Team 205 – Robotic Arm to Assist Motorized Chair Users

Parent Project Muscular Dystrophy

- Industry Sponsor: PPMD, Mr. Keith Van Houten
- Faculty Advisor: Prof. Stephen Moyer
- Team Lead: Brooke Harrington
- Team Members: Cavan Moriarty, Omar Abuljobain, Justin Lee, and Andrew Baum

VIRGINIA TECH

Agenda

Background and recap

- Background, Problem/Context, Value Proposition
- Solution Diagram

Solution Summary

- High Fidelity CAD
- Key/ Important Features
- How Solution Meets Objectives

Status of Obj. and Req.

- System Level Objectives – Derived Requirements and Verification Method
- Current Statuses of Objectives and Requirements

Critical Design

- Engineering analysis and calculations
- Drawings/Graphics for Important Components
- Test Plans and Bill of Materials

Project Management

- Updated Risk/Mitigated Risks
- Project Timeline and Budget

Background on Duchenne Muscular Dystrophy (DMD)

- DMD is a progressive disorder that deteriorates muscles over time
  - 1 in 5000 boys affected yearly
- Users with DMD require the use of motorized wheelchairs such as the Permobil M3 and Quantum
- People with DMD have limited range of motion (ROM) and often require a caretaker **assistive technologies** to help with everyday tasks

Figure 1: Robotic Arm Mounted on Wheelchair

Figure 2 Permobil M3 Corpus

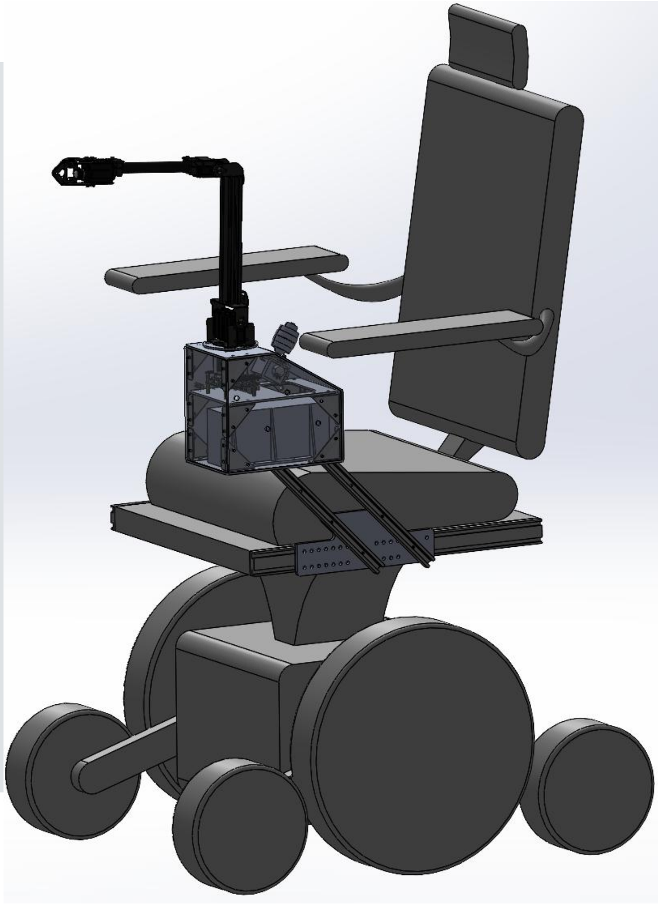
Problems with the Current ROM Assistive Robot

As mentioned by our customers and users, current solutions such as the Kinova JACO robotic arm are:

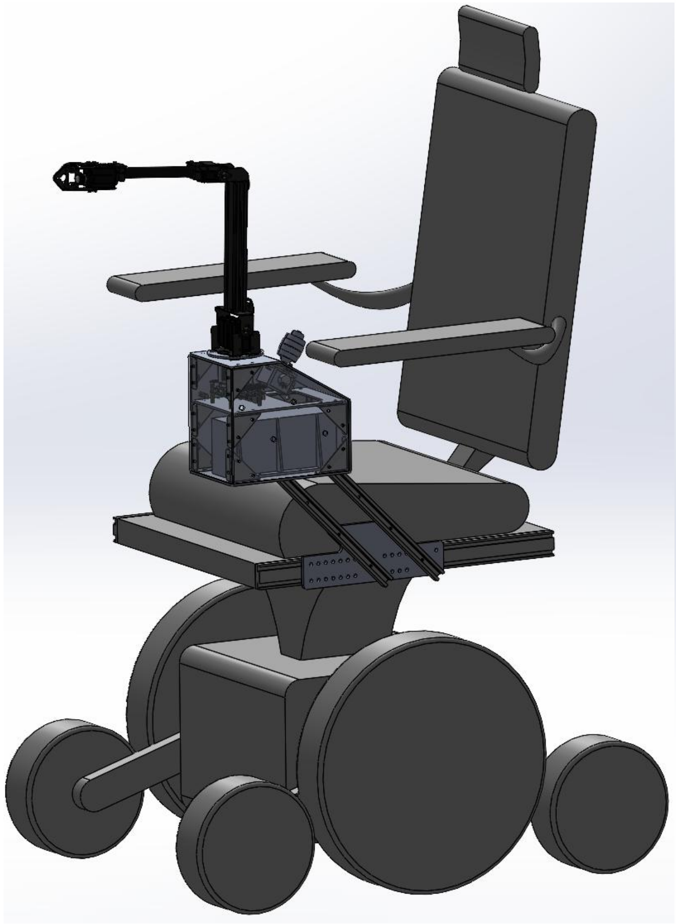
- Too Expensive (>$60,000)
- Difficult to Control
- Large/Bulky
- Difficult to remove for travel

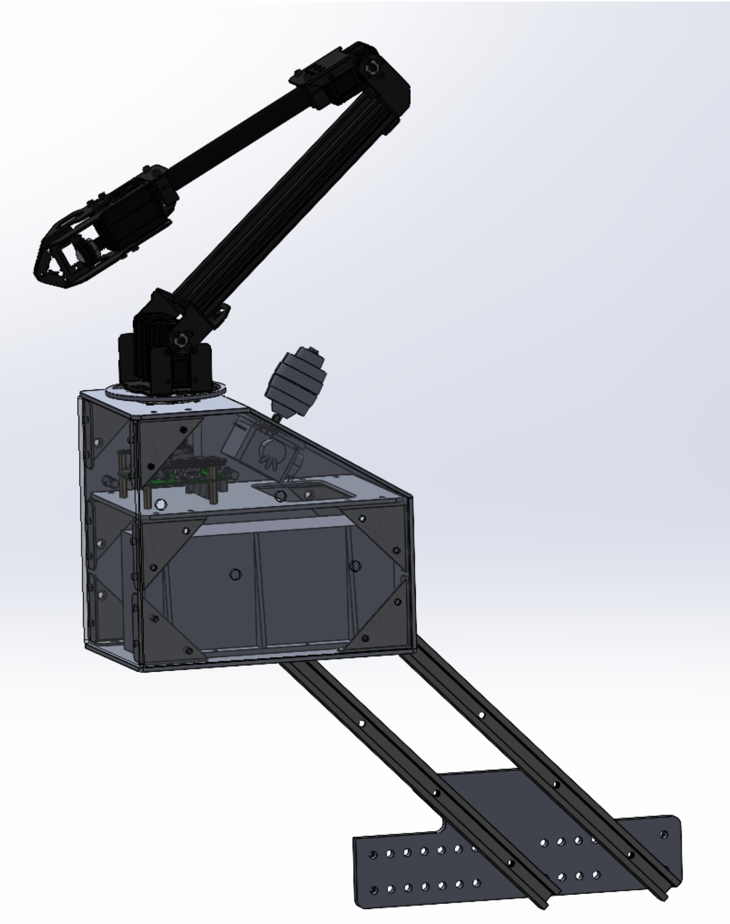
Team 205 has been tasked with developing an affordable, simple to control robotic arm that will accomplish two high value functions: pushing handicap door and elevator buttons, and bringing a cup to the user's face

Figure 3: Kinova JACO robotic arm



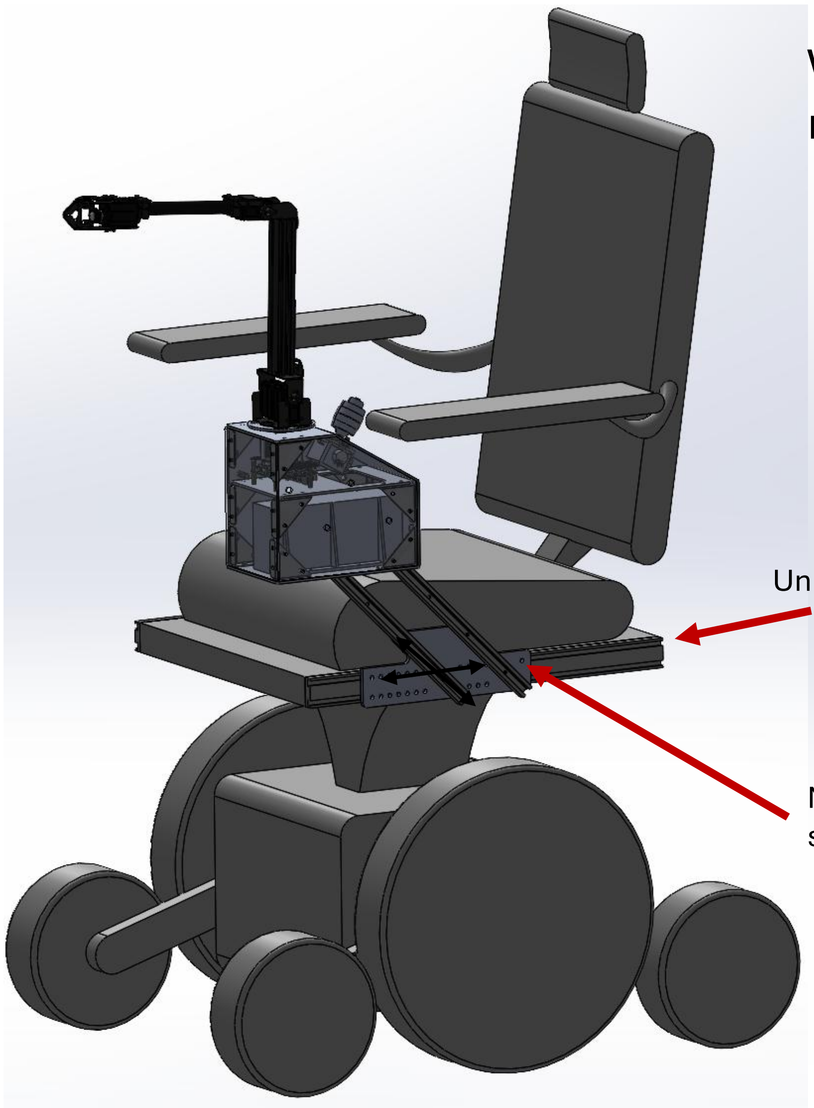
For Individuals with DMD who require enhanced ROM Assistance, an Affordable Robotic Aid Would increase independence with Everyday Tasks

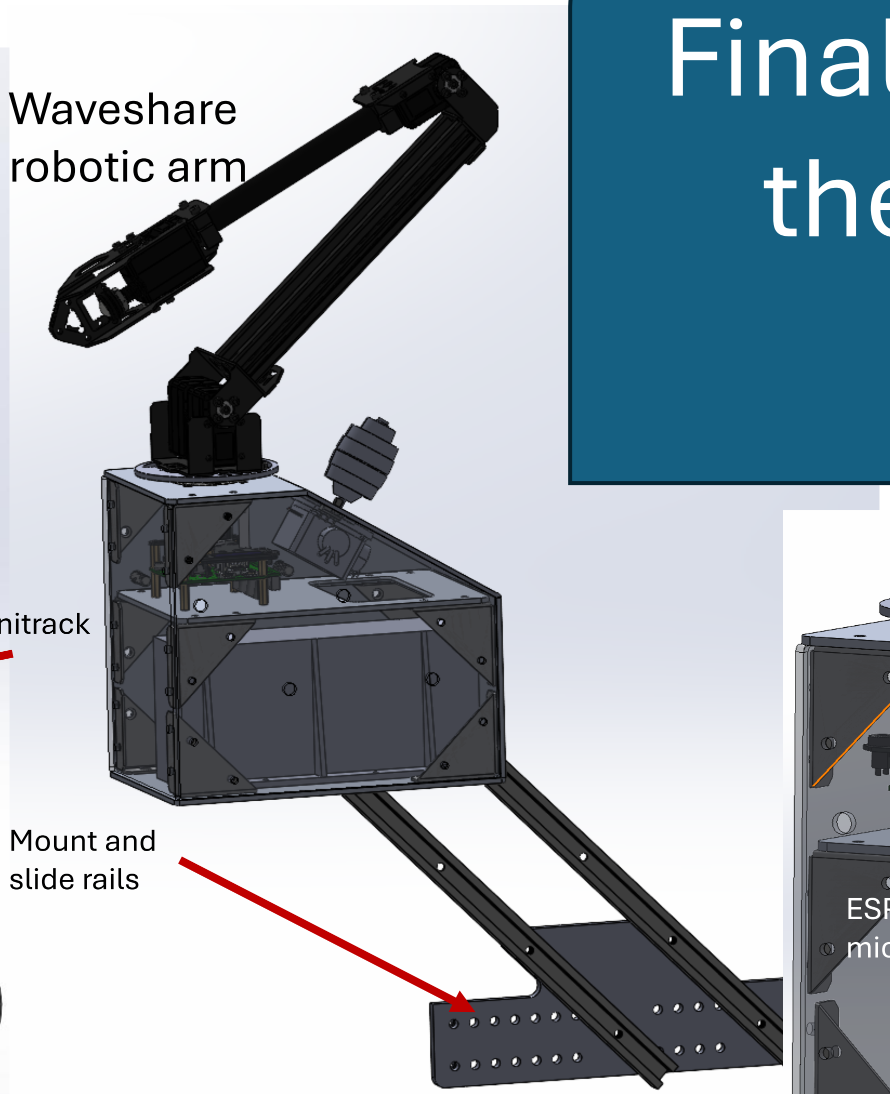


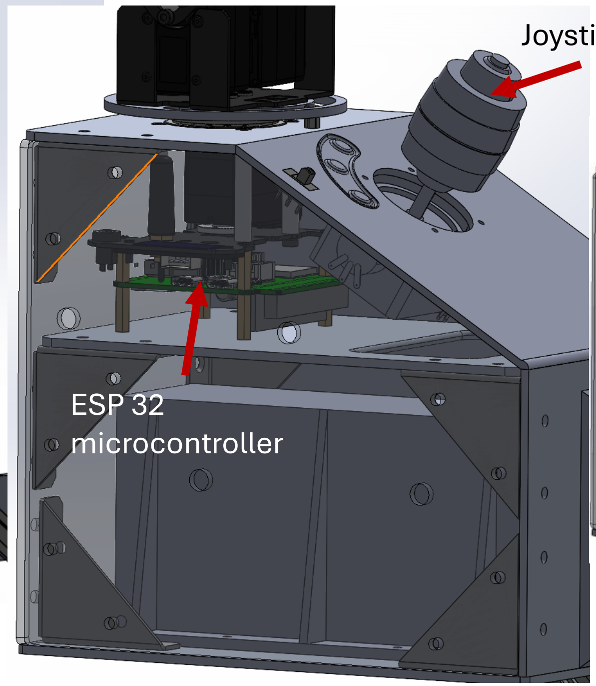


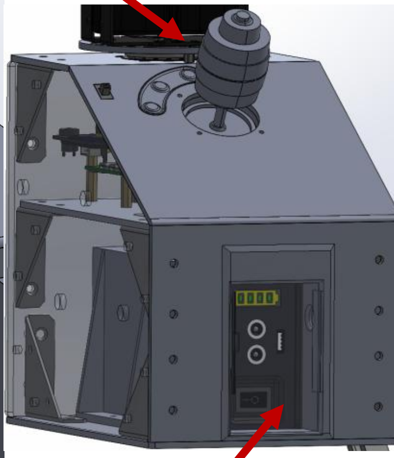
- Perform functions of high value to the user such as pushing a handicap door button
- Ease of use for the enhanced ROM capabilities
- Integrate robotic manipulation capabilities onto powered wheelchairs at an affordable price (<$5000)

Finalized CAD for the Case and Mount

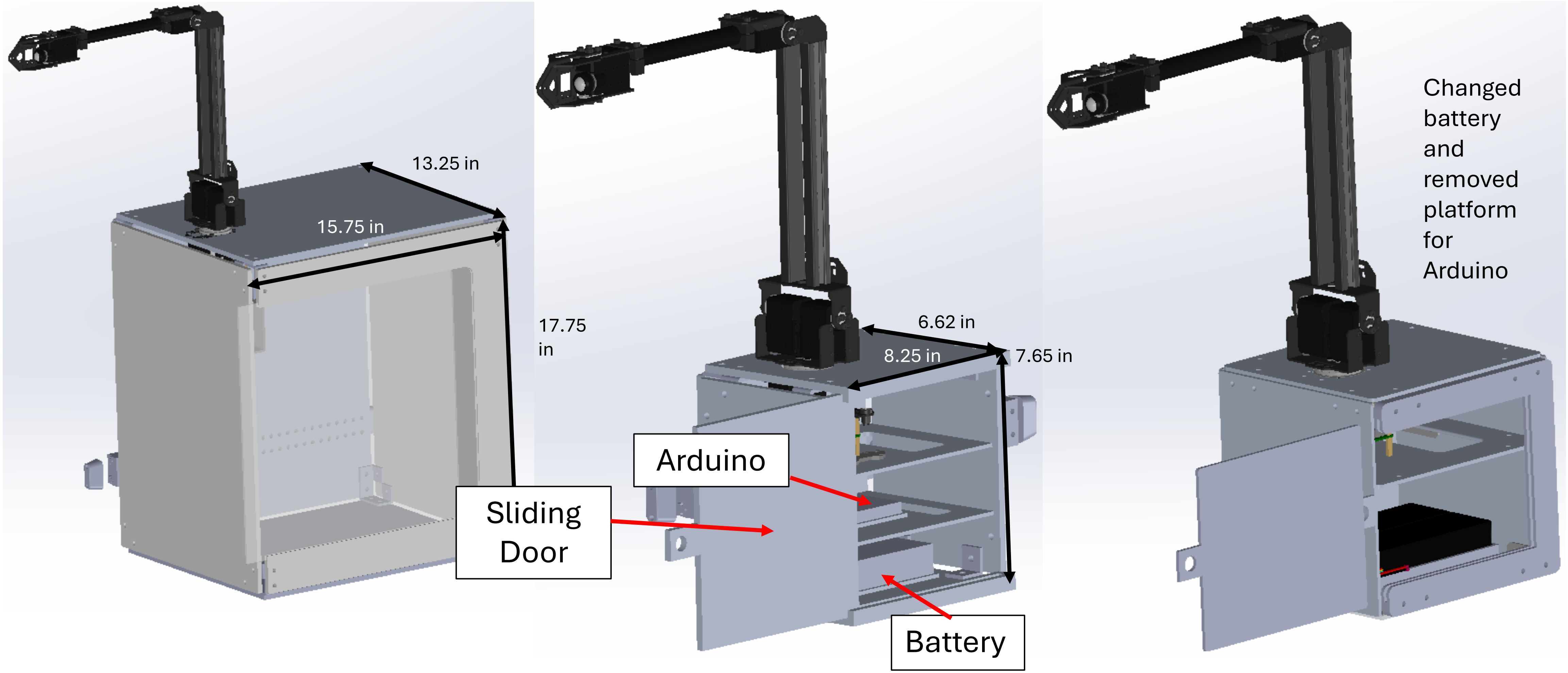




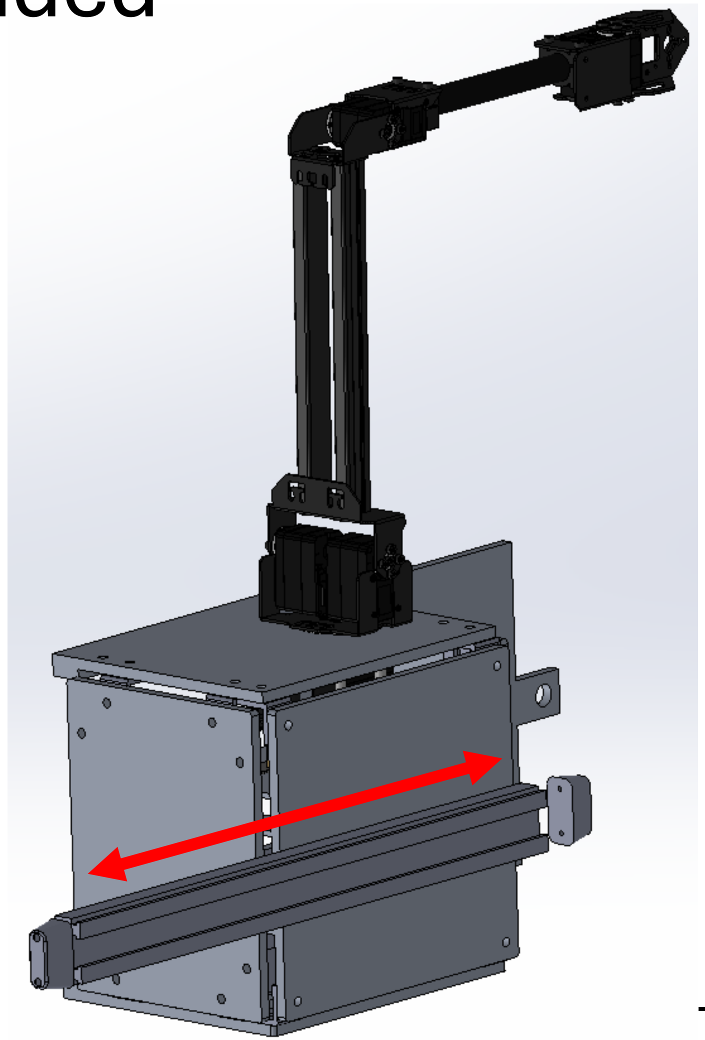




Since PDR: Refined structural and component containment design

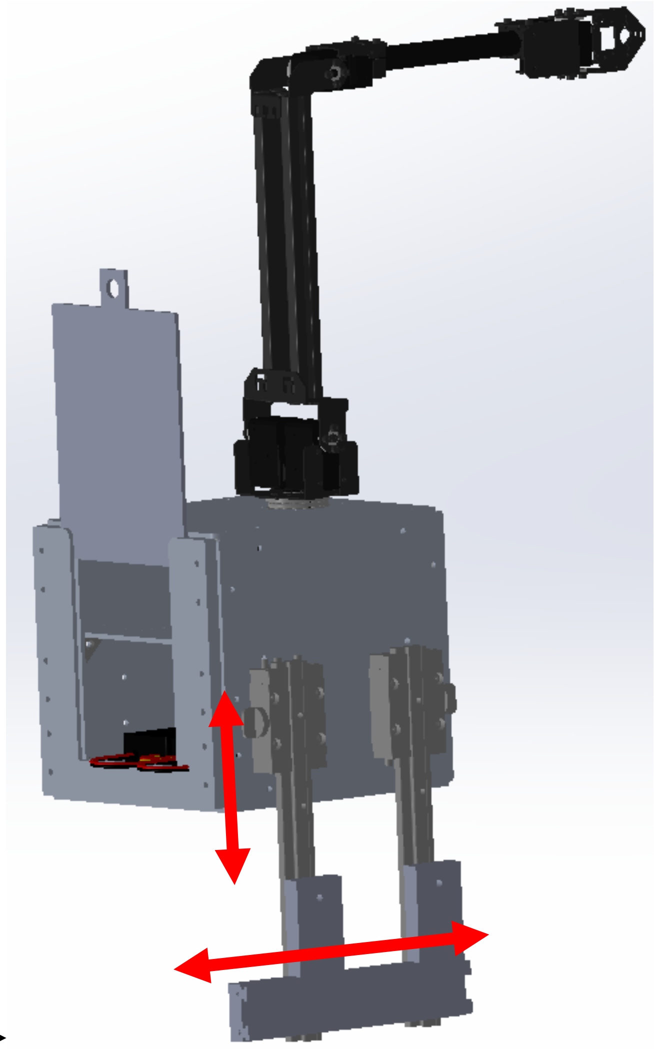


Progression of CAD since PDR – Vertical Adjustment Added

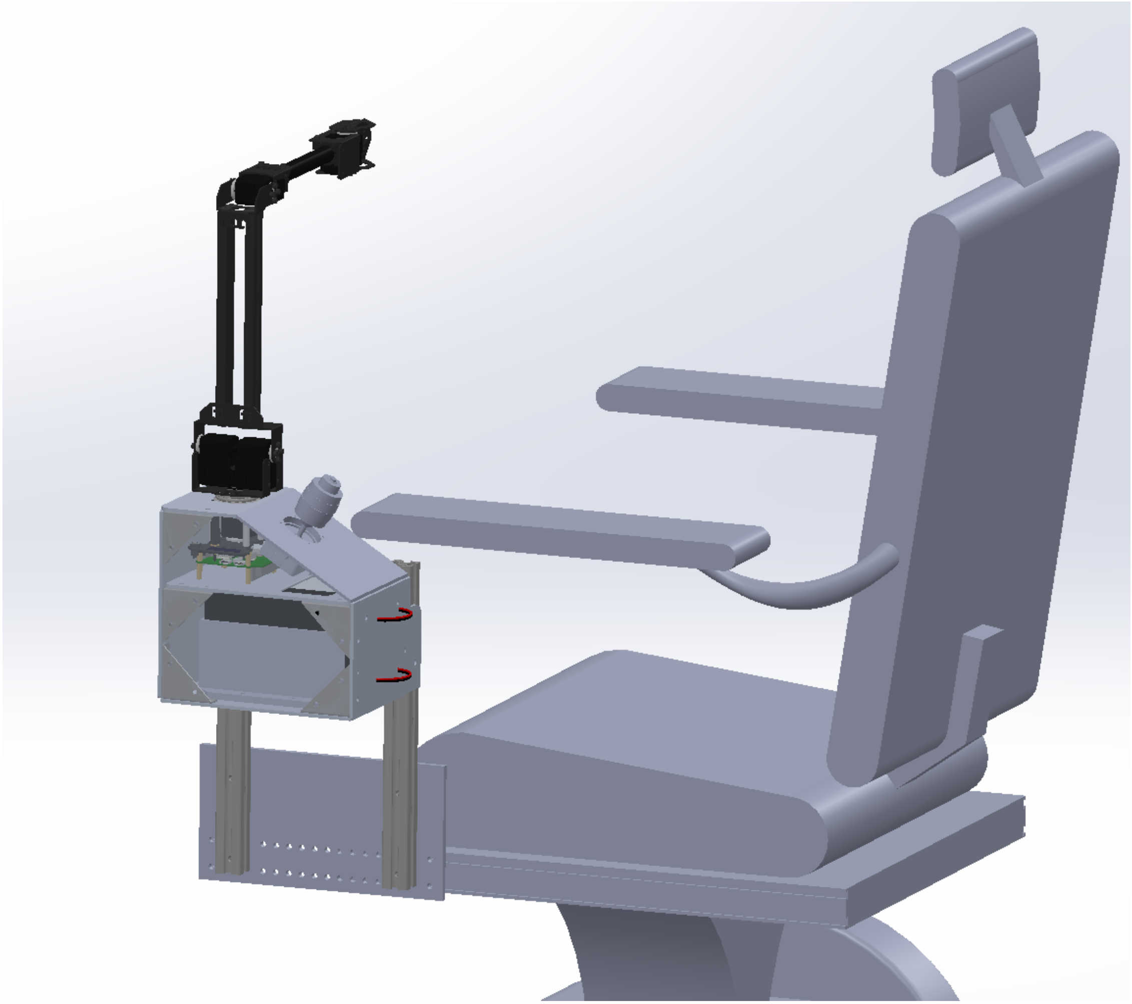


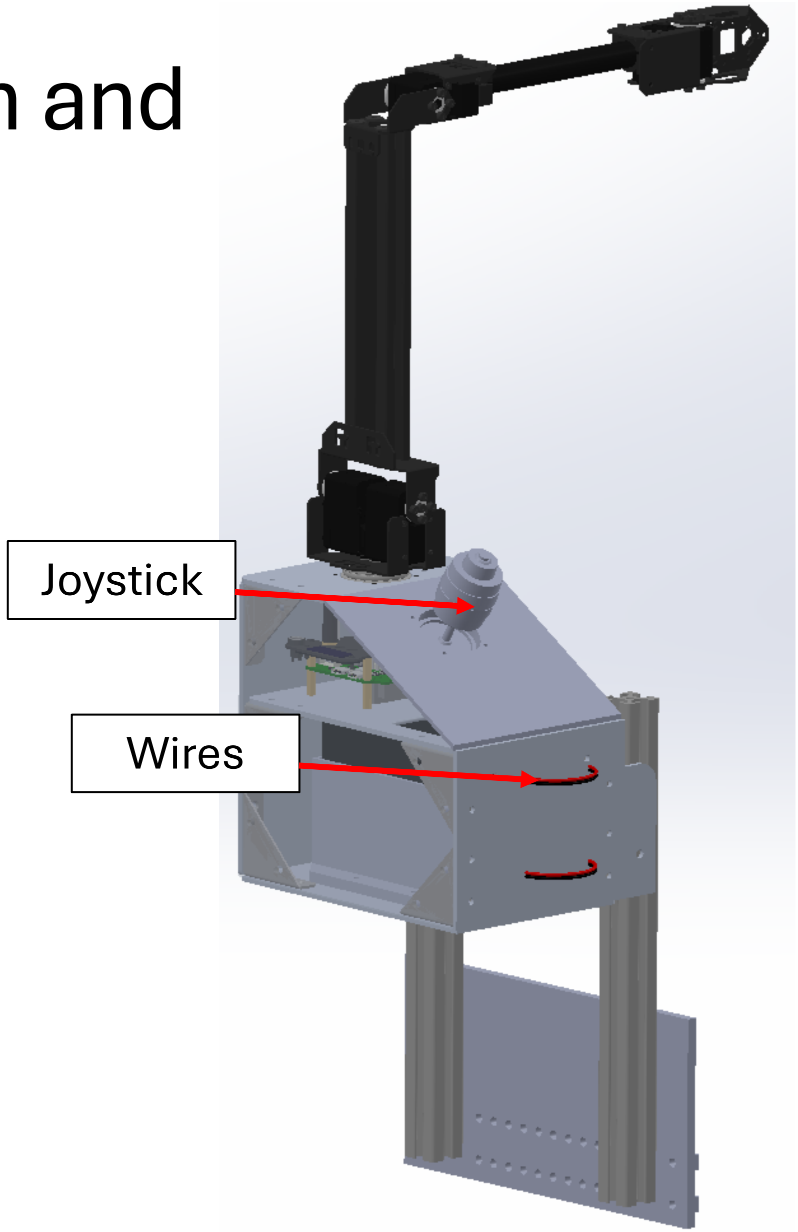
Added vertical adjustment in addition to the horizontal adjustment – adjusted when installed to user preference

Changed the location of the access door to allow the case to be mounted on either side of the wheelchair

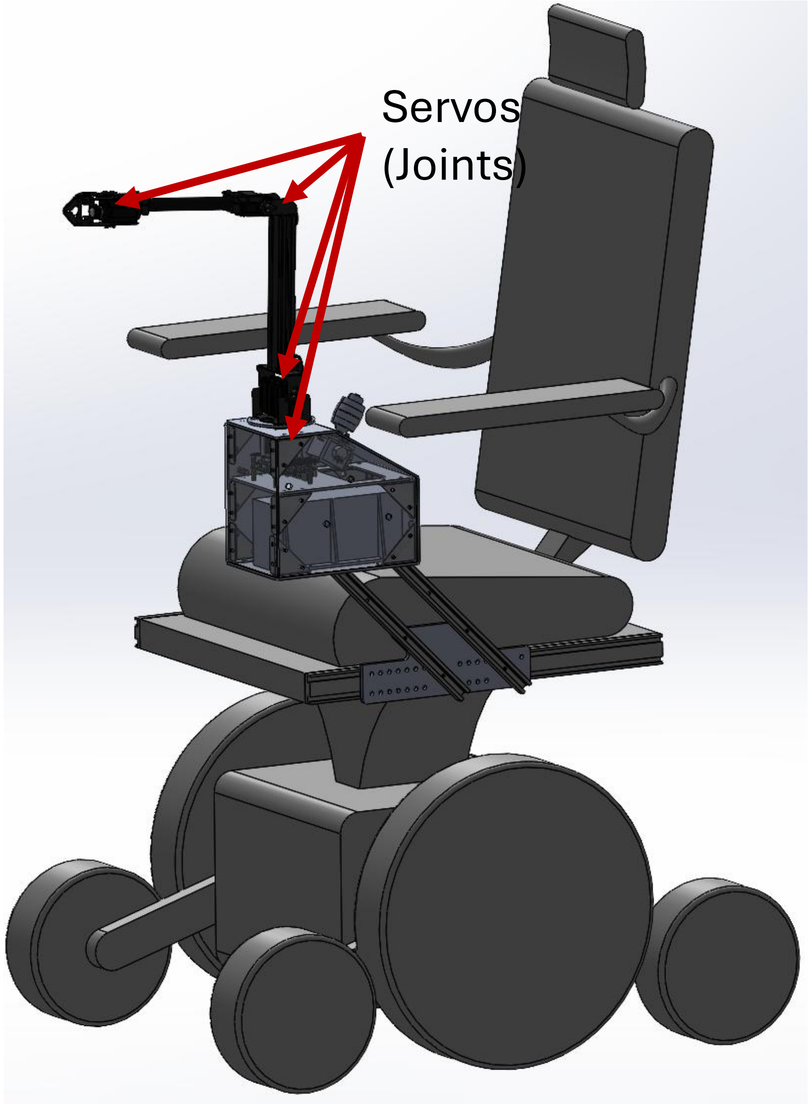


Addition of the Joystick after Decision and Redesign of the Vertical Adjustment

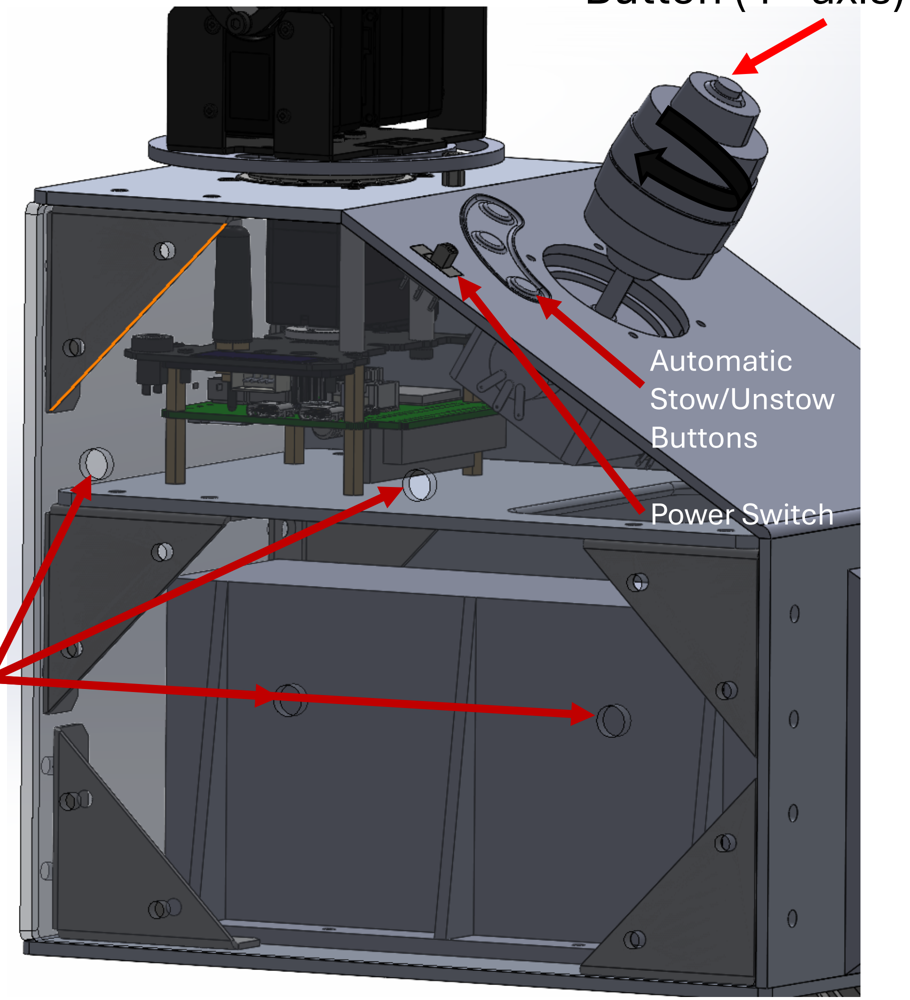




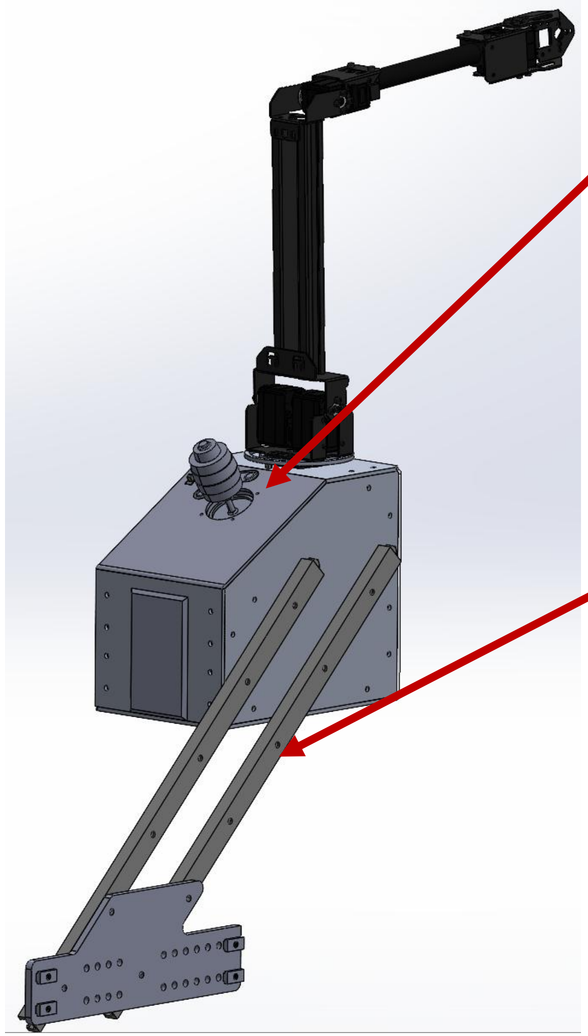
Finalized CAD – Key Features



- 4 DOF robotic arm – simple motion means simple control for users with limited dexterity in hands
  - 4 axis joystick – ease of control for end users
- Holes for mounting on either side – user request based on dominant hand

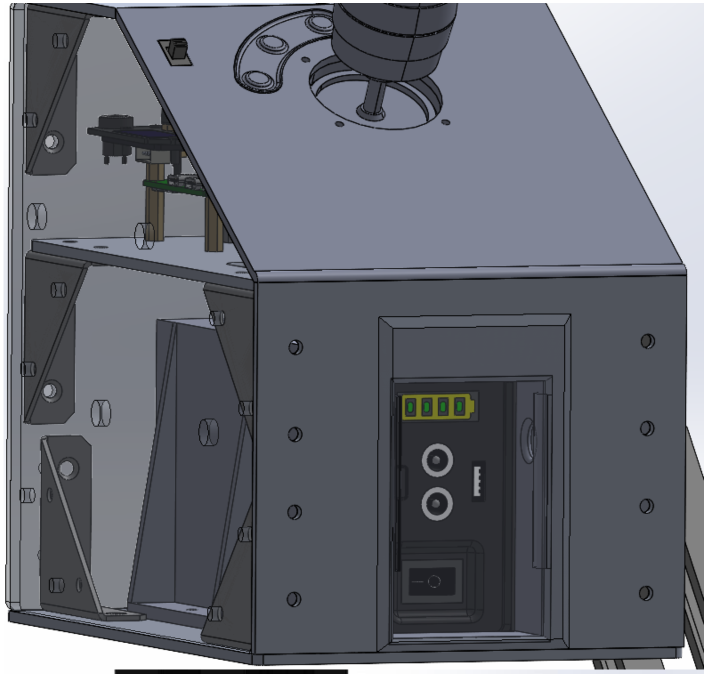


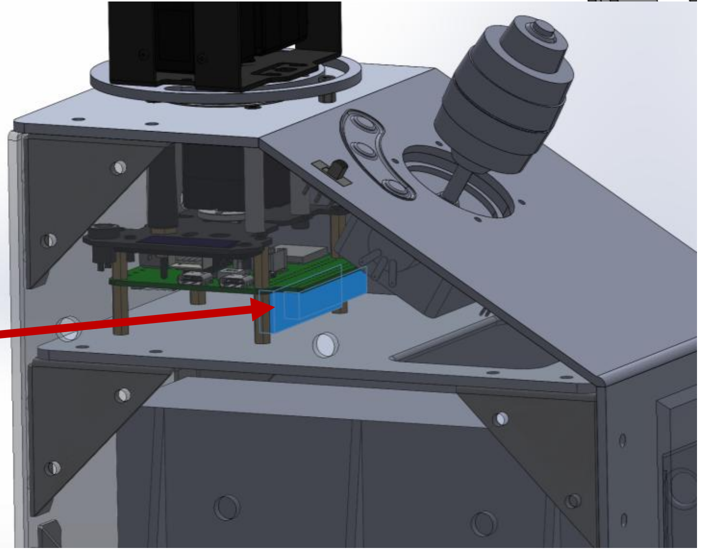
Finalized CAD – Key Features



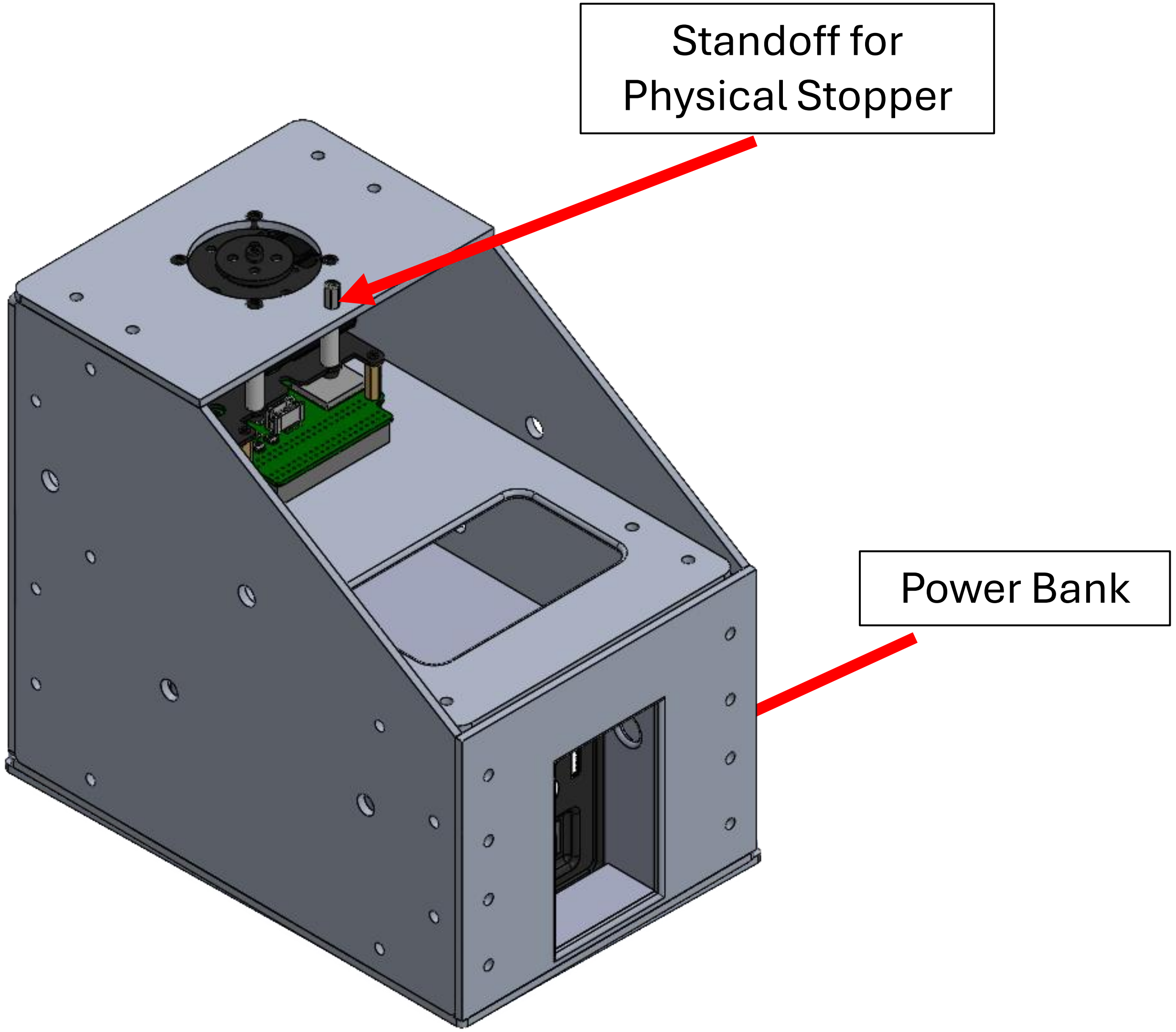
- Joystick mounted at an angle to resemble other joysticks on motorized wheelchairs
- Angled sliding rails that leave space under the case – users may have other assistive devices mounted here

- Power bank installed in the case for easy recharge – no need for battery removal per user request
- Turned the GPIO header to face the joystick – ease of wiring and maintenance

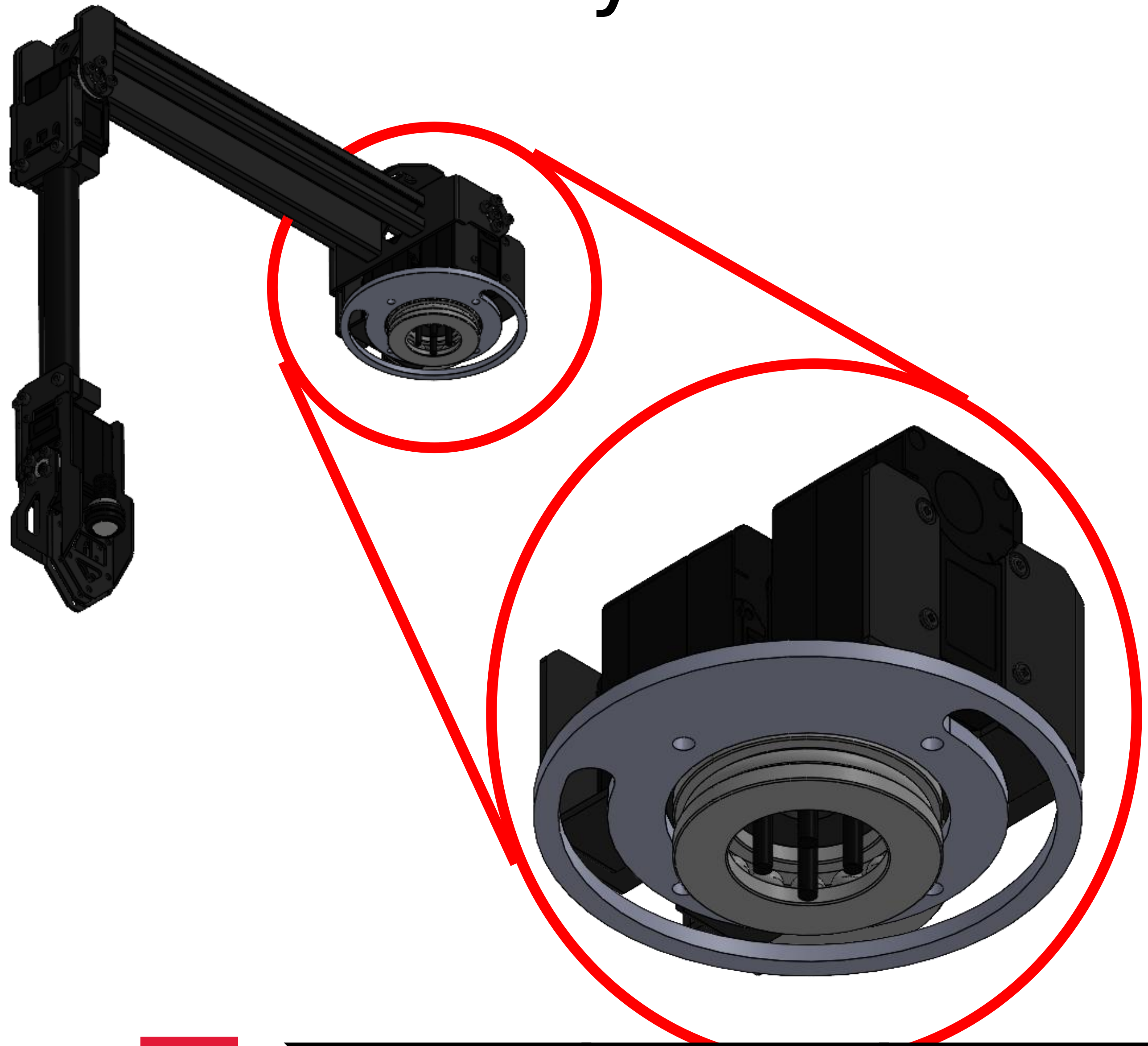


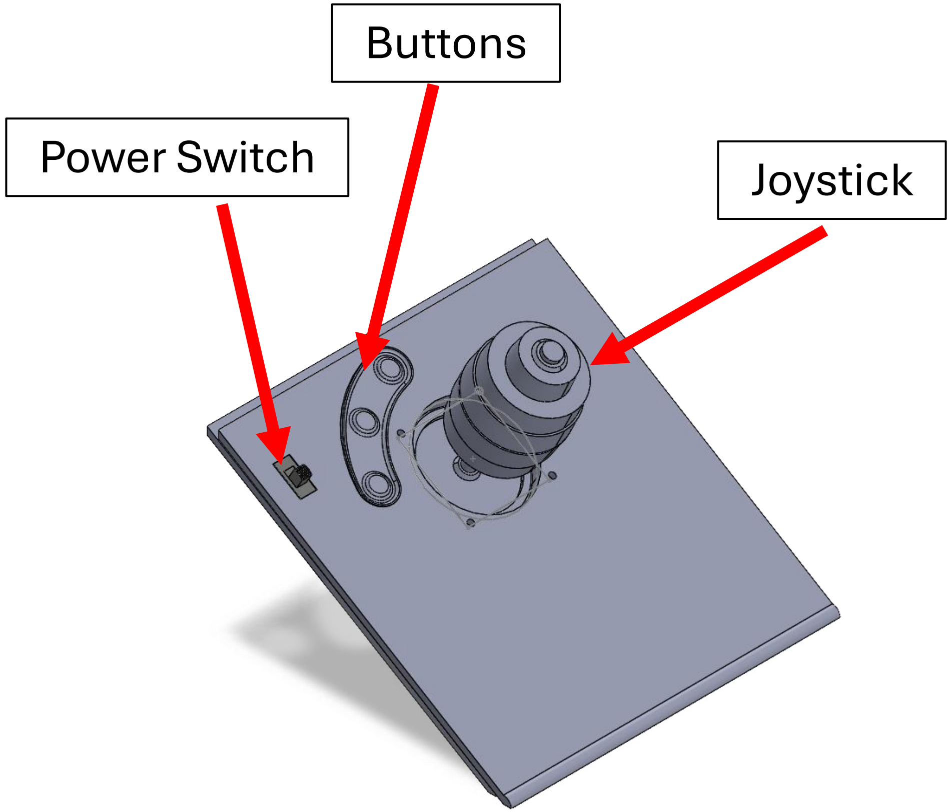


CAD Assembly

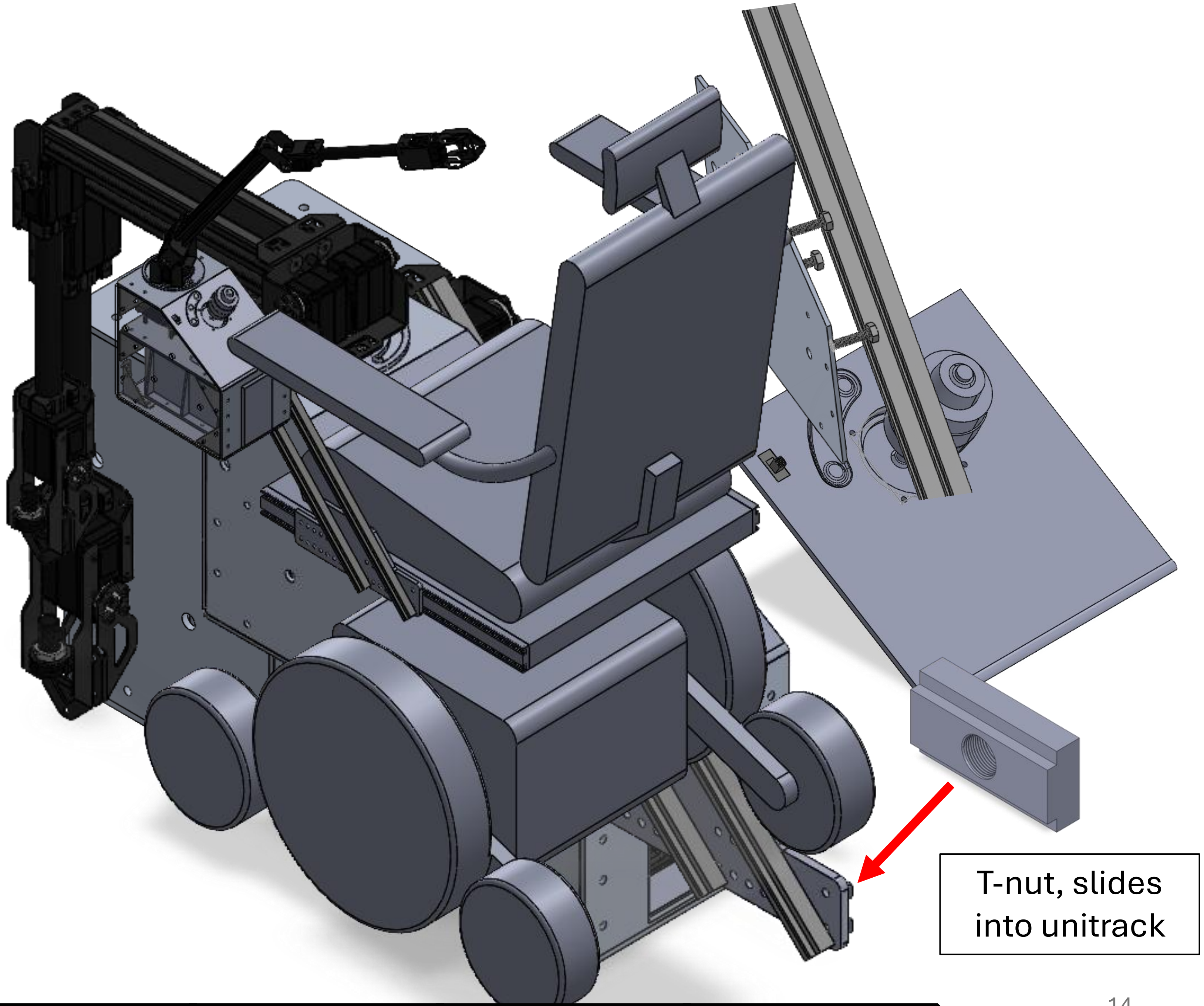


CAD Assembly



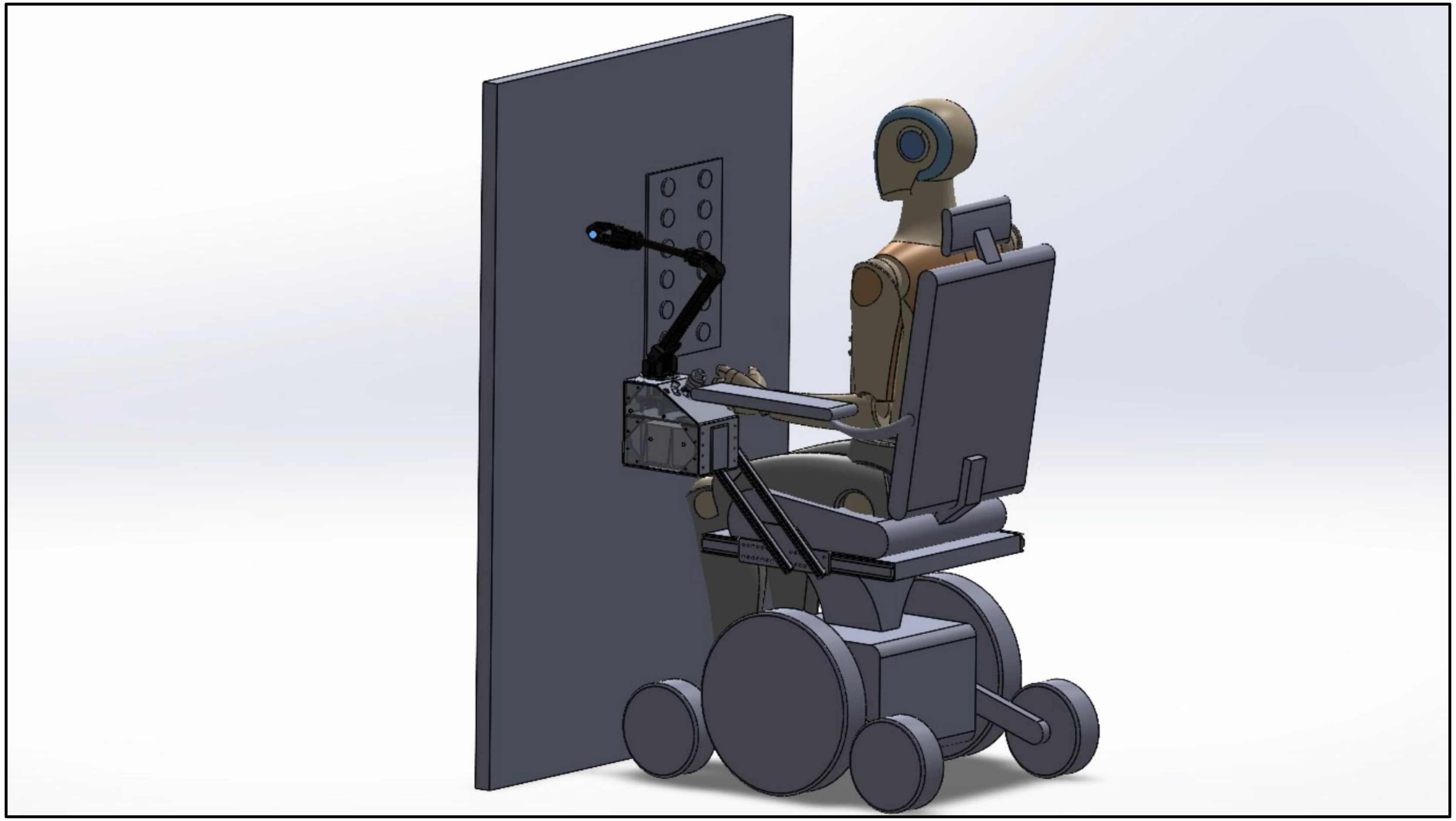


CAD Assembly



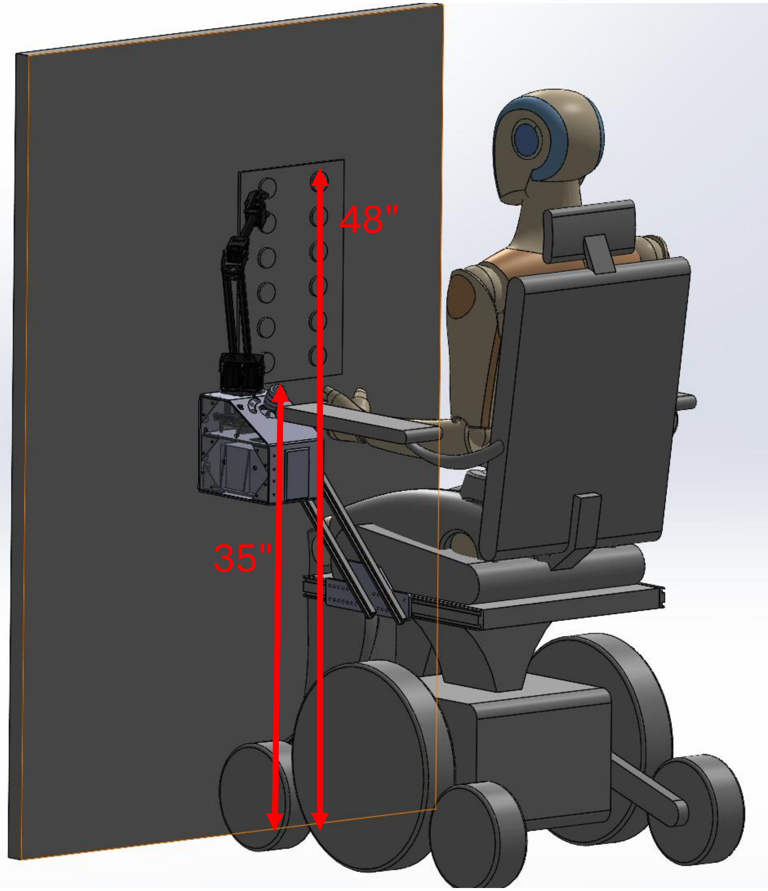
System Level Objectives

Demonstration of robotic arm mapping distance on wall

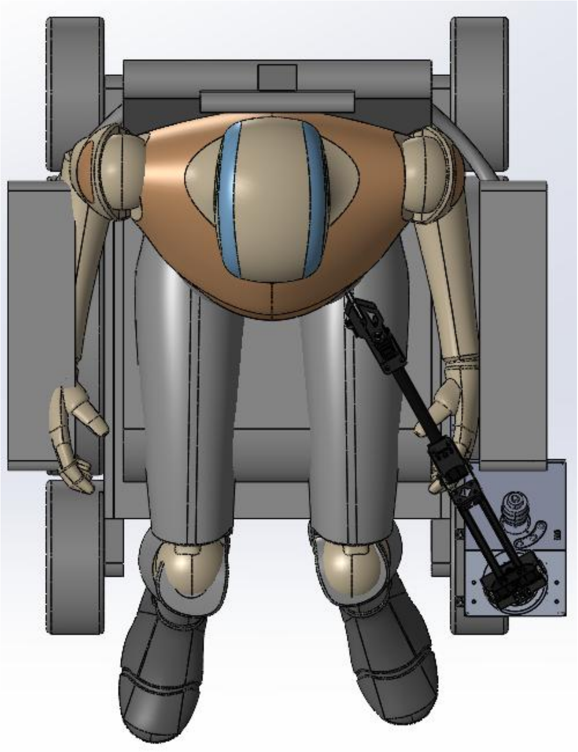


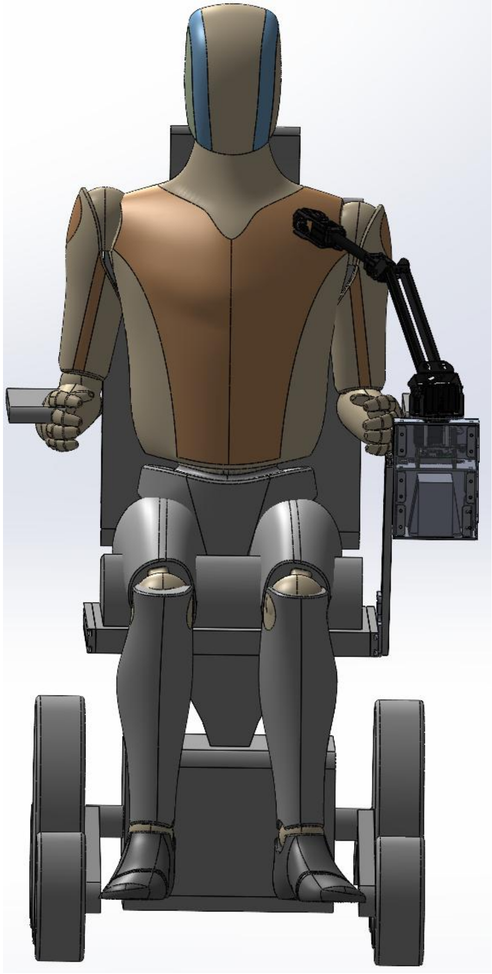
System Level Objectives

- The system will be able to depress an elevator button complaint with ADA standards found in [chapter 4 §407.4.6]



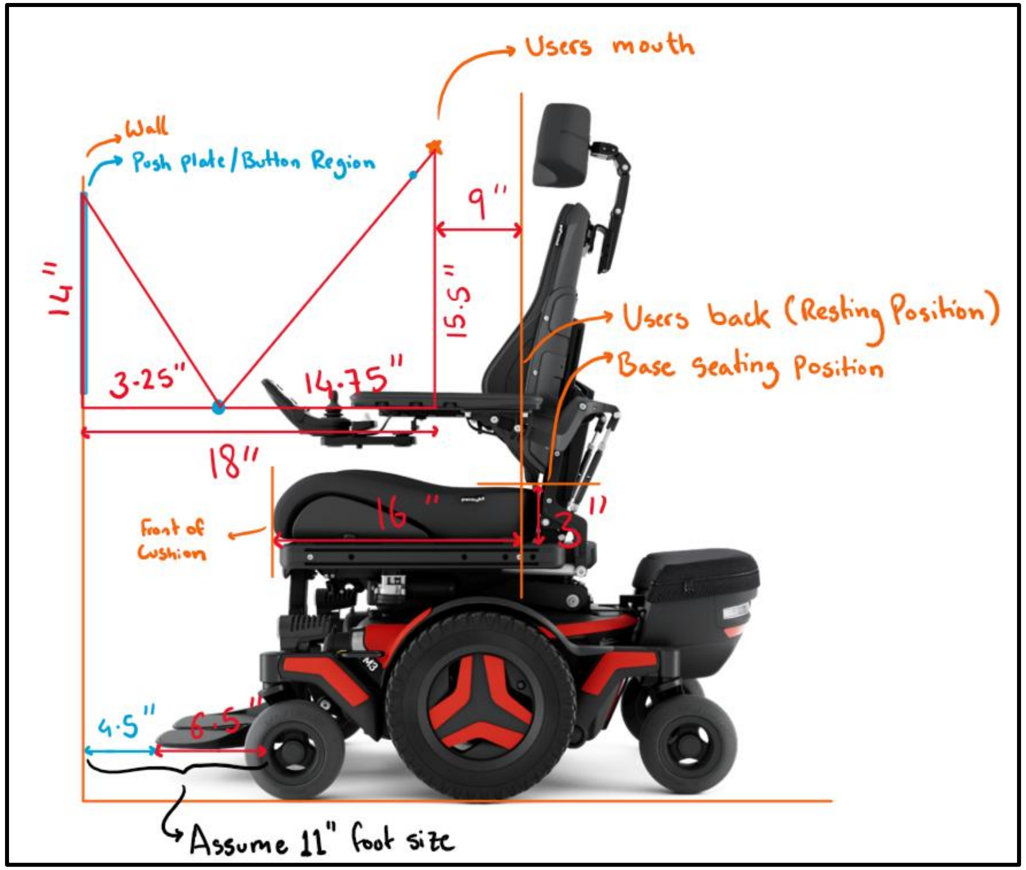
- Once the first objective is accomplished, the end effector will be modified to be able to bring a cup close to the users face for drinking with a straw

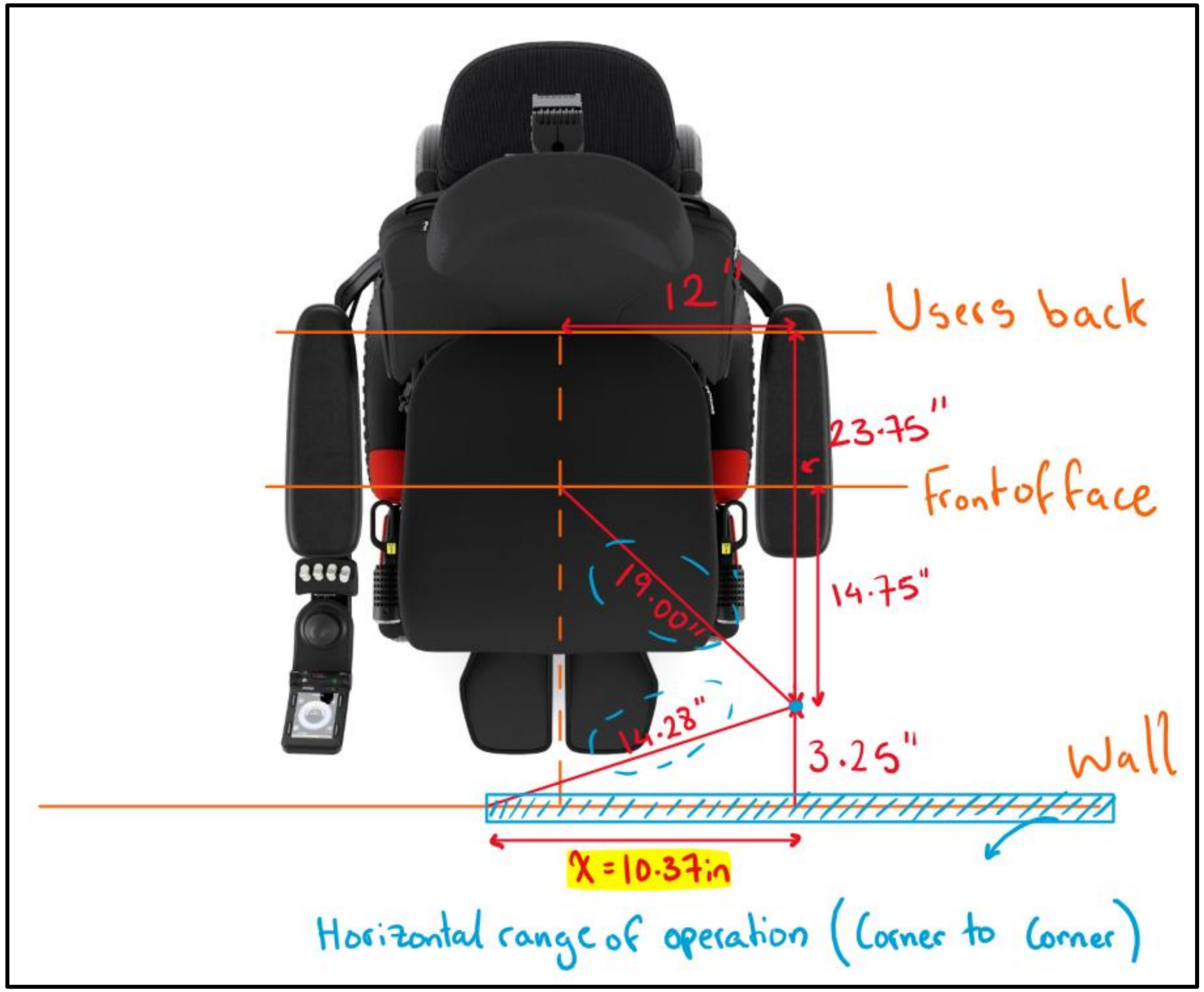




Variability to System Mapping (Directly Infront)

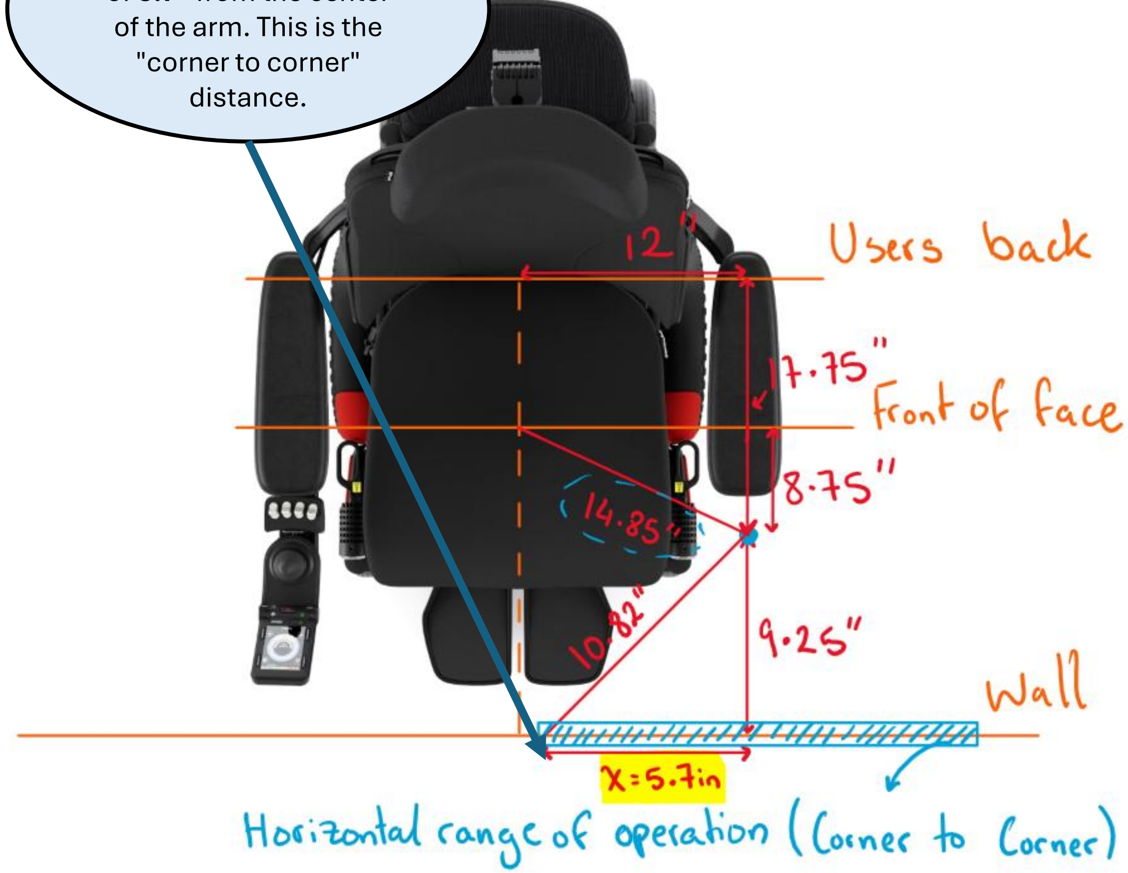
18" Armrest



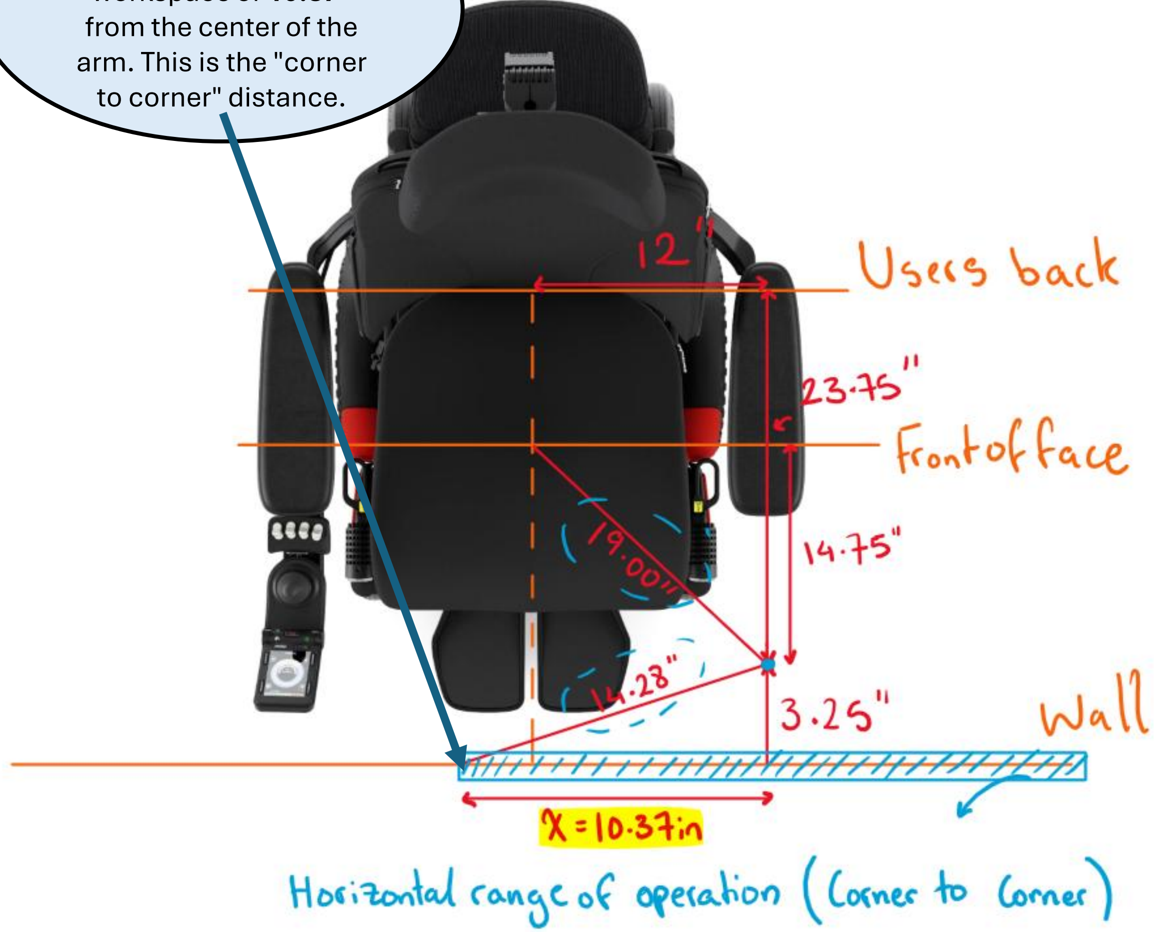


Variability to System Mapping (Directly infront)

A **13" armrest** provides a horizonal workspace of **5.7"** from the center of the arm. This is the "corner to corner" distance.

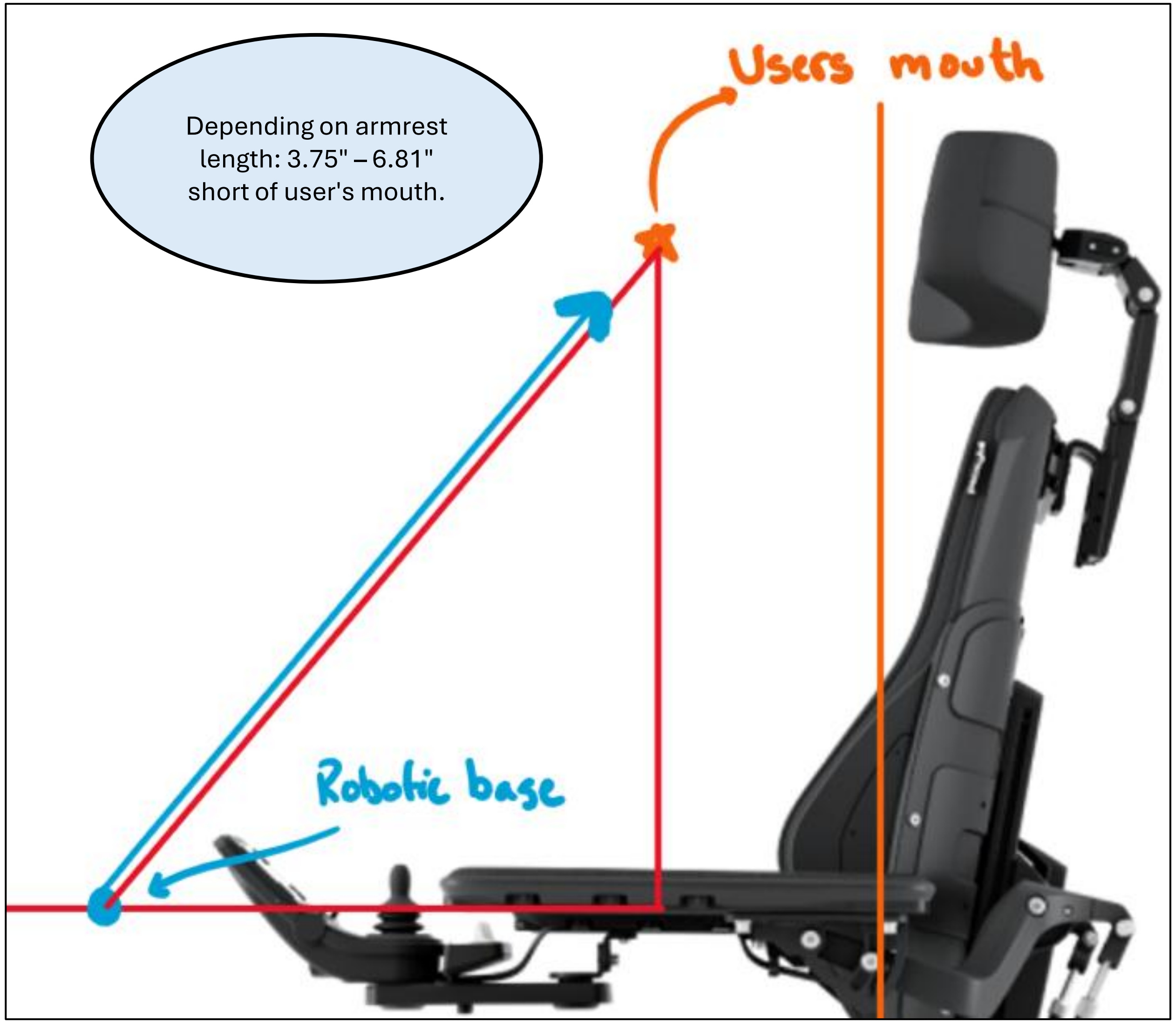


An **18" armrest** provides a horizonal workspace of **10.37"** from the center of the arm. This is the "corner to corner" distance.



Future Work Upon Primary Objective Completion

- Distance from robotic base to user's mouth: 21.47" for a 13" armrest, and 24.53" for an 18" armrest.
- Straw, extending from a custom cup, will provide extra distance.
- Secondary Objective:
    - Custom Cup
    - No tilt
    - 5" minimum extendable straw



Requirements Derived from System Level Objectives (1/2)

Technical Requirement ID	Requirement	Verification Method	Status
TR 1.1.0	The robot shall be controlled manually by a joystick and should have the ability to program automatic functions accessed through a button push	Test	In progress – test script started, test in spring
TR 1.2.0	The robot shall have no more than six degrees of freedom to reduce the complexity of controls	Inspection	Currently met with robot arm choice
TR 1.3.0	The robot shall have an operating radius that spans a minimum of 5 inches to a maximum of 2 feet from mount location with 180 degrees of rotation total	System Test	Test script written, test in spring
TR 1.4.1	The battery that powers the robot arm shall last 24 hours between charges	Analysis	Completed

Requirements Derived from System Level Objectives (2/2)

Technical Requirement ID	Requirement	Verification Method	Status
TR 1.3.1	The device should be able to reach and depress elevator buttons and the handicap door button between 35" and 48" as specified by ADA standards	Analysis	Completed
TR 1.3.2	The system shall have an accuracy of +- 10mm while at full extension	Analysis and test	Analysis Completed Test script written, test in spring
TR 3.0.0	The robot shall be installed in less than one hour and should be operational within thirty minutes of installation	Test	In progress – test script written, test in spring
TR 5.3.0	The robot shall have safeguards against unexpected de-energization	Inspection	In progress – implemented in code

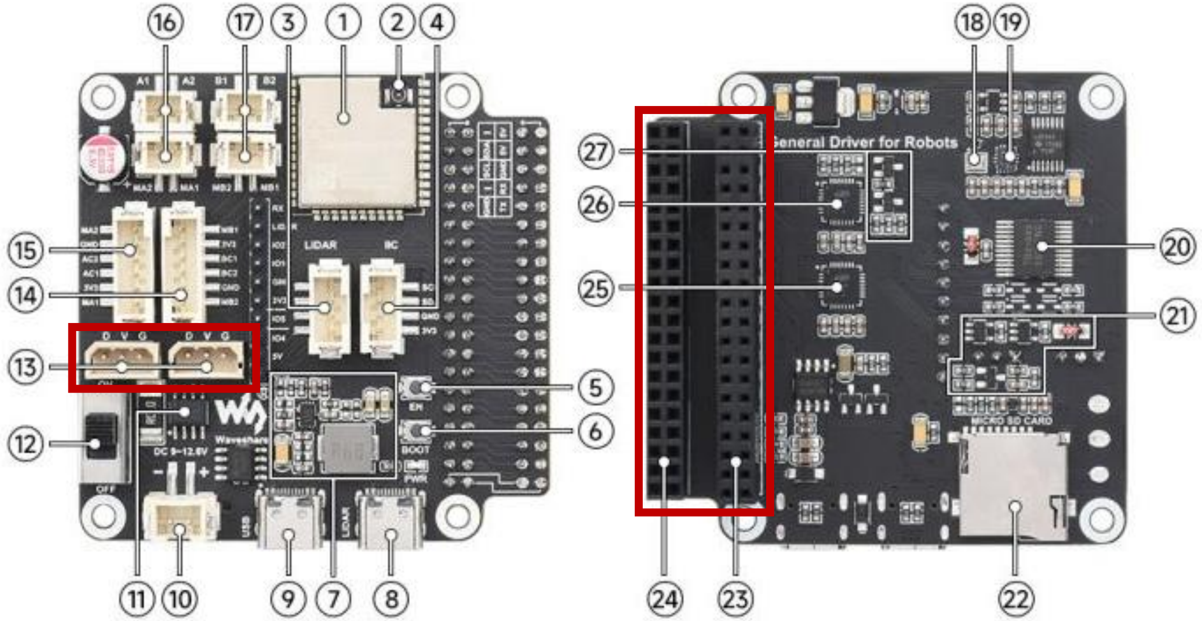
Test Plans for Spring Semester

TR 1.1.0

- **Description: Robot controlled manually by a joystick**

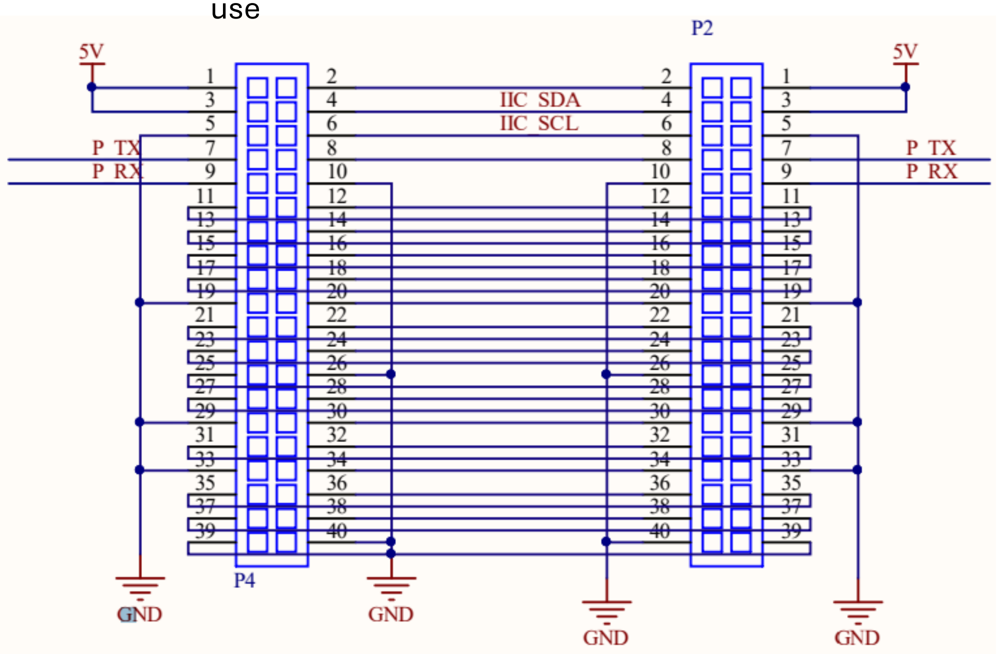
- Move the robot forward, backward, left, right, up, and down using the corresponding joystick movements and recording the value of the distance from the base of the arm to the end of the end effector
- **Equipment Used**: Waveshare robotic arm, 4-axis joystick, tape measure, computer

Electronics Interaction Between the Waveshare and the user

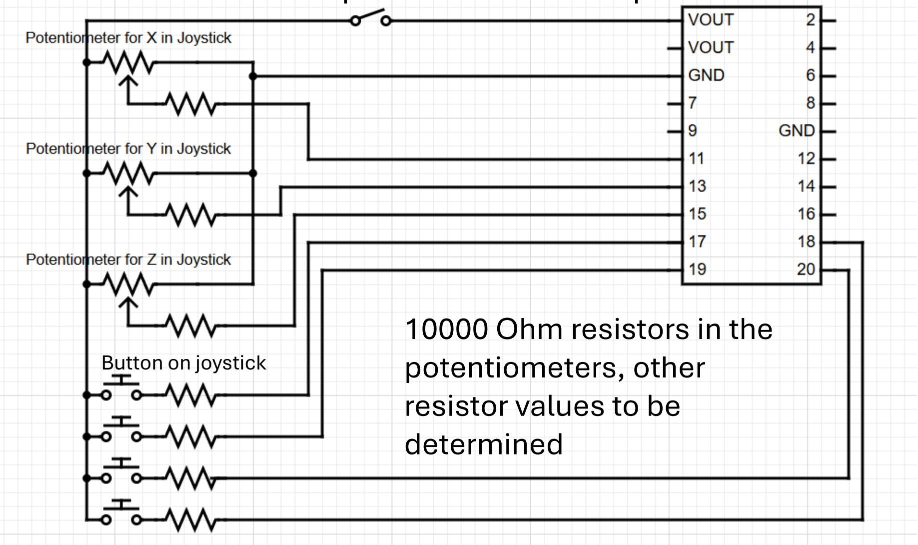


- The Waveshare robotic arm uses an ESP 32 microcontroller and additional drivers to control the robotic arm
- The motors are connected to the outputs labeled 13 which is the ST3215 bus servo interface – pass data (D), voltage (V), and ground (G)
- To connect the joystick and any additional buttons, we will make use of the general-purpose input/output (GPIO) pins, labeled 23 and 24 above, allowing the user to control the robotic arm

Layout of the GPIO pins – what pins we will use



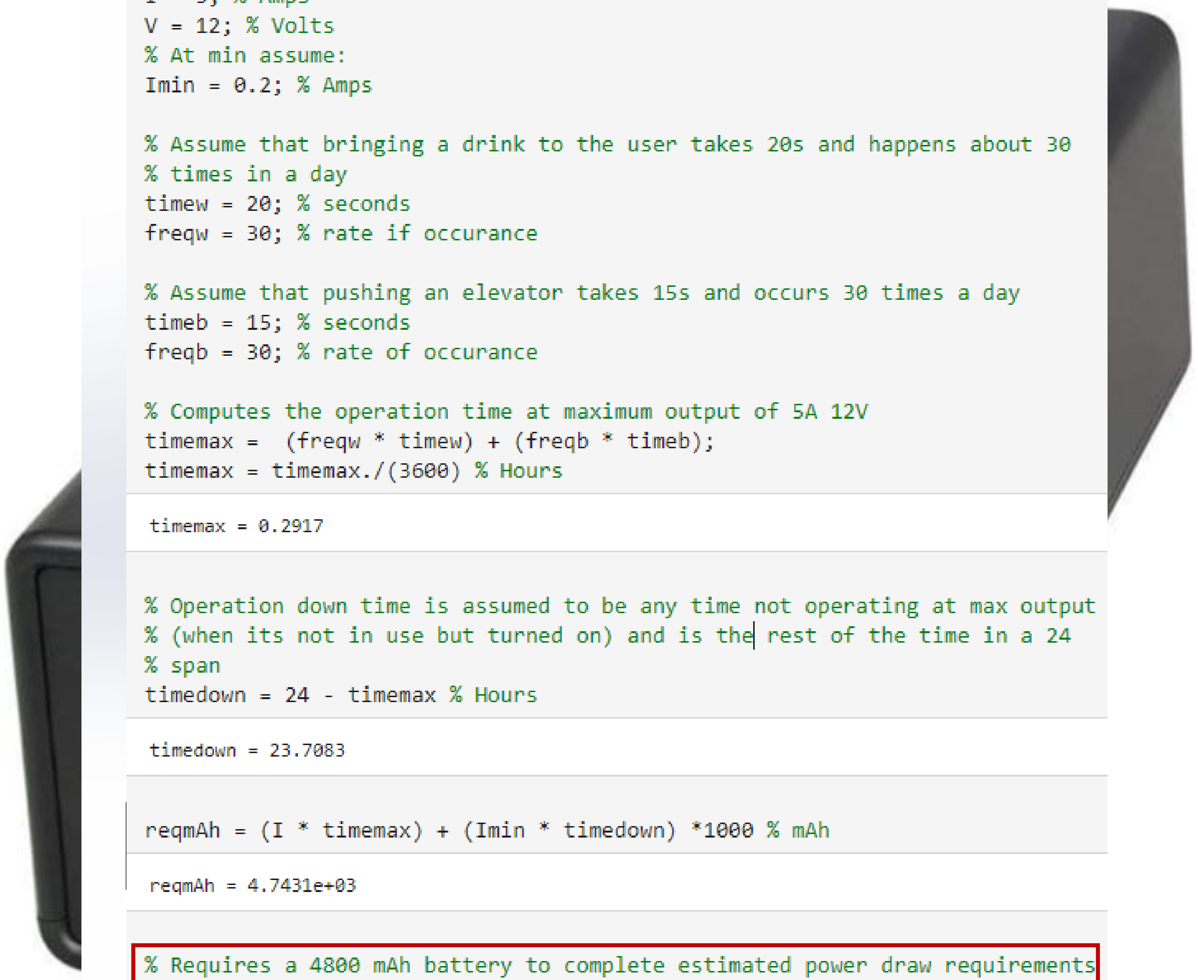
Switch to disconnect power – customer request



Automatic function buttons

Meeting Battery Requirements (TR 1.4 and 1.4.1-2)

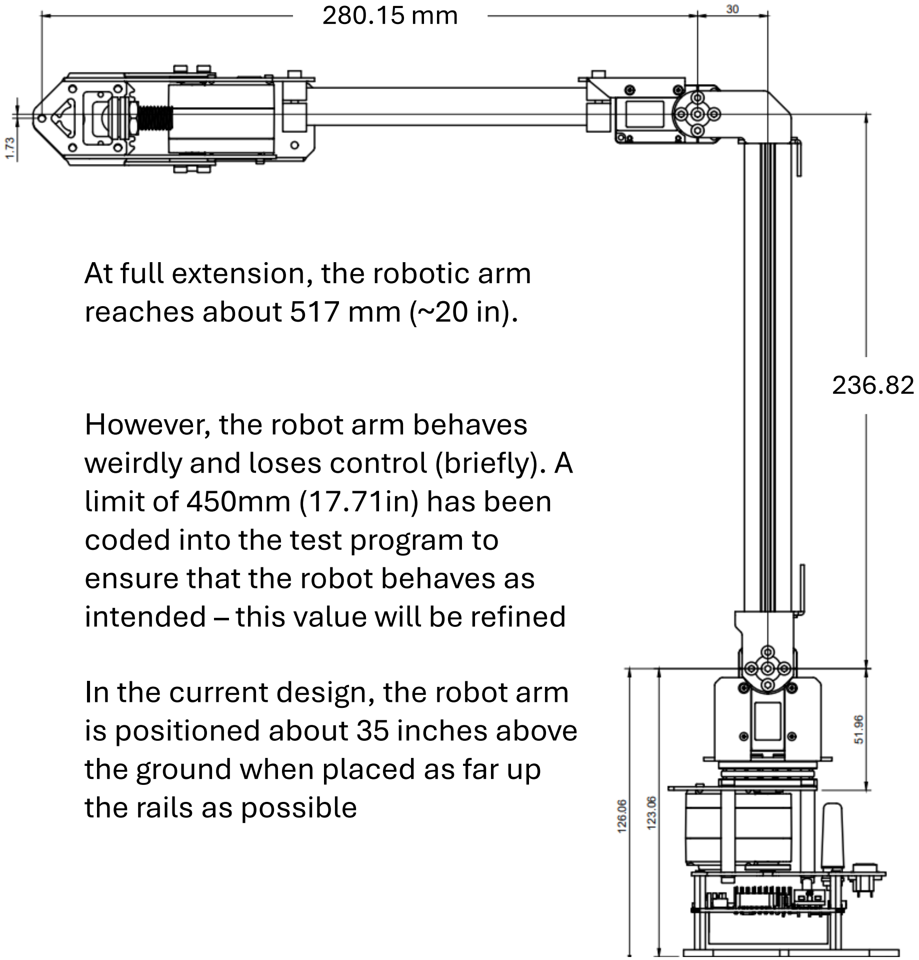
- The robotic arm should be independent of wheelchair battery and electronic systems (TR 1.4) – met with current design
- The robot should be able to operate a full day (24 hrs) before needing to be recharged (TR 1.4.1) - analysis
- The battery should remain installed in the case during charging (TR 1.4.2) – customer need – met with current design



```
% Refined calculation for battery requirements
% At max:
I = 5; % Amps
V = 12; % Volts
% At min assume:
Imin = 0.2; % Amps

% Assume that bringing a drink to the user takes 20s and happens about 30
% times in a day
timew = 20; % seconds
freqw = 30; % rate if occurance

% Assume that pushing an elevator takes 15s and occurs 30 times a day
timeb = 15; % seconds
freqb = 30; % rate of occurance

% Computes the operation time at maximum output of 5A 12V
timemax =  (freqw * timew) + (freqb * timeb);
timemax = timemax./(3600) % Hours

timemax = 0.2917

% Operation down time is assumed to be any time not operating at max output
% (when its not in use but turned on) and is the rest of the time in a 24
% span
timedown = 24 - timemax % Hours

timedown = 23.7083

reqmAh = (I * timemax) + (Imin * timedown) *1000 % mAh

reqmAh = 4.7431e+03

% Requires a 4800 mAh battery to complete estimated power draw requirements
% for one day, the team will likely purchase two to ensure battery is not
% completely drained before a 24 hr recharge requirement is up
```

- 12V 38400 mAh rechargeable lithium-ion battery
- Wall outlet for easy recharging – end user request
- Power switch for cutting power to the robot arm
- LED for battery status

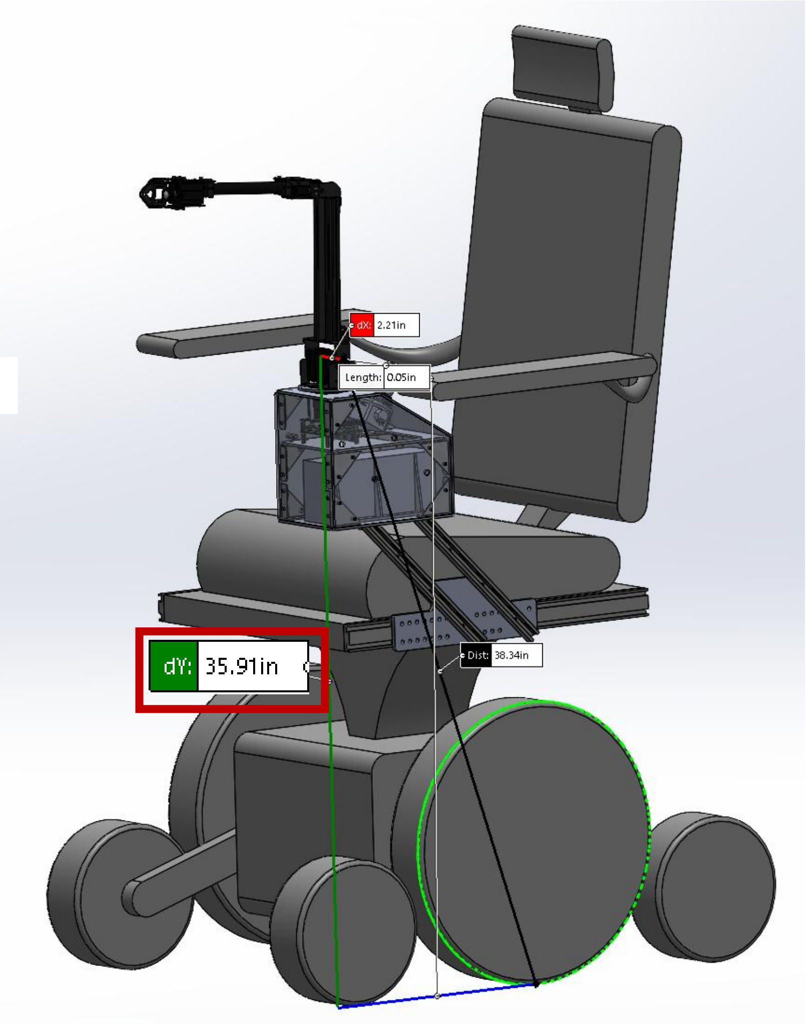
Accomplishing Pushing an Elevator Button – Meeting TR 1.3.1



At full extension, the robotic arm reaches about 517 mm (~20 in).

However, the robot arm behaves weirdly and loses control (briefly). A limit of 450mm (17.71in) has been coded into the test program to ensure that the robot behaves as intended – this value will be refined

In the current design, the robot arm is positioned about 35 inches above the ground when placed as far up the rails as possible



The base of the arm is nearly level with the lowest elevator button that complies with ADA standards in chapter 4 §407.4.6 at 35 inches

The top of the highest button compliant with ADA standards is 48 in above the ground, leaving about a 12 in span of buttons for the robot to press

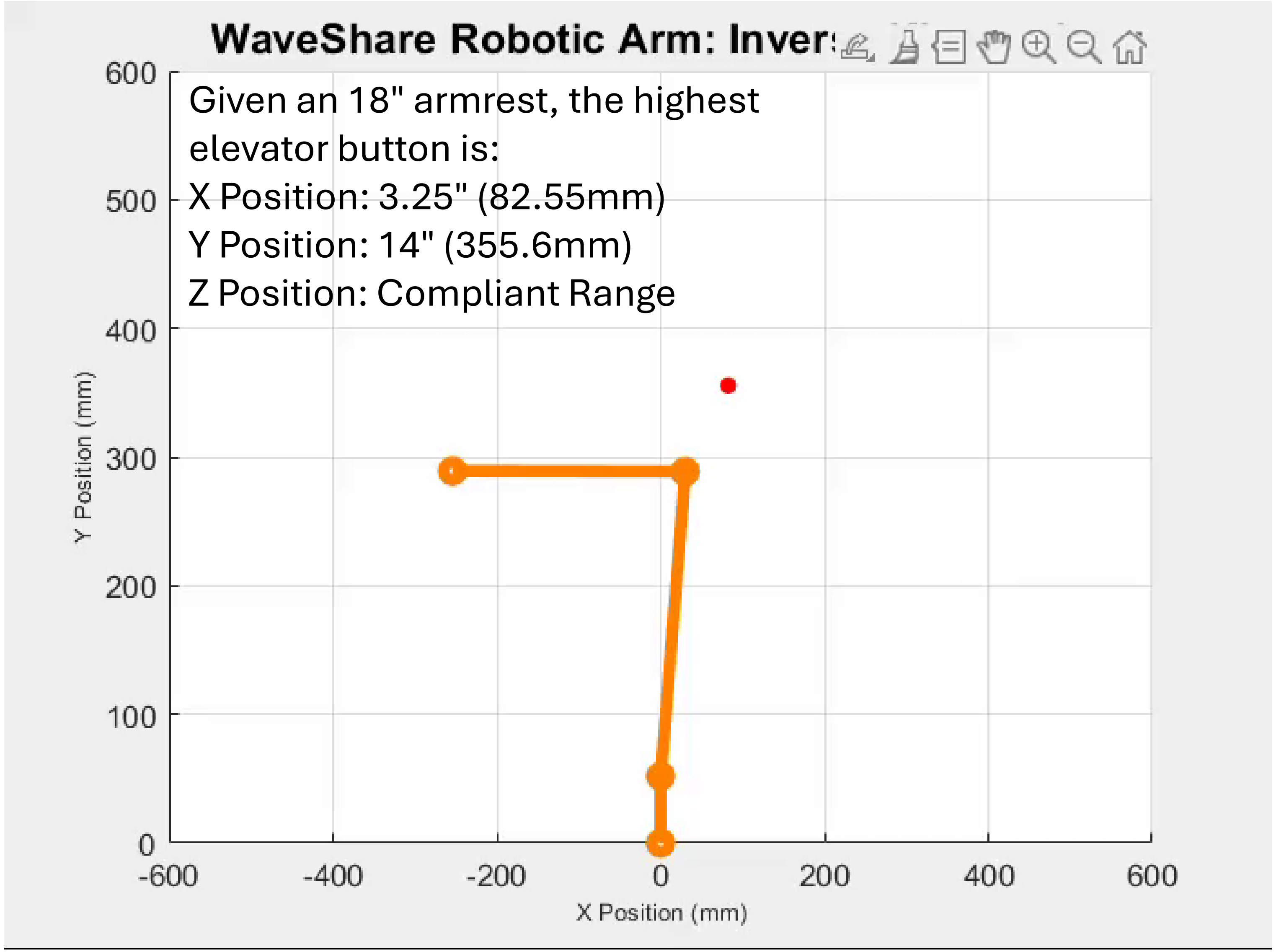
It may be necessary to turn the motorized chair sideways to reach the top button depending on the extrusion of the footrest and users foot size. It may also depend on the armrest length; however, it is not possible to design for every personalized wheelchair

Inverse Kinematics analysis for meeting TR 1.3.1

This MATLAB calculates joint values to obtain a desired position of the end effector…

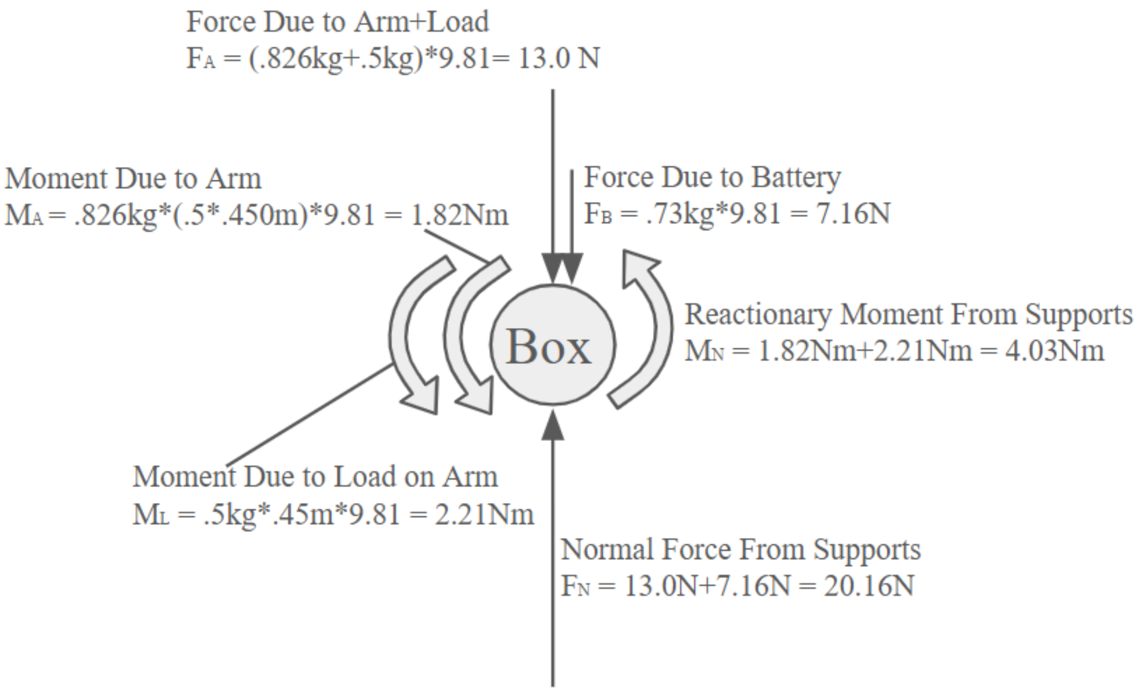
For the purposes of pressing an elevator button, the end effector will be treated as a link even though its a joint since the end effector will be closed for this operation

The end effector will reach the highest ADA complaint button, meeting our objective of pushing an elevator button



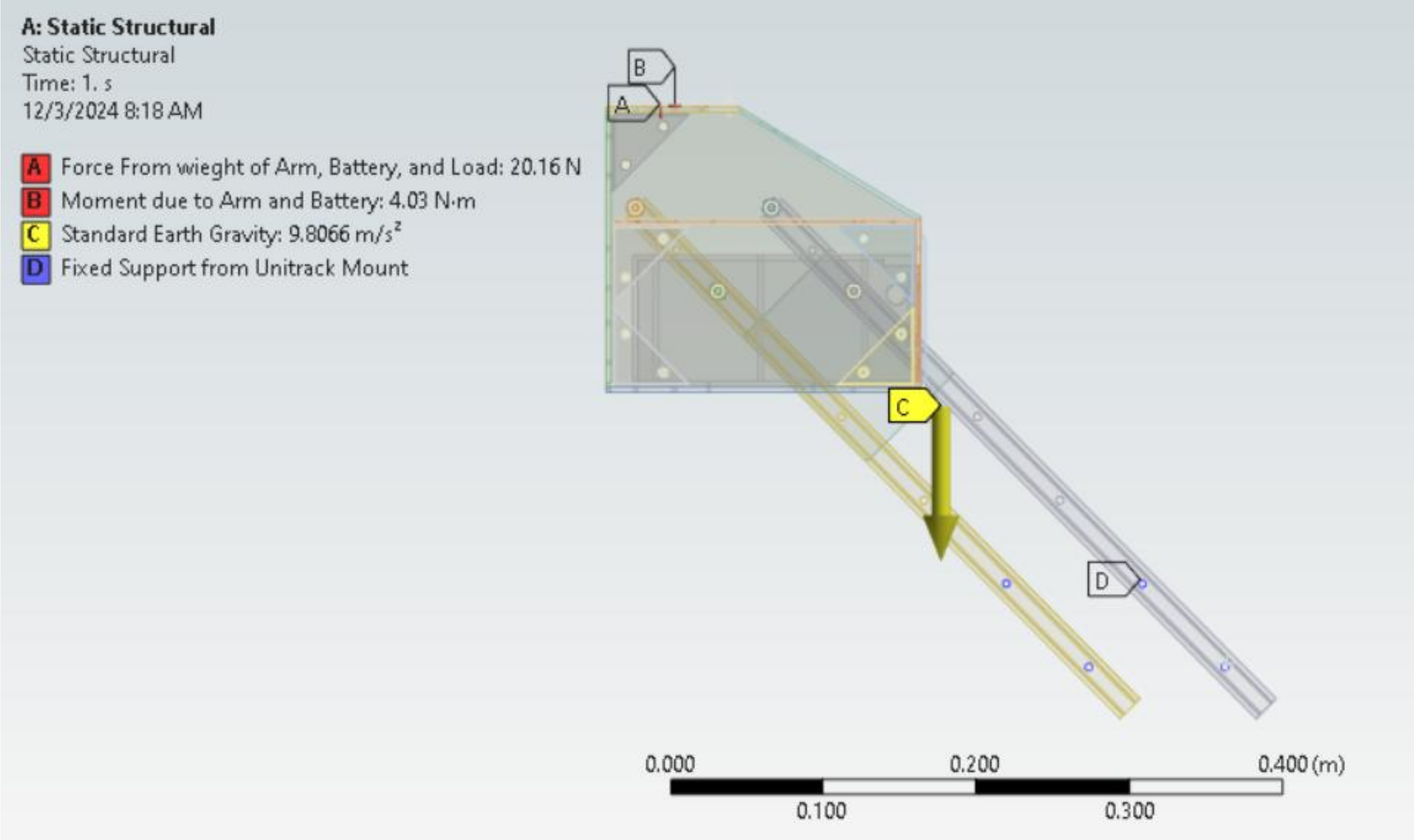
Free Body with Material Properties

- Material: 6061-T6 Aluminum Alloy
  - Ultimate Strength: 313.1MPa
  - Tensile Strength: 259.2 MPa
  - Density: 2713 kg/m^3
- FBD: From FBD we can extract details.
  - Total moment applied of 4.03Nm counterclockwise
  - Total force (not including weight of main model) of 20.16Nm
  - Change in weight due to human interaction is minimal



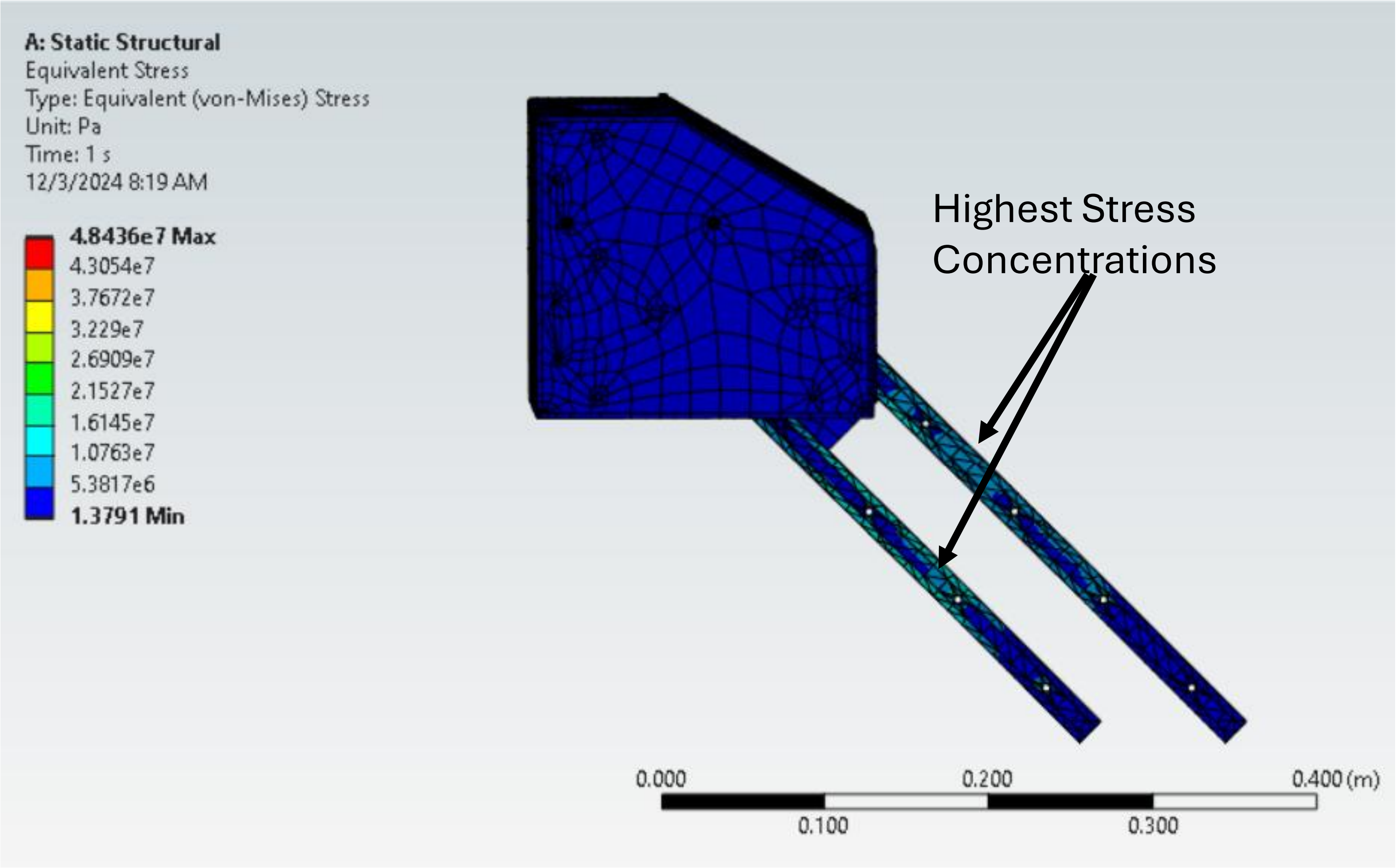
FEA Analysis Setup

- Load and moment from arm are applied at the point where the arm is mounted.
- Load is calculated using FBD from previous page, assuming the maximum load is being applied to the system at its maximum reach.
- Fixed supports are applied at the Unitrack mounting holes.



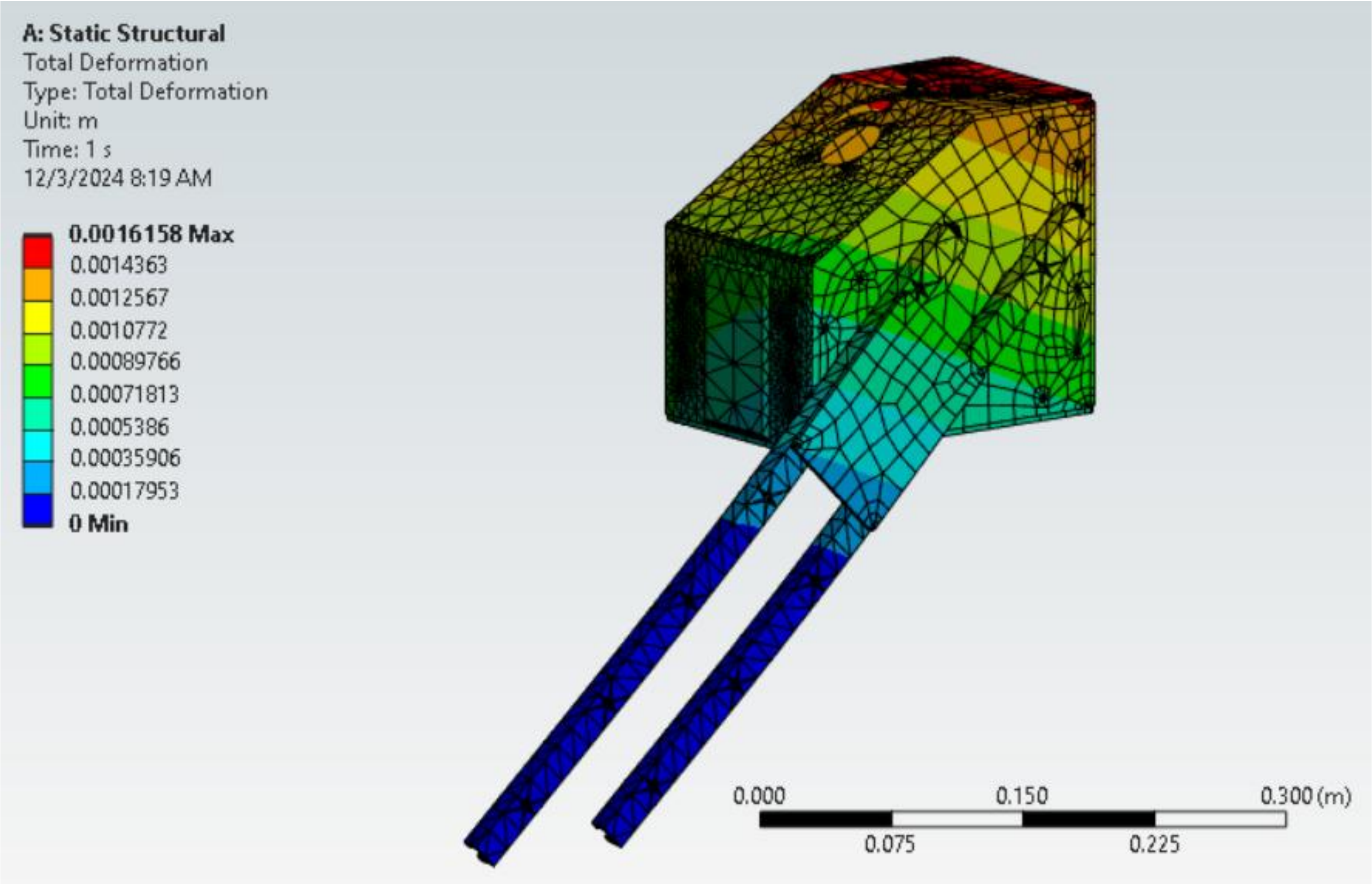
FEA Analysis to ensure our supports will not fail under ideal use case

- Ansys applies a mesh across parts, calculating the stress and deformation across the surface.
- Maximum Stress was found to be 48.44MPa, located in the T track.
- When compared to the Yield strength of Anodized Aluminum (275.8Mpa), FOS of 5.69.

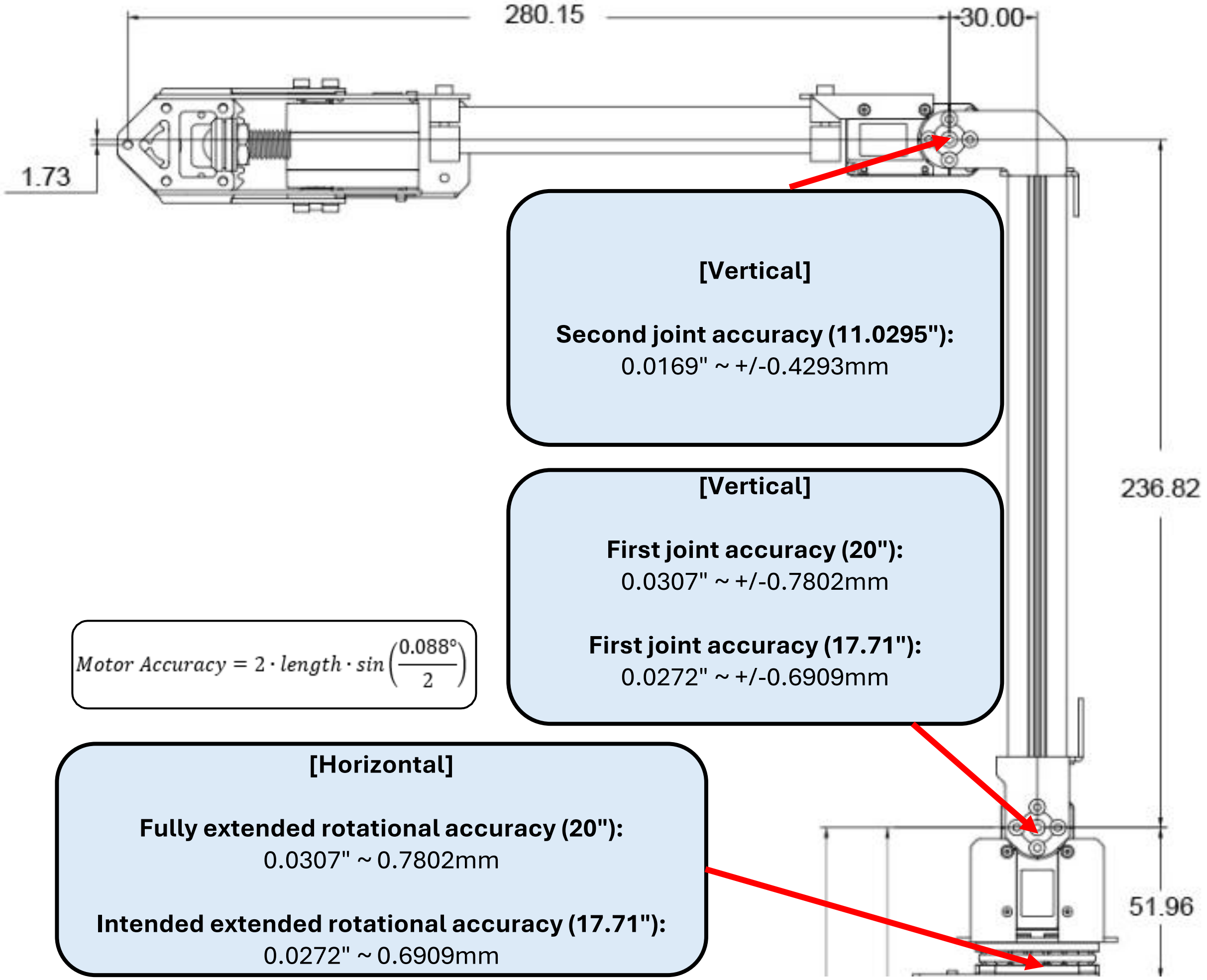


FEA Analysis to Ensure Accuracy of the Robotic Arm End Effector

- Maximum deformation of model was found to be in the T Track, which led to the box being shifted as a result.
- Maximum deformation of 1.62mm, which is within project allowances and does not move outside of accuracy bounds.



Accuracy of motors (Resolution) – TR 1.3.2



[Vertical]

Second joint accuracy (11.0295"):
0.0169" ~ +/-0.4293mm

[Vertical]

First joint accuracy (20"):
0.0307" ~ +/-0.7802mm

First joint accuracy (17.71"):
0.0272" ~ +/-0.6909mm

$$Motor\ Accuracy = 2 \cdot length \cdot sin\left(\frac{0.088°}{2}\right)$$

[Horizontal]

Fully extended rotational accuracy (20"):
0.0307" ~ 0.7802mm

Intended extended rotational accuracy (17.71"):
0.0272" ~ 0.6909mm

- The maximum horizontal resolution is 0.0307" while the maximum vertical accuracy is two times 0.0307" (0.0614"). The "diagonal error" amounts to 0.0687", which is within our error of +/- 10mm.
- As the coupled motor's resolution at the end effector fall well below the minimum diameter of a button (¾" as per ADA standards) the system will be able to navigate accurately to the desired location and complete our system objective.

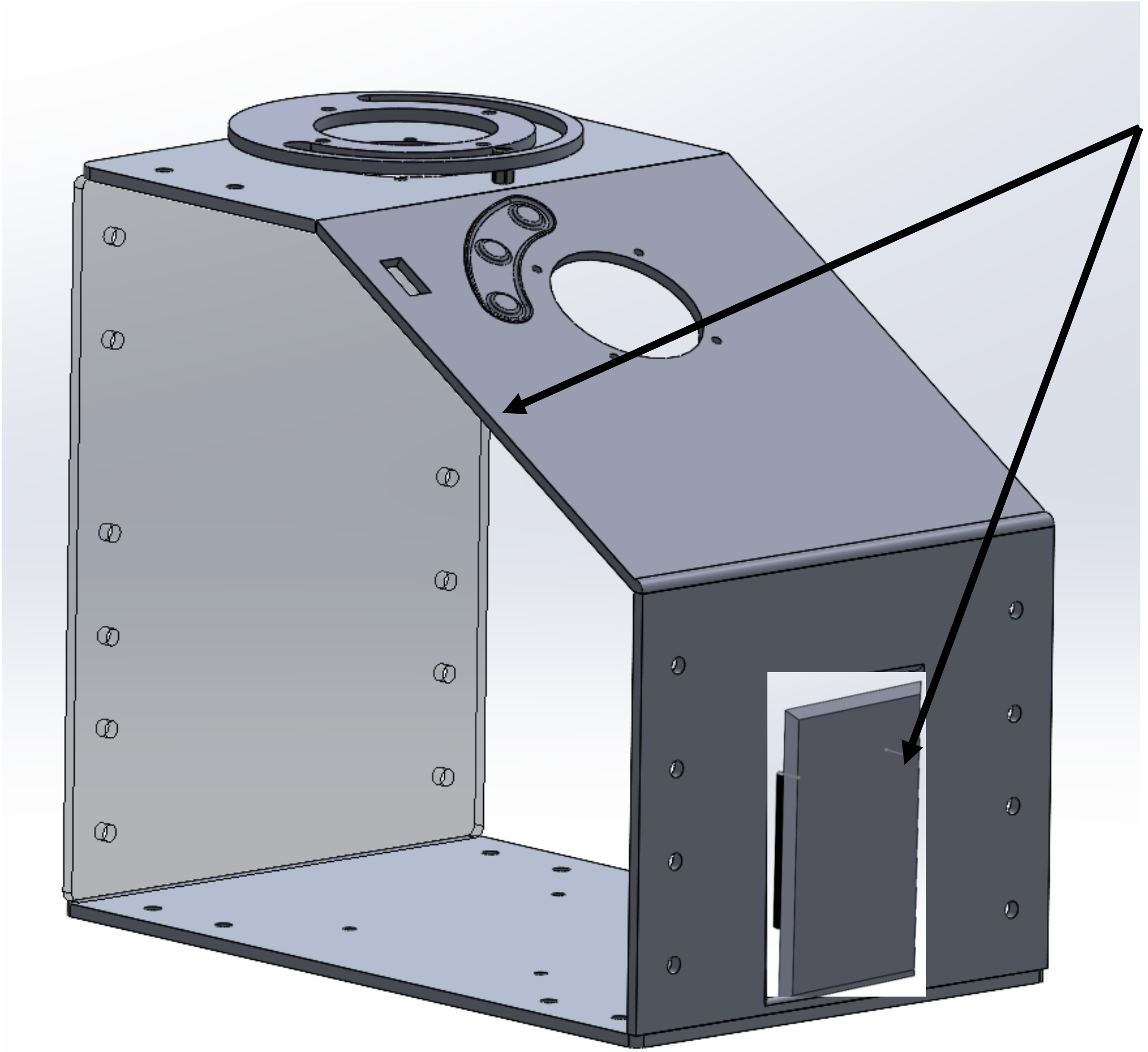
Managing Risks - Waterproofing the System

HIGH RISK

Water entering the case or robotic arm and contaminating electronics – causing system failure and possible sporadic movement of the robotic arm

Mitigation Plan:

Weld all the case except the sides where the case mounts to the wheelchair



Use rubber or neoprene gasket to line the exposed sides

When the sides are screwed in, the pressure created against the gasket restricts any water from entering the inside of the case

To cover the battery, use a plugin that fits inside the hole and is easily removable for charging

Managing Risks – System Waterproofing

- The servos and wires are currently exposed – prone to water damage
- Solution: Use a waterproof heat shrink
- Will be shrunk around linkages and left loose around the joints (servos)

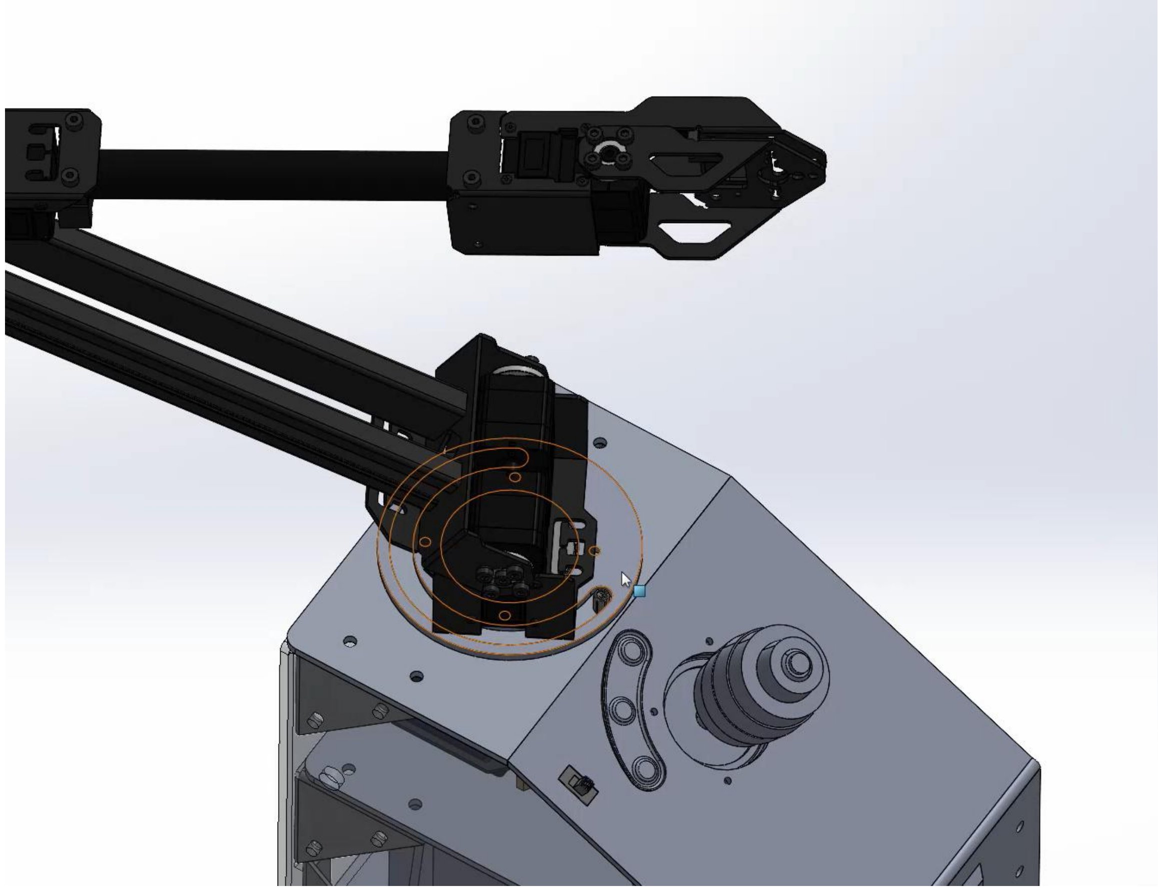
- Allows the arm to maintain its mobility
- Provides a seal around sensitive components
- The end effector will have a plastic "glove" that wraps around the two pieces to protect the servo

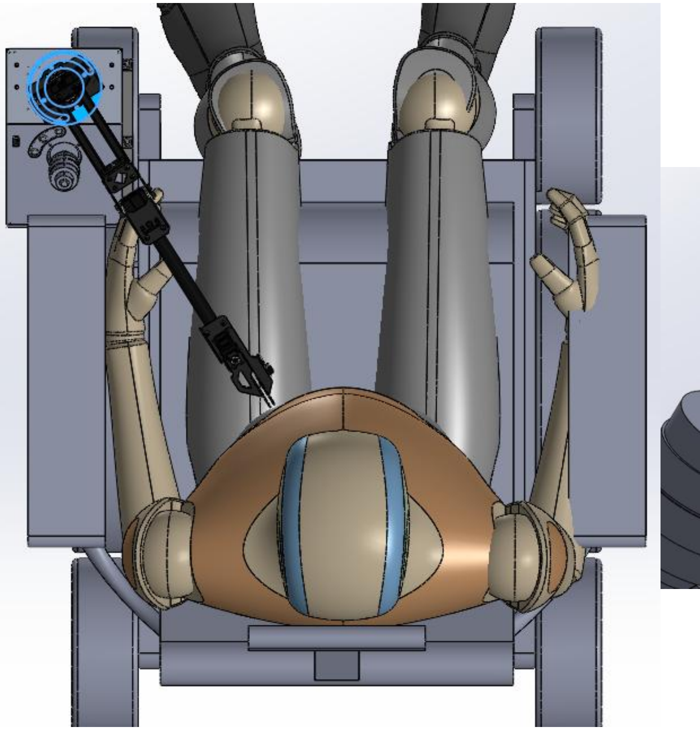
Managing Risks - Preventing the Robot from Hitting the User

HIGH RISK
There is a major risk of the robotic arm hitting the user when in use. This includes pinching, poking, hitting, and cutting (screws, sharp edges – there should be none)

A limiting plate and standoff are installed under the base of the arm

The standoff protrudes into the cutout in the plate which stops the motion of the arm if it exceeds the safe operating zone



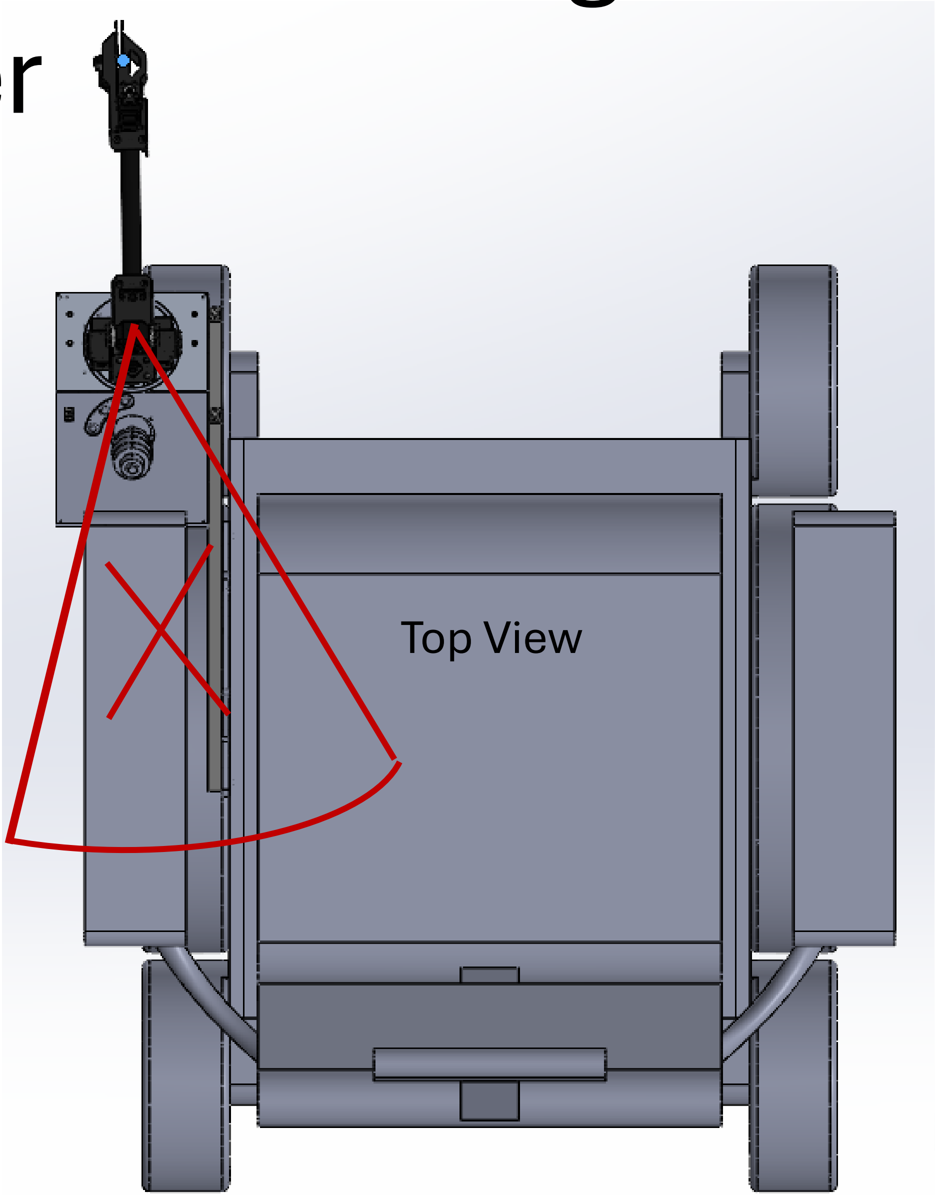


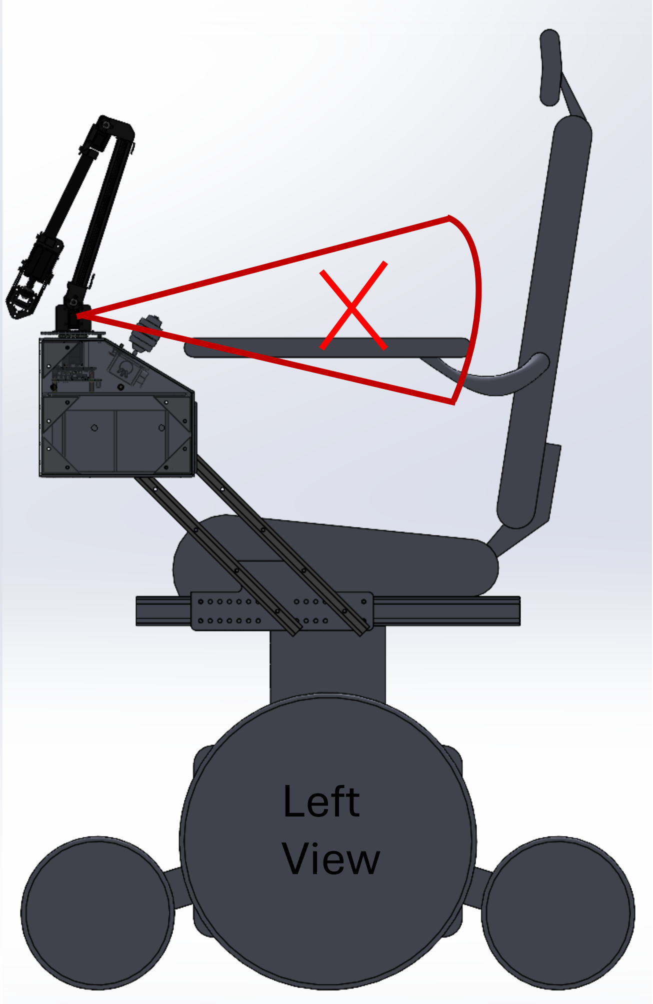
Managing Risks - Preventing the Robot from Hitting the User

Additionally, the team will code in a limited operating zone that allows the robot to act within a certain volume, however it will not be able to reach others

The excluded zone will be around the user's shoulder, arm, and hand and encompass the armrest

In the event that the programming stops fail, the physical stops should prevent the arm from hitting the user, and vice verse

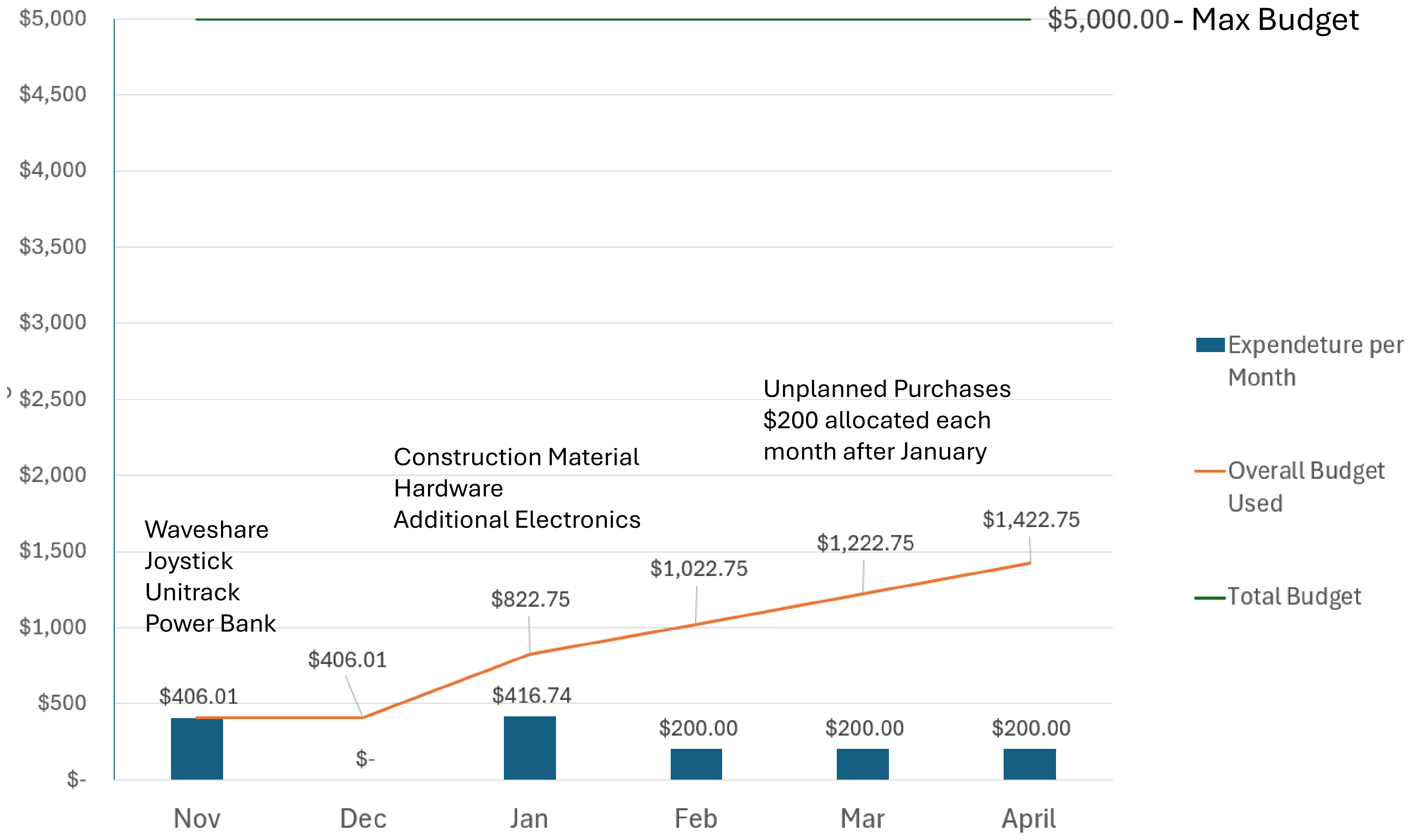




Bill of Materials

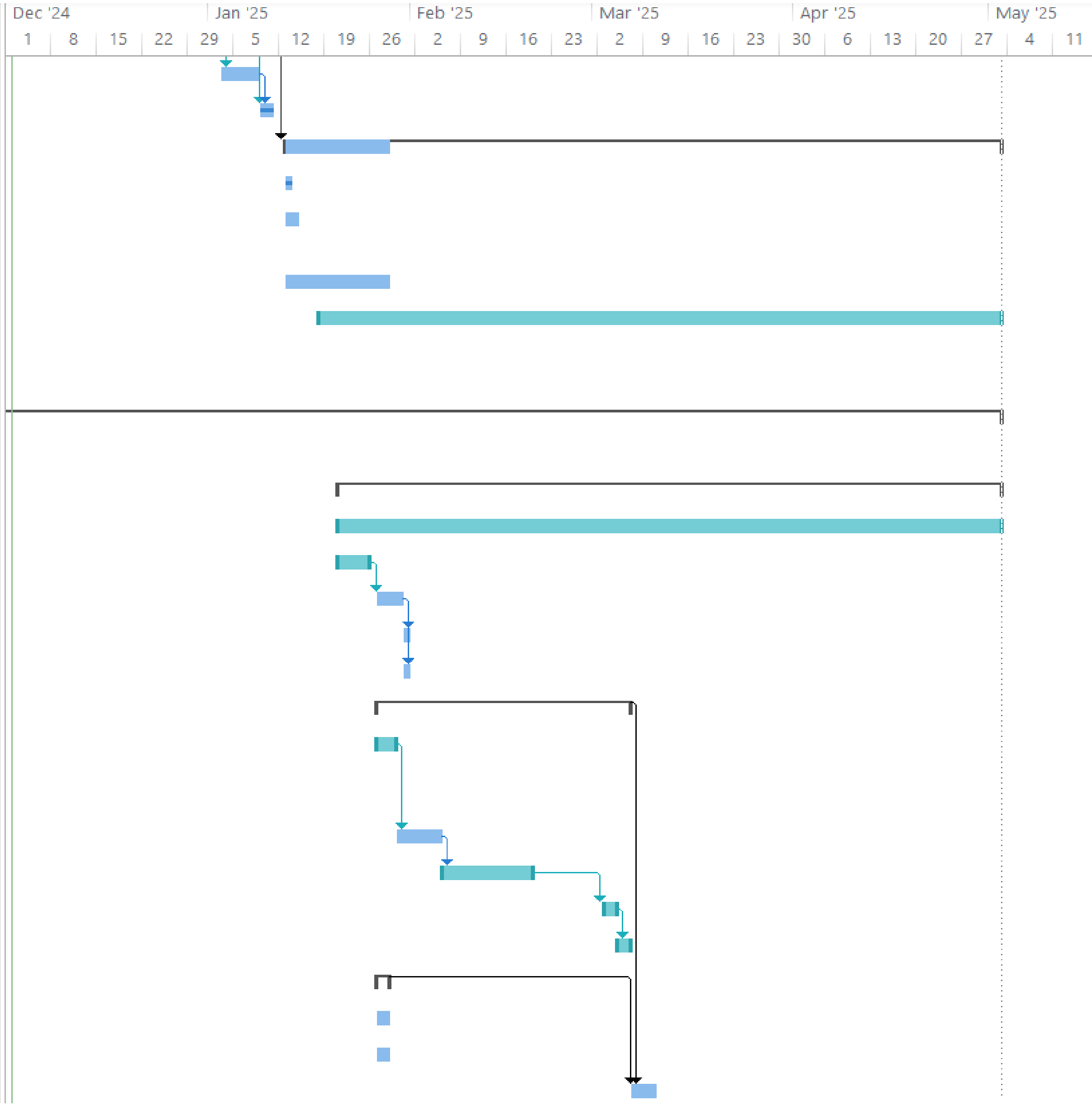
Budget - 205 - Robotic Arm to Assist Motorized Wheelchair Users												
Starting Budget =	$ 5,000.00											Any other
Projected Budget =	$ 4,177.25				Projected Used Budget =	$ (822.75)	Verification					
Available Budget =	$ 4,593.99				Used Budget =	$ (406.01)	$ (406.01)					
Item description	Part #	URL	Vendor	Quantity	Price Per Unit (No ship or tax)	Tax	Shipping	Shipping Method	Total Cost	Status (OPEN/PURCHASED)	Received (Y/N)	Purchase Month

Team 205's Budget Expenditures through the Year



Team 205 is Currently on Schedule for Spring Preparation

Task Name	% Complete	Start	Finish	Duration
Assign/ Complete individual tasks for CDR Report	0%	Fri 1/3/25	Wed 1/8/25	4 days
Prototype (Fitment Purposes)	100%	Thu 1/9/25	Fri 1/10/25	2 days
Spring Semester Preparation	**1%**	**Mon 1/13/25**	**Fri 5/2/25**	**80 days**
Order/ Recieve Joystick, UniTrack, and Battery	100%	Mon 1/13/25	Mon 1/13/25	1 day
Assess prototype and FEA analysis with advisor for approval of material orders	0%	Mon 1/13/25	Tue 1/14/25	2 days
Order/ Receive approved materials	0%	Mon 1/13/25	Tue 1/28/25	12 days
Develop code in Aruino IDE that utilizes joystick inputs	0%	Sat 1/18/25	Fri 5/2/25	76 days
Flesh out bugs with the Waveshare arm - determine if still best solution	0%			
Course Deliverables	**50%**	**Tue 9/3/24**	**Fri 5/2/25**	**173 days**
Spring Action Plan	0%	Thu 8/29/24	Thu 8/29/24	1 day?
Spring Semester	**0%**	**Tue 1/21/25**	**Fri 5/2/25**	**74 days**
Classes Begin	0%	Tue 1/21/25	Fri 5/2/25	74 days
Wire the battery, joystick, and robotic arm together	0%	Tue 1/21/25	Sat 1/25/25	5 days
Impliment Code from the Arduino IDE	0%	Mon 1/27/25	Thu 1/30/25	4 days
Test joystick according to test plan	0%	Fri 1/31/25	Fri 1/31/25	1 day
Test motion the of the robotic arm according to test plan	0%	Fri 1/31/25	Fri 1/31/25	1 day
Assemble the robot case	**0%**	**Mon 1/27/25**	**Thu 3/6/25**	**29 days**
Acquire and organize all raw materials (aluminum, screws, welding material, gasket material, jumper cables)	0%	Mon 1/27/25	Wed 1/29/25	3 days
Schedule time with Matt to weld and practice welding	0%	Thu 1/30/25	Wed 2/5/25	5 days
Weld the case and brackets together	0%	Thu 2/6/25	Wed 2/19/25	10 days
Install the robot arm and joystick	0%	Mon 3/3/25	Tue 3/4/25	2 days
Wire in the battery, joystick and ESP 32	0%	Wed 3/5/25	Thu 3/6/25	2 days
Mount the assembled arm and frame to the wheelchair	**0%**	**Mon 1/27/25**	**Tue 1/28/25**	**2 days**
Attach the unitrack to the wheelchair	0%	Mon 1/27/25	Tue 1/28/25	2 days
Modify the armrest to represent the Permobil	0%	Mon 1/27/25	Tue 1/28/25	2 days
Attach the system to the wheelchair	0%	Fri 3/7/25	Mon 3/10/25	2 days



VIRGINIA TECH
Parent Project Muscular Dystrophy

Conclusions and Future Work

Manufacturing:

- Bandsaw – Cutting aluminum stock to proper shape
- CNC Mill – Cutting slots and holes in aluminum plates for wire passthroughs and ventilation
- Drill Press – cutting and threading mounting holes
- Pliers or Bandsaw - Trimming the gauge material to length
- Acquiring electronics equipment (jumper cables, solder, switches, buttons, resistors, battery, etc.)

Assembly:

1. Weld and bolt: interior and exterior aluminum plates to create robotic case
    - Insert internal components such as battery, joystick, and Waveshare into robotic case
2. Weld truss onto vertical tracks
3. Bolt vertical tracks onto horizontal track
4. Bolt horizotal track onto Unitrack of the modified Jazzy wheelchair
5. Slide and bolt robotic case onto vertical tracks

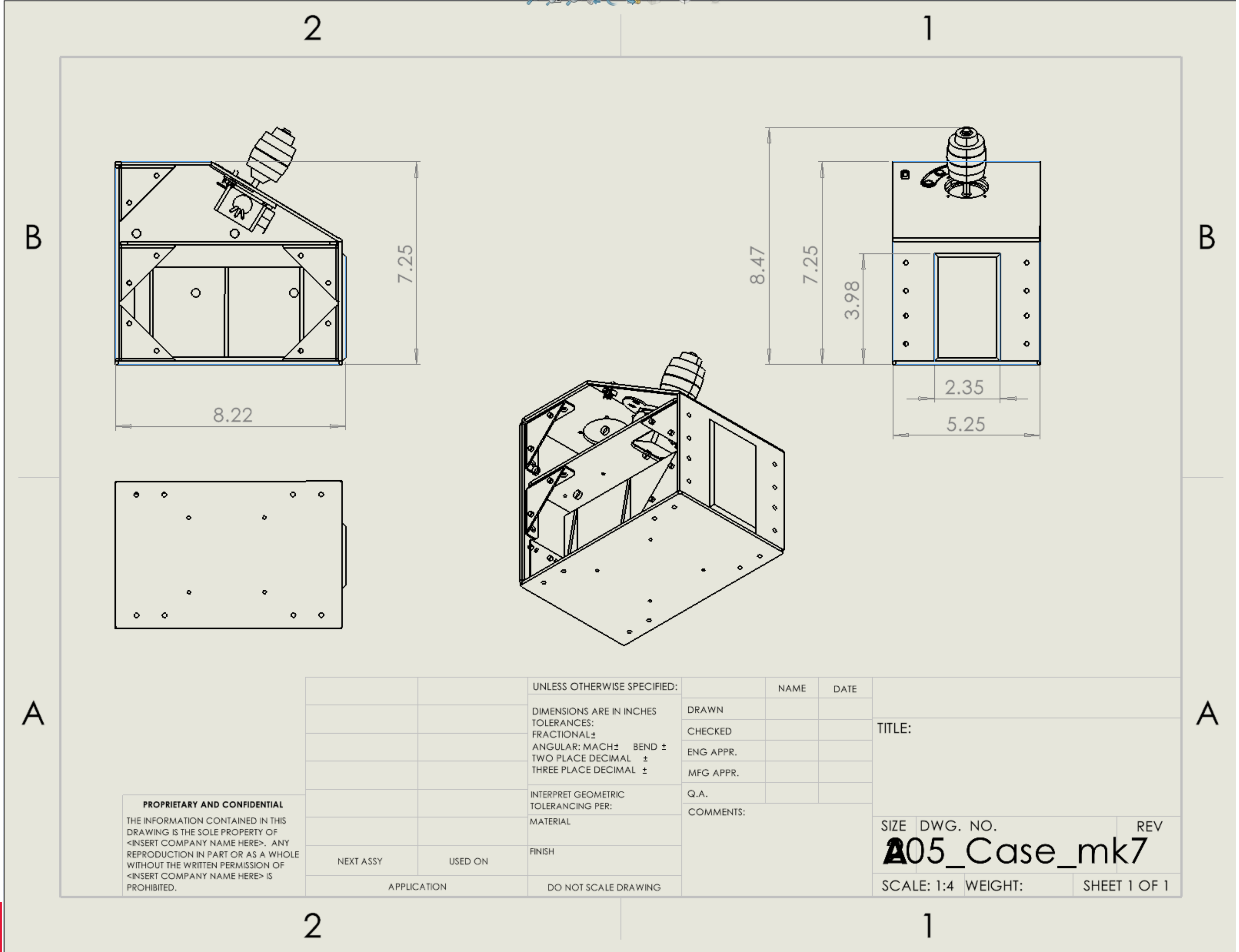
Conclusions:

- Team is on track to meeting the high value functions of pushing an elevator button
- System prototype fabrication will begin in the spring after preliminary testing
- Further system testing will be completed once fabrication is complete
- Mount the system on one of the end user's chairs to get feedback on prototype – make changes before May

Thank you for your time!

Questions?

Backup Slides, Additional Information



VIRGINIA TECH

Parent Project Muscular Dystrophy

Machines Used

- Bandsaw – Cutting stock aluminum to proper shape, Trimming the gauge material to length

- CNC Mill – Cutting slots and holes in aluminum plates for wire passthroughs and ventilation

- Drill Press – cutting and threading mounting holes

Assembly

1. Weld and bolt: interior and exterior aluminum plates to create robotic case (no.8 bolts and locknuts)(x20)
   - Insert internal components such as battery, joystick, and Waveshare into robotic case
2. Weld truss onto vertical tracks
3. Bolt vertical tracks onto horizontal track (18-8 pan head slotted screws and ¾" diam locknuts)(x4)
4. Bolt horizontal track onto Unitrack of the modified Jazzy wheelchair (M6 bolts and T-nuts)(x4)
5. Slide and bolt robotic case onto modified Jazzy wheelchair (no.8 bolts and locknuts)(x4)

Robot Arm Code – Important parts

```
global ser
ser = serial.Serial('COM3', baudrate=int(115200), timeout=1, dsrdtr=None)
```

Defines the serial connection (connection through USB port of computer

```
    keys = pygame.key.get_pressed()
    distance = math.sqrt(x**2 + y**2 + z**2)
    if keys[pygame.K_w] and distance < 450:
        x += 1
        time.sleep(.0001)
    if keys[pygame.K_s] and distance < 450:
        x -= 1
        time.sleep(.0001)
    if keys[pygame.K_a] and distance < 450:
        y += 1
        time.sleep(.0001)
    if keys[pygame.K_d] and distance < 450:
        y -= 1
        time.sleep(.0001)
    if keys[pygame.K_q] and distance < 450:
        z -= 1
        time.sleep(.0001)
    if keys[pygame.K_e] and distance < 450:
        z += 1
        time.sleep(.0001)
```

Polling for a keypress in a pygame window (this code was just to make sure we could control the arm ourselves)

Distance calculates the distance of the arm (base to end effector) for a future “if” condition

A bunch of “if” statements that determine what to do if a specific key is pressed

Robot Arm Code – Important parts

```
113    if distance > 449.9999999:
114        if x > 0:
115            x -= 2
116        if x < 0:
117            x += 2
118        if y > 0:
119            y -= 2
120        if y < 0:
121            y += 2
122        if z > 0:
123            z -= 2
124        if z < 0:
125            z += 2
126    if keys[pygame.K_z]:
127        t -= 0.01
128        time.sleep(.0001)
129    if keys[pygame.K_c]:
130        t += 0.01
131        time.sleep(.0001)
132    if keys:
133        space = " " *20
134        print("\r",[x,y,z,t,distance], space, end="" )
135    data = {
136        "T": 1041,
137        "x": x,
138        "y": y,
139        "z": z,
140        "t": t,
141    }
142    json_data = json.dumps(data)
143    ser.write(json_data.encode() + b'\n')
144    print(json_data)
145    time.sleep(0.0001)
```

If the distance ever exceeds our buffer value (450 in our case), then it brings the arm back in every direction by 2 mm – depending on which octant the end effector is in

126-131 controls the opening and closing of the end effector

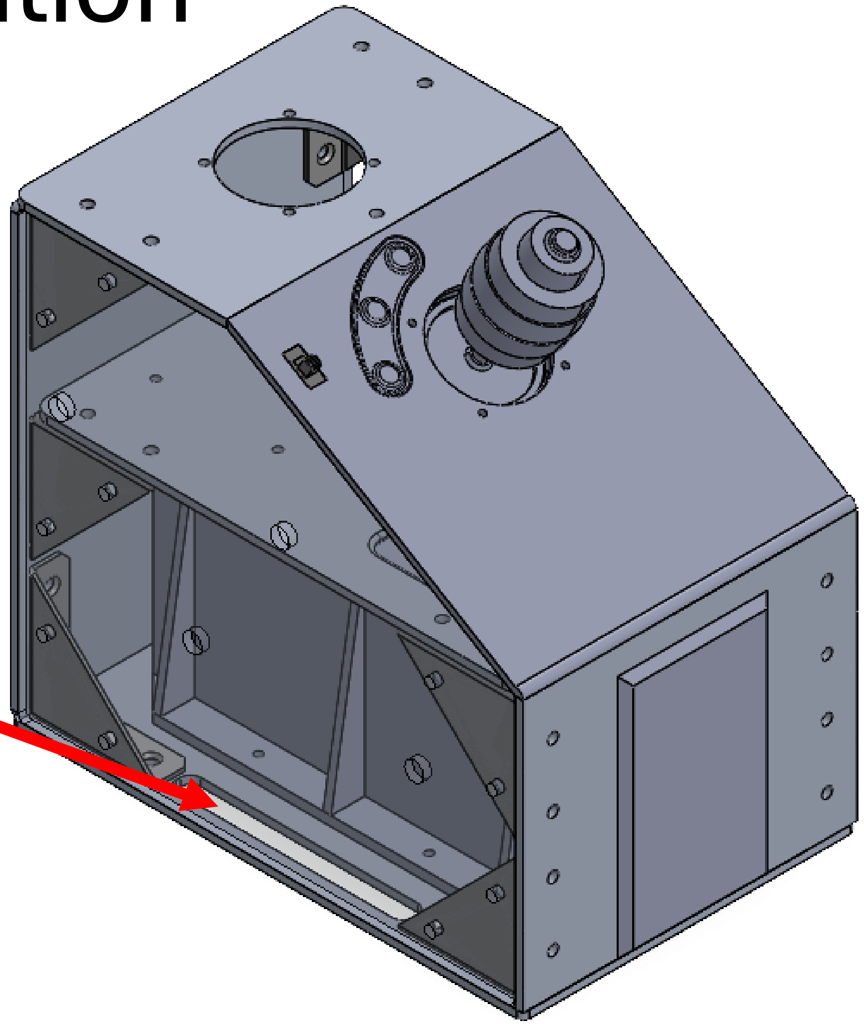
132 prints the coordinate output in the terminal

135 formats the JSON data to be sent in 143

142-145 formats the JSON data again, writes it over the serial connection, prints the sent command, then sleeps to keep the system unstressed

Optional Ventilation

Holes in the bottom plate for ventilation.



Variability to System Mapping (Directly Infront)

13" Armrest

- True dimensions vary from powered wheelchair to powered wheelchair.
- Two lengths (13" min, 18" max per Permobil specifications) are used to calculate the maximum profile the robotic arm can reach.

Forward Kinematics

- Transformation matrix (T) displays the position (p) and orientation (R) of the end effector in reference to the base of the robot

$$T = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_1 \\ r_{21} & r_{22} & r_{23} & p_2 \\ r_{31} & r_{32} & r_{33} & p_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Screws (S1,S2,S3,S4) are determined through robot geometry
- M is the initial transformation matrix, showing the initial position and orientation of the end effector

$$T = e\hat{}([S_1]*\theta_1)*e\hat{}([S_2]*\theta_2)*e\hat{}([S_3]*\theta_3)*e\hat{}([S_4]*\theta_4)*M$$

- Given desired θ values, the position and orientation of the end-effector can be determined

Load Capacity of Waveshare Robotic Arm

- The Waveshare load capacity has been tested, resulting in a maximum load of ….
- However, as the load increases, the end-effector becomes shaky which may lead to inaccurate positioning
  - Rubbers bands will be implemented to act as dampers

Software Testing of Waveshare Robotic Arm

- The Waveshare robotic arm has been tested with coordinate control
- Though we have established connection and control over the arm, we must shift our focus to integrating the 4-axis joystick with the arm in the spring
- Further debugging is required to ensure reliable response and fluid motion from the arm