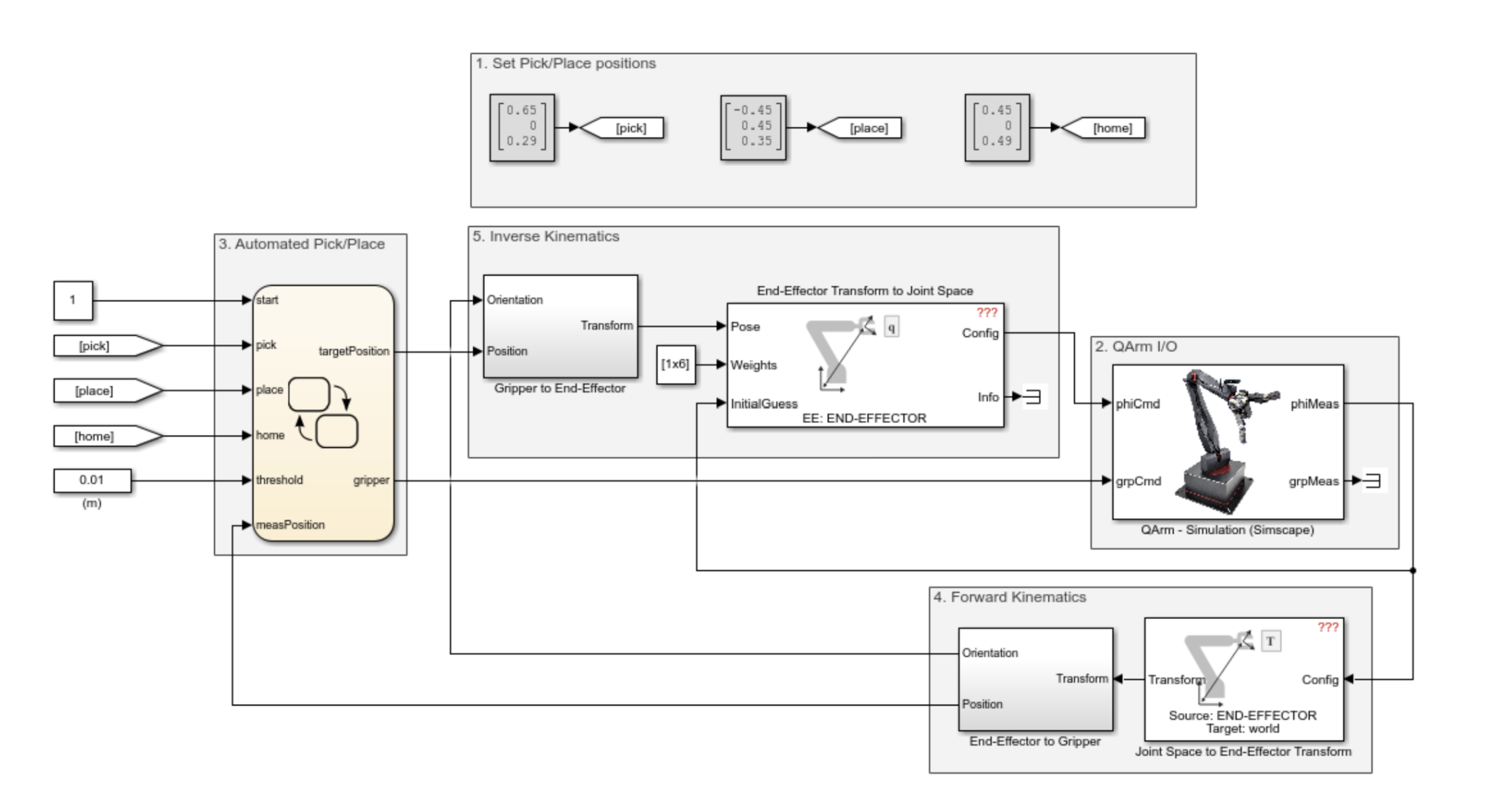
Building, Scaling, and Re-Engineering the Quanser 4 DOF Robotic Arm Overview

This presentation explores my experience with the Quanser QArm, a 4 DOF serial manipulator with tendon based gripper, showcasing its integration with Simulink, Python, and ROS for education and research.

How Does It Work?



Need for Advanced Educational and Research Robotics Platform

The robotics education and research landscape faced several significant challenges before the development of the Quanser QArm:

Reliability Issues

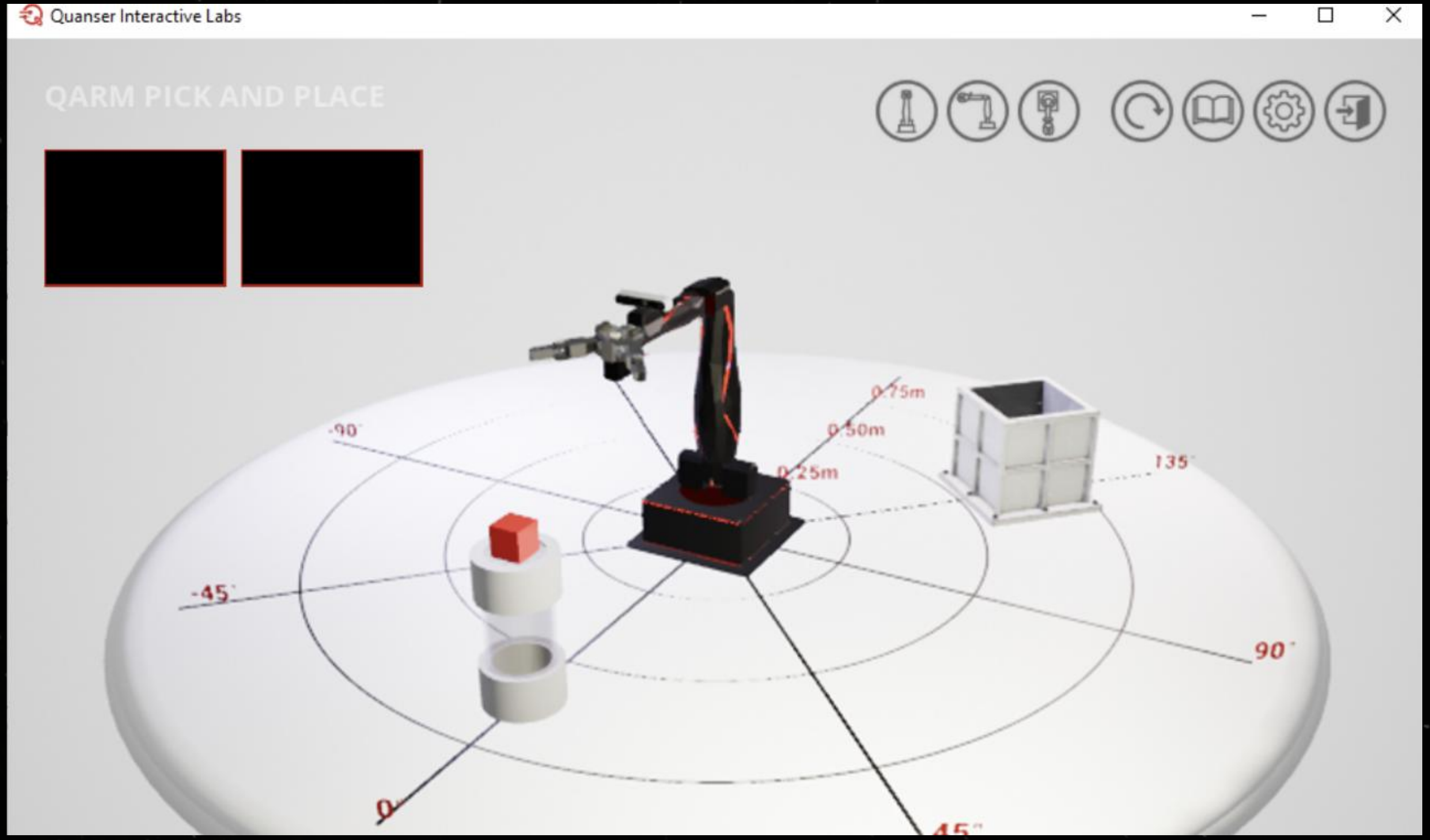
Students needed accessible yet sophisticated manipulators to bridge theoretical concepts with practical applications

Research Bottlenecks

Researchers struggled with platforms that restricted rapid prototyping and multi-language programming environments

Closed Architecture Limitations

Existing systems lacked the open framework needed for customization and expandability in academic settings



Develop a Modular, Scalable 4 DOF Robotic Arm

1

Design Versatile Manipulator

Create a robotic arm with large workspace and precise control capabilities to handle diverse educational and research applications

- Ensure accuracy within educational budget constraints
- Design for durability in high-usage academic environments

2

Enable Multi-Platform Integration

Develop hardware and software interfaces supporting multiple programming environments to maximize accessibility

- MATLAB/Simulink compatibility for control theory applications
- Python and ROS support for research flexibility

3

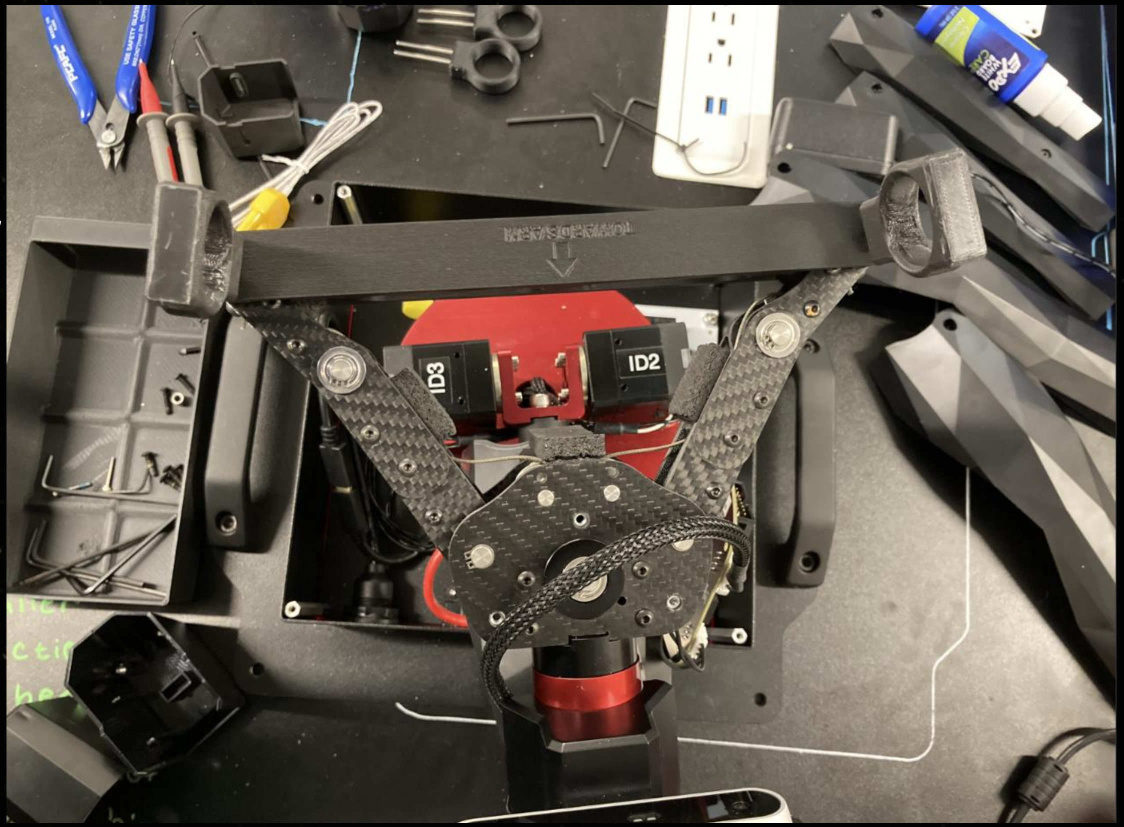
Support Advanced Research

Build a platform capable of supporting cutting-edge research in machine learning, assistive robotics, and automation

- Facilitate sensor integration for data-driven applications
- Enable real-time control algorithm deployment

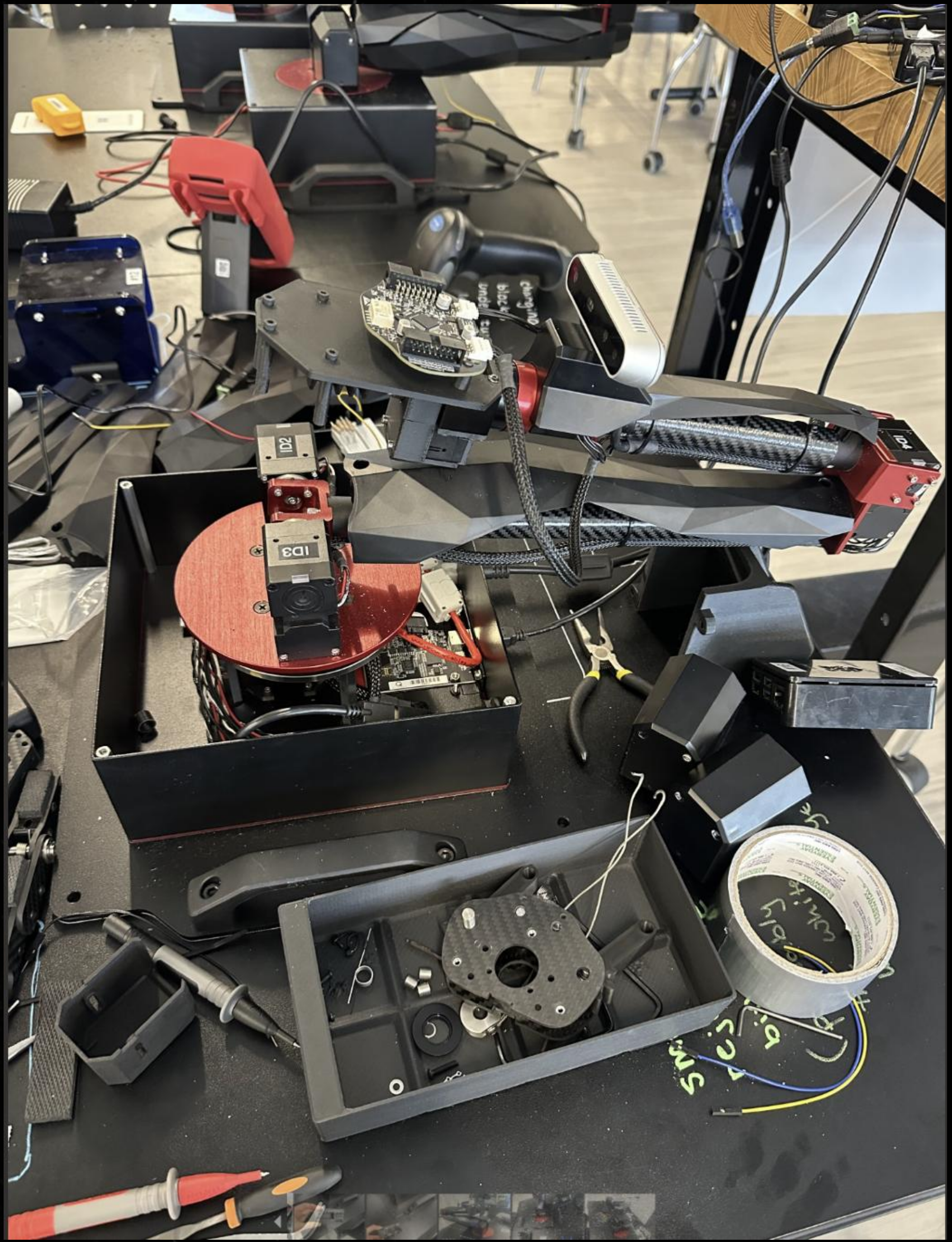
Issue Example: Diagnosing Inconsistent Gripper Failures on the Qarm

- Gripper behavior was unpredictable across identical arms:
  - Gripping failed at 45° one day, succeeded the next
  - Some units failed instantly with maxed current draw
- Connected multimeter in series and sampled in Python via INA219 ADC
- Confirmed software-reported values matched real current (±0.05 A)
- Disabled 0.3 A software current limit to investigate hardware limits:
  - 0.3-0.4 A: stable grip
  - 0.45 A: failure after ~10 sec
  - 1.1 A: failure after ~3 sec, required full reset



Issue Example: Diagnosing Inconsistent Gripper Failures on the Qarm

- Gripper servo powered via STM32-controlled VRM (1 A rated)PCB analysis showed no PTC or resettable fuse failure likely due to: VRM current limit or thermal shutdown or downstream IC protection
- The gripper shares power with the wrist motor (not isolated)
- Detected current spikes in gripper when shoulder moved → evidence of: Shared ground return path or poor decoupling
- Total theoretical stall current across all servos = 21.3A
- External PSU = 12 V, 16 A max → insufficient margin



What were the main sources of mechanical inefficiency in the original cable-driven gripper design?

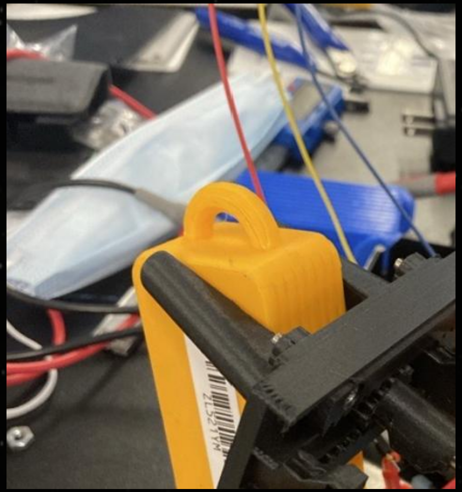
The original cable-driven gripper had several issues:

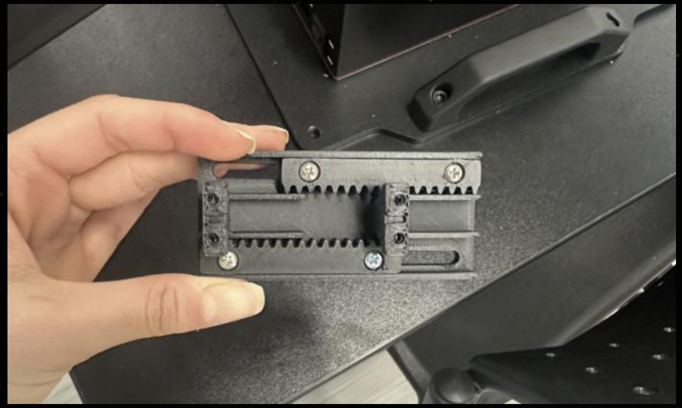
- **Nonlinear force transmission** due to elastic deformation in the cable.
- **Asymmetric tensioning**, causing imbalance between the two gripper fingers.
- **Excessive spring preload**, which pushed back against the motor and caused it to operate near stall continuously.
- **Manufacturing variability** in spring constant and cable length, leading to inconsistent current loads between arms.

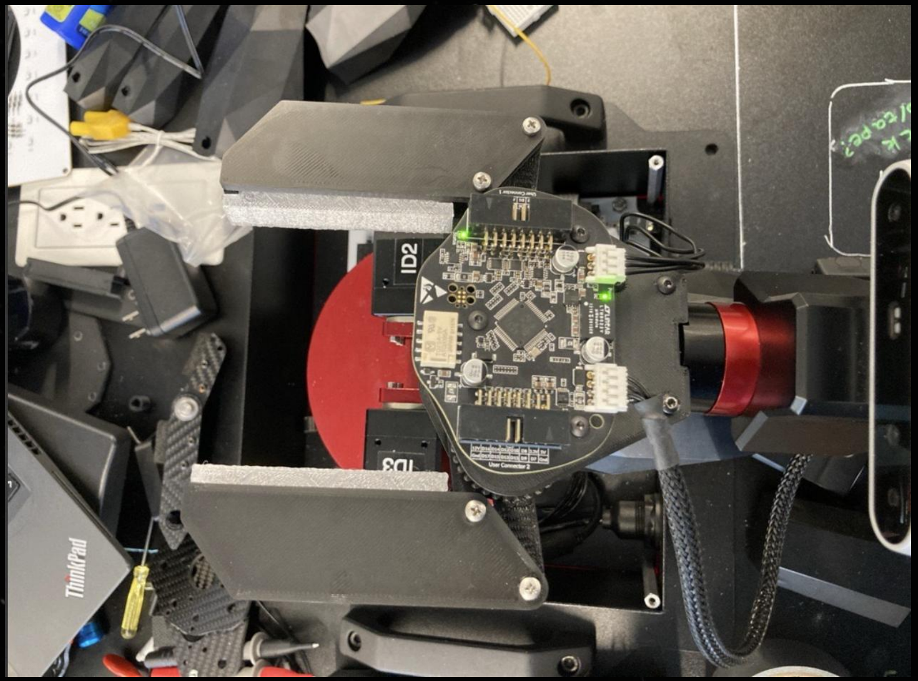
  All of these led to high and unpredictable current draw, and triggered premature overcurrent shutdowns.

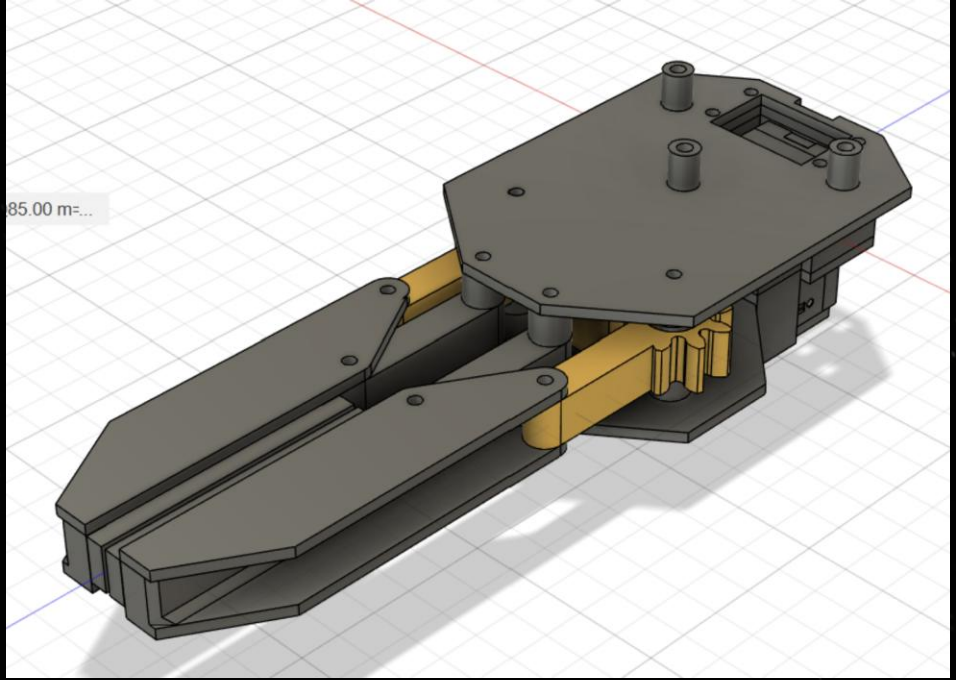
Gripper Redesign

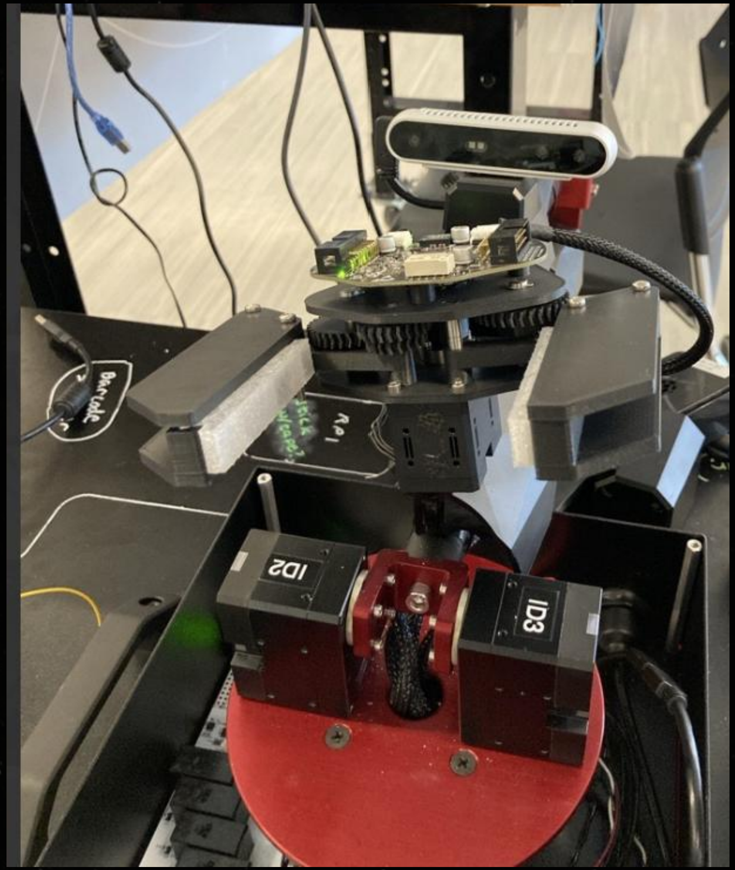
- Replaced Quanser’s cable-driven, spring-loaded gripper:
- Original: high preload = constant stall current
- Variable spring force and cable tension = inconsistent current draw
- 1st prototype: PLA rack-and-pinion
- Reduced current but backdrivable, poor holding torque
- Final: Worm gear + linkage + dovetail interface
- Non-backdrivable: no holding current needed
- Interchangeable TPU/silicone gripper pads for varied compliance
- Reduced current draw from 400-700 mA → ~130 mA consistently





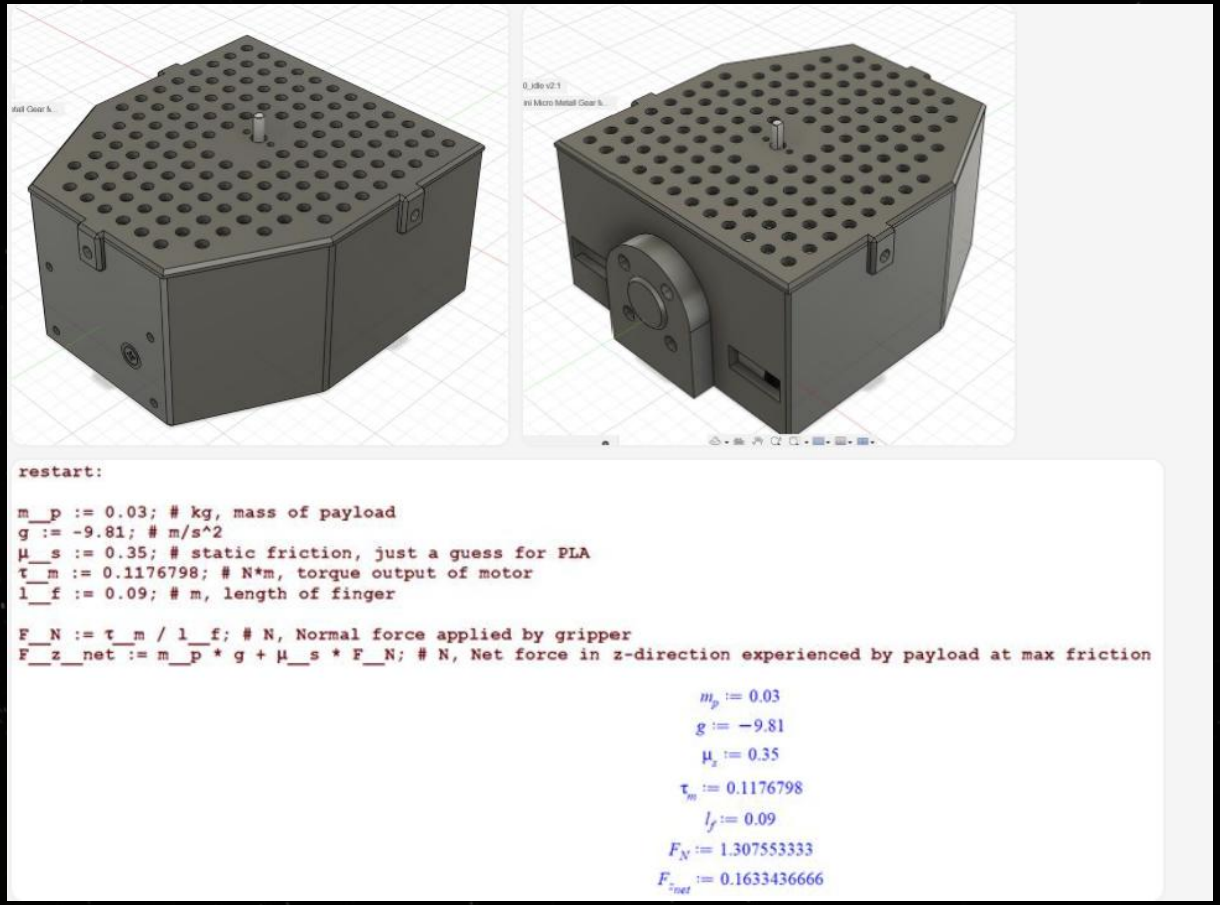


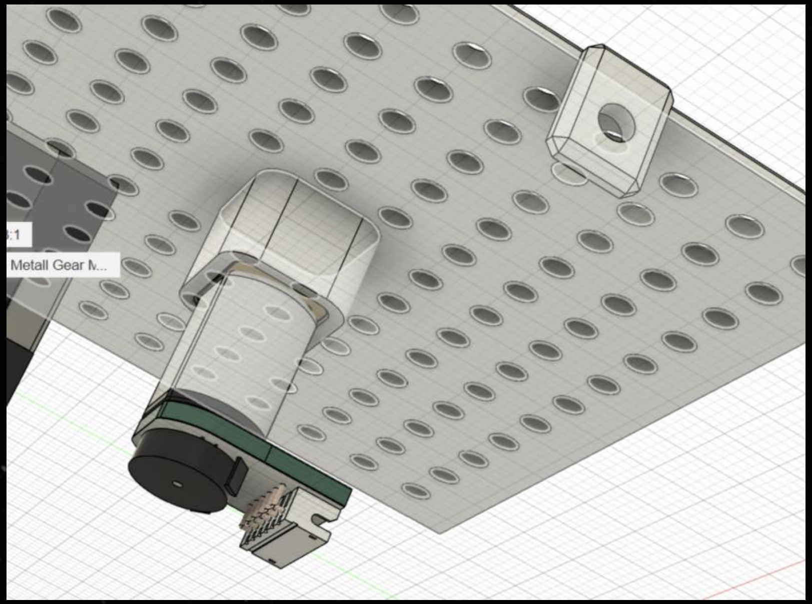

85.00 m=...


ID2
ID3

Gripper Redesign

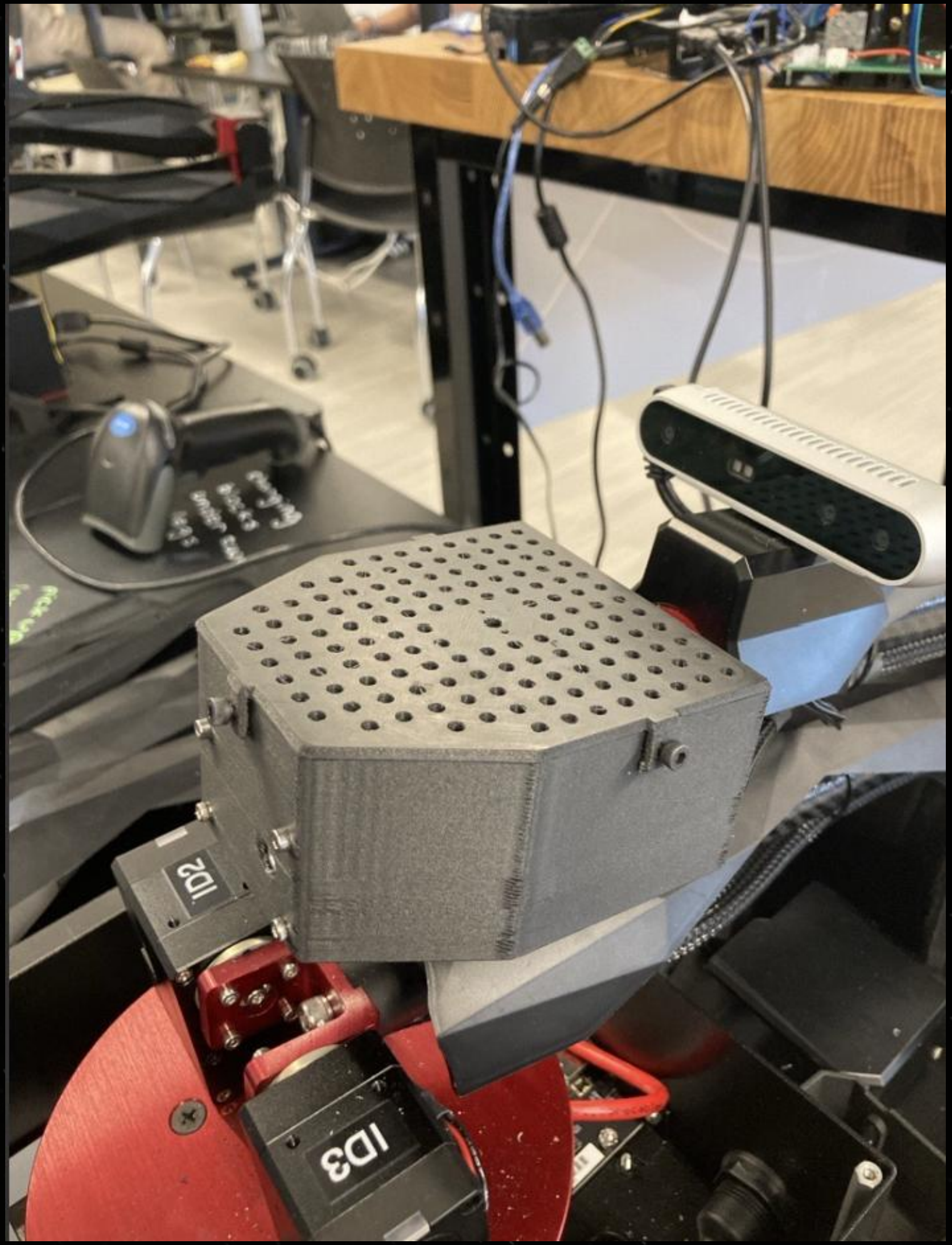
- Motor: N20 gearmotor (16 kg·cm stall torque, 100:1 gearbox)
- Control: DRV8871 H-bridge + Raspberry Pi GPIO
- Feedback: Quadrature encoder with pigpio (interrupt-based)
- Implemented Python PID loop:Closed-loop rotation control in ticks or degrees
- Modular mount: Captive nut inserts, M1.6 motor mounting, swappable gripper tops
- Designed for rapid student prototyping and testing





N20 Gearmotor Control

•Motor: N20 gearmotor (16 kg·cm stall torque, 100:1 gearbox)
•Control: DRV8871 H-bridge + Raspberry Pi GPIO Feedback: Quadrature encoder with pigpio (interrupt-based)
•Implemented Python PID loop:Closed-loop rotation control in ticks or degrees
•Tuned with anti-windup and derivative filtering
•Modular mount: Captive nut inserts, M1.6 motor mounting, swappable gripper tops
•Designed for rapid student prototyping and testing



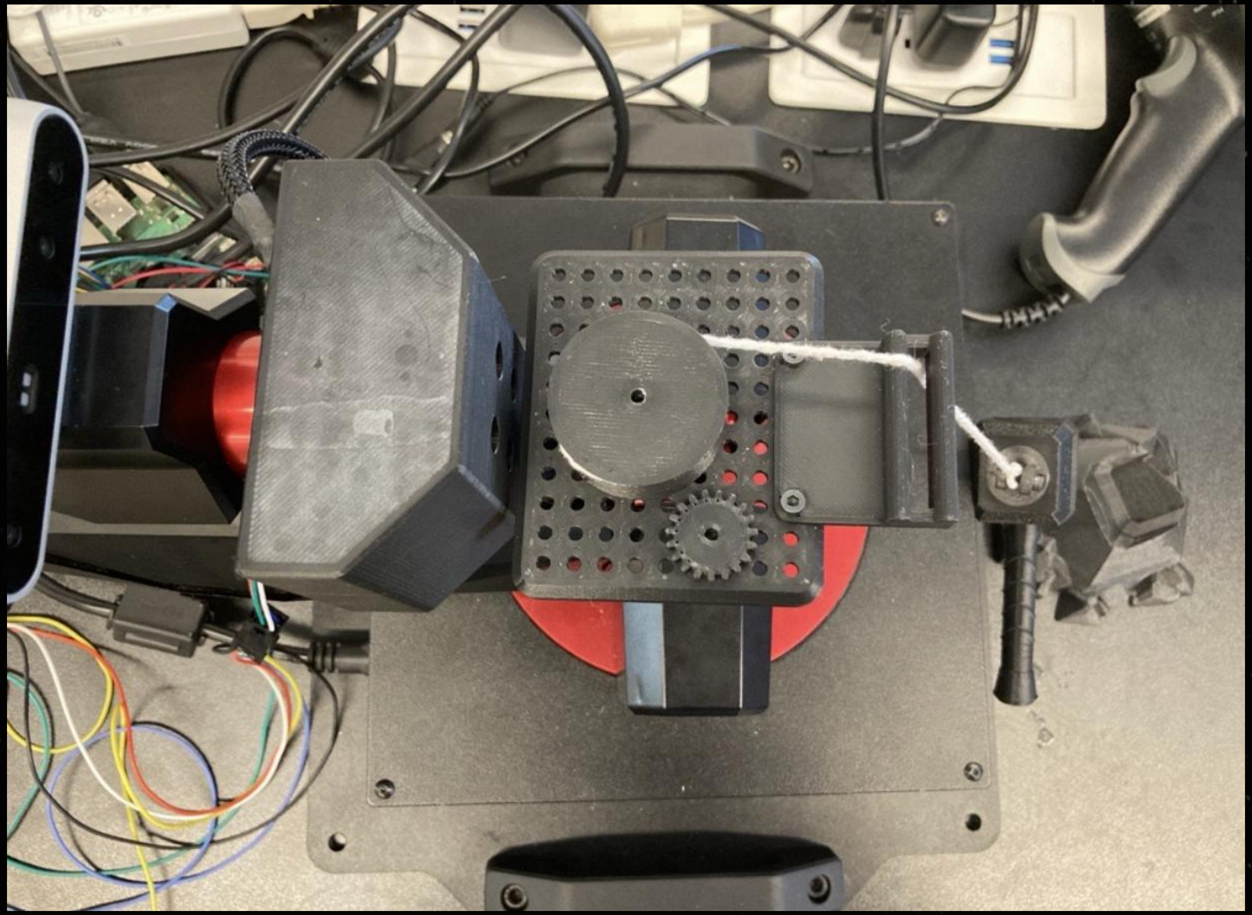
Evaluating cost, scalability, and educational value when choosing between retrofitting vs. redesigning

- I compared the BOM and fabrication costs of retrofitting (e.g., replacing the servo with a lower-draw alternative) versus a ground-up redesign using N20 gearmotors. The redesign had a marginally higher one-time cost but offered modularity, better current characteristics, and a more maintainable architecture.
- For educational value: The modular mount allows students to test different drive mechanisms (worm, rack, direct drive). The encoder integration enables exposure to closed-loop control.
- Using Pi GPIO and open-source drivers gave students hands-on embedded systems experience.
- Scalability was supported by low-cost, repeatable parts (PLA prints, M3/M1.6 fasteners, off-the-shelf drivers.

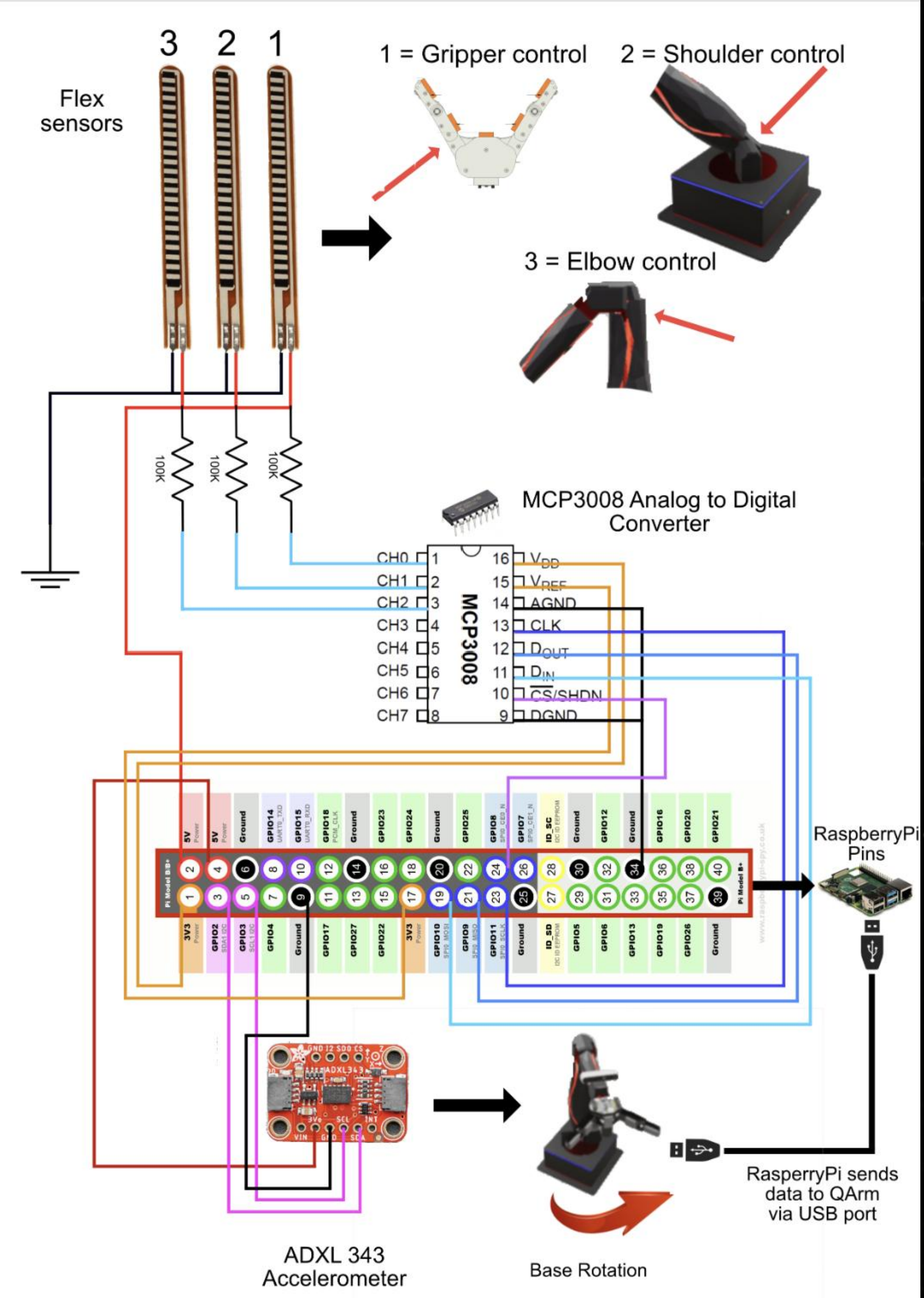
Documentation

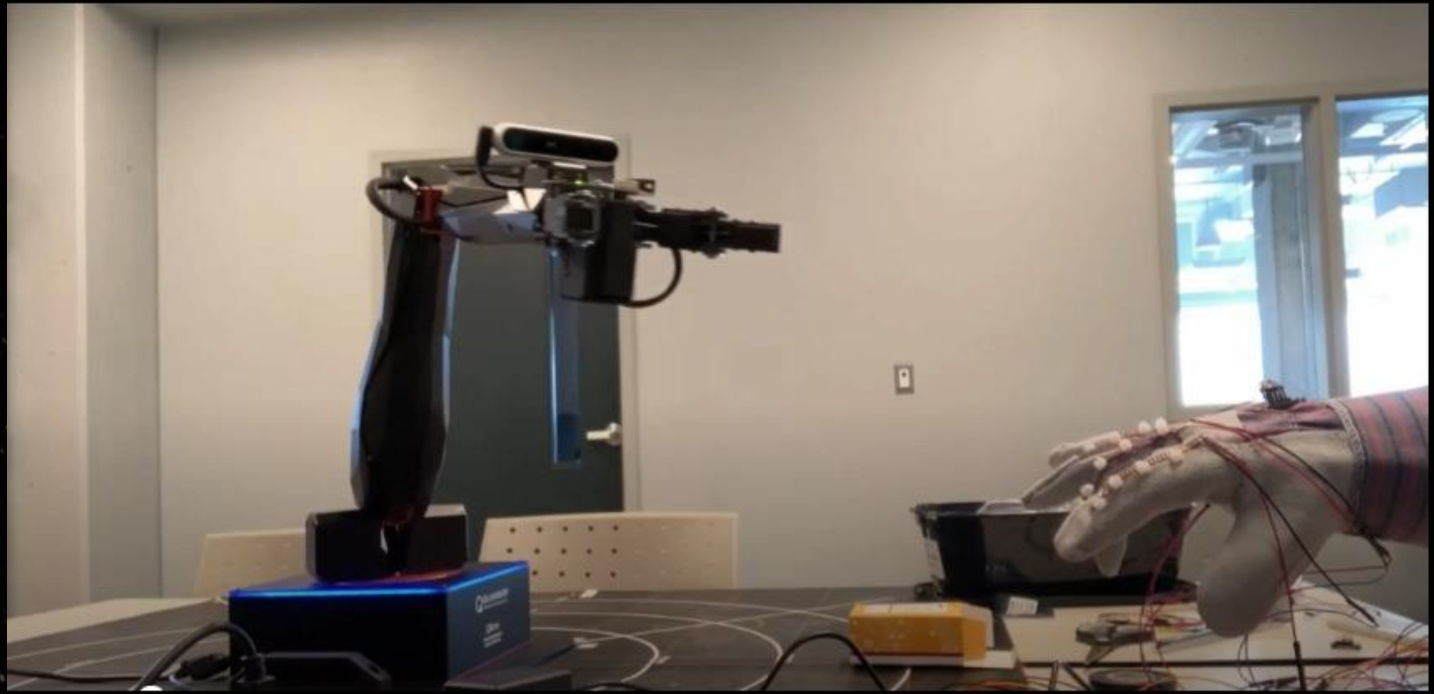
Documentation

Documentation



Documentation





Result: Impact on Education and Research

30+

ARMS

Adopted all Qarms for hands-on robotics education

100%

Error Free

Reduction in gripper failure

6+

Courses

Using QArm for vision-based and voice-controlled assistive robots

100%

Human Centered design

Improvement in understanding of kinematics, control, and path planning

Alignment with Robotics Research Focus

Shared Technological Values

K-Scale Labs and the Quanser QArm project share fundamental design philosophies:

Scalable Robotics Architecture

Both prioritize modular design approaches that enable scaling from educational to research-grade applications

Iterative Development Process

Establish structured feedback mechanisms for continuous hardware and software improvement cycles

Human-Robot Interaction

Shared goals in advancing intuitive interfaces between humans and robotic systems

Adaptive Control Systems

Mutual emphasis on developing responsive control algorithms that adapt to changing conditions