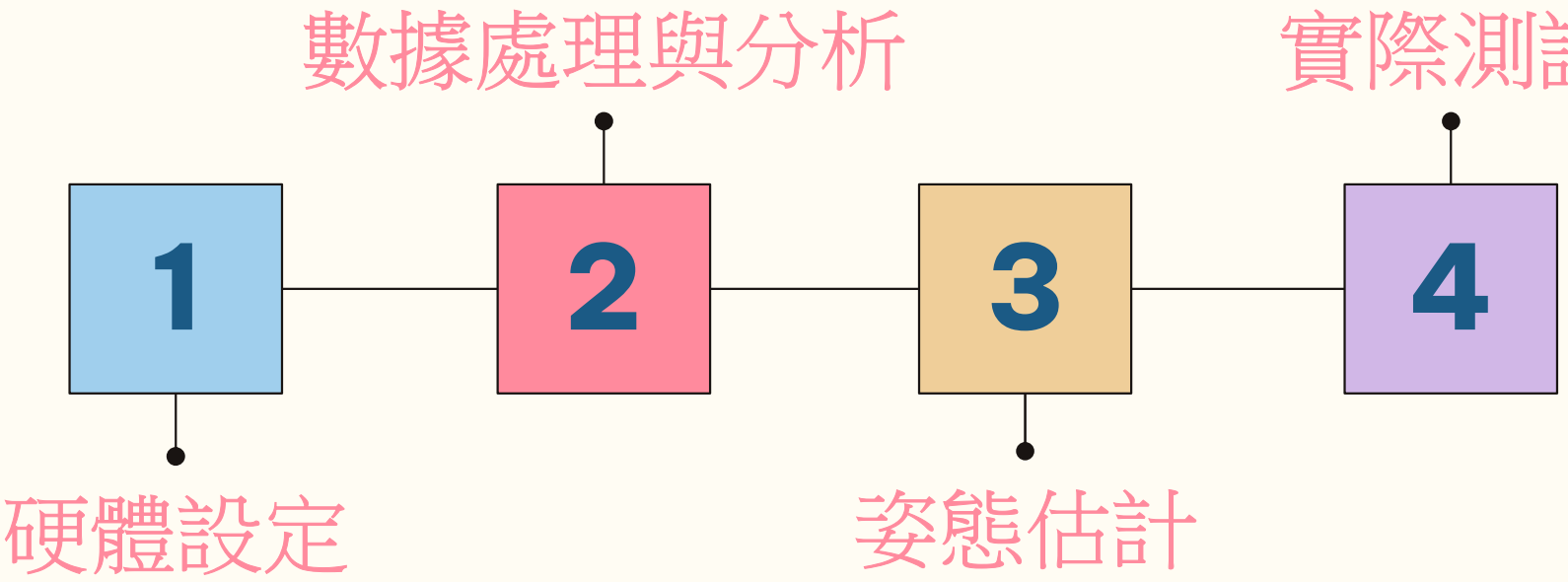
實習專題報告

新生兒髖關節超音波檢測

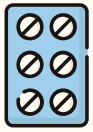
許博森 黃子庭

系統介紹

研究流程



系統概括



感測器採樣

以Arduino UNO 控制
MPU6050六軸感測器
測量繞軸旋轉的變化量

數據處理

將採集到的數據進行前置處理
並運算旋轉情形

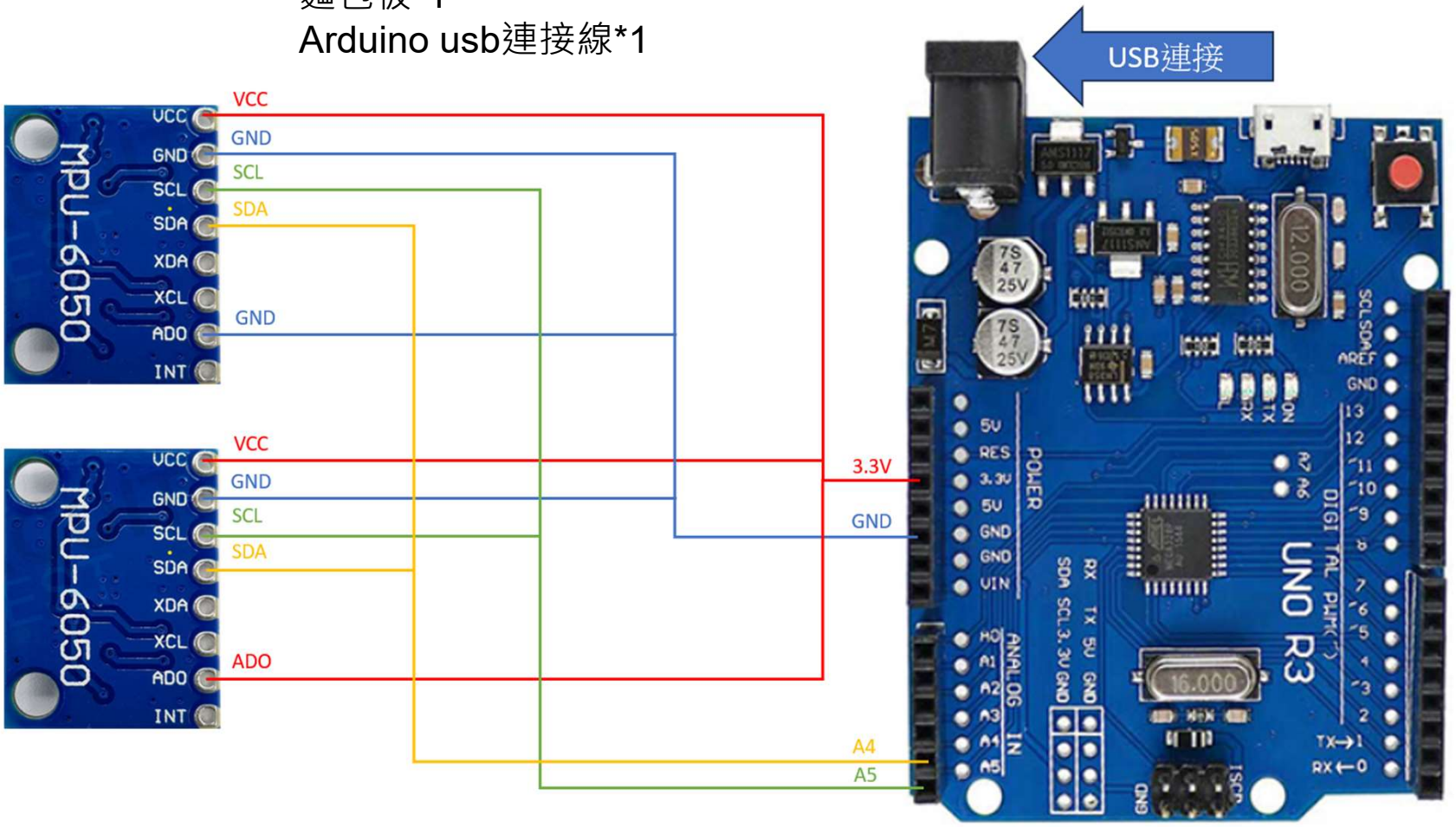
顯示姿態估計

匯出姿態估計結果
並顯示兩感測器的相對狀態

硬體設定

硬體設備

Arduino UNO R3板*1
MPU6050加速度陀螺儀傳感器*2
跳線數條
麵包板*1
Arduino usb連接線*1



硬體設定

```
#include <Wire.h>
#include <MPU6050.h>

MPU6050 mpu1;
MPU6050 mpu2;

// Timers
unsigned long timer = 0;
float timeStep = 0.01;

// Pitch, Roll and Yaw values
float pitch1 = 0, roll1 = 0, yaw1 = 0;
float pitch2 = 0, roll2 = 0, yaw2 = 0;
```

```
void setup()
{
  Serial.begin(115200);

  // Initialize MPU6050 mpu1
  while(!mpu1.begin(MPU6050_SCALE_2000DPS, MPU6050_RANGE_2G, 0x68))
  {
    Serial.println("Could not find a valid MPU6050 sensor1, check wiring!");
    delay(500);
  }
  mpu1.calibrateGyro();
  mpu1.setThreshold(3);

  // Initialize MPU6050 mpu2
  while(!mpu2.begin(MPU6050_SCALE_2000DPS, MPU6050_RANGE_2G, 0x69))
  {
    Serial.println("Could not find a valid MPU6050 sensor1, check wiring!");
    delay(500);
  }
  mpu2.calibrateGyro();
  mpu2.setThreshold(3);

  Serial.println("start computing!");
}
```

硬體設定

```
void loop()
{
  timer = millis();

  Vector norm1 = mpu1.readNormalizeGyro();
  pitch1 = pitch1 + norm1.YAxis * timeStep;
  roll1 = roll1 + norm1.XAxis * timeStep;
  yaw1 = yaw1 + norm1.ZAxis * timeStep;

  Vector norm2 = mpu2.readNormalizeGyro();
  pitch2 = pitch2 + norm2.YAxis * timeStep;
  roll2 = roll2 + norm2.XAxis * timeStep;
  yaw2 = yaw2 + norm2.ZAxis * timeStep;

  Serial.print("(");
  Serial.print(pitch1);
  Serial.print(",");
  Serial.print(roll1);
  Serial.print(",");
  Serial.print(yaw1);
  Serial.print(")");
  Serial.print("(");
  Serial.print(pitch2);
  Serial.print(",");
  Serial.print(roll2);
  Serial.print(",");
  Serial.print(yaw2);
  Serial.println(")");

  // Wait to full timeStep period
  delay(1);
}
```

在還沒有完成四元數字之前，先使用歐拉角計算
(會友萬象死鎖問題)

(pitch,roll,yaw) (pitch,roll,yaw)
Sensor1 Sensor2

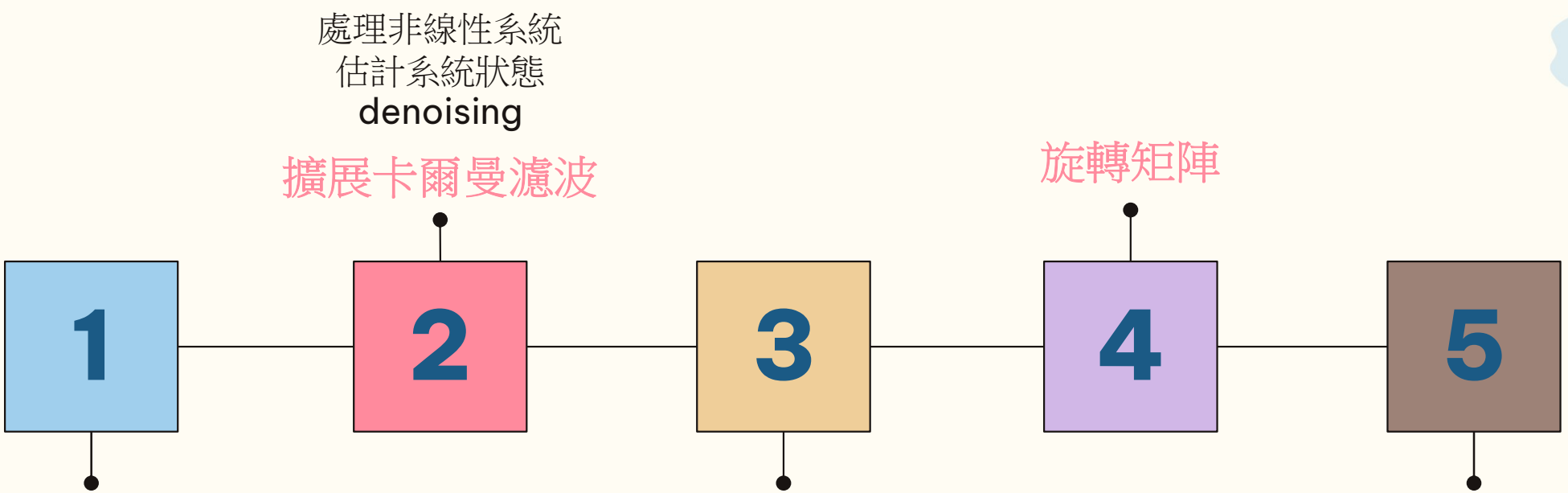
數據處理與分析

數據處理

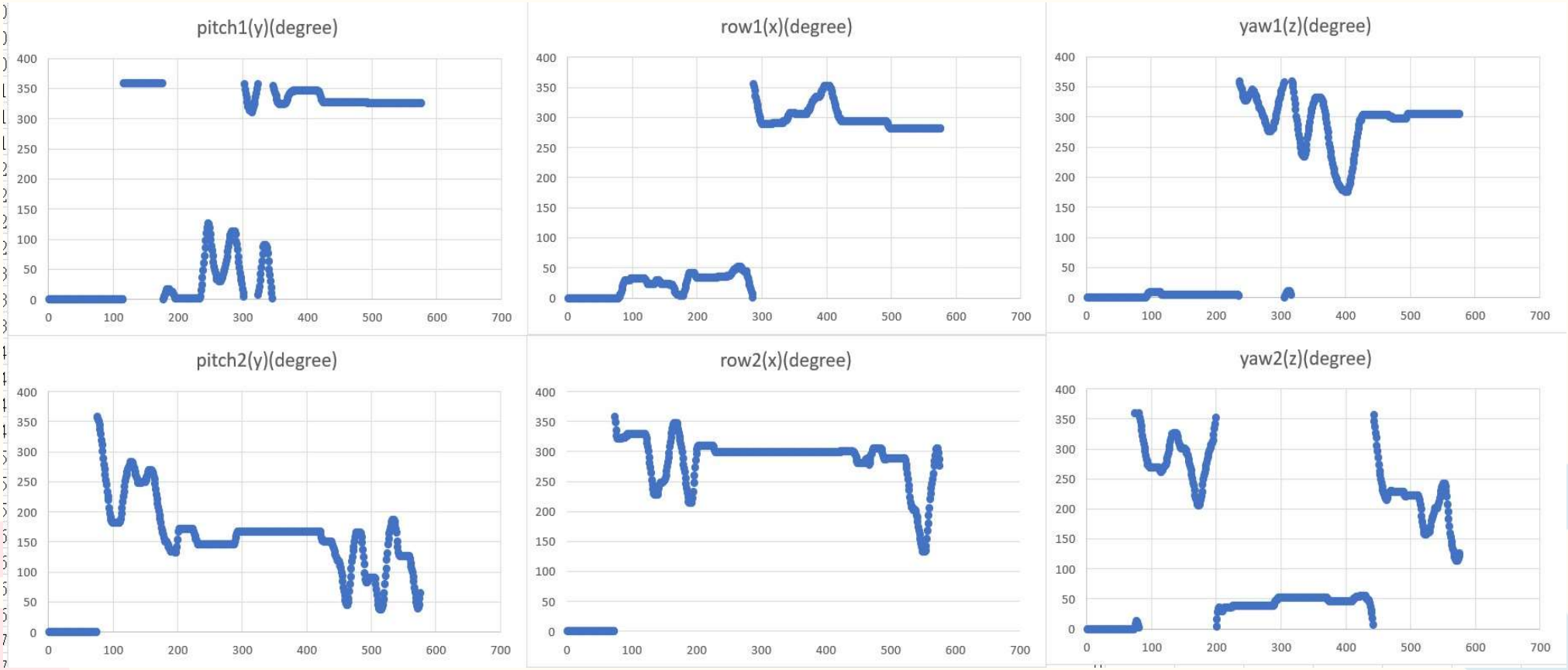
Online

檢測同時顯示兩感測器相對姿態結果
後續規劃匯出csv數據
以利與超音波截圖做比對

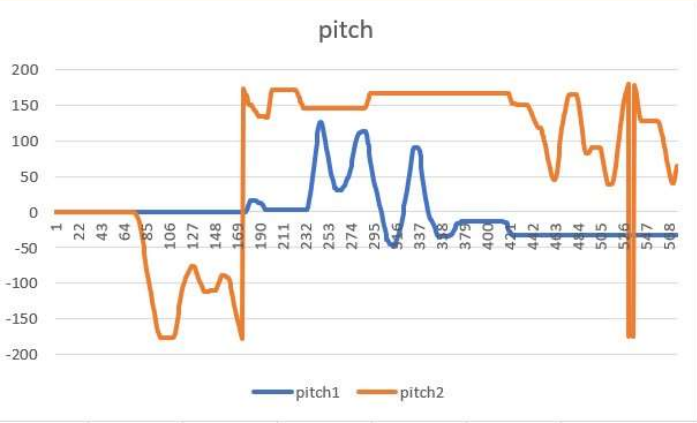
處理程序


處理非線性系統
估計系統狀態
denoising
擴展卡爾曼濾波
旋轉矩陣
1
2
3
4
5
取得陀螺儀數據
pitch、roll、yaw
(radians)
四元數
代表繞軸旋轉量
方便計算且避免萬向鎖
顯示影像
套用到三軸上的單位
向量以顯示影像

數據分析


pitch1(y)(degree)
row1(x)(degree)
yaw1(z)(degree)
pitch2(y)(degree)
row2(x)(degree)
yaw2(z)(degree)

數據分析


pitch
200
150
100
50
0
-50
-100
-150
-200
1 22 43 64 85 106 127 148 169 190 211 232 253 274 295 316 337 358 379 400 421 442 463 484 505 526 547 568
pitch1
pitch2


row
200
150
100
50
0
-50
-100
-150
-200
1 22 43 64 85 106 127 148 169 190 211 232 253 274 295 316 337 358 379 400 421 442 463 484 505 526 547 568
row1
row2


yaw
200
150
100
50
0
-50
-100
-150
-200
1 22 43 64 85 106 127 148 169 190 211 232 253 274 295 316 337 358 379 400 421 442 463 484 505 526 547 568
yaw1
yaw2

數據分析


相對位移pitch
250
200
150
100
50
0
-50
-100
-150
-200
1 22 43 64 85 106 127 148 169 190 211 232 253 274 295 316 337 358 379 400 421 442 463 484 505 526 547 568

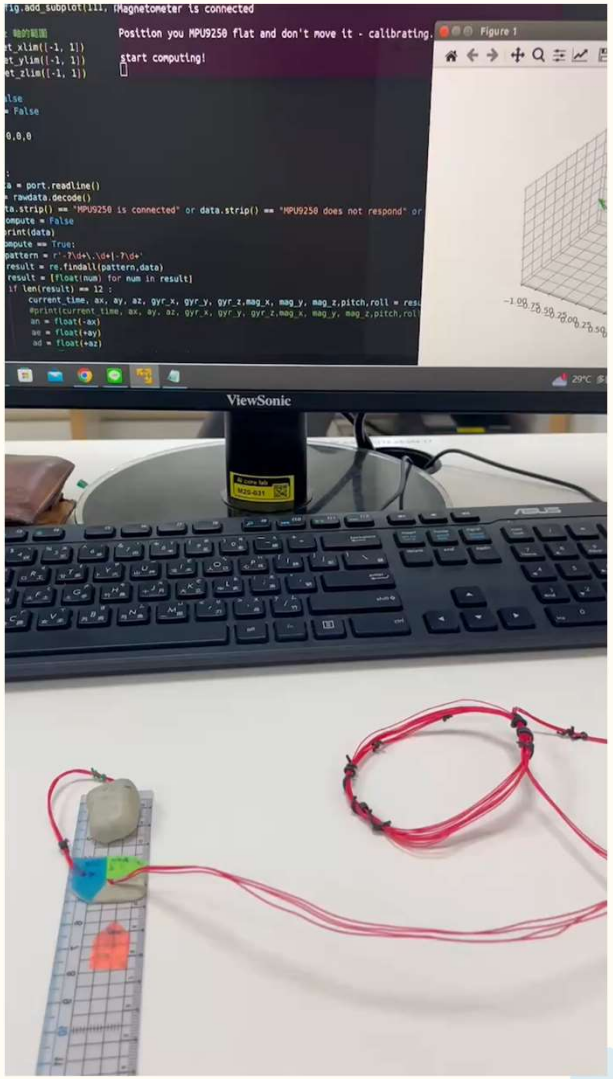

相對位移row
300
250
200
150
100
50
0
-50
-100
-150
-200
-250
1 22 43 64 85 106 127 148 169 190 211 232 253 274 295 316 337 358 379 400 421 442 463 484 505 526 547 568


相對位移yaw
300
250
200
150
100
50
0
-50
-100
-150
-200
1 22 43 64 85 106 127 148 169 190 211 232 253 274 295 316 337 358 379 400 421 442 463 484 505 526 547 568

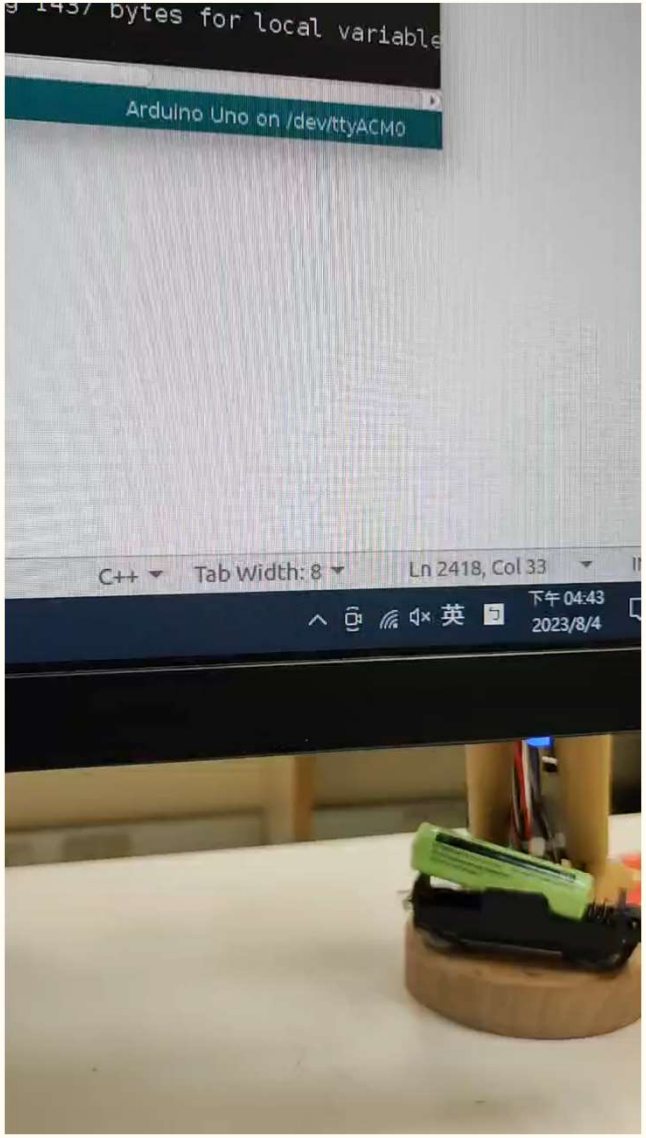
姿態估計

單一感測器

```
roll = rpy[0] ##radian
pitch = rpy[1]
yaw = rpy[2]

last_time = current_time

ax_subplot.cla()  # 清除目前的 figure

# 設定 x, y, z 軸的範圍
ax_subplot.set_xlim([-1, 1])
ax_subplot.set_ylim([-1, 1])
ax_subplot.set_zlim([-1, 1])

xyz = pitch,roll,yaw

# 建立一個旋轉矩陣
r = R.from_euler('xyz', xyz, degrees=True)

# 將旋轉矩陣應用到每個箭頭向量上
rotated_vectors = r.apply(vectors)

# 畫出旋轉後的箭頭
for v, c in zip(rotated_vectors, colors):
    ax_subplot.quiver(0, 0, 0, v[0], v[1], v[2], color=c)

plt.draw()
```



單一感測器


bytes for local variable
Arduino Uno on /dev/ttyACM0
C++
Tab Width: 8
Ln 2418, Col 33
英
下午 04:43
2023/8/4

雙感測器