

基於 ROS 之掃地機器人自主導航與目標點任務實現

Final Project Report for YZU 113-2 ME382A "Robot Operating System"

Name: 游竣翔
Student ID Number: 1100959

Abstract—本報告介紹一款模擬掃地機器人的自主導航系統，該系統建構於 ROS (Robot Operating System) 平台，並透過 Gazebo 模擬環境整合 URDF 機構建模、SLAM 建圖技術以及 Navigation Stack 進行路徑規劃與導航。機器人具備基本感知能力（如雷射與相機）並可於掃描建圖後，根據預設清掃路線執行巡邏任務，若接收到使用者於地圖中點選的目標位置，將會立即前往該點並清掃完後返回原點結束任務。本專案展示 ROS 在模擬環境中實現自主移動與人機互動控制的能力。

Keywords—ROS, SLAM, Gazebo, 掃地機器人, 導航

I. 專題目標

本專題旨在於 ROS 與 Gazebo 環境中建立一具備雷射與相機之自主掃地機器人模型，透過 SLAM 技術建立環境地圖，並整合導航功能，使機器人能沿固定路線清掃，並在接收到目標位置後自動前往、清掃，最後返回原點待命。

II. 系統設計

A. 硬體模擬架構

掃地機器人以 URDF/xacro 格式設計，包含：

- 差速驅動底盤 (DiffDrive plugin)
- Hokuyo 雷射掃描器 (Laser plugin)
- 攝影機模擬元件 (Camera plugin)

B. 感知與地圖建構

- 利用 slam_toolbox 建立室內地圖
- Lidar 感測器回傳掃描結果顯示於 RViz
- 使用 teleop_twist_keyboard 手動控制建圖

C. 導航模組整合

- 採用 move_base 搭配 amcl 進行機器人定位與導航
- 導入 nav2d 地圖伺服器與 costmap 靜態障礙物辨識

D. 系統架構圖

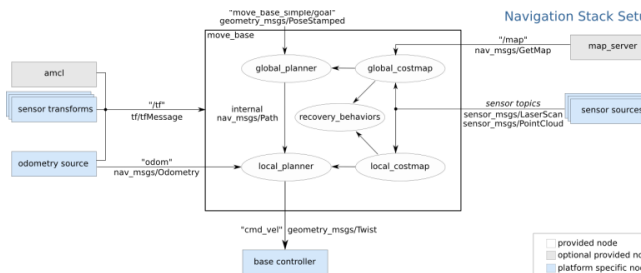


Fig. 1. System Architecture Diagram

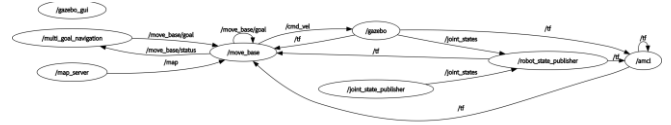


Fig. 2. System Architecture Diagram

III. 測試

測試結果與分析，驗證系統功能

A. 測試結果

自 Fusion 360 設計掃地機器人模型後，匯出並轉換為 URDF 檔案，後續整合差動輪驅動系統、Lidar 雷射感測器與攝影機模組，最終成功於 Gazebo 模擬環境中順利啟動並執行各項模擬任務，驗證整體設計的可行性與穩定性。

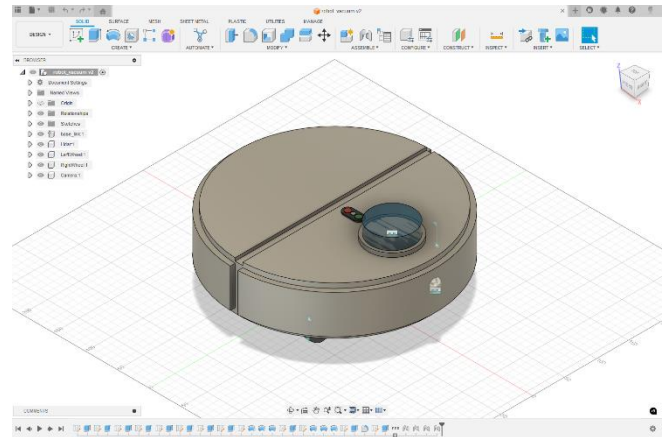


Fig. 3. Fusion-designed robot vacuum.

於 Gazebo 中整合房間的 STL 圖檔與掃地機器人之 URDF 模型，透過新增 world.launch 與 room.xacro 檔案，有效完成環境與機器人模型的載入與顯示，成功建立可供導航與模擬的虛擬室內場景。

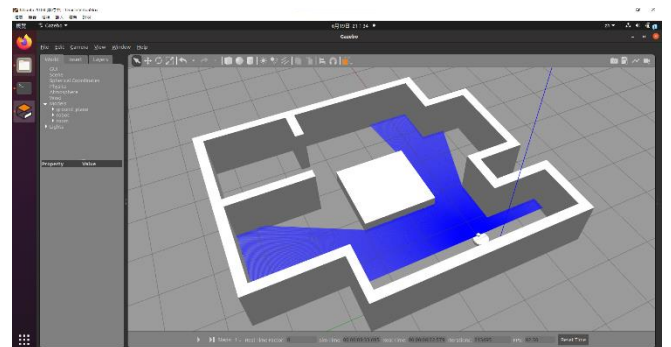


Fig. 4. Gazebo environment

使用 TurtleBot3 的 SLAM 功能對虛擬房間進行地圖建構，期間新增 slam.launch 以載入自訂場景，最終成功生成 .yaml 與 .pgm 地圖檔，以及新增 config 相關檔案與 navigation.launch 檔案，並實現在已掃描完成的地圖上移動機器人進行後續導航測試。

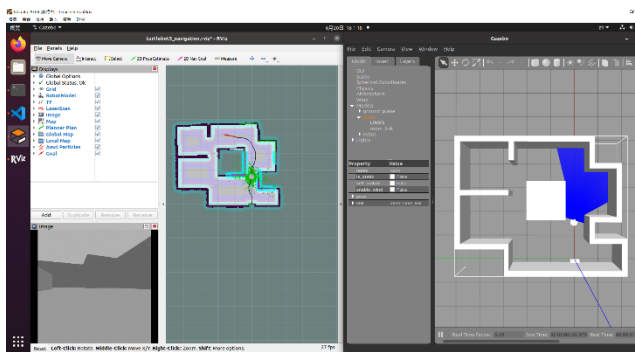


Fig. 5. Move the robot vacuum

B. 挑戰與解決方法

- TF 定義錯誤導致感測器偏移

問題：雷達資料於 RViz 中顯示與實際位置不符。

解法：修正 base_link 至 laser 之間的 TF 定義，並確認 odom 系統來源一致性。

- 地圖與機器人座標不同步

問題：導航時機器人偏離目標路線。

解法：整合 amcl 進行重定位，並優化 costmap 參數與 global/local planner 設定。

- 機器人尺寸過小導致模擬失敗

問題：URDF 定義中的機器人尺寸過小（如半徑僅 0.01m），導致 Gazebo 物理引擎無法正確計算碰撞與動態行為，進而模擬失敗或異常跳動。

解法：依實際需求重新調整 URDF 中的尺寸參數，並確保質量與慣性矩符合物理比例，重啟模擬後觀察機器人行為是否正常。

- 導航機器人路徑失敗

問題：標記 2D Nav Goal，機器人未依規劃路徑移動。

解法：待解決。

C. 網址連結

- Github :

https://github.com/jun-xiangg/robot_vacuum_description

- YouTube :

<https://youtu.be/HHn0eCB3yPI>

IV. 心得與反思

透過本學期的課程和期末專題，我深入學習了 ROS（Robot Operating System）在實際機器人應用中的整合能力，並從中體會到機器人系統設計的複雜性與挑戰。本專題的任務包含 URDF 模型建立、Gazebo 模擬環境設置、SLAM 地圖建構與 Navigation Stack 的整合，在每個階段都需要大量查閱官方文件與社群資源，逐步解決遇到的技術困難。

我特別從 TF 架構的學習中收穫良多，理解機器人各感測器與連接座標的空間對應對導航與視覺辨識的重要性。此外，在設定 AMCL 與 move_base 過程中，我學會了如何調整參數以提升路徑規劃與定位的穩定性。這些經驗讓我不僅熟悉 ROS 的架構，也更具備系統性思考與除錯的能力。

整體而言，本次專題不只是一次程式開發的練習，更讓我體會到跨領域知識整合的重要性，例如機構設計、感測器應用與軟體工程的結合。

V. 結論

本篇報告展示了於 ROS 平台上建構自主掃地機器人模擬系統的完整流程，涵蓋 URDF 模型建立、感測器整合、SLAM 地圖建構、以及導航功能的實現。透過 GAZEBO 模擬與 RVIZ 監控，機器人能在虛擬環境中完成環境探索、固定路徑清掃與使用者指定目標點的自動移動任務。專題過程中克服多項挑戰，如模型尺寸比例錯誤、TF 定義不正確與導航不穩定等問題，並透過不斷調整參數與驗證，使系統達成預期功能。本專題證實 ROS 架構的彈性與強大，有助於未來實體機器人之開發與應用。

REFERENCES

- [1] <https://github.com/rayyichen310/Final-Project-Report-for-Robot-Operating-System>
- [2] <https://github.com/scybd/Sweeping-Robot>
- [3] https://blog.csdn.net/qq_31329259/article/details/107466993
- [4] https://blog.csdn.net/qq_44339029/article/details/136671867
- [5] https://classic.gazebosim.org/tutorials?tut=ros_gzplugins
- [6] <https://robot-ros.com/robot/35376.html>
- [7] <https://weishih.wordpress.com/category/ros/>