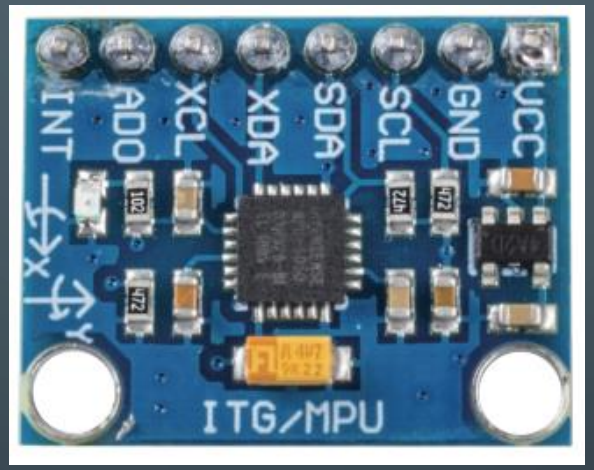
SENSOR MEASUREMENT PROJECT

BY ANDREW SCHALK, LIAM MCGUIRE, AND CHENG CHEN

MPU 6050

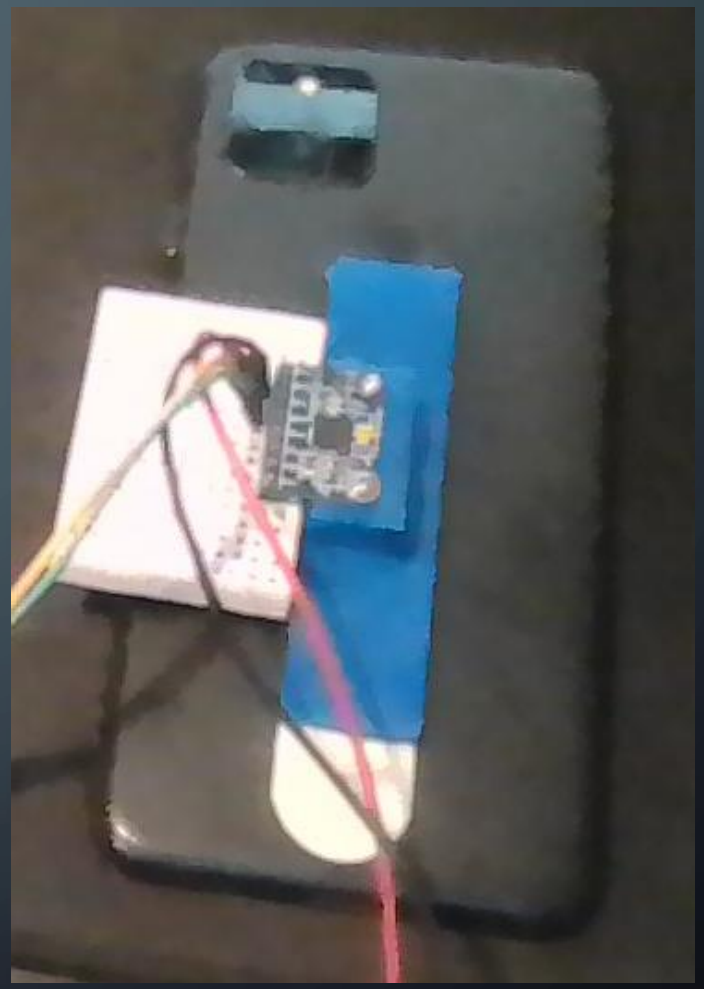
- 6 DOF gyroscope and accelerometer



MPU 6050

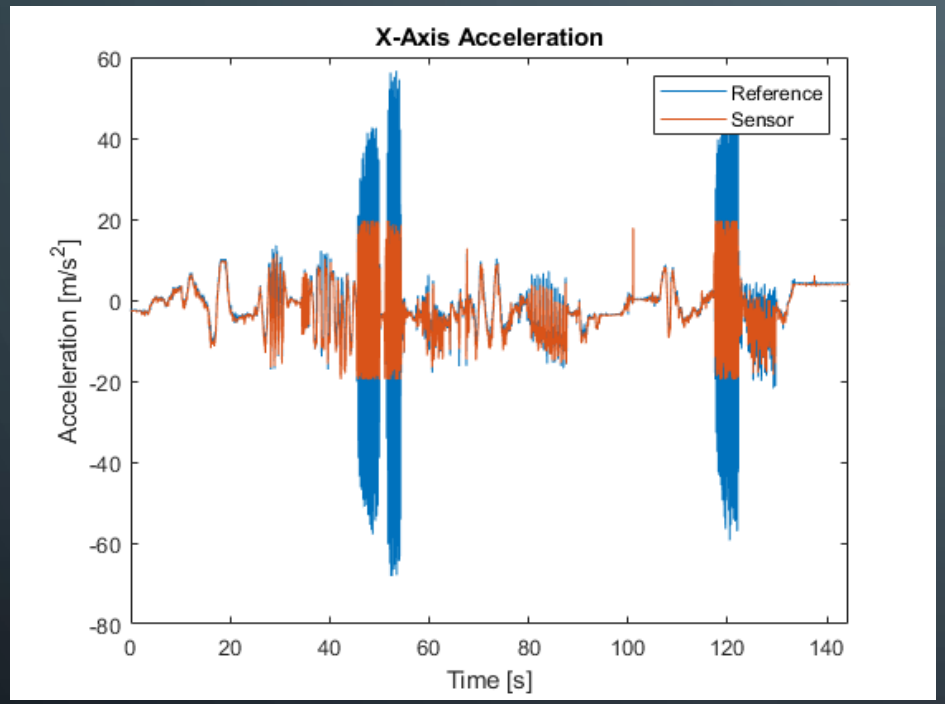
Testing Setup

- Ground Truth: Smart phone with sensor LSM6DSR
  - Assumption: Will be accurate enough to act as effective ground truth
  - Accelerometer Range: $\pm 157 m/s^2$
  - Gyroscope Range: $\pm 2000°/s$

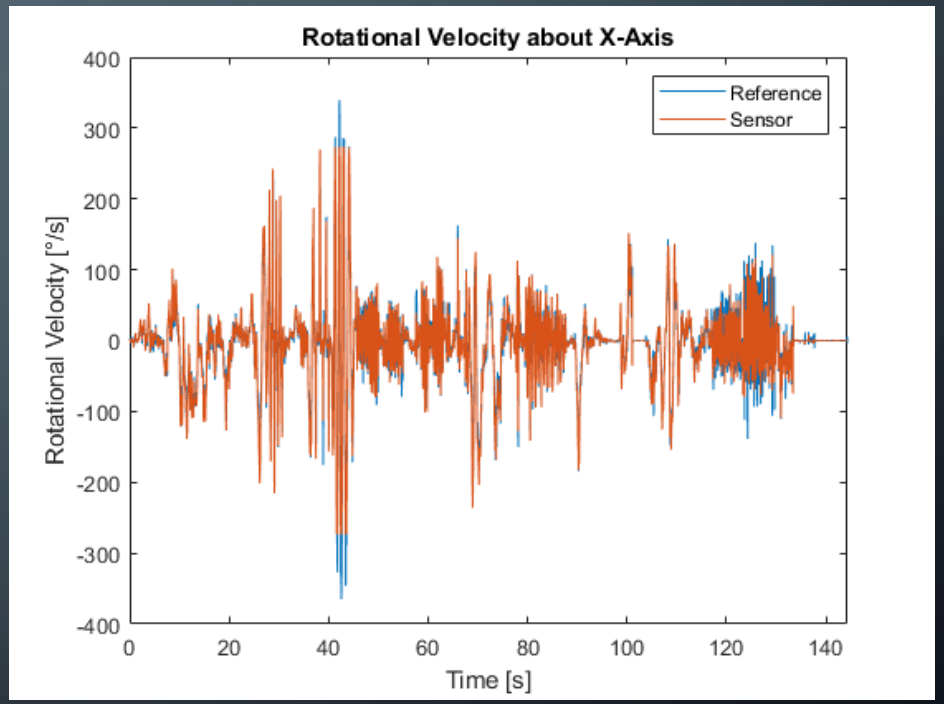


MPU 6050

Testing Data



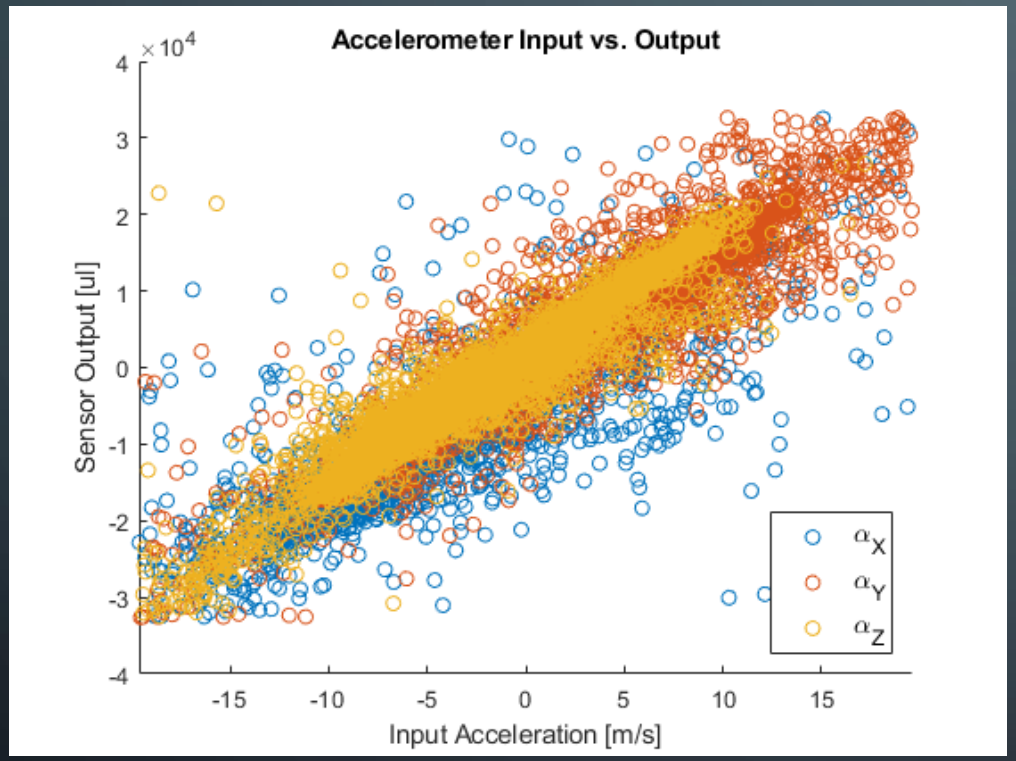
MPU Range: $\pm 19.62\ m/s^2$

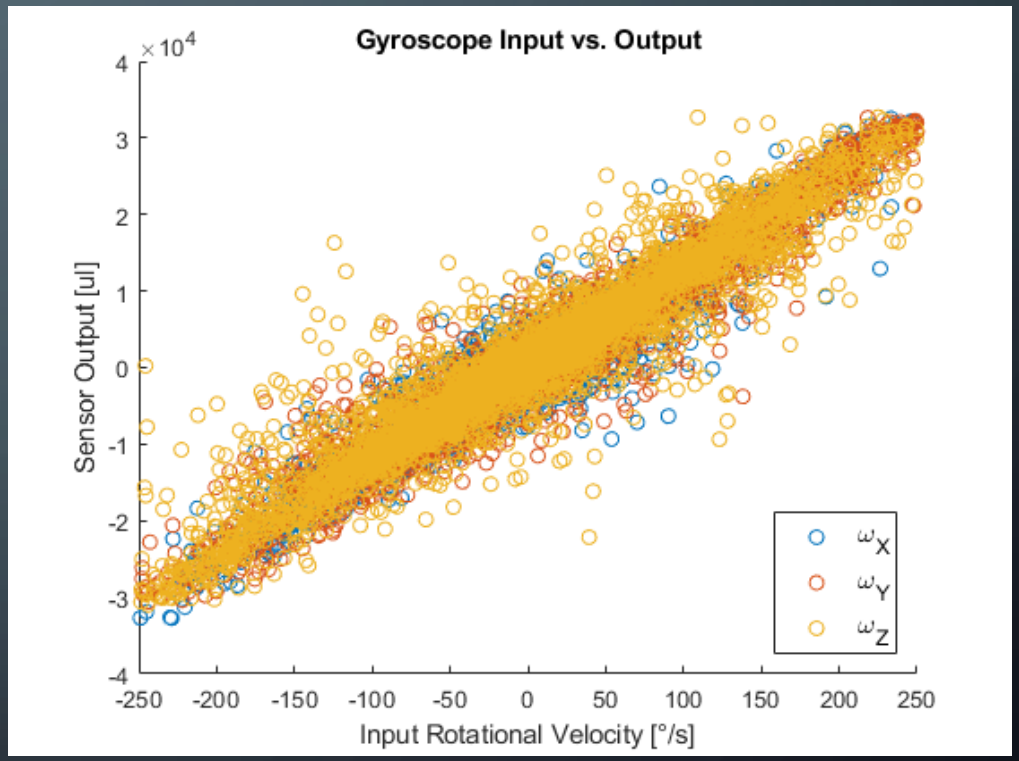


MPU Range: $\pm 250°/s$

MPU 6050

Testing Data

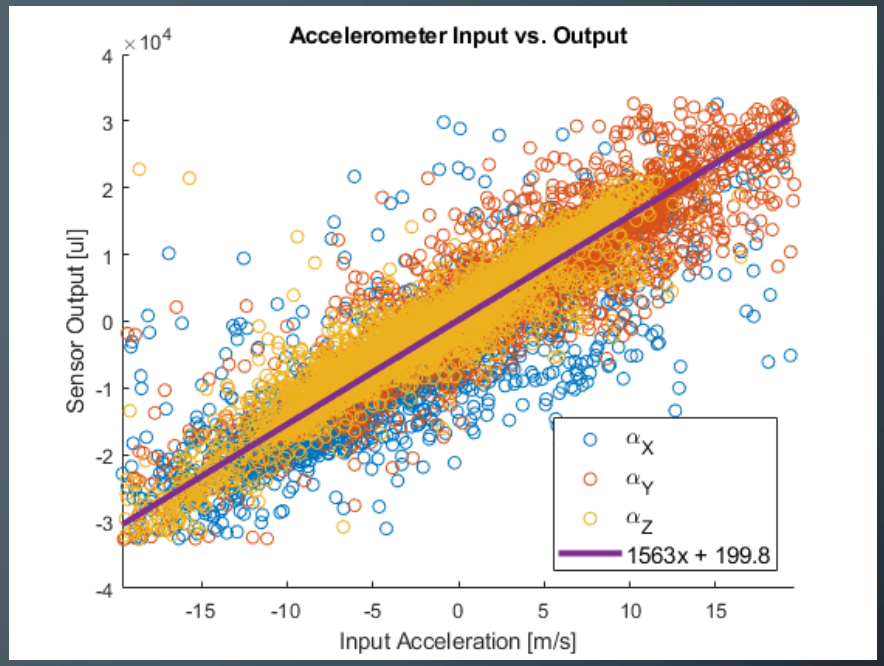


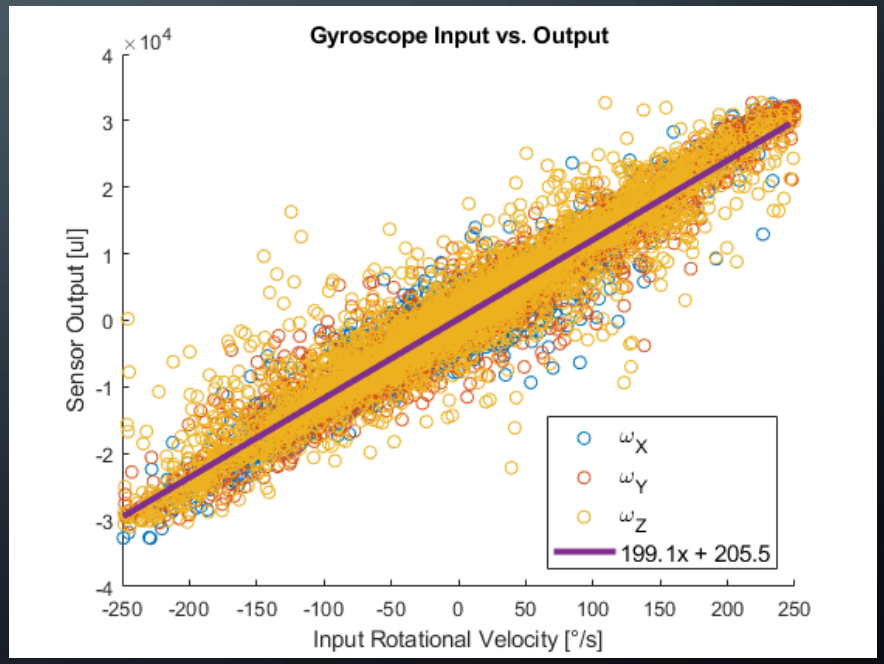


MPU 6050

Test Results

- Sensitivity
  - Accelerometer: 1563 [ul/m/s^2]
  - Gyroscope: 199.1 [ul/°/s]
- Accuracy (Measurement-Tolerance)
  - Accelerometer
    - Standard Deviation: 1.656 [m/s^2]
    - Max Error: 33.08 [m/s^2]
  - Gyroscope:
    - Standard Deviation: 9.011[°/s]
    - Max Error: 155.3 [°/s]

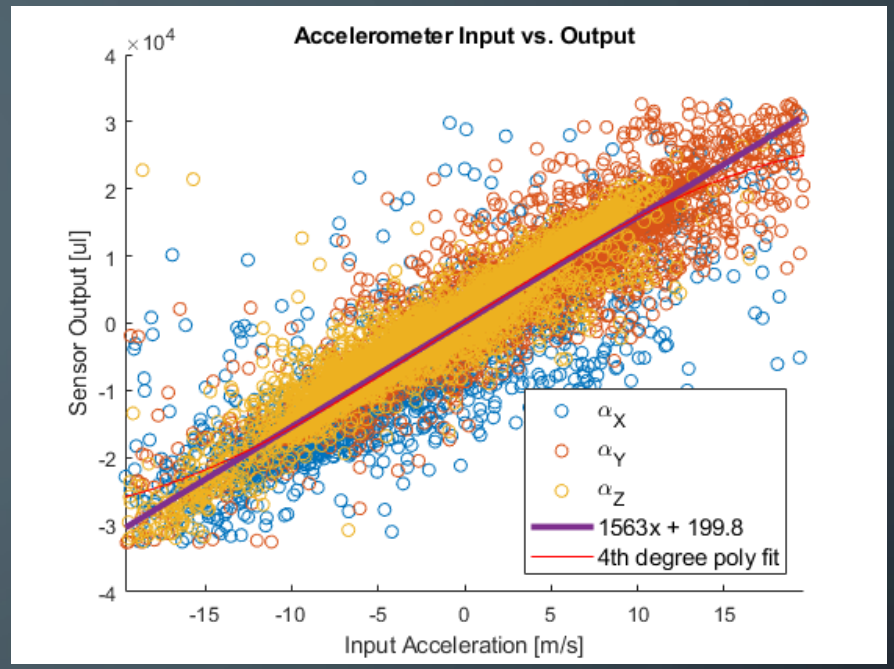


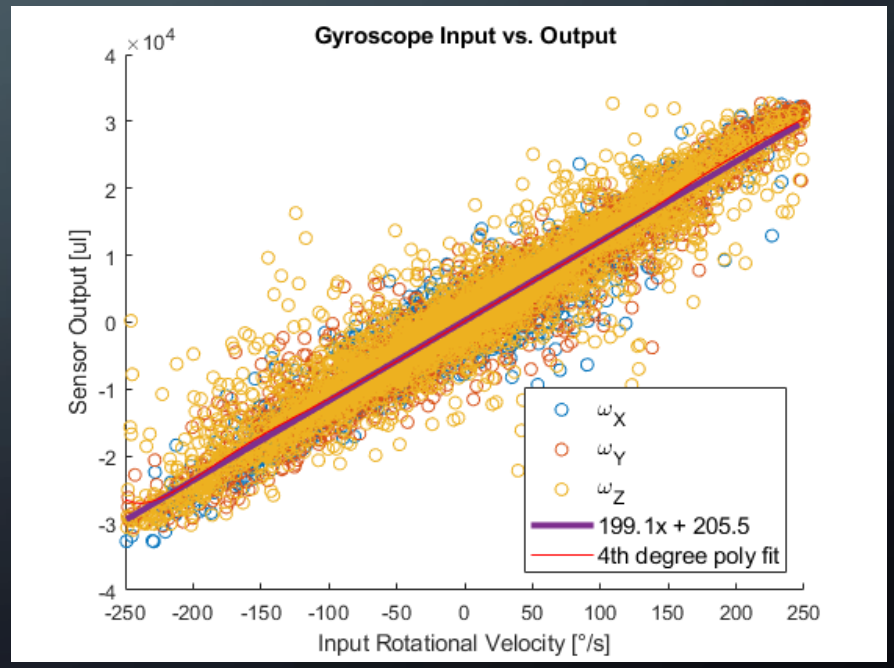


MPU 6050

Test Results

- Non-Linearity (max deviation from linear fit)
  - Accelerometer: 1.651 [m/s^2]
  - Gyroscope: 5.192 [°/s]





MPU 6050

Main Take-aways

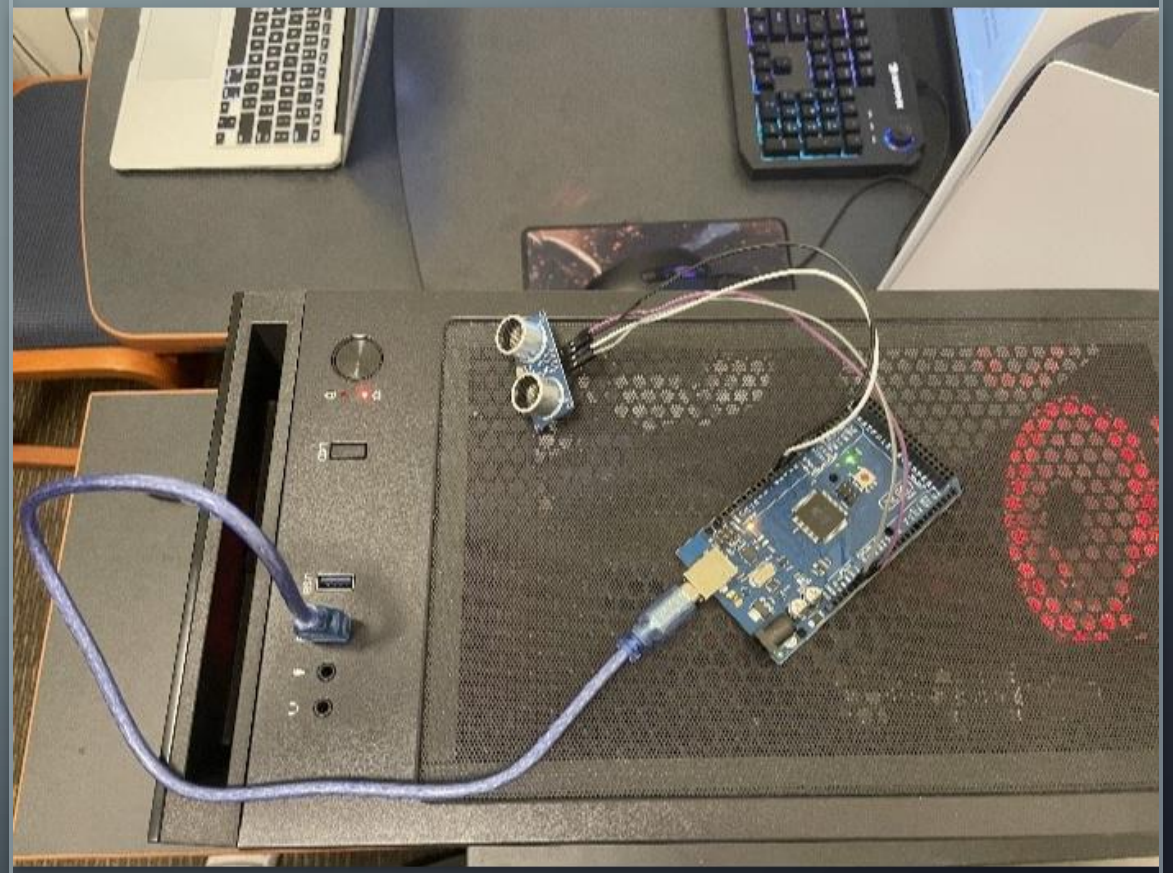
- Highly linear output
- Reasonably accurate and sensitive for some applications

ULTRASONIC SENSOR

- Sends pulse signals and detects if there is a return signal

ULTRASONIC SENSOR

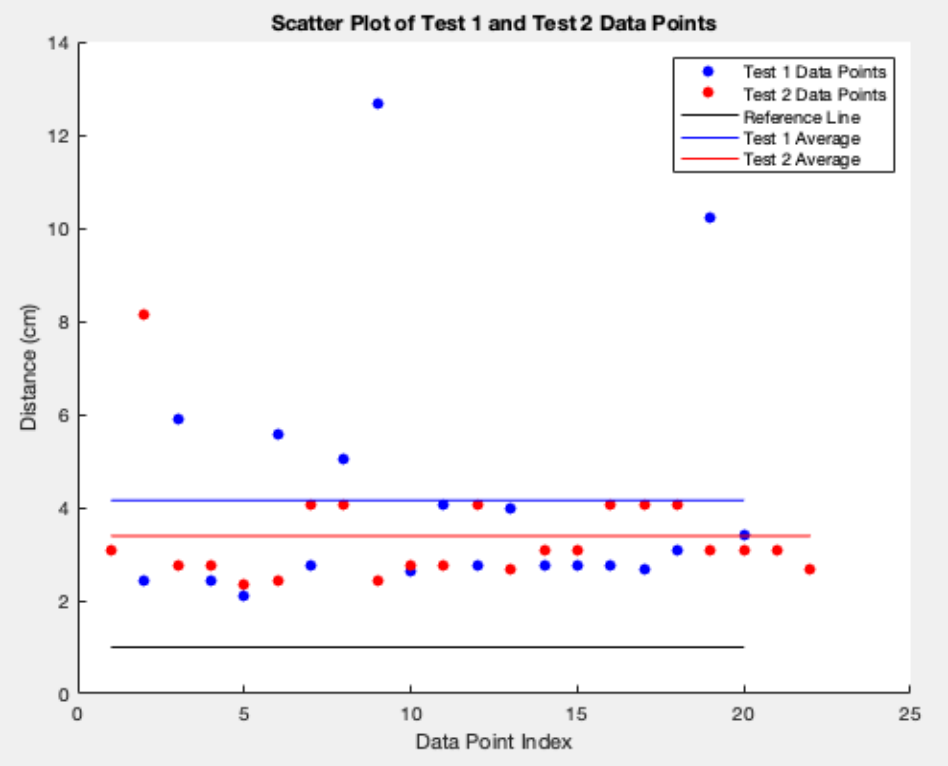
- Testing: Sensor kept above the ground to avoid interference and in open wide room
- Measured distances from sensors ranging from 1cm to 500cm
- Data collected used to test measurement capabilities



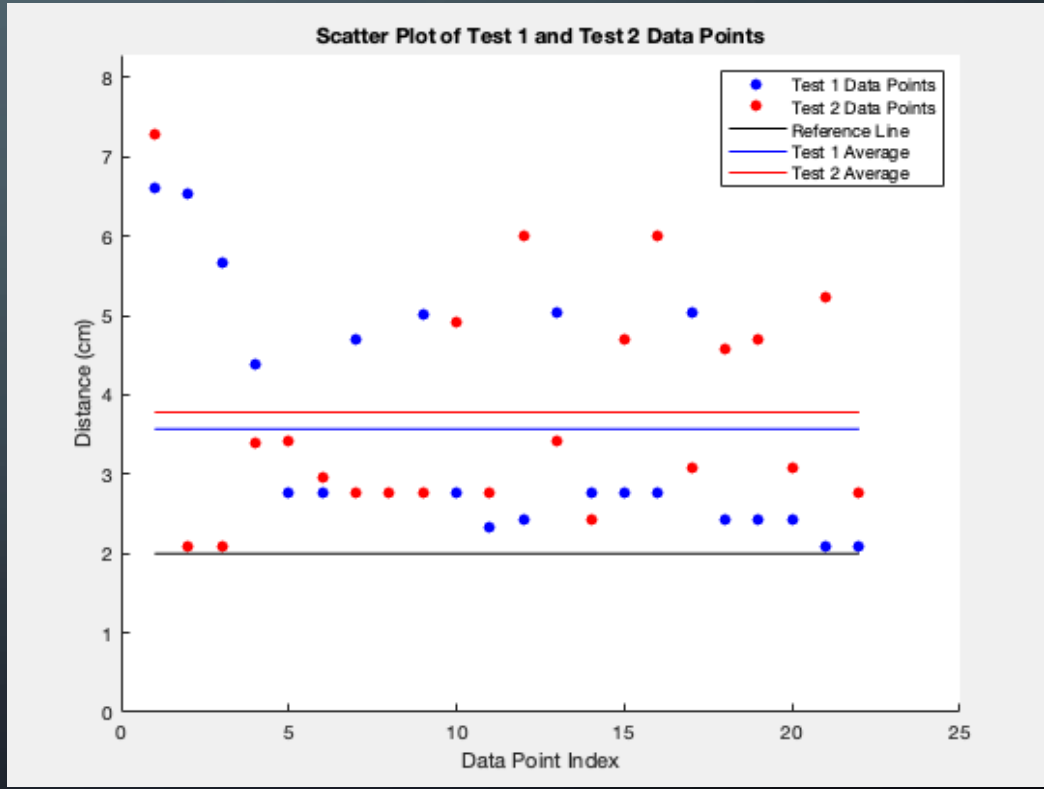
ULTRASONIC SENSOR

- Test Data and Plots

1cm

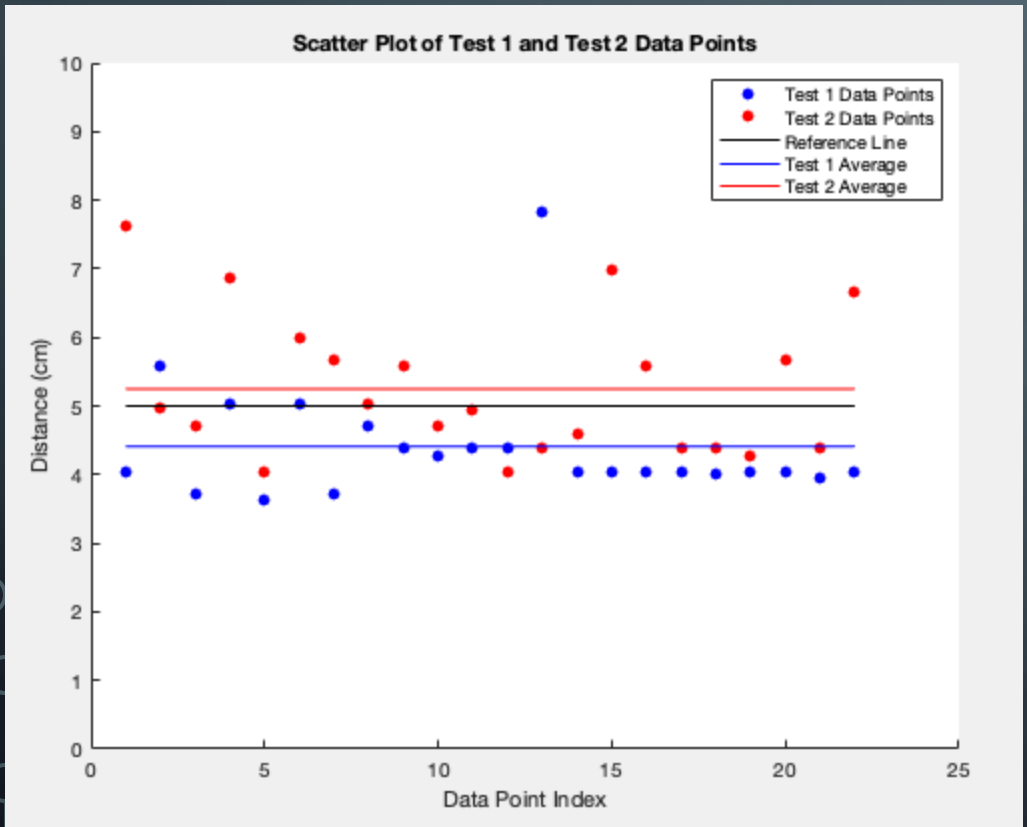


2cm

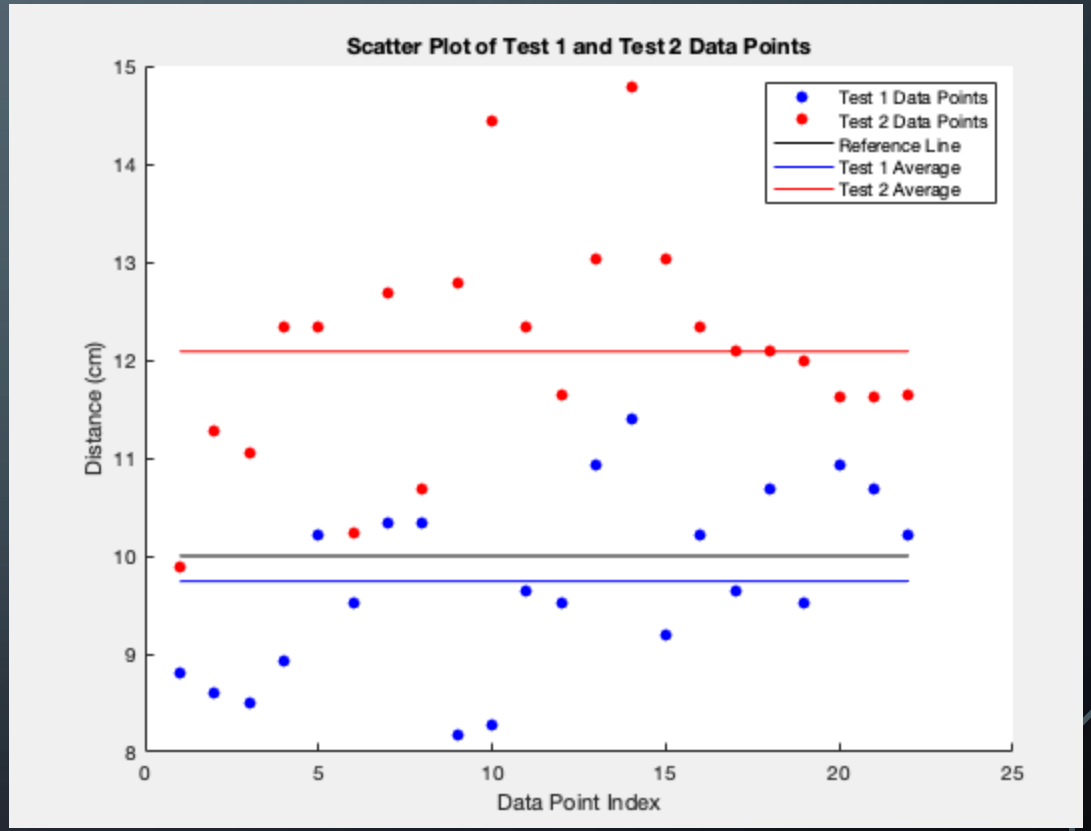


ULTRASONIC SENSOR TESTS CONT.

5cm

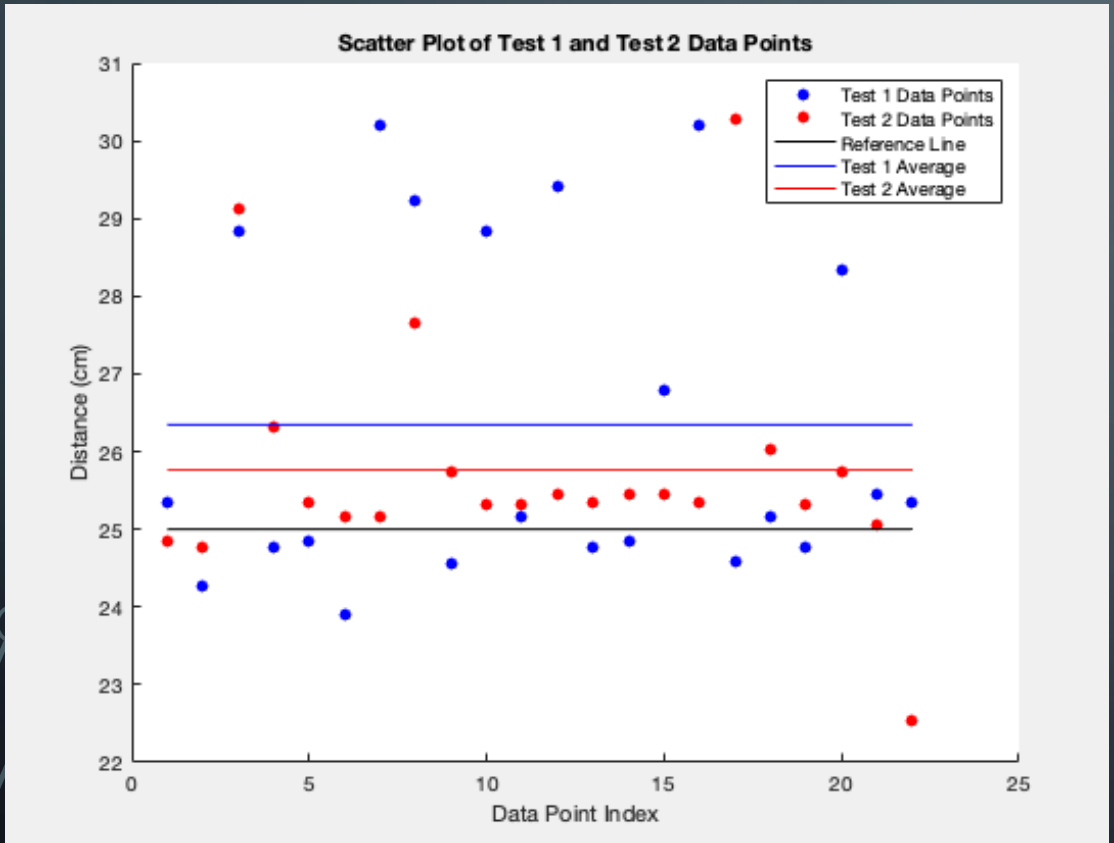


10cm

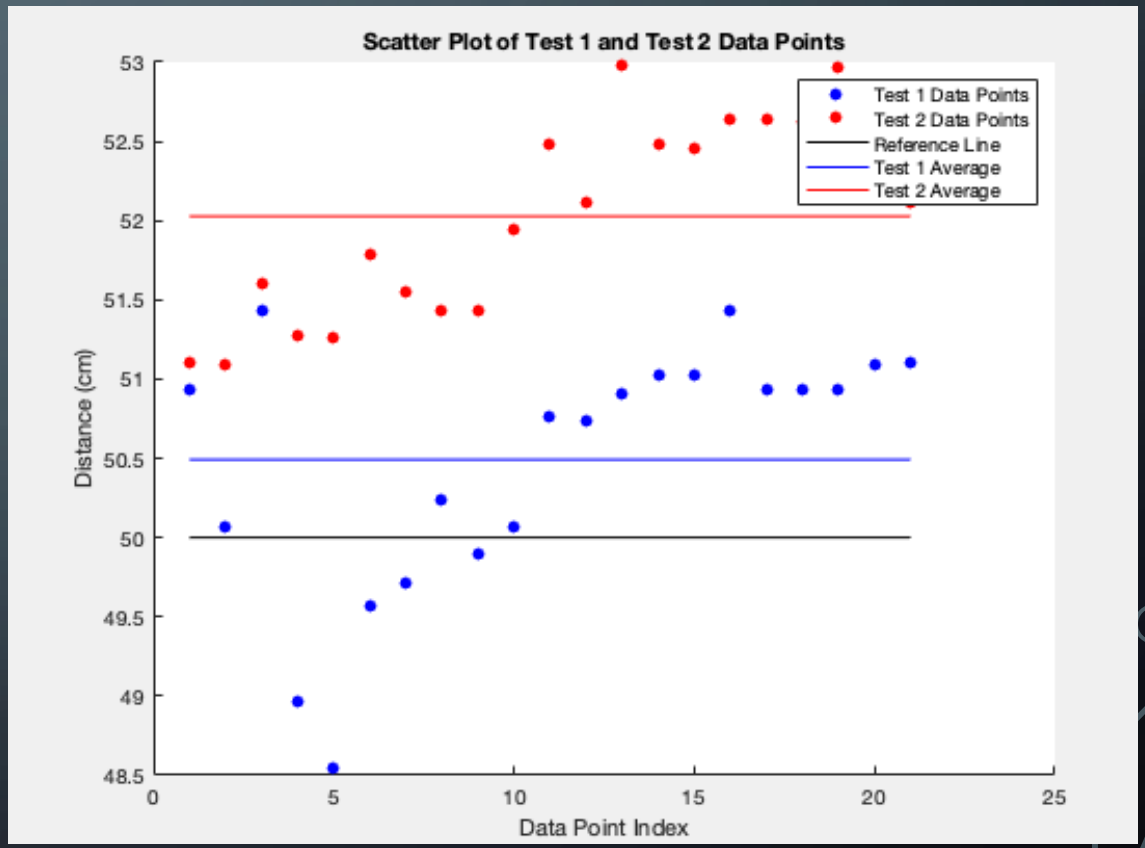


ULTRASONIC SENSOR

25cm

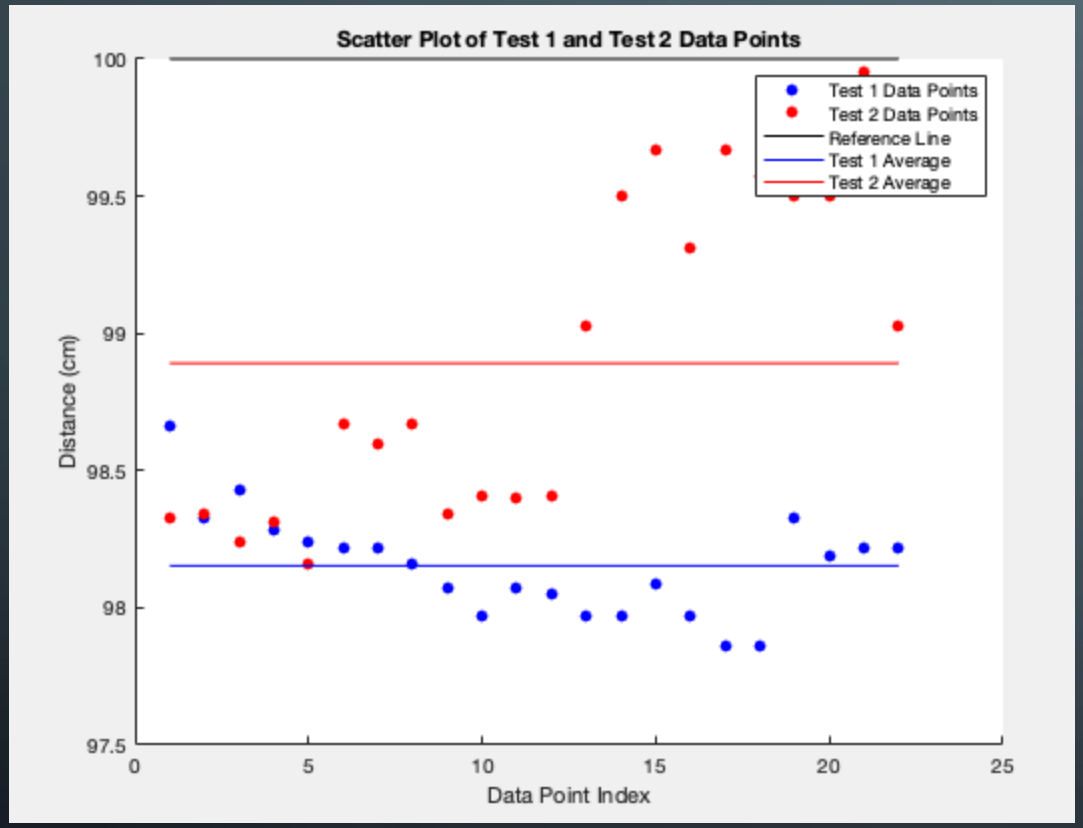


50cm

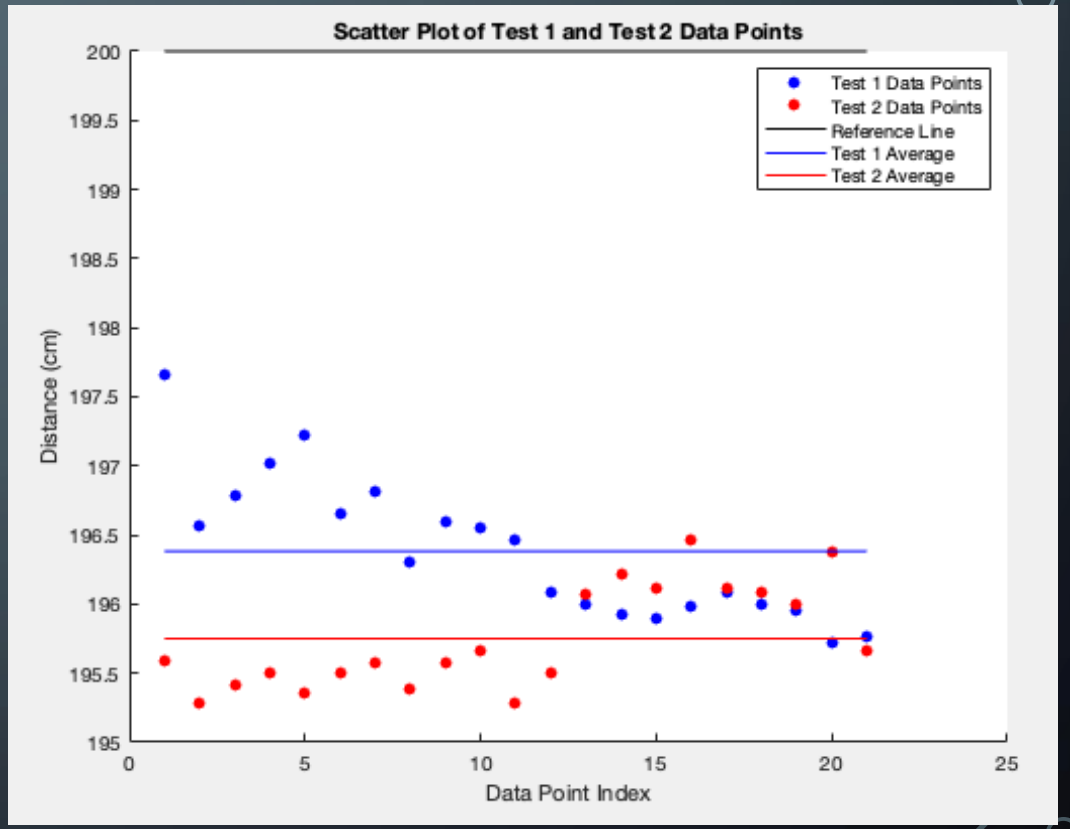


ULTRASONIC SENSOR

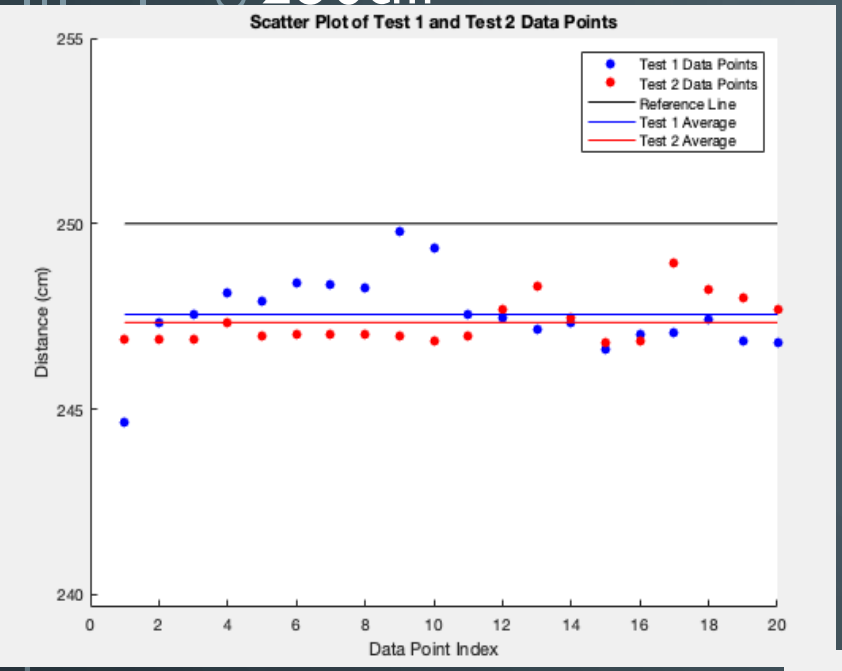
100cm



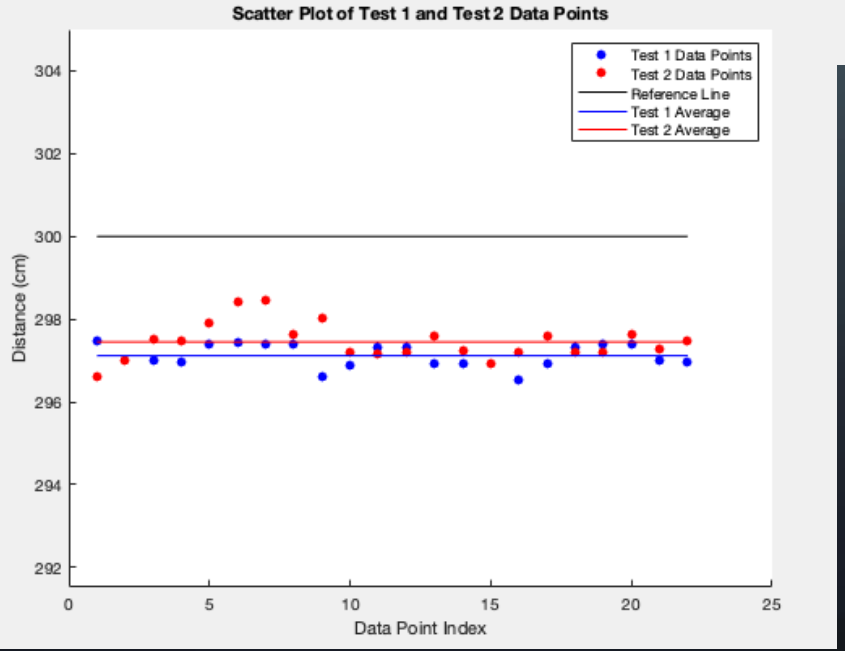
200cm



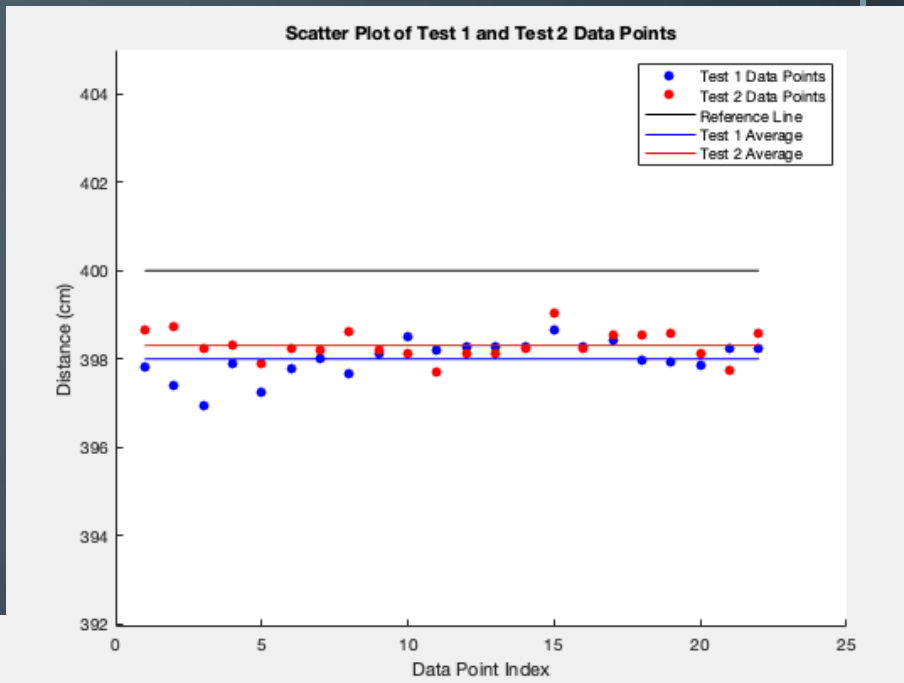
250cm



300cm



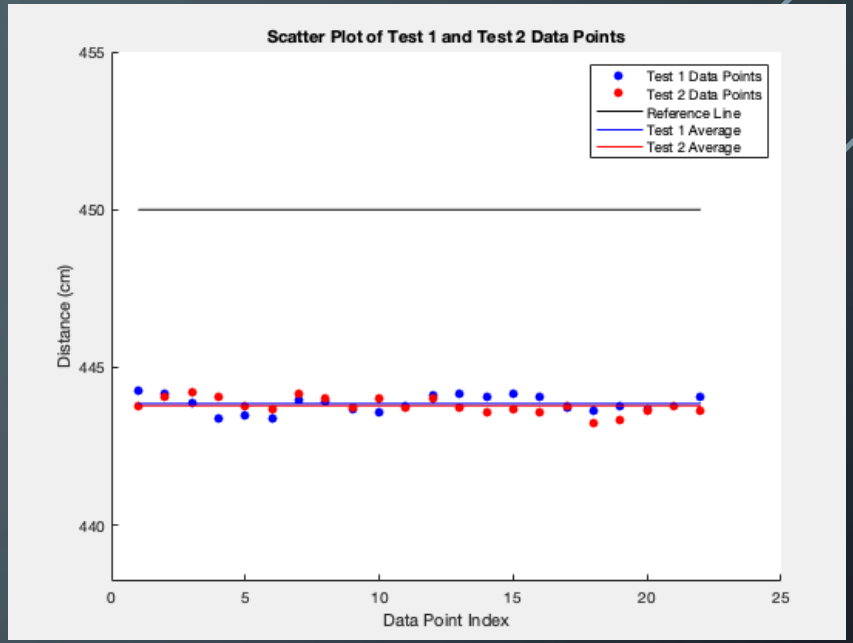
400cm



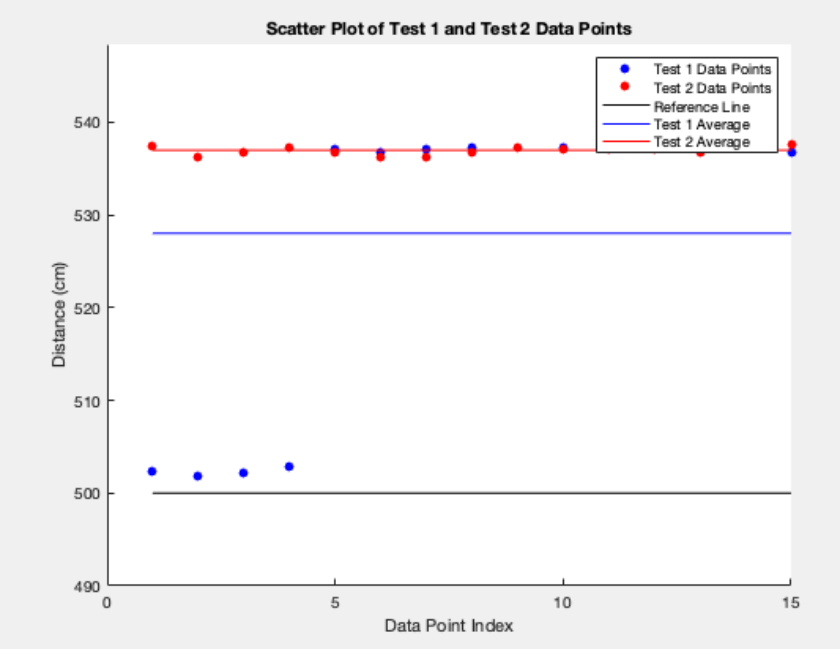
ULTRASONIC SENSOR

- Range
  - if not in proper position, sensor would read 1200cm
  - 25cm to 450cm
  - Data points were too inaccurate before 25cm and weakens begins at 450cm
- Repeatability
  - Test was conducted twice to measure repeatability
  - Within the range, average measurements between the two tests are close, therefore the tests have good repeatability

450cm



500cm



ULTRASONIC SENSOR

- Accuracy and Precision

  1cm-250cm are not accurate or precise

  300cm-400cm are somewhat accurate and precise

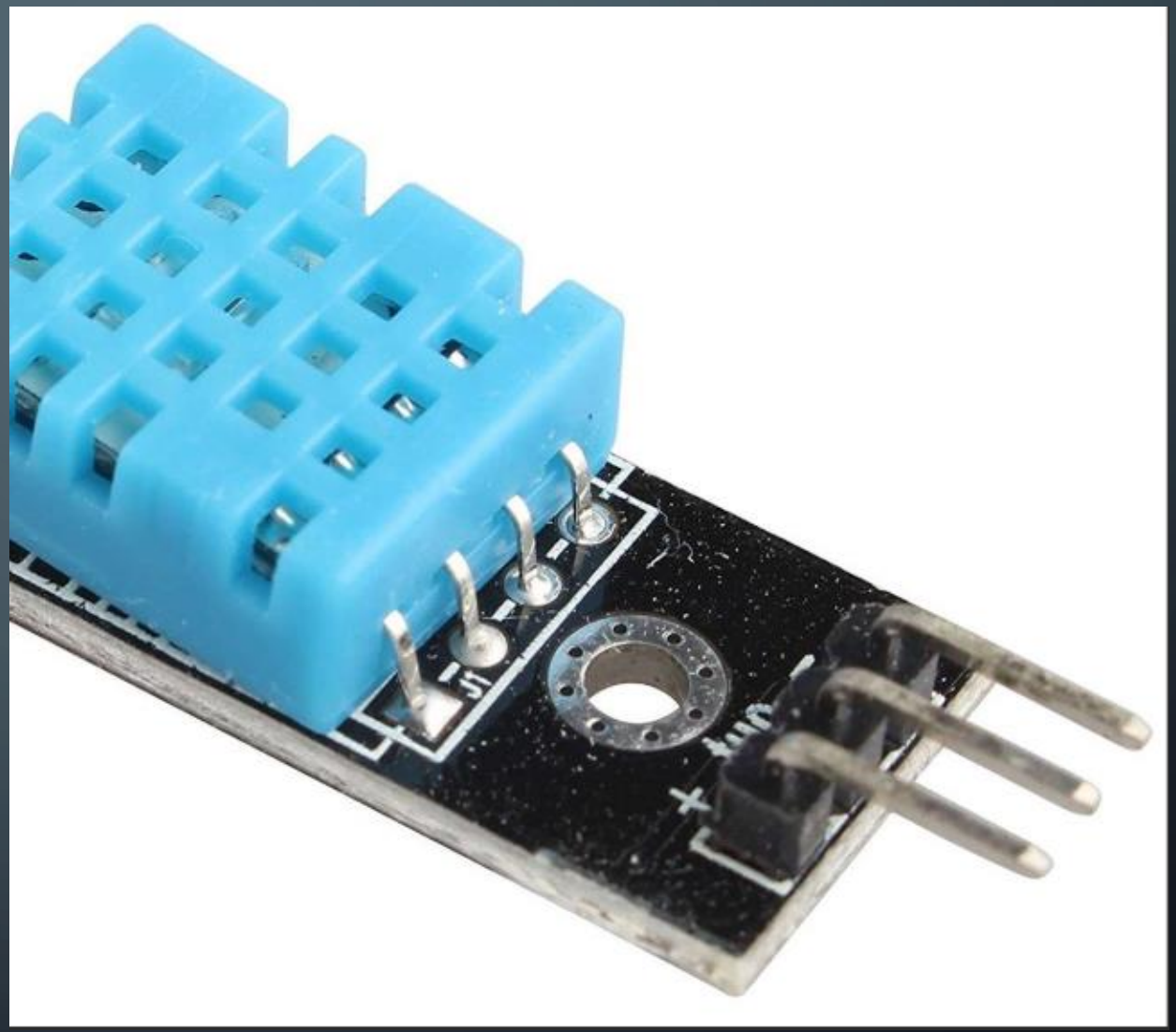
  450cm-500cm are precise but not accurate,

- Static Error

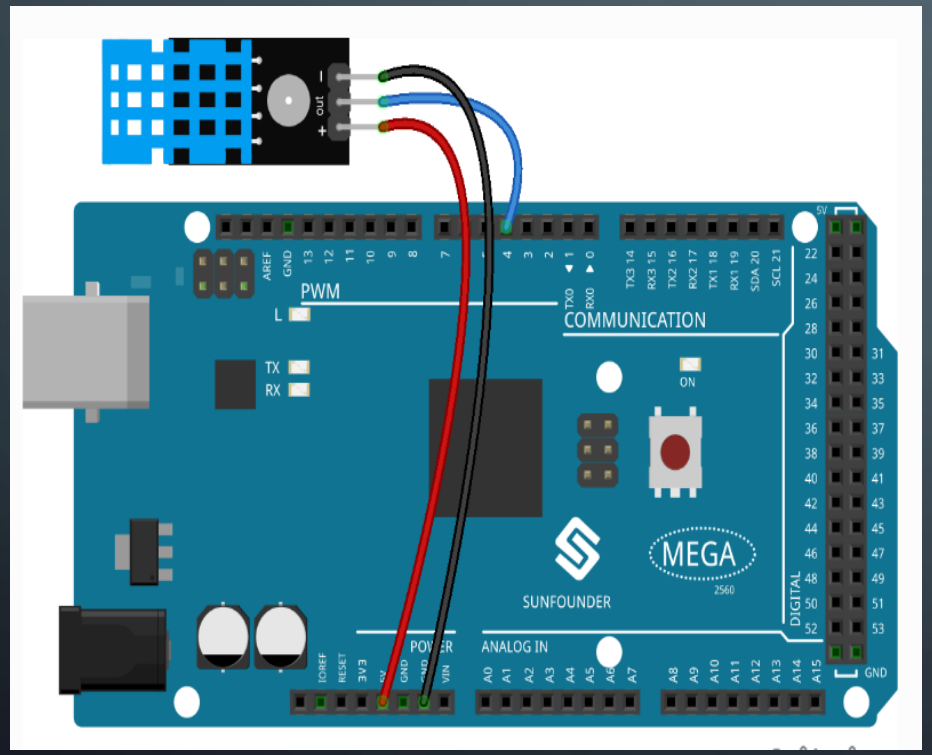
  No Consistent Static error was found between distances

DHT-11

The DHT-11 sensor measures temperature and humidity, operating between 3V to 5.5V. Temperature range: 0°C to 50°C, humidity: 20% to 90%. Accuracy: ±1°C for temperature and ±1% for humidity.



CONNECTIONS AND CODE



sketch_feb18a.ino

```ino
#include "DHT.h"

#define DHTPIN 4  // Set the pin connected to the DHT11 data pin
#define DHTTYPE DHT11 // DHT 11

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  Serial.println("DHT11 test!");
  dht.begin();
}

void loop() {
  // Wait a few seconds between measurements.
  delay(2000);

  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (it's a very slow sensor)
  float humidity = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float temperature = dht.readTemperature();

  // Check if any reads failed and exit early (to try again).
  if (isnan(humidity) || isnan(temperature)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  // Print the humidity and temperature
  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.println(" *C");
}
```

MEASURE

RANGE	•I chose to use boiling water, a desiccant, and ice for my range measurements
ACCURACY	• a temperature and humidity sensor was used for reference
STATIC ERRORS	• Measure data using Matlab

MEASUREMENT METHODS

Range

- I put the sensor over boiling water to test the maximum value of its humidity and temperature.
- I placed the sensor on an ice cube to test it for the lowest temperature measurement
- I put the sensor in a desiccant to test its lowest humidity level

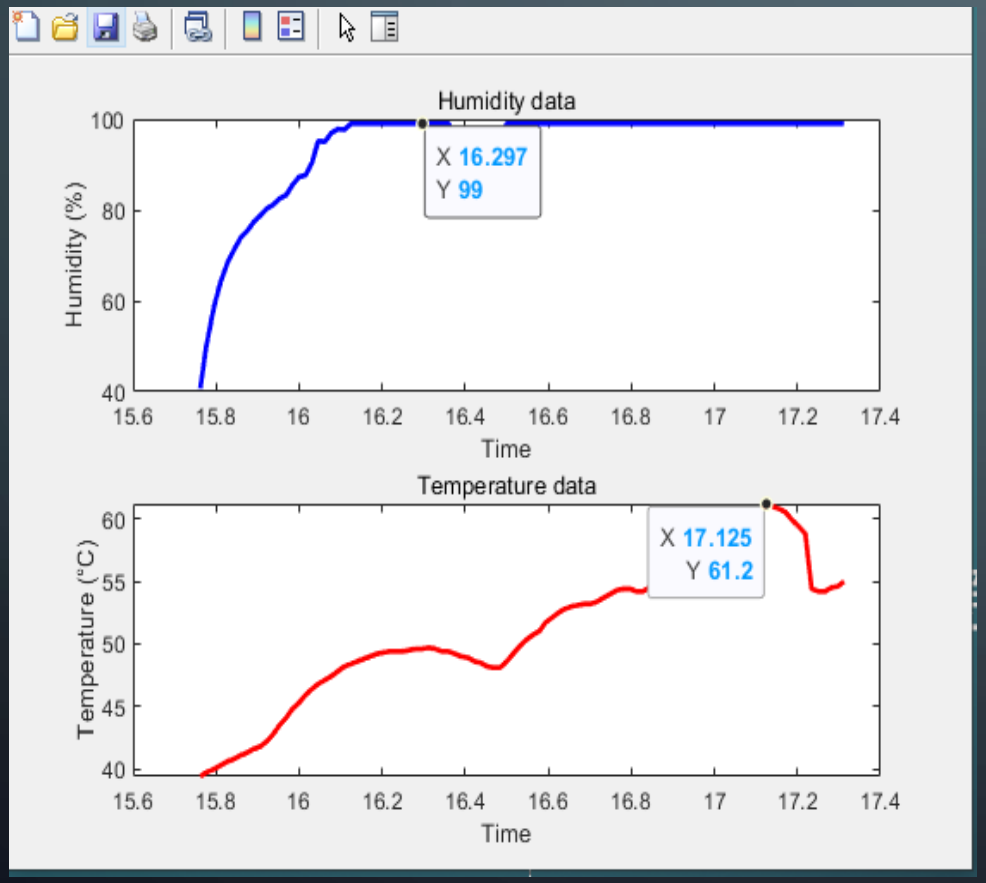
Accuracy

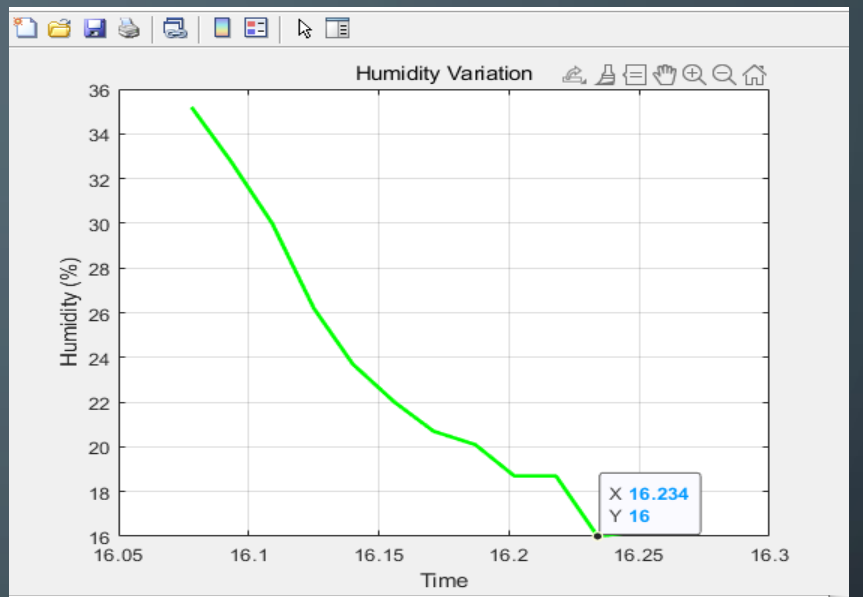
- I made three comparisons, all indoors, using another temperature and humidity sensor to compare the accuracy of the DHT-11 measurement values.

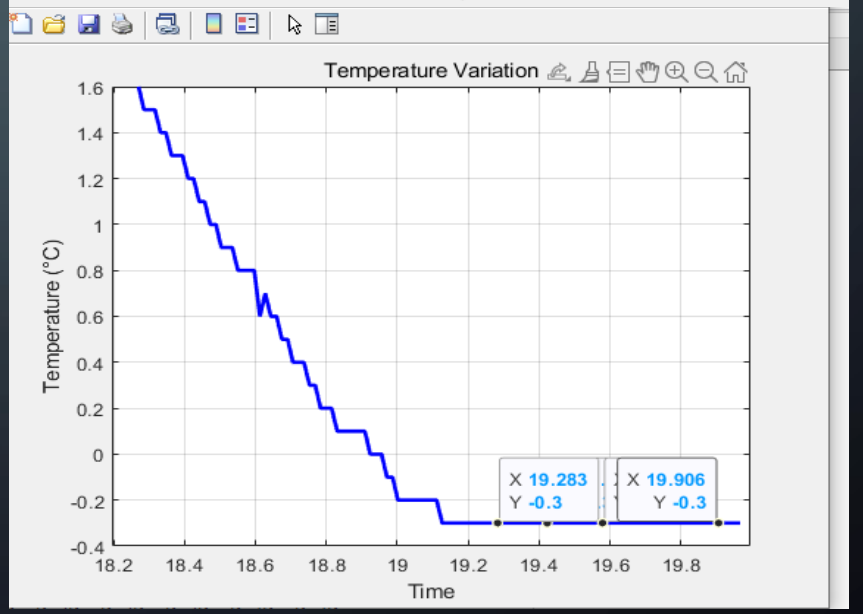
Static Errors

- I tested the data of DHT-11 under static conditions and used matlab to calculate the error

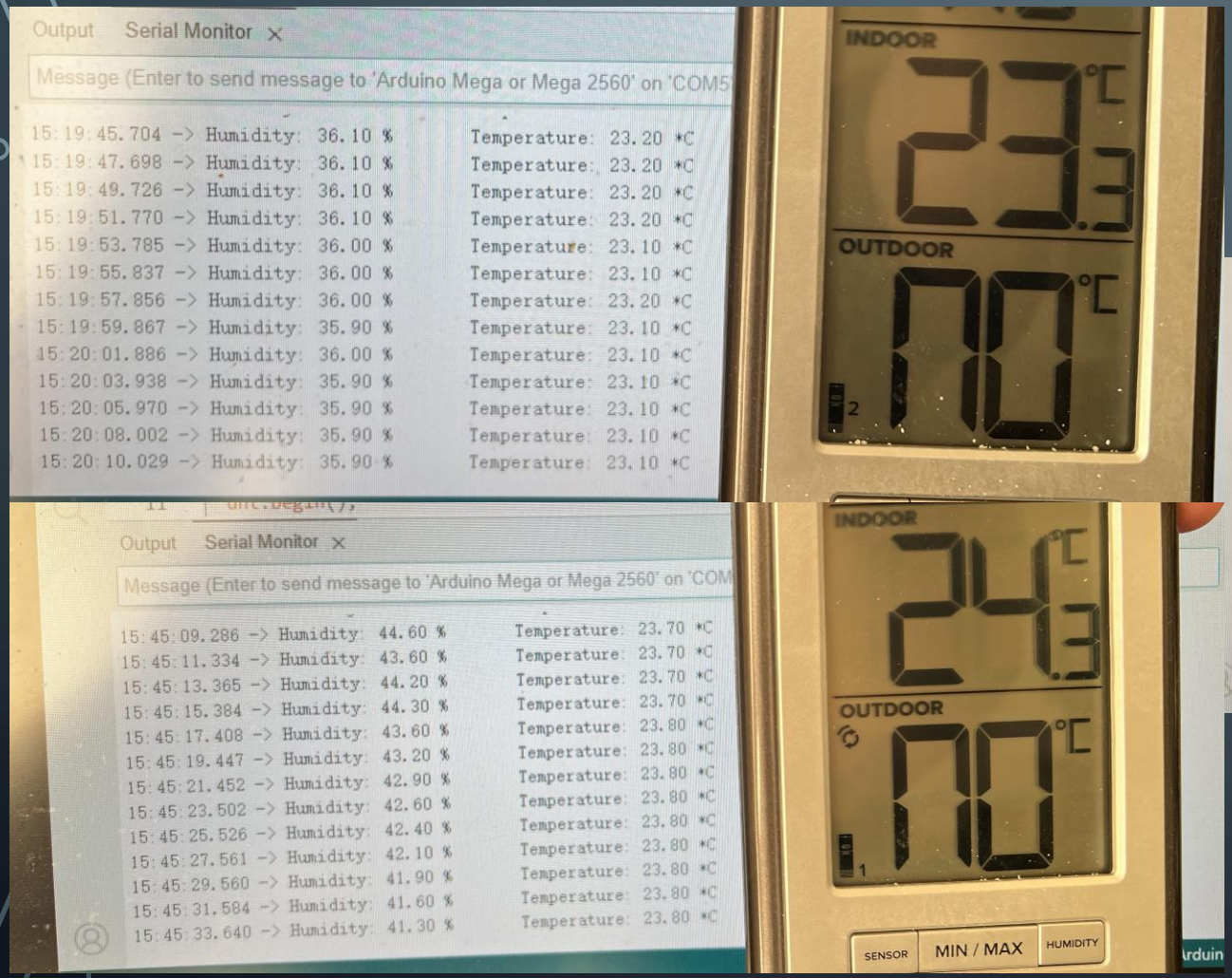
RANGE DATA

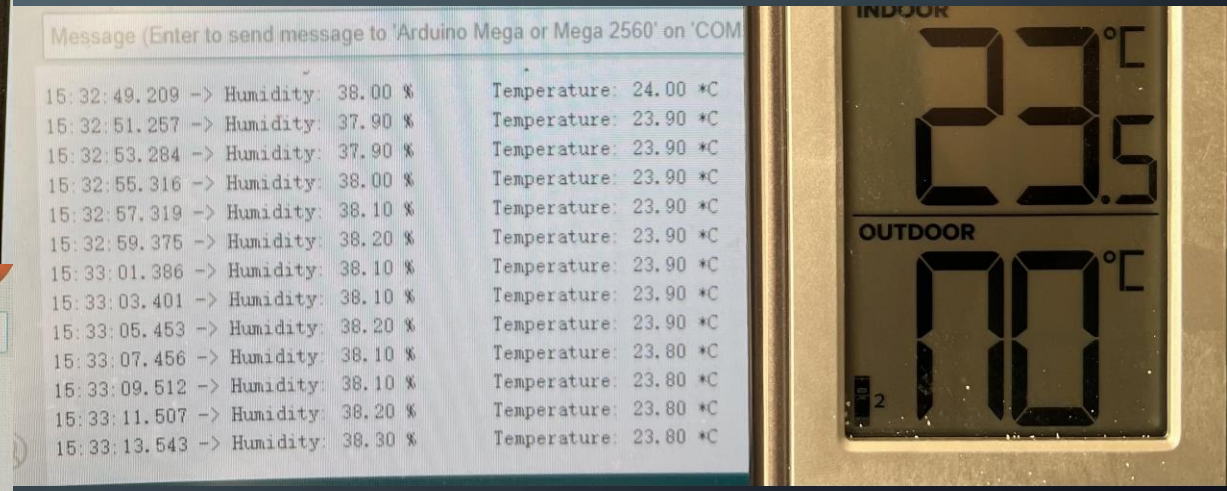

Humidity data
X 16.297
Y 99
Humidity (%)
Time
Temperature data
X 17.125
Y 61.2
Temperature (°C)
Time


Humidity Variation
X 16.234
Y 16
Humidity (%)
Time

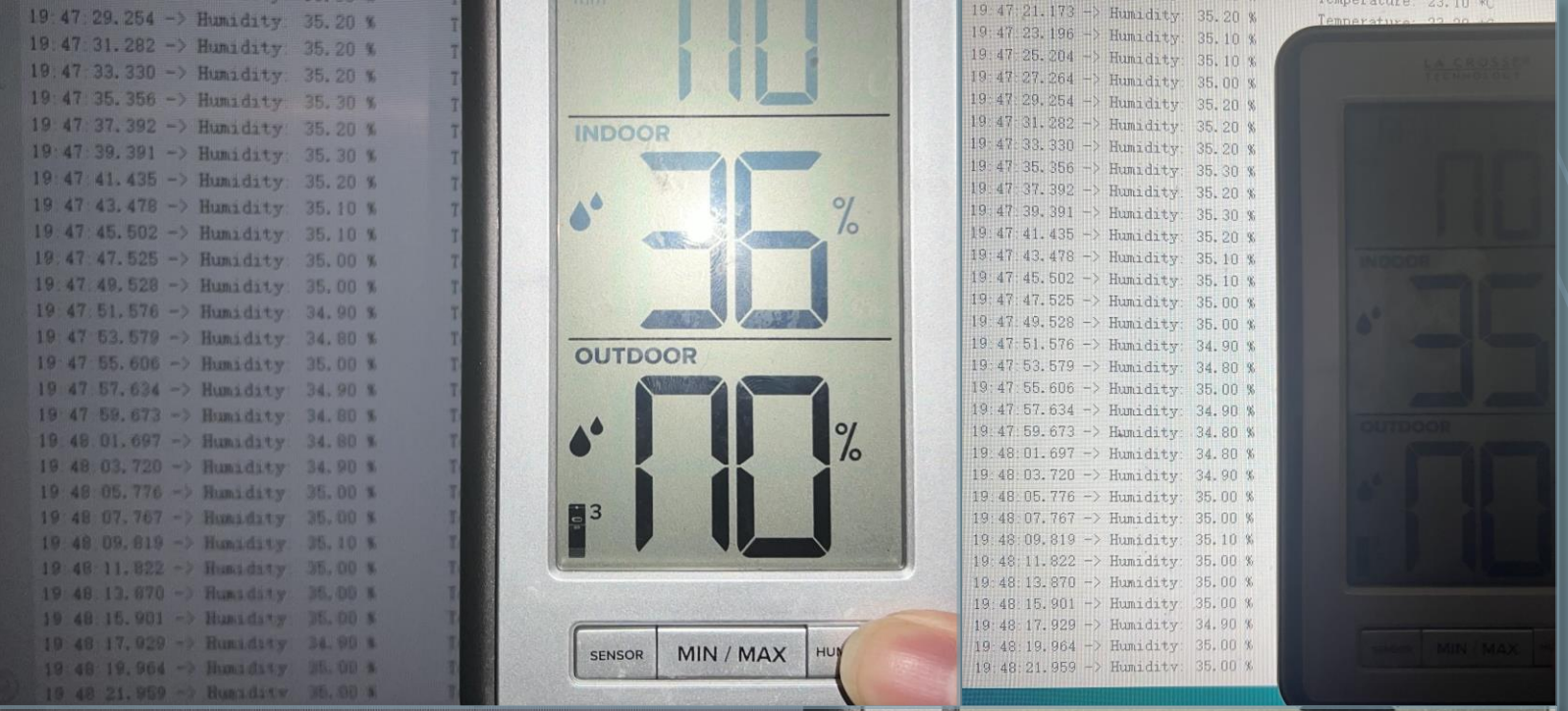

Temperature Variation
X 19.283
Y -0.3
X 19.906
Y -0.3
Temperature (°C)
Time

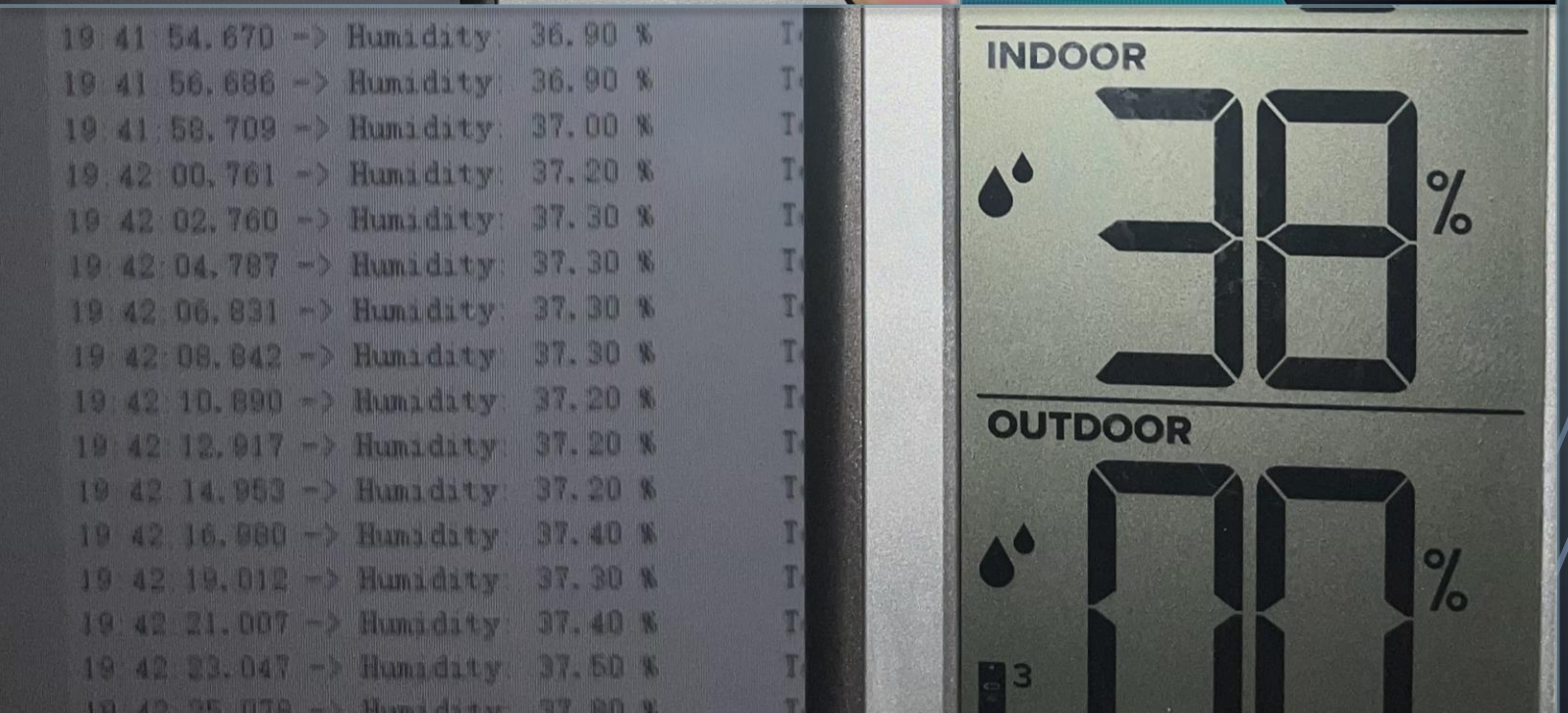
ACCURACY TEMPERATURE





I tested it three times with half an hour's interval. The first time the error was 0.2, the second time the error was 0.5, and the third time the error was 0.3.

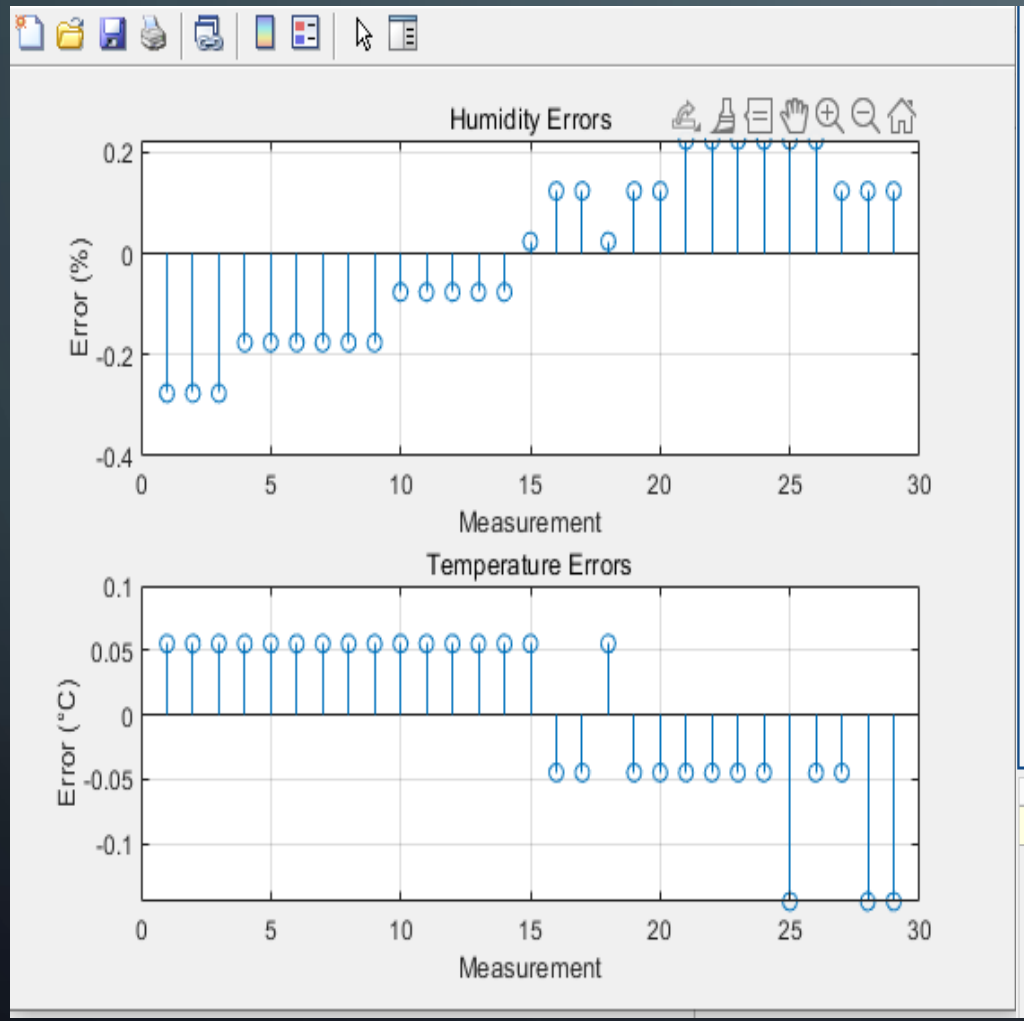




ACCURACY HUMIDITY

- The same test was conducted three times with an interval of 30 minutes each time, and the error was within 1%

STATIC ERRORS




```
humidity = [
    35.00 35.00 35.00 35.10 35.10 35.10 35.10 35.10 35.10 35.20 ...
    35.20 35.20 35.20 35.20 35.30 35.40 35.40 35.30 35.40 35.40 ...
    35.50 35.50 35.50 35.50 35.50 35.50 35.40 35.40 35.40
];

temperature = [
    23.80 23.80 23.80 23.80 23.80 23.80 23.80 23.80 23.80 23.80 ...
    23.80 23.80 23.80 23.80 23.80 23.70 23.70 23.80 23.70 23.70 ...
    23.70 23.70 23.70 23.70 23.60 23.70 23.70 23.60 23.60
];

% Calculate Mean Values
mean_humidity = mean(humidity);
mean_temperature = mean(temperature);

% Calculate Errors
humidity_errors = humidity - mean_humidity;
temperature_errors = temperature - mean_temperature;

% Calculate Standard Deviation
humidity_std_dev = std(humidity_errors);
temperature_std_dev = std(temperature_errors);

% Display Results
disp(['Mean Humidity: ', num2str(mean_humidity)]);
disp(['Mean Temperature: ', num2str(mean_temperature)]);
disp(['Humidity Standard Deviation: ', num2str(humidity_std_dev)]);
disp(['Temperature Standard Deviation: ', num2str(temperature_std_dev)]);
```

命令行窗口

不熟悉 MATLAB? 请参阅有关快速入门的资源。

```
>> untitled
>> untitled
Mean Humidity: 35.2759
Mean Temperature: 23.7448
Humidity Standard Deviation: 0.17455
Temperature Standard Deviation: 0.068589
```

QUESTIONS?