

Drone Implementation

無人機專題實作

專題領域：系統組 組別：B135 指導教授：呂仁碩 教授

組員姓名：吳瞳、邱彥廷、陳煜清、陳仕昕

Abstract

近年來，無人機在各方面，不論是娛樂、物流、科學探勘都有很多應用。而之前實驗室就有學長在專題中實作了無人機的控制器，但僅限於 roll(翻滾)、pitch(俯仰)、yaw rate(偏擺速率)三者的 PID control。延續之前的研究成果，我們希望能夠在無人機上增加更多功能，提升其實用性。而在這次的專題中，我們的目標是應用各種感測模組，實作出室內自動定高(Z 軸位置鎖定)、自動定點(X、Y 軸位置鎖定)的功能。

因為無人機的飛行對周遭的人具有相當的危險性，但定點功能又必須讓無人機自由飛行才能測試。因此，我們最初將精力集中在定高功能上，希望能在有穩定的定高功能後，再去開發定點功能。

而在定高功能方面，我們選擇利用 PID controller 作為控制器。在開發初期，經過一段時間的測試、調整，我們成功讓無人機在測試平台上實現穩定的定高功能。然而在室內的飛行測試中，因為電池供電問題、距離感測錯誤等影響，無人機時常會失控、墜機，造成螺旋槳葉片損毀等問題。

為了解決在自由飛行時容易失控的問題，我們先後嘗試了更改距離感測器、更改控制器架構、調整無人機配重，並建立新的測試平台等手段，最後成功達到穩定的定高功能。而定點方面，我們採用光流計作為感測器，但由於測試環境不夠寬闊，難以進行有效的測試，對測試人員的安全也有威脅。因此我們停止定點功能的開發，並開始尋找專題的其他方向。

在專題的後半時間，我們決定更改專題方向。而最後我們成功的做出了以下功能：穩定自動起降、按照事先規劃的數個高度依序飛行、遠端調整無人機參數、遠端透過 GUI 監測無人機飛行狀況。

Introduction

此專題使用的無人機為一四軸飛行器(Quadcopter)，由機身、馬達、驅動馬達的電子調速器(ESC)、螺旋槳、無線傳輸模組、測距模組及 Synopsys IoTDK 嵌入式開發板組成，如圖 1。控制高度的控制器在開發板上透過 C 語言實作，並利用無線傳輸模組與 PC 端的軟體溝通，得到實時的飛行資料如高度、姿態等等。

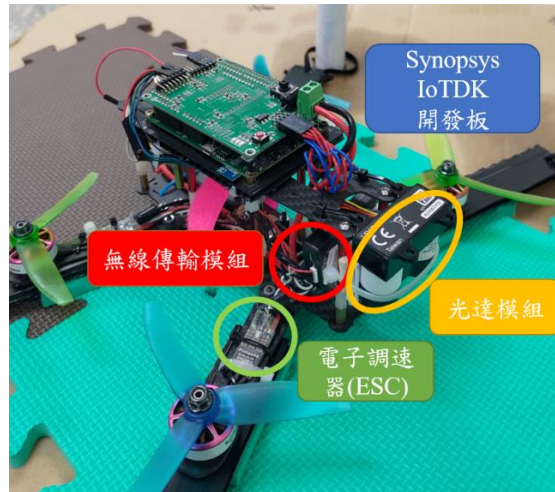


圖 1: 無人機及其組成

一、 原理分析與系統設計

1.1 感測器

無人機上的感測器分成兩部分，分別是量測及計算無人機姿態(Attitude)的陀螺儀，及量測高度(Altitude)的感測器。儘管偵測高度的感測器本身並非專題的內容，但感測器重要性在於測量的數值會直接影響控制器運作。因此我們總共嘗試過三種不同的感測器，分別是超音波測距模組、光達測距模組，以及氣壓計。最後選用光達測距模組。

I. 光達測距模組

光達測距模組的測量範圍約為十公分至數十公尺。雖然最遠的測量距離遠超出室內定高的需求，但由於光達是使用雷射，而雷射不容易發散，不容易偵測到地面以外的物體，且此模組的解析度約為 2.5cm，適合室內使用，因此最後選擇使用光達作為測量高度的方法。光達模組有兩種回傳高度的方式，一是 PWM，二是利用 I2C。我們選用 I2C，主因是 pulse width 是利用時間差傳遞訊息，為一類比量，容易受到干擾，而 I2C 傳遞的是數位訊號，具有較為穩定的特性。

1.2 控制系統設計及調整

I. 控制系統設計

圖 2(上)是無人機系統的方塊圖，主要由無人機的控制系統，及無人機本身組成。控制系統會根據控制目標如高度(Target altitude)和姿態(Target attitude)，及量測的高度(Measured altitude)和姿態(Measured attitude)等，輸出一組控制訊

號給馬達的控制器。此訊號為 PWM 訊號，當 PWM 的 duty cycle 越大，馬達的轉速越高，能提供較大的推力。

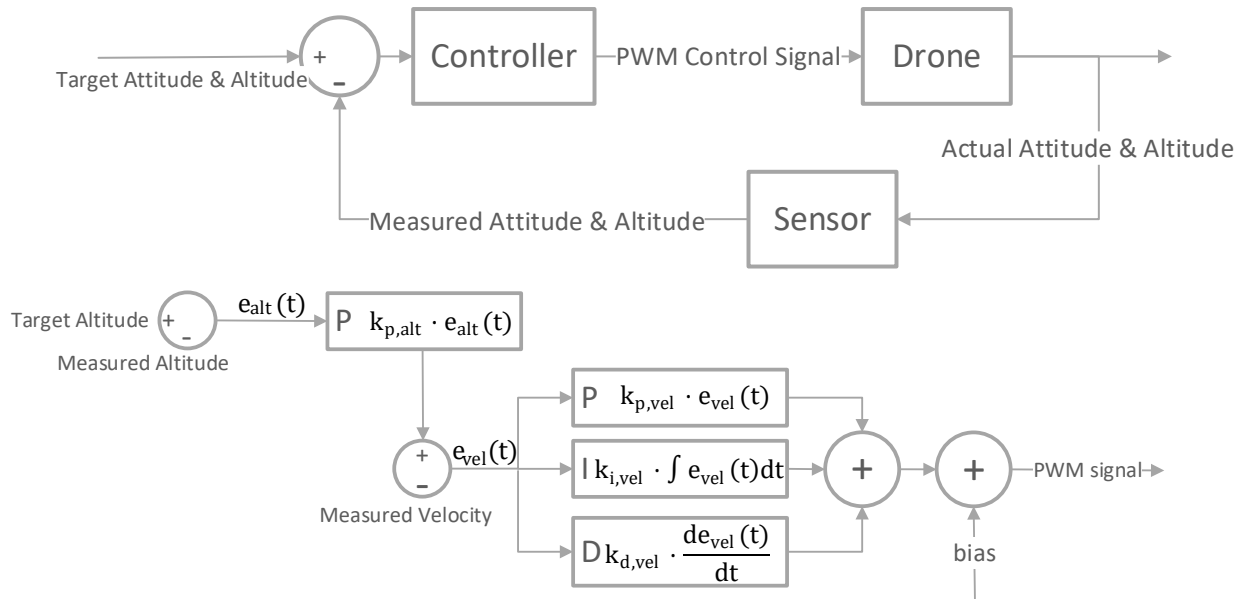


圖 2: 控制系統與高度控制器方塊圖

控制系統由多個不同的控制器組成。分別有控制無人機翻滾(Roll)、俯仰(Pitch)、偏擺速率(Yaw rate)及高度(Altitude)的控制器。此專題主要設計的是高度的控制器，roll、pitch 及 yaw rate 的控制器是由實驗室的學長提供，我們僅微調其中的參數。

II. 高度控制器設計

圖 2(下)為高度控制器方塊圖。首先，控制器會將目標高度與實際高度的差乘上固定的增益(gain)，轉變成目標的向上速度。接著再將目標的向上速度與實際向上速度的相減，作為 PID 控制器的輸入。

III. 高度控制器參數調整方法

在實務上，系統常是未知的，在設計控制器前，我們必須先做系統鑑別(system identification)，利用 MATLAB 中的 system identification toolbox，輸入實驗取得的輸入輸出資料，可以快速的得到 MATLAB 對系統的預測。如果事前對系統的動態有一定的認識，例：極點與零點的數量，則 MATLAB 的預測可以越準確。

我們假設高度對油門的系統是一個二階的系統，這種系統架構針對 step 訊號輸入不會出現穩態誤差，設計控制器也相對容易，因此我們選用這種架構做為系統的假設。高度數據對應到的是由光達測距模組所量出的高度讀數，油門則是對應到 PWM 數值。為了得出開迴路的飛行資料，我們使用無人機的手動飛行模式，使用遙控器手動操控，取得一段飛行的數據。

將濾波過的數據匯入 system identifier，除了上述假設的二階系統，我們也

額外嘗試了其他幾種極零點數量組合，最後 MATLAB 估算出最符合原數據的 system transfer function 為：

$$\frac{-0.012s^2 + 0.0048s + 0.00081}{s^2 + 0.07746s + 0.1014}$$

與濾波後的原數據比對後，估算出來的系統有接近 70% 的正確性。

最後我們根據此系統設計 PID 控制器，但試飛結果並不如預期，無人機出現不穩定的劇烈震盪。我們認為原因在於整個系統十分複雜，因此我們得到的 transfer function 不能有效代表真實系統。首先，系統是非線性的，除了 ground effect 外，PWM 的 duty cycle 和高度不是線性的關係。此外，系統也是時變 (time-variant) 的，例如電池的電壓會隨著飛行時間增加而下降，使得在相同的 PWM duty cycle 下，螺旋槳提供的推力會隨著時間下降。

由於我們無法得到有效的 transfer function，我們後來改用直覺、經驗及試誤法當作調整參數的方法。需要調整的參數總共有四項，分別是控制高度的 $k_{p,alt}$ ，及控制速度的 $k_{p,vel}$ 、 $k_{i,vel}$ 及 $k_{d,vel}$ 。在調整 $k_{p,alt}$ 時，此項的物理意義是將目標及量測高度差轉化為目標的向上速度。因此在調整此項時，最主要的依據是我們會設定當無人機與目標高度的差距是多少時，無人機要以多快的速度追到此差距。例如我們會設定當高度差是 50cm 時，無人機需要以 50cm/s 的速度追上目標，則 $k_{p,vel}$ 會設定成 1。

至於 $k_{p,vel}$ ，此項是將速度差轉換成 PWM 的 duty cycle。在調整之前，我們要事先猜測初始的數值。因此我們先量測在飛行中通常的向上及向下的速度為何，以及 PWM duty cycle 大致的變化範圍，以此預估大致的參數值。在微調時則是會逐步加大 $k_{p,vel}$ ，直到無人機產生如上下震盪等較為不穩定的狀況前為止，以增加控制器對速度差的反應能力，但又不會過於不穩定。

至於 $k_{i,vel}$ ，積分項能消除 steady-state error，然而過大的 $k_{i,vel}$ 會使得系統的 overshoot 增加。因此我們在調整時，是盡量增加 $k_{i,vel}$ ，但又不會增加到使系統產生過大的 overshoot。在我們實驗過程中，我們發現此項對於解決電池電壓隨時間逐步下降有幫助，積分項能夠逐漸增加以抵銷電壓下降的影響。

最後，因為調整完 $k_{p,vel}$ 及 $k_{i,vel}$ 後，我們認為無人機的反應已經符合我們要求。因此我們沒有完整的調整 $k_{d,vel}$ ，僅設定 $k_{d,vel}$ 為一個小的數值。

1.3 無人機飛行模式設計

除了控制器本身，我們有設計幾個飛行的模式供無人機使用，大致可以分為自動模式、定高模式及手動模式。在自動模式下，無人機會依序依照事先設定好的數個高度目標飛行，一旦完成所有高度目標，無人機會自動降落。在定高模式下，無人機會起飛至事先設定好的一個高度目標，並且在達到此高度目標後，使用者可以改變目標高度。最後使用者能輸入降落的指令，而無人機會自動降落。在手動模式下，定高的控制器並不會運作，完全由使用者決定馬達的控制訊號。

1.4 地面站

在此專題之前，無人機僅能由人為操作遙控器飛行。而我們希望除了遙控器外，也能透過 PC 端的地面站軟體控制無人機的飛行。我們使用 Mission Planner，能提供無人機飛行路徑規劃、參數調整及無人機控制面板等功能。此軟體與無人機的溝通方式是透過 MAVLink 傳輸協定。圖 3 是此軟體實際操作畫面。我們實作的功能主要分成 4 項。遠端調整控制器參數，不必重新燒錄程式至開發板上，加快控制器開發速度。規劃無人機路徑，透過地面站事先規劃數個高度目標，並在飛行前傳送至無人機端，使無人機自動完成這些高度目標，並於完成後自動降落。於飛行中手動調整飛行高度，及即時回傳飛行資料如高度、姿態、控制器資料等，除了能即時監測無人機的飛行狀態外，收集下來的資料也有利於控制器參數調整。



圖 3: 地面站操作介面

二、 實驗環境與結果

2.1 測試環境

I. 滑軌平台

因為在參數不確定的情況下，直接讓無人機自由飛行具有相當的危險性，所以一開始我們使用束帶將無人機固定在滑軌上進行測試，並設有鐵架阻隔確保安全，如圖 4(左)所示。在這種測試平台上，我們能夠實現十分穩定的定高功能，然而出於滑軌載重、高度限制、氣流效應等考量，無人機無法良好的重現自由飛行下的情況，因此我們設計了第二種測試平台。

II. 旗桿平台

在二種平台中，我們將無人機的四腳用細繩綁在旗桿上，如圖 4(右)所示：此種測試平台的好處是無人機處在的環境開放，且能進行約 150 公分以下的定高飛行。經過實際的測試後，此平台較能還原自由飛行的環境。在此平台上調整而得的控制器參數也能應用於自由飛行。此平台的缺點在於桿子與細繩間的摩擦力較大，需在桿子上塗潤滑油以降低摩擦力。

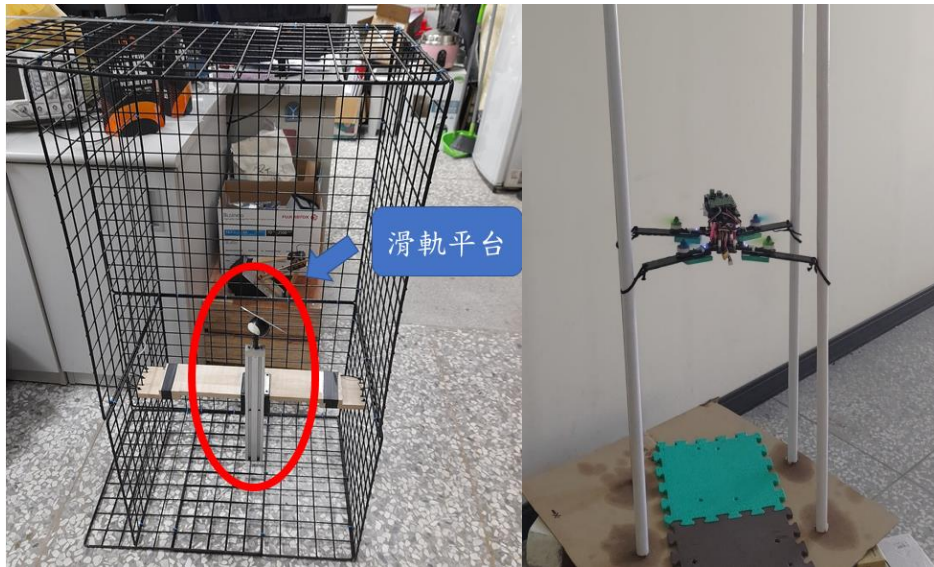


圖 4:滑軌測試平台與旗桿平台

2.2 實驗結果

我們的實驗結果分成兩個部分，一是固定飛行高度，如圖 5(上)。第二種是設定數個不同的目標高度讓無人機追至目標高度，如圖 5(下)。在第一部分中，可以看到在無人機達到目標高度後，大多數時間能夠穩定維持在目標高度。而在第二部分中，可以看到無人機除了能夠上升至目標高度，也能上升下降到其他目標高度，且在數個高度間切換皆沒有太大的 overshoot，十分穩定。然而在圖 5(上)約 70 秒圈選處及圖 5(下)約 140 秒圈選處，無人機出現突然下降的情況。我們認為此現象並非高度控制器運作異常，因為控制器在之後仍將無人機修正至目標高度。我們認為可能的原因為供電或是電子調速器(ESC)異常所致，但目前仍無法確切的釐清問題來源。

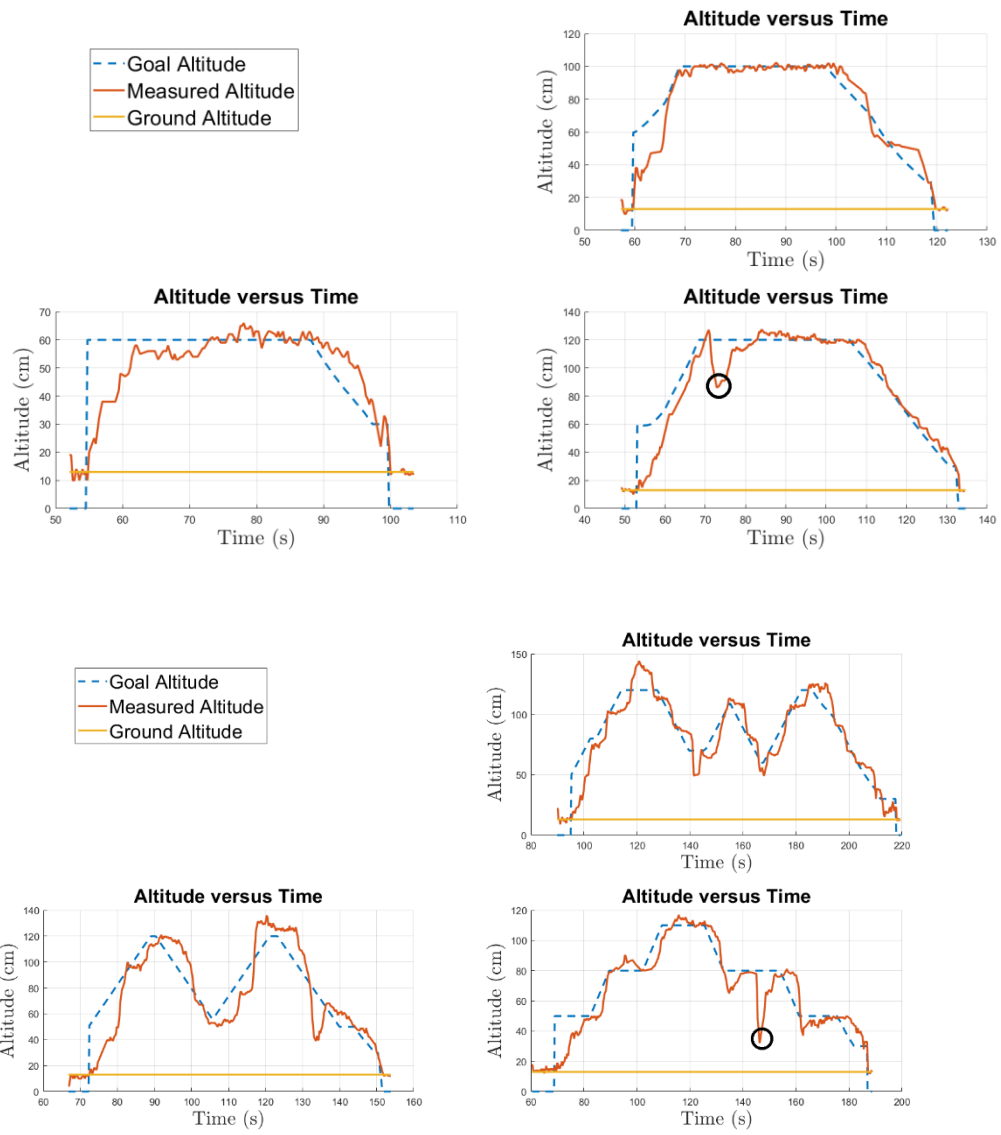


圖 1: 固定目標高度(上)及即時改變目標高度(下)之高度數據

心得感想

這個專題是實驗室近年開始的題目，我們認為最大的挑戰是我們的無人機可以說是從零開始打造，都是學長們一個個把馬達、旋翼、供電系統等組裝上而成的，最核心的飛控模組也是用開發版替代。雖然我們接手時已經有已經成型的無人機了，但在飛行表現上還是極度依賴人工遙控修正，操作上非常不便，除了隨時調整 roll(翻滾)、pitch(俯仰)要修正水平飄移，還要兼顧油門維持高度。在開發的過程中，我們也遇到不少軟硬體的問題，像是在傳輸資料時程式有可能會卡住，一旦卡住馬達的 PWM 就會中斷，緊接著就是墜機。感測器也不是隨時都是準確的，尤其是放在馬達高速運轉的無人機上時。畢竟這個題目才剛開始，我們和學長們都只能慢慢摸索，互相交流來找出解決方案。很感謝學長們的熱心幫忙，在調校感測器時提供我們不少建議，飛機多次失控摔壞，他們總能很快的修好給我們做下一次的試飛。

設計實驗上我們也花了許多心思，因為螺旋槳旋轉的速度極快，只要稍微碰到就可能嚴重割傷，如何在限制無人機自由保證人員安全的同時做出貼近現實的測試環境是一大課題。途中我們也歷經了多次的試作，嘗試了多種素材，才作出最後的平台。因為如此，這次專題沒有人受傷，製作平台也累積了不少實作經驗。

最後在我們開發定點功能陷入瓶頸時，老師建議我們在所剩不多的時間中把我們有的定高功能作好，與地面站結合作出完整的應用。也多虧老師的建議，在最後讓我們有個方向，做出不愧付出努力的成果。