Imperial College London
Department of Aeronautics

MEng Group Design Project 2019

Autonomous Drone Delivery

Executive Summary

MEDICAIR

ADD01: Project Management

Marij QURESHI, Jean LAUNAY

ADD02: Flight Control

Robert ATANASIU, Pablo FERRER BERTOMEU, Joel LEOW, Lucian NITA, Jakob STEWART, Tian WANG

ADD03: Hardware development

Yiftach CATS, Caden HO, Alexey PANSKOV, Jinay PATEL

ADD04: On-board sensing

Jia Wei KHOR, Gerald LOW, Brian NG, Vishnu MENON, Łukasz ŚLIWIŃSKI, Rafal ZUKOWSKI

ADD05: Experimental setup, safety and prototyping

Harry ALLEN, Haoming MA, Jordan MILLAR, Benjamin WONG

Project Supervision

Dr Mirko KOVAC, Dr Ajit PANESAR, Dr Thulasi MYLVAGANAM, Dr Nigel MACCARTHY, Dr Vito TAGARIELLI, Prof. Ferri ALIABADI

Project Advisory

Mr Roland HUTCHINS, Mr Ian PARDEW

***Abstract* - This project comprised 22 students designing and testing an autonomous drone as a proof of concept for delivery of small medical equipment. Outdoor drone navigation was done through a GPS/IMU sensor fusion via an extended Kalman filter, and both D*Lite and potential fields were used for obstacle avoidance. Precision landing was solved with a RGB camera tracking QR codes and an autonomous moving platform tracking the drone using a Terabee 3Dcam. An origami-based gripper actuated using a pump was used to pick up, store and release payloads up to 2kg. The concept was simulated in Gazebo and tested both indoors and outdoors at Imperial College Silwood Campus subject to regulation and safety measures. All but one system were successfully tested and laid the foundations for future drone delivery projects.**

1 Introduction

The use of drones for delivery is a topic widely discussed and known for its great potential to meet demand fast, at low cost and in remote locations. The scope of this project was set to design a drone delivery prototype capable of picking up and delivering a delicate item such as an egg. There was not a prescribed mission requirement so it was the role of the team to find relevant applications of this technology and define the challenges it would set.

1.1 Mission definition

The growing ageing population has led to a rapid increase of emergency care demand in developed countries. Health systems, under strong pressure to meet response targets without compromising clinical service, require larger and larger investments. However, in the UK - like other countries - funding for emergency care has only risen by 17% over the last last 4 year while demand has increased by 30% [3].

The implementation of an autonomous network of drones offers a fast and efficient solution to tackle the rising demand. Therefore, the mission of this project was set to design a drone capable of providing medical supplies to emergency locations saving the time spent in triage - time spent allocating the appropriate resources for an emergency case. The complexity was further increased by adding the feature of moving target tracking to deliver medical supplies directly to the ambulance after an on-site assessment by the deployed emergency care technicians.

It is worth mentioning that while this report is focused in the employment of this technology in the health care sector it could be used to a broad range of different applications.

1.2 Technical challenges

The technical challenges that this mission presents are shown in Figure 1. To design this prototype predefined groups (Flight control, Sensors, Hardware and Experimental Set-up) were restructured in functional teams in charge of tackling a particular part of the mission. This report presents the contribution of the 9 functional teams towards the design and implementation of the drone delivery prototype.

2 GPS Navigation

2.1 Mission requirement

Part of the objective of the mission was to remotely track and converge to a moving target. To accomplish this, the evaluation and characterisation of GPS emitting devices, both as a receiver on the drone and on the moving target, as well as the tracking algorithm to reach the target in an efficient manner, were considered.

2.2 Sensor Selection

2.2.1 RTK GPS

RTK GPS (Real-time-Kinematics Global-Positioning-System) is a robust device that offers superb positioning accuracy. The system employs one or multiple base stations and a rover mounted on a desired vehicle. The base stations have accurate data about their own global position. Using this information and measuring the signal from the satellites they can calculate the momentary error of the GPS positioning. The base stations send appropriate corrections to the rover, which enables very accurate tracking of the vehicle [23].

2.2.2 GPS on moving target

An Android Smartphone was chosen for the following arguments:

1. Widespread adoption (1.4bn Android users [4]) across the world bringing no additional cost
2. Availability of complementary sensors for sensor fusion
3. Seamless integration with ROS

2.3 RTK GPS

At the start of the project, the team was supplied with Here+ RTK GPS which seamless worked with PX4 software and could be used for state estimation. Various accuracy tests were performed to quantitatively assess the accuracy of the used GPS.
Unfortunately, the provided base module stopped working. For testing, only the rover module was used which nonetheless partly sufficed for the purposes of the project.

2.4 GPS on moving target

2.4.1 Remotely sending phone location data

A novel approach to an easy to setup, yet secure transmission of ROS (GPS) data over the cloud was developed. This makes use of the following technologies: SSH, Websockets, and Port forwarding. An SSH webserver *serveo.net* was utilised to aid development time. Finally, in order to send packets of data over the cloud, ROS messages were parsed into JSON and communicated over Websocket servers implemented through ROSBridge. This establishes a full-duplex communication between the server and client.

2.5 Proportional Navigation

For the tracking phase of the algorithm, a modern variation of the classical pure proportional navigation algorithm was chosen. The acceleration command normal to the line of sight (LOS) between the drone and the target is generated according to the relation [2]:

$$a_\perp = -\lambda|\dot{r}|\frac{r}{|r|} \times \frac{r \times \dot{r}}{r \cdot r} \tag{1}$$

where λ is a proportionality constant called the navigation constant, r is the LOS vector and $\dot{r}$ is its derivative in time. This was supplemented with an additional acceleration LOS direction using a PD controller in order to increase closing velocity. This is:

$$a_\parallel = K_p r + K_d \dot{r} \tag{2}$$

where K_p and K_d are the proportional and derivative gains respectively. The total acceleration input to the drone was the sum of these two components.

3 Obstacle Avoidance

3.1 Sensor Selection

Ultrasonic sensors have been dropped early in the design phase due to its likeliness in receiving noise, especially when operated outdoors [1]. The low scanning frequency of Radar also makes it impractical for a flying quadcopter as that would hugely restrict the cruise speed to ensure the path ahead is clear of obstacles [15]. Stereo-vision cameras are able to produce depth output by using two RGB cameras simultaneously [10]. It's light weight property makes it one of the ideal candidates for this project. Yet, extra computation is required to produce stereo-vision and object detection for identifying the location of obstacles, which would either be too computationally expensive with the Odroid-XU4 unit or would require extra weight and power supply for a separate computation unit. Laser rangefinders are therefore adopted for its low weight and low computational cost for depth information. The downside of it is the degraded range and accuracy in outdoor conditions. TeraOne sensors, which is capable of a range up to 11 m, can only produce a single depth reading over its 3° field of view (FOV) [21], which is too narrow to produce a continuous depth reading for obstacles closer then 4 m should 3 sensors be used. Terabee 3D CAM is therefore planned to form part of the sensor package to provide a complete depth reading over its wider FOV within a shorter range (up to 4 m) [20]. The 3D CAM is dropped later due to its incompatibility with Odriod-XU4.

3.2 Algorithm Selection

To determine the desired path for our drone and avoid obstacles on the way, we need to link the initial configuration of quadcopter's C-space to the goal configuration by a smooth series of collision free configurations (C-free).[18] The chosen algorithm needs to generate a trajectory that optimises the path distance and hence minimises the mission time under constant velocity assumption. Algorithm robustness must be an important factor in the selection process in order to accommodate sensors uncertainty and measurement noise.[11] Memory limitations affect the scalability of the algorithm on one hand, but good performance in complex situations is desired. Also, path planner's execution time can affect the reaction time of the physical system.

A number of different solutions both discrete time and continuous time were compared. Receding horizon control method [14] [13] as well as implementing a non-linear controller using echo state network [16] were considered to be too computationally expensive. Using mixed integer linear programming approach [6], or employing classical graph search algorithms [17] after performing a cell-decomposition of the C-space were also not selected as they are only suited for offline implementation and are not able to work in unknown dynamic environments. Chosen algorithm which was implemented was D* Lite, a search algorithm that gives the shortest path to goal using heuristics to focus the search and reduce the execution time. A simplified version of potential fields method was also used as a backup solution.

3.3 Algorithm Implementation

The core D* Lite implementation was based on Koenig's original paper[12], however, some small modifications such as adapting it to work with a moving goal, using a hash map to speed up the program were used. Safety features were also added to ensure it is robust enough to adapt state estimation errors, inexact sensor readings and be able to terminate safely in case it will reach the bounds of the computational power available. A backup obstacle avoidance program based on a modified potential field method [8] was also implemented.

3.4 Performance

Final field test was unsuccessful for the main D* lite algorithm, but some useful data could be acquired and main causes of failure could be diagnosed. Memory and processing power available on the Odroid represented the main bottleneck. However, safety features proved to be working and useful in the event of memory overloading and the drone successfully landed autonomously. Considering the successful Gazebo simulations, it is still believed that this algorithm is suited for use, but further tests are required. Continuous vision sensors can be used for obstacle detection, a better processor should be tested for computing the trajectory and the mission should be performed in an environment partially mapped. After these tests it could be could be quantitatively stated how these factors affected the mission performance. Fortunately, the backup algorithm was robust enough and even if it did not produce

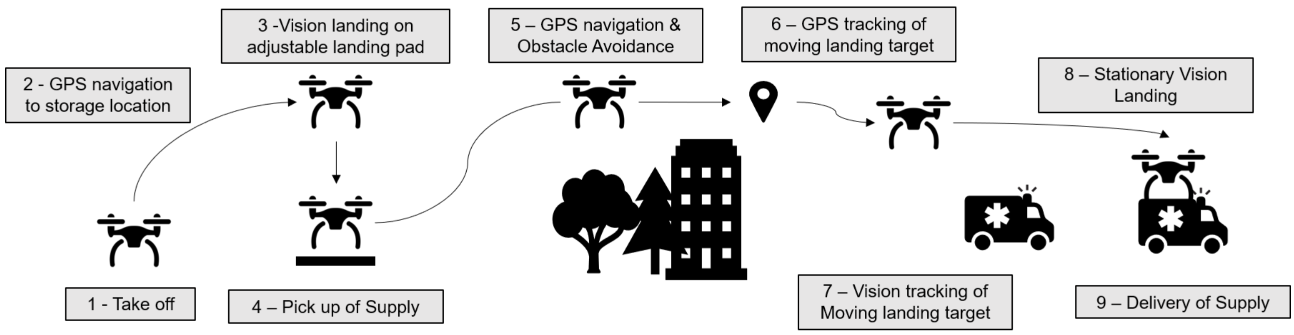

Figure 1: Mission profile diagram

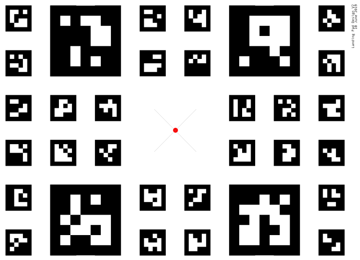

Figure 2: Vision sub-system layout

the minimum distance path, it successfully avoided the obstacles.

4 Vision Tracking

4.1 Algorithm Literature Review

In order to determine the relative pose (translations and rotations) of the camera on the drone and the landing target, an algorithm to detect the target in the image frame and another to estimate the relative pose in the drone's body fixed coordinate from the coordinates of the landing target on the image frame. Furthermore, the chosen algorithms must be capable of running at more than 5fps on the ODROID-XU4, as required by the landing trajectory control algorithm.

4.1.1 Object Detection

As computer vision, and particularly object detection, is a widely researched subject, a number of candidate algorithms are available for achieving it.

Initially, state-of-the-art, deep-learning based approaches such as Faster R-CNN, SSD and YOLOv3-Darknet were considered. A customized, slim build of YOLOv3 network was trained and tested on the ODROID-XU4 and was only able to attain an average frame rate of 1 fps. Furthermore, the algorithm was using all of ODROID's compute and memory resources, leaving little for other running processes.

Fiducial marker based tracking is a popular approach to determine relative pose in Augmented Reality (AR) applications. This involves placing an easily-detectable, predefined marker (ie. AR tags) in the environment and leveraging classical computer vision techniques to detect it. This is chosen as the backbone of the object detection methodology for this project as they are well-researched, efficient and supported by a number of open-source toolkits.

4.1.2 Pose Estimation

The pose of a calibrated camera can be uniquely determined from a minimum of four co-planar but non-collinear points. Therefore, using the four corners of the marker in the image coordinates with the intrinsic properties of the camera (camera matrix and Brown's distortion coefficients), the camera's pose can be estimated.

4.2 Algorithm Implementation

4.2.1 Landing Pad

A landing pad (figure 2 was designed with 28 small and 4 large AR tag around its centre, where the target will be placed. As each AR tag is unique, it is then possible to obtain the relative pose of the camera from the centre of the landing pad by detecting at least one marker. Furthermore, a mix of large and small markers were chosen to allow detection of markers at high and low altitudes.

5 Moving Landing Platform

5.1 Need for the Platform

Whether the mission comprising delivery of medications can be considered successful, will primarily depend on its duration. Therefore every leg of the mission must be completed in a shortest possible time. It was however recognised that performing autonomous cargo pick-up of fragile goods with a centimetre precision, might constitute a time bottleneck in the mission profile. It is caused by limited control authority of drones, particularly those in the early phases of design and the absence of accurate dynamic models. Furthermore, reduced precision of actuation as well as degraded attitude control and stability are phenomena inherently associated with ground effect, existing during near-surface operations [5]. To counteract them, a moving landing platform was designed upon which the cargo item will be positioned and actuated towards the centre of grappling mechanism of the approaching drone. Its functionality will render the pick-up procedure successful in the first attempt, thus saving a lot of precious time, which would otherwise be wasted for multiple pick-up attempts of the drone alone.

5.2 Platform Architecture

The platform consists of a custom-built rails enabling in-plane motion of the landing pad, actuated with use of two stepper motors. The pad, given available time for development, was constructed from 4 layers of cardboard of $[0/90]_S$ stacking sequence, thus providing good resistance to bending, necessary during drone touchdown. The edges of the pad are supported by a box made of Thermoset material. On top of the landing pad, a set of high-contrast markers is placed along with the cargo item.The platform is equipped with an upward-facing 3D TeraBee Camera, providing a depth-map of the approaching aerial vehicle. These depth-related data are processed by an independent CPU, which triggers actuation.

5.3 Sensors

The TeraBee 3D camera, an infrared ToF camera with a field-of-view of $74° \times 57°$ and a maximum range of 4m, is the only sensor used for the moving landing pad. The camera is placed on the moving landing pad and is pointed upwards facing the approaching drone. The camera generates both an infrared image and a point cloud of the drone. From the infrared image, the centre of the circular gripper is tracked using circular Hough Transform, a detection algorithm for circles. Then the x,y coordinates from the image is mapped back to the point cloud to obtain the actual x,y,z coordinates of the gripper.

5.4 Autonomous Tracking

The main concept behind the automation of moving landing platform is using MATLAB to simulate the data sent into landing pad while it is under manual control. The manual interaction is built via the Pronterface with the help of which the G code command can be sent directly into controller board (mpx-3) of landing platform. By monitoring the data transfer though the USB port, it can be found that mpx-3 only takes hexadecimal data as input. Thus, the data can be mocked in MATLAB and convert to hexadecimal format by using the function *dec2hex*, and the information is outputted though the port by MATLAB function *fprintf*. For convenience, we recorded all the necessary sample command and build a landing pad library to control the moving landing platform.

6 Delivery Mechanism

6.1 Initial Concept

The delivery mechanism consists of an origami gripper that actuates to lift the payload, and an encapsulation mechanism to prevent the payload from slipping out during flight due to sudden perturbations such as gusts. Thirteen concepts for the gripper were generated which were broadly classified into soft robotic [19] and mechanical systems. A trade-off analysis was carried out to aid in the selection process. The final choice was made for a pneumatically actuated origami gripper that is encased within an airtight membrane.

6.2 Design Choice

Figure 3 shows the final design of the capture mechanism. The blue arms are the rotating supports. Using a servo motor, it was possible to rotate the structure 180 degrees and lock into position thus protecting it from perturbations. The golden connector was 3D printed to provide extra holes in case of blockage. The idea behind this structure is that when sizing the origami up for full scale production, the rotating mechanism will automatically be sized up to accommodate this.

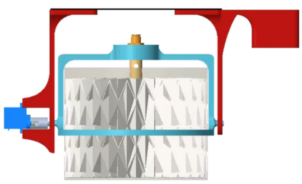

Figure 3: Design Choice Diagram

6.3 Origami Design

The origami gripper was based on a tessellation of the water-bomb pattern in a cylindrical arrangement [9], [22]. This structure exhibits auxetic behaviour meaning it has a negative Poisson ratio [7]. Thus, by actuating it in compression it was possible to create a gripping force.

6.4 Finite Elements Analysis and Validation

A finite element analysis was carried out on origami structure to optimise for the geometry and number of circumferential panels. The numerically obtained results were compared with a mathematical calculation of the classical solution of a simple thin-walled cylinder. The FEA was also further validated by experimental data which showed similar trends. Additionally, forces on an elliptical payload were also simulated and compared with an experimental failure pin-test of an hen egg. The pin failure load of an egg was 7 times as much as the simulated load, which provided sufficient confidence that the gripper would not damage the payload. This shows the benefit of soft compliant robotics where the mechanism can produce a significant gripping force while also being gentle as to not damage the payload.

6.5 Experimental Results

The experimental results proved the outstanding lifting capabilities of the delivery mechanism. Where the gripper was capable of inducing a lifting force of up to 25.85N (2.63Kg). This is 50 times greater then the required force to lift the prescribed payload.

6.6 Integration

A full integrated system was achieved meaning that the actuation of the vacuum pump as well as the servo motor was preform autonomously. By using a force sensor which was attached to the membrane which indicated when contact was made with the payload. This also indicated when sufficient gripping force was applied meaning that the payload could be lifted. The actuation was found to take approximately 13.5 seconds for the lifting force to be sufficient to pick up an egg. Thus, take-off will commence only 20 seconds after the actuation of the vacuum pump.

6.7 Conclusion

A variety of tests have been conducted on the mechanism with the intent of quantifying its performance limits and to show that the mechanism performs as expected. It has been shown that the capture mechanism can encapsulate a fragile payload without damaging it. This functioning prototype has the potential to be readily scaled up for industrial use.

7 Integration & Simulation

7.1 MATLAB Simulation

To verify the control algorithms being designed, simulations were performed on several platforms throughout the project. MATLAB and Simulink were used as an early-stage simulator which aimed at checking the accuracy and robustness of the trajectory generated by the target tracking and obstacle avoidance algorithms. The mathematical model of the quadcopter is constructed in MATLAB using Newton-Euler equations of motion, and it is linearised by assuming the drone is closing to the hovering state. At the same time, a dual-loop PID controller is used to represent the built-in PX4 controller in Pixhawk. Thus, MAT-

LAB is able to visualise the trajectory generated and test if the drone is able to stably achieve the desired state.

Simulink provides a chance to interact the algorithm with the virtual world. By building a 3D virtual world with pick-up point, obstacle and moving ambulance, we could test if the algorithm could generate the ideal trajectory which allows the drone to finish entire mission.

7.2 Gazebo Simulation

In the Gazebo simulator, we were able to emulate the sensors and sensor data such that we could test our control algorithms at a more realistic level. This also allowed us to verify, troubleshoot, and refine our control algorithms at the finite update rate as it is in reality, which was an important step from the MATLAB simulations which had real-time data input. Furthermore, we were able to build an expectation for the behaviour of the drone under autonomous control, such that any unexpected behaviour by the drone could be safely and promptly dealt with. Using Gazebo, we were successful in verifying all individual mission phases and algorithms: Vision Landing for egg pickup and delivery, D* Lite and Modified Potential Fields for obstacle avoidance, and Proportional Navigation for target tracking.

7.3 Integration of Algorithms

Until this point, all algorithms have been tested isolated in both environments: virtual world created in *Gazebo* and real world. But, the sole purpose of this project was to design and build a functioning drone that can perform the required mission profile. As a result, all the individual scripts had to be put together into a universal common form. The condition behind this was to ensure that the scripts will not interfere with each other, for example *vision tracking* will not start while *Potential Fields* was running.

Figure 4 presents the order in which the algorithms will run during the mission profile. Even though a computational symbiosis that would allow parallel running of obstacle avoidance and PN tracking would be ideal, the specification of the Offboard computer did not allow the implementation of such a delicate design. With further improvements of hardware and computational iterations, the algorithms would be allowed to intertwine with each other, thus ensuring a more fluent deployment of the mission, from both a computational and energy-saving standpoint.

7.4 Hardware Integration

There are two main components to the UAV system: the on-board computer and the flight controller. The on-board computer is tasked with controlling the drone autonomously and performing the calculations for computer vision and obstacle avoidance. On the other hand, the flight controller is a low-level linear controller which carries out state estimation through an EKF and controls the drone through a PID feedback loop. The two systems are integrated using MAVROS, an open source library which allows sending MAVLink commands from ROS. The on-board computer can subscribe to the flight controller's data and use that information to autonomously perform the mission.

7.5 Sensor Integration

The Companion Computer chosen was an Odroid XU4 board with 16GB eMMC flash storage, 2GB of RAM, running a Samsung Exynos5422 8-core processor. The Linux distro flashed is Ubuntu Mate 18.04, along with a stable version of ROS Melodic. Figure 5 illustrates the file system arrangement in Odroid and the node topology of the full integrated system.

8 Experimental Considerations

8.1 Prototyping

3D printing was the method of choice for rapid prototyping, primarily making use of the Ultimaker 3 Extended printer to manufacture rigid on-board components using polylactic acid (PLA), and the compliant, origami gripper mechanism using thermoplastic polyurethane (TPU). Initial rigid prints used the less dense acrylonitrile butadiene styrene (ABS) but severe warping problems caused large deviations from desired geometry, rendering prints unusable, and eventually PLA was chosen instead.

Over the course of the project, design considerations for 3D-printing were increasingly incorporated into the desired components, as the design process evolved to include not just geometrical or functional requirements and/or limitations, but also aspects of 3D printing, such as print direction and the reduction of print times by avoiding the printing of excessive support material.

8.2 Safety and Flight Tests

The team's safety objective was to complete all flight tests without harm to any persons or property, by implementing the necessary policies, physical hardware, and safety measures to reduce the probabilities of any incident occurring to as close to zero as possible.

Flight tests were conducted indoors in the Student Workshop (CAGB 223), and subsequently outdoors at Pond Field, which is part of the college's Silwood Park campus. This was made possible by conducting the necessary risk assessments for all drone-related work at both locations, and the formulation of standard operating procedures (SOPs) that ensured that all control measures that had been introduced were also duly and responsibly implemented. A comprehensive but non-exhaustive set of emergency response procedures (ERPs) were also devised to detail the necessary actions required to be taken in the event of any incident occurring. The green light from the department was eventually given as the above documentation and considerations fell into place, and a total of 70 flights were conducted successfully with no major incidents to report.

In addition, the dedicated safety team also compiled an Operations Manual, the first of its kind in Imperial, detailing all efforts made to enable outdoor flight testing possible under the scope of this project.

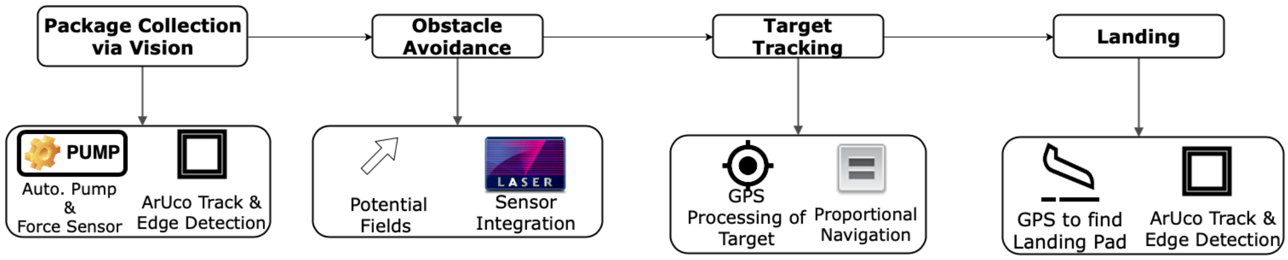

Figure 4: Algorithm Integration Topology

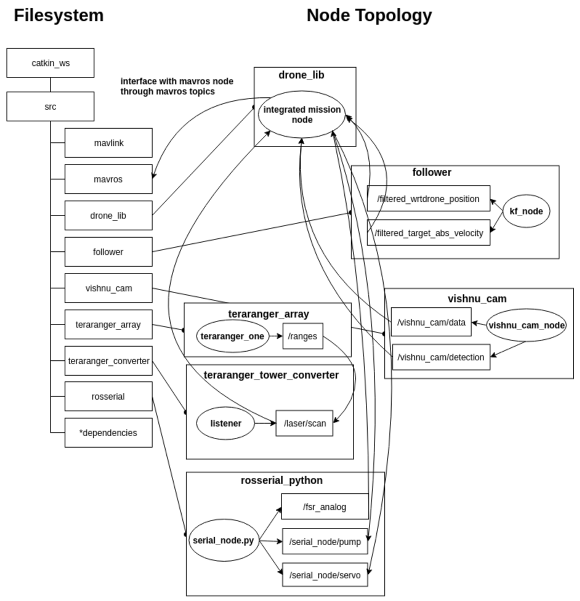

Figure 5: Filesystem and ROS computation graph showing node topology on overall system

This manual serves as a semi-official document that provides a safe and robust framework which ensured the responsible use of unmanned aerial systems, and can be used as the groundwork for future drone work in the college, and form the basis of future department- or college-wide drone policies.

9 Conclusion

Despite having a very ambitious goal and mission profile, this project was a success, with the team managing to demonstrate proofs of concept for most and working simulations for all subsystems pursued. The final drone was able to navigate to GPS waypoints through a variety of obstacles, track a mobile GPS accurately and land. A capture and delivery system was successfully tested along with an autonomous moving landing platform, leaving only a precision vision-landing system to be real-life tested in future iterations.

A significant challenge of this project, differentiating it from many others, was testing outdoors under exposure to the elements, however this better demonstrated the difficulties that such a concept might encounter when deployed on a large scale. We hope that with further work on the vision-landing system and more integrated outdoor tests, this project could evolve into a practical solution for transporting urgent medical equipment, saving many lives and inspiring the next generation of autonomous drone delivery technologies.

References

[1] J. Borestein. *Obstacle Avoidance with Ultrasonic Sensors*. Tech. rep. IEEE Journal of Robotics and Automation, 1988.

[2] A. Borowczyk. “Autonomous Landing of a Multirotor Micro Air Vehicle on a High Velocity Ground Vehicle”. In: *International Federation of Automatic Control* (2017).

[3] Comptroller and Auditor Genera. “NHS Ambulance Services”. In: (23 January 2017), p. 71.

[4] Gartner. N. D. “Global mobile OS market share in sales to end users from 1st quarter 2009 to 2nd quarter 2018”. In: ().

[5] Fabio Riccardi Davide Del Cont Bernard Mattia Giurato and Marco Lovera. *Ground effect analysis for a Quadrotor Platform*. Tech. rep. April 2017.

[6] James A Devens. “Autonomous Navigation with Obstacle Avoidance for Unmanned Aircraft Systems using MILP”. In: (2016).

[7] Shuzhi Sam Ge and Yun J Cui. “DMiura ori based curved surface origami structures”. In: (November 11, 2014).

[8] Shuzhi Sam Ge and Yun J Cui. “Dynamic motion planning for mobile robots using potential field method”. In: *Autonomous robots* 13.3 (2002), pp. 207–222.

[9] Brandon H. Hanna. “Modeling and Testing of Bistable Waterbomb Base Configurations”. In: (September, 2014.).

[10] S. Hrabar. *3D Path Planning and Stereo-based Obstacle Avoidance for Rotorcraft UAVs*. Tech. rep. International Conference on Intelligent Robots and Systems, 2008.

[11] Minyang Kang et al. “An empirical study on robustness of UAV path planning algorithms considering position uncertainty”. In: *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*. IEEE. 2017, pp. 1–6.

[12] Sven Koenig and Maxim Likhachev. “D^* Lite”. In: *Aaai/iaai* 15 (2002).

[13] Brandon Brandon Douglas Luders. "Robust trajectory planning for unmanned aerial vehicles in uncertain environments". PhD thesis. Massachusetts Institute of Technology, 2008.

[14] Jacob Mattingley, Yang Wang, and Stephen Boyd. "Code generation for receding horizon control". In: *2010 IEEE International Symposium on Computer-Aided Control System Design*. IEEE. 2010, pp. 985–992.

[15] J. Patel. *Review of Radar Classification RCS Characterisation Techniques for Small UAVs or Drones*. Tech. rep. IET Radar, Sonar and Navigation, 2018.

[16] Bogdan Pugach et al. "Nonlinear controller for a UAV using Echo State Network". In: *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE. 2017, pp. 124–132.

[17] B Moses Sathyaraj et al. "Multiple UAVs path planning algorithms: a comparative study". In: *Fuzzy Optimization and Decision Making* 7.3 (2008), p. 257.

[18] Yasmina Bestaoui Sebbane. *Smart autonomous aircraft: flight control and planning for UAV*. Crc Press, 2015.

[19] Matus Stefek. "Soft robotics. Technical repor". In: (February 5, 2016).

[20] *Terabee 3D CAM 80x60 Specification sheet*. Tech. rep. Terabee, 2019.

[21] *TeraRanger One Specification sheet*. Tech. rep. Defense and Security Symposium, 2019.

[22] B. Goodwine N. Turner and S. Mihir. "A review of origami and its applications in mechanical engineering". In: (2015).

[23] Lambert Wanninger. *Introduction to Network RTK*. `http://www.wasoft.de/e/iagwg451/intro/introduction.html`. Accessed: 2019-06-12. 16 June 2008.