

逢甲大學學生報告 ePaper

以 M5Stack 感測器實作

PHM 精密量測機械手臂機台

Implementation with M5Stack sensor

PHM Precision Measuring Robotic Arm Machine

作者：余浩璋

系級：精密系統二學位學程

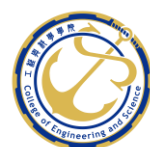
學號：D1057913

開課老師：蔡鈺鼎

課程名稱：精密量測

開課系所：精密系統設計學士學位學程

開課學年：111 學年度 第 1 學期



## 中文摘要

故障預測及健康管理 (Prognostic and Health Management, PHM)，被廣泛地應用在各種專業領域，像是軍事、航太等領域皆常出現，而在工業上常被應用於檢測設備可能故障跡象並確認起因、建立設備老化模型並預測其使用之壽命，了解其健康狀況並對規劃未來方向。

本次實驗以 M5Stack C Plus 感測機械手臂機台之螺桿，量測其螺桿於行進中之穩定度，使用 WI-FI 功能傳輸資料搭配 PuTTY、MATLAB 等軟體進行資料傳輸及運算分析。本實驗對運行中的螺桿來回進行數次精密量測，收集數值並進行視覺化分析，可以預測機器的健康狀況並加以改善，可運用 AI 技術進一步地進行預測，可提高系統的運行可靠性，減少系統的維修費用和提高維修準確性。

**關鍵字：***M5Stack*、*PHM*、感測器、機械手臂、量測



## Abstract

Failure prediction and health management (Prognostic and Health Management, PHM) is widely used in various professional fields, such as military, aerospace and other fields, and is often used in industry to detect possible signs of equipment failure and confirm Causes, establish equipment aging models and predict their service life, understand their health status and plan for future directions.

In this experiment, the M5Stack C Plus is used to sense the screw of the robotic arm machine and measure the stability of the screw during travel. The WI-FI function is used to transmit data and PuTTY, MATLAB and other software are used for data transmission and calculation analysis. In this experiment, several precision measurements are made on the running screw back and forth, and the values are collected and analyzed visually, which can predict the health status of the machine and improve it. AI technology can be used to further predict and improve the operating reliability of the system. Reduce system maintenance costs and improve maintenance accuracy.

*Keyword : M5Stack 、PHM 、Sensor 、Robotic Arm 、Measurement*

## 目錄

---

中文摘要.....	1
Abstract.....	2
目錄.....	3
圖目錄.....	4
附錄目錄.....	5
第一章-研究簡介.....	7
1.1 技術與規格調查.....	7
1.2 文獻回顧.....	7
1.2.1 M5Stack C Plus.....	7
1.2.2 MATLAB.....	8
1.3 方法調查.....	8
1.4 研究目的.....	8
1.5 研究內容.....	9
第二章-方法論述.....	10
2.1 量測方法.....	10
2.2 資料傳輸.....	10
2.3 資料視覺化.....	10
2.4 資料信賴區間.....	11
2.5 資料比較.....	12
第三章-實驗量測與結果討論.....	16
3.1 感測器振動量測.....	16
3.2 結論與未來.....	16
參考文獻.....	17
附錄.....	18
附錄一、量測過程.....	18
附錄二、資料傳輸.....	20
附錄三、相關理論.....	21
附錄四、M5Stack C Plus Arduino Code 環境安裝.....	22
附錄五、資料分析.....	25

## 圖目錄

---

圖 1-1 三次實驗分別的 XYZ Boxplot.....	11
圖 1-2 三次實驗分別的 X Boxplot 比較.....	11
圖 1-3 三次實驗分別的 Y Boxplot 比較.....	11
圖 1-4 三次實驗分別的 Z Boxplot 比較.....	11
圖 2-1 第一次試驗之未轉換數據圖.....	12
圖 2-2 第二次試驗之未轉換數據圖.....	12
圖 2-3 第三次試驗之未轉換數據圖.....	12
圖 3-1 第一次試驗之三角函數轉換數據圖.....	13
圖 3-2 第二次試驗之三角函數轉換數據圖.....	13
圖 3-3 第三次試驗之三角函數轉換數據圖.....	13
圖 4-1 第一次試驗之傅立葉轉換數據圖.....	14
圖 4-2 第二次試驗之傅立葉轉換數據圖.....	14
圖 4-3 第三次試驗之傅立葉轉換數據圖.....	14
圖 5-1 三次實驗之所有 XYZ 資料波形.....	15
圖 5-2 三次實驗之 X 資料波形.....	15
圖 5-3 三次實驗之 Y 資料波形.....	15
圖 5-4 三次實驗之 Z 資料波形.....	15

## 附錄目錄

附錄一、量測過程.....	18
圖 1-1 M5Stack 量測程式.....	18
圖 2-1 M5Stack 量測環境擺設示意圖.....	18
圖 2-2 M5Stack 量測環境擺設示意圖.....	18
圖 2-3 M5Stack 運作中執行量測圖.....	18
圖 2-4 Arduino 預設 Wi-Fi 及全域變數程式碼.....	19
圖 2-5 Arduino void setup 程式碼.....	19
圖 2-6 Arduino void loop 程式碼.....	19
圖 2-7 Arduino void I2C_Write_NBytes 程式碼.....	19
圖 2-8 Arduino void Btn_test 程式碼.....	19
圖 2-9 M5 Stack 待機狀態.....	20
圖 2-10 M5 Stack 已連接 Wi-Fi.....	20
圖 2-11 M5 Stack 開始進行量測.....	20
圖 2-12 M5 Stack 結束量測，顯示總平均結果.....	20
附錄二、資料傳輸.....	20
圖 3-1 PuTTY Configuration 設定參數連接感測器.....	20
圖 3-2 PuTTY 資料傳輸畫面.....	21
圖 3-3 資料彙整為 CSV 檔至指定路徑.....	21
圖 3-4 CSV 檔中之 acceleration 資料.....	21
附錄三、相關理論.....	21
圖 4-1 傅立葉轉換 頻域.....	21
圖 4-2 傅立葉轉換 時域.....	21
附錄四、M5Stack C Plus Arduino Code 環境安裝.....	22
圖 5-1 下載驅動器.....	22
圖 5-2 開啟裝置管理員.....	22
圖 5-3 尋找 COM 序列埠.....	22
圖 5-4 尋找驅動程式.....	22
圖 5-5 使用下載好的驅動程式.....	23
圖 5-6 安裝 Arduino IDE.....	23
圖 5-7 開啟 Preference 貼上網址 <a href="https://dl.espressif.com/dl/package_esp32_index.json">https://dl.espressif.com/dl/package_esp32_index.json</a> ，並關閉.....	23
圖 5-8 開啟工具找到開發板(boards).....	23
圖 5-9 開啟工具找到開發板管理器(boards manager).....	24
圖 5-10 尋找 esp32.....	24
圖 5-11 打開程式庫.....	24
圖 5-12 打開管理庫.....	24

## 以 M5Stack 感測器實作 PHM 精密量測機械手臂機台

圖 5-13 搜尋 M5StackCPlus 並下載.....	25
附錄五、資料分析.....	25
圖 6-1 WAVE 分析之 Python 程式碼執行過程.....	25
圖 6-2 WAVE 分析之 Python 程式碼執行過程.....	25
圖 6-3 WAVE 分析之輸出 csv 檔.....	26
圖 6-4 M5 Stack 結束量測，顯示總平均結果.....	26
圖 6-5 MATLAB 常態分佈下之 95%信賴區間計算程式碼.....	26
圖 6-6 MATLAB 三次測驗之 95%信賴區間計算結果.....	26



# 第一章 研究簡介

## 1.1 技術與規格調查

傳統之 PHM 週期大致分為三大主要步驟，觀察、分析及決策，從數據分析至人機介面，又可細分為七項步驟，這種配置是針對已研究及定義的系統，然而大多數 PHM 工程均並未包含其定義，及 PHM 之演進。中小型企業也存在資料量無法提供運算之問題，因此延長 PHM 週期對於中小型企業十分重要。

工廠內有諸多機台和廠務系統核心運轉設備健康管理設備，如馬達、空壓機、水泵、冰水主機等；藉由安裝震動加速規、溫度貼片、流量計、壓力計、數位電表等感測器，收集運轉設備健康狀態的關鍵數據，並透過人性化儀表板提供遠端監控及數據分析，協助企業主及廠務即時掌控設備效能及維修時程，提升產程應對靈活性及整體生產力。

CBM 以及 PHM 與之前描述的維護策略類型不同，是採用數據驅動技術協助我們有效地設置維護活動之時間，找出維護頻率，與其相對成本之間的平衡點，而兩者的區別在於它們在檢測有缺陷之系統條件時的不同響應。

CBM 在檢測時間過後會立即啟動干預機制，可能需要更換或修理設備的某個組件，即使還可以正常運轉更長的時間，不影響其他部件。從生產效率的角度來看，檢測到故障後立即進行干預，可能導致設備工作週期在不適切的階段中止。

PHM 是某時間預測組件之使用壽命，以瞭解未來需執行維護之時間點，相較之下更可降低維護成本，因為可以不犧牲安全性及效率的充分利用每個組件。

## 1.2 文獻回顧

### 1.2.1 M5Stack C Plus

M5Stack 為 2016 年在深圳成立的公司，其意涵為 Modular、5cm、Stackable。分別為模組化、印刷電路板之長寬僅有 5 公分，猶如積木可堆疊的。核心為 Core 積木，由 ESP32 擔任主控晶片，底部可彈性換裝不同的 Module，如無線通訊模組、電池模組等。Module 能以一個 Core 為基礎進行複數堆疊連接；Core 也可向外連接不同的 Unit(各種感測器或致動器)積木，如加速度感測器、震動馬達等。



以 M5Stack 感測器實作 PHM 精密量測機械手臂機台

除上述外，M5Stack 還提供機械、機構類型配件、底座等，如智慧插座、迷你車等。使用它的好處是開發的人員不需焊接、接線，自由拼湊即可。

### 1.2.2 MATLAB

MATLAB 又稱矩陣實驗室，為美國 The MathWorks 公司出品之商業數學軟體，用於數值運算，有許多附加功能，它也適合不同專業領域之應用，不管自動控制系統設計、影像識別、機器與深度學習、金融建模及分析等。還有 Simulink 提供視覺化開發環境，常用於系統類比、動態及嵌入式系統開發等方面。

## 1.3 方法調查

本實驗將以 M5 Stack C Plus 對機械手臂機台的螺桿進行振動感測的測試，使用 Arduino IDE 撰寫需要應用之程式，並將該程式燒錄至 M5 Stack C Plus 中。本實驗在 M5 Stack C Plus 中寫入 WI-FI 功能，不需連接 Serial 即可透過 PuTTY 連接 IP 位址即可傳輸資料至 csv 檔或是 SQL 等資料庫，即可將該資料進一步地使用 MATLAB 或是 Python 分析，可將資料視覺化並結合視覺平台進一步地將資料可視化。實驗收集之數據也可進一步進行 AI 分析預測機器之健康管理狀態。

## 1.4 研究目的

本文主要以 M5 Stack C Plus 對機械手臂機台的螺桿進行 PHM (Prognostic and Health Management, PHM) 故障預測與健康管理分析，藉由感測器所接收的振動資料所繪製的曲線圖及其延伸進行分析，本實驗希望能以 M5 Stack C Plus 結合 MATLAB 進行運算，並分析數據達到對於機器設備的健康管理，進行精密的實作與量測，給予施加優化

## 1.5 研究內容

第二章為方法論述，首先是對於本實驗使用 M5 Stack C Plus 進行實際量測分析時的環境、目標物及量測情況，再進一步地介紹所使用的數值方法及測試方式，並進行趨勢擬合。

第三章為實驗結果與討論，對於使用本感測器所實驗的結果進行探討，及可能的優化方案，進行兩個以上案例的比較，進一步地進行改善。

第四章為結論與未來，總結本實驗的所有內容及要點，為前幾章之結果與比較，作為本文的結論。

## 第二章 方法論述

### 2.1 量測方法

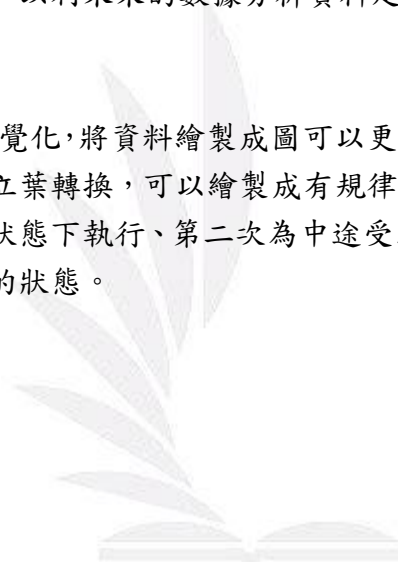
本次量測將 M5 Stack C Plus 透過黏土黏在機械手臂機台的螺桿上，使用先前用 Arduino IDE 寫好的程式燒錄進去以提供量測，將移載平台復歸原位，以復歸位置作為起始點，待感測器啟動時，移載平台開始由手動操控移動，來回一次，本實驗進行三次實驗，第一次為正常實驗，不受外力因素影響正常的執行一次、第二次則有輕微敲擊、第三次則有重擊，以方便比對三種不同量測方法的圖形。

### 2.2 資料傳輸

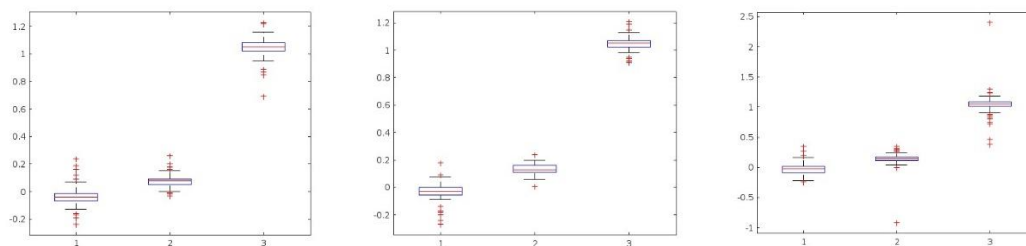
透過 PuTTY 使用 Wi-Fi 連結 IP 位址，將 M5Stack 所收集的資料傳輸至裝置中，並彙整為 csv 檔表格，以利未來的數據分析資料足以更有效率地分析。

### 2.3 資料視覺化

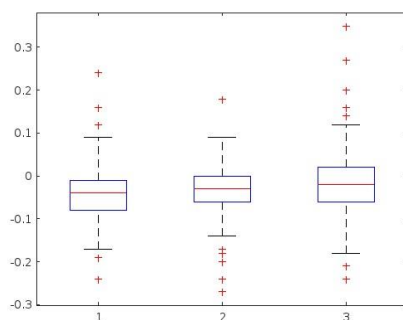
透過 MATLAB 將資料視覺化，將資料繪製成圖可以更直覺地觀測並比較數據，再將混亂的波形進一步傅立葉轉換，可以繪製成有規律周期、更乾淨的數據。本次實驗第一次執行為正常狀態下執行、第二次為中途受到輕微敲擊的影響下執行，第三次為中途受到重擊時的狀態。



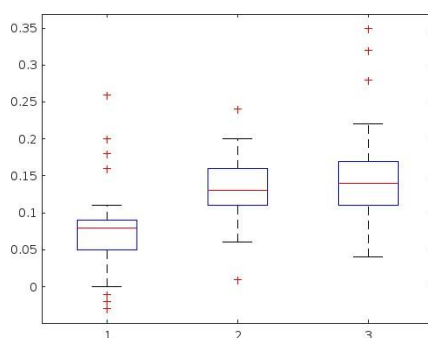
## 2.4 資料信賴區間



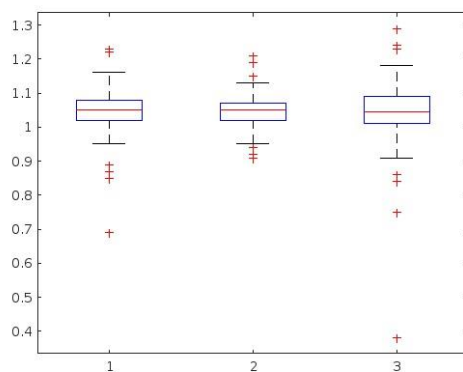
(圖 1-1 三次實驗分別的 XYZ Boxplot)



(圖 1-2 三次實驗分別的 X Boxplot 比較)



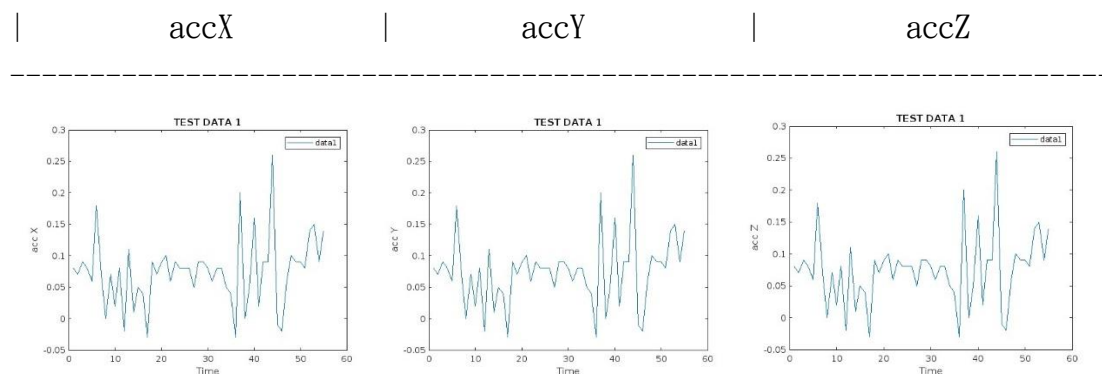
(圖 1-3 三次實驗分別的 Y Boxplot 比較)



(圖 1-4 三次實驗分別的 Z Boxplot 比較)

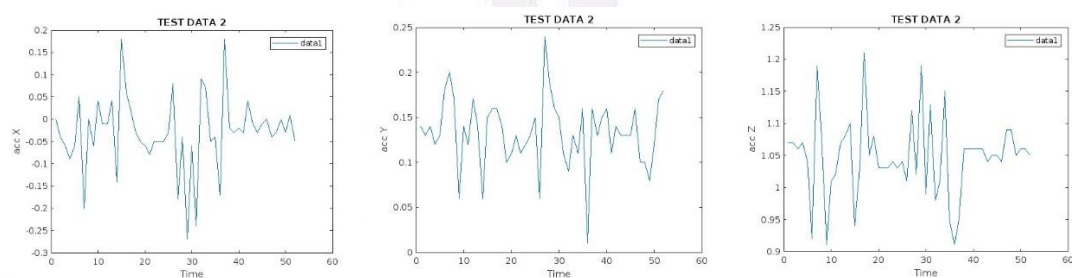
## 2.5 資料比較

### 未轉換數據圖形



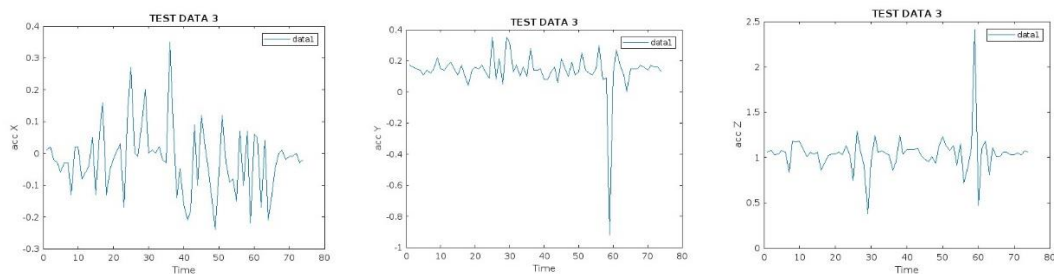
(圖 2-1 第一次試驗之未轉換數據圖)

### 第一次實驗(未受外力影響)



(圖 2-2 第二次試驗之未轉換數據圖)

### 第二次實驗(受到輕微敲擊)



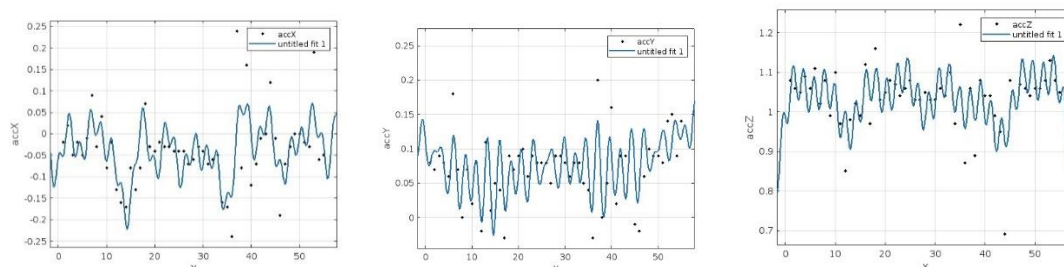
(圖 2-3 第三次試驗之未轉換數據圖)

### 第三次實驗(受到重擊)

## 三角函數轉換數據圖形

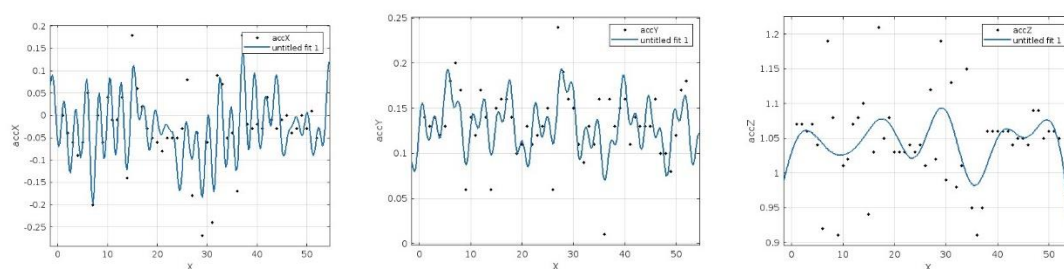
| accX | | accY | | accZ | |

---



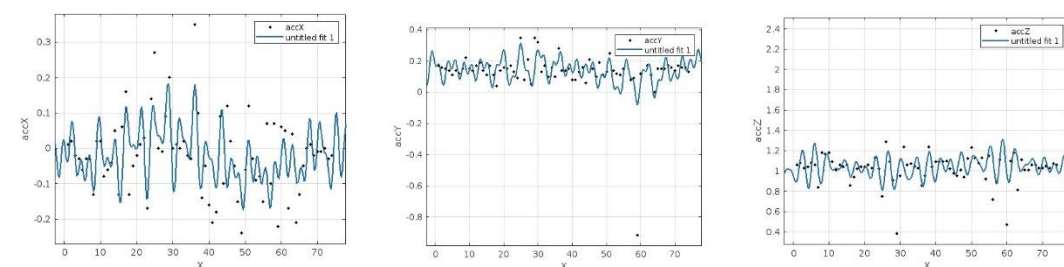
(圖 3-1 第一次試驗之三角函數轉換數據圖)

### 第一次實驗(未受外力影響)



(圖 3-2 第二次試驗之三角函數轉換數據圖)

### 第二次實驗(受到輕微敲擊)



(圖 3-3 第三次試驗之三角函數轉換數據圖)

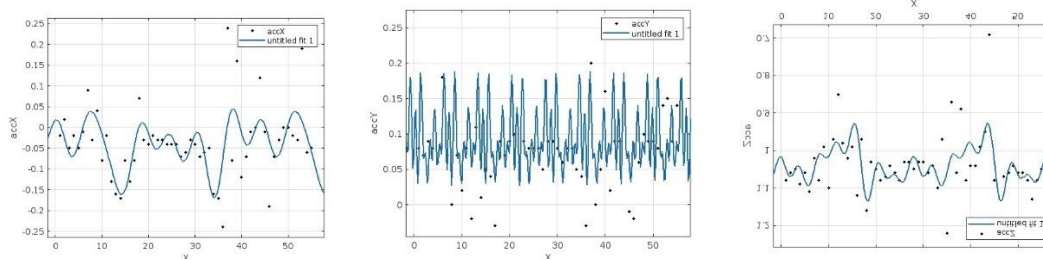
### 第三次實驗(受到重擊)

## 傅立葉轉換數據

accX

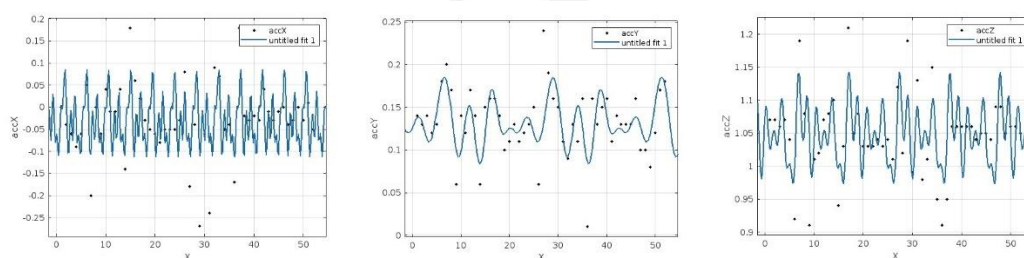
accY

accZ



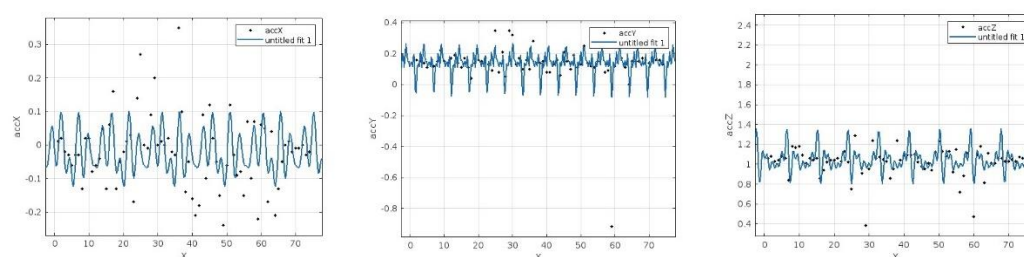
(圖 4-1 第一次試驗之傅立葉轉換數據圖)

第一次實驗(未受外力影響)



(圖 4-2 第二次試驗之傅立葉轉換數據圖)

第二次實驗(受到輕微敲擊)

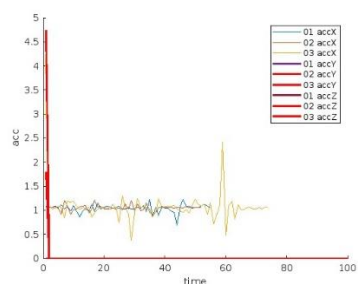


(圖 4-3 第三次試驗之傅立葉轉換數據圖)

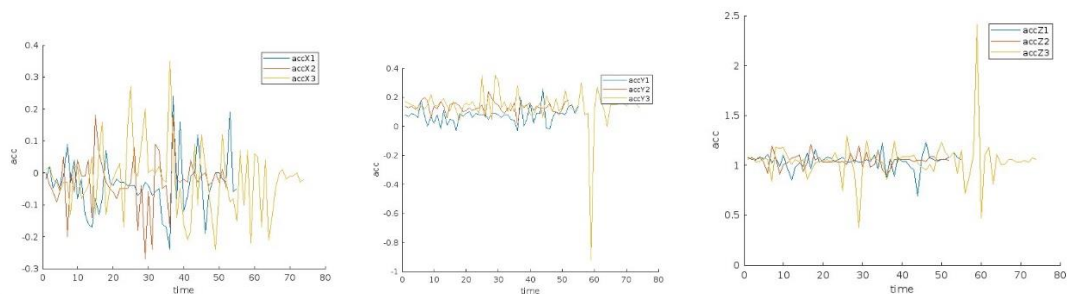
第三次實驗(受到重擊)

## 以 M5Stack 感測器實作 PHM 精密量測機械手臂機台

本實驗將所量測到的資料進一步地使用 MATLAB 軟體分析計算，可以得知這些數據的平均值、變異數、分布情形及 70% 的信賴區間等，可以再使用其他分法來檢測並預測機器健康狀況



(圖 5-1 三次實驗之所有 XYZ 資料波形)



(圖 5-2 三次 X 資料波形)(圖 5-3 三次 Y 資料波形)(圖 5-4 三次 Z 資料波形)

## 第三章 實驗量測與結果討論

### 3.1 感測器振動量測

本次實驗使用 Wi-Fi 功能連接感測器，連接到 Wi-Fi 後按下按鈕即可開始感測振動及溫度，本次感測的目標為振動，因此只會傳送振動的資料。本系統的傳輸程式碼使用暫存器的撰寫方式執行程式，資料會儲存在 tempA、tempB、tempC 三個暫存器中，善用指標可以減少許多程式碼執行所造成的記憶體增加。

### 3.2 結論與未來

本研究提出以 M5 Stack 感測器對機械手臂機台進行振動量測，先將欲量測的機台開啟，使用黏土將感測器固定在移載平台上，使用 Wi-Fi 功能透過軟體 PuTTY 進行資料傳輸，再使用 MATLAB、或其他程式語言進行其他分析，未來可透過 AI 分析並預測機台的使用健康狀況。

實行 PHM 可分為六個步驟：

- 一、蒐集設備數據，震動值在 PHM 中只是其中一項參數，其他還可以收集分析的數據有溫度、電流、電壓等。
- 二、數據處理分析，將收集到的數據透過後續處理轉化為濾波、頻譜圖等資料訊息。
- 三、設備檢測監測，參考各類參數指標來制定門檻值規範等，達到故障警報的功能。
- 四、設備狀態評估，將各式數據化資料記錄累積成趨勢圖表、故障或異常診斷紀錄等評估設備目前狀態。
- 五、設備故障預測，總合數據，可透過 AI 演算法預測下次故障或發生異常的可能，並提前預防。
- 六、保養計畫實行，制定循環計畫，將此六步驟為一週期實現設備健康管理計畫，延長設備使用壽命。

在本次實驗中，實施的步驟為一、二，收集震動數據並把數據處理轉化為頻譜圖等資料，再加以分析。



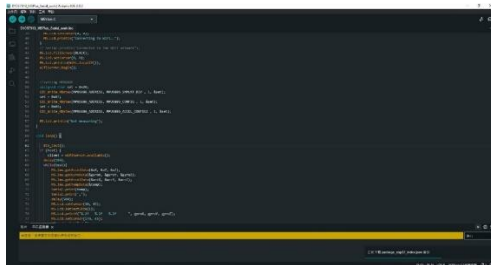
## 參考文獻

---

- [1] 透過 PHM 機制降低設備突發性故障，  
<https://udn.com/news/story/11726/5741289>，2021/09/13，2022/12/29
- [2] 知識分享|不超過3分鐘看懂「PHM」！，  
<https://www.getit01.com/p2018021525411590/>，2022/12/29
- [3] 富鴻網 PHM 設備管理健康方案，<https://www.fhnet.com.tw/PHM.php>，  
2022/12/29
- [4] 鼎新電腦 設備也要健康管理！看科技如何管理設備健康？，  
<https://www.digiwin.com/tw/blog/5/index/2839.html>，2022/12/30
- [5] CIRCUS Pi 運用 M5Stack Core Basic 打造 IoT 智慧植栽，  
[https://www.circuspi.com/index.php/2021/09/14/m5stack\\_iot\\_planting/](https://www.circuspi.com/index.php/2021/09/14/m5stack_iot_planting/)，2022/12/30
- [6] MAKERPRO 科技創新實作社群/媒體 ESP32 的積木開發套件 - M5Stack 有何特色？，  
<https://makerpro.cc/2019/11/esp32-with-m5stack-explore/>，2022/12/30
- [7] Wiki MATLAB，<https://zh.m.wikipedia.org/zh-tw/MATLAB#:~:text=MATLAB%E6%98%AF%E4%B8%80%E7%A8%AE%E7%94%A8%E6%96%BC,%E3%80%81FORTRAN%EF%BC%89%E7%B7%A8%E5%AF%AB%E7%9A%84%E7%A8%8B%E5%BC%8F%E3%80%82>，2022/12/30
- [8] Wiki Fourier Transformation，<https://zh.wikipedia.org/zh-tw/%E5%82%85%E9%87%8C%E5%8F%B6%E5%8F%98%E6%8D%A2>，2022/12/31
- [9] 圖解傅立葉分析，<https://hackmd.io/@sysprog/fourier-transform>，  
2022/12/31
- [10] Wave - Read and write wave files，  
<https://docs.python.org/3/library/wave.html>，2022/12/31
- [11] PHM 如何提高設備的維護效率？ - 固德科技，  
<https://www.goodtechnology.com.tw/blog/21004.html>，2023/01/03
-

## 附錄

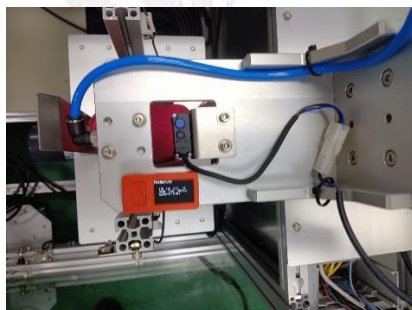
### 附錄一、量測過程



(圖 1-1 M5Stack 量測程式)



(圖 2-1 M5Stack 量測環境擺設示意圖)



(圖 2-2 M5Stack 量測環境擺設示意圖)



(圖 2-3 M5Stack 運作中執行量測圖)

## 以 M5Stack 感測器實作 PHM 精密量測機械手臂機台

```
1 #include <M5StickPlus.h>
2 #include "WiFi.h"
3 WiFiServer wifiServer(80);
4
5 const char* ssid = "Tera_2695";
6 const char* password = "Terrance186";
7 float ax, ay, az;
8 int last;
9 boolean test = false;
10 String s;
11 WiFiClient client;
12
13 // Accel
14 float accX = 0;
15 float accY = 0;
16 float accZ = 0;
17 // Gyro
18 float gyroX = 0;
19 float gyroY = 0;
20 float gyroZ = 0;
21 float temp = 0;
22 float tempA, tempB, tempC;
23 float sum = 0;
```

(圖 2-4 Arduino 預設 Wi-Fi 及全域變數程式碼)

```
24 void setup() {
25   pinMode(13, OUTPUT);
26   Serial.begin(9600);
27   M5.Lcd.setRotation(3);
28   M5.Lcd.fillScreen(BLACK); //clean I/O & fill with black
29   M5.Lcd.setCursor(0, 0); //set text position
30   M5.Imu.init();
31   I2C.begin(I2C_ADDR);
32   // Wi-Fi building
33   WiFi.begin(ssid, password);
34
35   while (WiFi.status() != WL_CONNECTED) {
36     delay(1000);
37     M5.Lcd.fillScreen(BLACK);
38     M5.Lcd.setCursor(0, 0);
39     M5.Lcd.println("connecting to WiFi.");
40   }
41   // Serial.println("Connected to the WiFi network");
42   M5.Lcd.setCursor(0, 0);
43   M5.Lcd.println(WiFi.localIP());
44   wifiServer.begin();
45
46   //setting MPU6050
47   unsigned char i2c = 0x68;
48   I2C_Write_NBytes(MPU6050_ADDRESS, MPU6050_SMPLRT_DIV, 1, &i2c);
49   i2c = 0x07;
50   I2C_Write_NBytes(MPU6050_ADDRESS, MPU6050_CONFIG, 1, &i2c);
51   i2c = 0x00;
52   I2C_Write_NBytes(MPU6050_ADDRESS, MPU6050_ACCEL_CONFIG2, 1, &i2c);
53
54   M5.Lcd.println("Not measuring");
55 }
```

(圖 2-5 Arduino void setup 程式碼)

```
56 void loop() {
57   btn_test();
58   if (test) {
59     client = wifiServer.available();
60     delay(100);
61     while (test) {
62       M5.Imu.getAccelData(&ax, &ay, &az);
63       M5.Imu.getGyroData(&gyroX, &gyroY, &gyroZ);
64       M5.Imu.getTempData(&temp, &tempA, &tempB, &tempC);
65       Serial.print(temp);
66       Serial.println("");
67       delay(500);
68       M5.Lcd.setCursor(10, 45);
69       M5.Lcd.setTextSize(3);
70       M5.Lcd.println("X: %.2f Y: %.2f Z: %.2f", gyroX, gyroY, gyroZ);
71       M5.Lcd.setCursor(10, 45);
72       M5.Lcd.println("a/s");
73       M5.Lcd.setCursor(10, 40);
74       M5.Lcd.println("X: %.2f Y: %.2f Z: %.2f", accX * 1000, accY * 1000, accZ * 1000);
75       M5.Lcd.setCursor(10, 60);
76       M5.Lcd.println("mg");
77       M5.Lcd.setCursor(10, 75);
78       M5.Lcd.println("temperature : %.2f C", temp);
79       M5.Imu.getAccelData(&ax, &ay, &az);
80       s = String(ax) + "," + String(ay) + "," + String(az);
81       client.println(s);
82       Serial.println(s);
83       btn_test();
84       delay(100);
85     }
86     delay(10);
87   }
88 }
```

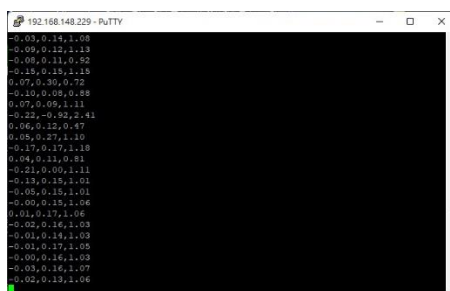
(圖 2-6 Arduino void loop 程式碼)

```
98 void I2C_Write_NBytes(uint8_t driver_Addr, uint8_t start_Addr, uint8_t number_Bytes, uint8_t *write_Buffer) {
99
100   Wire1.beginTransmission(driver_Addr);
101   Wire1.write(start_Addr);
102   Wire1.write(*write_Buffer);
103   Wire1.endTransmission();
104
105 }
106
```

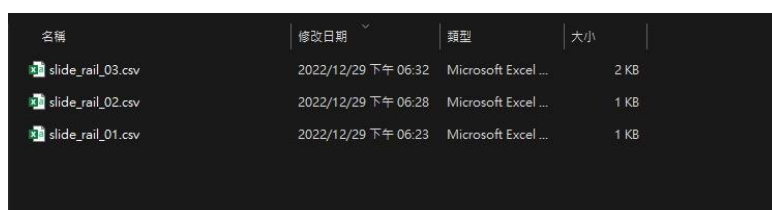
(圖 2-7 Arduino void I2C\_Write\_NBytes 程式碼)



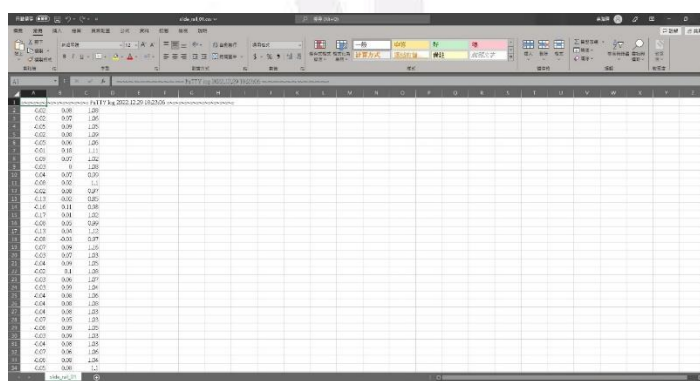
以 M5Stack 感測器實作 PHM 精密量測機械手臂機台



(圖 3-2 PuTTY 資料傳輸畫面)



(圖 3-3 資料彙整為 CSV 檔至指定路徑)



(圖 3-4 CSV 檔中之 acceleration 資料)

### 附錄三、相關理論

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \xi} dx, \xi \text{ 為任意實數。}\xi \text{ 的定義域為頻域。}$$

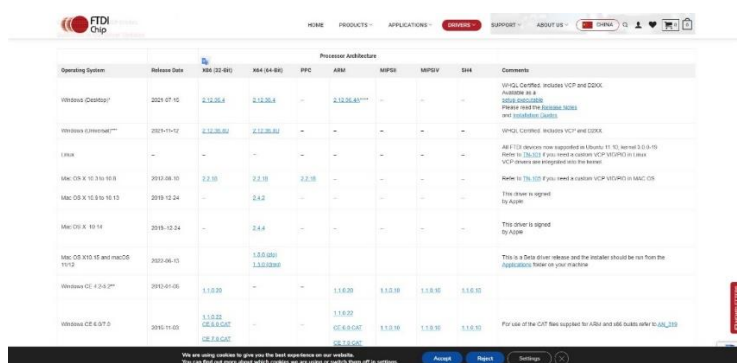
(圖 4-1 傅立葉轉換 頻域)

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi) e^{2\pi i \xi x} d\xi, x \text{ 為任意實數。}x \text{ 的定義域為時域。}$$

(圖 4-2 傅立葉轉換 時域)

以 M5Stack 感測器實作 PHM 精密量測機械手臂機台

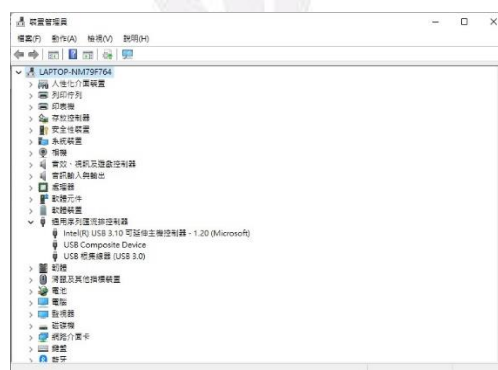
## 附錄四、M5Stack C Plus Arduino Code 環境安裝



(圖 5-1 下載驅動器)



(圖 5-2 開啟裝置管理員)



(圖 5-3 尋找 COM 序列埠)

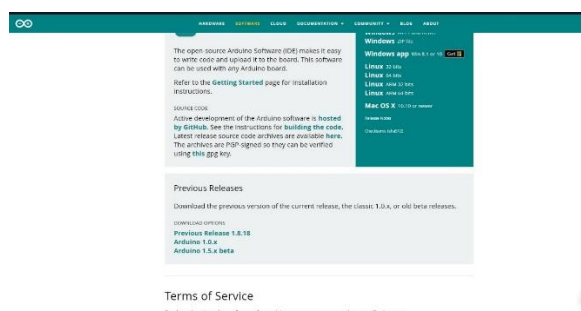


(圖 5-4 尋找驅動程式)

## 以 M5Stack 感測器實作 PHM 精密量測機械手臂機台



(圖 5-5 使用下載好的驅動程式)



(圖 5-6 安裝 Arduino IDE )



(圖 5-7 開啟 Preference 貼上網址

[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)，並關閉。)



(圖 5-8 開啟工具找到開發板(boards))

以 M5Stack 感測器實作 PHM 精密量測機械手臂機台



(圖 5-9 開啟工具找到開發板管理器(boards manager))



(圖 5-10 尋找 esp32)



(圖 5-11 打開程式庫)



(圖 5-12 打開管理庫)

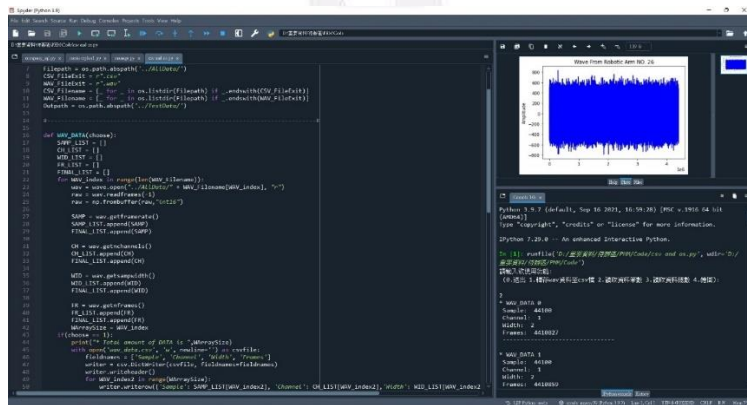




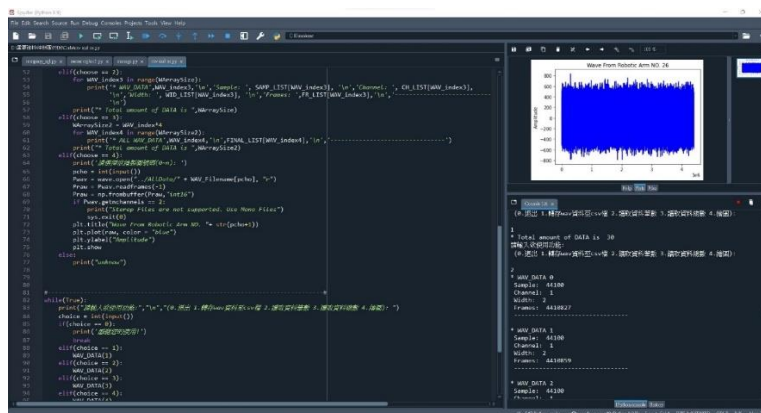
(圖 5-13 搜尋 M5StackCPlus 並下載)

## 附錄五、資料分析

本實驗之資料亦可透過程式語言將資料進一步地分析，機器振動之聲波亦可作為 PHM 之分析資料，本實驗使用 Python 分析機器之振動 WAV 檔使用 Wave 套件進行分析，可獲取相關之 channels、sampwidth、framerate 等相關資料，並寫入 CSV 檔或繪製成圖再進行轉換。

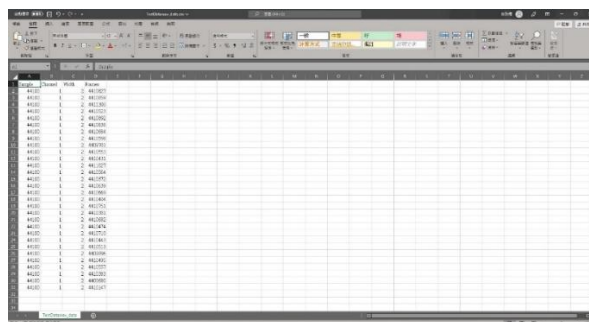


(圖 6-1 WAVE 分析之 Python 程式碼執行過程)



(圖 6-2 WAVE 分析之 Python 程式碼執行過程)

以 M5Stack 感測器實作 PHM 精密量測機械手臂機台



(圖 6-3 WAVE 分析之輸出 csv 檔)

		mean mu	std	VAR	pdf	rawdata
1	01 accX	-0.0375	0.0866	0.0075	1×100 double	55×1 double
2	02 accX	-0.0300	0.0842	0.0071	1×100 double	52×1 double
3	03 accX	-0.0201	0.1076	0.0116	1×100 double	74×1 double
4	01 accY	0.0740	0.0548	0.0030	1×100 double	55×1 double
5	02 accY	0.1331	0.0388	0.0015	1×100 double	52×1 double
6	03 accY	0.1362	0.1397	0.0195	1×100 double	74×1 double
7	01 accZ	1.0404	0.0834	0.0070	1×100 double	55×1 double
8	02 accZ	1.0479	0.0630	0.0040	1×100 double	52×1 double
9	03 accZ	1.0441	0.2167	0.0470	1×100 double	74×1 double

(圖 6-4 M5 Stack 結束量測，顯示總平均結果)

```

pdx1 = fitdist(slide_rail_01.accX, 'Normal')
cix1 = paramci(pd)

pdy1 = fitdist(slide_rail_01.accY, 'Normal')
ciy1 = paramci(pd)

pdz1 = fitdist(slide_rail_01.accZ, 'Normal')
ciz1 = paramci(pd)

pdx2 = fitdist(slide_rail_02.accX, 'Normal')
cix2 = paramci(pd)

pdy2 = fitdist(slide_rail_02.accY, 'Normal')
ciy2 = paramci(pd)

pdz2 = fitdist(slide_rail_02.accZ, 'Normal')
ciz2 = paramci(pd)

pdx3 = fitdist(slide_rail_03.accX, 'Normal')
cix3 = paramci(pd)

pdy3 = fitdist(slide_rail_03.accY, 'Normal')
ciy3 = paramci(pd)

pdz3 = fitdist(slide_rail_03.accZ, 'Normal')
ciz3 = paramci(pd)
    
```

(圖 6-5 MATLAB 常態分佈下之 95%信賴區間計算程式碼)

```

pdx1 =
NormalDistribution
Normal distribution
mu = -0.0374545 [-0.0688578, -0.0148413]
sigma = 0.0866072 [0.0729115, 0.106686]
cix1 = 2x2
-0.0609 0.0729
-0.0140 0.1067

pdy1 =
NormalDistribution
Normal distribution
mu = 0.074 [0.0591875, 0.0888125]
sigma = 0.0547925 [0.0461279, 0.0674958]
ciy1 = 2x2
-0.0609 0.0729
-0.0140 0.1067

pdz1 =
NormalDistribution
Normal distribution
mu = 1.04036 [1.01782, 1.0629]
sigma = 0.083377 [0.0701921, 0.102707]
ciz1 = 2x2
-0.0609 0.0729
-0.0140 0.1067

pdx2 =
NormalDistribution
Normal distribution
mu = -0.03 [-0.0534423, -0.00655765]
sigma = 0.0842033 [0.0705664, 0.104423]
cix2 = 2x2
-0.0609 0.0729
-0.0140 0.1067

pdy2 =
NormalDistribution
Normal distribution
mu = 0.133077 [0.12228, 0.143874]
sigma = 0.0387824 [0.0325015, 0.0480951]
ciy2 = 2x2
-0.0609 0.0729
-0.0140 0.1067

pdz2 =
NormalDistribution
Normal distribution
mu = 1.04788 [1.03034, 1.06543]
sigma = 0.0630075 [0.0528033, 0.0781373]
ciz2 = 2x2
-0.0609 0.0729
-0.0140 0.1067

pdx3 =
NormalDistribution
Normal distribution
mu = -0.0201351 [-0.0450717, 0.00480146]
sigma = 0.107633 [0.0926518, 0.128439]
cix3 = 2x2
-0.0609 0.0729
-0.0140 0.1067

pdy3 =
NormalDistribution
Normal distribution
mu = 0.136216 [0.103847, 0.168585]
sigma = 0.139713 [0.120266, 0.16672]
ciy3 = 2x2
-0.0609 0.0729
-0.0140 0.1067

pdz3 =
NormalDistribution
Normal distribution
mu = 1.04405 [1.033852, 1.09426]
sigma = 0.216687 [0.186526, 0.258574]
ciz3 = 2x2
-0.0609 0.0729
-0.0140 0.1067
    
```

(圖 6-6 MATLAB 三次測驗之 95%信賴區間計算結果)