

不確定的環境中無人機的階層式協力運動規劃

蔡苡雋

國立政治大學 資訊科學系
台北市文山區指南路二段 64 號
107753044@nccu.edu.tw

李蔡彥

國立政治大學 資訊科學系
台北市文山區指南路二段 64 號
li@nccu.edu.tw

摘要—路徑規劃一直是無人機自動化研究中重要的課題，其目的在於確保無人機的安全性及效率，在移動中隨時更新與即時規劃路徑，讓無人機順利到達目標點。無人機的機上運算對於複雜或廣大的區域，需要消耗很多計算時間，而對於無人機的安全性來說，需要較短的規劃時間來達到飛行安全的目的。在本論文中，我們以快速搜索隨機樹(RRT)路徑規劃演算法為基礎，提出了一種階層式協力計算架構，針對機上與基地台不同的運算能力，設計非同步合作規劃，嘗試解決在單層路徑規劃上較難同時確保規劃即時性與路徑最優性的問題。本系統可用於無人機行駛在不確定的環境中即時規劃路徑，並確保無人機在飛行時的安全，及產出有效率的路徑。我們以四種不同環境條件進行即時模擬飛行實驗，實驗結果顯示本架構可有效降低飛行花費時間或是飛行路徑長度，並且能在有提供風場資訊的環境中，選擇更順風的路徑飛行。

關鍵詞：無人機、運動規劃、路徑規劃、快速搜索隨機樹

I. 簡介

近年來無人機的全球市場大幅成長，現已成為商業、政府和消費應用的重要工具，能夠支持諸多領域的解決方案，例如物流運輸、農業植保、安防救援、資訊蒐集、巡檢勘查、中繼通訊及影視攝影等。而路徑規劃是無人機自動化研發中具挑戰性的問題之一，因為無人機進行任務的環境往往是部分未知或容易變動的地區，對於複雜或廣大的區域來說，往往會消耗很多計算時間，而對於即時規劃(Real-time planning)來說，亦需要較短的規劃時間來確保安全性，所以基於路徑長短、避障安全以及計算時間的考量下，一個好的路徑規劃演算法與自主規劃模型就顯得格外重要。

目前無人機動態路徑規劃主要有兩種方式，第一種是由無人機上的傳感器(Sensor)蒐集環境數據資料，透過通訊方式傳回基地台(Base station, BS)運算出全局的(Global)最佳路徑，再將路徑指令傳給無人機執行。但由於計算時間過長或是傳輸的延遲，導致無人機有可能無法即時避障，而可能有安全上的問題。第二種方式是在無人機上進行機上運算(Onboard computing)來自主規劃路徑，但因為機上計算力一般較為不足，很有可能無法在寬廣或複雜的三維環境中即時地運算出全局最佳路徑，而只能計算出局部的(Local)最佳路徑。這種只規劃局部路徑的規劃模式又稱為滾動窗口(Rolling window)規劃[1]，雖然能夠即時規劃，保障無人機的安全性，但會有陷入局部最小值(Local minima)的問題，無法運算出確保可到達目的地的路徑。

為解決以上問題，本論文以能夠做機上運算的無人機為對象，開發一套自主路徑規劃系統，配合能與之通

訊的基地台，在不確定的三維環境中，同時考量環境中的障礙物以及風場，使無人機能自主規劃起點到終點的路徑，並搭配無人機與基地台的合作規劃架構，結合遠端規劃與機上規劃的優點，在行駛的過程中利用機上運算進行即時規劃來確保安全性，同時基地台產出近似最優的全局路徑，以供無人機在線上即時修正，使無人機既可跳脫局部最小值，面對可能變化的環境，又可確保即時安全性，以完成飛行任務。本研究最後會透過實驗模擬環境驗證所設計系統之有效性。

II. 相關研究

A. 運動規劃

運動規劃(Motion planning)是機器人學(Robotics)中常見的研究問題，用於搜尋能將機器人從起始組態(Start configuration)移動到目標組態(Goal configuration)的合法組態序列(Sequence of valid configurations)。而一個基本的運動規劃問題則是計算一條連接起始組態到目標組態的連續路徑，同時避免與已知障礙物發生碰撞。連接起點位置和終點位置的序列點稱為路徑，構成路徑的策略則稱為路徑規劃。

根據環境的資訊及問題廣度，運動規劃分為全局運動規劃(Global motion planning)和局部運動規劃(Local motion planning)。在全局運動規劃中，環境的信息多是全部已知的，因此我們常把全局運動規劃稱為靜態或離線(Offline)運動規劃，其規劃之準確性取決於能取得環境資訊的準確度。全局運動規劃可以求得最優解，但是需要預先知道環境的準確資訊，當環境發生變化或會遇到未知障礙物時，此方法可能就無法適用。在局部運動規劃中，環境信息可以是全部未知或部分已知，因此局部運動規劃稱為動態或線上(Online)運動規劃，重點在考慮機器人當前局部環境中的信息，讓機器人可以立即反應環境變化並具有良好的避障能力，但此方法需要機器人擁有環境信息的探測與處理能力，以及較高的機上運算速度，才能夠針對環境變化進行動態規劃，確保機器人的安全性。而局部運動規劃由於使用局部環境信息，因而無法得到全局最優的結果，甚至會因為局部最小值問題而無法到達最終目標空間。

B. 路徑規劃演算法

路徑規劃的演算法有很多[2]，根據其自身優缺點，其適用範圍也各不相同。低維的問題可以利用幾何算法解決，或是將網格覆蓋在組態空間(C_{space})中，將空間離散化，並利用搜尋演算法(Search algorithm)進行路徑規

劃。高維空間的問題在計算上較為棘手的，工作空間中的人工位能法(Potential-field)[3]雖是有效的經驗法則(Heuristic)，但是會有局部最小值的問題。基於採樣的演算法(Sample-based algorithms)則有可能避免局部最小值的問題，也能解決高維的問題，雖然僅具隨機完整性(Probabilistic Completeness)(無法確定不存在任何路徑)，但是隨著花費的時間越來越多，找到解答(如有)的機率會提高到 1.0。目前基於採樣的算法被認為是高維空間中運動規劃較有效的技術。

C. 快速搜索隨機樹

快速搜索隨機樹(Rapidly-exploring random tree, RRT)[4]是近年來發展起來的路徑規劃演算法，屬於基於採樣的路徑規劃方法，透過在組態空間隨機採樣中逐步建構搜尋樹，使搜尋樹逐漸覆蓋未探索的區域，最後到達目標點，找到一條合法的路徑，此方法由於避開了對全局空間的建立，在高維空間與複雜環境的表現優於其他演算法(如: A*、BFS、Potential-field)，因此本研究選擇快速搜索隨機樹演算法作為路徑規劃的方法。

III. 系統架構與設計

本研究所設計之系統架構與流程如圖 1 所示，系統一開始會先進行初始化，包括參數設置、環境圖建置等，再進行預規劃(Preplanning)，產生出初始的全局路徑供無人機執行，接著會進入程式主階段，分為全局規劃(Global Planning)與局部規劃(Local Planning)兩個程序迴圈，全局規劃交由基地台負責規劃全局路徑、平滑化路徑以及根據無人機回傳的資料更新環境變化，持續將最好的路徑結果回傳給無人機；局部規劃交由無人機利用基地台傳回的全局路徑來規劃局部路徑，局部規劃的範圍小，是

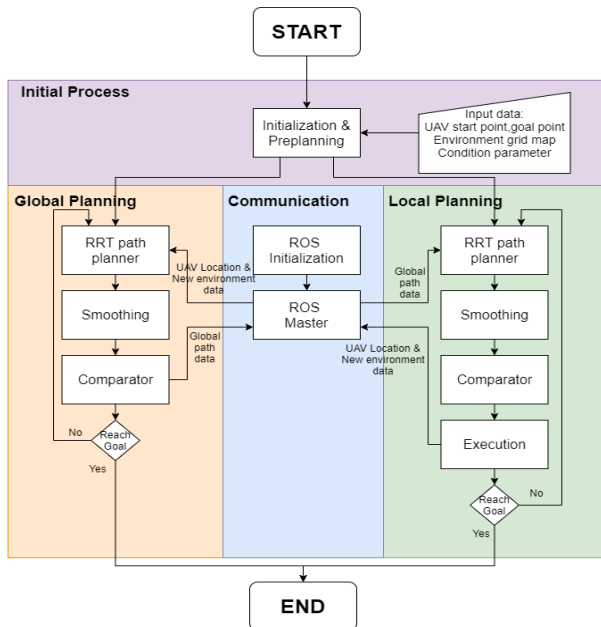


圖 1 系統架構與程式流程圖

根據無人機上的傳感器感測範圍而定，其計算頻率高，目的在於確保無人機在不確定的環境中能夠更即時的更新規劃來達到安全性，局部路徑平滑化後會與原本的全

局路徑做比較，並根據比較結果來修正路徑。路徑決策完後便會執行路徑，執行路徑的同時也會感測周圍的環境，並將最新的環境資訊以及無人機位置持續傳回給基地台做後續的全局規劃。

本系統通訊方式交由機器人作業系統(ROS)處理，ROS 通信模組是一種實現模組間 P2P 的鬆耦合(Loose coupling)網絡連接的處理架構，通訊協議使用標準的 TCP/IP 接口，能提供節點(Node)間基於 Topic 的非同步(Asynchronous)數據流通訊，而本系統分別將基地台與無人機作為節點，並相互發布以及訂閱。

此階層式合作規劃架構的好處在能讓無人機於保持動態規劃的同時，也兼顧全局的最優性，局部規劃可以快速反應環境變化，即時修正全局路徑來達到更高效率以及安全性，而全局規劃能以全局的視野來提供無人機全局較優的路徑，幫助無人機脫離局部最小值。

以下將會針對本論文所提出之階層式合作架構中幾個關鍵模組做說明。

A. RRT 路徑規劃模組(RRT path planning)

此模組用於規劃無人機路徑，其演算法如圖 2 所示。

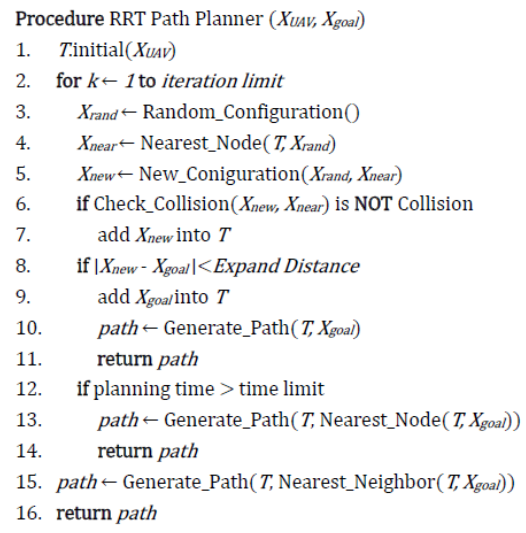


圖 2 路徑規劃演算法

搜尋樹 T 的根節點為原點 X_{UAV} ，接著會進入迴圈擴展搜尋樹，擴展的方式為先在組態空間中隨機找一個點 X_{rand} ，並在搜尋樹中找尋與它最近的點 X_{near} ，從 X_{near} 往 X_{rand} 的方向生長一個 ϵ 單位，形成新節點 X_{new} ，在確認完 X_{new} 為安全的點後，就會將新節點 X_{new} 加入搜尋樹 T 中，新節點會考慮與障礙物距離、路徑前進方向、風場向量，來決定此點是否安全。能夠跳出迴圈的狀況有三種，第一種為順利搜尋到目標點，當新節點與目標點 X_{goal} 非常靠近時，如果兩點間沒有障礙物，就會將目標點 X_{goal} 加到搜尋樹，並產生完整路徑。第二種與第三種狀況分別是超過搜尋次數限制以及時間限制，由於無人機是以動態規劃來運作，因此需要對規劃時間做限制，超過限制時，會以距離目標點最近的節點來產生路徑。產生路徑的方法為以輸入的節點當第一個節點，開始呼叫其父節點，並以此父節點繼續呼叫下個父節點，重複此步驟直到父節點為根節點，將這些節點串連便形成了一條路徑。

B. 平滑模組(Smoothing)

平滑模組是接在 RRT 路徑規劃模組後，由於 RRT 產生的路徑不會是平滑的路徑，因此需要平滑模組的幫助，平滑化使用基於插植的路徑平滑，主要參數有更新資料權重 α 、平滑權重 β 以及保持疊代條件 Tolerance，平滑演算法如圖 3 所示。

```

Procedure Smoothing (path)
1. smooth path ← path
2.  $\alpha$  ← weight_data
3.  $\beta$  ← weight_smooth
4. change ← tolerance
5. while change ≥ tolerance
6.   change ← 0
7.   for i ← 1 to len(smooth path)
8.     for j ← 0 to 2
9.       odd ← path[j][j]
10.      new ← smooth path[j][j]
11.      new_pre ← smooth path[j-1][j]
12.      new_next ← smooth path[j+1][j]
13.      new_saved ← new
14.      new ← new +  $\alpha$  * (odd - new) +  $\beta$  * (new_pre + new_next - 2 * new)
15.      if Check_Collision(new_pre, new) is NOT Collision
16.        if Check_Collision(new, new_next) is NOT Collision
17.          if Get_Wind_Score(new_pre, new, new_next) ≥ Get_Wind_Score(new_pre, new_saved, new_next)
18.            smooth path[j][j] ← new
19.            change ← change + |new - new_saved|
20. return smooth path

```

圖 3 路徑平滑演算法

演算法中 *path* 為原始路徑，*smooth path* 為欲輸出之平滑路徑， α 為更新資料權重， β 為平滑權重，兩者皆用於使用梯度下降法來調整平滑速度，Tolerance 為疊代條件，用於 while 迴圈判斷式，迴圈會重複的平滑化路徑，直至某次路徑平滑改變量小於 Tolerance，便會輸出 *smooth path*。演算法的關鍵在於第 14 行的公式，平滑化的新節點改變量為 $\alpha * (old - new) + \beta * (new_{pre} + new_{next} - 2 * new)$ ， α 控制保留上次平滑結果的權重， β 為控制每次平滑量的權重，當新節點計算完成後就會檢查此點與前後兩點間是否有障礙物，如果安全且此點的風場分數大於原本的點，就會將新節點取代掉舊節點。

C. 比較模組(Comparator)

比較模組用於比較兩條路徑的優劣勢，此路徑比較演算法如圖 4 所示。

```

Procedure Comparator (path1, path2, XUAV)
1. Xpath1 ← Nearest_Node(path1, XUAV)
2. Xpath2 ← Nearest_Node(path2, XUAV)
3. L1 ← Get_Path_Length(Xpath1, path1) + |XUAV - Xpath1|
4. L2 ← Get_Path_Length(Xpath2, path2) + |XUAV - Xpath2|
5. if Check_Collision(Xpath1, path1) is Collision
6.   L1 ← L1 + penalty score
7. if Check_Collision(Xpath2, path2) is Collision
8.   L2 ← L2 + penalty score
9. W1 ← Get_Wind_Score(Xpath1, path1)
10. W2 ← Get_Wind_Score(Xpath2, path2)
11. if L1 - W1 < L2 - W2
12.   return path1
13. else
14.   return path2

```

圖 4 路徑比較演算法

本研究將風場向量圖用於提升路徑品質，在路徑規劃時能夠選擇相對順風的路徑，以降低無人機在飛行中消耗的能源。此模組會先各在兩條路徑中找到與無人機最靠近的一點，並與路徑相連接，形成一條從無人機開始到目標點的路徑，並計算其路徑節點間歐氏距離加總變為路徑總長(L1、L2)，接著檢查此路徑上是否有障礙物，如果有則在路徑長上加上一個處罰分數，再計算路徑的風場分數，其分數越高代表此路徑越順風，最後會比較兩條路徑的長度與風場分數，並傳回較好的路徑。

風場分數的計算方式如下：

$$S_w = \sum_{i=1}^n (V_d \circ \frac{V_w}{|V_w|}) \times \frac{area}{weight \times 1600} \quad (1)$$

在其公式中，風場分數(S_w)為路徑上各節點無人機的飛行速度與風速向量內積的加總，並帶入風力計算所需的無人機相關參數(如無人機面積及重量)。風力計算的假設是在標準狀態下(氣壓為 1013hPa，溫度為 15°C，空氣單位體積重量 $r=0.01225$ kN/m³，重力加速度 $g=9.8$ m/s²)，根據風壓所得到的加速度。由於我們要比較的是兩個路徑相對的順風程度，最後一項對特定飛機而言是常數，因此也可以省略。

D. 執行模組(Execution)

在得到路徑後，無人機將負責決策路徑、執行路徑與感測環境等工作。在執行路徑前，會先判斷無人機的全局路徑是否安全，當全局路徑不再安全時，會改用局部規劃的路徑，而全局路徑安全時，會比較局部路徑與全局路徑，並選擇較優者。執行路徑在模擬環境中，會計算無人機速度在此路徑中下一秒會前進的位置座標，並感測無人機可偵測範圍內的環境資料，返回新的環境資訊與無人機最新位置。

IV. 實驗設計與結果分析

本研究在模擬環境中進行實驗測試，基地台使用環境之作業系統為 Ubuntu 16.04，ROS 版本為 ROS Kinetic Kame。而無人機的機上運算將以筆記型電腦做模擬取代，同樣使用 Ubuntu 16.04 以及 ROS Kinetic Kame，並將計算速度設為基地台的十分之一，實驗結果則透過 matplotlib 提供即時的資料視覺化來顯示障礙物與路徑，驗證路徑正確性以及效率。

本論文所提出架構為 Collaborative Planner，是利用基地台與無人機上的合作來規劃路徑，而與本論文所提出之架構比較的對照組共三種，第一種為 BS Control Planner，是大多無法機上規劃的無人機使用的模式，第二種為 UAV Control Global Planner，主要為機上運算能力強的無人機使用的模式，第三種為 UAV Control Local Planner，主要用於機上運算能力不高的無人機，只專注於局部範圍內的路徑規劃。

測試環境設計如圖 5 所示，共有四種測試環境模型，環境內所有的障礙物模型都以長方體組成，黃色方塊為未知障礙物，需要等待無人機靠近時才會被感測到，紅色方塊則為已知障礙物，藍點為起始點，黑點為目標點。

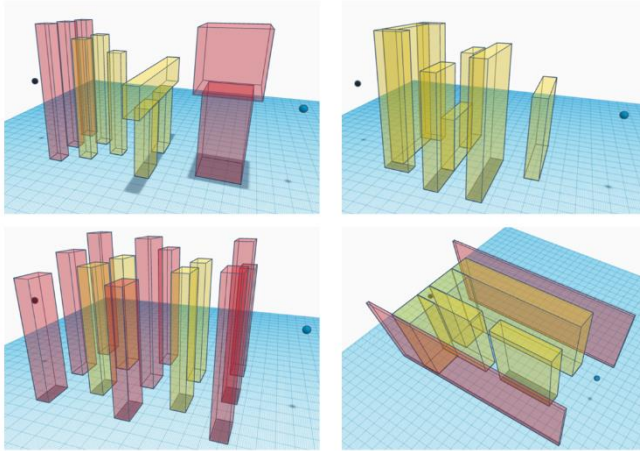


圖 5 測試環境設計圖

左上:實驗 1 右上:實驗 2 左下:實驗 3 右下:實驗 4

實驗中基地台與無人機通訊延遲設為固定的 0.5 秒，無人機上的傳感器感測距離設為 20 公尺，無人機飛行最高速度設為每秒 10 公尺，無人機與障礙物安全距離設定為 2 公尺。實驗進行時，如果沒有合法的路徑供無人機行駛時，會讓無人機在原地等待，等待的時間將會計入無人機停滯時間，並影響總花費時間，實驗會在四個測試環境中對每個實驗組做 20 次實驗並取其平均值。

表 1 測試環境 1 實驗結果

	總花費時間(停滯時間)	總路徑長
Co Planner ¹	36.95 (0)	233.617
BS Planner ²	37.98 (0.51)	234.955
UAV G Planner ³	43.31 (8.15)	220.253
UAV L Planner ⁴	36.39 (0)	231.642

表 2 測試環境 2 實驗結果

	總花費時間 (停滯時間)	總路徑長
Co Planner ¹	57.12 (0.82)	352.733
BS Planner ²	67.05 (1.73)	399.512
UAV G Planner ³	95.89 (16.05)	319.402
UAV L Planner ⁴	>100 (未完成)	175.457(未完成)

表 3 測試環境 3 實驗結果

	總花費時間 (停滯時間)	總路徑長	風場分數
Co Planner ¹	52.59 (0)	326.238	150.899
BS Planner ²	55.24 (0.25)	339.454	128.748
UAV G Planner ³	48.71 (2.35)	286.057	35.669
UAV L Planner ⁴	46.39 (0)	199.901	89.901

¹ Collaborative Planner

² BS Control Planner

³ UAV Control Global Planner

⁴ UAV Control Local Planner

表 4 測試環境 4 實驗結果

	總花費時間(停滯時間)	總路徑長
Co Planner ¹	139.88 (9.87)	724.281
BS Planner ²	137.69 (13.46)	672.850
UAV G Planner ³	314.13 (213.13)	600.472
UAV L Planner ⁴	>350 (未完成)	171.664(未完成)

綜觀四個實驗數據(表 1~4)，可以證實 Collaborative Planner 及 BS Control Planner 在總花費時間是遠優於 UAV Control Global Planner 及 UAV Control Local Planner，而 Collaborative Planner 又較優於 BS Control Planner，並且在有提供風場資訊的環境中(如實驗 3)，系統自動選擇更順風的路徑做飛行。

V. 結論與未來研究

本研究以通訊規劃及機上規劃的合作為目標，完成一套合作式路徑規劃系統，在不確定的環境中，保持動態規劃讓無人機安全到達目標點。在演算法方面，我們使用 RRT 路徑規劃演算法，根據三維環境佔用網格圖與風場向量圖，以及無人機前進向量作為計算考量，在極短的時間內計算出近似最佳解。在系統架構中，我們提出了一種階層式協力計算架構，透過無人機與基地台的階層式合作規劃，將局部規劃交由無人機確保安全性與即時性，全局規劃交由基地台來控制路徑的最佳性，此模式結合局部規劃與全局規劃的優點，並補足互相的缺點。最後透過實驗結果的觀察，驗證本論文提出的方法在與其他三種方法相比之下，可以有效降低花費時間以及路徑長度，並在風場環境中選擇較優的路徑行駛。

本系統目前只針對無人機單純的兩點移動做路徑規劃，尚未對特定的任務如物流運輸、資訊蒐集、巡檢勘查、影視攝影等應用做規劃考量，未來本系統可朝向應用導向發展。多機的協同可提升探索效率，未來也能加入多機合作規劃。

致謝

本研究經人工智慧普適研究中心 (PAIR Lab) 授權科技部計畫「前瞻無人機智慧系統：核心技術研發與情境應用展示 (4/4)」執行，於計畫編號 MOST110-2634-F-004-001 補助下完成，特此致謝。

參考文獻

- [1] K. Liang, C. Zhao., and J. Guo, "Path Planning Based on Fuzzy Rolling Rapidly-exploring Random Tree for Mobile Robot," School of Computer Science and Technology, NUST, Nanjing 210094, China, vol. 34, no. 5, pp. 642-648, 2010.
- [2] L. Yang, J. Qi, J. Xiao, and X. Yong, "A literature review of UAV 3D path planning," in Proceeding of the 11th World Congress on Intelligent Control and Automation, 2014: IEEE, pp. 2376-2381.
- [3] H. Yang, Q. Jia, and W. Zhang, "An Environmental Potential Field Based RRT Algorithm for UAV Path Planning," in 2018 37th Chinese Control Conference (CCC), 2018: IEEE, pp. 9922-9927.
- [4] S. M. LaValle, J. J. Kuffner, and B. Donald, "Rapidly-exploring random trees: Progress and prospects," Algorithmic and computational robotics: new directions, no. 5, pp. 293-308, 2001.