

Automated Surgical System Integrating an Intelligent Soft Robotics Surface and a Smart Robotic Arm

ETSD-0100-9-12

INTRODUCTION

Performing surgery presents challenges due to risks of the patient moving during the operation. Existing solutions, such as rigid immobilizers, lack adaptability, such that constant manual assistance is often needed. To address this, a flexible pneumatic platform integrated with a robotic arm and camera feedback was developed to enhance surgical support of the patient. The system features a grid of soft, air-powered silicone muscles, creating a dynamic surface adaptable to various surgical needs. The robotic arm, guided by real-time camera feedback, enables precise manipulation and works in tandem with the platform. While some limitations in precision and range of motion exist within the platform, this system enhances surgical control and stability. Additionally, its modular design allows for broader healthcare applications, such as preventing bedsores by detecting and adjusting pressure on specific body areas. This innovative approach offers an intelligent, adaptive solution to improve surgical outcomes and patient care, thereby relieving stress from surgeons and doctors.



Quest Surgical National. (2025). Alumni-Hand and Dura-Hand. <https://questnational.com.au/pro-duct-details/alumi-hand-and-dura-hand/>

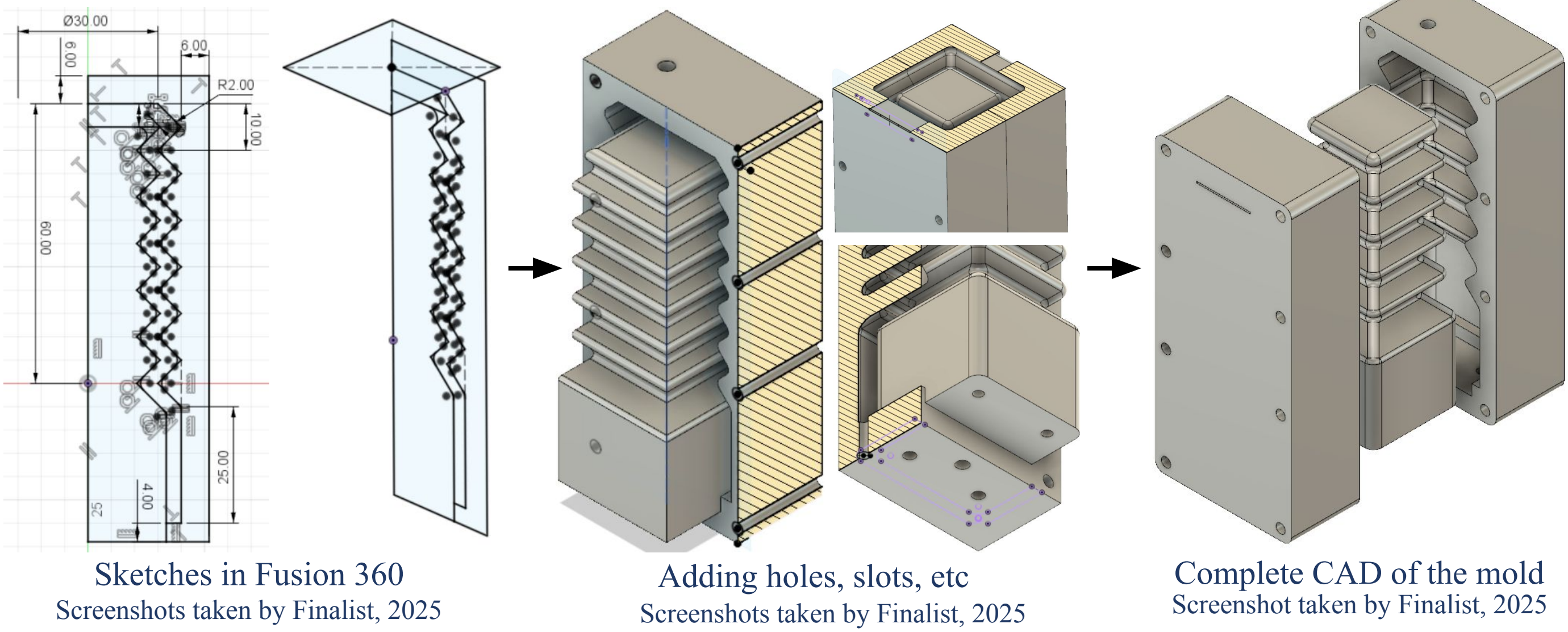
OBJECTIVES

- To create a system able to manipulate and control objects while being safe for human interaction.
- Tilt objects to a specified angle
- Move objects to a specified location
- Actively read and adapt to the current situation
- Have smooth, predictable movements
- Have a modular platform that can easily be expanded and changed to fit custom needs
- The arm will have accurate and precise movements
- The arm will aid the platform in controlling objects

Engineering Design

- The interactive surface is composed of 16 independent muscles arranged in a 4x4 square.
- Each muscle is made out of injection-molded silicone 10A/B
- Horizontal restriction rings both inside and outside the muscles prevent them from deforming
- Each can extend and retract for a greater range of motion
- They are modular in design and can be custom arranged by changing the hole pattern on the mounting plate.

Creating the Silicone Muscle Mold in Fusion 360

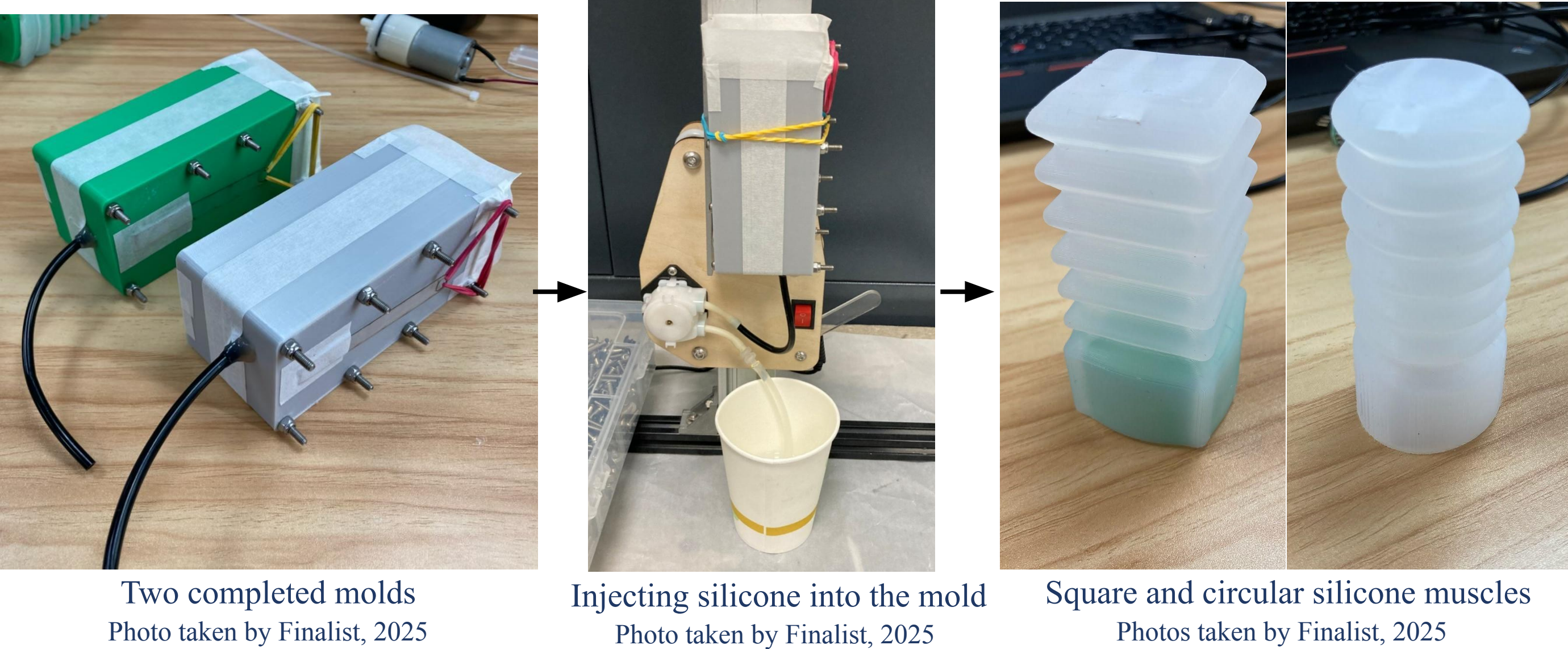


Sketches in Fusion 360
Screenshots taken by Finalist, 2025

Adding holes, slots, etc
Screenshots taken by Finalist, 2025

Complete CAD of the mold
Screenshot taken by Finalist, 2025

Creating the Silicone Muscles

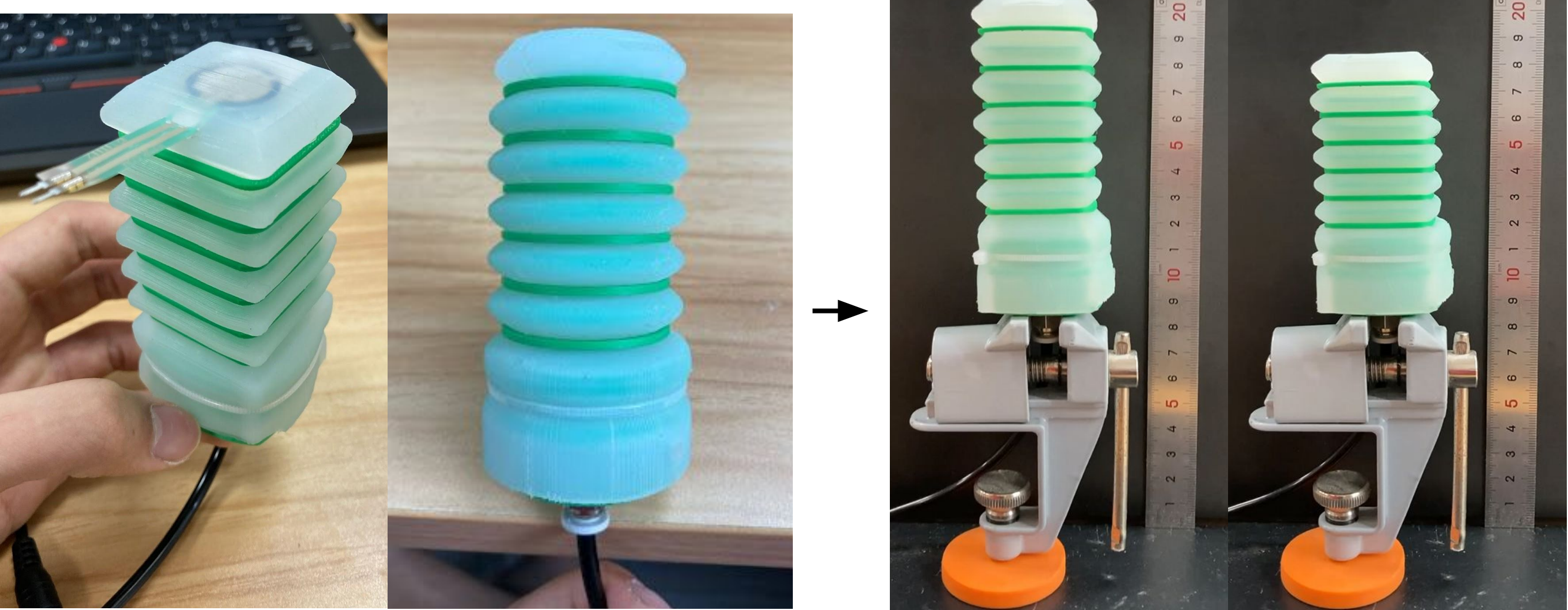


Two completed molds
Photo taken by Finalist, 2025

Injecting silicone into the mold
Photo taken by Finalist, 2025

Square and circular silicone muscles
Photos taken by Finalist, 2025

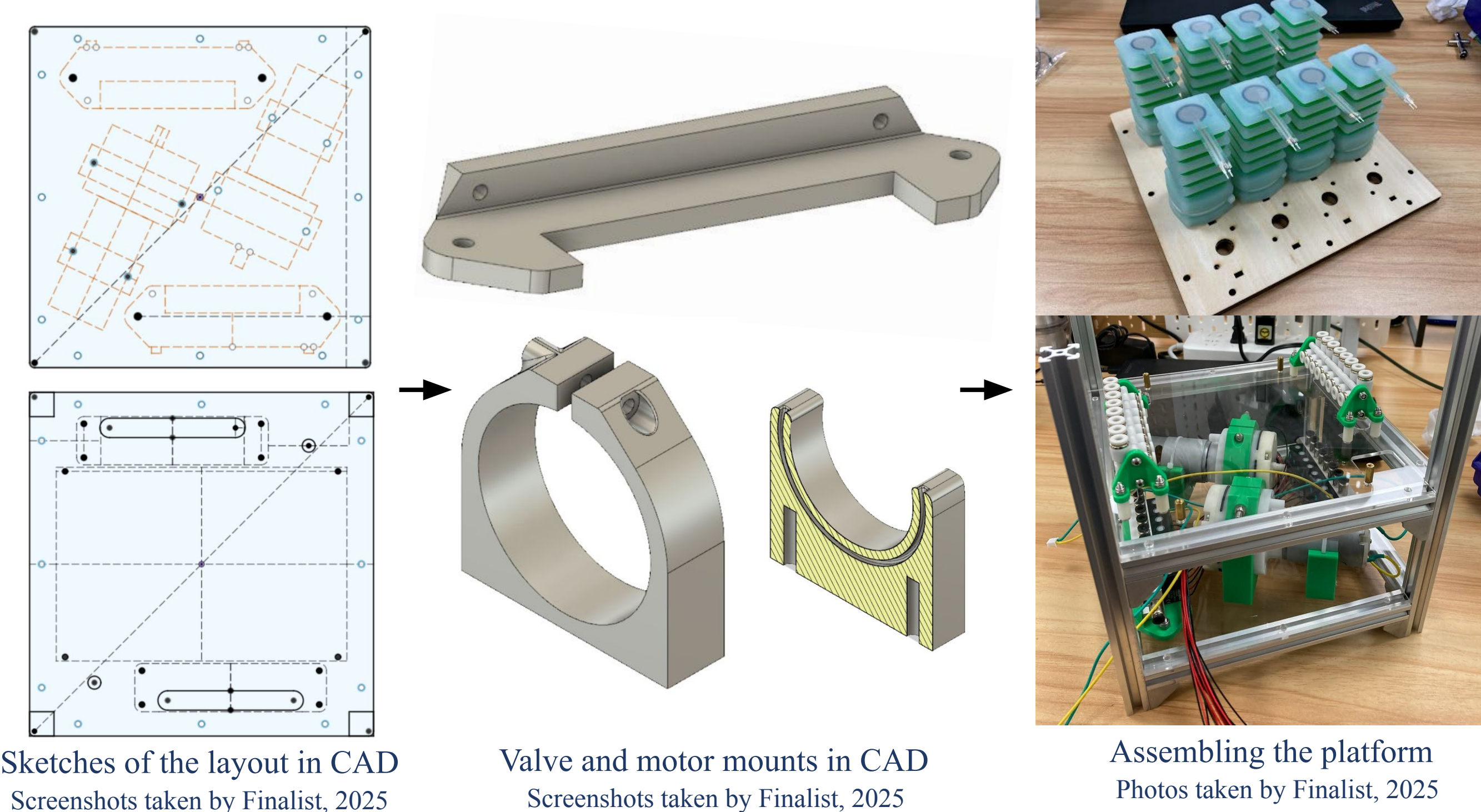
Assembling, Testing, and Iterating the Muscles



Complete silicone muscles
Photos taken by Finalist, 2025

Testing extension/retraction of muscle
Photos taken by Finalist, 2025

Structure and Housing

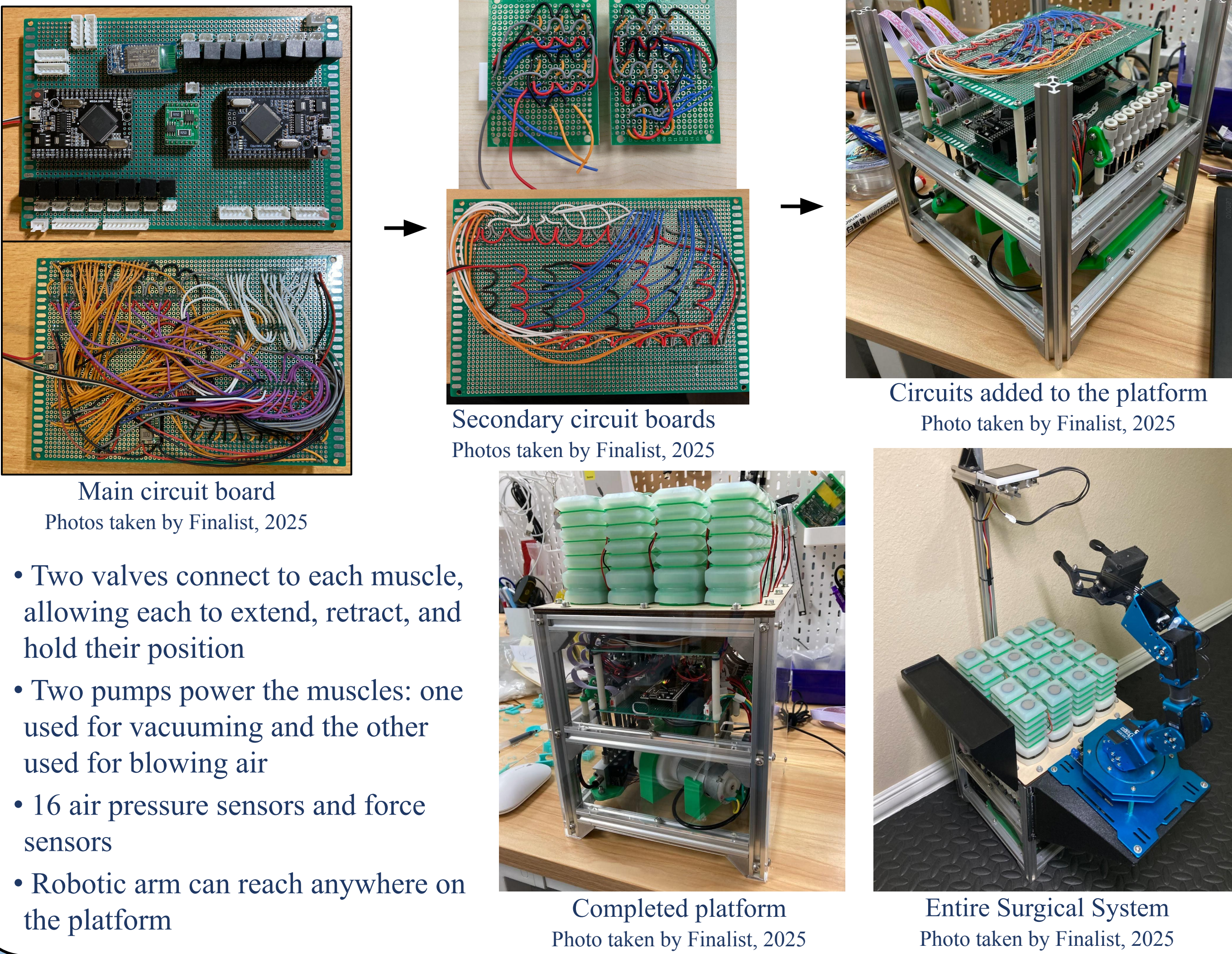


Sketches of the layout in CAD
Screenshots taken by Finalist, 2025

Valve and motor mounts in CAD
Screenshots taken by Finalist, 2025

Assembling the platform
Photos taken by Finalist, 2025

Electronics



Main circuit board
Photos taken by Finalist, 2025

Secondary circuit boards
Photos taken by Finalist, 2025

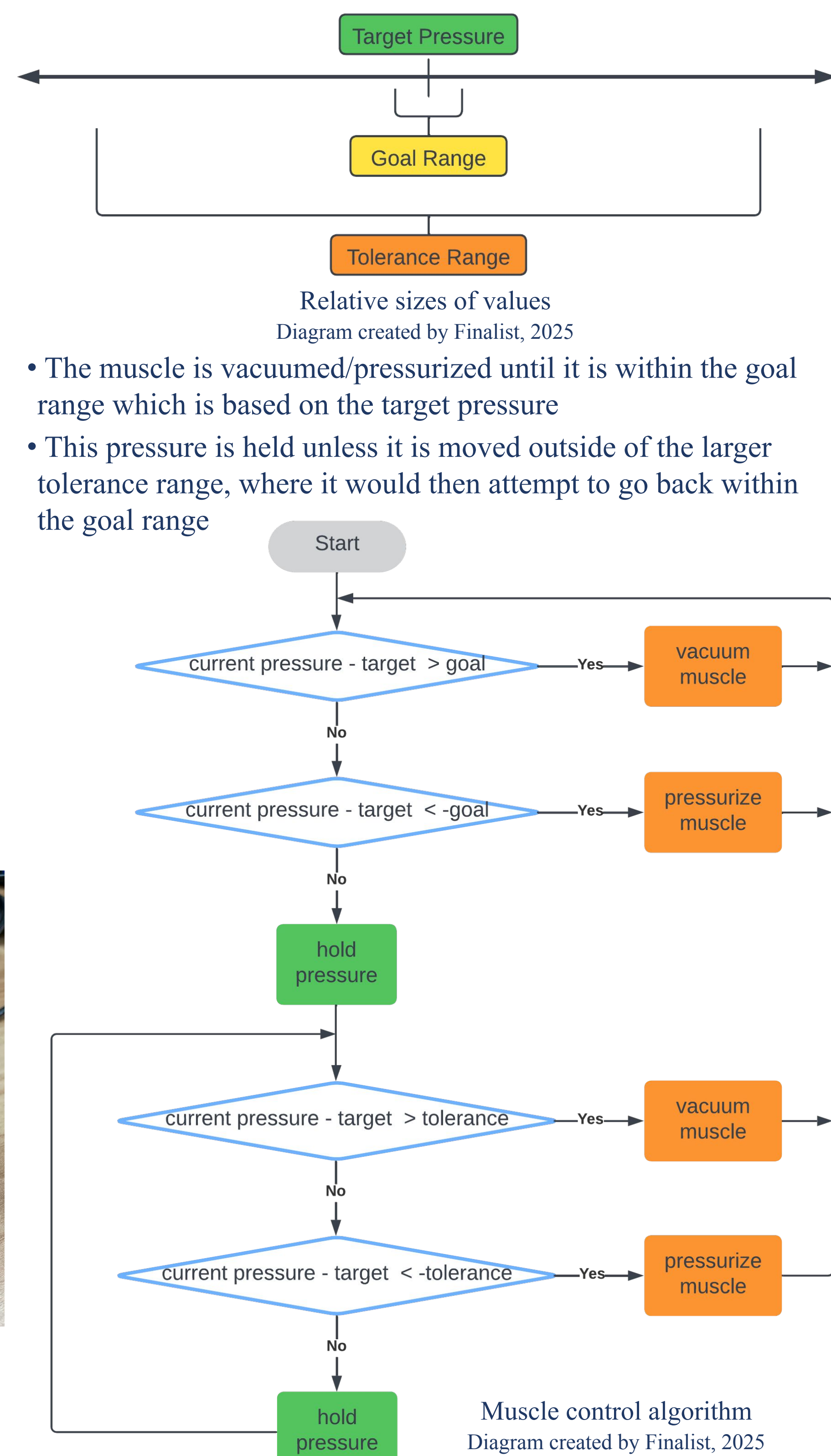
Circuits added to the platform
Photo taken by Finalist, 2025

- Two valves connect to each muscle, allowing each to extend, retract, and hold their position
- Two pumps power the muscles: one used for vacuuming and the other used for blowing air
- 16 air pressure sensors and force sensors
- Robotic arm can reach anywhere on the platform

Completed platform
Photo taken by Finalist, 2025

Entire Surgical System
Photo taken by Finalist, 2025

Programming



Relative sizes of values
Diagram created by Finalist, 2025

- The muscle is vacuumed/pressurized until it is within the goal range which is based on the target pressure
- This pressure is held unless it is moved outside of the larger tolerance range, where it would then attempt to go back within the goal range

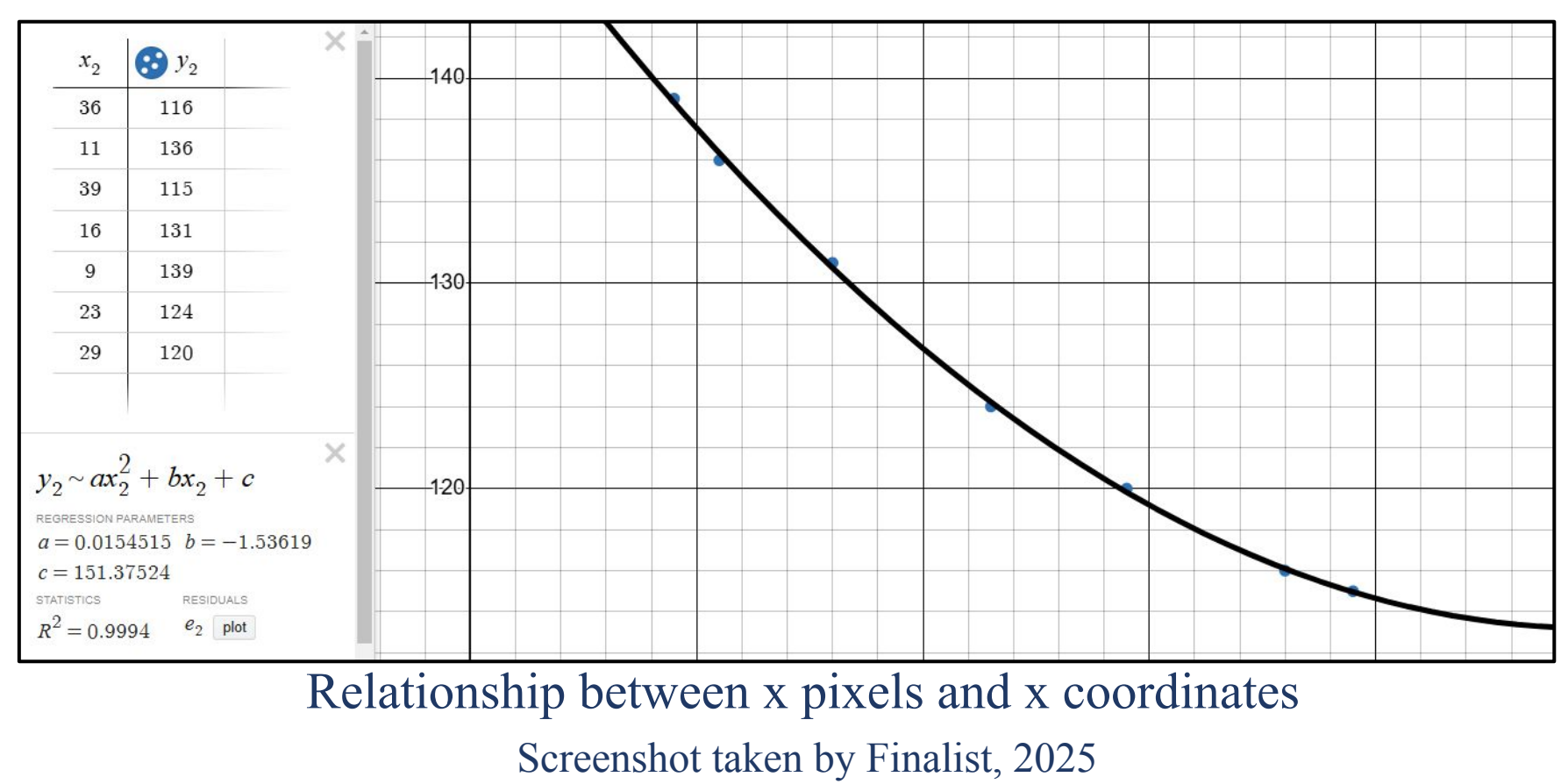
- This logic prevents the muscle from shaking as it tries to reach the target pressure, resulting in a smooth motion

- The K210 Vision Module detects and tracks objects on the platform
- April Tags, color detection, and other methods provided the location and angular data of the object
- Through the serial port, the data is sent to the Arduino and turned into instructions for the muscles and arm

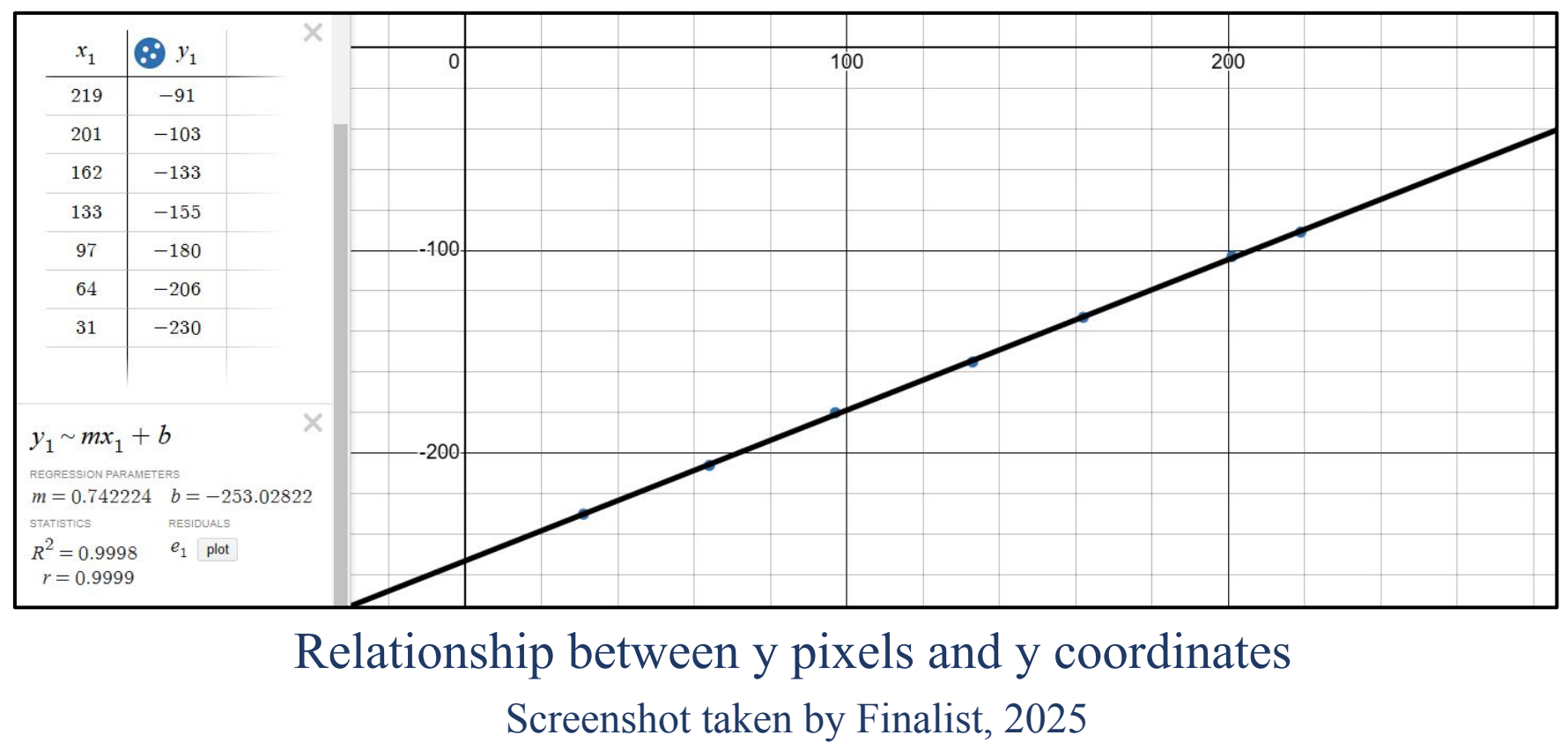
K210 Vision Module
Photo taken by Finalist, 2025

The camera gathers the pixel location of the object, which must then be converted into real 3D coordinates to control the arm

- Data was collected of the pixel vs 3D coordinates
- Regressions were performed to find the best fit curve



Relationship between x pixels and x coordinates
Screenshot taken by Finalist, 2025



Relationship between y pixels and y coordinates
Screenshot taken by Finalist, 2025

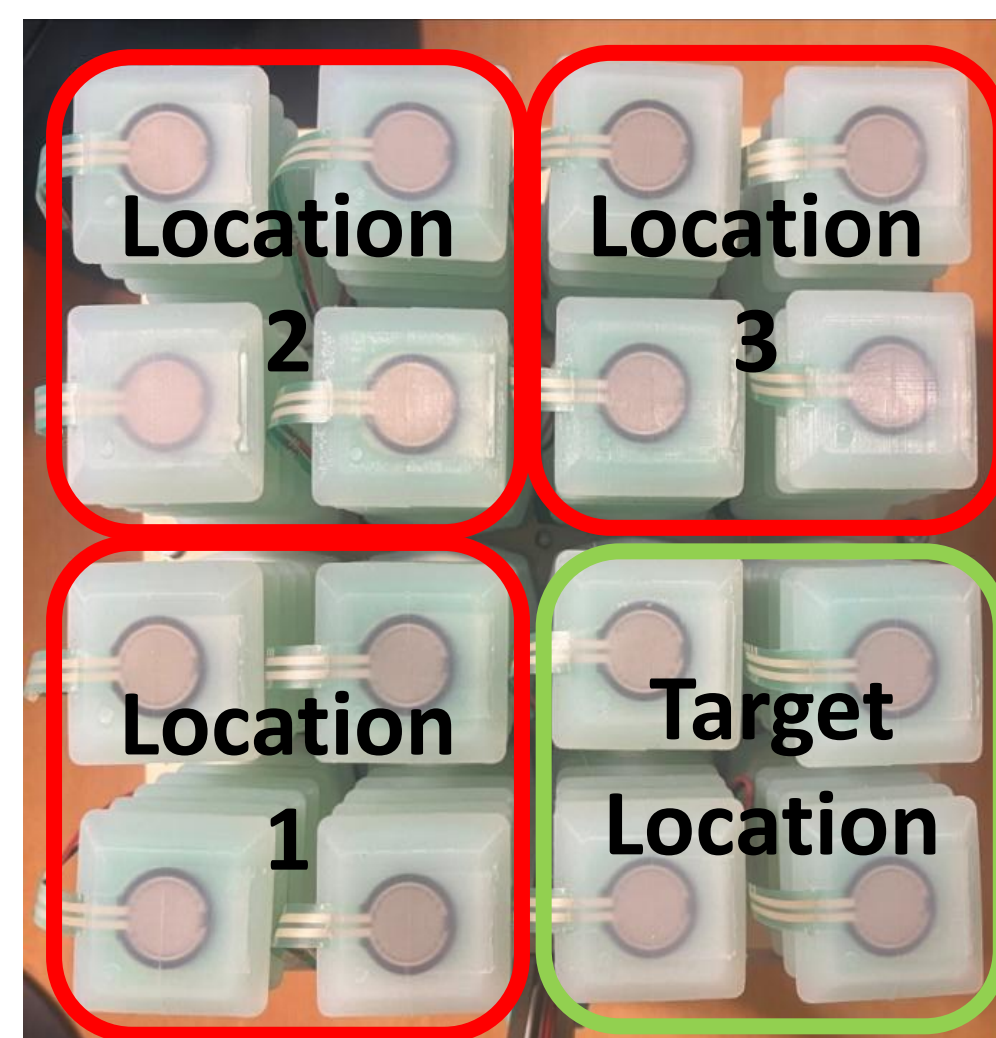
To effectively control the arm, inverse kinematics were used to generate servo angles from a 3D coordinate. These angles were then converted into servo values and sent to the robotic arm.

```
void moveToPos(float x, float y, float z, float phi, float speed) {  
    float ang_6 = atan2(y, x);  
    float hyp = -sqrt(pow(x, 2) + pow(y, 2));  
  
    float w4yp = hyp - armlen[2] * cos(phi);  
    float w4z = z - armlen[2] * sin(phi);  
  
    float delta = pow(w4yp, 2) + pow(w4z, 2);  
    float c4 = (delta - pow(armlen[0], 2) - pow(armlen[1], 2)) / (2 * armlen[0] * armlen[1]);  
    float s4 = sqrt(1 - pow(c4, 2));  
    float ang_4 = atan2(s4, c4);  
  
    float s5 = ((armlen[0] + armlen[1] * c4) * w4z - armlen[1] * s4 * w4yp) / delta;  
    float c5 = ((armlen[0] + armlen[1] * c4) * w4yp + armlen[1] * s4 * w4z) / delta;  
    float ang_5 = atan2(s5, c5) - PI/2;  
    float ang_3 = phi - ang_5 - ang_4 - PI/2;  
    float ang_2 = phi - ang_5 - ang_4 - PI/2;  
  
    tanServoPos[2] = round((237.141 * ang_3 + 495));  
    tanServoPos[3] = round((-237.141 * ang_4 + 495));  
    tanServoPos[4] = round((237.141 * ang_5 + 495));  
    tanServoPos[5] = round((237.141 * ang_6 + 867.5));  
}
```

Generating servo positions through inverse kinematics
Screenshot taken by Finalist, 2025

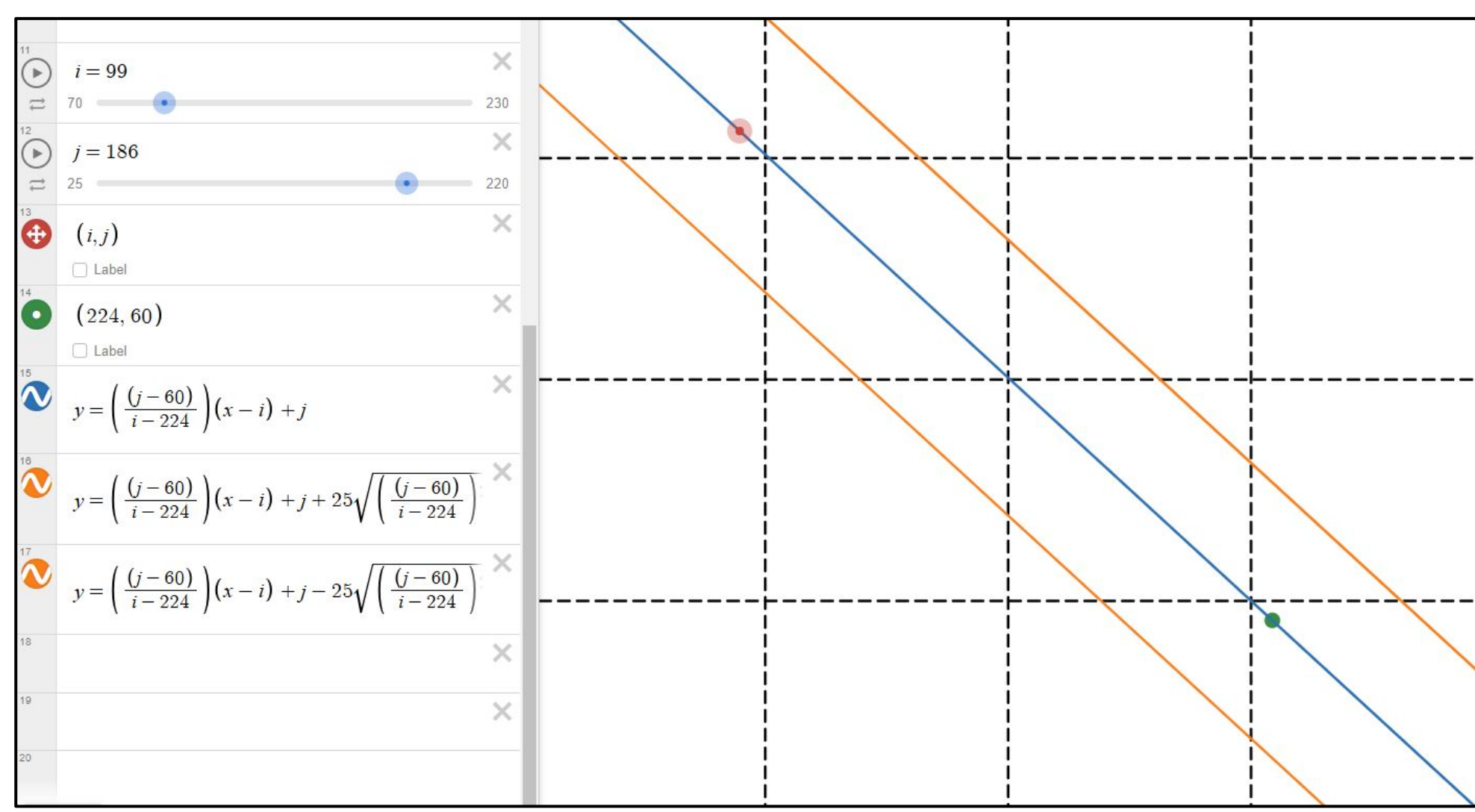
RESULTS

To test object manipulation, the platform attempted to move a 50mm cube to a set location from varying starting positions based on the camera feedback



Visual created by Finalist, 2025

- Generating the path from the cube to the target location
- Target location in bottom right (224, 60)
 - Current cube location is (i, j)
 - The three lines are the generated path
 - The black dashed lines represent the silicone muscles
 - The necessary muscle movements are found based on this path



Visual created by Finalist, 2025



Photos taken by Finalist, 2025

Sequence of transporting the cube from starting location 2 to target location
The white bounding box and cross (center) is created by the camera showing detection of the cube

	Location 1	Location 2	Location 3
Trial 1	Success	Success	Success
Trial 2	Fail	Fail	Success
Trial 3	Success	Success	Success
Trial 4	Success	Success	Fail
Trial 5	Success	Success	Success
Trial 6	Fail	Success	Success
Trial 7	Success	Success	Success
Trial 8	Success	Success	Success

Table created by Finalist, 2025

83% Success Rate: Failure occurred only when the cube got stuck; increasing the muscle per area ratio would solve this issue

The model silicone hand represents the patient's hand during surgery. The device attempts to move the hand to a target position and hold it stable, thereby assisting the surgeon in successfully performing the operation.

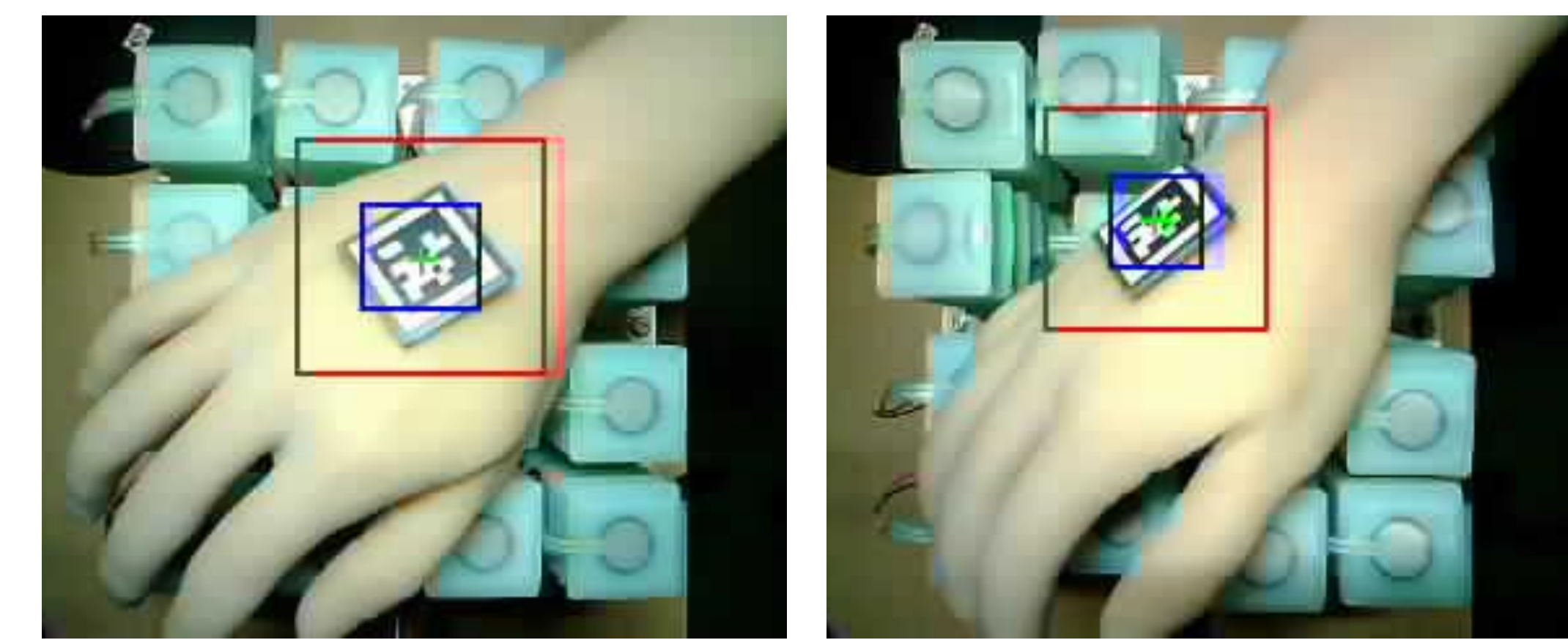


Photo taken by Finalist, 2025

Photo taken by Finalist, 2025

- The red square is the region of interest and follows the April Tag
- If unable to find the April Tag, the ROI will expand
- The smaller ROI increases the frame rate by reducing the searching size
- The blue square is the bounding box of the April Tag
- The green cross is the center of the April Tag

The configuration shows the target x and y angle that the platform will try and tilt the hand to

Config	Trial 1	Trial 2	Trial 3
X: -20 Y: 0	Success	Success	Success
X: 0 Y: -30	Success	Success	Success
X: 40 Y: 0	Success	Fail	Success
X: 0 Y: 50	Success	Success	Fail
X: -10 Y: 20	Success	Success	Success
X: 20 Y: 30	Success	Success	Success
X: 30 Y: -40	Success	Fail	Success
X: -40 Y: -50	Fail	Success	Fail

Table created by Finalist, 2025

79% Success Rate: Failure only occurred when the angle was too high; adjusting the geometry/material of the muscles to allow for a greater range of motion would solve this issue

Testing object detection and manipulation with the robotic arm

- The camera searches the side platform for the cotton ball through color detection.
- These pixel coordinates are sent to the Arduino through the serial
- Pixel coordinates are converted into real life 3D coordinates
- Through inverse kinematics, servo angles are generated and then converted into servo values
- The robotic arm receives these positions and grabs the cotton ball, with the camera actively guiding it
- The arm then transports the cotton ball to the center of the platform

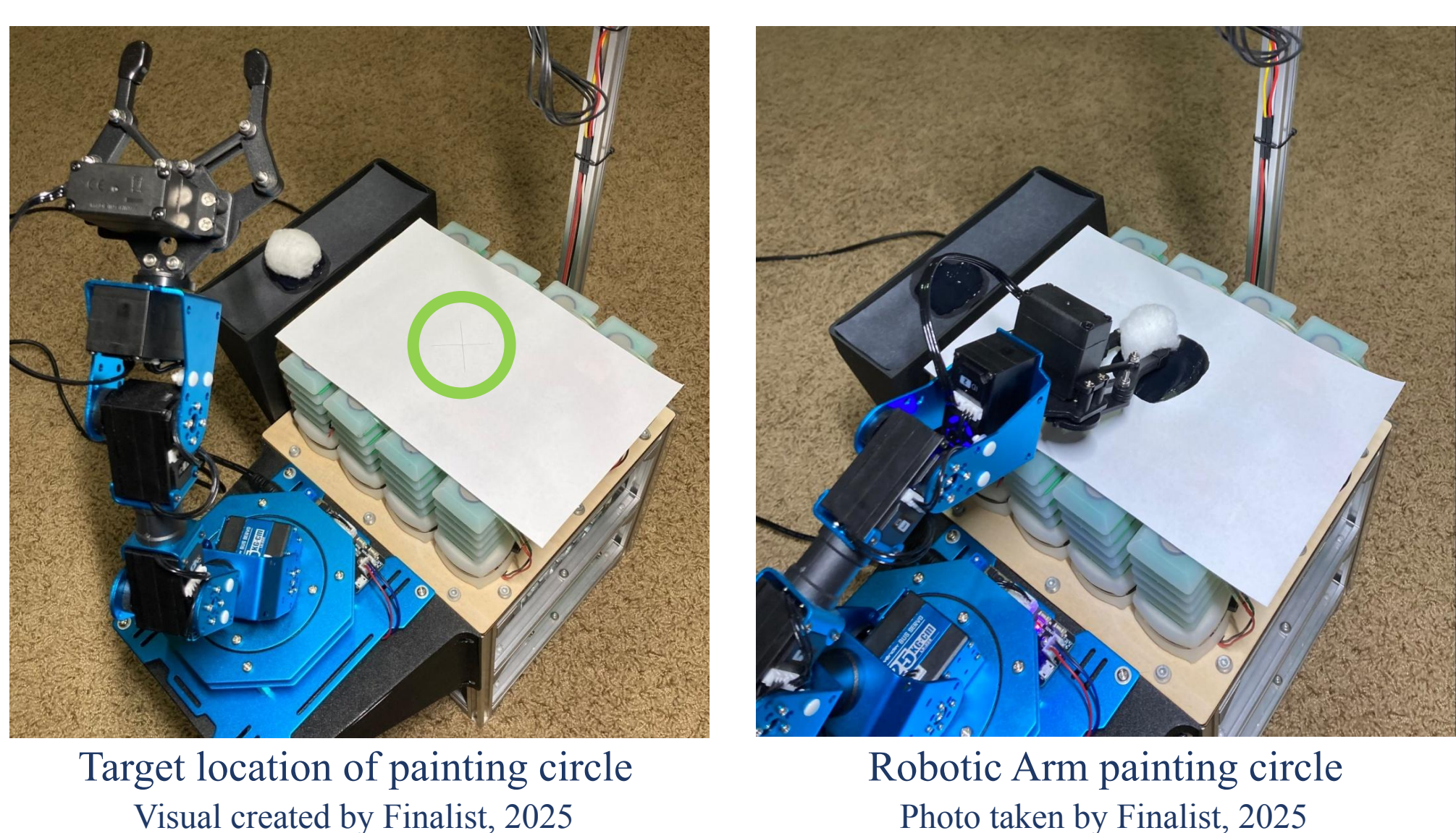
24 trials were performed with a 100% success rate of detecting, grabbing, and transporting the cotton ball.



Grab and drop locations
Visual created by Finalist, 2025

Robotic arm moving the cotton ball
Photo taken by Finalist, 2025

This experiment simulates alcohol-assisted wound disinfection by wiping a soaked cotton ball onto the platform, thus "cleaning" it



Target location of painting circle
Visual created by Finalist, 2025

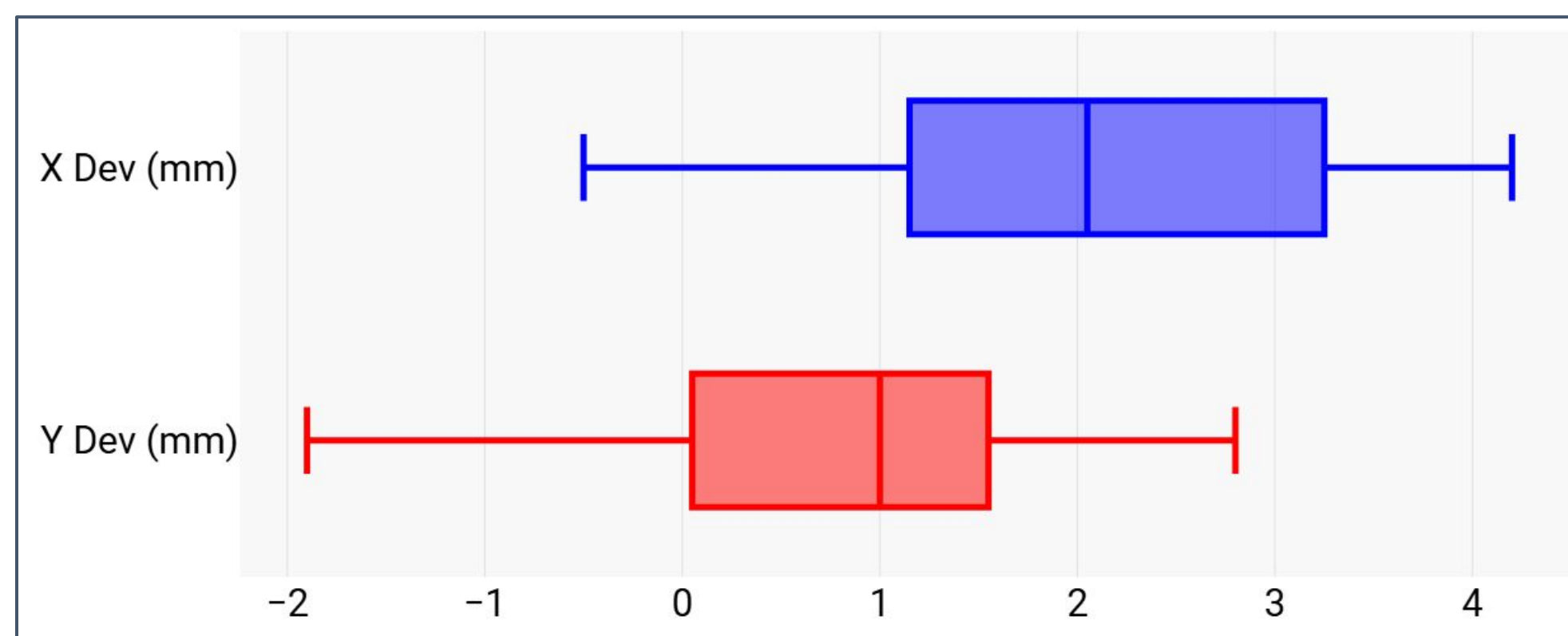
Robotic Arm painting circle
Photo taken by Finalist, 2025

- The cotton ball, dipped in black paint, is detected by the camera
- The arm grabs and moves the cotton ball to the platform
- The arm then "wipes" the center of the platform in a circular motion
- Once done, the arm returns the cotton ball to the side platform

This tests both the arms accuracy and precision: the X and Y deviations are measured from the center of the drawn circle to the center of the platform.

	1	2	3	4	5	6	7	8	...	Avg	SD
X Dev (mm)	-0.5	3.3	2.6	4.2	3.2	1.8	1.1	0.8	...	2.0	1.5
Y Dev (mm)	-1.1	2.8	2.2	0.5	-1.0	1.7	-0.3	-1.9	...	0.8	1.7

Table created by Finalist, 2025



Graph created by Finalist, 2025

The robotic arm is quite accurate and precise as it is able to control the cotton ball down to the millimeter. Improvements could be made to the rigidity of the arm and its servos to decrease the x and y deviations. The algorithms for controlling the arm in 3D space could also be further optimized to increase accuracy.

FUTURE RESEARCH

- A larger platform in different arrangements
- Smaller muscles for higher resolution and finer control
- Reducing the size of the gaps in between modules
- Optimizing muscle geometry for more extension and retraction
- Detecting objects without the use of April Tags
- Visual sensors (ex. LiDAR) to assist in detecting the object
- Advanced code for higher precision and control of the muscles
- Enhanced algorithm for moving and manipulating objects
- Ability to directly measure each muscles length
- Full integration of the arm and platform in tandem
- Higher quality arm to improve precision and smoothness

CONCLUSIONS

- Able to tilt and move objects to a specific location
- The modular design allows for custom sizes and arrangements
- The camera and air pressure sensors provided constant input into the feedback control loop
- Can detect and track objects based on color and April Tags
- The soft, silicone material allows for safe human interaction
- Able to actively adjust to changing scenarios
- Fully automated once given a target location or angle
- High control over the muscles as they have smooth motions and can hold pressures steadily
- The robotic arm is consistent when grabbing and moving objects
- The arm can move to any position following a custom path

REFERENCES

- Johnson, B. K., Naris, M., Sundaram, V., Volchko, A., Ly, K., Mitchell, S. K., Acome, E., Kellaris, N., Kepling, C., Correll, N., Humbert, J. S., & Rentschler, M. E. (2023). A multifunctional soft robotic shape display with high-speed actuation, sensing, and control. *Nature Communications*, 14(1), 4516. <https://doi.org/10.1038/s41467-023-39842-2>
- Kotner, J., Black, J., Call, E., Gefen, A., & Santamaria, N. (2018). Microclimate: A critical review in the context of pressure ulcer prevention. *Clinical Biomechanics*, 59, 62–70. <https://doi.org/10.1016/j.clinbiomech.2018.09.010>
- Leithinger, D., & Ishii, H. (2010). Relief. *DSpace@MIT* (Massachusetts Institute of Technology). <https://doi.org/10.1145/1709886.1709928>
- Nakagaki, K., Fitzgerald, D., Ma, Z. (John), Vink, L., Levine, D., & Ishii, H. (2019). inFORCE. *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '19*. <https://doi.org/10.1145/3294109.3295621>
- Yoshida, S., Sun, Y., & Kuzuoka, H. (2020). PoCoPo: Handheld Pin-based Shape Display for Haptic Rendering in Virtual Reality. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. <https://doi.org/10.1145/3313831.3376358>