Date of current version 12/08/2021.

Mirror Movement Robotic Arm

Garrett P. Tjernagel, William R. Prody, and Branson P. Elliott

“This work was supported in part by the University of North Dakota.”

*ABSTRACT*

The aim of this research paper is to introduce a new method for designing a robotic arm from shoulder to fingertips that is fully controlled by mirroring the motion of a human. Prior research was done in this area before conducting calculations and modeling for the arm. Using prior research along with innovative ideas from the group, the Mirror Movement Robotic Arm was proposed. The design process for the MMRA is discussed along with different options for the design and why the final design was chosen. Kinematic calculations are shown for the arm in the design process along with modeling for the arm. Controller flow for the robotic arm and hand are discussed and the designed circuits for the controller are shown. After all of the calculations and modeling was done the group was able to find a feasible design for a 6 DOF robotic arm and 3 DOF hand..

*INDEX TERMS* Kinematics, DH Table, Controller Flow, Controller Circuit, DOF, Modeling, Actuation, Adduction, Abduction, Gyroscope, Arduino DUE

1. *INTRODUCTION*

Continued improvement in the field of robotics leads to ever more impressive engineering and design. Improvements to the speed, size, and access to technology allows more and more people to innovate. This paper describes the work and research done to create a robotic arm controlled by gyroscopic data sampling. To achieve high speed while maintaining acceptable levels of accuracy and precision, development of kinematics, modeling, control systems, and power systems are all required to provide an end user with as seamless of an experience controlling a robotic arm with as little training as possible.

1. *METHODS AND THEORY*

In this section we will describe the process taken in designing the mirror movement robotic arm using. The ///order in which the project design was done followed the process described by Puig et al.

1. Problem Definition

When it comes to the area of robotic modeling based on human functions, one of the more difficult parts to imitate is the arm and hand. There has been some extensive research done in this area over recent years, however, it seems that there haven’t been any studies that tried to replicate a human arm from shoulder to fingertips. Our group decided to take on this task and try to replicate the function of a human arm to be as accurate and responsive as possible. The applications for the use of this controlled arm could be in environments that are too dangerous for humans, or advancement in healthcare.

1. Concept Design

The beginning stage for designing the arm started with discussing where rotation and how many degrees of freedom (DOF) were needed. According to H. Kim et al., human arm motion is redundant in the fact that it contains 7 DOF, however it only requires 6 DOF to position the wrist and orient the palm in 3D space. This became a driving force in our decision to design for 6 DOF arm. We also aimed to create the most realistic feel possible so we will mount the arm on a vertical pedestal to represent the arm attached to a body torso.

The joint motion of the arm will be driven through a series of rotational actuators. The shoulder joint will sit atop the pedestal and contain two degrees of freedom. The combination of these two degrees of freedom working simultaneously will create a third DOF. At the elbow joint, there will be a single rotating motion to create the curl of the forearm up to the bicep and vice versa. We will apply a forearm rotating motion to supplement for wrist rotation and palm orientation. Based on the anatomy of the human arm, when the wrist is rotated, the forearm rotates likewise. This concept again applies a realistic feel to the model, and adds a fifth DOF. The final degree of freedom will come at the combination of the wrist and base of the hand. The motion will represent the pitch of the wrist, giving it the freedom to move up and down.

When it came to the hand a few ideas were thrown around especially in regards to thumb movement. For the finger actuation, there was the possibility to have motors for rotating each finger joint, however there would not be enough room nor money to implement this. Cable-driven actuation was chosen for the fingers instead. The thumb was originally going to be designed using a ball joint in the hand and a motor to rotate the joint. Instead the adduction and abduction of the thumb was chosen to be imitated using a sweeping motion along with cable driven actuation.

1. *RELATED RESEARCH*

So far, the majority of robotic manipulators have been only small sections of the body like the hand. In regards to the hand, there are multiple robotic hands that have been made but most of them used tripod type grasping. The other robotic arms that were reviewed seemed to have a fairly slow response time to input. Ideally, the response time for this arm will be as close to zero as possible. Other arms also use wired connections to control the robotic arm, and this project aims to be able to control the arm wirelessly. Along with this comes a unique feature to this arm in which the end goal is for any person to be able to slip on the controlling gauntlet and have a natural feel to the control.

For many of the naturally controlled robotic arms, meaning those controlled by mimicking human movement, many were large and bulky when it came to design and operation. L. Chen et al describe ways to implement AI enabled wearable manipulator controllers. The group plans to use some of the ideas here focusing mainly on the wearable side of things with less emphasis on the AI as that would be harder to implement, despite the merit and enhancements that the AI could provide. S. Alvarez-Rodriguez and F. G. Peña Lecona discuss using an nth degree order system of sensors to control an nth-Degrees of Freedom (DOF) arm. The group found some of the ideas in their work especially valuable since we are using 3 main sensors, 4 feedback sensors, as well as 5 auxiliary sensors to control our system, not to mention all wirelessly to enable freedom of movement for the user. W. G. Hao et al also talk about control of a 6-DOF arm with efficient trajectory planning and speed control, which can be implemented into the groups system effectively as the group seeks to implement a finely controlled arm as efficiently as possible. Building again off the wireless nature of the groups design, N. RNaveen specifically talks about Arduino communication protocols, of which are specifically useful for the design of the MMRA.

1. *EQUATIONS*

In order to find the kinematics of the hand, the transformation matrix first had to be found from the DH table. The DH table is found by analyzing the joints of the arm and hand. The number of joints determines the number of steps needed to go from the first joint to the last. Each DH table has a spot for rotation in Z(theta), translation in z(d), rotation in x(alpha), and translation in x(a), along with the variable. An example of the table is shown in Table 1.

| Order |  | d | a |  | Var. |
| --- | --- | --- | --- | --- | --- |
| 0-1 | 0 | d1 | 0 | 0 | 1 |
| 1-2 | 15 dgs | 0 | d2 | 0 | 2 |
| 2-3 | 0 | d3 | 0 | 0 | 3 |

Table 1 - DH Table Format

After the DH table is calculated the homogeneous transformation matrix can be found. This is done by multiplying the different operations in each row of the DH table together. Equation (1) shows the transformation matrix, , for the first row of a DH table.

(1)

When the transformation matrix for each step was found, the overall transformation matrix was the product of each step multiplied in order. So for a DH table with three steps, the overall transform would be,

(2).

Using the overall transformation and the transformation for each step, the jacobian can be found to describe the forward and inverse motion of the system. The jacobian of a system is found with the equation,

(3)

Where,

(4)

and,

(5)

for revolute joints.

1. *DESIGN*
2. Hand
3. Control

The control of the hand is fairly simple. First, the finger actuation motion is controlled by steel and rubber wires that act like tendons in the hand. Each finger has three joints with the steel cable and rubber cable tied off at the tip of the finger. The rubber cable for each finger will be tied off at the wrist joint when the hand is in its resting position. The steel cable for each finger is tied to a servo motor in the forearm of the robot. When motion from the control glove is input, the servo motor will turn and pull on the steel cable which will bend the joints of the fingers. When the controlling finger is moved back to its rest position, the tension in the rubber cable pulls the robotic finger back to the rest position. The fingers will not have any side to side motion and therefore will just have one degree of freedom to bend down. The thumb is different from the other fingers due to the extra degree of freedom in the thumb. With the sweeping motion, the control glove will register the side to side inputs of the thumb for the sweeping motion in the robotic thumb. As far as the bending motion, all the fingers have one degree of freedom for curling the fingers down.

1. Materials

The hand itself will be 3-D printed from the CAD model which will require printing material. For the actuation of the fingers both steel and rubber cable is needed for the hand. Finally, a 9mm bolt is needed for the thumb sweeping as well as the wrist. There will need to be 7 servo motors for the hand, 5 for the finger actuation, 1 for thumb sweeping, and 1 for the wrist.

1. Modeling

When it came to modeling the hand, the group decided to make it as close to the same size as a regular human hand as possible. L. Tian et al. offers the idea to measure the length of each “link” on a human hand which is what was done to scale the 3-D model (n. pag.). The hand model was created using tinkercad and will be converted to an STL file when the hand is printed. H. Mnyusiwalla et al. points out that, when modeling a tendon based actuating hand, the designer needs to be careful of friction with the tendons. They go on to advise designers to minimize sliding surfaces, using pulleys, and making the tendon paths as straight as possible (p. 812). In implementation the straightest paths from the finger to the wrist were made and guides were inserted for the steel and rubber cables. There was not enough room in the fingers and hand for pulleys however, and the least number of guides possible were used to help reduce the friction. A gyroscope mount is included on the backside of the hand and two hooks were attached to the base of the thumb to get the sweeping motion. The 3-D model of the hand is shown below in figure 1.

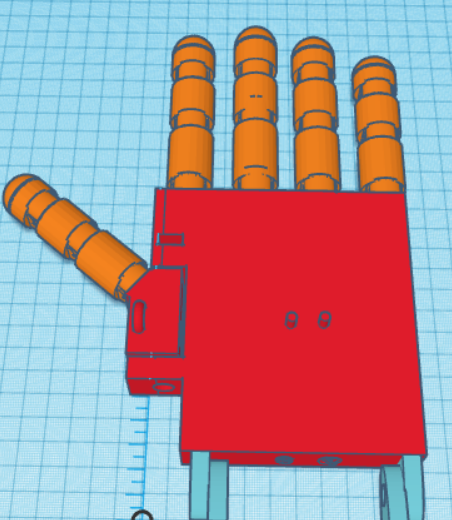


Figure 1 - Top View of Hand

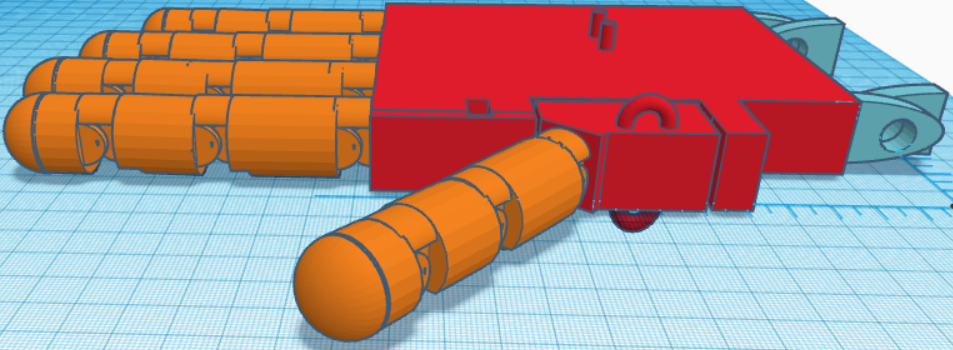


Figure 2 - Side View of Hand

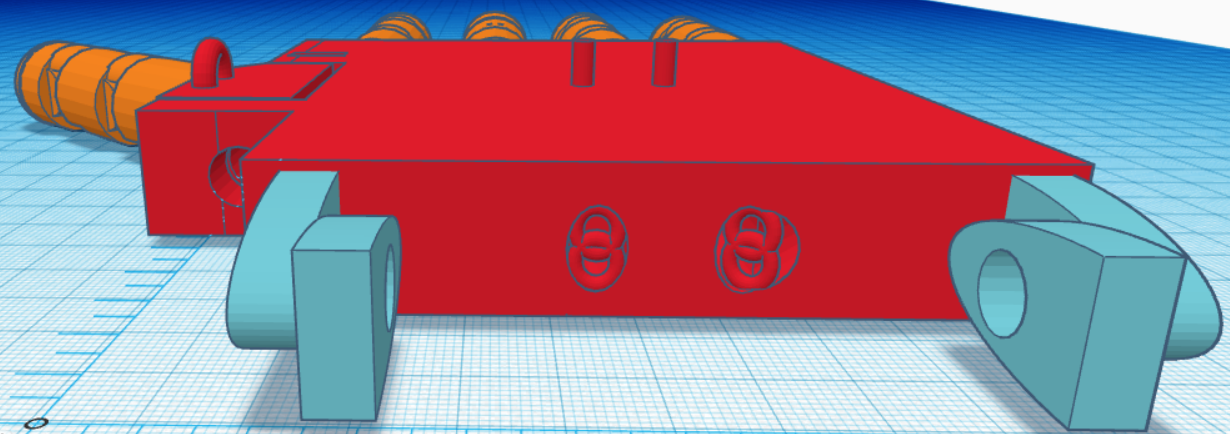


Figure 3 - Backside View of Hand

When viewing the side of the hand, one can see the stops on the backside of each joint of each finger. This will keep the fingers from being able to go past the resting point. There are notches in the links of the fingers as well in order to keep the links from stopping each other. Finally, in the back view one can see the two openings for the steel and rubber cables along with some of the guides. There is also a 9mm hole for the thumb adduction and abduction.

1. Kinematics

In order to calculate and analyze the motion of the hand, the kinematics of the hand were evaluated and will be explained in the following paragraphs. The first step for the kinematics is the DH table. Two separate DH tables were made for the hand, one that works for the pinkie, middle, ring, and index fingers, and one that works for the thumb motion. Both of the DH tables can be seen below.

| Order |  | d | a |  | Var |
| --- | --- | --- | --- | --- | --- |
| 0-1 | 0 | 0 | d1 | 0 | 1 |
| 1-2 | 0 | 0 | d2 | 0 | 2 |
| 2-3 | 0 | 0 | d3 | 0 | 3 |

Table 2 - DH table of fingers

The index finger and the ring finger are the same length so the d1-d3 values will stay the same for each of those. For the middle and pinky fingers, the d1-d3 values will increase and decrease in size respectfully. The DH table for the thumb is shown in table 3.

| Order | Theta | d | a |  | Var |
| --- | --- | --- | --- | --- | --- |
| 0-1 | 0 | 0 | 0 | - | 1 |
| 1-2 | - | 0 | d2 | 0 | 2 |
| 2-3 | 0 | 0 | d3 | 0 | 3 |
| 3-4 | 0 | 0 | d4 | 0 | 4 |

Table 3 - DH table of thumb

The lengths for each distance were measured on tinkercad and a value of the different lengths in inches can be found in table 4. The distances found were measured from the middle of one joint to the middle of the next joint.

| Finger | d1 | d2 | d3 | d4 |
| --- | --- | --- | --- | --- |
| Index and ring | 1.35 | 1.1 | 0.75 | N/A |
| middle | 1.47 | 1.1 | 0.69 | N/A |
| pinky | 1.17 | 0.96 | 0.5 | N/A |
| thumb | 0 | 1.2 | 0.96 | 0.66 |

Table 4 - Distance values measured for each finger

After the DH tables have been made, the overall transformation matrix can be found. This is done by finding the homogeneous transformation matrix for each step and then multiplying them together. This was done using MATLAB and the code and results can be seen below.

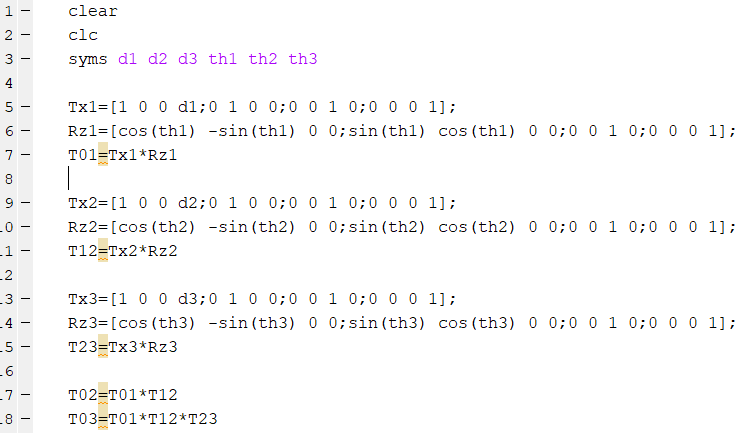


Figure 4 - Transform code for four fingers

In order to find the transform matrix for each of the different fingers, the d1-d3 values from Table 4 would be plugged in.

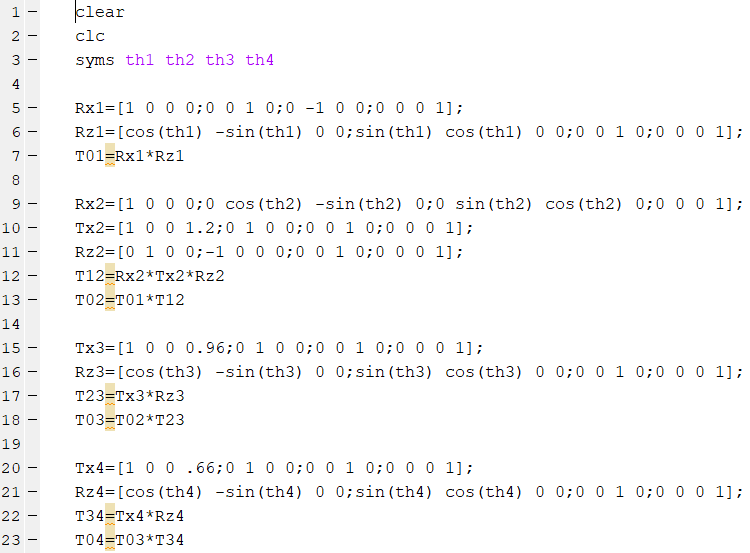


Figure 5 - Transform code for Thumb

To see the resulting transform matrices for the fingers and thumb see Appendix B. After the transform matrices are found, the jacobian can be calculated. The code for calculating the Jacobian for the fingers and thumb is shown below and the resulting Jacobian Matrices can be found in Appendix C.

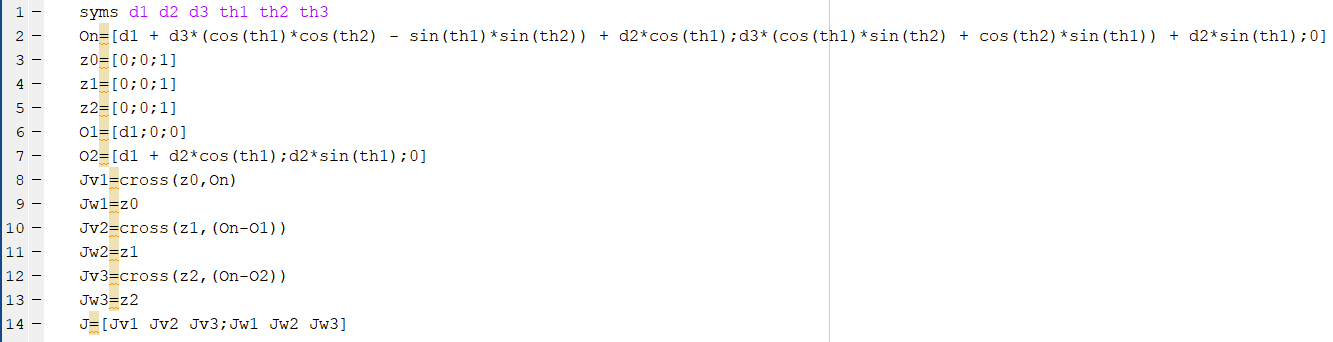


Figure 6 - Jacobian Calculations for Fingers

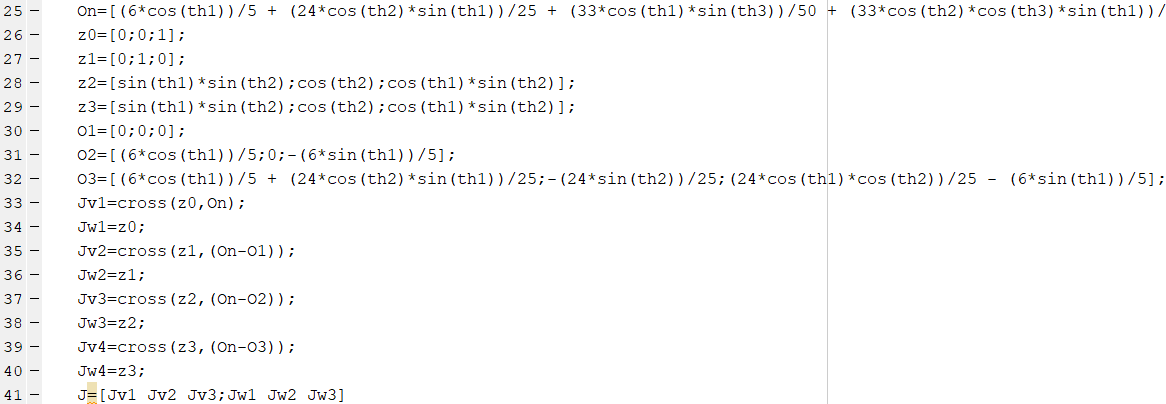


Figure 7 - Jacobian Calculations for Thumb

1. Arm
2. Control

Control of the arm mechanics will be done through a series of electric drives, 2 arduino microcontrollers, multiple gyroscopes, and feedback signals. As a group, we designed for 6 degrees of freedom which requires three separate joints. The joints of discussion are shoulder, elbow, and wrist. As stated above, the goal is to create a robotic arm that mirrors the movement to that of the gauntlet controller. The arm gauntlet controller will contain an arduino due microcontroller, 3 gyroscopes, a radio transmitter, and a battery power supply. The robotic arm will also utilize an arduino due, 3 gyroscopes, 5 motors with encoders, and a power supply.

The physical location of the gyroscopes on both the robotic arm and the gauntlet are as follows: one on the center point of the upper arm link, one on the center point of the lower arm link, and one on the back of the hand. The three gyroscopes will allow us to place where the user’s arm is located at any point in time using vector math to place the arm in space. Positional data values will be taken from the gyroscopes of the gauntlet and transmitted from the gauntlet microcontroller to the robotic arm microcontroller. This positional data will be sampled multiple times in a matter of milliseconds for precise locating before getting relayed to the arm. The radio transmitter located on the arm will receive positional data from the gauntlet. The arm microcontroller will process data, compute movement, and relay signals to the relevant motors. At the same time, the arm control gyroscopes will be sampled for feedback positional data to be fed into a control loop compared to the received data in order to acquire precise movement of the motors. Each motor of the arm will have an encoder attached that will be fed into a feedback control loop allowing for precise control of the system.

1. Materials

The joint parts will be designed via CAD modeling and then 3D printed. This will provide a very sturdy base for each rotational motion and movement of the arm joints. The links for the upper and lower arm sections will be made from rigid aluminum plates. These will provide a solid foundation from joint to joint as well as a lightweight design. The electric drives will be 12V DC motors coupled with encoders.

1. Modeling

As stated before, when modeling the arm, we aimed for 6 DOF. We also aimed to design an arm that is as similar to a human arm as possible. These objectives will , in theory, provide us with a robotic arm that will behave as similar to a human arm as time and money will allow. Figures 8 and 9 show the different directions of motion at each joint of the arm as well as the pedestal which de hope t

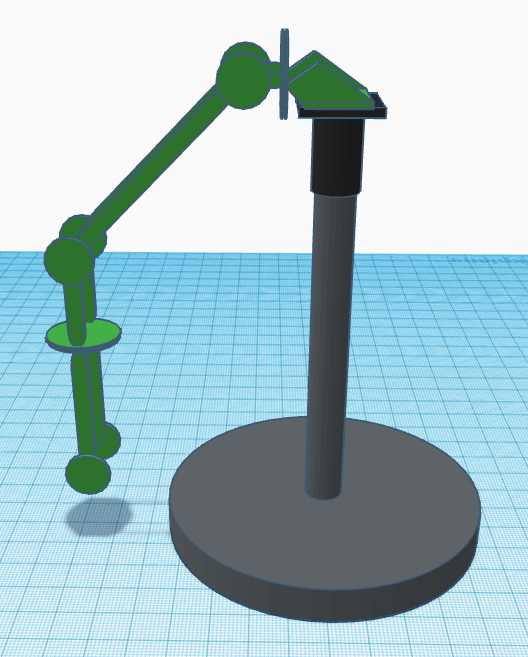


Figure 8. Frontside View of Arm joints and motion

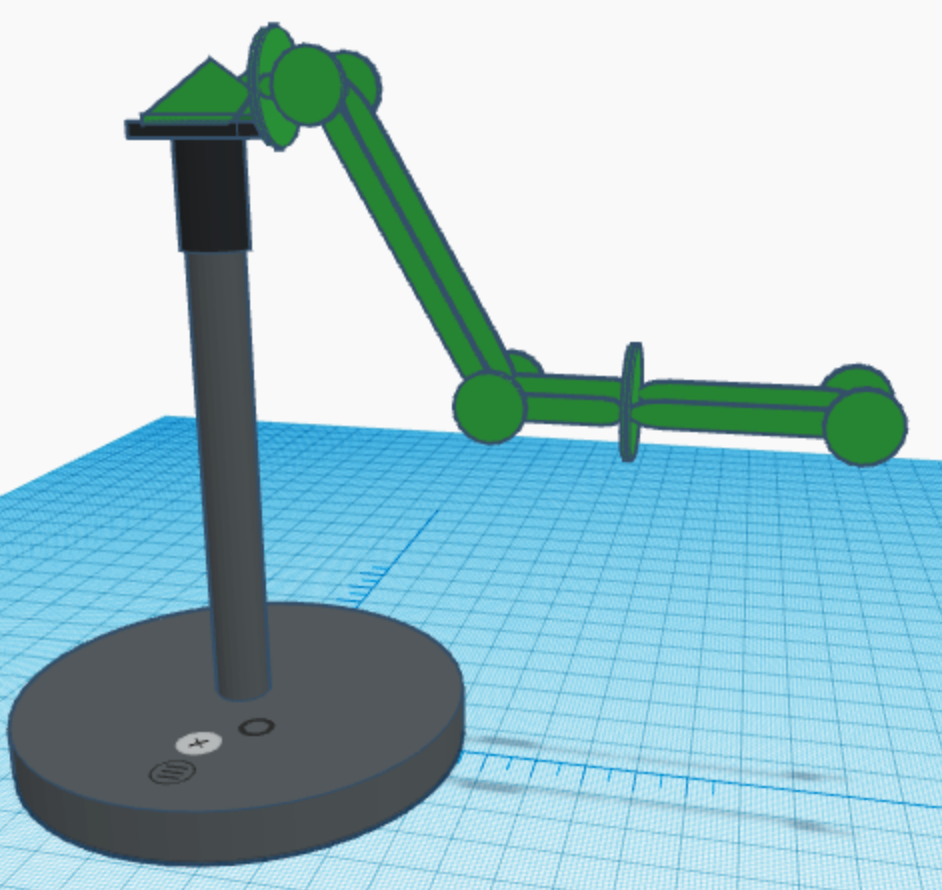


Figure 9. Backside View of Arm joints and motion

1. Kinematics

To determine the orientation and location of the end effector (hand), the kinematics for the arm

were evaluated. To do this, we determined the DH table of the robotic arm. The DH table can be seen in Table 5 below.

| Order | 𝜽(z) | d(z) | a(x) | 𝛼(x) | Var |
| --- | --- | --- | --- | --- | --- |
| 0-1 | 0 | 0 | a1 | -90 | 𝜽1 |
| 1-2 | 0 | 0 | a2 | 0 | 𝜽2 |
| 2-2.5 | 0 | 0 | a3 | 90 | 0 |
| 2.5-3 | 90 | 0 | 0 | 90 | 𝜽3 |
| 3-4 | 0 | d4 | 0 | -90 | 𝜽4 |
| 4-4.5 | 0 | 0 | 0 | 90 | 0 |
| 4.5-5 | 0 | d5 | 0 | 0 | 𝛄 |

Table 5 - DH Table of the arm

C. Control Systems

1. Gauntlet Controller Control Flow

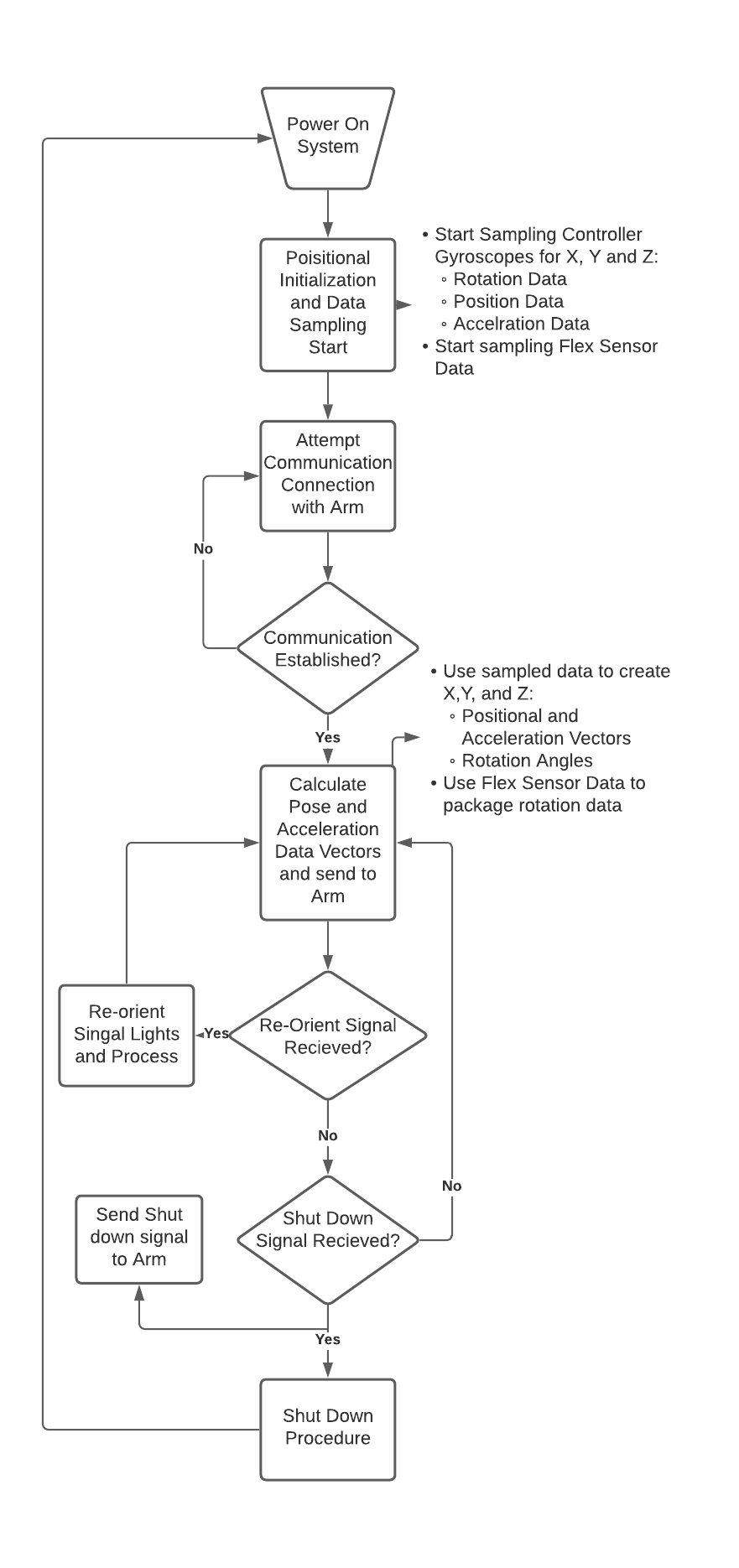


Figure 10. Control Flow Diagram of Gauntlet Controller

This system seems complex in form and function, which it is, but broken down, it can be visualized quite simply. After powering on the gauntlet controller, an initialization sequence is started. The gyroscopes will start sampling the rotational, and acceleration data. This will then, depending on the final system, be initialized in a position designed by the operator, or will be collected automatically with orientation to gravity. After initializing, the controller will attempt to establish a connection to the arm. After which the arm will move into an initial position, indicating to the user that the arm is ready to be controlled. During this process, the gyroscopes will be continuously sampled for data, and sent to the arm for processing. During this sampling loop, the controller will be checking for error signals from the arm as well as the controller in the event that data packets sent are incomplete or data packets are incorrect by a large margin. In such an event a signal is sent to the arm indicating that it needs to initialize once more.

1. Arm Control Flow

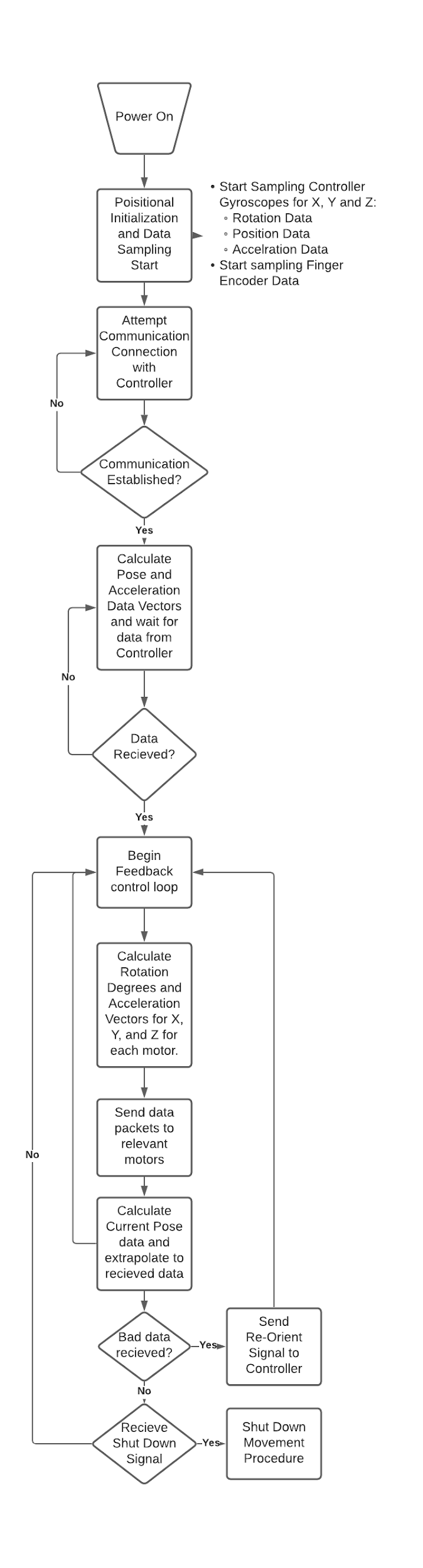


Figure 11. Control Flow Diagram of Arm

Similar to the gauntlet controller, the control flow of the arm is simple once broken down. After powering on, the arm begins sampling its gyroscopic data for rotation, position, and acceleration. After which it waits for a communication connection to the controller. After which the arm begins its initialization process. Which will either be a series of movements to ensure correct calibration of the motors, or slow matching movement to the controller. Initialization ends, and the control feedback loop starts. The data sampling of the arm is compared to the data sent from the controller and will output rotation and acceleration values for the motors in the arm and hand. This loop is continuously checking the output desired pose data and is checked against the current pose data in order to determine how fast or slow the system needs to be moving. In the event that an error occurs by way of incomplete data packets or inability to carry out a command, the arm will move to the startup position and initialize once more. After the controller receives the power off command, a second command is communicated to the arm to move it back to the starting position. This way, the arm will always know the starting position and will make initialization simple and effective for the next power up.

1. Materials

For the controller base the group decided on using aluminum base parts to connect 3D printed rotating cuffs. This allows for a lightweight ‘gauntlet’ to mount all the following components to. Since our system will be resource and sensor intensive, the Arduino DUE was selected for its high processing power and large number of pin options. The large number of pin options will allow us to connect the multiple gyroscopic and flex sensors as well as the radio module for fast data transmission to the robotic arm. To power the controller, the power draw of the system is relatively low and will only require one 9V battery.

The controller circuit is outlined in the following figure.

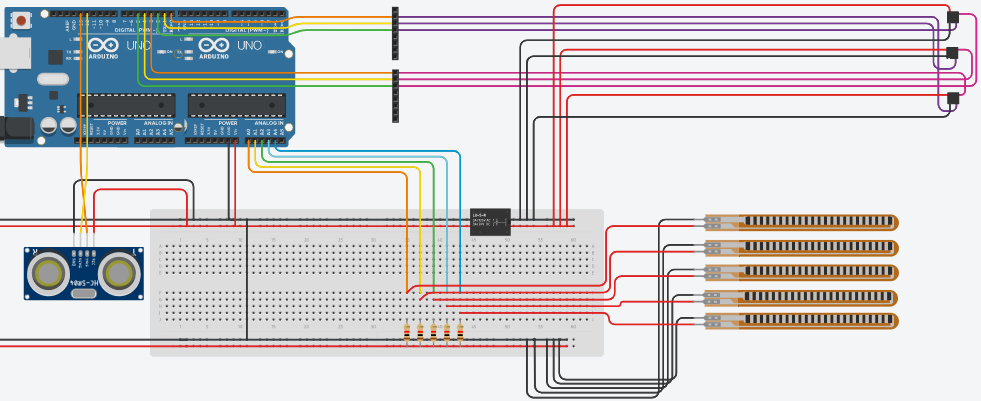


Figure 12. Control Circuit of Gauntlet Controller

Each gyroscope and flex sensor will be connected to one of the pins on the arduino, the radio transmitter will connect to the serial data pins, and there are two power busses for the circuit, one 5V for the flex sensors and a 3.3V for the arduino DUE and the gyroscope modules.

The arm controller circuit is outlined in the following figure.

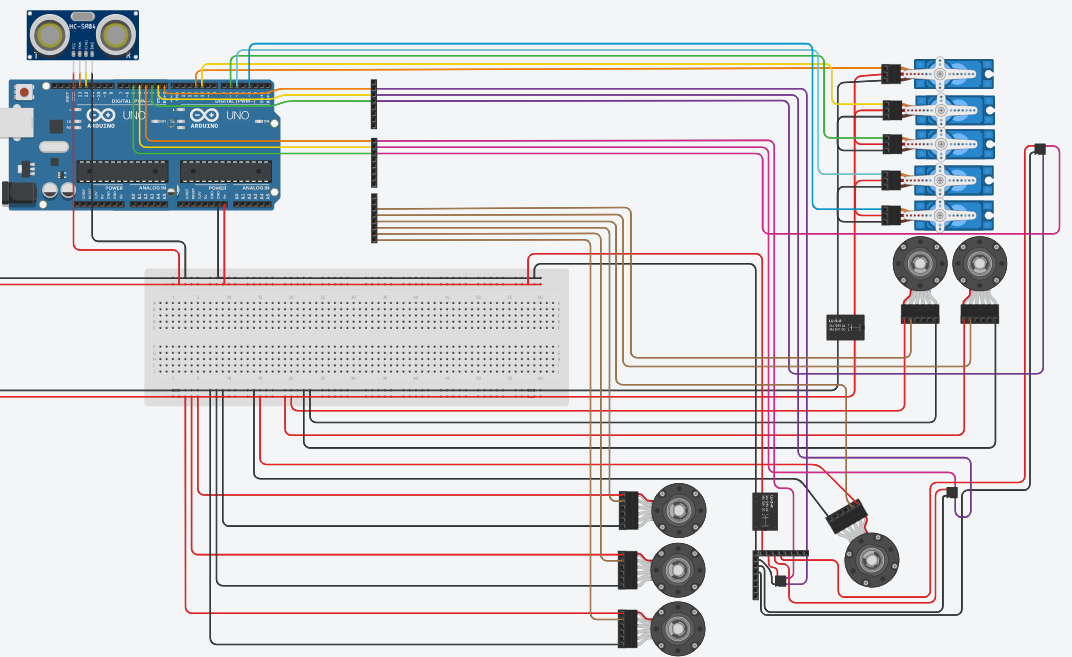


Figure 13. Control Circuit of Arm

The gyroscopes are all connected on a separate power bus with the arduino similar to the controller. Each hand servo has its own power bus as do the six arm motors and encoders.

For a complete parts list see Appendix A.

1. Modeling

For the model there were a few questions to ask to help with the control of the system as a whole. A regular human forearm will rotate slightly with the wrist, so a cuff was needed to keep the rotation of the wrist and upper arm separate from the whole rotation in order to correctly correlate the positional data in the control loop. This will be done by inserting a rotating cuff at the wrist which is attached to the glove and the end of the forearm section. The second thing to consider is that there is only one joint located on the robot for the ‘elbow’, so ridgid aluminum plates will run along the sides of the upper and lower arm with a joint at the elbow only allowing rotation in the X direction of the joint. Finally there will be a final cuff and shoulder strap to hold the upper part of the brace in place to the user. All of these things in tandem will allow for the controller to kinematically mimic the desired movements of the arm. When collecting data and extrapolating the controller and arm rotational and acceleration vector values they will be more closely related and will require less processing of the microcontroller.

D. Power System

1. Robotic Arm

The driving motors for the joint movement will be 12V motors thus they will require a 12V DC power supply. The Arduino DUE microcontroller has an recommended input voltage of 7-12V with a maximum input voltage level of 16V. To ensure safe operation and function, we will use a 12V supply for both the arm motors and microcontroller. Our robotic arm will remain stationary on a solid surface so we will use a wired power supply driven from a wall outlet. From the outlet, an AC to DC converter will be connected to convert and step the voltage level down to a usable 12Vdc. A 12V power bus will be used to operate the arm motors. The servo motors used for finger motion will require a 5V supply. A 5V power bus will be used for their operation. Gyroscopes on the arm will be connected to various pins of the Arduino DUE microcontroller.

1. Gauntlet Controller

The gauntlet controller will not contain any motors but will have an Arduino DUE for signal processing. As stated above, the recommended input voltage of the microcontroller is 7-12V. Additionally, we want our gauntlet controller to be free of wired connections as we would like to be able to move around and operate the arm from various ranges. In order to achieve this, we will use a 9V battery power supply to operate the gauntlet arduino microcontroller. This will be mounted to the gauntlet controller to maintain free range of motion. Gyroscopes and flex sensors will be connected to various pins on the microcontroller. The radio transmitter will be connected to the serial data pins of the microcontroller.

References

[1] H. Mnyusiwalla, P. Vulliez, J.-P. Gazeau, and S. Zeghloul, “A new dexterous hand based on bio-inspired finger design for inside-hand manipulation,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 6, pp. 809–817, Jun. 2016.

[2] J. E. Puig, N. E. Rodriguez, and M. Ceccarelli, “A methodology for the design of robotic hands with multiple fingers,” *International Journal of Advanced Robotic Systems*, vol. 5, no. 2, p. 22, 2008.

[3] L. Tian, N. M. Thalman, D. Thalmann, and J. Zheng, “Nature grasping by a cable-driven under-actuated anthropomorphic robotic hand,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 17, no. 1, p. 1, 2019.

[4] L. Chen, H. Sun, W. Zhao, and T. Yu, “Robotic Arm Control System Based on AI Wearable Acceleration Sensor,” *Mathematical Problems in Engineering*, 03-Mar-2021. [Online]. Available: https://www.hindawi.com/journals/mpe/2021/5544375/. [Accessed: 12-Oct-2021]

[5] S. Alvarez-Rodríguez and F. G. Peña Lecona, “th Order Sensor Output to Control -DoF Serial Robot Arms,” *Journal of Sensors*, 11-Jan-2021. [Online]. https://www.hindawi.com/journals/js/2021/8884282/.[Accessed: 12-Oct-2021].

[6] W. G. Hao, Y. Y. Leck and L. C. Hun, "6-DOF PC-Based Robotic Arm (PC-ROBOARM) with efficient trajectory planning and speed control," 2011 4th International Conference on Mechatronics (ICOM), 2011, pp. 1-7, doi: 10.1109/ICOM.2011.5937171

[7] N. RNaveen, “Arduino Communication Protocols (Wired and Wireless for IoT),” *Technobyte*, 03-Nov-2019. [Online]. Available: https://technobyte.org/arduino-communication-protocols-wired-wireless-for-iot/. [Accessed: 12-Oct-2021].

[8] Talli, Amit, and Vinod Kumar V Meti. “Design, Simulation, and Analysis of a 6-Axis Robot Using Robot Visualization Software.” *IOP Science*, 2020, https://iopscience.iop.org/article/10.1088/1757-899X/872/1/012040.

[9] Pachaiyappan, S, et al. “Design and Analysis of an Articulated Robot Arm for Various Industrial Applications.” *IOSR Journal*, 2014, https://www.iosrjournals.org/iosr-jmce/papers/NCCAMABS/V

[10] Reddy, Gurudu Rishank, and Venkata Krishna Prashanth Eranki. “Design and Structural Analysis of a Robotic Arm.” *DIVA Portal*, 2016, https://www.diva-portal.org/smash/get/diva2:1068547/FULLTEXT02.

[11] P. Bora and V. Nandi, "Low cost shadow function based articulated robotic arm," *2015 International Conference on Energy, Power and Environment: Towards Sustainable Growth (ICEPE)*, 2015, pp. 1-4, doi: 10.1109/EPETSG.2015.7510079.

[12] H. Kim, L. M. Miller, N. Byl, G. M. Abrams and J. Rosen, "Redundancy Resolution of the Human Arm and an Upper Limb Exoskeleton," in *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 6, pp. 1770-1779, June 2012, doi: 10.1109/TBME.2012.2194489.

APPENDIX

1. Parts List

| Part | Count | Cost/Unit |
| --- | --- | --- |
| Arduino DUE | 2 | 40.30 |
| GY-512 MPU6050 | 7 | 2.99 |
| 4PC 20kg Digital 180deg Servo Motors | 2 | 18.99 |
| 12V DC motor with encoder | 6 | 19.99 |
| 4PC NRF24L01 2.4G Radio Transmitter | 1 | 7.89 |
| AC to DC Power Converter | 1 | 20.99 |
| 9V Battery Holder | 1 | 6.99 |
| Misc Wire and Heat Shrink Kit | 1 | 20-30.00 |
| Dupont Connector Kit | 1 | 23.99 |
| Soldering Supplies | 1 | 15.98 |
| Steel Cable | 1 | 18.98 |
| Elastic Band Cable | 1 | 4.39 |
| Assorted Bolts and Connectors | 1 | 20.00 |
| Braided Wire | 1 | 8.00 |
| 2mm Braided rubber | 1 | 10-15.00 |
| Steel Wire | 1 | 10.00 |

<https://store-usa.arduino.cc/products/arduino-due>

1. Fingers and Thumb transform

The transform for each of the fingers with the variables d1-d3 included are shown below.

| cos(th3)\*(cos(th1)\*cos(th2) - sin(th1)\*sin(th2)) - sin(th3)\*(cos(th1)\*sin(th2) + cos(th2)\*sin(th1)) | - cos(th3)\*(cos(th1)\*sin(th2) + cos(th2)\*sin(th1)) - sin(th3)\*(cos(th1)\*cos(th2) - sin(th1)\*sin(th2)) | 0 | d1 + d3\*(cos(th1)\*cos(th2) - sin(th1)\*sin(th2)) + d2\*cos(th1) |
| --- | --- | --- | --- |
| cos(th3)\*(cos(th1)\*sin(th2) + cos(th2)\*sin(th1)) + sin(th3)\*(cos(th1)\*cos(th2) - sin(th1)\*sin(th2)) | cos(th3)\*(cos(th1)\*cos(th2) - sin(th1)\*sin(th2)) - sin(th3)\*(cos(th1)\*sin(th2) + cos(th2)\*sin(th1)) | 0 | d3\*(cos(th1)\*sin(th2) + cos(th2)\*sin(th1)) + d2\*sin(th1) |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 |

Result of Finger Transform Matrix

| cos(th4)\*(cos(th1)\*sin(th3) + cos(th2)\*cos(th3)\*sin(th1)) + sin(th4)\*(cos(th1)\*cos(th3) - cos(th2)\*sin(th1)\*sin(th3)) | cos(th4)\*(cos(th1)\*cos(th3) - cos(th2)\*sin(th1)\*sin(th3)) - sin(th4)\*(cos(th1)\*sin(th3) + cos(th2)\*cos(th3)\*sin(th1)) | sin(th1)\*sin(th2) | (6\*cos(th1))/5 + (24\*cos(th2)\*sin(th1))/25 + (33\*cos(th1)\*sin(th3))/50 + (33\*cos(th2)\*cos(th3)\*sin(th1))/50 |
| --- | --- | --- | --- |
| sin(th2)\*sin(th3)\*sin(th4) - cos(th3)\*cos(th4)\*sin(th2) | cos(th3)\*sin(th2)\*sin(th4) + cos(th4)\*sin(th2)\*sin(th3) | cos(th2) | - (24\*sin(th2))/25 - (33\*cos(th3)\*sin(th2))/50 |
| sin(th2)\*sin(th3)\*sin(th4) - cos(th3)\*cos(th4)\*sin(th2) | sin(th4)\*(sin(th1)\*sin(th3) - cos(th1)\*cos(th2)\*cos(th3)) - cos(th4)\*(cos(th3)\*sin(th1) + cos(th1)\*cos(th2)\*sin(th3)) | cos(th1)\*sin(th2) | (24\*cos(th1)\*cos(th2))/25 - (6\*sin(th1))/5 - (33\*sin(th1)\*sin(th3))/50 + (33\*cos(th1)\*cos(th2)\*cos(th3))/50 |
| 0 | 0 | 0 | 1 |

Resulting Transform Matrix for Thumb with d values substituted

1. Fingers and Thumb Jacobian

| - d3\*(cos(th1)\*sin(th2) + cos(th2)\*sin(th1)) - d2\*sin(th1) | - d3\*(cos(th1)\*sin(th2) + cos(th2)\*sin(th1)) - d2\*sin(th1) | -d3\*(cos(th1)\*sin(th2) + cos(th2)\*sin(th1)) |
| --- | --- | --- |
| d1 + d3\*(cos(th1)\*cos(th2) - sin(th1)\*sin(th2)) + d2\*cos(th1) | d3\*(cos(th1)\*cos(th2) - sin(th1)\*sin(th2)) + d2\*cos(th1) | d3\*(cos(th1)\*cos(th2) - sin(th1)\*sin(th2)) |
| 0 | 0 | 0 |
| 0 | 0 | 0 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |

Jacobian Matrix for the Fingers

| (24\*sin(th2))/25 + (33\*cos(th3)\*sin(th2))/50 | (24\*cos(th1)\*cos(th2))/25 - (6\*sin(th1))/5 - (33\*sin(th1)\*sin(th3))/50 + (33\*cos(th1)\*cos(th2)\*cos(th3))/50 | cos(th2)\*((24\*cos(th1)\*cos(th2))/25 - (33\*sin(th1)\*sin(th3))/50 + (33\*cos(th1)\*cos(th2)\*cos(th3))/50) + cos(th1)\*sin(th2)\*((24\*sin(th2))/25 + (33\*cos(th3)\*sin(th2))/50) |
| --- | --- | --- |
| (6\*cos(th1))/5 + (24\*cos(th2)\*sin(th1))/25 + (33\*cos(th1)\*sin(th3))/50 + (33\*cos(th2)\*cos(th3)\*sin(th1))/50 | 0 | cos(th1)\*sin(th2)\*((24\*cos(th2)\*sin(th1))/25 + (33\*cos(th1)\*sin(th3))/50 + (33\*cos(th2)\*cos(th3)\*sin(th1))/50) - sin(th1)\*sin(th2)\*((24\*cos(th1)\*cos(th2))/25 - (33\*sin(th1)\*sin(th3))/50 + (33\*cos(th1)\*cos(th2)\*cos(th3))/50) |
| 0 | - (6\*cos(th1))/5 - (24\*cos(th2)\*sin(th1))/25 - (33\*cos(th1)\*sin(th3))/50 - (33\*cos(th2)\*cos(th3)\*sin(th1))/50 | - cos(th2)\*((24\*cos(th2)\*sin(th1))/25 + (33\*cos(th1)\*sin(th3))/50 + (33\*cos(th2)\*cos(th3)\*sin(th1))/50) - sin(th1)\*sin(th2)\*((24\*sin(th2))/25 + (33\*cos(th3)\*sin(th2))/50 |
| 0 | 0 | sin(th1)\*sin(th2) |
| 0 | 1 | cos(th2) |
| 1 | 0 | cos(th1)\*sin(th2) |

Jacobian Matrix For Thumb kinematics