

# Robot Controller through Gestures

## Team 41

Eric Zhou(yirui2), Guang Yin(guangy2), Haoduo Yan(haoduoy2)

TA: Hanyin Shao

Spring 2022

# Introduction

Traditional Robot Controller

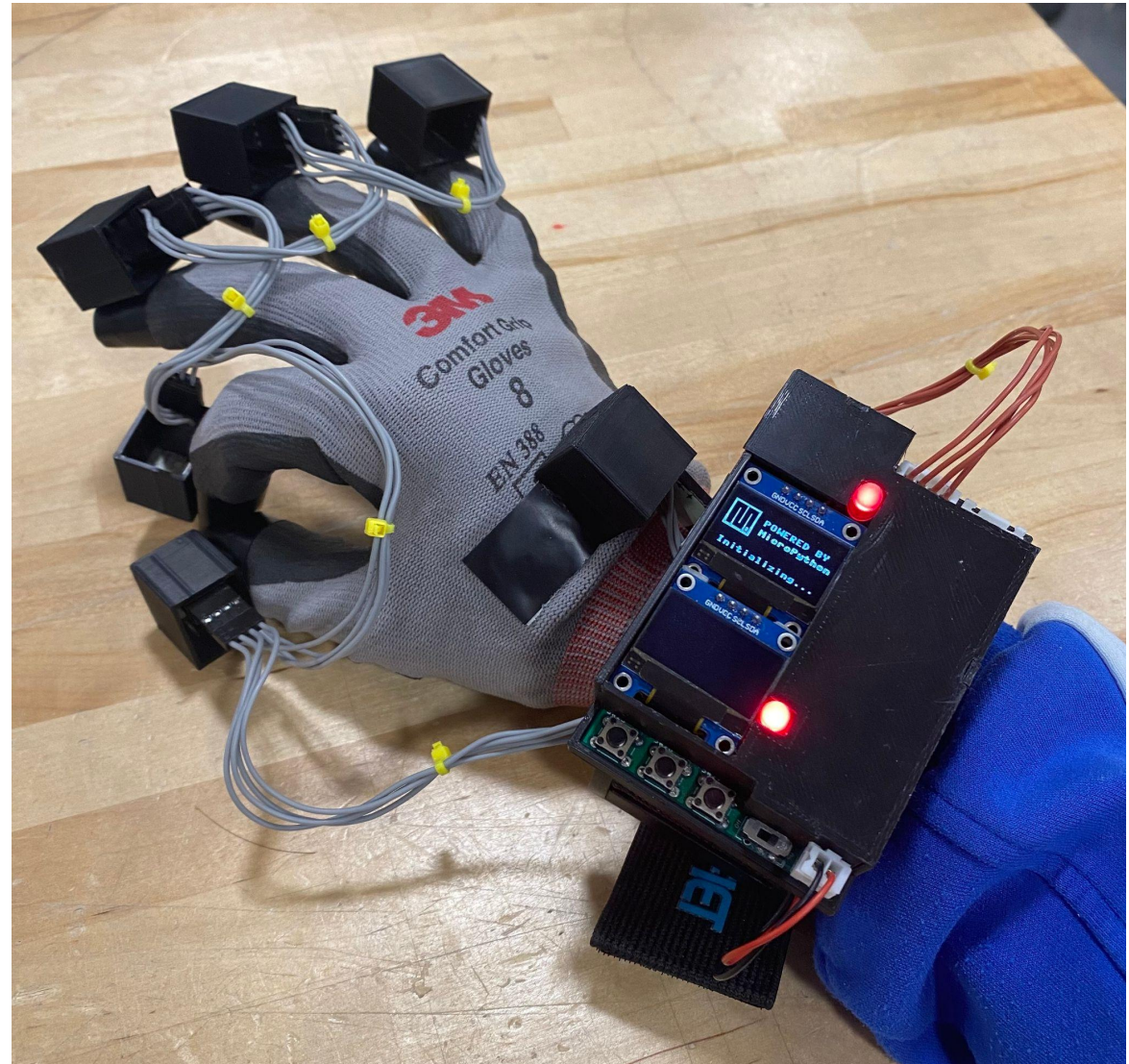


- ✗ Unresponsive Enough
- ✗ Not Easy to Use
- ✗ Unengaging

Can we have a **cooler** and more **intuitive** tool?



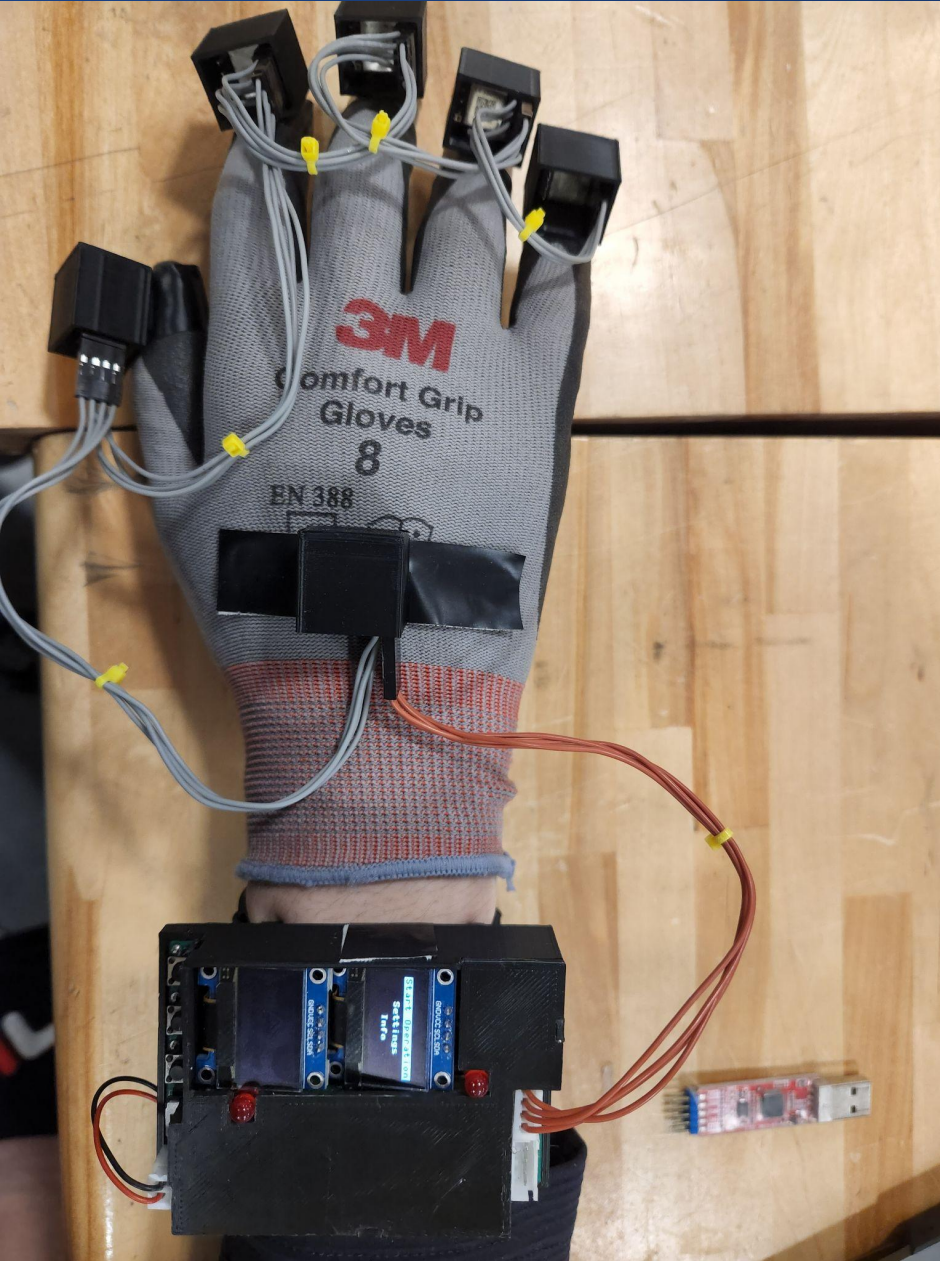
# No Problem!



# Objective







- The position and orientation of fingers and palm can be collected through IMUs.
- Hand movements and gestures can be translated to robot control actions respectively.
- Users can get feedback through the glove.

# Design



Fig. Hardware Overview

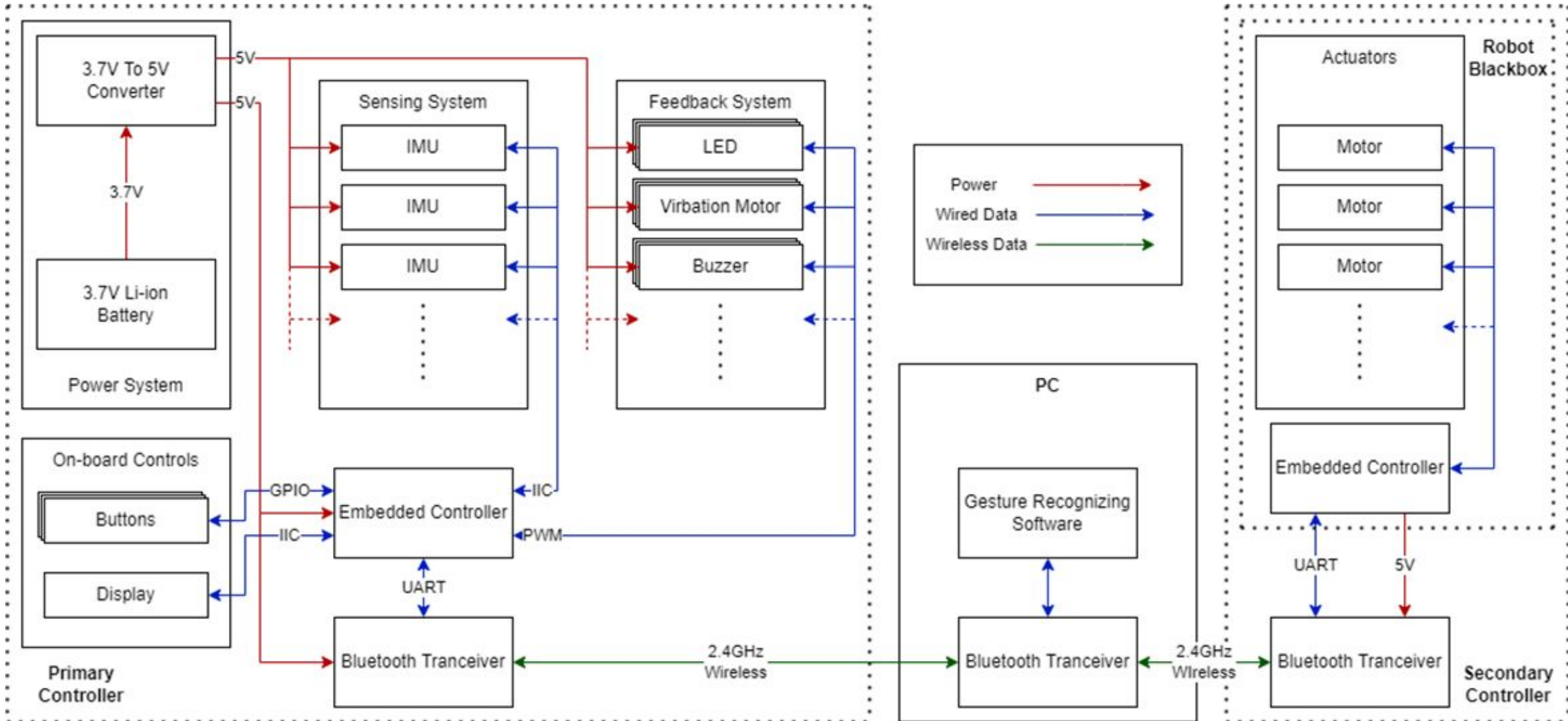
Control Board

## Subsystems

- Human Positioning System
- Gesture Control System
- Robot Feedback System



# Block Diagram





# PCB Board Overview

Vibration Motor  
I2C Ports  
ESP32 Main  
Reset Buttons  
LED

Buzzer

Front  
(w/o Screen)

Front  
(w/ Screen)

ESP32 Peripheral  
Buttons  
OLED Screen  
Battery

Micro-USB port

Back

Power Regulator  
USB to UART



# Human Positioning System



TERMINAL	PROBLEMS	277	OUTPUT	DEBUG CONSOLE	GITLENS	JUPYTER
Middle	roll: -19.756, pitch: -14.161, yaw: 45.235. Q0: 0.911, Q1: -0.110, Q2: -0.178, Q3: 0.356, AX: 2.359, AY: -3.163, AZ: 8.800					
Ring	roll: -19.188, pitch: 17.280, yaw: 53.400. Q0: 0.860, Q1: -0.214, Q2: 0.058, Q3: 0.460, AX: -2.866, AY: -3.034, AZ: 8.728					
Little	roll: -11.871, pitch: -7.514, yaw: -9.962. Q0: 0.988, Q1: -0.108, Q2: -0.056, Q3: -0.093, AX: 1.254, AY: -1.948, AZ: 9.307					
Thumb	roll: -12.265, pitch: -28.489, yaw: 41.017. Q0: 0.912, Q1: -0.011, Q2: -0.265, Q3: 0.313, AX: 4.527, AY: -1.766, AZ: 8.144					
Hand	roll: -1.835, pitch: -9.968, yaw: -41.714. Q0: 0.930, Q1: -0.046, Q2: -0.076, Q3: -0.356, AX: 1.704, AY: -0.306, AZ: 9.671					
Index	roll: -12.678, pitch: -6.871, yaw: 53.823. Q0: 0.888, Q1: -0.071, Q2: -0.103, Q3: 0.443, AX: 1.144, AY: -2.091, AZ: 9.302					

Fig. Gesture data collected from IMUs and sent to PC through Bluetooth

**Euler Angles:** Roll, Pitch, Yaw, in degrees

**Quaternions:** Q0, Q1, Q2, Q3

**Acceleration:** AX, AY, AZ, in  $\text{m/s}^2$



- Using L2 algorithm to recognizes gestures



- Translates gestures into robot commands



- Sends commands to robot via Bluetooth

## 0) Prerecord Gestures

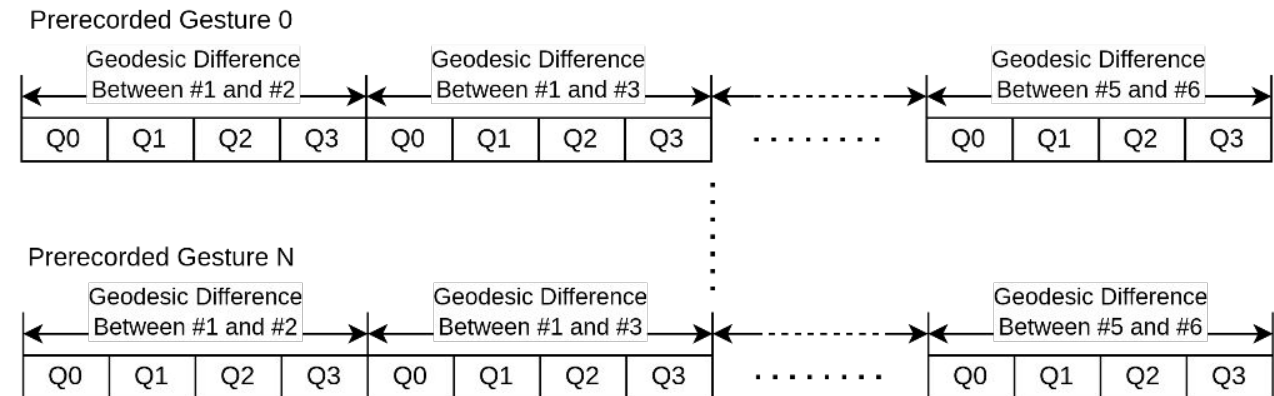


Fig. Principles of L2 Algorithm

## 1) Get quaternions

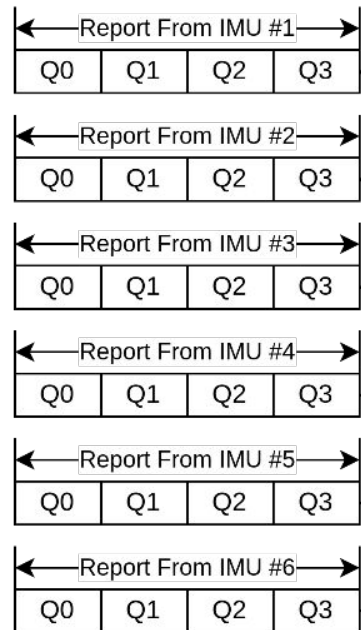


Fig. Principles of L2 Algorithm



## 2) Calculate geodesic distance

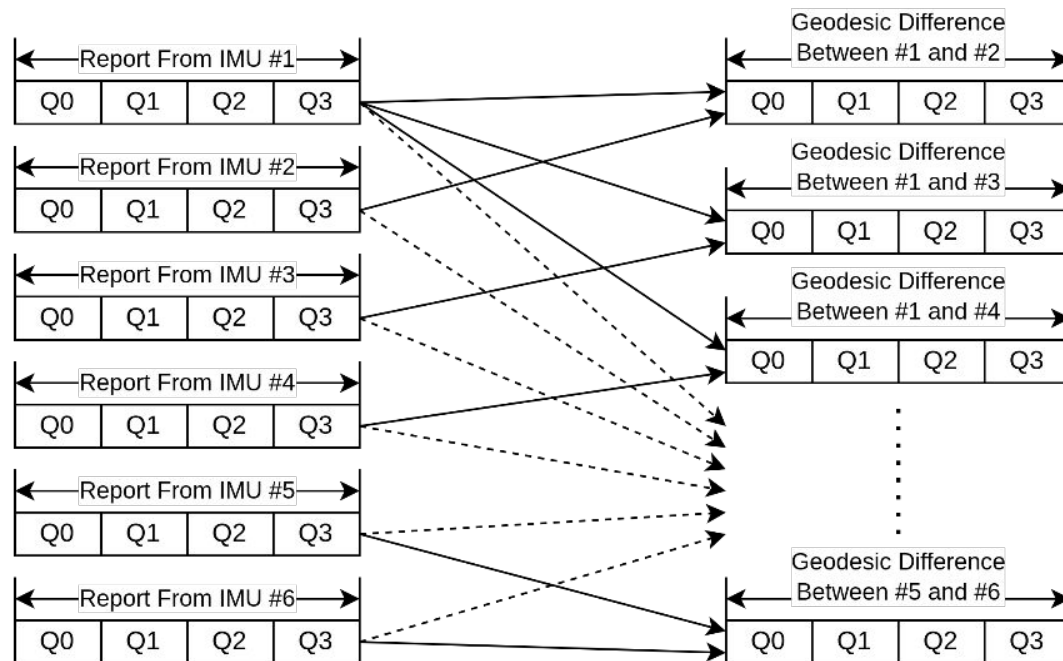


Fig. Principles of L2 Algorithm

## 3) Concatenate into one vector

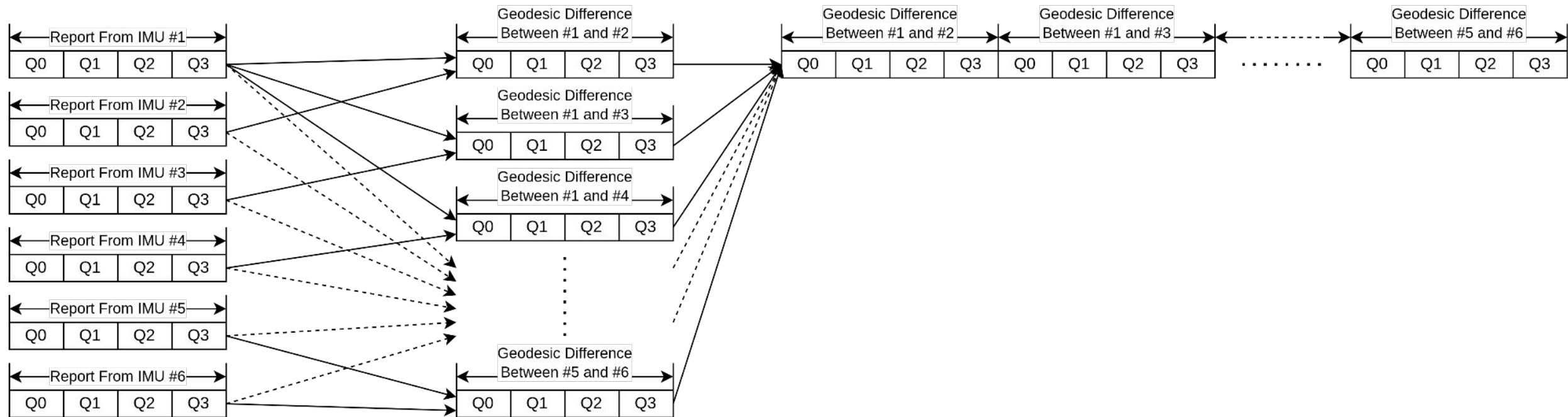


Fig. Principles of L2 Algorithm

## 4) Compute the L2 distance

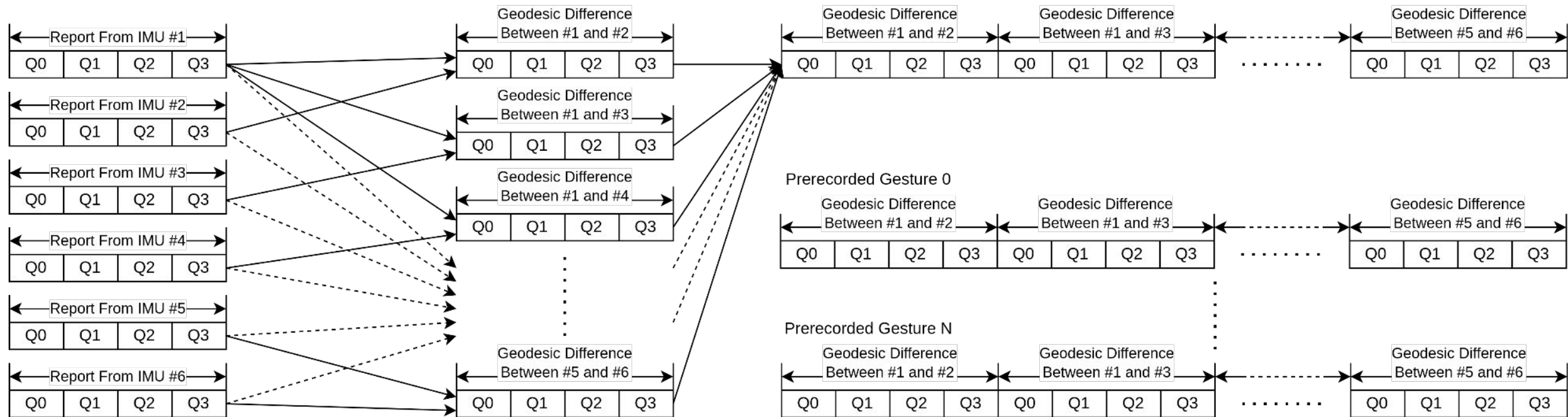
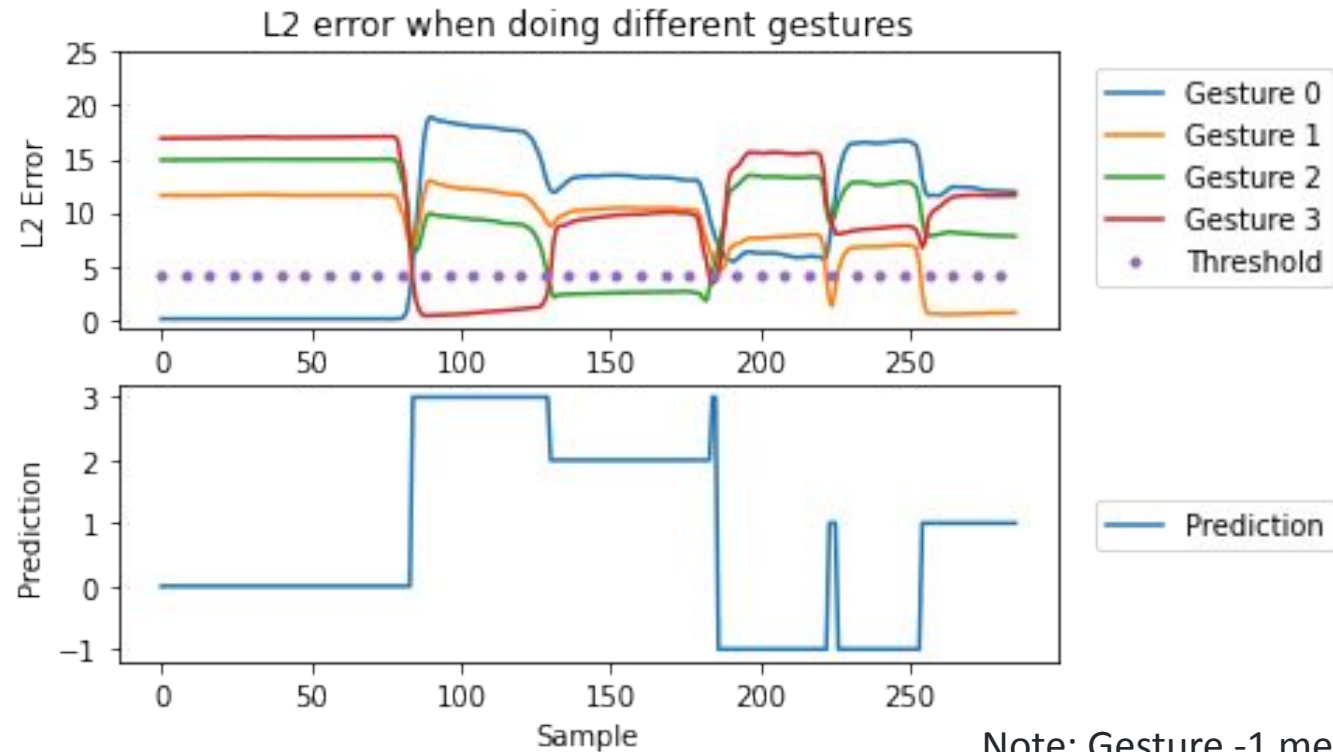


Fig. Principles of L2 Algorithm



## 5) Predict the gesture



Note: Gesture -1 means no prediction

Fig. Example of L2 Algorithm in Operation

# Example of Saved Gestures



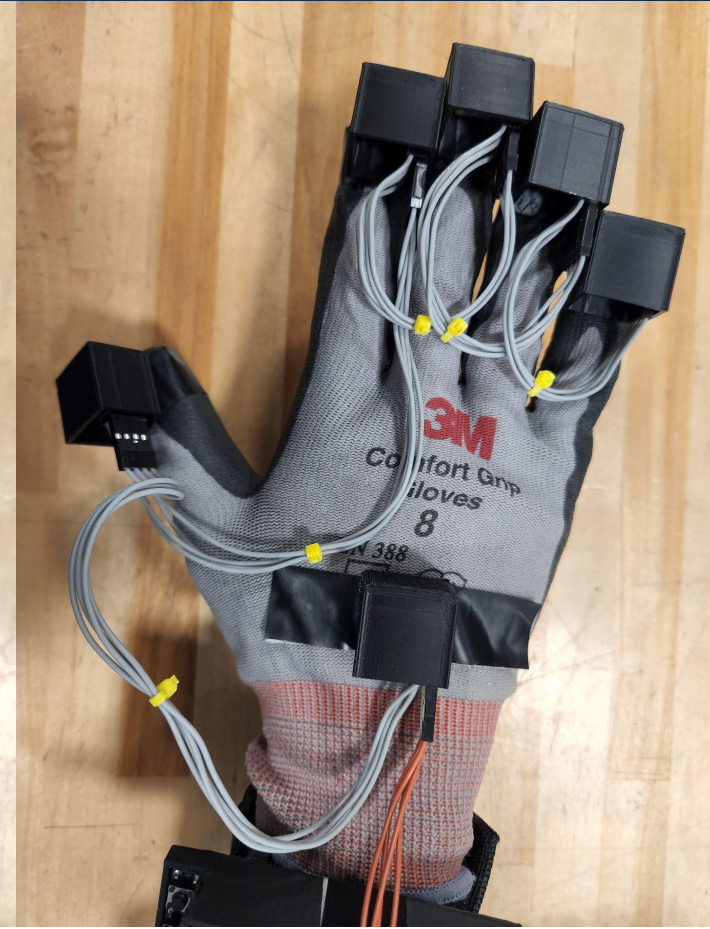
**Hold**  
**(Gesture 0)**



**Chassis**  
**(Gesture 1)**



**Gimbal**  
**(Gesture 2)**



**Shoot**  
**(Gesture 3)**



# More About L2 Algorithm



Example output of software running on the PC

```
Gesture 0 l2: 21.574106040980926
Gesture 1 l2: 15.935477902904832
Gesture 2 l2: 14.441595901942796
Gesture 3 l2: 0.608011788139928
```

Prediction: 3



```
Gesture 0 l2: 12.003243266082979
Gesture 1 l2: 0.15055034716838211
Gesture 2 l2: 10.151618564238449
Gesture 3 l2: 11.415297980643851
```

Prediction: 1



```
Gesture 0 l2: 29.310272787902104
Gesture 1 l2: 22.60062787570088
Gesture 2 l2: 14.549955075964574
Gesture 3 l2: 34.62843018384689
```

Prediction: 404



Fig. Examples of L2 Algorithm in Operation

Note: Threshold: 4, Prediction 404 means no prediction



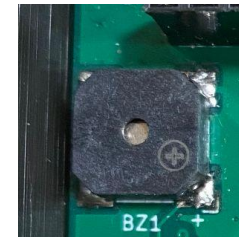
Feedbacks include:

- Controller operating status
- Bluetooth connection status
- Messages from robot
- Possible warnings and errors

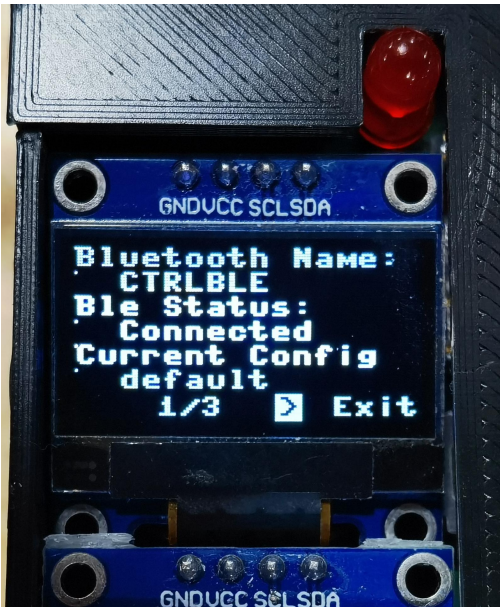


Feedback will be displayed through:

- LED
- Vibration motor
- Buzzer
- Display



# More About Display



Create Config  
Files Onboard



Showing Operation  
Status

Showing System  
Information

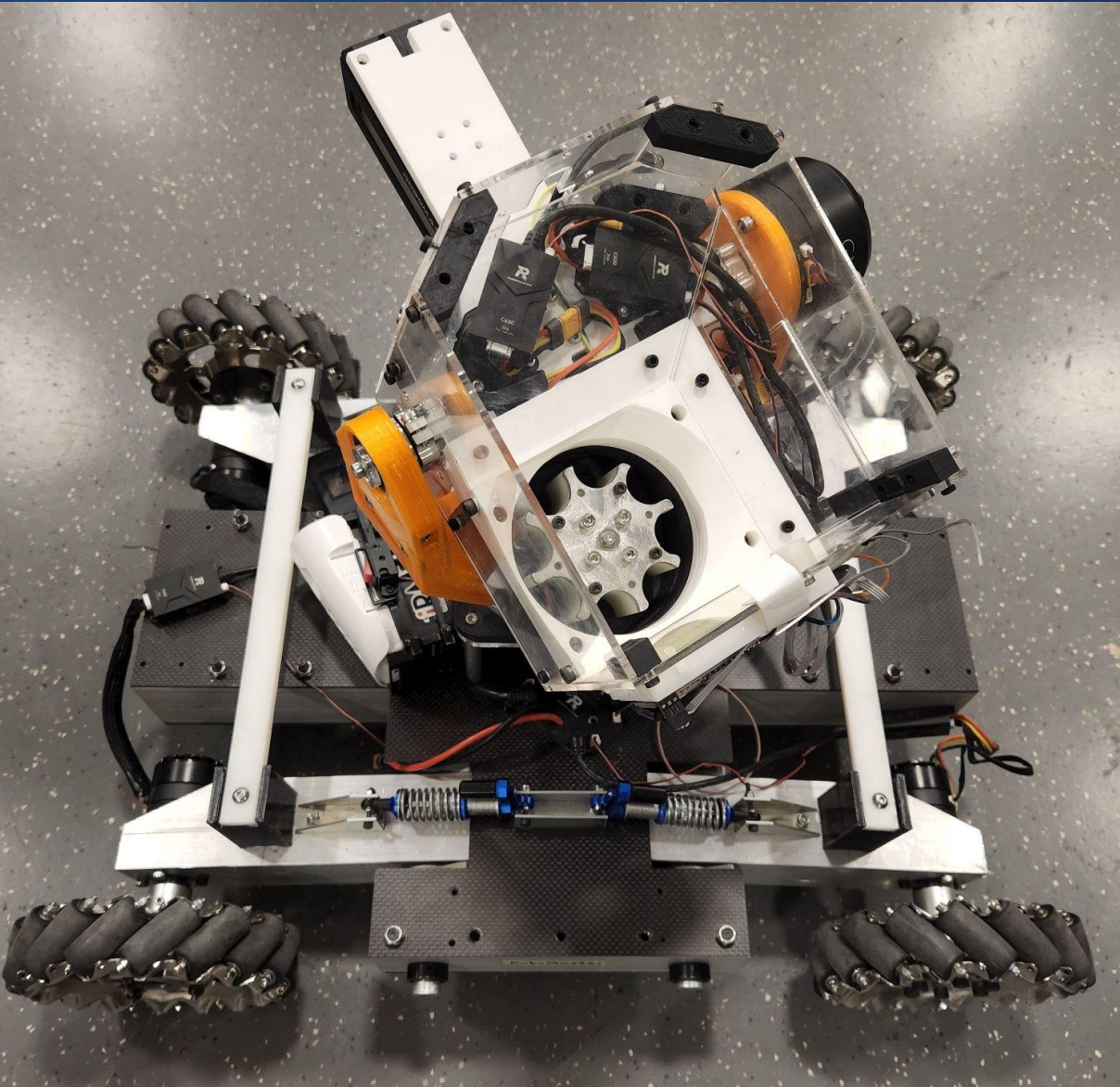


Manage Config  
Files Onboard





# Functionality & Instruction Set



**Hold**

Stop all action

**Chassis**

Move forward or back

Turn left or right

Rotate clockwise or  
counter clockwise

**Gimbal**

Move up or down

Move left or right

**Shooter**

Shoot





# Conclusion

## Successes

- Controller is fully functional.
- Also universal and customizable.
- We make use of the IMUs and identify gestures with high accuracy.

## Challenges

- Bluetooth connection is unstable.
- Readings from IMUs are greatly influenced by the magnetic field of surrounding environment.
- Drawing the PCB board according to our needs.

1. Remove intermediate PC
2. Build a UI for recording more gestures
3. Revise the gesture recognition algorithm
4. Reduce magnetic interference
5. Extend battery life
6. Tidier exterior design

The End