Proceedings of the URECA@NTU 2015-16

Autonomous Indoor MAV by Simultaneous Localization and Mapping

URECA Student
Hoang Minh Chung
School of Mechanical and Aerospace Engineering

URECA Supervisor
Professor Xie Lihua
School of Electrical and Electronics Enineering

***Abstract* – This article presents a tested implementation of Simultaneous Localization and Mapping (SLAM) and path planning on indoor MAVs, that enable MAVs to fly, navigate, and avoid obstacles in GPS-denied environments without constant human pilot input. The MAV is able to estimate its 6 DOF states and perceive the surrounding using onboard sensors. All data collecting and fusing are implemented and optimized to be able to operate by low-cost and light-weight onboard computer. The paper also discusses the choices of hardware combination that prove to overcome popular MAV physical limitations and successfully operate in the tested environment.**

Keywords – Simultaneous Localization and Mapping, SLAM, Quadcopter, X8 Copter, Autonomous, Obstacle Avoidance, Path Planning

1. INTRODUCTION

The recent advent of medium and small size UAVs has opened up vast opportunities for improving quality of life. In a near future, UAVs will potentially relieve humans from the burden of small-volume transportation, wide-area search and rescue, urban surveillance, aerial photography or even personal entertainments. The advantages of these flying contraptions are popular: affordable, portable, scalable, and sometimes open-sourced. However, there are also some existing challenges that are currently limiting the applications of UAVs. One of the most challenging issues of UAVs is its limited ability to operate without constant human supervision. This shortcoming negates the possibility of the UAV to fly without line-of-sight supervision of the teleoperator, including both outdoor long range flight and indoor autonomous flight.

One of the main causes leading to such limitation is the problem of localization, the ability of the UAV to determine its position and heading. If the UAV operates outdoor, it can use commercial positioning systems such as GPS to localize and navigate between target waypoints. However, such advantage is not available indoor, where GPS signals could not penetrate through walls. Moreover, the close proximity of obstacles and walls in the indoor environments requires centimeter-accuracy localization for the UAV to navigate without collisions.

Figure 1: DR1 Alvis – X8 Version

There has been a few recent attempts to overcome the indoor localization challenge. One of them is localization by trilateration of multiple distances measured by Ultra Wide Band (UWB) radio waves from multiple known anchors in the environment [1, 2]. This method proves to be capable of high accuracy, high data rate, thanks to the simple calculation mechanism. However, just like the GPS, this method render the copter “blind” to the environment and unable to avoid obstacles or walls. Therefore, this approach is only suitable for autonomous flight when the environment is known and the mission is as simple as straight traversing from one waypoint to another. Another type of attempt is to apply Simultaneous Localization and Mapping (SLAM) algorithm with various types of sensors, such as inertial measurement units (IMU), 2D laser scanners, cameras or point cloud sensors, and estimate vehicle states with the surrounding map repeatedly. Such approach proves to be effective [3], despite being computationally expensive. Moreover, because it enables the robot to perceive its surrounding in real time, the robot is then able to sense walls and obstacles and avoid accordingly. In fact, there has been several implementations [4, 5], proving SLAM to be potentially the best solutions for indoor flight.

Another issue, specifically pertaining to UAV, is its limited ability to carry payload. Unlike its equivalents such as the Unmanned Ground Vehicles (UGVs) and Unmanned Surface Vehicles (USVs), UAVs relies on motors and propellers to hover and stabilize.

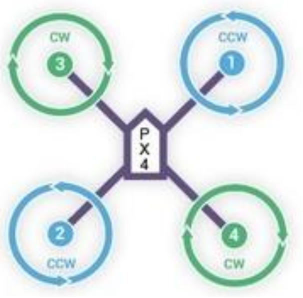

Figure 2a: X4 propulsion configuration

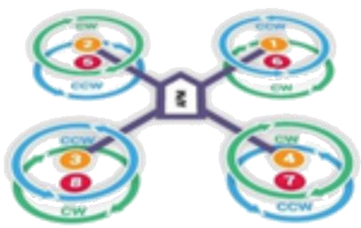

Figure 3b: X8 propulsion configuration

Figure 4: Pixhawk board

Figure 5: RaspberryPi 2 board

These motors and propellers stall at their aerodynamic limits[6] while demand high electrical power density. Moreover, indoor flight also requires the UAV size to be as small as possible to be able to go through narrow corridors or windows. These factors eventually limits the choices of sensor, quality of data, and capability of on-board processor, which ultimately poses challenges not only on the robustness of algorithms, but also on the fine combinations of hardware: copter frame construction method, power source type, sensor selection and on-board computer choice.

Therefore, this project is an attempt to realize an indoor copter using SLAM, capable of obstacle avoidance, self-position control, intelligent path planning and autonomous flight without manual supervision. This work tackles not only the flight concept and software architecture but also focuses on the hardware combinations and actual test flights, with the ultimate aim of having an operational indoor autonomous UAV.

2. HARDWARES SETUP

The copter is designed to be modular so that different combinations of hardware would still be functional and changes of one module would not affect the performance of others. This modular design approach also allows each module to be developed and optimized independently. In general, a complete system comprises of four different modules: Power, Propulsion, Low-level control and High-level control.

2.1. POWER MODULE

Power module comprises of devices that distribute and supply energy at appropriate voltage and current for other modules to function. A typical power module comprise of a lithium polymer battery (LiPo), voltage regulator, voltage sensor, current sensor and power distribution board. Besides, LiPo battery is selected to have a good combination of weight, voltage and capacity necessary for other modules to run smoothly.

2.2. PROPULSION MODULE

Propulsion module comprises of brushless motors and propellers in proper combinations to generate sufficient thrust force that lifts and manoeuvres the whole UAV. In this project, because of indoor-flying requirement, X4 and X8 configurations are preferably considered to keep the drone footprint small.

Most of the time, X4 configuration is used. However, if the payload is heavy, X8 configuration is deployed which however compromise in terms of aerodynamic efficiency. A glimpse of the X8 configuration platform, named DR1 Alvis is captured in Figure 1. Besides, the combinations of motors and propellers are evaluated based on the battery voltage, limit current, required thrust and rotation speed. In this project, the selected combination is AXI 2216/20 motor with APC 10 inch diameter propeller.

2.3. LOW-LEVEL CONTROL MODULE

Low-level control module is in charge of the high frequency control of attitude and position. The module fuses data from gyroscope, accelerometer and magnetometer and barometer at high frequency to estimate instantaneous attitude, then actuate propulsion module accordingly to achieve stable flight. The rate of control output into motors can be as high as one kilohertz, which requires a high frequency real-time control system. In this project, Pixhawk board is used for low level control because of its high frequency crystal and good capability real-time processor STM32.

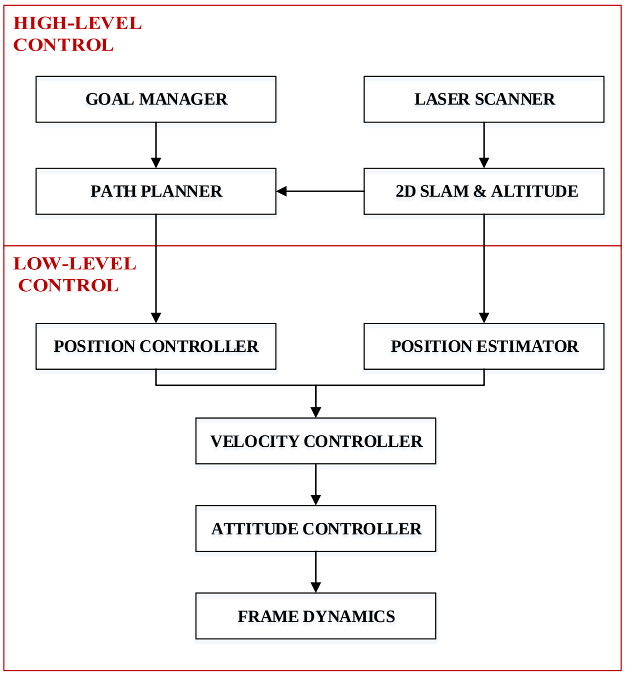

Figure 6: Overall software architecture

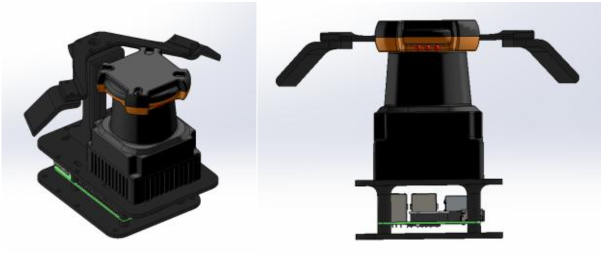

Figure 7: Dual purpose mechanism: 2D SLAM and local altitude measurement

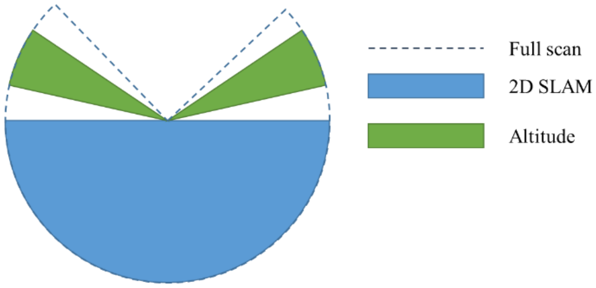

Figure 8: Laser scanner data usage (Top view)

2.4. HIGH-LEVEL CONTROL MODULE

High-level control module is in charge of computationally expensive processes that enables the drone to fly more intelligently. Specifically in this project, the high-level control collects laser range data to localize itself in an unknown environment and, with a predefined goal, try to navigate and explore without colliding into obstacles. In this project, RaspberryPi 2 board is selected for high-level control, because of its low-cost and good compatibility with Linux. The laser scanner to collect range data for SLAM is selected to be Hokuyo UTM-30LX for its low noise, 30m range, high data rate and light weight. Besides, Robotic Operating System is also deployed on Linux environment run SLAM implementation and path planning. Besides, the communication between high and low level boards is achieved via serial ports with Mavlink protocol.

2.5. FRAME CONSTRUCTION

The main frame, such as main body, main arms are designed and machined from carbon composite sheets in order to have high rigidity and durability against heavy impacts.

On the other hand, some structural members, such as propeller guards, laser scanner mount are purposely designed to be sacrificial during crashes, in order to absorb the impacts and protect the main components and electronics such as sensors and processors. Therefore, these parts are 3D printed out of PLA plastic, with low fill density in order to break upon heavy impacts. The sample construction of the X8 version is shown in Figure 1, revealing differently manufactured components such as composite arms and 3D printed propeller guards, etc.

3. SOFTWARE ARCHITECTURE

The overall software architecture is classified into two layers: low-level control and high-level control, as illustrated in Figure 5. Generally, low-level control is defined as the control mechanism for the UAV to stabilize and navigate locally, while high-level control is in-charge of conceptually abstract and computationally expensive tasks such as visual navigation, machine learning, or computer vision. Because of different software requirements, low-level processes needs high update rate, strict timing and fast through-put while high-level processes demands large memory and fast processors. Therefore, in order to optimize the performance, these different processes are run on different hardware, as mentioned in previous part of this report. However, it is not strictly implied that high-level control processes has to be done entirely on high level control hardware, or vice-versa. Some tasks such as Position Control, even though classified as low-level, could still be done on high-level hardware for ease of debugging while insignificantly affect the performance of the UAV.

3.1 GOAL MANAGER

The Goal Manager is in-charge of the overall outcome of the mission. In general, because the mission could break down into a set of one or multiple waypoints in an unknown environment, Goal Manager decides when to start and end the mission and which goal to pursue at a particular time.

3.2. PATH PLANNER

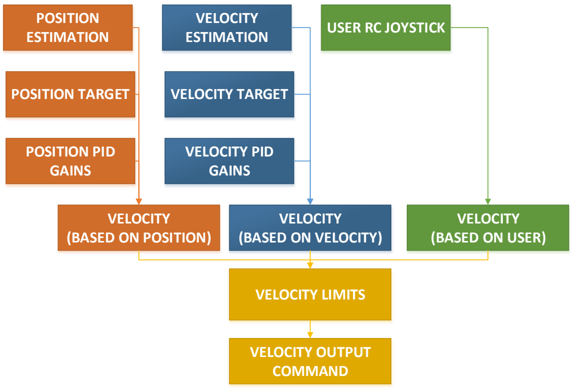

Figure 9: Position and Velocity PID control loop

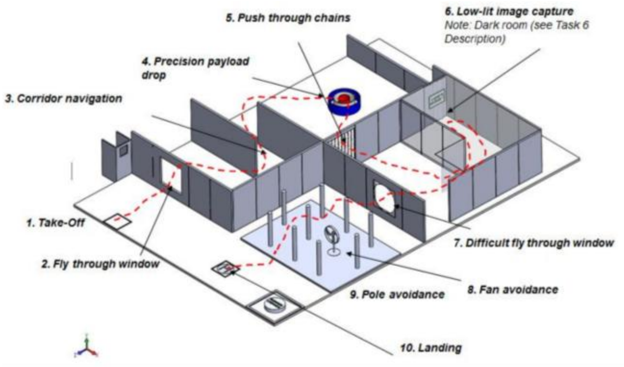

Figure 10: SAFMC 2016 – Category D2 challenge

Path Planner receives goals from Goal Manager and plan for the least costly path based on the instantaneously know map. The path is computed by a selected algorithm algorithm and is comprised of multiple points lining up to the goal. Besides the path is also updated in real time to help the copter optimize path, compensate for drifting as well as avoiding newly detected obstacles.

In this project, we use open-source Navigation package provided by ROS. The reason is that this package provide a complete and modular solutions for path planning, which includes many user interface and utilities modules that would be very time consuming yet barely productive to reinvent the wheel. Moreover, because the package itself is modular, it is also possible for us to quickly modify the source code and implement different path planning algorithms.

In default source-code, ROS Navigation package provides two basic options for path planning algorithms: Dijkstra's algorithm and A* search algorithm. After comparison flight tests, A* search algorithm is found to be more suitable for our UAV flight, because of its informed search mechanism yielding faster response time. Moreover, even though Dijkstra' algorithm produce more consistent path overtime, A* search is still more preferable because of drifting during flight and non-static obstacles that requires fast response.

3.3. 2D SLAM AND ALTITUDE

In this project, the laser scan data is not only used for 2D planar position estimator but also used for altitude measurement. In fact, each set of laser range data covers 270 degrees centred about the x-axis on the xy-plane. From this set of data, the central 180 degrees is used for 2D localization and mapping, while 25 degrees from each side is deflected by a mirror to the ground for altitude measurement.

Firstly, for 2D localization and mapping, even though there are a number of available algorithms and implementations [4, 5, 7], Hector SLAM is selected for several reasons. Firstly, many implementation of 2D SLAM assumes planar motion model and only takes into account three degrees of freedom, which introduces errors if used for this project's 6DOF application. Secondly, many of these approaches also assume the robot equipped with high performance computer, hence perform more complicated and computationally expensive such as pose graph optimization [8], which case is too expensive and barely necessary for this project micro UAV navigation. On the other hand, Hector SLAM uses optimized scan matching [9] to obtain 2D state estimation, which approach is practical for limited computation capability of UAV on-board computer and also sufficiently accurate for UAV low speed navigation. Thirdly, the Hector SLAM approach has been properly packaged into a ROS module and well tested by the open-source community.

Secondly, for local altitude measurement, a segment from the full set of range data is extracted and reflected downward to calculate the distance from copter to the floor. This approach is also verified in [4] and [5] to works well in structured and rectilinear environment, which is practical for most indoor flight applications. However, while the laser beam is radial direction, the mirror is flat and oriented at fixed angle, hence theoretically there will be error incurred by non-perpendicular floor projection. In order to overcome this limitation, only a small segment of 25 degrees is used for altitude measurement to ensure that errors from non-perpendicular reflection is negligible compared to actual reading. Moreover, another source of error is the mirror vibration during flight, causing the laser scanner to captured non-reflected range measurements. Because these measurements are from surrounding walls and obstacles, the reading is much greater than the reflected reading from floor. In order to improve the altitude measurement, a high pass filter is deployed to avoid these "bumps".

3.4. LOW-LEVEL CONTROL

The low-level control block contains mainly of position velocity and attitude PID control loops. Firstly, the high rate attitude PID control is already implemented within the Pixhawk firmware with a good record of test flights by the open-source community. Therefore we avoid reinventing the wheel by directly inheriting the existing work and only tune the PID gains till the UAV is sufficiently stable for our purpose.

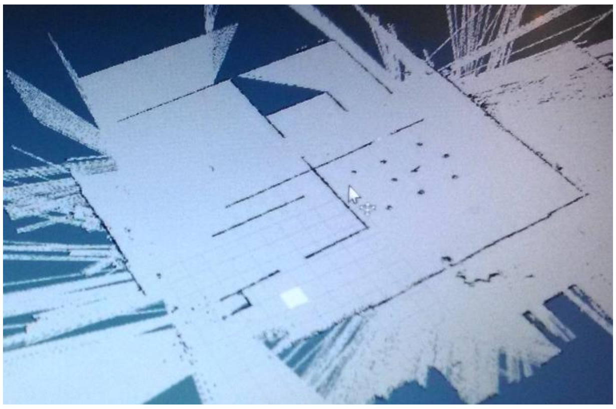

Figure 11: Map of SAFMC maze layout, processed in real-time by DR1 Alvis

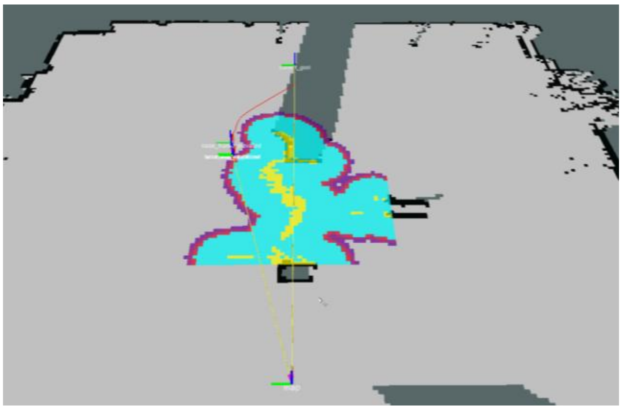

Figure 122: A snapshot of the online SLAM and path planning process. Red trail shows the planned path

On the other hand, even though Pixhawk firmware also has already implemented PID control loops for position and velocity, it has a limitation of ignoring user RC input. In fact, the position and velocity control loops are enabled upon activation of a special flight mode named "Offboard", which would subsequently attempt the navigate the copter by input velocity targets or position targets and completely ignore user joystick inputs until deactivated. Moreover, Offboard mode also lacks a limits on velocity of the copter. These limitations render the use of Pixhawk "Offboard" mode dangerous for both UAV operator and the UAV itself. Hence, the position and velocity PID control loops are redesigned and implemented, taking into account user inputs and velocity limits, in order to improve safety during flights.

Figure 7 illustrates the concept of position and velocity PID controls. Even though not drawn explicitly, this control segment should be understood as part of a complete closed-loop control mechanism, including attitude control and feedback sensors data.

4. FLIGHT TESTS AND RESULTS

The system is tested in several different cases, with increasing difficulty, to gradually tune the position and velocity control and also to account for any shortcomings from the mechanical design of the frame. The tests scenarios include 1) flying in square shape waypoints in empty corridor, 2) flying in a straight line, with one obstacle in the middle, 3) flying through door 1.5m-wide door and 4) fly through a maze.

Firstly, the empty space waypoint flight was the most basic flight, which was used to fine tune the PID gains for XY position control and altitude hold control. Moreover, it is also used to tune and limit the flight speed between waypoints so that the overshooting is within 1m and the flight is sufficiently safe for indoor autonomous flight.

Secondly, the 2 waypoints flight with one obstacle in the middle is used to test and fine tune path planning. In fact, to account for the lateral drift of the copter, the walls and obstacles are "inflated" to further restraint path planning and make the resulted plan further away from walls and obstacles. A glimpse into the inflation is shown in Figure 12, where the path planning algorithm is forced to come up with a trail around the inflated balloon shown in red, keeping the MAV at a safe distance from the obstacle shown in yellow color. However, if the inflation is too big, the algorithm will prevent the copter from entering doors or flying through narrow corridors, which results in mission failure. Moreover, this test is also used to determine the appropriate rate of planning so as to reduce workload for onboard computer. In fact, due to limited computation resource of onboard lightweight hardware, unnecessary path planning could delay the SLAM process, eventually affect the mapping rate, localization rate, and ultimately reduce position control performance.

Thirdly, the flight test navigating through door is another test used to verify the performance and robustness of path planning algorithm.

Last but not least, the copter is brought for a challenging in Singapore Amazing Flying Machine Competition (SAFMC), which is an annual MAV contest between various universities and institutions across Singapore. The challenge for the copter is to autonomously fly through the maze illustrated in figure 9, while performing various tasks such as ball dropping or photo capturing of predefined locations. Our copter managed to take off, enter door, navigate through corridors and escapes the exit door. However, due to the presence of adverse blowing fan, the copter exhibits self-rotating phenomenon while attempting to counter the wind. Moreover, the battery also drains, voltage drops quickly causing the gradual loss of thrust and the copter eventually land near the fan. The reason for the lack of agility against wind is suspected to be the limit of velocity control, while the shortcoming of battery is recognized to be a mechanical limitation. These issues are also addressed in subsequent works on this copter. Despite these limitation, the copter manage to almost finish the mission and produce the map of the maze as shown in Figure 11.

5. CONCLUSION

In this work, we have managed to design, construct and program a copter to fly operate autonomously based on simultaneous localization and mapping (SLAM). The hardware of the copter is designed to be modular in order to optimize independent modules of the copter for different requirements, without affecting others. The control of the copter is separated into two main layers: low-level and high-level control, in order to optimize the performance and modularity of the software. In the end of the project, the copter is able to hover, plan path, avoid obstacle and control position autonomously without manual control input. The copter is also tested in the maze of Singapore Amazing Flying Machine (SAFMC) competition, proving the concept is a potential solution for indoor autonomous MAV navigation.

6. ACKNOWLEDGMENT

We wish to acknowledge the funding support for this project from Nanyang Technological University under the Undergraduate Research Experience on CAmpus (URECA) programme.

7. REFERENCE

[1] K. Guo, Z. Qiu, C. Miao, A. H. Zaini, C.-L. Chen, W. Meng, *et al.*, "Ultra-Wideband-Based Localization for Quadcopter Navigation," *Unmanned Systems,* vol. 4, pp. 23-34, 2016.

[2] A. Benini, A. Mancini, and S. Longhi, "An imu/uwb/vision-based extended kalman filter for mini-uav localization in indoor environment using 802.15. 4a wireless sensor network," *Journal of Intelligent & Robotic Systems,* vol. 70, pp. 461-476, 2013.

[3] S. Thrun, "Probabilistic robotics," *Communications of the ACM,* vol. 45, pp. 52-57, 2002.

[4] I. Dryanovski, W. Morris, and J. Xiao, "An open-source pose estimation system for micro-air vehicles," in *Robotics and automation (ICRA), 2011 IEEE international conference on*, 2011, pp. 4449-4454.

[5] S. Grzonka, G. Grisetti, and W. Burgard, "Towards a navigation system for autonomous indoor flying," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, 2009, pp. 2878-2883.

[6] J. D. Anderson Jr, *Fundamentals of aerodynamics*: Tata McGraw-Hill Education, 2010.

[7] A. Bachrach, R. He, and N. Roy, "Autonomous flight in unknown indoor environments," *International Journal of Micro Air Vehicles,* vol. 1, pp. 217-228, 2009.

[8] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *Robotics, IEEE Transactions on,* vol. 24, pp. 1365-1378, 2008.

[9] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," in *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on,* 2011, pp. 155-160.