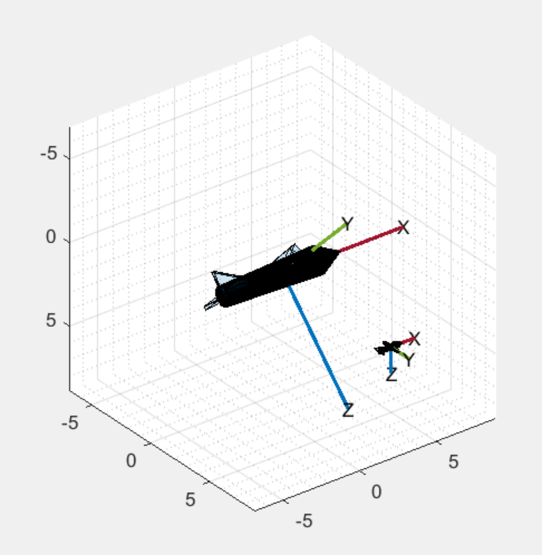


Figure 3: Frame with 90 Degree Roll.

In robotics, the coordinate system is defined in n, o, and a vector when refereeing to the frame relative to the joint. To determine the relative orientation of the human hand a mathematical model of the n, o, a vector must be found. These are the x, y, z relative to the joints frame of reference, rather than a universal frame of reference. First, V1 is defined as the vector from the P17, knuckle on the pinky to P5, the knuckle on the index finger. V2 is defined as the vector from the knuckle on the pinky to P0, the base of the palm. The a vector is defined as the cross product of V1 and V2. This produces a vector orthogonal to the two vectors. The a vector is then normalized by dividing the x, y, and z component by the magnitude of the vector. The o vector is defined as V1 normalized. The n vector is the cross product of o and a. This results in a vector orthogonal to both vectors in a similar direction to V2. This is shown in figure 4.

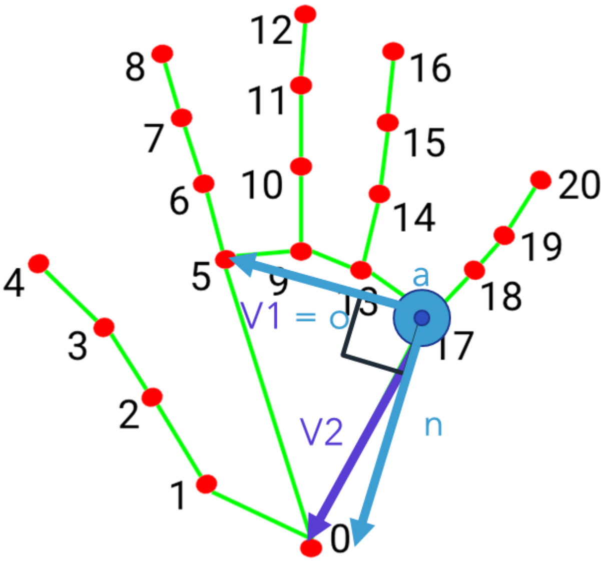


Figure 4: Euler representation of hand frame

Traditionally roll is the rotation about n, pitch is the rotation around o, and yaw is the rotation around a. For this implementation it is intuitive to have roll as the rotation about a, pitch as the rotation around o, and yaw as the rotation around n. The SciPy spatial transformation library used set the yaw value as the yaw minus the pitch. By adding the pitch to the yaw this was canceled out. To use this coordinate system on Baxter, a 90-degree offset was added to the pitch.

The coordinate system for Baxter is not consistent with the coordinate system for the camera. The x value for Baxter is the z for the camera. This is set to a constant value when working with a traditional camera. The y axis for Baxter is the x axis of the camera. The z vector for Baxter is the inverse of the y axis for the camera. When using the left arm, a plus one offset is used.

The inverse kinematics script was unable to impact roll pitch and yaw. Yaw was treated as negative roll in the tf library available in python 2. Sending RPY data to the robot results in a change in roll and pitch, but not yaw. By sending quaternion data to the IK function results in a change in yaw and pitch, but not roll. Because of this limitation, the user must determine if they would like roll or pitch.

Media Pipe Hands is compatible with Android, IOS, C++, Python, and JavaScript, but not Linux. Baxter runs on python 2 on an Ubuntu Linux system. For this reason, the MediaPipe is run on an external windows computer. The messages are transferred to the Baxter computer using UDP messaging in python. A server is run on the Media Pipe program, which sends messages to the Linux based client which connects with it.

Baxter uses ROS or Robot Operating System. The main goal of ROS is to easily add and communicate between systems in a robotic system. A system publishes data of a specific data type to a ROS topic. A different system can subscribe to that topic and receive the data that was published.

The UDP_Publisher python script receives the UDP message and packages it into a pose message. A pose message has a x,y,z positions, and x,y,z,w orientation components for the quaternion message type. This is then published to a "/my_Pos" topic, which will be received by the Baxter code.

The flowchart of the Baxter IK program can be found in the appendix. First the input type is chosen. This allows a choice for position and angles to be set by a constant value or set by a the MediaPipe program. The RPY type is set to chose between roll or yaw. This must be consistent with the MediaPipe program. Finally, the right or left limb is chosen. The Subscriber is then activated to listen to the my_Pose topic. This then runs an asynchronous callback function which converts the data to an array and then sleeps for 1s. The robot is then enabled and initialized.

The robot then loops while not in shutdown. First, the pose data is received from the callback function. The RPY and Quat variables are set based on the data. The main poses object is populated based on the user's selections. This sets the position and orientation of pose. The required joint angle for the given pose is found through the IK function. These joint angles are then applied to the robot. The joints move within a set tolerance to the robot. 0.1 radians or 5.729 degrees allows for quick motion with acceptable error.

IV. Results

The MediaPipe library was able to determine the positions of the key joints in the hand. Figure 5 shows the robot imitating the operator's motion. The robot is in the setting such that the position and orientation are defined by the user. The orientation is set to account for roll and not pitch.

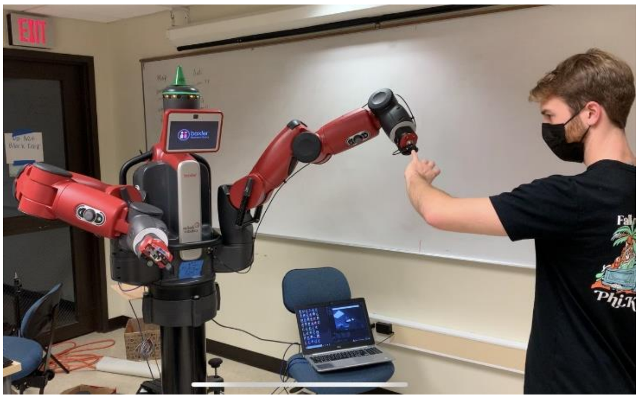


Figure 5: Baxter Mirroring Human Position.

The precision of the library varied with the background. The lab used had white walls and many bright overhead lights. This reduced the contrast on the hand and resulted in less precise detections of the hand frame compared to other environments. The z position from the library was subject to drift. The pitch of the hand pointed perpendicular to the camera frame was not consistently 90 degrees. Overtime this would vary between 80 and 110 degrees as library continued to estimate the dimensions and orientation of the hand.

The RGB camera on the RealSense was less effective than the web cam. This resulted in inconsistent detections of the hand and not reliable for experimentation. The depth returned from the camera was accurate within a specific range but was not consistent outside of that range.

The Euler angles of the hand were found in the MediaPipe program and displayed. It was found that yaw was inconsistent between 40 and -40 degrees. At shallow angles a small change in the z component of either knuckle will result in a large change in angle. For this reason, any yaw between -40 and 40 was set to zero.

The messages where successfully sent to the Linux machine at a high rate. It was found that if poses where published to the my_Pose topic at a high rate, the messages would back up causing a delay. Overtime this resulted in a large delay between action from the user and response from the robot. By limiting the rate at which the messages are published to 10 messages per second.

The Inverse Kinematic function of the robot was unable to a set position considering roll, pitch, and yaw. By setting the RPY type, roll and pitch or pitch and yaw where successfully implemented.

The Baxter robot has a lower range of motion than ideal. This means there are many positions within the workspace that Baxter is unable to reach given set of joint angles. The robot is also slow to move to position further decreasing its usage. Because of the slow movement, the robot was able to move between set points but was unable to mimic smooth motion. With a set z component for position, the robot moved its arm in a 2d plane based off the movements of the user's hand.

V. Future Works

A major limitation to the Baxter robot is its inverse kinematics function. This function is incapable of adjusting for roll, pitch, and yaw together. This function does not check if the determined joint angles are acceptable for the Baxter robot. As a 7 DOF robot Baxter is capable of multiple joint angles for a given position. If a found solution is not acceptable, other solutions should be tried. This would improve the speed and overall performance of the robot.

The position coordinates of the hand could be found in a more precise manner. The RGB camera on the RealSense 435 was not consistent enough to be used on this application. The higher level D455 or depth camera from another manufacturer may be capable directly determining the z components of the hand. If multiple traditional cameras are used, the z component could be identified using stereo vision or other image processing techniques.

This process could be implemented on another robot. A robot with quicker movements would have a more realistic motion. A robot with a better range of motion would also be ideal. Once implemented on an improved setup, this procedure could be used in a real setting. This would be ideal for complex motions in a disaster setting. This could also be used for custom parts manufacturing. Eventually, this could be used in surgery where the movements of a surgeon's hands could be scaled for more precise surgeries.

VI. Acknowledgment

Thank you to Dr. Sahin, and Nikhil Deshmukh for all their help and input on this project. Without them this project would not have had the same level of success

VII. References

[1] L. Zhang, D. Pang and P. Ma, "Character recognition for automotive electrical box components based on Machine vision," 2018 International Conference on Advanced Mechatronic Systems (ICAMechS), 2018, pp. 117-121, doi: 10.1109/ICAMechS.2018.8506926.

[2] D. Fries, J. Phillips, M. McInnis and C. Tate, "Rapid Multi-Sensor Fusion and Integration Using AI-enabled Machine Vision for Real-Time Operator Physiological Status," 2021 IEEE Research and Applications of Photonics in Defense Conference (RAPID), 2021, pp. 1-2, doi: 10.1109/RAPID51799.2021.9521454.

[3] P. -Y. Chen, W. -L. Chen, N. -S. Pai, M. -H. Chou, G. -S. Huang and C. -Y. Chang, "Implement of a 6-DOF manipulator with machine vision and machine learning algorithms," *2017 International Conference on Applied Electronics (AE)*, 2017, pp. 1-5, doi: 10.23919/AE.2017.8053583.

[4] W. A. A. Praditasari, R. Aprilliyani and I. Kholis, "Design and Implementation of Interactive Virtual Museum based on Hand Tracking OpenCV in Indonesia," *2021 8th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 2021, pp. 253-256, doi: 10.23919/EECSI53397.2021.9624265.

[5] A. Chaikaew, K. Somkuan and T. Yuyen, "Thai Sign Language Recognition: an Application of Deep Neural Network," *2021 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering*, 2021, pp. 128-131, doi: 10.1109/ECTIDAMTNCON51128.2021.9425711.

[6] K. Kaneko *et al.*, "Humanoid Robot HRP-5P: An Electrically Actuated Humanoid Robot With High-Power and Wide-Range Joints," in *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1431-1438, April 2019, doi: 10.1109/LRA.2019.2896465.

[7] M. P. Polverini, A. Laurenzi, E. M. Hoffman, F. Ruscelli and N. G. Tsagarakis, "Multi-Contact Heavy Object Pushing With a Centaur-Type Humanoid Robot: Planning and Control for a Real Demonstrator," in *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 859-866, April 2020, doi: 10.1109/LRA.2020.2965906.

[8] I. Rodríguez-Moreno, J. M. Martínez-Otzeta, I. Goienetxea, I. Rodriguez and B. Sierra, "A New Approach for Video Action Recognition: CSP-Based Filtering for Video to Image Transformation," in *IEEE Access*, vol. 9, pp. 139946-139957, 2021, doi: 10.1109/ACCESS.2021.3118829.

[9] D. Guidotti, F. Leofante, A. Tacchella and C. Castellini, "Improving Reliability of Myocontrol Using Formal Verification," in *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 4, pp. 564-571, April 2019, doi: 10.1109/TNSRE.2019.2893152.

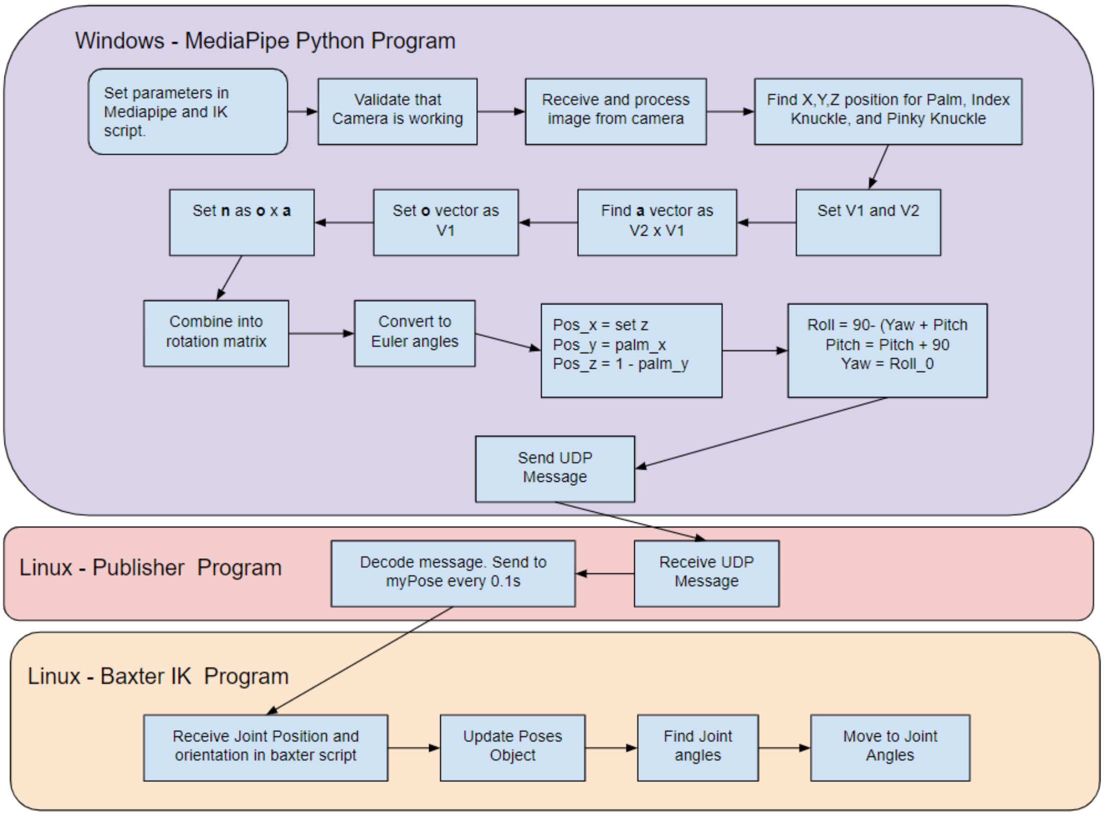
[10] K. Bartol, D. Bojanić, T. Petković and T. Pribanić, "A Review of Body Measurement Using 3D Scanning," in *IEEE Access*, vol. 9, pp. 67281-67301, 2021, doi: 10.1109/ACCESS.2021.3076595.

[11] N. Renotte, “Ai hand pose estimation with MediaPipe detect ... - youtube,” *YouTube*, 25-Apr-2021. [Online]. Available: https://www.youtube.com/watch?v=EgjwKM3KzGU. [Accessed: 14-Dec-2021].

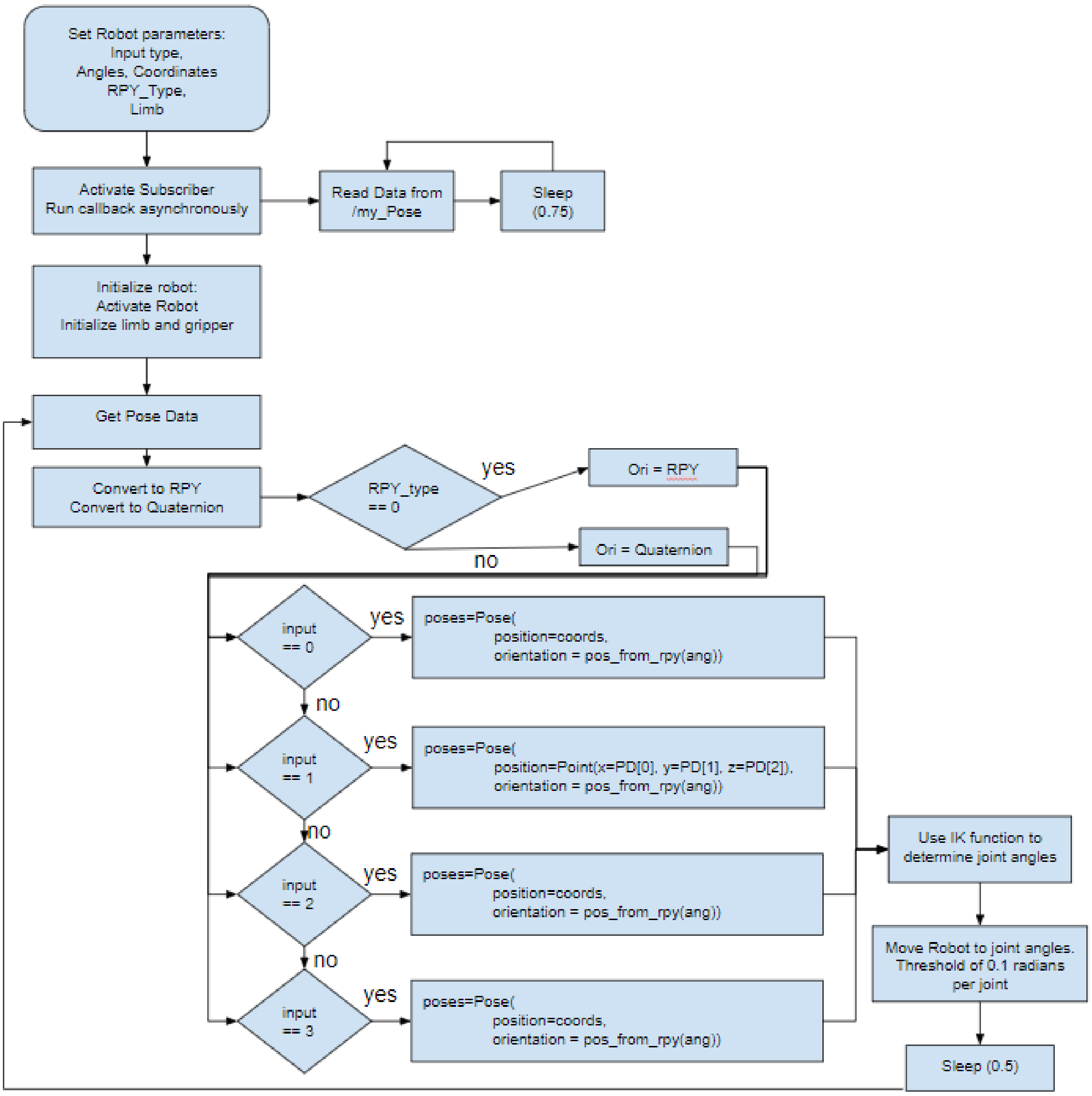
[12] Sergio, “Distance detection with Depth Camera,” *Youtube*, 11-Mar-2021. [Online]. Available: https://www.youtube.com/watch?v=mFLZkdH1yLE&list=PLYn_Gw6hOCXrtBbpMDT42jEHyGvrBbmhB&index=2&t=712s. [Accessed: 14-Dec-2021].

VIII. Appendix

I. FLOWCHART OF OPERATIONS


Windows - MediaPipe Python Program
Set parameters in Mediapipe and IK script.
Validate that Camera is working
Receive and process image from camera
Find X,Y,Z position for Palm, Index Knuckle, and Pinky Knuckle
Set V1 and V2
Find **a** vector as V2 x V1
Set **o** vector as V1
Set **n** as **o** x **a**
Combine into rotation matrix
Convert to Euler angles
Pos_x = set z
Pos_y = palm_x
Pos_z = 1 - palm_y
Roll = 90- (Yaw + Pitch
Pitch = Pitch + 90
Yaw = Roll_0
Send UDP Message
Linux - Publisher Program
Receive UDP Message
Decode message. Send to myPose every 0.1s
Linux - Baxter IK Program
Receive Joint Position and orientation in baxter script
Update Poses Object
Find Joint angles
Move to Joint Angles

II. FLOWCHART OF BAXTER IK PROGRAM


Set Robot parameters:
Input type,
Angles, Coordinates
RPY_Type,
Limb
Activate Subscriber
Run callback asynchronously
Read Data from
/my_Pose
Sleep
(0.75)
Initialize robot:
Activate Robot
Initialize limb and gripper
Get Pose Data
Convert to RPY
Convert to Quaternion
RPY_type
== 0
yes
Ori = RPY
no
Ori = Quaternion
input
== 0
yes
poses=Pose(
position=coords,
orientation = pos_from_rpy(ang))
no
input
== 1
yes
poses=Pose(
position=Point(x=PD[0], y=PD[1], z=PD[2]),
orientation = pos_from_rpy(ang))
no
input
== 2
yes
poses=Pose(
position=coords,
orientation = pos_from_rpy(ang))
no
input
== 3
yes
poses=Pose(
position=coords,
orientation = pos_from_rpy(ang))
Use IK function to
determine joint angles
Move Robot to joint angles.
Threshold of 0.1 radians
per joint
Sleep (0.5)