

東 海 大 學

工業工程與經營資訊學系

碩士論文

IMU 感測系統於運動軌跡的應用

研 究 生：林晉裕
指 導 教 授：林育儒 教授

中 華 民 國 一 〇 七 年 七 月

The Application of IMU System to the Tracking

By
Jin-Yu Lin

Advisor : Prof. Yu-Ju Lin

A Thesis
Submitted to the Institute of Industrial Engineering and Enterprise
Information at Tunghai University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in
Industrial Engineering and Enterprise Information

July 2018
Taichung , Taiwan

應 IMU 感測系統於運動軌跡的應用

學生：林晉裕

指導教授：林育儒 博士

東海大學工業工程與經營資訊學系

摘 要

隨著運動參與度的提升，越來越多運動員依靠輔助學習的設備做學習，然而輔助學習設備的成本過高，導致輔助學習設備並不普及，因此本研究試圖以 IMU 感測元件搭配低通濾波器及 AHRS 姿態解算法建構一套軌跡重建系統，提供給運動員進行動作修改時有個準確的參考依據。其中的資訊包含了運動時間、軌跡、速度、加速度，協助使用者在於動作改良上的參考。

本研究使用 Arduino NANO 的微控制器 ATmega-328 控制 MPU-9250 感測元件收集物體在三維空間中角速度及加速度資訊，經由 HC-06 藍芽傳輸模組將數據傳輸到隨身電腦，利用 MATLAB 直接對所設定的傳輸埠進行數據接收並進行濾波，將濾波後數據經由 AHRS 姿態解算法計算出所須的資料後，建構出三維的軌跡模型，並且將基本資訊逐一顯示。

為了進行感測系統軌跡的驗證，本研究將感測器分別移動不同軌跡，根據感測系統所輸出的軌跡對照實際軌跡，最後查看兩者誤差程度，驗證研究之感測系統是否準確。

關鍵字詞：軌跡重建、低通濾波器、IMU、MPU-9250、HC-06、AHRS

The Application of IMU System to the racking

Student : Jin-Yu Lin

Advisor : Prof. Yu-Ju Lin

Department of Industrial Engineering and Enterprise Information
Thunghai University

ABSTRACT

Because of the participation in sports arise, more and more athletes use the equipment which is used to supply learning. But the cost of this equipment is too high, makes the learning equipment is not popular. This research attempts to construct a sensing system with IMU sensor and low-pass filter and AHRS to provide the accurate reference for athletes to modify their actions. The information which is provided by sensing system includes during times, tracking, speed, acceleration. These information will assist athletes in the reference of motion improvement.

We used the Arduino NANO compact board which has a microcontroller ATmega-328. ATmega-328 control the MPU-9250 to collect the motions and used the Bluetooth chip to transport the data to PC computer. Using MATLAB to run low-pass filter to filter out the noises. And used the data which had been filtered via AHRS to calculate the information which we need. Constructing three-dimensional tracking model , and appear all of the bases information.

In order to verify the tracking of the sensing system. We will move the sensor to different tracking. And we will compare the system tracking with actual tracking to verify that sensing system of this research is accurate.

Keywords: Tracking 、 Low Pass Filter 、 IMU 、 MPU-9250 、 HC-06 、 AHRS

致謝詞

大學四年一瞬即過，緊接著投入了研究所的生活，進入研究所不僅僅是家人的期望，也是對自己的期許，期許著自己能夠在這兩年之中充實自己。或許在研究所生涯的前期有些迷惘，不清楚自己在做什麼，同屆的同學都已經進入職場，而自己卻還這裡迷糊糊，過了半年的沉澱，開始驚覺自己該為了自己的未來好好做努力了，不能辜負了家人辛苦的支持與期待。

當時初入研究所時，對於研究所的內容一竅不通，在研究所期間所學到的遠遠超過大學四年所學的，謝謝指導老師林育儒老師的指導，老師不嫌麻煩，細心的教導，讓我從沒有半點該專業項目的基礎下還能順利學習，如果沒有老師的指導，即使經過這兩年的時間，我可能還是與兩年前剛從大學畢業時的我沒有什麼不同，這兩年的時間真的讓我受益良多。

在讀研究所的過程中，除了老師的教導之外，更多的是同儕的陪伴，沒有同儕的陪伴，或許在這兩年的期間內我就已經放棄了，謝謝學弟妹至翔、富群、博元、芷俞、孟勳、逢齊和順景的陪伴，讓我在研究所的生涯中不只是只有上課、開會與報告，還有著是大家一起放鬆娛樂的時間，這些回憶將會永遠留在心中，也祝福大家在未來的一年中可以順利的畢業，對於自己所做的研究中也可以有著不錯的成就。

謝謝系隊與校隊在過去六年的陪伴，尤其在研究所的這兩年中，因課業的關係導致練習時間少了許多，但我永遠記得，在球場上的我是如此的開心無壓力，雖然沒有取得最好的名次，但我珍惜與大家打球的時光，希望大家可以順利畢業，除此之外可以盡情的享受在球場上的自己，開心打球。

最後要感謝的是我的家人，在我就學的期間因課業與球隊的關係，明明家裡與學校只有距離麼一點距離，但我回家的次數卻不多，但家人們總是默默的支持，沒有你們的支持，這兩年我真的無法熬過，謝謝家人們的無私與關心，在過去是你們在照顧我，現在換我來照顧你們了。

林晉裕 謹誌於

東海大學工業工程與經營資訊學系研究所

中華民國一〇七年七月

目錄

摘要.....	i
ABSTRACT.....	ii
致謝詞.....	iii
目錄.....	iv
表目錄.....	vii
圖目錄.....	viii
第一章 緒論.....	1
1.1 研究背景.....	1
1.2 研究動機.....	2
1.3 研究問題.....	2
1.4 研究目的.....	3
1.5 研究架構.....	3
第二章 相關知識.....	4
2.1 加速度計.....	4
2.1.1 加速度計的介紹.....	4
2.1.2 加速度計的檢測機制.....	4
2.2 陀螺儀.....	6
2.2.1 陀螺儀的介紹.....	6
2.2.2 陀螺儀的檢測機制.....	6
2.3 磁力計.....	7
2.4 低通濾波器.....	8
2.5 四元數.....	8
第三章 文獻探討.....	10
3.1 以影像為基礎的動作辨識.....	10
3.2 慣性感測器的應用於人體姿態辨識.....	11
3.3 誤差校正法.....	13
第四章 系統架構建構.....	15
4.1 系統架構.....	15
4.1.1 硬體組成.....	16
4.1.2 姿態數據收集.....	16
4.1.3 資料分析.....	16

4.2 硬體設備.....	17
4.2.1 ATmega-328	17
4.2.2 Grove-IMU-10DOF	18
4.2.3 MPU-9250.....	19
4.2.4 HC-06.....	22
4.3 姿態數據分析與轉換.....	23
4.3.1 加速度計與陀螺儀飄移校正.....	23
4.3.2 低通濾波器進行濾波.....	24
4.3.3 AHRS 姿態解算數據計算.....	25
4.3.4 四元數驗證.....	33
4.3.5 地球坐標軸轉換.....	34
4.3.6 軌跡計算.....	35
第五章 系統實驗結果.....	38
5.1 環境及實驗設計.....	38
5.2 設備畫面.....	38
5.3 六軸 AHRS 與九軸 AHRS 比較.....	40
5.3.1 六軸 AHRS	40
5.3.2 九軸 AHRS	43
5.4 實驗結果與真實軌跡比較.....	45
5.4.1 平面 L 形軌跡	45
5.4.2 平面 W 形軌跡.....	48
5.4.3 三維運動軌跡.....	51
第六章 結論與未來研究.....	54
6.1 結論.....	54
5.1 未來研究.....	55
參考文獻.....	56
附錄一.....	57
附錄二.....	61

表目錄

表 2.1 旋轉表示法比較	9
表 4.1 Grove- IMU-10DOF 與 ATmega-328 連接對照表	18
表 4.2 MPU-9250 加速度計靈敏度對照表	20
表 4.3 MPU-9250 陀螺儀靈敏度對照表	21
表 4.4 MPU-9250 磁力計靈敏度表	21
表 4.5 HC-06 與 ATmega-328 連接腳位對照表	22
表 5.1 六軸 AHRS 轉換之尤拉角	41
表 5.2 九軸 AHRS 轉換之尤拉角	43
表 5.3 L 形三點座標	47
表 5.4 L 形軌跡誤差	47
表 5.5 W 形 5 點座標	49
表 5.6 W 形軌跡誤差	50
表 5.7 三維運動軌跡座標點	53
表 5.8 三維運動軌跡誤差	53

圖目錄

圖 2.1 運動物體及電容	4
圖 2.2 單向運動物體的相對加速度	5
圖 2.3 傅立葉轉換後波型	8
圖 3.1 Mahony 所提出的 PCF 架構	13
圖 3.2 Collinson 所討論之互補濾波器架構	13
圖 4.1 系統架構圖	15
圖 4.2 ATmega-328 晶片腳位功能對照	17
圖 4.3 姿態數據分析流程圖	23
圖 4.4 濾波前後比較圖	24
圖 4.5 加速度計傾角計算	25
圖 4.6 尤拉角示意圖	33
圖 4.7 加速度積分法	35
圖 4.8 公式 4.8 積分法	36
圖 4.9 公式 4.8 積分法誤差.....	36
圖 4.10 公式 4.10 積分法	36
圖 4.11 公式 4.10 積分誤差抵銷.....	36
圖 5.1 感測器設備	38
圖 5.2 感測器接線圖	39
圖 5.3 六軸感測器軌跡模型	40
圖 5.4 坐標軸晃動方向	42
圖 5.5 L 形軌跡 XY 平面軌跡圖	45
圖 5.6 L 形軌跡三維軌跡圖	46
圖 5.7 L 形軌跡三軸加速度值	46
圖 5.8 L 形軌跡三軸位移值	47
圖 5.9 W 形軌跡 XY 平面軌跡圖	48
圖 5.10 W 形軌跡三維軌跡圖	48
圖 5.11 W 形軌跡三軸加速度值	49
圖 5.12 W 形軌跡三軸位移值	49
圖 5.13 三維運動軌跡 - XY 平面軌跡.....	51
圖 5.14 三維運動軌跡 - XZ 平面軌跡	51
圖 5.15 三維運動軌跡 - YZ 平面軌跡	52
圖 5.16 三維運動軌跡 - 立體空間軌跡	52
圖 5.17 三維運動軌跡三軸加速度值	52
圖 5.18 三維運動軌跡三軸位移值	53

第一章 緒論

本研究緒論部分共分為 4 節，分別為：第一節 研究背景；第二節 研究動機；第三節 研究目的；第四節 研究問題。

1.1 研究背景

翻開近年來台灣參與的各項國際賽事的轉播與進場觀眾紀錄當中，可以發現到現今台灣民眾對於各項運動項目的投入與支持，除了對於運動比賽的投入與支持之外，民眾也隨著健康意識的抬頭，開始投入各項運動之中，希望藉由運動來取得更健康的生活，提升自身的生活品質。根據教育部體育署的統計數據：台灣規律人口運動人口自從 98 年的 24.4%，提升到 104 年的 33.4%，顯示了這幾年來台灣民眾規律運動參與率呈現增加的趨勢，除此之外，教育部體育署也針對非規律運動參與比例做調查，從 97 年以來已經連續 9 年超過維持八成以上的運動參與比例，代表著台灣民眾對於各項運動的高度參與。

除此之外，台灣政府也積極推動各項運動相關的政策，希望藉由政策吸引更多民眾進行規律的運動，而大多數成人的運動習慣都是從小就培養而成，「成人期健康的根基是來自於青少年時期的健康，相同的，成人期後的運動行為也和青少年時期的運動習慣，有明顯的相關。」(黃銘廣, 2004)，因此政府對於學生的運動習慣培養更是關注。

1.2 研究動機

在政府政策的倡導之下，現今已有許多學生及民眾投入於規律運動之中，更有部分學生不僅僅是進行規律運動而已，更朝著競技運動項目中投入，不管是一般的運動項目或是競技項目，最重要是有著正確的基礎動作，利用正確的基礎動作培養出正確的姿勢，避免日後因姿勢不良造成運動表現不佳，甚至造成一些無法預期的運動傷害，而競技運動項目對於動作及姿勢的要求又是更加嚴格。

在過去的練習中，也有一些方法可以幫助學生來學習各項動作，包括了專業老師的指導，錄製影片分析等等，隨著科技的進步，開始有一些輔助學習的設備出現，這些設備大大的提升學生在修改各式動作的能力。「國外訓練用一些輔助器材，先固定動作後才有辦法完全的做完整個動作。」(王建民，2018)然而，這些輔助學習的設備並不是太普及，甚至有些設備只有在一些專科大學中才有配備，資源的落差導致了學生在學習上的不平衡，以至於學生在最後的表現可能達不到理想的要求，更有可能因不良的動作姿勢，對於學生產生了一些傷害。

1.3 研究問題

隨著科技的進步，在各項運動的學習上，不再只是過去那般的傳統式教學，只能依靠學生們各自的體會能力，導致學習效果不一定非常顯著，在一些專業運動學校裡，開始配合一些運動輔助學習的設備，幫助學生在學習上有更多的資訊作為參考，提升學生學習速度，但是這些輔助學習的設備往往因為環境及設備本身的限制，導致只出現在較為專業的學校中，在一般的學校中可能就不那麼普及。

現在市面上的輔助學習設備普遍上售價都不便宜，除此之外，在使用上的時候可能還會有環境上的限制，像是某些設備必須待在相關的實驗室之中，讓一些能在室外進行的運動在使用上有所限制，因此，本研究想針對這方面進行研究，提供最佳的輔助學習設備。

1.4 研究目的

根據上述所提及的，為了讓學生更容易的去了解各項動作如何去學習或者是將本身的動作資訊顯示出來，作為修改動作的參考依據，因此，本研究希望利用加速度計配合陀螺儀建立一套軌跡重建系統，再由 AHRS 姿態解算計算出地球座標軸的感測數據，將數據經由藍芽傳輸到電腦設備，由 MATLAB 做濾波處理並計算速度及位移，最後將軌跡繪製出來，提供資訊給學生作為參考依據。

此套系統包含以下功能，攜帶式且簡易使用的輔助設備、完整的運動軌跡重建、提供各個視角的軌跡、於介面上顯示運動過程中各項數據。

1.5 研究架構

本研究共分為六個章節，第一章為緒論，主要說明本研究之研究背景、動機、目的、研究方法概述以及研究架構。第二章為相關知識，主要探討本研究會利用到的硬體設備以及資料處理方式，簡單介紹讓讀者大致上有所了解。第三章為文獻探討，主要探討加速度計、陀螺儀以及 AHRS 之過往的文獻資料，提供給讀者查閱其相關資料。第四章為研究模型之建構，說明整套系統是使用哪些工具，例如加速度計及陀螺儀之使用，以及數據傳輸方式。第五章則是展示整套系統實際運作模式以及運行之結果。第六章總結本研究所完成之研究成果、貢獻與未來研究方向。

第二章 相關知識

本章節會針對本研究所用到的相關知識做介紹。2.1 加速度計的原理及介紹，2.2 陀螺儀的原理及介紹，2.3 磁力計的原理及介紹 2.4 低通濾波器的原理及介紹 2.5 四元數介紹 2.5 AHRS 介紹。

2.1 加速度計

2.1.1 加速度計介紹

加速度計(Accelerometer):是屬於慣性測量元件(Inertial Unit)，可以用來偵測物體於三維空間中的加速度。根據牛頓第二運動定律: $F = ma$ ，物體的加速度($\frac{m}{s^2}$)與物體所受到的合力(N)成正比，與物體的質量(KG)成反比，物體的加速度方向與合力相同。而加速度計的作用力檢測機制擷取作用裡所產生的加速度。所以加速度計實際測量的值是力，而不是我們所認知的加速度；加速度計是藉由檢測施加於其中一個軸向的作用力來間接測量加速度值。

2.1.2 加速度計的檢測機制

加速度計最常見的檢測方式為電容的檢測，物體的加速度與物體運動時所產生的電容變化有關如圖 2.1。這種檢測技術的優點為精準度高、穩定度高、功耗低、且結構簡單，除此之外，這種檢測技術也不易受到外在的雜訊及溫度的波動影響。

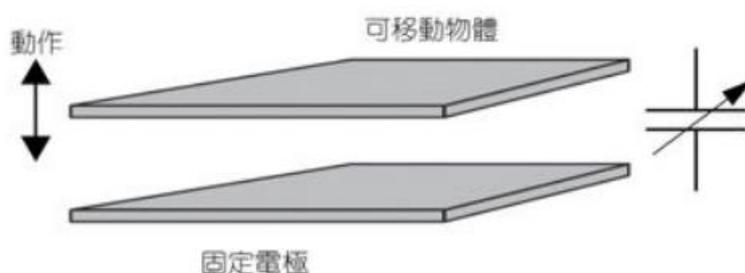


圖 2.1 運動物體及電容

(資料來源:Majid Dadafshar, 2014)

而電容結構可以分為單側或者是差分對，下圖 2.2 就以差分對加速度計作為介紹。整個加速度計包含了單個可移動物體，其中一側的表面，五體沿著彈簧方向放置，介於兩個固定的矽基板或是電極(做為另一平面)之間。物體相對於固定電極(d1 和 d2)的運動(動作 X)很明顯地會使電容產生變化(C1 和 C2)。最後再計算 C1 及 C2 的差值，就可以得到物體重心的位移及方向。

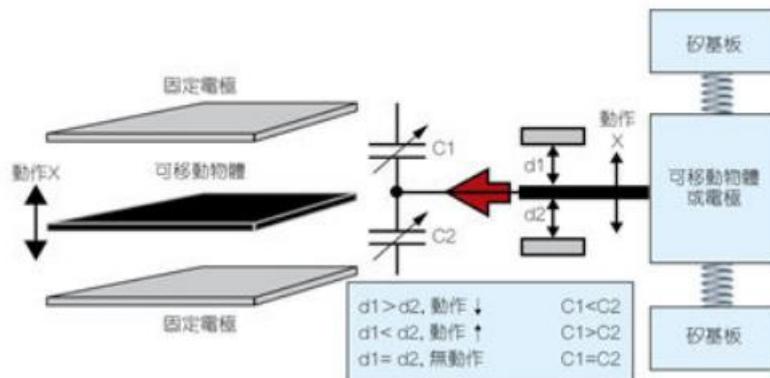


圖 2.2 單向運動物體的相對加速度
(資料來源:Majid Dadafshar, 2014)

上述過程可以簡單歸納成:對物體施力，作用力導致物體發生位移，進而產生電容變化，而物體重心的類比電壓經過電荷放大、訊號調節以及低通濾波器等，再經由 $\Sigma - \Delta$ 調變($\Sigma - \Delta$ ADC)轉為數位訊號，最後經由轉換公式將其資訊轉換成重力加速度的單位(g)。

接下來以一個簡單例子做舉例，假設有一 10-bit 類比數位轉換器，其 X 軸輸出值 $AdcRx=586$ ，而每個類比數位轉換器都有參考電壓值，假設再本例子當中為 3.3V，則轉換 10-bit 類比數位轉換器的輸出值公式為:

$$VoltRx = \frac{AdcRx \times v_{ref}}{1023} \quad (2.1)$$

將上述的 $AdcRx = 586$ 帶入公式 2.1 中，求得 $VoltRx = \frac{586 \times 3.3V}{1023} \cong 1.89V$

每個加速度計都會有零重力電壓標準值，為了求得正確的電壓值，需減去零重力情況下的電壓值，假這此例子的零重力電壓值為 1.65V，則電壓值(ΔVR_x)減去零重力電壓後為：

$$\Delta VR_x = 1.89V - 1.65V = 0.24V$$

求得電壓值後此時的輸出單位為伏特，還不是 $g(9.8m/s^2)$ ，為了最後的轉換，需要再將加速度計的靈敏度加入計算，靈敏度通常以 mV/g 作為單位，靈敏度可由產品的相關資料中取得，在此例子當中，假設靈敏度為 $478.5mV/g$ ，而轉化的公式為：

$$R_x = \frac{\Delta VR_x}{Sensitivity} \quad (2.2)$$

將 ΔVR_x 以及靈敏度數值帶入公式 2.2 中，求得 $R_x \cong 0.5g$ ， R_x 為加速度計再 X 軸方向的數值，Y 軸及 Z 軸算法與 X 軸相同。

(資料來源: STARLINO,2009)

2.2 陀螺儀

2.2.1 陀螺儀介紹

陀螺儀(Gyroscopoe)又名角速度計，其測量的單位為(度/秒)的角速度，測量物體的旋轉狀態，基於旋轉物體的旋轉軸所指的方向不受外力影響的原理來保持方向；使用陀螺儀時，施予一作用力，令它快速旋轉，運用多種方式感測軸所指的方向，並將數據回傳。

2.2.2 陀螺儀的檢測機制

每個陀螺儀測量圍繞著該軸的旋轉，舉例來說，一個三軸的陀螺儀會偵測在 X 軸、Y 軸及 Z 軸的旋轉，為了要以數字表示旋轉，先說明表示法：

R_{xz} ：向量力 R 在 XZ 平面的投影

接著定義 Z 軸與 R_{xz} 之間的角度關係：

$A_{xz} - R_{xz}$ ：與 Z 軸之間的夾角

陀螺儀就是偵測上述所提角度的變化速度，換句話說，陀螺儀的輸出數值會與角度的變化呈現線性相關，舉例來說，當時間為 t_0 ，繞 Y 軸 (A_{xz}) 的旋轉角度為 A_{xz0} ，而在時間為 t_1 時，該角度變化為 A_{xz1} ，則該角度變化速度為：

$$RateA_{xz} = (A_{xz1} - A_{xz0}) / (t_1 - t_0)$$

現實中的陀螺儀極少部分會直接輸出以 deg/s 為單位的數值，這部分和加速度計相同，陀螺儀也同樣需要類比數位轉換器，透過類似於 2.1.1 的公式，將輸出值轉換為 deg/s 為單位的輸出值，而陀螺儀的類比數位轉換器公式如下(此例題同樣使用 10bit 類比數位轉換器)：

$$\text{RateA}_{xz} = (\text{AdcGyro}_{xz} \times V_{\text{ref}}/1023 - V_{\text{zeroRate}}) / \text{Sensitivity} \quad (2.3)$$

AdcGyro_{XZ} 可以從類比數位轉換器模組取得，用來測量向量 R 在 XZ 平面投影所形成的旋轉角度的頻道，也就是說旋轉是繞著 Y 軸轉而形成。 V_{ref} 為類比數位轉換器參考電壓，上述例題以 2.3 為例。 V_{zeroRate} 為零速度電壓，也就是在沒有旋轉的情況下所測得的電壓值，本題以 1.23V 為例。 Sensitivity 為陀螺儀的敏感度，單位是 mV/deg/s，表示每增加 1 deg/s 的旋轉速度，則陀螺儀的輸出值會上升多少 mV，在此例題中， Sensitivity 的值為 2 mV/deg/s。

假設此例題 $\text{AdcGyro}_{XZ} = 571$ ，將 $\text{AdcGyro}_{XZ} = 571$ 及 $\text{Sensitivity} = 2\text{mV/deg/s}$ 帶入公式 2.3 之中求得 $\text{RateA}_{xz} \cong 306 \text{ deg/s}$ ，換句話說，這裝置繞著 Y 軸做旋轉的速度為 306 deg/s。

(資料來源: STARLINO,2009)

2.3 磁力計

磁力計為測量該地點磁場的儀器，也稱做磁強計、高斯計。在國際單位制之中描述磁場的物理量為磁感應強度，單位為特斯拉。由於 1 特斯拉代表著事非常強的磁場，因此地球科學上常用納特(nT)來當作測量單位。磁感應強度是矢量，具有大小及方向的性質，指測量磁感應強度大小的磁力計稱為標量磁力計，而可以測量特定方向磁場大小的磁力計稱作矢量磁力計。能夠測量磁場的物理原理有很多，根據不同原理進行分類，常見的矢量磁力計有磁通門磁力計、磁力阻磁力計。

2.4 低通濾波器

濾波器的用途在於抑制並衰減雜訊的強度，換句話說就是將雜訊以其他形式消耗掉，主要做法以頻率作為依據，決定哪些雜訊可以被濾除，低通濾波器為容許低頻率訊號通過，並將頻率高於截止頻率的訊號進行攔截。對於不同濾波器而言，每個頻率的訊號的減弱程度不同，在使用時可依照需求進行濾波器的設定。除此之外，低通濾波器也可以更改成高通濾波器，其效果剛好與低通濾波器相反。以圖 2.3 做介紹，黑色的波為原始的波，經過傅立葉轉換後發現原始波為兩個波所組成，而這兩種波的頻率並不一致，此時就可設定及截止頻率，將不需要的波去除。

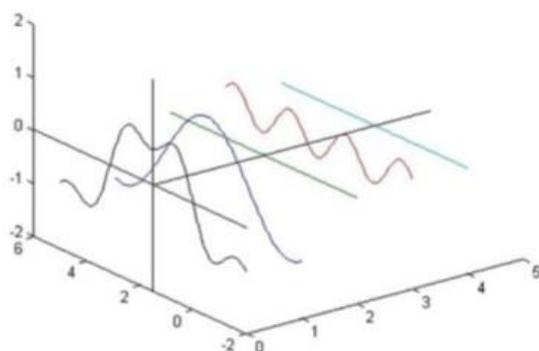


圖 2.3 傅立葉轉換後波型
(資料來源: Lapuda,2014)

2.5 四元數

旋轉為三軸座標轉變(縮放、旋轉、平移)最複雜的一種，較為多人使用及知悉的兩種旋轉表示法為旋轉矩陣和歐拉角。旋轉矩陣是使用了一 4×4 大小的矩陣來表示任意軸旋轉的變換矩陣，而歐拉角則是按照一定的座標軸順序做計算(如先 X 再 Y 最後 Z)，每個座標軸旋轉一定的角度來變換座標或是向量，而事實上他是一系列的座標軸旋轉的組合。

四元數於 1843 年由愛爾蘭數學家 William Hamilton 所提出的概念，從明的角度而言，四元數是複數的不可交換延伸。如果把四元數的集合想像成多為實數空間，四元數就代表著一個四維的空間，相較於一般複數只是一個二維空間。四元數為實部和虛部所組成，而四元數的虛部包含了三個虛數單位， i 、 j 、 k ，則四元數表示為 $q = W + xi + yj + zk$ 。

四元數大量使用於電腦繪圖(圖像分析)上所表示三圍物體的旋轉及方位。在發現四元數之前較為多人使用的是尤拉角及旋轉矩陣，但是這兩種方法都各有優缺點，在經過比較後才選擇了四元數來計算座標軸的旋轉，表 2.3 為三種方法的比較表。

表 2.1 旋轉表示法比較

	優點	缺點
四元數	<ol style="list-style-type: none"> 1. 可避免萬象節鎖 2. 只需一個四維的四元數就可計算繞任一過原點的向量旋轉 	<ol style="list-style-type: none"> 1. 計算較尤拉角複雜 (多一維度) 2. 理解較為困難，不直觀
旋轉矩陣	<ol style="list-style-type: none"> 1. 旋轉軸可以是任意向量 	<ol style="list-style-type: none"> 1. 矩陣使用 16 個元素 2. 4×4 矩陣計算的複雜度上升
尤拉角	<ol style="list-style-type: none"> 1. 容易理解 2. 形象直觀 3. 表示法簡單 	<ol style="list-style-type: none"> 1. 計算需要按照固定座標軸順序 2. 會造成萬象節鎖

(資料來源: CSDN)

第三章 文獻探討

人體的運動姿態軌跡追蹤相關的研究相當多，在這部分主要分成以攝影影像做辨識以及以感測器所得的資訊做辨識兩種。由於近年來科技的發展，感測器的辨識也越來越精準，相關的研究也逐漸增加中，在軌跡重建的部分，主要利用的資訊為兩種：一種為利用加速度計所測量計算而得的加速度，以及利用陀螺儀所測量計算而得的角速度，因為感測器的發展，研究的範圍也從偵測靜態姿態到運動姿態。本章節主要介紹內容如下：3.1 介紹過去普遍使用的軌跡辨識法，3.2 慣性感測器在人體姿態辨識上的應用 3.3 介紹結合加速度計及陀螺儀分析運動姿態。

3.1 以影像為基礎的動作辨識法

在過去的經驗中，如要針對動作辨識的作法，大部分都是以攝影機當作影像輸入的來源，將輸入的影像經過一連串的影像處理步驟來獲得軌跡的資訊。一般來說，以影像為基礎的動作辨識法，都是利用多部攝影器材擺置於各個不同的角度，以取得偵測對象的全方位動作資訊。在利用解讀後的影像資訊重建人體動作模型，而此種方法的困難點在於影像資訊的複雜程度較高，尤其此方法是利用多個攝影鏡頭所取得的影像資訊，可能包含了前景深景的資訊，利用這些資訊進行人體的動作捕捉。

在謝宗倫(2010)學者提出的研究中，利用單相機去重建物體的軌跡，因使用單一相機來重建三維世界座標，不僅可降低校正與環境架設的困難、增加系統的實用性，更可以降低開發成本與時間。而在 Urtasun(2005) 等人提出的研究中，在單一視角的影像資訊下，利用預先收集大量的影像資訊，透過人工觀察取出幾個重要的關鍵畫面的資訊，並利用內插法取得相同的取樣數，再經過 PCA 去除重複的資訊降低維度，再用 Hill-Climbing Algorithm 計算人體運動姿態的參數再結合人體動態模型，來重建出 3D 高爾夫揮桿動作，避免了在複數攝影機下的高計算複雜度。在林楷歲(2010)學者的研究中，林楷歲學者先利用高斯建模方法建立背景模型，使用背景相減法擷取出影片中的移動物，並採用型態學運算消除雜訊及連通段開(Connect Component)將多個移動物個別標記。接著擷取移動物的顏色特徵及座標位置，利用 CAM Shift 於畫面中搜尋移動物特徵進行多移動物偵測

並更新其特徵。最後分別計算其重心位置並進行修補後，記錄成一連串的軌跡座標。

但是利用影像偵測的方法比較適用於低階的三維影像重建的應用，對於需要較精準的應用的需求下，例如身體細部的旋轉或扭動等等，這些資訊比較難以用攝影鏡頭偵測而得，而這種方法還有使用環境的限制，一般來說，都是在固定的場所內架設攝影機來記錄資訊，對於某些需要在室外進行的運動使用上就有困難，上述的困難點還只是屬於使用複數攝影機的情況下，若只使用單一一部攝影機，則必須事先使對於拍攝環境做資料收集，再於事後將環境資訊給移除，這方面進行移動式的動作追蹤的情況下，難度又是更上一層。

3.2 慣性感測器的應用於人體姿態辨識

隨著 MEMS 技術的發展，慣性感測器的應用，從原先只應用於車輛或是醫療器材等等或是較為昂貴的領域，而現在逐漸發展於應用在個人的姿態辨識之中。在 M. J. Mathie(2004)的研究中有個實驗，M. J. Mathie 在實驗中只有使用加速度計做遠距離看護的系統，並針對慢性病病患的日常動作進行分類，例如坐姿、站姿、平躺等等的方向，針對各種類別的姿態設定適當的加速度門檻值，用以追蹤病患的現況，當加速度值在短時間內出現急遽的變化，可以推測判斷出病患可能出現跌倒的情況，此時可以即時的通知看護人員進行協助。

D. Fontaine(2003)等學者在研究文中提出在人體的手臂或是腿部裝置加速度計，利用物理觀念，將加速度值透過兩次積分，第一次積分獲得速度，再對速度做積分取得位移量，將位移量藉由 3D 模型中顯示，追蹤人體的動作姿態。但是這種方法因為只使用了加速度計，因此對於人體的轉動並無法偵測，只能針對位移的部分做追蹤，除此之外，若對加速度計做旋轉，會導致加速度計的座標軸出現偏移，與原先座標軸不同，最後在計算位移量時會產生問題。

伍星翰(2011)學者的研究中利用加速度感測器偵測三軸方向的加速度特性，用來辨識跌倒的方向，並結合 ZigBee 定位系統讓跌倒外的發生地點可以更快被找到，以便做最迅速的救援。此研究的另一種應用為手勢移

動軌跡辨識系統，利用加速度感測器紀錄手部移動軌跡的加速度值，透過無線方式傳輸至電腦在藉由動態時間校正(dynamic time warping, DTW)演算法來辨識移動軌跡。相同的，此種方法也沒有加入陀螺儀的輔助，若有旋轉時勢必會造成軌跡繪製上的錯誤。

K. King(2004)等人在年的研究中使用以慣性感測器取代昂貴且不易攜帶的高速攝影機來偵測人體動姿態。這篇文章使用的是自製的三軸加速度計及陀螺儀，藉此偵測人體的揮桿動作特徵，但是這篇研究最後的分析結果只給予角速度的波行變化，並沒有完整的軌跡計算。

邱柏鈞(2012)學者的研究中使用了加速度計以及陀螺儀作為結合的感測器，利用網路將資訊傳輸至雲端系統，其中處理了感測器的偏差校正以及分散式感測器的時間同步問題。接著針對持續旋轉的感測器座標做處理將資料顯示於地球座標下，最後將軌跡繪出。

H. Ghasemzadeh(2011)在研究中，提到一般而言專業的運動需要投入大量的培訓。而傳統的培訓方法，例如尋求旁人協助或是找專業教練一同培訓，以及在錄影的幫助下進行訓練可以有效的提高培訓成果。然而這些技術或資訊不包括運動員的動作細部細節，因此利用加速度計及陀螺儀結合而成穿戴裝置，提供各種物理資訊，包含速度、加速度、轉動角度等等，讓運動員在矯正動作時有參考依據。

3.3 誤差校正法

Li Xiaofang(2015)等學者於研究中使用了 ZUPT 以及卡爾曼濾波器來校正誤差，ZUPT 為 Zero velocity update 的縮寫，ZUPT 是一種針對慣性導航系統的誤差校正方法，其特性為低成本且有效的控制誤差的方法，這是利用了人們在行走時，當足部與地面接觸的瞬間速度為零的特性來做判斷，在研究的結果中顯示，基於離散性卡爾曼濾波器模型的 ZUPT 方法使用慣性感測系統可以得到高度精確的定位。

Mahony(2008)學者的研究提出利用兩種非線性互補濾波器架構計算姿態角控制無人機，改善了因陀螺儀的雜訊與偏差值而產生的誤差。下圖 3.1 為 Mahony 所提出的 Passiv Complementary Filter(PCF)架構，

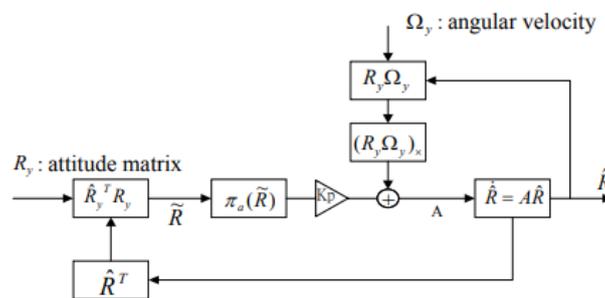


圖 3.1 Mahony 所提出的 PCF 架構

(資料來源: Mahony,2008)

Collinson(1996)書中討論有關於互補濾波器(Complementary Filter)之原理，其互補濾波器架構圖如圖 3.2 所示；此架構優點在於架構簡單，能夠有效降低程式運算量，而在整合訊號效果上也有不錯表現。

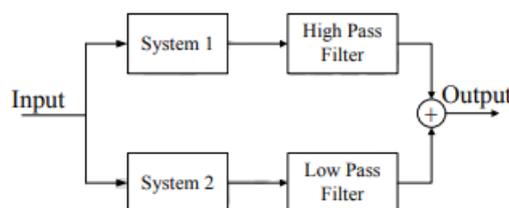


圖 3.2 Collinson 所討論之互補濾波器架構

(資料來源: Collinson, 1996)

互補濾波器之原理是把兩種不同特性的輸入訊號透過適當的濾波器做互相補償，以圖 3.2 為例: System1 之輸出特性為量測雜訊少，對輸入響應快，但會隨著時間而累積誤差；System2 之輸出特性為量測雜訊多，但長期平均精度較佳。所以在 System1 中使用高通濾波器取出高頻的部分，將低頻部分去除，System2 則是剛好相反，如此可得到響應快且雜訊少之輸出。

楊長圃(2004)於研究中利用 AHRS(Attitude and Heading Reference System)以四元數模型計算載具的姿態，並應用卡爾曼濾波器技術，整合三軸加速度及電子羅盤估算載具傾斜角度，推導機身的四元數，最後再轉至尤拉角，利用尤拉角配合直升機控制系統。在慣性導航系統上，將加速度積分兩次，計算出載具的速度及位移；由於積分計算過程中會累積誤差，導致結果逐漸偏離正確值，此研究利用卡爾曼濾波器結合 GPS，預測慣性導航系統的速度及位移誤差做回授修正。

第四章 系統架構建構

本研究透過蒐集文獻或是依照現有文獻做出選擇，挑選出適用於本研究之中的硬體設備以及資料處理方法，4.1 系統架構，4.2 硬體設備，4.3 資料分析與轉換。

4.1 系統架構

在本章節會介紹本研究所建立的感測系統，感測系統大致上可分為三大區塊，分別為感測器硬體組成、姿態數據收集以及數據分析與呈現的部分，下圖 4.1 為本研究的感測系統之系統架構圖。

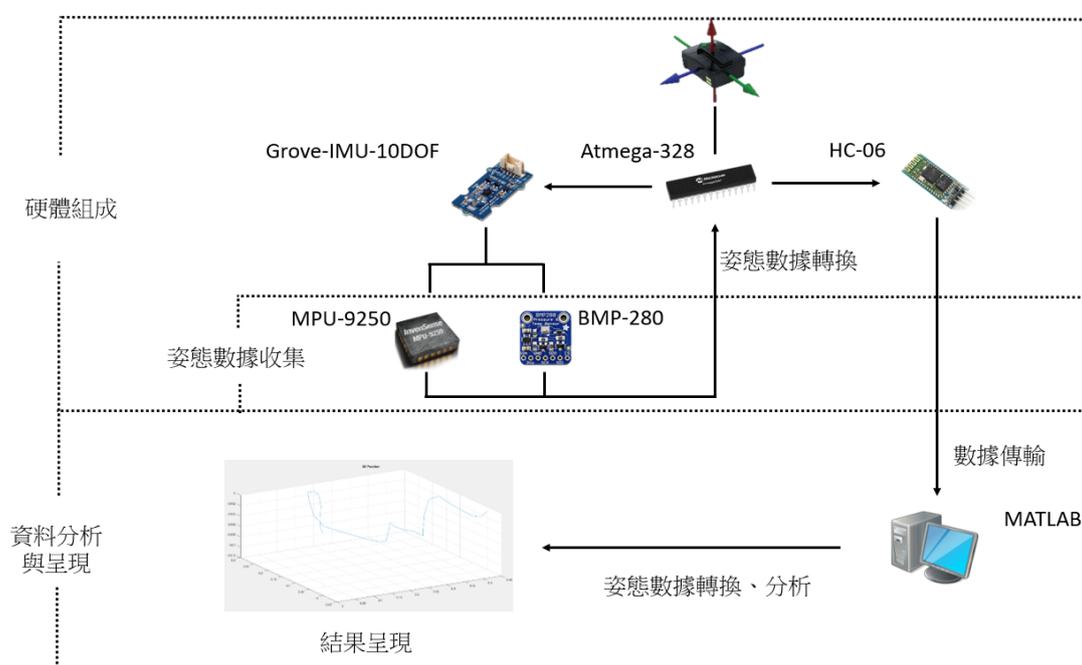


圖 4.1 系統架構圖

4.1.1 硬體組成

在硬體組成的部分，我們使用 ATmega-328 微控制器來控制整個感測系統，ATmega328 控制 Grove-IMU-10DOF 以及 HC-06 兩塊晶片，Grove-IMU-10DOF 晶片內包含了 MPU-9250 及 BMP-280 晶片，MPU-9250 晶片主要偵測感測物體的慣性感測資料，包含了感測物體的三軸加速度、三軸的角速度以及三軸的磁力計，加速度、角速度及磁場強度是構成後續 AHRS 姿態解算法的重要元素；而 BMP-280 晶片則是測量物體位置的大氣壓強度，用以偵測物體的海拔高度，在本研究中暫不需要此功能，若有需要更精準的校正，則可以以大氣壓力換算海拔高度，驗證地球座標軸下的 Z 軸位移；HC-06 則是負責將 ATmega-328 晶片所處理完之姿態數據傳輸至電腦。

4.1.2 姿態數據收集

姿態數據收集的部分，我們目的是建立一套能夠重建軌跡的系統，因此本研究需要收集感測物體的運動姿態，重建軌跡的原理可以利用將加速度值做二次積分可以得到位移的原理下手，將感測時間點的姿態數據轉換成該時間點的位置，最後繪圖而出；但這是在不包含轉動的情況下的運動軌跡，如果有轉動的產生，勢必會對軌跡上的轉換造成影響，因此本研究也須要偵測感測物體的轉動角度，以求更精準的運動軌跡；偵測物體的該地地磁強度則是因為陀螺儀的 Z 軸與地球座標軸的 Z 軸重疊，在做轉換時會造成無法計算出正確的 Z 軸數值，因此需要使用磁力計來做較正，使得所得結果更為精確。

4.1.3 資料分析與呈現

在取得運動姿態的相關數據之後，我們將資料匯入於 MATLAB 之中，用 MATLAB 做資料分析，將加速度轉換成速度及軌跡，最後繪圖出來，將軌跡及其他運動姿態資料顯示於屏幕中，提供運動員做為參考。

4.2 硬體設備

4.2.1 ATmega-328

ATmega-328 為 Atmel 發展的 megaAVR 產品線其中一個單晶片產品。這是一顆八位元 AVR 精簡指令型微控制器，是基於 RISC 的高性能設備，結合了 32KB ISP 閃存記憶與同時讀寫功能、1KB EEPROM、2KB SRAM、23 條通用 I/O 電纜、32 個工作寄存器、串行可編程 USART 等。Atmel ATmega328 微控制器可在一個時鐘週期內執行功能強大的指令，使設備能夠實現接近 1 MIPS 每 MHz 的吞吐量，同時平衡功耗與處理速度。根據設計，Atmel 微控制器可用於自動化與家庭及建築自動化。下圖 4.2 為 ATmega-328 晶片腳位功能對照圖，根據需求對應該腳位做宣告，控制輸出或是輸入格式等等。

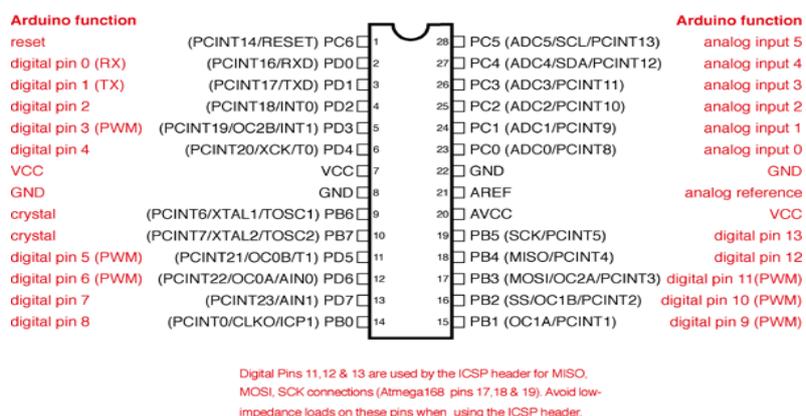


圖 4.2 ATmega-328 晶片腳位功能對照

在本研究中利用 ATmega-328 微控制器來控制 Grove-IMU-10DOF 偵測物體的運動姿態，事先於 Arduino 中撰寫運動姿態數據的轉換程式，再將程式燒錄於 ATmega-328 中，代替電腦執行程式。

4.2.2 Grove-IMU-10DOF

Grove-IMU-10DOF 是基於 MPU-9250 和 BMP-280，MPU-9250 仍然是最兼容的 9 軸運動跟踪設備，它結合了 3 軸陀螺儀，3 軸加速度計，3 軸磁力計和數字運動處理器（DMP），採用 3×3×1mm 封裝。關於 BMP280，從 Grove- IMU -10DOF 的主要變化，這次它被包裝在比 BMP180 小 65% 的非常緊湊的包裝。不僅尺寸小，功耗更低，低至 2.74uA@1Hz。

Grove- IMU-10DOF 通過 MPU-9250 和 BMP-280 的完美結合，Grove-IMU- 10DOF 可以為您提供總共 10 個數據軸，允許您構建移動項目，如智能手機，可穿戴設備，Quadrotor，甚至可以構建像 PSP 這樣的操縱桿控制器和 Xbox。

Grove-IMU-10DOF 共有 4 支 PIN 腳，分別為 VCC、GND、SDA、SCL，下表 4.1 為 Grove- IMU-10DOF 與 ATmega-328 所連接對應的對照表。VCC 與 5V 連接輸入電源，GND 連接 ATmega-328 的 GND 進行接地，而 SDA 與 SCL 則是代表著串列資料與串列時脈，連接至 A4 與 A5，由 A4 與 A5 接收類比訊號，在 ATmega-328 中進行數據處理。

表 4.1 Grove- IMU-10DOF 與 ATmega-328 連接對照表

Grove- IMU-10DOF	ATmega-328
VCC	5V
GND	GND
SDA	A4
SCL	A5

4.2.3 MPU-9250

原先本系統所使用的感測晶片為 MPU-6050，MPU-6050 是一種非常普遍使用的三維空間姿態傳感晶片，可以用來獲取三軸的加速度及角速度。但 MPU-6050 所輸出的數據有比較多的雜訊(噪聲)，若不進行濾波將會對整個系統的精準度造成嚴重影響，因此，本研究尋找另一塊感測晶片來取代 MPU-6050。

本研究最後找到由同一家公司所生產的晶片 MPU-9250，MPU-9250 是 InvenSense 第二代九軸 Motion Tracking 元件，適用於智慧型手機、平板裝置、穿戴式感測器以及其他消費性應用。MPU-9250 採用 3×3×1 mm QFN 封裝，是全球最小的九軸 Motion Tracking 元件，包含了三軸加速度計、三軸陀螺儀以及三軸數位羅盤。

MPU-9250 的三軸加速度計提供了 4 種不同的靈敏度範圍供使用者依照使用需求做調整，加速度計的靈敏度從 ± 2 到 ± 16 ，若要輸出成單位為 g 值的輸出數據，必須在 ATmega-328 中寫入公式 2.2 將 MPU-9250 輸出的 ADC 值轉換成 g 值，下表 4.1 為 MPU-9250 的加速度計靈敏度對照表，根據使用者需求之 g 值範圍，對照相對應的靈敏度數值與 conditions，而 conditions 的部分需要事先在 MPU-9250 的 c++檔中更改一行程式碼，先找出 void MPU9250 的初始化函式，再去更改加速度計所對應到的程式碼，MPU-9250 初始化函式如下：

```
void MPU9250::initialize()
{
    setClockSource(MPU9250_CLOCK_PLL_XGYRO);
    setFullScaleGyroRange(MPU9250_GYRO_FS_250);
    setFullScaleAccelRange(MPU9250_ACCEL_FS_2);
    setSleepEnabled(false);
}
```

若要更改加速度計的感測範圍，只需更改初始化函式中的 setFullScaleAccelRange 後面括號中的值，FS_2 所代表的是 $\pm 2g$ ，若想改成 $\pm 4g$ 的靈敏度，只需將 FS_2 更改成 FS_4，其他的靈敏度範圍依此類推。

表 4.2 MPU-9250 加速度計靈敏度對照表

conditions	g-range	sensitivity
AFS_SEL=0	±2	16384
AFS_SEL=1	±4	8192
AFS_SEL=2	±8	4096
AFS_SEL=3	±16	2048

(資料來源: MPU-9250 datasheet)

MPU-9250 的陀螺儀也與加速度一樣，提供了 4 種不同的靈敏度範圍供使用者按照需求做調整，而陀螺儀的靈敏度範圍可以從±250 到±2000做調整，與加速度計相同的，MPU-9250 的原始輸出數據為 ADC 值，需要經過轉換才能換成以 deg/s 為單位的輸出值，必須在 ATmega-328 中寫入公式 2.3 將 ADC 轉為角速度的輸出值，下表 4.2 為 MPU-9250 的陀螺儀靈敏度對照表，根據使用者需求之角速度值範圍，對照相對應的靈敏度數值與 conditions，而 conditions 的部分需要事先在 MPU-9250 的 c++檔中更改一行程式碼，先找出 void MPU9250 的初始化函式，再去更改陀螺儀所對應到的程式碼，MPU-9250 初始化函式如下：

```
void MPU9250::initialize()
{
    setClockSource(MPU9250_CLOCK_PLL_XGYRO);
    setFullScaleGyroRange(MPU9250_GYRO_FS_250);
    setFullScaleAccelRange(MPU9250_ACCEL_FS_2);
    setSleepEnabled(false);
}
```

若要更改加速度計的感測範圍，只需更改初始化函式中的 setFullScaleGyroRange 後面括號中的值，FS_250 所代表的是±250deg/s，若想改成±500的靈敏度，只需將 FS_250 更改程 FS_500，其他的靈敏度範圍依此類推。

表 4.3 MPU-9250 陀螺儀靈敏度對照表

conditions	degree-range	sensitivity
FS_SEL=0	± 250	131
FS_SEL=1	± 500	65.5
FS_SEL=2	± 1000	32.8
FS_SEL=3	± 2000	16.4

(資料來源: MPU-9250 datasheet)

MPU-9250 最後一組感測元為磁力計，但磁力計與上述兩種感測器有所不同，磁力計在這邊只有一種感測範圍，因此不需要另外做調整，只需了解在 MPU-9250 中如何將磁力計部分的 ADC 輸出值轉換成以奈特(nT)為單位的輸出值。因磁力計只有一組靈敏度，因此不需要另外撰寫程式調整磁力計的靈敏度範圍，只需撰寫轉換 ADC 值的程式碼，下表為 MPU-9250 磁力計的靈敏度表。

表 4.4 MPU-9250 磁力計靈敏度表

	Full-Scale Range	Sensitivity
Magnetometer	± 4800	0.6

(資料來源: MPU-9250 datasheet)

4.2.4 HC-06

本研究所設計的感測系統主要針對運動員使用，為了儘可能不影響到運動員的表現，我們決定使用藍芽傳輸設備，原先於 WIFI 與藍芽兩者之間考據，由於在運動場之中，WIFI 可能會受到環境影響以及 WIFI 的耗電量大，導致必須為了藍芽配置更大容量的電池，而對於裝置於運動員身上的感測器來說則是越小越好，因此本研究最後選擇藍芽傳輸書據。

HC-06 共有 4 支 PIN 腳，分別為 VCC、GND、TX、RX，下表 4.5 為 HC-06 與 ATmega328 所連接對應的腳位。VCC 與 5V 連接輸入電源，GND 與 ATmega-328 的 GND 連接進行接地，TX 與 RX 分別為資料傳輸與資料接收，在 HC-06 與 ATmega-328 中都有配置，一般來說，正常通訊時 HC-06 都是連接其他設備的 TX，若 HC-06 要接收自己本身所傳送出來的資料則是 TX 與 RX 進連接，通常這方法都是用來檢測 HC-06 是否正常。在本研究中的數據是由 ATmega-328 中處理完，藉由 HC-06 進行傳輸，因此要由 ATmega-328 的 TX(傳送)連接 HC-06 的 RX(接收)。

表 4.5 HC-06 與 ATmega-328 連接腳位對照表

HC-06	ATmega-328
VCC	5V
GND	GND
TX	RX
RX	TX

(資料來源:本研究整理)

4.3 姿態數據分析校正與轉換

本研究的目的是為利用 IMU 感測系統建構出一套軌跡重建系統，因此本研究需要由 IMU 感測器中獲得原始資料進行換算，在本章節中會介紹本研究如何去換算這些姿態數據，以及相關數據的誤差校正方法。下圖 4.3 為姿態數據的分析流程圖。

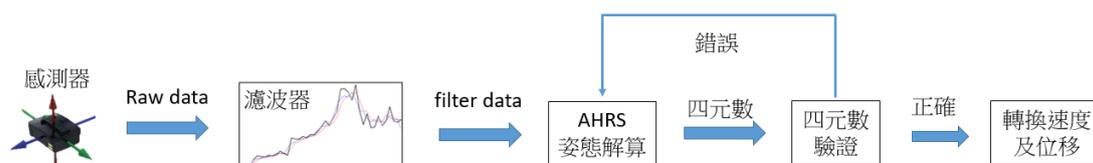


圖 4.3 姿態數據分析流程圖

4.3.1 加速度計與陀螺儀飄移校正

不管是 MPU-6050 或是 MPU-9250，任何一塊慣性感測元件都因製造過程的關係，每塊晶片都會有自本身的漂移值，也就是晶片在靜止狀態下，加速度計的三軸中除了 Z 軸之外的值都應該為 0，Z 軸的加速度因受到地心引力的影響，將會有個 1g 的值存在，若 Z 軸測得的數值不為 1g，則該軸也有飄移值的存在；在陀螺儀中，三軸的轉動角度都為 0，而有飄移值的存在的話，數值則必定不為 0，變化將會以飄移值為原點出發，按照感測物體狀態做變化，因此必須在換算其他姿態數據前先將飄移值消除，否則最後呈現出來的結果必定會與現實情況有所落差。以下程式碼為在本研究中撰寫於 ATmega-328 的程式碼，該程式碼為本研究中如何去自動消除飄移值的方法。

```
unsigned short times=50;
for(int i=0;i<times;i++)
{
    accelgyro.getMotion9(&ax, &ay, &az, &gx, &gy, &gz, &mx, &my, &mz);
    axo += ax; ayo += ay; azo += az;
    gx0 += gx; gy0 += gy; gzo += gz;
```

$$\begin{aligned}
& \} \\
& axo /= times; ayo /= times; azo /= times; \\
& gxo /= times; gyo /= times; gzo /= times;
\end{aligned} \tag{4.1}$$

在啟動系統後，系統會自動記錄前 50 筆由 MPU-6050 測得的 ADC 輸出資料，並且自動加總，記錄完 50 筆 ADC 輸出資料後，按照公式 4.1 將加速度計三軸及陀螺儀三軸共六軸的各自加總值做平均，平均後的值就是 MPU-9250 在靜止狀態下的飄移值。計算完飄移值後，在每次計算三軸加速度或是陀螺儀之前都必須先扣掉該軸的飄移值，扣除完畢後在執行公式 2.2 或公式 2.3 計算出該軸的加速度 g 值及轉動角度，而加速度計的 Z 軸扣除完畢後必須加上受地心引力引響的 g 值，因此在 Z 軸的部分要另外加上 0.981，經過消除飄移值的感測資料才有辦法進行下一步驟的姿態解算。

4.3.2 低通濾波器進行濾波

現今也有許多軌跡重建系統，大部分的系統都是建構對於即時的偵測系統，在系統中對於加速度計及陀螺儀的雜訊濾波都使用卡爾曼濾波器或是擴展式卡爾曼濾波器來進行濾波，但卡爾曼濾波器主要用於預測及更新，他需要的是一段原始資料來進行預測和更新，對於系統來說，若短時間內有較大的變化值出現的話，將會造成卡爾曼濾波器的反應沒有那麼即時，更新的速率可能會跟不上變化的速度，因此在本研究中的做法為，收集一個完整動作的資訊，將所有資訊進行低通濾波，用低通濾波的特性，將雜訊波進行濾除，保留下有用的資訊，下圖 4.4 為未濾波及進行濾波的比較圖。

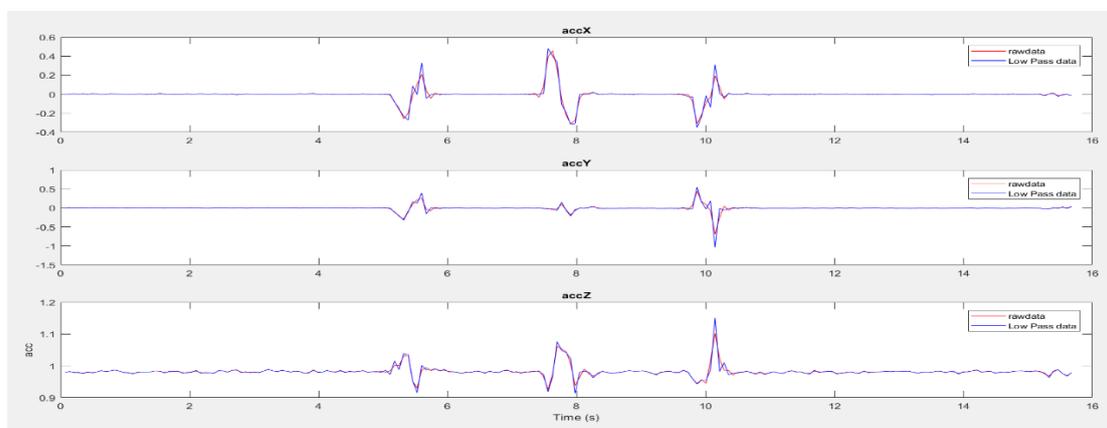


圖 4.4 濾波前後比較圖

4.3.3 AHRS 姿態解算數據換算

AHRS 是一種姿態解算的方法，而 AHRS 是自動航向基準系統 (Automatic Heading Reference System) 的簡稱。而 AHRS 計算的方式是基於四元數為基礎，目前使用四元數來進行 AHRS 姿態解算的算法被廣泛採用於四軸飛行器上。該算法源自英國 Bristol 大學的 Ph.D Sebastian Madgwick，他在 2009 年開發並發布了該算法。下面將真對本研究所使用的 AHRS 程式碼進行詳細介紹。

在 AHRS 姿態解算法之中，我們會使用到加速度、角速度、磁力計來進行運算，而加速度計除了自身的物理性質可以轉換成速度及位移之外，還可以計算出該軸的傾角，以下舉一個簡單例子：假設一個單軸的加速度計位於重力水平面上的時候，在垂直水平面上所受到的加速度為 $1g$ ，而在平行於水平面上所受到的加速度為 0 。因此當我們旋轉一個角度時，會在水平軸上產生一加速度的分量，透過他們的關係可以計算出該軸加速度計的傾角，如下圖 4.5 所示。

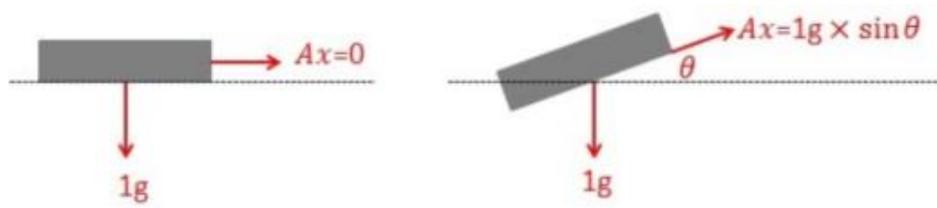


圖 4.5 加速度計傾角計算

(資料來源: 圖點博士)

以下 MadgwickAHRUpdateIMU 函式為沒有磁力計的情況下進行的 AHRS 姿態解算法，若有磁力計數值則進行下方的 MadgwickAHRUpdate 函式。

```
void MadgwickAHRUpdateIMU(float ax, float ay, float az, float gx, float gy, float gz)
{
    float recipNorm;
    float s0, s1, s2, s3;
```

```

float qDot1, qDot2, qDot3, qDot4;
float _2q0, _2q1, _2q2, _2q3, _4q0, _4q1, _4q2, _8q1,;
float _8q2, q0q0, q1q1, q2q2, q3q3;

qDot1 = 0.5f * (-q1 * gx - q2 * gy - q3 * gz);
qDot2 = 0.5f * (q0 * gx + q2 * gz - q3 * gy);
qDot3 = 0.5f * (q0 * gy - q1 * gz + q3 * gx);
qDot4 = 0.5f * (q0 * gz + q1 * gy - q2 * gx);          (4.2)

if(!((ax == 0.0f) && (ay == 0.0f) && (az == 0.0f))) {

```

先導入慣性感測器偵測而得的數據，由公式 4.2 計算出由陀螺儀所算出的四元數變化率，並檢視各軸的角速度值是否為 0。否則在接下來將加速度計進行歸一化處理的時候，會產生除以 0 的邏輯錯誤。

```

    recipNorm = invSqrt(ax * ax + ay * ay + az * az);
    ax *= recipNorm;
    ay *= recipNorm;
    az *= recipNorm;

```

將加速度計的數據進行歸一化處理。其中 `invSqrt` 是平方根的倒數，使用平方根的倒數而不是直接使用平方根的原因是使下面的 `ax`，`ay`，`az` 的運算速度更快。通過歸一化處理後，`ax`，`ay`，`az` 的數值範圍將變成 -1 到 +1 之間。

```

    _2q0 = 2.0f * q0;
    _2q1 = 2.0f * q1;
    _2q2 = 2.0f * q2;
    _2q3 = 2.0f * q3;
    _4q0 = 4.0f * q0;

```

```

_4q1 = 4.0f * q1;
_4q2 = 4.0f * q2;
_8q1 = 8.0f * q1;
_8q2 = 8.0f * q2;
q0q0 = q0 * q0;
q1q1 = q1 * q1;
q2q2 = q2 * q2;
q3q3 = q3 * q3;

```

宣告上述各式變數名稱的計算方式，以節省以下計算所需時間。

```

s0 = _4q0 * q2q2 + _2q2 * ax + _4q0 * q1q1 - _2q1 * ay;
s1 = _4q1 * q3q3 - _2q3 * ax + 4.0f * q0q0 * q1 - _2q0 * ay - _4q1 + _8q1
    *
    q1q1 + _8q1 * q2q2 + _4q1 * az;
s2 = 4.0f * q0q0 * q2 + _2q0 * ax + _4q2 * q3q3 - _2q3 * ay - _4q2 +
    _8q2 *
    q1q1 + _8q2 * q2q2 + _4q2 * az;
s3 = 4.0f * q1q1 * q3 - _2q1 * ax + 4.0f * q2q2 * q3 - _2q2 * ay;    (4.3)

```

由公式 4.3 計算出利用梯度下降法配合加速度值算出修正值，用以後續要修正由陀螺儀所算出的四元數變化率。

```

recipNorm = invSqrt(s0 * s0 + s1 * s1 + s2 * s2 + s3 * s3);
s0 *= recipNorm;
s1 *= recipNorm;
s2 *= recipNorm;
s3 *= recipNorm;
qDot1 -= beta * s0;
qDot2 -= beta * s1;
qDot3 -= beta * s2;

```

```

    qDot4 -= beta * s3;
}

```

(4.4)

公式 4.4 為將先前計算出來的四元數變化率進行更新，由四元數變化率扣除由公式 4.3 所算出的修正值乘以 beta(積分比例增益)。

```

q0 += qDot1 * (1.0f / sampleFreq);
q1 += qDot2 * (1.0f / sampleFreq);
q2 += qDot3 * (1.0f / sampleFreq);
q3 += qDot4 * (1.0f / sampleFreq);
recipNorm = invSqrt(q0 * q0 + q1 * q1 + q2 * q2 + q3 * q3);
q0 *= recipNorm;
q1 *= recipNorm;
q2 *= recipNorm;
q3 *= recipNorm;
}

```

(4.5)

在這步驟為四元數值的更新，公式 4.5 原始四元數加上將上述計算出的四元數變化率做積分而得的值。更新完四元數後須再進行一次歸一化，求出真正四元數。

MadgwickAHRSupdate 為同時具有加速度值、角速度值以及磁力計數值的情況下所進行的函式。

```

void MadgwickAHRSupdate(float ax, float ay, float az, float gx, float gy, float gz, float mx, float my, float mz)
{
    float recipNorm;
    float s0, s1, s2, s3;
    float qDot1, qDot2, qDot3, qDot4;
    float hx, hy;
    float _2q0mx, _2q0my, _2q0mz, _2q1mx;
    float _2bx, _2bz, _4bx, _4bz, _2q0, _2q1;
    float _2q2, _2q3, _2q0q2, _2q2q3, q0q0, q0q1, q0q2, q0q3, q1q1, q1q2;

```

```

float q1q3,q2q2, q2q3, q3q3;

if((mx == 0.0f) && (my == 0.0f) && (mz == 0.0f))
{
    MadgwickAHRSupdateIMU(ax, ay, az, gx, gy, gz);
    return;
}

```

先導入陀螺儀偵測而得的數據，判斷出三軸磁力計是否為 0，若為 0，在這裡可以當成沒有磁力計的 AHRS，因此直接呼叫 MadgwickAHRSupdateIMU，進行無磁力計的 AHRS 姿態解算。

```

qDot1 = 0.5f * (-q1 * gx - q2 * gy - q3 * gz);
qDot2 = 0.5f * (q0 * gx + q2 * gz - q3 * gy);
qDot3 = 0.5f * (q0 * gy - q1 * gz + q3 * gx);
qDot4 = 0.5f * (q0 * gz + q1 * gy - q2 * gx);

```

```

if(!((ax == 0.0f) && (ay == 0.0f) && (az == 0.0f)))

```

導入慣性感測器偵測而得的數據，此處的四元數變化率算法與 MadgwickAHRSupdateIMU 相同，因此由公式 4.2 計算出由陀螺儀所算出的四元數變化率，並檢視各軸的價速度值是否為 0。否則在接下來將加速度計進行歸一化處理的時候，會產生除以 0 的邏輯錯誤，磁力計在先前已檢測過，因此在這不需在檢驗一次。

```

{
    recipNorm = invSqrt(ax * ax + ay * ay + az * az);
    ax *= recipNorm;
    ay *= recipNorm;
    az *= recipNorm;
    recipNorm = invSqrt(mx * mx + my * my + mz * mz);
    mx *= recipNorm;
    my *= recipNorm;
    mz *= recipNorm;
}

```

將加速度計的數據與磁力計的數據進行歸一化處理。其中 invSqrt 是平方根的倒數，使用平方根的倒數而不是直接使用平方根的原因是使下面的 a_x 、 a_y 、 a_z 、 m_x 、 m_y 、 m_z 的運算速度更快。通過歸一化處理後， a_x 、 a_y 、 a_z 、 m_x 、 m_y 、 m_z 的數值範圍將變成-1 到+1 之間。

$$_2q0mx = 2.0f * q0 * mx;$$

$$_2q0my = 2.0f * q0 * my;$$

$$_2q0mz = 2.0f * q0 * mz;$$

$$_2q1mx = 2.0f * q1 * mx;$$

$$_2q0 = 2.0f * q0;$$

$$_2q1 = 2.0f * q1;$$

$$_2q2 = 2.0f * q2;$$

$$_2q3 = 2.0f * q3;$$

$$_2q0q2 = 2.0f * q0 * q2;$$

$$_2q2q3 = 2.0f * q2 * q3;$$

$$q0q0 = q0 * q0;$$

$$q0q1 = q0 * q1;$$

$$q0q2 = q0 * q2;$$

$$q0q3 = q0 * q3;$$

$$q1q1 = q1 * q1;$$

$$q1q2 = q1 * q2;$$

$$q1q3 = q1 * q3;$$

$$q2q2 = q2 * q2;$$

$$q2q3 = q2 * q3;$$

$$q3q3 = q3 * q3;$$

上述變數宣告為定義該變數的計算方法，避免以下需要大量的重複運算時造成運算量過大。

$$hx = mx * q0q0 - _2q0my * q3 + _2q0mz * q2 + mx * q1q1 + _2q1 * my * q2 + _2q1 * mz * q3 - mx * q2q2 - mx * q3q3;$$

$$hy = _2q0mx * q3 + my * q0q0 - _2q0mz * q1 + _2q1mx * q2 - my * q1q1 + my * q2q2 + _2q2 * mz * q3 - my * q3q3;$$

$$_2bx = \text{sqrt}(hx * hx + hy * hy);$$

$$_2bz = -_2q0mx * q2 + _2q0my * q1 + mz * q0q0 + _2q1mx * q3 - mz * q1q1 + _2q2 * my * q3 - mz * q2q2 + mz * q3q3;$$

$$_4bx = 2.0f * _2bx;$$

$$_4bz = 2.0f * _2bz;$$

在這部分將磁力計的偵測數據加入運算，計算出以地球磁場為依據的參考方向，而這方向後續計算中還會繼續使用。

$$s0 = -_2q2 * (2.0f * q1q3 - _2q0q2 - ax) + _2q1 * (2.0f * q0q1 + _2q2q3 - ay) - _2bz * q2 * (_2bx * (0.5f - q2q2 - q3q3) + _2bz * (q1q3 - q0q2) - mx) + (-_2bx * q3 + _2bz * q1) * (_2bx * (q1q2 - q0q3) + _2bz * (q0q1 + q2q3) - my) + _2bx * q2 * (_2bx * (q0q2 + q1q3) + _2bz * (0.5f - q1q1 - q2q2) - mz);$$

$$s1 = _2q3 * (2.0f * q1q3 - _2q0q2 - ax) + _2q0 * (2.0f * q0q1 + _2q2q3 - ay) - 4.0f * q1 * (1 - 2.0f * q1q1 - 2.0f * q2q2 - az) + _2bz * q3 * (_2bx * (0.5f - q2q2 - q3q3) + _2bz * (q1q3 - q0q2) - mx) + (_2bx * q2 + _2bz * q0) * (_2bx * (q1q2 - q0q3) + _2bz * (q0q1 + q2q3) - my) + (_2bx * q3 - _4bz * q1) * (_2bx * (q0q2 + q1q3) + _2bz * (0.5f - q1q1 - q2q2) - mz);$$

$$s2 = -_2q0 * (2.0f * q1q3 - _2q0q2 - ax) + _2q3 * (2.0f * q0q1 + _2q2q3 - ay) - 4.0f * q2 * (1 - 2.0f * q1q1 - 2.0f * q2q2 - az) + (-_4bx * q2 - _2bz * q0) * (_2bx * (0.5f - q2q2 - q3q3) + _2bz * (q1q3 - q0q2) - mx) + (_2bx * q1 + _2bz * q3) * (_2bx * (q1q2 - q0q3) + _2bz * (q0q1 + q2q3) - my) + (_2bx * q0 - _4bz * q2) * (_2bx * (q0q2 + q1q3) + _2bz * (0.5f - q1q1 - q2q2) - mz);$$

$$s3 = _2q1 * (2.0f * q1q3 - _2q0q2 - ax) + _2q2 * (2.0f * q0q1 + _2q2q3 - ay) + (-_4bx * q3 + _2bz * q1) * (_2bx * (0.5f - q2q2 - q3q3) + _2bz$$

```

* (q1q3 - q0q2) - mx) + (-_2bx * q0 + _2bz * q2) * (_2bx * (q1q2 -
q0q3) + _2bz * (q0q1 + q2q3) - my) + _2bx * q1 * (_2bx * (q0q2 +
q1q3) + _2bz * (0.5f - q1q1 - q2q2) - mz);

```

此處的用意與 MadgwickAHRSupdateIMU 的 S0、S1、S2、S3 相同，但計算方法不太一樣，此處的修正值不僅僅使用加速度，在這裡也包含了三軸的磁力計數據，利用兩種感測資料配合梯度下降法算出修正值，用以後續要修正由陀螺儀所算出的四元數變化率。

```

recipNorm = invSqrt(s0 * s0 + s1 * s1 + s2 * s2 + s3 * s3);
s0 *= recipNorm;
s1 *= recipNorm;
s2 *= recipNorm;
s3 *= recipNorm;

qDot1 -= beta * s0;
qDot2 -= beta * s1;
qDot3 -= beta * s2;
qDot4 -= beta * s3;
}

```

此處與 MadgwickAHRSupdateIMU 相同，將先前計算出來的四元數變化率進行更新，由四元數變化率扣除由公式 4.3 所算出的修正值乘以 beta(積分比例增益)。

```

q0 += qDot1 * (1.0f / sampleFreq);
q1 += qDot2 * (1.0f / sampleFreq);
q2 += qDot3 * (1.0f / sampleFreq);
q3 += qDot4 * (1.0f / sampleFreq);
recipNorm = invSqrt(q0 * q0 + q1 * q1 + q2 * q2 + q3 * q3);
q0 *= recipNorm;
q1 *= recipNorm;
q2 *= recipNorm;

```

```

q3 *= recipNorm;
}

```

此步驟由更新後的四元數變化率進行四元數的更新，原始四元數加上四元數變化率做積分而得的值。更新完四元數後須再進行一次歸一化，求出真正四元數。

4.3.4 四元數驗證

經由 AHRS 姿態解算法計算出代表受測物體的姿態四元數後，接著我們必須驗證所得的四元數是否正確，若正確代表四元數沒問題可以繼續做轉換的步驟，若出現問題，則必須回到 AHRS 之中，檢查是否有出現錯誤，或是那些參數需要更動。

本研究驗證四元數的方法為利用四元數計算出尤拉角(Euler Angle)，若由四元數計算而得的尤拉角角度正確，則四元數即為正確的代表受測物體的運動姿態。尤拉角代表的是三軸 X、Y、Z 軸所旋轉的角度，而 X、Y、Z 軸所旋轉的角度所對應到的尤拉角名稱為 Roll、Pitch、Yaw，下圖 4.6 為尤拉角與 X、Y、Z 軸的示意圖。

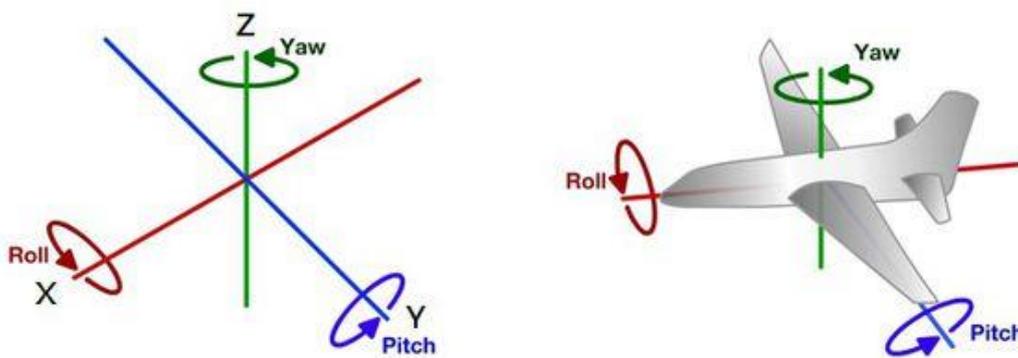


圖 4.6 尤拉角示意圖

(資料來源: STARLINO,2009)

為了驗證四元數是否正確，本研究利用四元數來計算尤拉角，四元數與尤拉角之間有轉換公式，本研究將公式撰寫於 ATmega-328 中，並將結果顯示出來方便驗證，下面公式 4.6 為四元數轉換尤拉角的公式。

$$\text{roll} = (\text{atan2f}(q_0 * q_1 + q_2 * q_3, 0.5f - q_1 * q_1 - q_2 * q_2)) * \text{RAD};$$

$$\text{pitch} = (\text{asinf}(-2.0f * (q_1 * q_3 - q_0 * q_2))) * \text{RAD};$$

$$\text{yaw} = (\text{atan2f}(q_1*q_2 + q_0*q_3, 0.5f - q_2*q_2 - q_3*q_3))*\text{RAD}; \quad (4.6)$$

由四元數所轉換出來的尤拉角原先為徑度，因此我們必須轉換成度數，在程式的開頭，本研究已先定義變數 RAD 為徑度轉度數的公式，因此在這裡只需乘上 RAD 即可。

4.3.5 地球坐標軸轉換

此章節要介紹的是本研究用來轉換地球座標軸的方法，此方法是基於四元數的計算，因此在這裡要先講解四元數的基本計算定義。先假設有兩組四元數，下面計算定義都用這兩組四元數做運算。

$$q_1 = w_1 + x_1i + y_1j + z_1k$$

$$q_2 = w_2 + x_2i + y_2j + z_2k$$

四元數的加法定義：

$$q_1+q_2 = (w_1+w_2) + (x_1+x_2)i + (y_1+y_2)j + (z_1+z_2)k$$

四元數的乘法定義(利用分配律)：

$$\begin{aligned} q_1*q_2 = & (w_1*w_2 - x_1*x_2 - y_1*y_2 - z_1*z_2) + \\ & (w_1*x_2 + x_1*w_2 + y_1*z_2 - z_1*y_2)i + \\ & (w_1*y_2 - x_1*z_2 + y_1*w_2 + z_1*x_2)j + \\ & (w_1*z_2 + x_1*y_2 - y_1*x_2 + z_1*w_2)k \end{aligned}$$

由於 $q = w + xi + yj + zk$ 可以分為純量 w 與向量 $xi + yj + zk$ ，所以為了方便表示，將 q 表示為 (S,V) ，其中 S 表示純量 w ， V 表示向量 $xi + yj + zk$ ，所以乘法又可以表示為：

$$q_1 * q_2 = (S_1 + V_1)*(S_2 + V_2) = S_1*S_2 - V_1.V_2 + V_1 \times V_2 + S_1*V_2 + S_2*V_1$$

其中 $V_1.V_2$ 表示為兩者向量的內積， $V_1 \times V_2$ 表示為兩向量的外積

接著要介紹如何使用四元數表示三維空間的任意旋轉，假設有一向量 $P(X,Y,Z)$ 對著一單位四元數 q 做旋轉，則將 P 視為無純量的四元數 $Xi+Yj+Zk$ ，則向量的旋轉公式為下：

$$\text{Rot}(P) = qPq^* \quad (4.7)$$

公式 4.7 中 q^* 為四元數 q 的共軛(Conjugate)，即為 $q^* = w - xi - yj - zk$

了解以四元數為基礎的向量旋轉公式後，接著要將公式實際帶入計算之中， $Rot(P)$ 為本研究想要得到的結果，即為經轉換成地球座標軸後的加速度向量， P 為本研究所使用的感測器測得的三軸加速度，將他組合成一向量，然而三軸加速度只有 3 個數值，而四元數有 4 個數值，照上面所介紹的，將 P 視為無純量的四元數，因此可以將 P 表示為 $0+xi+yj+zk$ ，轉換後的 $Rot(P)$ 也是相同的表示方法 $0+X'i+Y'j+Z'k$ ，最後取值只需要取 X' 、 Y' 、 Z' 即可， X' 、 Y' 、 Z' 即為轉換後加速度，再由 X' 、 Y' 、 Z' 計算出感測物體的速度及位移。

4.3.6 軌跡計算

經過上述的座標軸轉換之後，必須先將 Z 軸受到重力加速度的影響給扣除，因此將 Z 軸的加速度值減去 $1g$ ，再將加速度進行一次積分得到速度，再經過一次積分得到位移，利用位移來重建軌跡。假設起始位置的位移量及速度皆為 0，而 Δt 為每一筆取樣資料的時間差，再本研究中 Δt 固定為 0.07。以下公式 4.8 及公式 4.9 為我們所知的加速度轉換公式。

$$v_i = v_{i-1} + a_i \Delta t \quad (4.8)$$

$$s_i = d_{i-1} + v_{i-1} \Delta t + \frac{1}{2} a_i \Delta t^2 \quad (4.9)$$

在本研究中，對於速度及積分的算法稍作修改，以下用圖 4.7 來做說明。

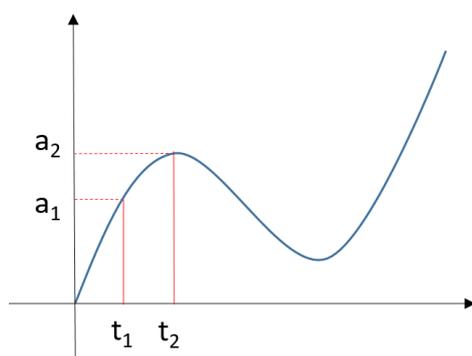


圖 4.7 加速度積分法

由上圖 4.7 得知在 t_1 時所對應的加速度為 a_1 ，當時間為 t_2 時所對應的加速度為 a_2 ，若是速度由公式 4.8 計算而得的，則該速度值為圖 4.8 灰色斜線區塊面積，依照公式 4.8 的算法將會有一部分的面積多計算，如圖 4.9 紫色區塊。因此在本研究中所使用的積分法有所不同如公式 4.10，本研究將

連續兩個時間點的加速度做平均，以平均的加速度(a')計算出面積如圖 4.10 斜線部分面積所示，若以此方法計算加速度積分，則將會有兩塊略為相同的區塊做為抵銷如圖 4.11 紫色區塊所示，誤差大小相較於公式 4.8 還要來得小。

$$v_i = v_{i-1} + \frac{a_i + a_{i-1}}{2} \Delta t \quad (4.10)$$

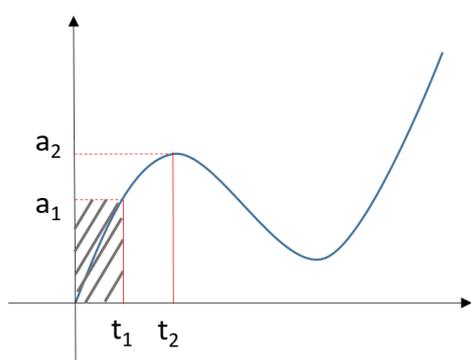


圖 4.8 公式 4.8 積分法

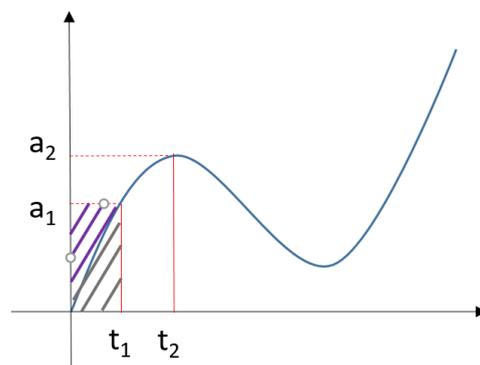


圖 4.9 公式 4.8 積分法誤差

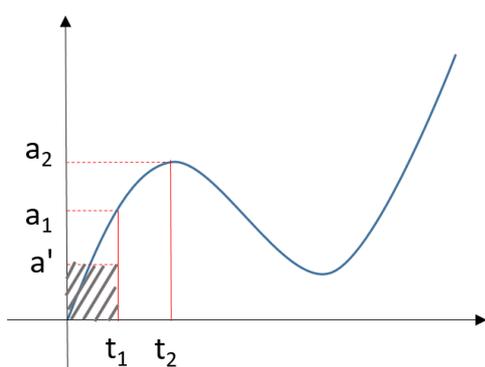


圖 4.10 公式 4.10 積分法

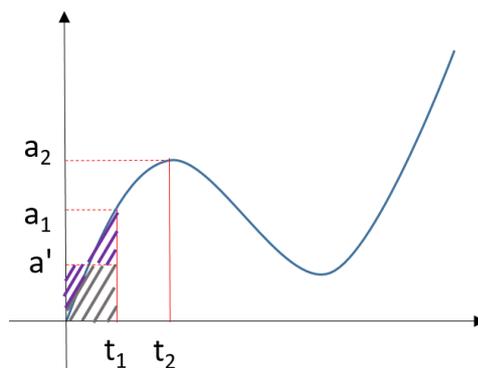


圖 4.11 公式 4.10 積分誤差抵銷

由於我們偵測的是運動員的姿態動作，所偵測的為三維空間的資訊，因此我們利用三軸的加速度值來計算，以各軸的加速度值計算各軸的速度及位移，即可知道在每個取樣點間，受測物體分別在三軸的位移量，再將為位繪製至於三維座標上，受測者即可知道在運動過程中的移動及轉動變化狀況。

在計算由加速度值轉換成速度的過程中，會有個物理性質上的盲點，當受測物體靜止時，加速度為 0 解且速度也應當要為 0，但以公式 4.8 或公式 4.10 來計算時，當加速度為 0 時，速度還是受到上一個時間點的速度影響，還是保持著相同的速度，因此在計算軌跡的時候，會造成受測物體以

等速度持續位移下去，並不是靜止的狀態，因此本研究在計算速度的過程中，加入了一個篩選條件，假設連續兩個或三個時間點的加速度為 0 時，直接將速度歸 0，如此一來才不會造成上述結果。

第五章 系統實驗結果

5.1 環境及實驗設定

為了測試本研究之系統能否執行以及驗證系統結果精準度，因此本研究將感測系統移動幾種不同軌跡後進行比較，將系統輸出結果與真實軌跡情形做比較，計算出本系統的誤差程度。

5.2 設備畫面

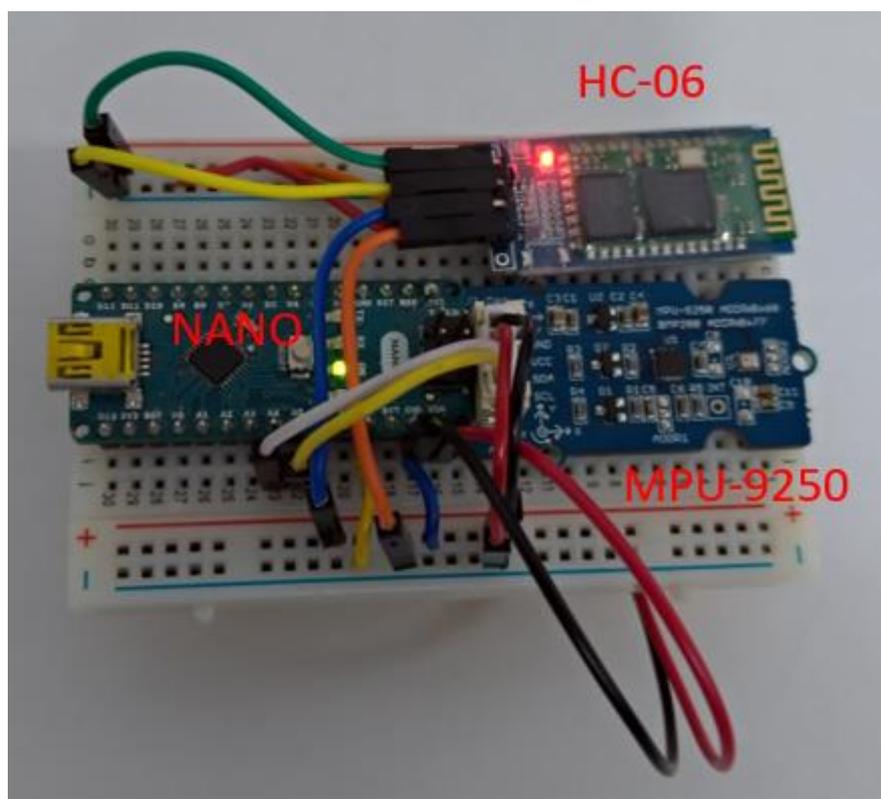


圖 5.1 感測器設備

圖 5.1 為本研究所使用感測器，主要由 MPU-9250 進行偵測數據，而 NANO 開發板即為控制器維 ATmega-328 的開發板，而 HC-06 主要負責本研究的數據傳輸，藉由藍芽傳輸，將數據傳輸至電腦進行分析，電源供應的部分因 NANO 開發板需要 9V 電源才能運作，而 9V 電池體積較維龐大，因此將 9V 電池安裝於麵包板後方；下圖 5.2 為本研究感測器接線圖。

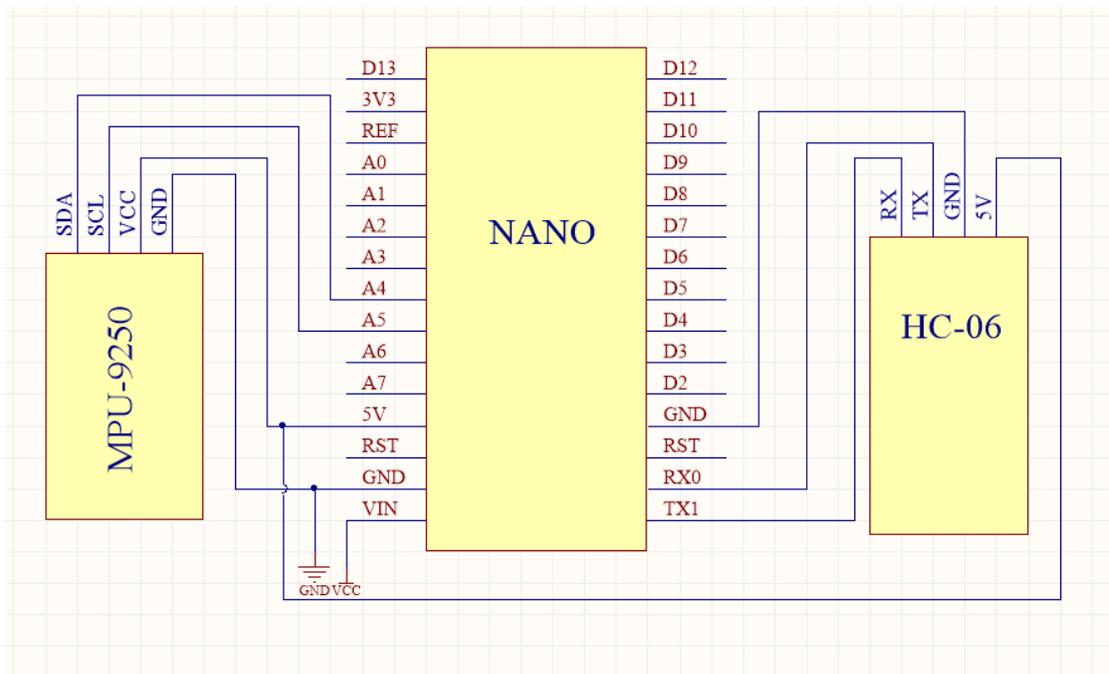


圖 5.2 感測器接線圖

5.3 六軸 AHRS 與 9 軸 AHRS 比較

5.3.1 六軸 AHRS

原先本研究所採用的 IMU 感測晶片為 MPU-6050，而 MPU-6050 所提供的感測資訊為六軸(三軸加速度、三軸陀螺儀)。本研究利用六軸的資訊進行 Madgwick 所提出的 AHRS 姿態解算，將六軸感測資訊帶入三軸磁力計皆為 0 的算法中求出代表運動姿態的四元數，搭配四元數及加速度向量轉換成地球座標軸下的加速度，在計算出速度及位移用以畫出軌跡模型，下圖 5.3 為六軸感測器偵測而得知數據所繪畫出的軌跡模型，由圖 5.3 可知六軸的感測器在做快速運動時會產生飄移，而飄移量會隨著時間而逐漸擴大。

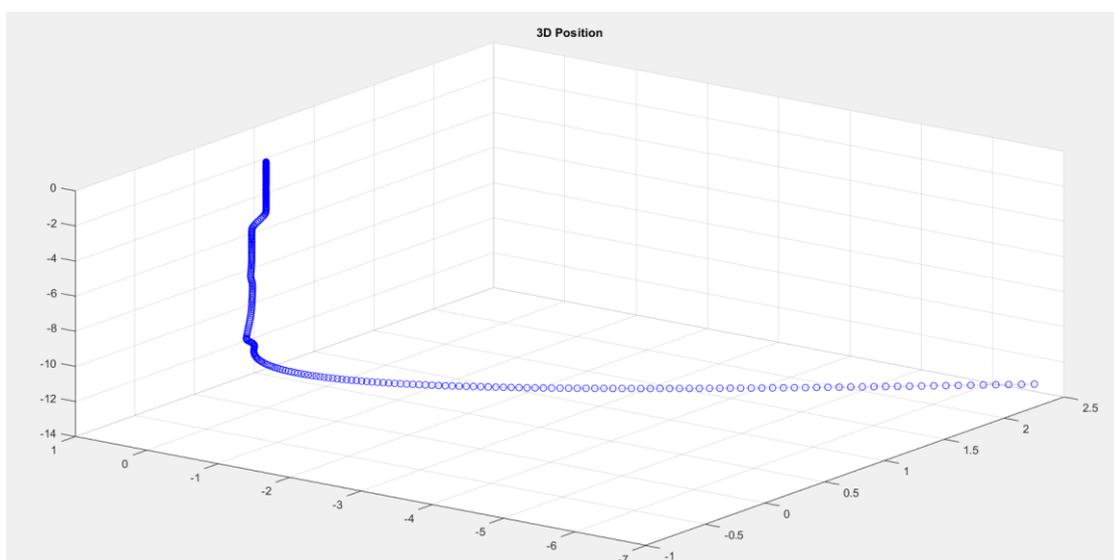


圖 5.3 六軸感測器軌跡模型

了解軌跡模型會產生飄移後，查閱原始加速度、角速度值皆為正常，因此問題可能出現在 AHRS 所計算而得的四元數，為了驗證四元數是否正確，本研究依照公式 4.6 計算出 Roll、Pitch、Yaw，轉動感測器查看 Roll、Pitch、Yaw 各自所代表的座標軸轉動角度是否正確。

表 5.1 六軸 AHRS 轉換之尤拉角

Roll	Pitch	Yaw	Roll	Pitch	Yaw	Roll	Pitch	Yaw
-0.15	0.04	0.07	-0.03	0.18	2.04	-54.42	6.43	47.32
-0.15	0.03	0.07	-0.03	0.17	2.06	-52.87	7.86	56.14
-0.15	0.03	0.07	-0.03	0.15	2.07	-48.08	15.84	63.51
-0.15	0.02	0.06	-0.02	0.14	2.08	-37.37	22.36	72.7
-0.14	0.02	0.08	-0.01	0.13	2.09	-29.82	21.58	79.07
-0.13	0.04	0.08	0	0.2	2.07	-24.32	17.41	74.28
-0.13	0.05	0.09	-0.01	0.24	2.09	-15.12	-4.14	71.43
-0.12	0.06	0.1	-0.03	0.22	3.16	-15.29	-6.34	66.77
-0.12	0.08	0.12	-0.05	0.21	3.87	-14.41	-7.09	61.93
-0.09	0.1	0.16	-0.09	0.23	4.03	-13.54	-8.13	56.31
-0.08	0.1	0.24	-0.15	0.14	4.79	-12.77	-8.66	52.21
-0.06	0.13	0.29	-0.21	0.05	5.39	-11.95	-9.25	47.71
-0.05	0.14	0.3	-0.24	-0.11	10.52	-11.03	-9.13	42.98
-0.05	0.15	0.35	-0.29	-0.29	18.19	-10.34	-11.58	37.51
-0.06	0.16	0.41	-0.41	-0.52	26.75	-9.01	-12.67	30.64
-0.07	0.16	0.46	-0.53	-0.68	33.03	-7.6	-12.97	25.16
-0.08	0.16	0.45	-0.54	-0.74	32.8	-6.37	-13.26	20.11
-0.08	0.17	0.49	-0.43	-0.33	26.56	-4.81	-13.28	14.01
-0.08	0.18	0.54	-0.48	-0.31	26.85	-3.51	-13.41	8.74
-0.09	0.2	0.63	-0.36	-0.04	38.07	-2.19	-13.43	3.39
-0.09	0.19	0.69	-0.46	-0.89	55.62	-1.1	-13.42	-1.1
-0.07	0.2	0.74	-1.02	-1.42	73.43	-0.24	-13.33	-4.66
-0.07	0.2	0.81	-1.67	-1.66	91.95	0.33	-13.22	-7.01
-0.07	0.19	0.92	-1.85	-0.8	109.13	0.59	-13.23	-8.28
-0.07	0.18	1.04	-1.57	3.73	118.45	0.35	-13.23	-7.39
-0.08	0.18	1.14	1.01	5.36	113.26	0.09	-13.2	-6.15
-0.07	0.17	1.29	-2.08	12.42	93.85	-0.04	-13.2	-5.56
-0.08	0.15	1.4	-8.97	19.35	70.18	-0.14	-13.16	-5.07

Roll	Pitch	Yaw	Roll	Pitch	Yaw	Roll	Pitch	Yaw
-0.06	0.15	1.56	-18.51	12.07	52.35	-0.26	-13.1	-4.47
-0.06	0.14	1.6	-27.62	2.53	47.73	-0.29	-13.08	-4.32
-0.04	0.18	1.71	-37.51	5.22	54.01	0.05	-12.89	-5.58
-0.04	0.18	1.76	-36.6	22.5	64.82	0.53	-12.77	-7.66
-0.04	0.19	1.79	-24.66	35.94	80.15	1	-12.64	-9.54
-0.03	0.18	1.77	-6.76	40.64	102.86	1.28	-12.54	-10.74
-0.03	0.18	1.8	12.35	36.32	125.63	1.4	-12.48	-11.24
-0.03	0.18	1.8	25.93	36.8	135.04	1.46	-12.43	-11.45
-0.06	0.17	1.81	26.06	43.68	130.75	1.48	-12.39	-11.59
-0.06	0.18	1.82	-1.13	52.23	101.96	1.48	-12.35	-11.59
-0.05	0.19	1.96	-28.12	40.29	73.04	1.49	-12.32	-11.66
-0.05	0.21	2.03	-42.39	17.62	59.78	1.49	-12.29	-11.67
-0.05	0.2	2.05	-50.33	6.15	48.51	1.5	-12.26	-11.67

上表 5.1 為某次實驗六軸 AHRS 轉換之尤拉角片段資料，感測器在初始狀態下，Roll、Pitch 軸之角度變化皆不大，Yaw 軸之角度會隨著時間做正弦波的變化，可利用 AHRS 中的 KP、KI 值做修改，將振幅縮小到一個範圍之內，在本次實驗中，本次實驗劇烈晃動感測器，發現經過劇烈晃動後，由四元數所轉換之尤拉角皆為異常，此次實驗為晃動 Z(Yaw)軸，如下圖所示，則 Z 軸之尤拉角變化程度將會與晃動程度成正比，成發散狀態持續飄移，另外兩軸則是固定在某些數值下做小幅度變化，為了修正此問題，本研究替換了原先使用的 MPU-6050，使用了具備 9 軸感測資訊的 MPU-9250。

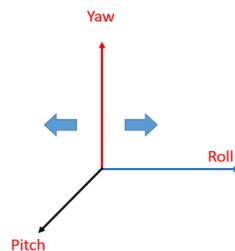


圖 5.4 坐標軸晃動方向

5.3.2 九軸 AHRS

為了修正飄移的問題，本研究將 IMU 感測元件更換為九軸感測器 MPU-9250，MPU-9250 相較於 MPU-6050 多了三軸的磁力計，在九軸的 AHRS 之中加入磁力計對於四元數變化率做修正，取得更為精準的四元數的變化率。在這裡與六軸 AHRS 有些微不同，因磁力計的加入，在靜止狀態下磁力計依然有數值，因此在靜止狀態下 Roll、Pitch、Yaw 皆有初始值，在本次實驗中，搖晃感測器後將感測器放回相同位置，再查看 Roll、Pitch、Yaw 是否與初始值相同以及是否產生飄移。

表 5.2 九軸 AHRS 轉換之尤拉角

Roll	Pitch	Yaw		Roll	Pitch	Yaw		Roll	Pitch	Yaw
0.04	-0.04	35.63		-12.91	-37.55	88.4		-1.4	0.8	43.55
-0.02	-0.06	35.49		1.4	-36.2	64.07		-1.24	0.7	42.96
0.06	-0.08	35.71		-10.92	-31.41	87.25		-1.23	0.73	42.21
0.05	-0.05	35.77		-27.11	-38.41	115.5		-1.18	0.7	41.53
0.06	-0.07	35.7		-11.1	-40.73	90.07		-1.04	0.62	40.99
-0.09	0.01	35.36		-24.54	-33.7	113.03		-0.96	0.55	40.39
-0.02	0	35.55		-11.37	-34.34	88.56		-0.87	0.52	39.89
Roll	Pitch	Yaw		Roll	Pitch	Yaw		Roll	Pitch	Yaw
0.03	-0.11	35.69		-6.83	-33.29	81.3		-0.71	0.45	39.54
-0.02	-0.01	35.67		-9.26	-29.7	87		-0.64	0.36	39.18
-0.07	0.06	35.5		-6.39	-23.04	80.19		-0.58	0.43	38.59
-0.08	0.05	35.45		-4.42	-19.08	76.15		-0.57	0.33	38.11
-0.01	0	35.64		-3.05	-15.69	73.42		-0.44	0.29	37.92
0.01	-0.03	35.54		-1.52	-12.52	69.35		-0.48	0.29	37.52
0.08	-0.09	35.78		-0.2	-14.17	65.27		-0.37	0.24	37.28
-0.03	-0.03	35.73		0.88	-12.16	63.17		-0.28	0.24	37.05
-0.01	0	35.8		0.74	-9.79	66.76		-0.31	0.28	36.76
-0.01	0.03	35.71		1.22	-8.78	65.91		-0.23	0.23	36.65
0	0.01	35.83		1.16	-8.43	65.35		-0.29	0.24	36.31
0.07	-0.04	36.01		1.09	-8.14	64.77		-0.25	0.19	36.15

Roll	Pitch	Yaw		Roll	Pitch	Yaw		Roll	Pitch	Yaw
-0.02	0.03	35.97		0.77	-7.78	66.28		-0.28	0.15	35.87
-0.14	0.05	35.77		0.45	-7.62	65.57		-0.36	0.16	35.51
-0.03	-0.01	36.03		0.22	-7.23	66.35		-0.25	0.11	35.39
-0.03	-0.05	35.92		0.34	-6.92	64.37		-0.15	0.08	35.37
0.05	-0.09	36.1		0.25	-6.51	62.53		-0.11	0.04	35.46
0.02	0	36.16		0.15	-6.09	61.87		-0.13	-0.03	35.57
-0.09	0.04	35.93		0.13	-5.77	61.27		-0.15	-0.01	35.43
-0.23	0.15	35.56		0.08	-5.44	60.59		-0.16	-0.14	35.5
-0.15	0.08	35.79		-0.09	-5.12	60.08		-0.08	-0.13	35.57
-0.12	0.02	35.83		-0.21	-4.65	59.62		-0.1	0	35.34
-0.09	0.05	35.81		-0.31	-4.24	59.05		-0.14	0.1	34.92
-0.07	0.05	35.82		-0.4	-3.89	58.46		-0.1	0	35.1
-0.04	-0.03	35.89		-0.5	-3.57	57.86		-0.09	0.02	35.15
-0.05	-0.02	37.04		-0.59	-3.21	57.27		-0.11	0.06	34.99
0.39	0.3	32.38		-0.69	-2.86	56.6		-0.2	0.01	34.89
0.54	0.58	29.52		-0.79	-2.54	55.97		-0.06	-0.01	35.05
0.98	0.5	32.41		-0.88	-2.21	55.35		-0.1	0.04	34.94
0.51	-0.13	33.47		-1	-1.91	54.75		-0.07	0.05	34.9
0.57	-0.23	33.97		-1.1	-1.62	54.14		-0.14	0.11	34.66
0.59	-0.25	34.42		-1.19	-1.35	53.49		-0.05	0.02	34.8
0.62	-0.3	34.86		-1.24	-1.09	52.77		0	-0.03	34.96
0.61	-0.32	35.19		-1.32	-0.84	52.1		0.03	-0.05	35.1
0.2	0.3	34.91		-1.38	-0.57	51.45		-0.01	-0.1	35.23
0.6	0.73	32.77		-1.42	-0.35	50.75		0	-0.09	35.4
1.2	-2.08	40.1		-1.47	-0.13	50.04		-0.05	-0.08	35.28
1.82	-8.88	38.34		-1.49	0.05	49.34		-0.15	-0.17	35.29
1.57	-16.38	42.68		-1.5	0.15	48.63		-0.22	-0.08	35.02
0.81	-15.73	47.94		-1.53	0.32	47.91		-0.14	-0.05	35.1
-3.65	-19.03	64.99		-1.55	0.48	47.19		-0.04	-0.1	35.36
-9.99	-28.33	81.97		-1.52	0.61	46.46		-0.18	-0.01	35

Roll	Pitch	Yaw		Roll	Pitch	Yaw		Roll	Pitch	Yaw
-18.88	-36.38	97.85		-1.44	0.66	45.75		-0.1	-0.04	35.15
-4.9	-34.69	73.06		-1.42	0.71	45.02		-0.12	0	35.02
0.95	-32.59	63.72		-1.4	0.76	44.29		-0.15	0.04	35.09

5.4 實驗結果與真實軌跡比較

本研究將進行三個階段的驗證，先由較為簡單的圖形 L 圖形做驗證，接著進行較多轉折的 W 圖形做驗證，驗證多轉折之圖形是否會造成影響。為了比較真實圖形與系統輸出軌跡圖形的位置，但在實驗中的軌跡可能非直線運動，因此在做檢驗的部分，主要測量轉折點與原點的距離。

5.4.1 平面 L 形軌跡

在 L 圖形的部分依圖 5.3 及圖 5.4 所示，圖 5.5 及圖 5.6 分別為本次 L 圖形實驗的加速度值及計算出的位移值。由圖 5.3 得知在此圖形中大致上可以找出 A、B、C 三點，為了驗證距離與軌跡是否為正確，本研究將計算任意兩點間的線段距離，與真實所畫之圖形的線段距離是否相符。

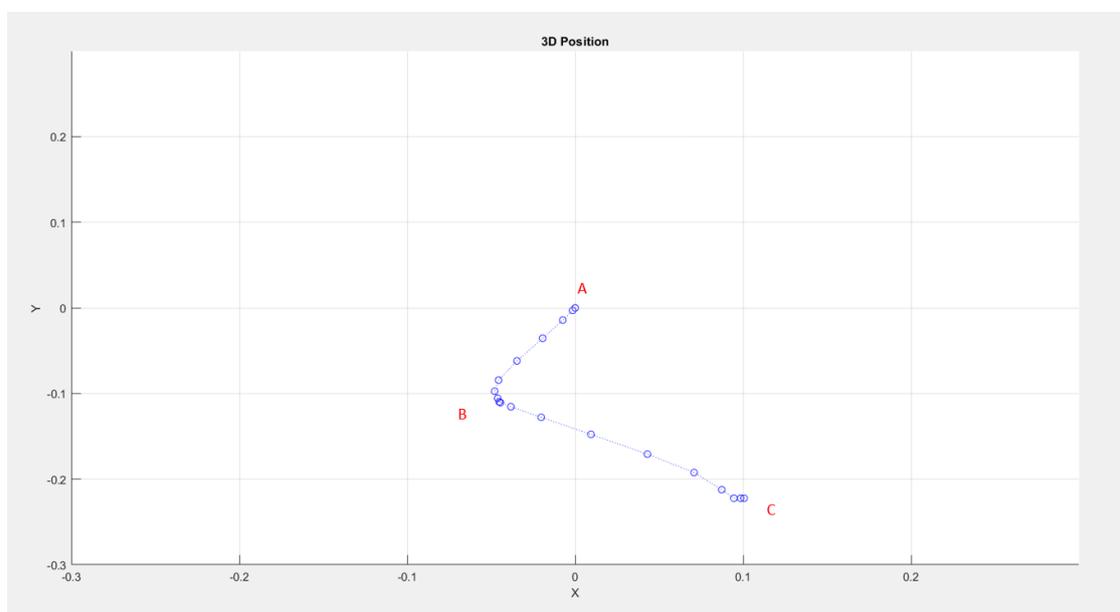


圖 5.5 L 形軌跡 XY 平面軌跡圖

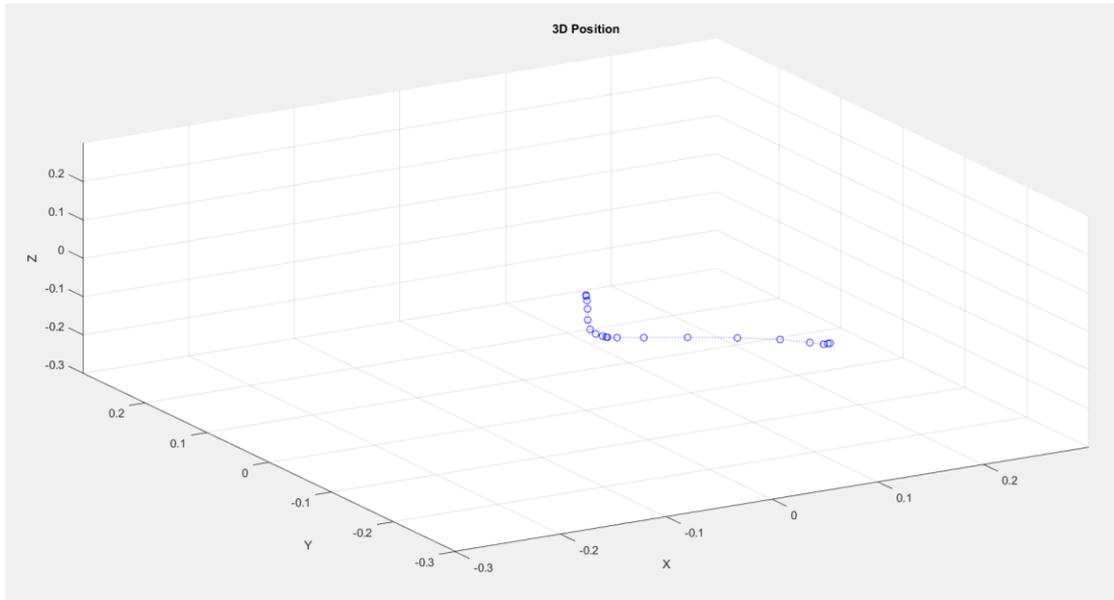


圖 5.6 L 形軌跡三維軌跡圖

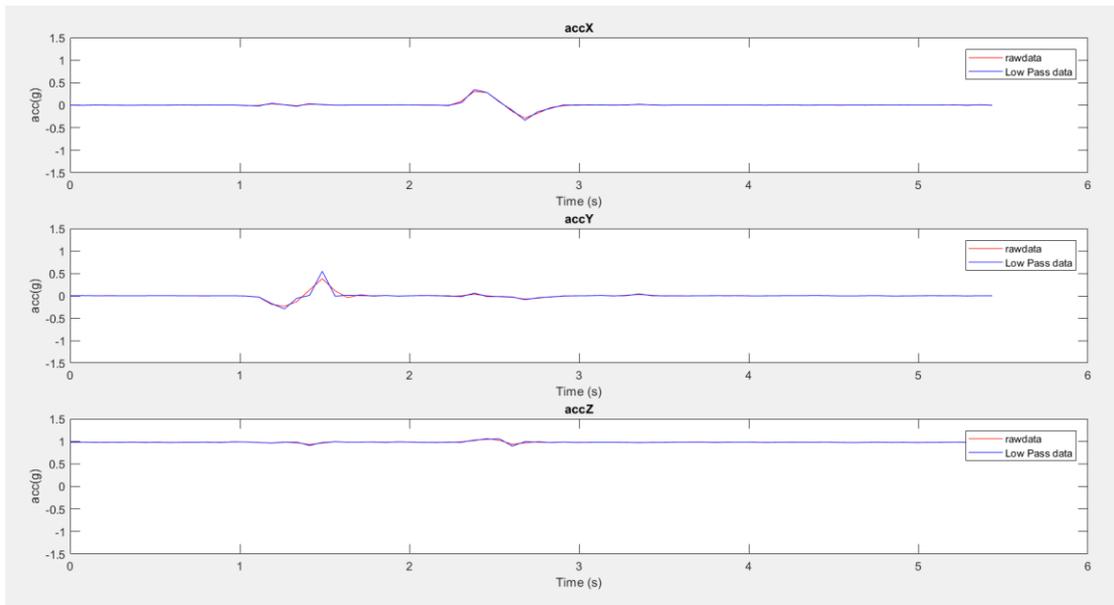


圖 5.7 L 形軌跡三軸加速度值

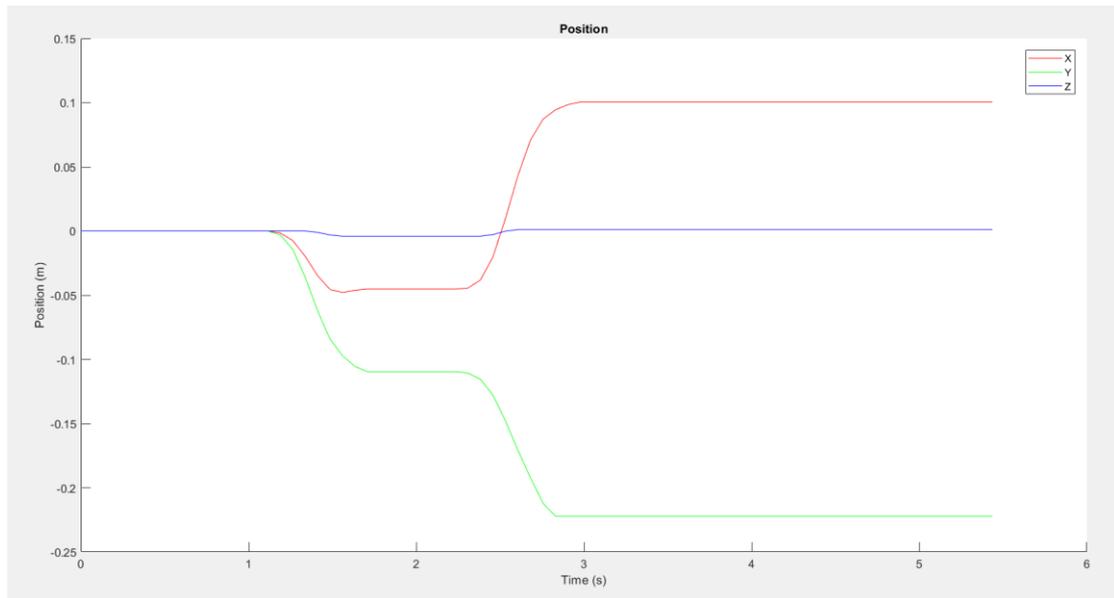


圖 5.8 L 形軌跡三軸位移值

表 5.3 L 形三點座標

A	(0 , 0)
B	(-4.78 , -9.73)
C	(10.05 , -22.23)

表 5.4 L 形軌跡誤差

	真實軌跡	系統軌跡	誤差率
\overline{AB}	11.4	10.84	4.9%
\overline{BC}	20.7	19.4	6.3%
\overline{AC}	24.7	24.39	1.3%

5.4.2 平面 W 形軌跡

在 W 圖形的部分依圖 5.7 及圖 5.8 所示，圖 5.9 及圖 5.10 分別為本次 W 圖形實驗的加速度值及計算出的位移值。由圖 5.7 得知在此圖形中找出 A、B、C、D、E 五點，在本次實驗中，本實驗將計算任意兩點的直線距離，與真實所畫之圖形的線段距離是否相符。

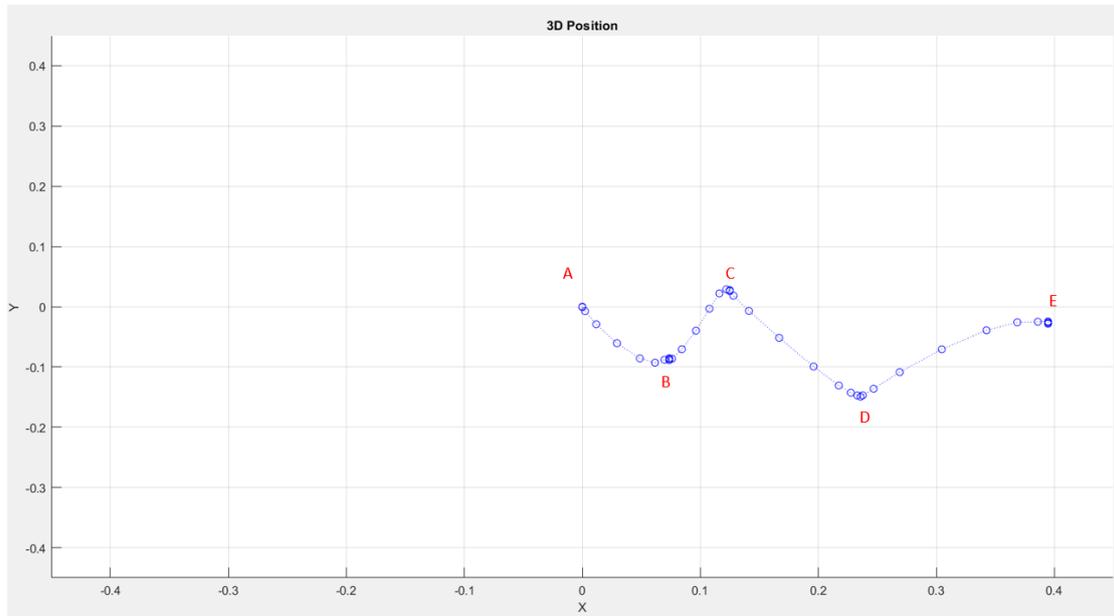


圖 5.9 W 形軌跡 XY 平面軌跡圖

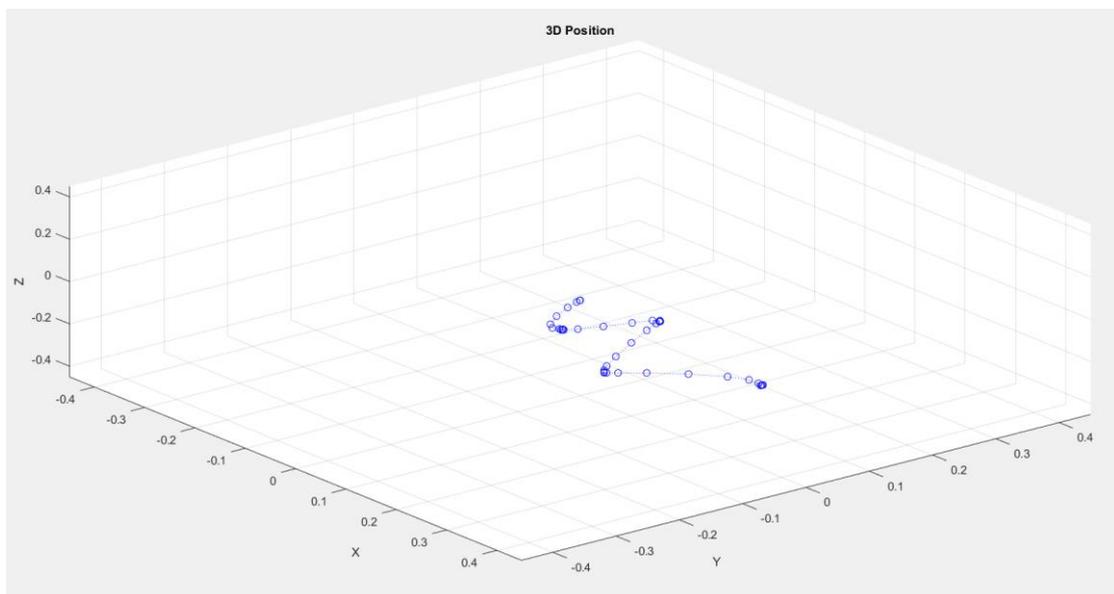


圖 5.10 W 形軌跡三維軌跡圖

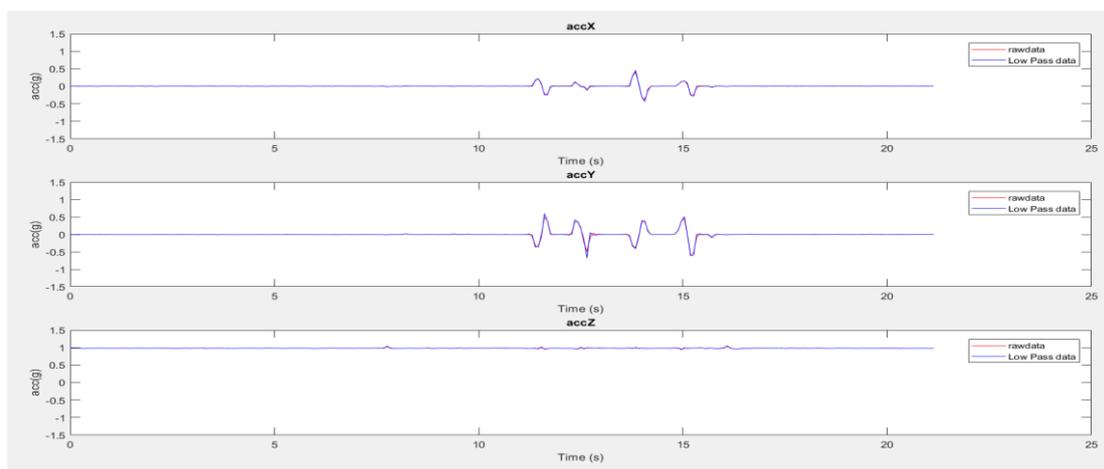


圖 5.11 W 形軌跡三軸加速度值

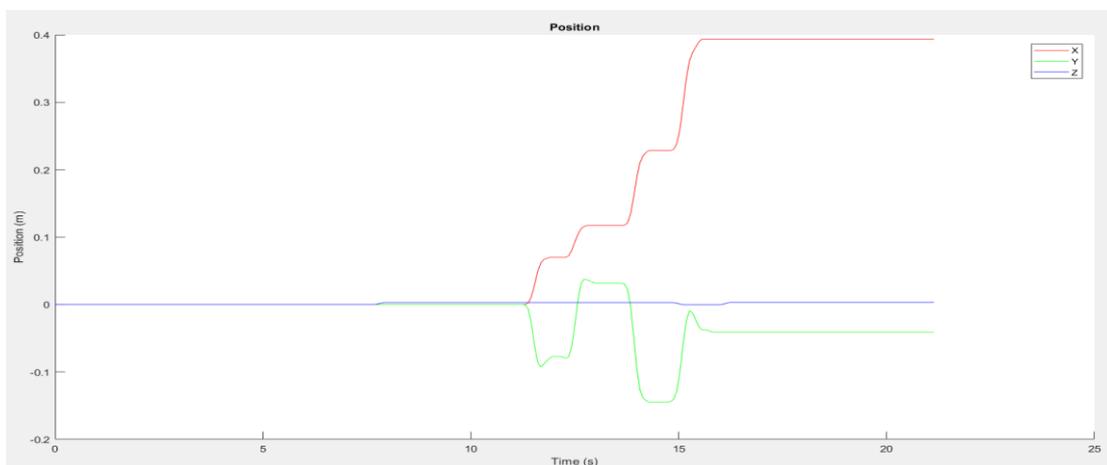


圖 5.12 W 形軌跡三軸位移值

表 5.5 W 形五點座標

A	(0 , 0)
B	(6.16 , -9.28)
C	(12.2 , 2.9)
D	(23.57 , -14.94)
E	(39.46 , -2.82)

表 5.6 W 形軌跡誤差

	真實軌跡	系統軌跡	誤差率
\overline{AB}	12.18	11.1	8.5%
\overline{AC}	14.638	13.7	6.1%
\overline{AD}	30.5	27.9	8.5%
\overline{AE}	38.08	39.6	3.8%
\overline{BC}	14.24	13.59	6.5%
\overline{CD}	22.87	21.26	7%
\overline{DE}	18.28	19.89	8.8%

5.4.3 三維運動軌跡

在三維運動軌跡的實驗中，本次實驗將於桌面進行一段平面運動後朝位於桌面之下的椅子進行移動，查看加入 Z 軸的因素是否會造成影響，由圖 5.11 來看，在本次實驗中先標註 4 個端點，本次實驗將利用這些端點進行軌跡驗證。

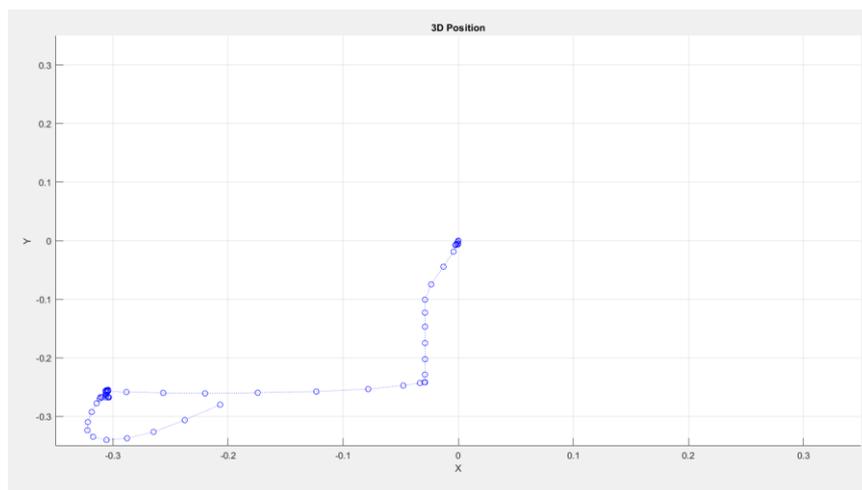


圖 5.13 三維運動軌跡 - XY 平面軌跡

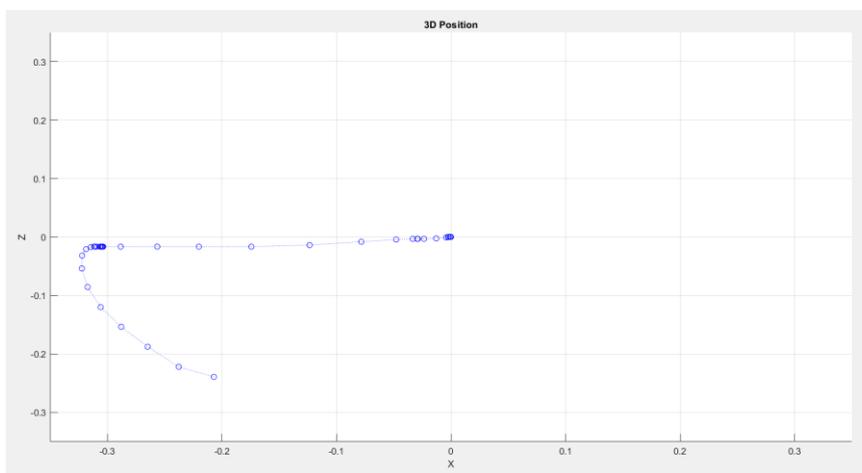


圖 5.14 三維運動軌跡 - XZ 平面軌跡

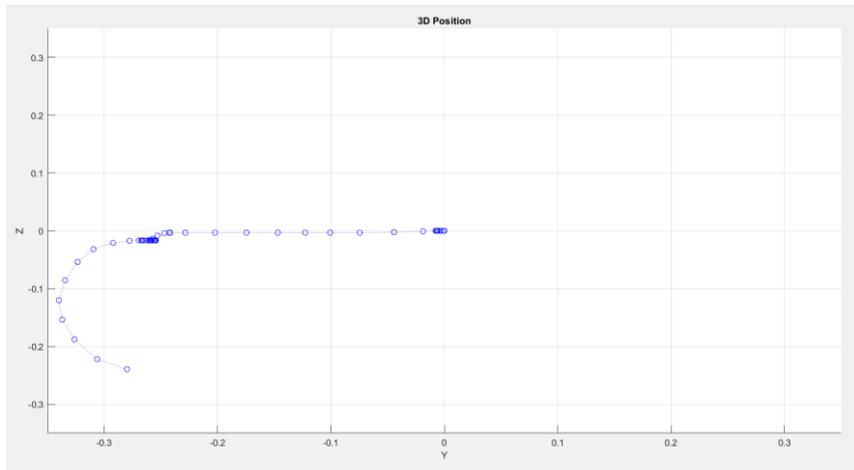


圖 5.15 三維運動軌跡 - YZ 平面軌跡

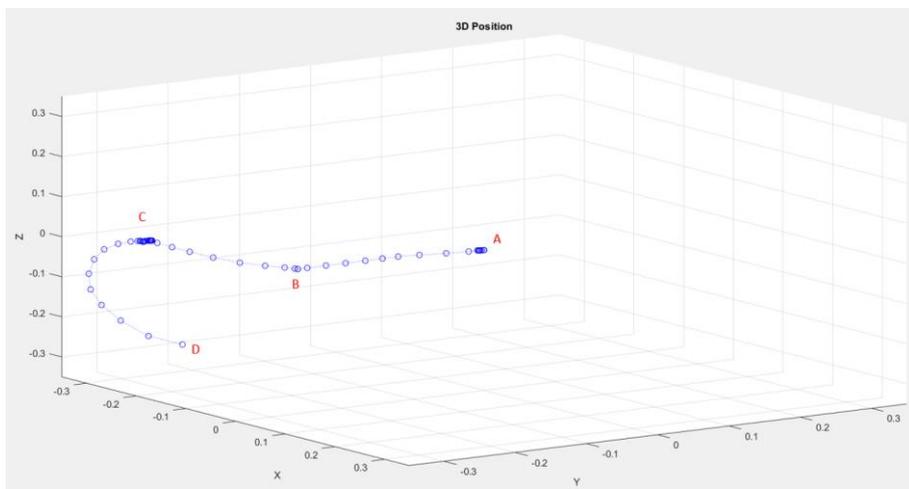


圖 5.16 三維運動軌跡 - 立體空間軌跡

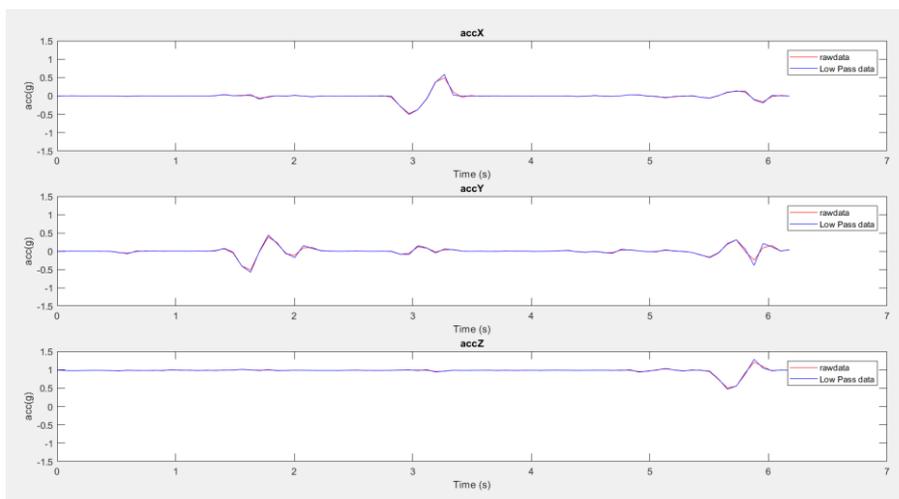


圖 5.17 三維運動軌跡三軸加速度值

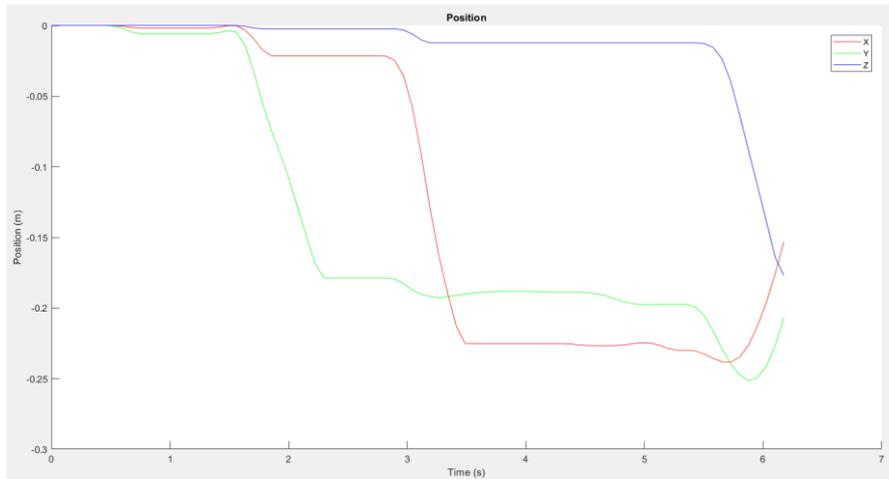


圖 5.18 三維運動軌跡三軸位移值

表 5.7 三維運動軌跡座標點

A	(0 , 0)
B	(-2.14 , -17.87)
C	(-25.5 , -18.83)
D	(-15.31 , -20.7 , -17.71)

表 5.8 三維運動軌跡誤差

	真實軌跡	系統軌跡	誤差率
\overline{AB}	19.4	17.99	7.26%
\overline{AC}	28.7	31.69	10.4%
\overline{BC}	21.8	23.38	7.2%
桌面與椅子落差	18.5	17.71	4.2%

第六章 結論與未來研究

6.1 結論

本研究建構一套適用於輔助運動員改善動作的軌跡重建系統，此系統成功將慣性感測元件所偵測到的動作資訊透過藍芽傳輸，並利用 MATLAB 將動作資訊經過低通濾波器將雜訊去除，以及進行 AHRS 姿態解算求得代表該時間點之動作的四元數，接著利用四元數搭配三軸加速度所組成的向量進行座標軸的轉換，計算出屬於地球座標軸的加速度，有了地球座標軸下的三軸加速度後即可計算出三軸的各自位移量，並使用各軸的位移量進行繪圖，將每個時間點的點連接起來即是本系統重建之軌跡。最後於驗證軌跡時，本研究所重建之軌跡確實與真實軌跡有所相符，而誤差率也控制在一定程度之下，證明該系統確實可以提供運動員改善動作時的參考依據。

本研究透過 IMU 感測器與 AHRS 所建構出的軌跡重建系統做出以下之貢獻：

1. 本研究成功使用 IMU 感測器與 AHRS 姿態解算法做結合，建構出一套軌跡重建之系統，提供運動軌跡、姿態數據以及各項數據。
2. 本研究所建立之系統，經由驗證之後，其系統之誤差皆在一定範圍之中，代表軌跡還原程度高。
3. 過往的運動輔助設備被認為是屬於高單價的輔助儀器，因此對於普通的運動員來說負擔太高，而本研究之軌跡重建系統成功將成本降低，與價格動輒上萬元的攝影輔助設備有一定之落差，而本研究之系統所輸出之模型可任意調整三維視角觀測，更有助於運動員觀察動作。
4. 市面上所販售之輔助設備通常都侷限某項運動之中，對於較少數人進行的運動中，其輔助設備可能極難取得，而本研究之系統可適用於各項運動中，只要加速度及角速度不要超過感測器範圍，本研究之系統皆可輸出軌跡，若在其他領域也須用到軌跡重建的部分也可使用本研究之系統，並不局限於運動項目中。

6.2 未來研究

隨著科技的發展，感測器的精準度也隨之提高，在過往只能只用一些較為低階的感測晶片來偵測數據，如今已發展出較為高階的晶片，除了精準度的提升之外，在同一塊晶片中所搭配的功能也隨之提高，在本研究中所使用之感測晶片為 MPU-9250，搭配 HC-06 藍芽傳輸做數據的傳輸，考慮到成本及方便性，本研究僅能建構出可行之模型，但本研究之模型還有以下缺陷尚未解決：

1. 本研究所使用的 IMU 感測元件為 MPU-9250，搭 I2C 傳輸協定，然而其傳輸速度並非十分即時，造成取樣資料時間間隔較大，因此若在短時間內做出急劇的方向改變等等，有可能造成動作資訊接收上的落差，除此之外在 AHRS 姿態解算的部分，也會隨著時間間隔的關係造成更新速率較慢。
2. 本研究系統之硬體設備皆為簡易模型，目前尚有空間做改善，只擷取所需晶片，重新進行電路布局，重新設計後的感測器應當可以節省相當大空間，令使用者在使用上更具方便性。
3. 目前本研究之系統為 Arduino 搭配 MATLAB 所建構而成，兩種開發平台使用之語言不同，Arduino 使用 C 語言控制 ATmega-328 控制器，而 MATLAB 則進行龐大運算及繪圖功能。若能使用 MATLAB 取代 Arduino 控制 ATmega-328 控制器，在操作上將會簡易許多。
4. 本研究之軌跡目前只是將軌跡連線，並將結果同時顯示出來。若能依照所對應時間點進行軌跡紀錄做成動畫，勢必有助於使用者對於動作的了解。

參考文獻

- [1] 伍星翰(2011)。線性加速度感測器的研究與其應用(碩士論文)。中央大學，桃園市。
- [2] 林楷崑(2011)。利用影像處理技術進行前景偵測與軌跡重建(碩士論文)。佛光大學，宜蘭縣。
- [3] 邱柏鈞(2012)。基於體感網路的高爾夫揮桿分析(碩士論文)。交通大學，新竹市。
- [4] 楊長圃(2004)。無人直昇機慣性導航系統(碩士論文)。元智大學，桃園市。
- [5] 謝宗倫(2010)。單相機高速運動物體 3D 軌跡重建的新方法(碩士論文)。亞洲大學，台中市。
- [6] Collinson, R. P. G. (1996). *Introduction to Avionics* (pp. 231-237). Microwave Technology Series 11, Chapman & Hall, 1st.
- [7] Fontaine, D., David, D., and Caritu, Y. (2003). Sourceless human body motion Capture. *Proceedings of Smart Objects Conference (SOC 2003)*, Grenoble, 2003.
- [8] Ghasemzadeh, H. & Jafari, R. (2011). Coordination analysis of human movements with body sensor networks: A signal processing model to evaluate baseball Swings. *IEEE Sensors Journal*, 11(3), 603–610.
- [9] King, K., Yoon, S. W., Perkins, N. C. & Najafi, K. (2004). The dynamics of the golf swing as measured by strapdown inertial sensors. *Proceedings of the 5th international conference on the engineering of sport*, 2, 276–282.
- [10] Li Xiaofang, Mao Yuliang, Xie Ling, Chen Jiabin, Song Chunlei(2014). Applications of zero-velocity detector and Kalman filter in zero velocity update for inertial navigation system. *IEEE Navigation and Control Conference*, 88 – 90.
- [11] Mahony, R., Hamel, Y., & Jean-Michel Pflimlin(2008). Nonlinear Complementary Filter on the Special Orthogonal Group. *IEEE Transactions on Automatic Control*, 53, 1203-1218.
- [12] Mathie, M. J., Basilakis, J., & Celler, B. G. (2001). A system for monitoring posture and physical activity using accelerometers. *IEEE Engineering in Medicine and Biology Society*, 4, 3654–3657.
- [13] Urtasun, R., Fleet, D. J., & Fua, P. (2005). Monocular 3d tracking of the golf Swing. *IEEE Conference on Computer Vision and Pattern Recognition*, 2, 932–938.

附錄一 ATmega-328 程式碼

```
#include "Wire.h"
#include "I2Cdev.h"
#include "MPU9250.h"
MPU9250 accelgyro;
I2Cdev I2C_M;
uint8_t buffer_m[6];
int16_t ax, ay, az;
int16_t gx, gy, gz;
int16_t mx, my, mz;
float Axyz[3];
float Gxyz[3];
float Mxyz[3];
float axo=0,ayo=0,azo=0;
float gxo=0,gyo=0,gzo=0;
#define sample_num_mdate 5000
volatile float mx_sample[3];
volatile float my_sample[3];
volatile float mz_sample[3];
static float mx_centre = 0;
static float my_centre = 0;
static float mz_centre = 0;
volatile int mx_max = 0;
volatile int my_max = 0;
volatile int mz_max = 0;
volatile int mx_min = 0;
volatile int my_min = 0;
volatile int mz_min = 0;
void setup()
{
    Wire.begin();
    Serial.begin(38400);
```

```

accelgyro.initialize();
Barometer.init();
delay(1000);

unsigned short times=20;
for(int i=0;i<times;i++)
{
    accelgyro.getMotion9(&ax, &ay, &az, &gx, &gy, &gz, &mx, &my, &mz);
    axo += ax; ayo += ay; azo += az;
    gx0 += gx; gyo += gy; gzo += gz;
}
axo /= times; ayo /= times; azo /= times;
gx0 /= times; gyo /= times; gzo /= times;
}
void loop()
{
    getAccel_Data();
    getGyro_Data();
    getCompassDate_calibrated();
    MadgwickAHRSupdate(Axyz[0],Axyz[1],Axyz[2],radgyro[0],radgyro[1],
radgyro[2],Mxyz[0],Mxyz[1],Mxyz[2]);
    Serial.print(millis());Serial.print("\t");
    Serial.print(Axyz[0],4);Serial.print("\t");Serial.print(Axyz[1],4);Serial.print("\t");
    Serial.print(Axyz[2],4);Serial.print("\t");
    Serial.print(Gxyz[0]);Serial.print("\t");Serial.print(Gxyz[1]);Serial.print("\t");
    Serial.print(Gxyz[2]);Serial.print("\t");
    Serial.print(Mxyz[0]);Serial.print("\t");Serial.print(Mxyz[1]);Serial.print("\t");
    Serial.print(Mxyz[2]);Serial.println("\t");
    delay(5);
}
void getAccel_Data(void)
{
    accelgyro.getMotion9(&ax, &ay, &az, &gx, &gy, &gz, &mx, &my, &mz);

```

```

    Axyz[0] = ((double) ax-axo) / 4096;
    Axyz[1] = ((double) ay-ayo) / 4096;
    Axyz[2] = ((double) az-azo) / 4096+0.981;
}
void getGyro_Data(void)
{
    accelgyro.getMotion9(&ax, &ay, &az, &gx, &gy, &gz, &mx, &my, &mz);
    Gxyz[0] = ((double) gx -gxo)* 250 / 32768;
    Gxyz[1] = ((double) gy -gyo)* 250 / 32768;
    Gxyz[2] = ((double) gz -gzo)* 250 / 32768;
    radgyro[1] = Gxyz[1]*DEG_TO_RAD;
    radgyro[2] = Gxyz[2]*DEG_TO_RAD;
    radgyro[3] = Gxyz[3]*DEG_TO_RAD;
}
void getCompass_Data(void)
{
    I2C_M.writeByte(MPU9150_RA_MAG_ADDRESS, 0x0A, 0x01);
    delay(10);
    I2C_M.readBytes(MPU9150_RA_MAG_ADDRESS, MPU9150_RA_MAG_XOUT_L,
6, buffer_m);
    mx = ((int16_t)(buffer_m[1]) << 8) | buffer_m[0] ;
    my = ((int16_t)(buffer_m[3]) << 8) | buffer_m[2] ;
    mz = ((int16_t)(buffer_m[5]) << 8) | buffer_m[4] ;
    Mxyz[0] = (double) mx * 1200 / 4096;
    Mxyz[1] = (double) my * 1200 / 4096;
    Mxyz[2] = (double) mz * 1200 / 4096;
}
void getCompassDate_calibrated ()
{
    getCompass_Data();
    Mxyz[0] = Mxyz[0] - mx_centre;
    Mxyz[1] = Mxyz[1] - my_centre;
    Mxyz[2] = Mxyz[2] - mz_centre;
}

```

```

}
void get_calibration_Data ()
{
    for (int i = 0; i < sample_num_mdate; i++)
    {
        get_one_sample_date_mxyz();
        if (mx_sample[2] >= mx_sample[1])mx_sample[1] = mx_sample[2];
        if(my_sample[2] >= my_sample[1])my_sample[1] = my_sample[2];
        if (mz_sample[2] >= mz_sample[1])mz_sample[1] = mz_sample[2];
        if (mx_sample[2] <= mx_sample[0])mx_sample[0] = mx_sample[2];
        if(my_sample[2] <= my_sample[0])my_sample[0] = my_sample[2];
        if (mz_sample[2] <= mz_sample[0])mz_sample[0] = mz_sample[2];
    }
    mx_max = mx_sample[1];
    my_max = my_sample[1];
    mz_max = mz_sample[1];
    mx_min = mx_sample[0];
    my_min = my_sample[0];
    mz_min = mz_sample[0];
    mx_centre = (mx_max + mx_min) / 2;
    my_centre = (my_max + my_min) / 2;
    mz_centre = (mz_max + mz_min) / 2;
}
void get_one_sample_date_mxyz()
{
    getCompass_Data();
    mx_sample[2] = Mxyz[0];
    my_sample[2] = Mxyz[1];
    mz_sample[2] = Mxyz[2];
}

```

附錄二 MATLAB 程式碼

```
clear
clc
rawdata = xlsread('all data.xlsx','test');
time = (rawdata(:,1)-rawdata(1,1))/1000;
T= (rawdata(20,1)-rawdata(5,1))/15/1000;
rawaccX = rawdata(:,2);
rawaccY = rawdata(:,3);
rawaccZ = rawdata(:,4);

filtCutOff = 5;
[b, a] = butter(1, (2*filtCutOff)/(1/T), 'low');

dataaccX = filtfilt(b,a,rawaccX);
dataaccY = filtfilt(b,a,rawaccY);
dataaccZ = filtfilt(b,a,rawaccZ);
quat(:,1) = rawdata(:,11);
quat(:,2) = rawdata(:,12);
quat(:,3) = rawdata(:,13);
quat(:,4) = rawdata(:,14);
accX = dataaccX;
accY = dataaccY;
accZ = dataaccZ;
rawacc = [accX accY accZ];

for i = 1:length(accX)
    acc4(i,:) = quaternRotate([accX(i) accY(i) accZ(i)], quaternConj(quat(i,:)));
end

threshold = 0.05;
acc = acc4;
acc(:,3) = acc(:,3)-0.981;
adjacc=[];
for i=1:length(accX)
    if (acc(i,1) < threshold && acc(i,1)> (-1)*threshold)
        adjacc(i,1) = 0;
    else
        adjacc(i,1)=acc(i,1);
    end
end
```

```

end
if (acc(i,2) < threshold && acc(i,2) > (-1)*threshold)
    adjacc(i,2) = 0;
else
    adjacc(i,2)=acc(i,2);
end
if (acc(i,3) < threshold && acc(i,3) > (-1)*threshold)
    adjacc(i,3) = 0;
else
    adjacc(i,3)=acc(i,3);
end
end

unaccumulatevel(1,:)=0;
for i=2:length(accX)
    if (adjacc(i,1) ==0)
        unaccumulatevel(i,1) = 0;
    else
        unaccumulatevel(i,1) = (acc(i,1)+acc(i-1,1))*0.5*T;
    end
    if (adjacc(i,2) ==0)
        unaccumulatevel(i,2) = 0;
    else
        unaccumulatevel(i,2) = (acc(i,2)+acc(i-1,2))*0.5*T;
    end
    if (adjacc(i,3) ==0)
        unaccumulatevel(i,3) = 0;
    else
        unaccumulatevel(i,3) = (acc(i,3)+acc(i-1,3))*0.5*T;
    end
end

vel(1,:)=unaccumulatevel(1,:);
for i=2:length(accX)
    if (unaccumulatevel(i,1)==0 && unaccumulatevel(i-1,1)==0)
        vel(i,1) = 0;
    else
        vel(i,1) = unaccumulatevel(i,1)+vel(i-1,1);
    end
end

```

```

end
for i=2:length(accX)
    if (unaccumulatevel(i,2)==0 && unaccumulatevel(i-1,2)==0)
        vel(i,2) = 0;
    else
        vel(i,2) = unaccumulatevel(i,2)+vel(i-1,2);
    end
end
for i=2:length(accX)
    if (unaccumulatevel(i,3)==0 && unaccumulatevel(i-1,3)==0)
        vel(i,3) = 0;
    else
        vel(i,3) = unaccumulatevel(i,3)+vel(i-1,3);
    end
end

untotalpos(1,:) = 0;
for i = 2:length(accX)
    untotalpos(i,1)=(vel(i,1)+vel(i-1,1))*0.5*T;
    untotalpos(i,2)=(vel(i,2)+vel(i-1,2))*0.5*T;
    untotalpos(i,3)=(vel(i,3)+vel(i-1,3))*0.5*T;
end
pos(1,:) = untotalpos(1,:);
for i =2:length(accX)
    pos(i,1) = untotalpos(i,1)+pos(i-1,1);
    pos(i,2) = untotalpos(i,2)+pos(i-1,2);
    pos(i,3) = untotalpos(i,3)+pos(i-1,3);
end
pos = pos*9.8;

figure(1)
hold on;
plot(time, pos(:,1), 'r');
plot(time, pos(:,2), 'g');
plot(time, pos(:,3), 'b');
title('Position');
xlabel('Time (s)');

```

```

ylabel('Position (m)');
legend('X', 'Y', 'Z');
hold off;

figure (2)
hold on;
subplot(3,1,1),plot(time, rawacc(:,1), 'r');
hold on;
title('accX');
subplot(3,1,1),plot(time, rawaccX(:,1), 'b');
legend('rawdata','Low Pass data');
hold off;
hold on;
subplot(3,1,2),plot(time, rawacc(:,2), 'r');
hold on;
title('accY');
subplot(3,1,2),plot(time, rawaccY(:,1), 'b');
legend('rawdata','Low Pass data');
hold off;
hold on;
subplot(3,1,3),plot(time, rawacc(:,3), 'r');
hold on;
title('accZ');
subplot(3,1,3),plot(time, rawaccZ(:,1), 'b');
hold off;
xlabel('Time (s)');
ylabel('acc');
legend('rawdata','Low Pass data');
hold off;

```

```

figure(6)
hold on;

subplot(3,1,1),plot(time, adjacc(:,1), 'r');
grid on;
title('acc');
subplot(3,1,2),plot(time, adjacc(:,2), 'r');
grid on;

```

```
subplot(3,1,3),plot(time, adjacc(:,3), 'r');  
grid on;  
hold off;
```

```
figure (7)  
hold on;  
plot3(pos(:,1),pos(:,2),pos(:,3),'b:o');  
set(gca,'xlim',[-1 1]);  
set(gca,'ylim',[-1 1]);  
set(gca,'zlim',[-1 1]);  
title('3D Position');  
grid on;  
hold off;  
hold of;
```