CS407 Specification

Ben De Ivey, Will Seymour, Jon Gibson, Alex Henson

December 10, 2015

1 Introduction

Airborne drones, as referred to in this project, are unmanned aerial vehicles (UAV) capable of sensing their surroundings and performing some processing or computation in an automated fashion. They are used for a variety of purposes, including filming[3], exploration[4], and even selfie taking[1]. Along with the numerous uses of drones in the military setting[7], the applications for a network of monitoring drones are broad. For this project, the type of airborne drones that will be used and talked about are smaller and more compact than those, for example, that are military UAVs.

Over the last couple of decades, research and development has been done looking into replacing humans with drones in potentially dangerous situations[2][6]. Unstable mines, flooded tunnels, or areas with high levels of radiation of harmful gas are examples of places where it is unsafe for a human to go, but fine for a drone. With suitable sensors and algorithms, it is possible for drones to perform the same tasks as humans in these situations but in a much safer way.

While the main focus of this project is create a framework for a drone network to undertake a specified task, an application of this project is firefighting in situations too dangerous, expensive or impractical or human to perform. For example, California frequently has issues with forest fires. It has been shown that 80%-96% of area burned by forest fires in California were caused by 1% of the largest fires[5]. Unmanned flying drones in a network, capable of detecting initial outbreaks of a fire, could potentially suppress the fire before the situation were to get worse, either entirely dealing with the fire, or giving the fire response teams longer to reach the site and less danger when they arrive.

In this project, we propose to tackle the problem with a drone network, where if any one drone detects a fire, then that information is immediately broadcast to others, and appropriate action taken.

2 Overview and Intended Functionality

The intended outcome of the project is to accurately model airborne sensor networks (drone networks) operating over a well defined area of physical space. While the fire fighting case study presented will target a specific problem, the software produced as part of the project will operate on more general problems. At a high level, our simulated networks will be able to monitor a range of different properties using any continuous linear sensors, returning data to the fixed base station from which they were tasked. Properties might be tasked as compiling a map of a property over an area or finding only the maximum value of a property within an area (such as temperature). It will also be possible to specify actions to perform when certain conditions are met (such as when a fire has got out of control).

The network model will solve the problem of work division, with individual nodes electing a leader and having that leader distribute the tasked workload between the swarm. The other main problems associated with modelling mobile sensor networks are the routing of individual nodes and the efficient transmission of communications data over long distances. In terms of directing individual nodes, drones will be assigned airspace and maintain a model of which units are operational in each area, negotiating passing space when required. When a node needs to communicate data back to the base station it will choose an appropriate method of doing so. This may involve connecting directly to the base station or by passing the message to a nearby drone which is closer to the base station.

The network simulation should also have a concept of fault tolerance. If an individual node fails then the system should be able to detect this and reallocate work without compromising the integrity of the data collected. A node may fail in a number of ways: becoming unresponsive, providing incorrect responses, providing responses outside of the specified time intervals, and in other ways as arbitrary (or byzantine) failure.

If time and resources permit, we intend to deploy the developed software to physical drones in order to better demonstrate the network. This will show how the network reacts to real world conditions which is especially relevant in terms of fault tolerance.

3 Interfaces

In order to simulate a drone network and formulate the interaction between drones and the base station, it will be necessary to use various different interfaces to achieve the intended functionality. The main program which will be used is NS-3, free software which provides discrete event network simulation, which can construct a theoretical simulation of the drone network and the tasks that it will perform. NS-3 will be useful for a number of reasons. Firstly, using NS-3 will allow for the testing of how the physical drones will/should perform in practice, simultaneously giving a framework of the code with which they will be programmed. NS-3 uses Python or C++, providing full support for each language, and therefore the creation of a theoretical simulation in NS-3 would also equate to a skeleton program for the coding of the physical drones, in the event that these languages are used for drone functionality. Additionally, if it is the case that the intended outcome of the project is not fully realised on a physical level, it will still be possible to create a scalable model solution for airborne sensor networks for a given problem set.

However, NS-3 is merely a library which provides a set of network simulation models that users can interact with to set up a simulation and provide an output. In other words, because these events are simply function calls, it is also necessary to use a graphical front end for NS-3, so that the simulation can be readily visualised. For this task, NetAnim will be used. NetAnim is an offline animator which animates a simulation using an XML trace file collected during simulation, creating a GUI that will give information and statistics about the simulation. The advantage of using NetAnim is that it will not only provide a visual display for the simulation, but it will be possible to step through a simulation one event at a time and pause the simulation at a given time, as well as printing routing tables and nodes at various points in time. These features will allow for much more flexibility than standard output alone.

The drones which will be used in the project are Parrot AR 2.0 Power Edition quadcopter drones. Given that these drones are not in and of themselves programmable and autonomous, a Raspberry Pi or Arduino (small, single-board computers) will be mounted onto the drone itself and programmed in order to give instructions to the drone for basic autonomous functionality, including any sensor manipulation which may be required for the project. Finally, in order to perform the task of interaction between the drone and its base station, a custom-made language will be required such that problem detection and correction can be performed autonomously. This language will be responsible for processing any information passed by the drain to the base station, as well as handling the subsequent commands which will be passed back to the drone.

4 Scheduling

The project can be broken down into N major components: Network communications, directing drones in physical space, specifying and executing input, deployment to real drones. These sections are estimated to be of roughly equal size. Networking and physical routing will be developed in parallel, with the intention of having them complete by the start of term two. During term two, the tasking logic will be constructed, along with the method by which the user inputs instructions for the network to complete.

5 Management and Development Strategies

Throughout the course of the project, the team will partake in weekly meetings with the project supervisor. Progress tracked throughout the project against the scheduling plan in section 4 will ensure that the project's time is managed appropriately.

At the midway point through the project, a progress report will be written with details for the interested parties of how the project is progressing, any revised development plans, and a revised and detailed specification.

Due to the small size of the project team and the fluctuating nature of time each team member has available, a development methodology inspired by Agile methods will be used. Two week sprints ending in a meeting with the project supervisor, with another meeting half way through the sprint, enable the progress of the project in a flexible but structured way. Sprint goals can be decided upon at the start of the fortnightly period based on the current state of the project, as well as the estimated available time of the project members during the period.

To manage the project's material, a git repository will be created, ensuring consistency amongst the project members' work, and safety for the code base. Not only does this reduce duplication of work, but it also prevents data loss in the event of the unexpected. Additionally it allows code to be reverted if updates break previous functionality.

References

[1] Lily Robotics Inc. Lily - camera. reinvented, 2015.

[2] Vijay Kumar, Daniela Rus, and Sanjiv Singh. Robot and sensor networks for first responders. *Pervasive Computing, IEEE*, 3(4):24–33, 2004.

[3] Skyvantage LTD. Stunning aerial filming photography flying uav drones, london uk, 2015.

[4] Gem Systems Advanced Magnetometers. Uavs - pathway to the future - gem systems, 2015.

[5] Stephen J Pyne et al. *Fire in America. A cultural history of wildland and rural fire.* Princeton University Press, 1982.

[6] Niramon Ruangpayoongsak, Hubert Roth, and Jan Chudoba. Mobile robots for search and rescue. In *Safety, Security and Rescue Robotics, Workshop, 2005 IEEE International*, pages 212–217. IEEE, 2005.

[7] Noel Sharkey. The automation and proliferation of military drones and the protection of civilians. *Law, Innovation and Technology*, 3(2):229–240, 2011.