105 年「機電實作專題成果展構想書」

作品名稱：四軸飛行載具室內建模與定位

中山大學/ 機械與機電工程學系/ 蘇傳翔/ B023022032
中山大學/ 機械與機電工程學系/ 張楚涵/ B023022058
中山大學/ 機械與機電工程學系/ 李鈺謙/ B023021035
中山大學/ 機械與機電工程學系/ 鄭郁潔/ B023021044
指導教授：劉耿豪

一、摘要

災害的發生經常導致建築物的崩塌和內部結構改變，造成搜救行動上的困難，因此無人搜救載具的室內定位能力變得尤其重要。然而最常見的定位系統 GPS(Global Positioning System)並不適用於室內環境，考慮到搜救的機動性和救援效率，本次研究以四軸飛行器(Quadrotor)為主體，搭載微型電腦和 Xtion pro live 景深相機，配合機器人作業系統 ROS 及 RTABMAP、Moveit!、Rviz 和 Mavros 等 Packages，藉由測距自主規畫路徑與 3D 建模(3D Modeling)，克服空間上的限制和室內定位的困難，達成救援前先行探測災後環境的目的。

關鍵詞: 四軸飛行器、路徑規劃、室內 3D 建模、室內定位

二、動機與目的

因應台灣地震發生頻繁，包括此次 2 月 6 日地震造成台南市 117 人死亡。在觀察許多新聞播報救援過程中，救難人員總是因不了解倒塌建物的內部環境，進而造成救援效率不高，使受難者生還機率低。考慮到這些因素，我們便想到是否有方法能夠使救難人員能更迅速且方便的瞭解建物內部環境及影像，來加快救援進度，因此聯想到四軸飛行器的平台，將體積小、攜帶方便的特性應用在救災方面，必定能對救災能有助益，而不只侷限於空拍和娛樂的功用而已。

本次研究的目的主要是建構出一台專用的四軸載具，並利用四軸載具的機動性與其可操控性，搭配景深相機(RGBD Camera)，實現室內環境的 3D 建模與載具自身在室內空間的自我定位。此研究未來可應用於地震救災的先行偵搜，災難現場影像資料回傳，進而到逃生路徑的規劃。此成果不僅可以有效的減少人力的派遣，增加救援效率，也可以降低救災人員生命安全的風險，進而達到「機器人輔助人類進行救援行動」的目的。

三、研究介紹

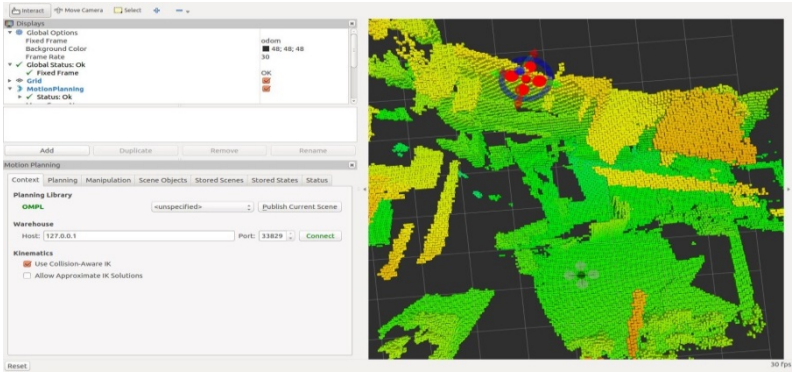
此次研究主要分為軟體與硬體兩個部分。軟體方面是以機器人作業系統 ROS 作為主架構，而我們使用的是 ROS Indigo 版本，利用 Gazebo 與 Moveit! packages 來進行模擬 3D 建模與直升機的控制，接著再應用到現實中的直升機上。硬體部分，我們所使用到的有景深相機 Xtion pro live、飛控板 Pixhawk 及微型電腦 Odroid XU4。以 ROS 架構，使用 RTABMAP 與 Moveit! packages 進行即時的攝影、建模與定位，再藉由自己寫的 controller package 透過 Mavros 對四軸飛行器進行位置控制，來實現即時 3D 建模、貼圖與飛行器路徑控制。

ROS[1]全名為 Robot Operating System，ROS 並不是一個獨立的系統而是一種作業環境，因此 ROS 須安裝在 Ubuntu 作業系統上，因為 ROS 在 Ubuntu 上有軟體庫。ROS 的運作模式為透過各個 Packages 的溝通，來完成一個完整機器人的運作模式，各個 Package 都有自己的功能，包含處理數據的節點(Nodes)，每個節點都有不

同的工作程序，像是攝影、定位、路徑規劃等，而各個 Packages 透過發佈和訂閱主題(Topic)來傳送訊息，達成彼此間的溝通。

RTABMAP[2]的全名為 Real-Time Appearance-Based Mapping，是一種基於貝葉斯(Bayesian)環路閉合檢測器，並利用 RGB-D 圖形和同步定位與地圖構建演算法 SLAM(Simultaneous Localization and Mapping)來進行 3D 建圖的方法，本此研究將 RTABMAP 搭配 Octomap[3]以八元樹(三維空間的樹狀數據結構)的形式儲存成立體地圖，調整解析度，藉此省下大量的儲存空間，透過查詢某個方塊點是否可以通過，來實現導航的功用。

Moveit![4]是由 ROS 一系列移動操作的 Packages 所組成，包含運動規劃、操作控制、感知、碰撞檢測等，本此研究利用 Moveit!來規劃路徑，將 Octomap 匯入 Moveit!，並配合 Rviz 顯示出四軸飛行器的模型及在 3D 建模中的定位，觀察其運動的狀態。



圖一:實際操作 Rviz 利用 Moveit 規畫路徑

Gazebo 是一個跨平台的獨立軟體，可以模擬真實世界中各種的模型和物體，包括物體特性的調整、環境的設置、機器人種類和數目多寡等，其執行檔能夠處理基本相關的物體運動計算和感測器的數據，並將其視覺模擬化。本次研究利用 ROS 連結 Gazebo，將設定好的資訊匯入 Gazebo，除了可以在實際測試之前模擬飛行器的運動狀況並存取資訊進行修改，也可以大大的減少線材、電池和其他材料的損耗，不用擔心因為硬體機構、馬達上的問題或者軟體問題而導致飛行器暴衝造成危險，並減少在硬體上除錯的時間。

Mavros[4]是 ROS 中由 MAVLink 擴充的節點，在本次研究中用來支持飛控板 Pixhawk 與地面站間的通訊，並利用 Mavros 的指令來控制飛行器的運行動作。

四、研究方法及步驟

在此次的研究中有兩大部分，分別為 3D 建模和四軸飛行器的控制，不過在實作之前，我們會先將建模與控制以模擬的形式進行，以確保飛行器能夠如預想的方式完成路徑。

1. 模擬

模擬的主要目的為測試軟體和硬體執行的狀態。

首先測試軟體的部分。開啟 Gazebo，並將 PX4 Firmware 與 Gazebo 裡的四軸飛行器連接上，接著開啟 Mavros，透過 Mavros 指令控制 Gazebo 裡的飛行器，並配置 RGBD camera，將 RGBD camera 建出的模型的 Transform 設定完成，使之以 Octomap 的形式顯示在 Moveit!上，再透過 Moveit!的 OMPL(Open Motion Planning Library)規劃出路徑，最後，將路徑傳給 Controller，透過 Mavros 指令來命令 Gazebo 裡面的飛行器完成路徑。

在硬體部分，利用簡單的程式執行 Mavros 命令，讓四軸飛行器起飛至兩公尺的高度接著降落，來測試 Pixhawk 在 Offboard 模式的功能，確認飛行器上硬體的執行狀態。

2. 實作

(1) 將四軸飛行器組裝完成後，安裝微型電腦及 Xtion pro Live。

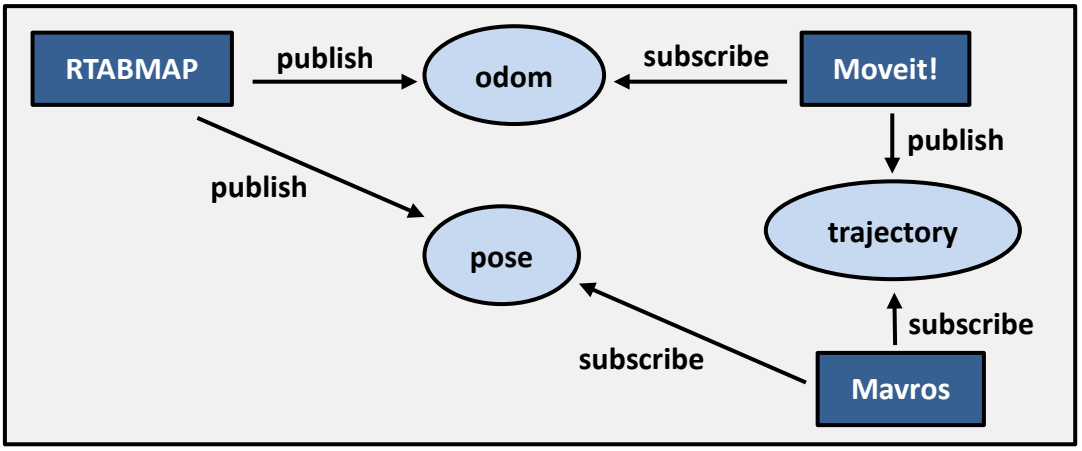
(2) 在微型電腦上安裝 Ubuntu 14.04，並安裝 ROS Indigo 和 RTABMAP、Moveit!及 Mavros 等 Packages。

(3) 開啟 RTABMAP 獲得點雲圖(point cloud)及世界固定坐標系 Odom，來完成建模。

(4) 寫一個將 RTABMAP 的 Odom(測程資訊)轉成位置資訊 Pose 的程式，因為 RTABMAP 的 Odom 的資料型態是 nav_msgs，而我們所需的資料型態是 geometry_msgs(基本上 Odom 與 Pose 相同，不過資料儲存的方式不同)。

(5) 寫一個程式 subscribe 步驟(4)程式所 publish 的 Pose 及 Moveit!規劃出的路徑(waypoint)，演算後將下一個位置 publish 給 Mavros。

(6) 利用 Mavros 命令直升機到指定位置。



圖二:從建模到控制所 publish 及 subscribe 的 Topic

圖三:本次研究所使用的四軸飛行器

五、結果與討論

ROS 是開源的機器人開發平台，在其架構下，它建立了機器人各大領域導航、操作、機構和感知的溝通平台，讓全世界的研究人員能夠共享各種機器人相關的研究資訊和演算法，且支援各式各樣的程式語言，包括 C++和 Python，如果本次的研究在應用上的可行度夠高，則其他研究人員就不必重頭寫 Controller 及 Driver，可藉此省下成本和時間並加以進行測試和修改。而現實生活中已經有利用四軸飛行器在救災現場中進行空拍的實際例子，能夠即時的回傳現場資訊和狀況，以便救災路線的規劃，四軸飛行器的應用範圍廣泛，且機動性高、體積小和攜帶方便，容易製造也方便控制，擁有很高的潛能和開發性，若應用在救災現場，可以減少人力的損耗和傷亡，也能夠即時掌握救災現場的現況和建築物內部改變後結構。

然而 ROS 是一種基於消息傳遞的分佈式多進程架構，在通信機制的部分會消耗很多系統的資源，再者，無人機最重要的地方在於其穩定性與安全性，若主節點崩潰導致訊息無法傳送，或下游節點發生問題導致系統出現錯誤，則很有可能會造成機台的失控，因此必須備份節點以防萬一，並改良其間的通訊方式以改善效能問題，而隨著技術人員不斷的改良，並配合開發者在自主導航這部分的需求，正持續的優化 ROS 的系統，而符合工業運行標準的 ROS2 也在開發中，未來 ROS 在自主定位、無人駕駛系統的性能與穩定性會更加完善，並應用在生活中。

我們在這次專題中寫得 Controller 也可能會因 ROS 的優化，需改變其資料型態。在未來，我們需要減輕飛行器的重量以增加他的飛行時間，並對所使用到的 Packages 進行參數優化，以達成運用最低效能來達成基礎運算量，增加運算效率，也可運用新上市的 RGBD camera 的體積小的特性，來減少四軸的體積及重量。

六、致謝

這次的研究能夠完成，首先要感謝的是西班牙馬德里科大 Vision4UAV 實驗室的 Fu Changhong 博士帶領我們進入 ROS 的領域，並激發我們學習 ROS 的相關知識；並感謝成大航太系的林浩銘同學，在本次研究過程中遇到難關時，帶我們了解 RTABMAP 及 controller 的傳輸規則並更加深入 ROS 的架構，因為有他們的協助，使得研究更加完美，對於第一次接觸這 ROS 系統架構的我們是很大的助力。

七、參考文獻

【1】Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., ... & Ng, A. Y. (2009, May). ROS: an open-source Robot Operating System. In *ICRA workshop on open source*

【2】RTABMAP,”http://wiki.ros.org/rtabmap”

【3】Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, *34*(3), 189-206

【4】Moveit!,”http://moveit.ros.org/”