Tracking of a Fleet of Paparazzi Drones

Isabelle Santos
A20325602
isantos1@hawk.iit.edu

Mengliang Li
Axxxxxxxx
mli80@hawk.iit.edu

Dheeraj Prajwal Bojja Venkata
A20329437
prajdheeraj@gmail.com

***Abstract*—The safety pilot of a drone can almost only rely on their eyes to determine the state of a drone they are monitoring. In the context for instance of a search and rescue mission, many drones could be involved, and the pilot may lose sight of the drone they are monitoring, and some drones may go beyond the range of the safety pilot's remote control. This project proposal envisions and describes the design and implementation of a smartphone app to monitor and control a fleet of drones with an interface similar to that of GoogleSky.**

I. Introduction

A. Context

Paparazzi is a free and open-source software project [1] comprising the hardware and software necessary deploy a drone autopilot system. Paparazzi has been used by universities, private companies and drone enthusiasts.

Paparazzi is articulated around a ground control system (GCS) and the actual drone. The drone is composed of modules for receiving orders from the ground station, modules for sending status reports to the ground station, and flight control systems and actuators. The GCS is generally a computer running the Paparazzi software, with an antenna to emit and receive messages to and from the drone and a standard RC transmitter.

During the flight of a drone, there are two pilots:

- the GCS operator inputs flight parameters and sees the flight data in real-time. He follows the trajectory of the drone on a map;
- the safety pilot observes the drone and holds ready to take command of in case anything unexpected happens. He follows the trajectory of the drone in the sky.

1) Limitations of the current system:

a) Quality of information transmission: The safety pilot communicates with the GCS operator (for instance by talkie-walkie) in order to get the drone flight data. The transmission can be of poor quality, resulting in incomprehensions. If the communication between both operators is cut off, the flight becomes dangerous.

b) Information reception time: There is a high latency between the moment when the drone sends out information and the moment when the safety pilot receives the information.

c) Drone locating: The safety pilot has no way of easily finding the position of a drone. If the safety pilot loses sight of a drone, they must go to the GCS, look at the map, and try to spot the drone.

d) Drone identification: The safety pilot can not identify a drone that they see without prior knowledge. Two drones may look very similar, and distinguishing them by sight from a distance may be impossible.

e) Fleet management: The safety pilot usually has only a standard RC model remote control, so typically a pilot only follows one drone at a time. This limits the feasibility of a fleet of drones.

f) Flight range: If a drone flies out of the range of the GCS and the safety pilot's remote control, it can continue to follow its flight plan. However, it is no longer possible to send orders to that drone, nor is it possible to retrieve information. If something unexpected happens when the drone is out of range, the drone could very well get lost, injure people or damage property.

B. Presentation

This project would consist in creating a smartphone app for Android that would interface with the Paparazzi system to reduce the workload for the drone safety pilot, enable the management of more than one drone at a time, and enable safe flight beyond the range of the GCS under certain conditions. Since the app can be used on a smartphone, it can be carried around easily by the safety pilot for field use.

This app enables the pilot to visualize information concerning all registered drones. This app may also be used to monitor and control a fleet of drones. In particular, the pilot should be able to access all relevant information concerning a drone, as well as localize or identify a drone in the sky. The app also enables the pilot to send orders to a drone from the smartphone.

It should also be possible for the drone to send a message to the safety pilot. The smartphone then indicates the position or the direction of the drone. Such messages can be relative to the drone status (e.g. low battery), or the drone environment if said drone is outfitted with sensors.

The final part of this project is to create an Vehicular ad-hoc Network (VANET) within the flock of drones and test their connectivity and communication. In other terms, there will be a Control Center (laptop/smartphone) that will monitor one of the drones which is designated as a Road Side Unit (RSU) and will communicate with the other drones which are designated as On Board Units (OBUs or nodes).

The communication system will be based off of current advancements in collective behaviour and may be inspired by existing communication systems such as ADS-B [2], [3] which allows air-to-air communications as well as air-to-infrastructure communications. The work will be based on existing the existing Paparazzi framework, and will use associated software such as the mission status report module, the real-time plotter (RTPlotter), and the server application (AppServer) [1].

Possible applications for this project include to monitor traffic as to avoid traffic jams where an accident may have occurred, or search and rescue missions where deploying people could be dangerous.

C. Related Work

Following an aircraft in flight has been the subject of a number of studies over the years.

Many smartphone based solutions such as FlightAware [4] or Flight Radar [5] enable the user to know the position, flight status and other information concerning most commercial flights in the world, and follow flights live.

Solutions also exist for controlling drones with a smartphone. For instance ARDrone Flight is the companion app to the Parrot ARDrone for controlling the drone and recording videos [6], [7]. However, it does not provide visual tracking using the phone's camera, and it is not open-source. The PPRZonDroid [8] is an application that you can use to control Paparazzi aircraft with your android device. However, it only replaces the GCS and the user can only follow the trajectory of a drone on a map.

GoogleSky [9] features tracking. It is a Google app that shows the celestial sphere on the smartphone screen by linking the screen display with the screen orientation. In the case of this project, the stars of GoogleSky would be replaced by a fleet of micro-drones.

Recent studies have also been done on the subject of swarming of ground robots or aerial drones [10], collective behavior, and flying ad-hoc networks (FANET) [11], [12]. Many applications for such FANETs are envisionned such as meteorological studies [13], thermal detection for hang-gliders [14] or search and rescue missions [15].

Studies in [16], [17] shows how the ad hoc network in the context of unmanned aerial systems (UAS) is a promising solution.

The apps that currently exist do not integrate drone control, drone monitoring and tracking of multiple drones in a single app. Therefore, even though this project is inspired by existing applications, it is unlike each of those applications.

II. Implementation

The app is divided into four layers

- the user interface,
- the model,
- the network,
- the collective behaviour.

A. User interface

This layer consists of what the safety pilots sees and interacts with through their smartphone.

The app has four distinct screens:

- Flight listing: the list of all drones currently tracked by the app
- Flight data: all information relevant to a selected drone
- Tracking: symbolic representation of the drones over the image taken by the phone camera
- Preferences: network connection parameters, selection of warnings to be displayed,...

This layer includes the threads necessary to switch from one screen to another.

B. Model and controller

This layer includes all the information specific to each drone and the smartphone device. The main elements of the model are the UAV and the Fleet.

1) UAV flight data: This is the most important part of the model, since it concerns critical data. The safety pilot makes decisions concerning the drones based on this data. For each drone, the data that we consider is the following:

- Name, colour and id
- Vertical speed
- Ground speed: this is the speed of the drone relatively to the local Earth frame
- True air speed: this is the speed of the drone relatively to the body of air it is flying through
- Orientation characterised by the roll, pitch and heading of the drone
- Coordinates of longitude and latitude
- Height: this is the distance from the drone to the ground directly under it
- Altitude: this is the vertical position of the drone above the mean sea level

The information concerning each drone is transmitted by the parser of the Networks layer, and written to the screen of the User Interface layer on a regular basis.

2) Fleet: A fleet contains several drones and is controlled by a surveillance device.

The Fleet class is a doorway between the view and the model. Fleet manages the list of drones which can change dynamically.

3) Device data: The data regarding the device includes the orientation and location of the phone. This is used in combination with the location of the drone to provide the tracking feature. The following data regarding the device is considered:

- Orientation characterised by the roll, pitch and azimuth of the device
- Coordinates of altitude, longitude and latitude
- Cone of visibility of the camera

The LocationManager and SensorManager classes of the android library are used to provide this information.

4) User preferences: The preferences that may be set by the user are of several sorts:

- Network preferences: the user may need to modify the IP address or connexion ports

- Warning preferences: the user may want to activate/deactivate certain types of warnings based on the type of mission performed
- ...

C. Networks

This layer includes a parser for converting messages received into data usable by the other layers for the device-drone communications.

The parser does the following tasks:

- establish a connexion;
- receive messages;
- acknowledge the reception of messages;
- process messages;
- format the collected data;
- transmit the collected data to the other modules.

A dedicated thread establishes the connexion with the appServer. Another thread acknowledges the reception of TCP messages.

D. Collective behaviour

This layer includes all the rules for the drones to interact amongst themselves.

If two drones are within communication range, they can compute the distance separating them by measuring the time to transmit messages.

III. Tests

Only the app and the ad hoc network between drones are being tested. It is assumed for this project that Paparazzi and the smartphone are free of bugs.

The features to be tested include the display of the app on the smartphone, the conformity of the information displayed to the safety pilot, and the communications between drones.

The Paparazzi software can be used to simulate the flight of drones. Information can be sent to and from the simulated drones in real-time using the network created by AppServer, the server launched by Paparazzi to obtain information from the drones. It is then possible to connect a smartphone to this network and obtain the simulated flight data and send instructions to the simulated drone.

A. Tests in a simulated environment

B. Tests in a real environment

IV. Conclusion and future work

Conclusions will be drawn from this project, future work will be envisioned.

Acknowledgment

The authors would like to thank Xiang-Yang Li.

References

[1] "Paparazzi, The Free Autopilot," paparazziuav.org, Release of 2015-01-20. [Online]. Available: http://wiki.paparazziuav.org/wiki/Main_Page

[2] "FAA ADS-B Services," www.faa.gov/nextgen/ads-b/broadcastservices/, Release of 2015-01-20. [Online]. Available: http://www.faa.gov/nextgen/ads-b/broadcastservices/

[3] "FAA Announces Automatic Dependent Surveillance-Broadcast Architecture," www.faa.gov/news/press_releases/news_story.cfm?newsId=5520, Release of 2015-01-20.

[4] "FlightAware, Live Flight Tracking," flightaware.com, Release of 2015-01-20. [Online]. Available: http://flightaware.com/

[5] "Flight Radar 24, Live Air Traffic," flightradar24.com, Release of 2015-01-20. [Online]. Available: http://flightradar24.com

[6] "ARDrone Flight Recorder," ardrone2.parrot.com/apps/flight-recorder/, Release of 2015-01-20. [Online]. Available: http://ardrone2.parrot.com/apps/flight-recorder/

[7] "Parrot AR.Drone," ardrone2.parrot.com, Release of 2015-01-20. [Online]. Available: ardrone2.parrot.com

[8] "PPRZonDroid," wiki.paparazziuav.org/wiki/PPRZonDroid, Release of 2015-01-20.

[9] "Google Sky Maps," play.google.com/store/apps/details?id=com.google.android.stardroid, Release of 2015-01-20. [Online]. Available: https://play.google.com/store/apps/details?id=com.google.android.stardroid

[10] G. Vàsàrhelyi, Cs. Viràgh, G. Somorjai, N. Tarcai, T. Szrnyi, T. Nepusz, T. Vicsek, "Outdoor flocking and formation flight with autonomous aerial robots," *IEEE IROS Conference*, 2014.

[11] Q. Vey, A. Pirovano, J. Radzik, and F. Garcia, "Aeronautical ad hoc network for civil aviation," in *Communication Technologies for Vehicles - 6th International Workshop, Nets4Cars/Nets4Trains/Nets4Aircraft 2014, Offenburg, Germany, May 6-7, 2014. Proceedings*, 2014, pp. 81–93. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-06644-8_8

[12] M. Royer, A. Pirovano, and F. Garcia, "Survey on context-aware publish/subscribe systems for VANET," in *Communication Technologies for Vehicles, 5th International Workshop, Nets4Cars/Nets4Trains 2013, Villeneuve d'Ascq, France, May 14-15, 2013. Proceedings*, 2013, pp. 46–58. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-37974-1_4

[13] G. Hattenberger, G. Cayez, and G. Roberts, "Flight tests for meteorological studies with MAV," in *IMAV 2013, International Micro Air Vehicle Conference and Flight Competition*, Toulouse, France, Sep. 2013, p. pp xxxx. [Online]. Available: https://hal-enac.archives-ouvertes.fr/hal-00936235

[14] J. Eckert, D. Eckhoff, and R. German, "Flying Ad-Hoc Network Communication for Detecting Thermals: Feasibility and Insights," in *3rd International Conference on Innovative Computing Technology (INTECH 2013)*. London, UK: IEEE, August 2013, pp. 333–338.

[15] G. D. Cubber, D. Doroftei, Y. Baudoin, D. Serrano, K. Berns, C. Armbrust, K. Chintamani, R. Sabino, S. Ourevitch, and T. Flamma, "Search and Rescue robots developed by the European ICARUS project," vol. 1, 2013.

[16] O. Bouachir, A. Abrassart, F. Garcia, and N. Larrieu, "A Mobility Model For UAV Ad hoc Network," in *ICUAS 2014, International Conference on Unmanned Aircraft Systems*, Orlando, United States, May 2014, pp. pp 383–388. [Online]. Available: https://hal-enac.archives-ouvertes.fr/hal-00998651

[17] O. Bouachir, "Design and implementation of communication architecture for civil mini-UAVs," Theses, Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier), Dec. 2014. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01092318