Indoor Flight of AR Drone 2.0 UAV in Vicon Motion Capture System

John M. Donald

University of New South Wales at the Australian Defence Force Academy

Swarming behavior with artificial intelligence have been extensively simulated as computer based by researchers and students, using unmanned aerial vehicles with autopilot software will allow theses simulations to enter the real-world. In this project, the Paparazzi UAV autopilot software is to be integrated with the Vicon motion capture system allowing an AR Drone 2.0 to fly autonomously. The solution provided indicates that the Paparazzi software can be integrated with a Vicon system. The solution has the potential to be used on various unmanned aerial vehicle platforms.

Contents

I. Introduction 2
II. Background 2
III. Concept Map 3
IV. Methodology 3
V. Results and Discussion 5
   A. Quantitative Results 5
   B. Qualitative Results 6
VI. Project Management 7
VII. Engineering Lessons Learned 8
VIII. Conclusions 8
IX. Recommendations 8

References 9

Abbreviations

3D Three Dimensional
GCS Ground Control Station
GUI Graphics User Interface
GPS Global Positioning System
IDE Integrated Development Environment
INS Inertial Navigation Systems
PPRZ Paparazzi UAV software source code
RPM Revolutions Per Minute
UAVs Unmanned Aerial Vehicles
UDP User Datagram Protocol
VICON Motion Capture and Detection System

1 PLTOFF, School of Engineering & Information Technology. ZEIT4501
I. Introduction

There is significant interest in autonomous micro unmanned aerial vehicles (UAVs) and their potential applications [1] [2]. There have been substantial computer simulations regarding single and multiple UAVs in agriculture environments such as herding animals [3] [4]. There has been significant progress in the field of autonomous UAV flight with several commercial and open source autopilot software available such as Paparazzi UAV, Ardupilot and Pixhawk. This project concerns itself with providing the capability of autonomous drone flight in an indoor environment, which will enable students and researchers to further testing and evaluating computer simulations into a real-world environment. By utilizing a motion capture system, repeatable and accurate experiments can be conducted allowing simulations to be tested and tuned to real world effects in a consistent manner. The integration of the Paparazzi UAV autopilot software with the Vicon motion capture system will allow various UAV platforms to be flown increasing the range of testing simulations.

II. Background

Modern micro unmanned air vehicles are equipped with numerous sensors that are technically sophisticated to enable autonomous flight capability. The on-board sensors could be a camera, either downward or forward looking which can be used for optic-flow guidance [5]; other sensors include accelerometers to enable inertia navigation systems (INS) which is commonly used in conjunction with global positioning systems (GPS) [6] to provide accurate guidance, however, INS systems by themselves suffer from INS drift which has a cumulative effect resulting in inaccurate positioning over time which may not be appropriate for repeatable experiments. Motion capture systems, such as the Vicon system, provide the ability to accurately detect the drone position within millimeter accuracy [7]. Due to the sensors not being on-board like other methods of guidance if the drone crashes it can be expected that there is no reduction in the confidence of the sensor providing the guidance method, unlike on-board sensors which may be damaged leading to costly repairs or drone replacement.

An important aspect of software engineering is the ability to modify existing software to adapt or upgrade hardware to provide an entirely new capability. With the increasing popularity of open-source code development, which are generally well maintained and supported by global communities the ability to develop a new capability can be conducted with a rapid development time and a relatively low cost. This is due to the burdens of hardware prototyping and software development being greatly reduced. During this project the Paparazzi UAV Autopilot software is utilized as it already commercial grade and proven ready for use with full flight planning features. The Paparazzi UAV Autopilot Software (PPRZ) is a flexible system that allows for modularity in systems to be uploaded to the drone. The most important feature of PPRZ is the is that users have full access to the source code, this enables a white box working environment. Paparazzi UAV software has been utilized by researchers since providing mostly outdoors capability using GPS as the primary guidance method [8]. In a GPS denied environment TU Delft University has extensively utilized a motion capture system ‘Optitrack’ [9] [10] which is encouraging to understand that it is achievable to integrate the Vicon system, a project which hasn’t currently been conducted.

The AR Drone 2.0 quadrotor UAV is a cheap and robust drone which has a proven capability with the PPRZ system which allows for rapid development of the source code to integrate with Vicon system. The AR Drone 2.0 has an on-board computer with a Linux operating system [11]. There are several sensors on-board including a 3-axis accelerometer for INS capability, a 3-axis magnetometer, barometer, forward looking camera, downward looking camera for optic-flow capability and ultra sound sensor [11]. Even though the AR Drone 2.0 has these sensors, using the Vicon motion capture system will provide a more accurate and reliable guidance method which can then be transferred to other UAV platforms.

Figure 1. AR Drone 2.0 With reflective markers used in the Vicon system to capture position.
Section 3 introduces the concept map of the Paparazzi UAV software used to investigate possible approaches to integrate the PPRZ system and Vicon system together. Section 4 provides the methodology of the approaches and discusses their outcomes. Section 5 reports and discusses the results of the project. Section 6 discusses the project management conducted throughout the project. Section 7 summarizes important engineering lessons learned during the project. Section 8 concludes the work presented in this project. Finally, section 9 discusses recommendation for future works to continue the work presented in this project.

III. Concept Map

The PPRZ autopilot communications is divided into three segments – Airborne, Ground and Communication [12]. The Airborne segment is uploaded onto the drone and is responsible for communication telemetry data regarding the on-board sensors and hardware status, battery voltage level and drone position (which conventionally is provided by the on-board GPS module). The Ground segment is situated on a laptop and is responsible for the ground control station interface and flight plan commands sent to the drone via a datalink. The Communication segment is linked to the ground segment and provides communication between the ground segment and the drone via the Ivy bus which enables single or multiple drone communication.

The PPRZ software is divided into directories which closely follow the communication segments. The airborne section allows for inclusion of the drone platform, airframe files which allows for specific sensor modules to be incorporated such as how the GPS fix is to be realized i.e. from a GPS module for outdoor flights or datalink method used for indoor flights. The performance of the drone can be changed in the airframe file where inputs to the PID controller such as roll, pitch, yaw and thrust rates can be adjusted as well as motor rotation rates. The ground segment incorporates the flight plan directory where waypoints and flight routes can be set as well as safety perimeters for the drone flight area. Also in the ground segment, the communications segment is included where modules can be added to interface with the internal Ivy bus and externally via sockets to Wi-Fi networks.

The PPRZ software is capable of receiving and handling with multiple coordinate system formats such as cartesian x, y, z; east north up (ENU) and GPS nmea latitude, longitude sentences. The airborne segment is capable of navigation in either of these coordinate systems, but the ground control station always displays the drone position in GPS format.

The PPRZ system has a safety caveat hard coded into the software which cannot be removed which is that it requires a GPS 3-D fix to initiate launch, starting of engines or continued flight plan [13]. This is to ensure the PPRZ system has a reputation for safety. However, this safety mechanism can be satisfied by means other than GPS alone. Carrying on with previous work by TU Delft University within the Optitrack motion capture system [9] [10], the safety mechanism can be overcome by a “remote GPS message” sent to the drone via the Ivy bus from the Ground segment to the Airborne segment on the drone. The “remote GPS message” can be delivered in a short or long format either is recognized by the PPRZ system as an indoors position fix.

Using the Vicon motion capture system to act as the position sensor for the drone requires the drone, ground control laptop and Vicon system to be connected to the same Wi-Fi network. This requires the AR Drone 2.0 software to be modified so that it is converted from using its on-board Wi-Fi to set up a network to joining a Wi-Fi network, this is done using a procedure outlined by Bart Remes et al [14]. Once this configuration change occurs all communications between the ground segment and airborne segment is directed through the hosting Wi-Fi modem.

IV. Methodology

To integrate the Vicon system with the PPRZ software initially a network UDP socket was required to establish communications so that position data could be passed from the Vicon system to the PPRZ software. A UDP socket was considered the best method as the data was only required to be flowing one-way and the rate of position update (100 Hz) would mean that any lost packets would not be required to be received. The next step is to process the Vicon position data which is in a cartesian x, y, z format to a useable format that PPRZ would treat as acceptable to provide a GPS 3-D fix. Then finally, communicate to the drone to direct a flight movement.
Initially, due to the AR Drone 2.0 having an on-board W-Fi modem it was thought possible to use this to provide a drone to Vicon direct link utilizing the airborne segment. In previous versions of PPRZ a GPS module was available known as ‘gps_udp’, however, when this module was incorporated it was discovered that multiple internal links in the software had been removed and through extensive attempts to rejuvenate these links more issues arose. This lead to contacting the PPRZ developers to question the airborne segment as an interface for the UDP connection and due to the low processing power of the on-board computer it was advised that this was not possible without causing other issues with other on-board hardware such as the motors.

This lead to the second option, establishing a UDP socket connection with the ground segment. Following the work conducted by TU Delft University [9] [10] in the Optitrack motion capture system it was thought this was the most plausible method to establish communications and conduct the required coordinate transformation to create the ‘remote GPS message’. Firstly, all the required files and directories where copied into a separate environment known as ‘Vicon Link’ separate from the PPRZ software. The Optitrack module used by TU Delft University was modelled to ensure the method used to communicate with Ivy bus and utilize the PPRZ math functions such Euler’s angles remained but the UDP connection and extracting the Vicon position data was created. The UDP connection between the ground segment and Vicon system was successful as well as the coordinate transformation from cartesian x, y, z into ENU position which is required for the ‘remote GPS message’. This directory was then re-inserted back into the PPRZ software and a specialized tool was created so that a user can select and initiate Vicon Link from the ground segment Paparazzi window. However, when Vicon Link was initiated it appeared that the computer “crashed”, possible causes thought of at the time was the Ivy bus had not been installed on the ground station or there was a continuous loop in Vicon Link when calling the Ivy bus or an issue with how the Ivy bus was being utilized to communicate with the drone i.e. the socket the Ivy bus was attempting to communicate with was not available. It should be noted that when the Vicon Link module is running outside of the PPRZ software the processing rate is in real-time with no noticeable delays.

While working on the second option a third option was initiated to eliminate the possibility the Ivy bus was not a working communication link on PPRZ anymore. This option involved utilizing the Vicon Link module created previously and extra hardware devices known as XBee Wi-Fi modems. With one XBee device connected to the ground station laptop and the other connected to the AR Drone 2.0 USB port. XBee Wi-Fi modems were chosen as they provide easy set-up between similar devices, they can be assigned to a network or create a Wi-Fi network between themselves which has further possibilities with multiple drone swarms. Once again, the Vicon Link module was removed from the PPRZ software and the XBee devices connected to USB ports on the ground station computer, the data was transformed and successfully transmitted between XBee devices. Then the receiving XBee device was connected to the AR Drone 2.0 without PPRZ uploaded the data was transmitted once again and via telnet the drone busy box the remote GPS message could be viewed, this indicates the data can be successfully transmitted. However, when the PPRZ software is uploaded to the AR Drone 2.0 the voltage delivered to the USB port is reduced to a level below which the XBee device operates (3.5 Volts). This indicates that investigation into the module that controls the USB port voltage needs to be conducted.

A fourth option was also investigated, this was the sensor solution. Involving utilizing the on-board accelerometers used for INS and the on-board magnetometers to align with the local earth magnetic field which is another method to provide a GPS 3-D fix. This method had previously only been conducted successfully outdoors [15] and was discounted to be performed in this project due to the fact that there are large metal structures surrounding the Vicon arena which would likely provide unreliable position data and due to the previously mentioned INS drift issue.

Returning to the second option investigation, Vicon Link between the ground segment and the Vicon system, it was discovered that the computer “crash” was an extremely long computation processing issue. After initiating Vicon Link on the PPRZ ground segment window approximately 10 minutes of processing then occurred, where the computer screen ‘greyed out’, the computer screen returns to normal with the expected data required to build the remote GPS message such ENU position and speed appeared in the terminal window. Inspecting the ground control station Graphics interface indicated that there was a GPS 3-D fix. This indicates that the Vicon Link does successfully transform the Vicon position coordinates into ENU coordinates then transmits the remote GPS message using the Ivy bus.
V. Results and Discussion

A. Quantitative Results

![Graphs showing GPS distribution](image)

Figure 3. GPS Distribution Outdoors. **Left:** Latitude Distribution Histogram with a bell curve of the distribution with a mean $\mu = -35.291142^\circ$ and a standard deviation $\sigma = 5.98 \times 10^{-6}$ decimal degrees. **Right:** Longitude Distribution Histogram with a bell curve of the distribution with a mean $\mu = 142.168126^\circ$ and a standard deviation $\sigma = 8.97 \times 10^{-6}$ decimal degrees.

To understand any improvements offered by using the Vicon system the accuracy of the drone position was assessed outdoors. It was found that the accuracy of the Neo-M8 GPS module was $\pm 3.58$ meters in the latitude direction and $\pm 3.12$ meters in the longitudinal direction. This accuracy is clearly not acceptable when attempting to create a capability that is accurate and repeatable, this validates the aim to develop integration of the PPRZ system with a highly accurate position sensing system as the Vicon system.

<table>
<thead>
<tr>
<th>Way Point</th>
<th>Vicon Position [cm]</th>
<th>ENU Position [hex]</th>
<th>ENU speed [m/s]</th>
<th>Drone GPS Position [decimal degrees]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td>Latitude</td>
</tr>
<tr>
<td>Home</td>
<td>32.743</td>
<td>-30.779</td>
<td>0.720</td>
<td>69175296</td>
</tr>
<tr>
<td>P1</td>
<td>178.003</td>
<td>122.590</td>
<td>-0.333</td>
<td>373417984</td>
</tr>
<tr>
<td>P2</td>
<td>-120.032</td>
<td>117.216</td>
<td>-3.079</td>
<td>4043428867</td>
</tr>
<tr>
<td>P3</td>
<td>-117.005</td>
<td>-180.452</td>
<td>-3.853</td>
<td>4051513347</td>
</tr>
<tr>
<td>P4</td>
<td>178.648</td>
<td>-174.393</td>
<td>1.098</td>
<td>375212033</td>
</tr>
</tbody>
</table>

Table 2: Comparison of Distance Between Way-Points.

<table>
<thead>
<tr>
<th>Leg</th>
<th>Vicon Distance [m]</th>
<th>Drone GPS Distance [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>P1 - P2</td>
<td>2.9803</td>
<td>0.0053</td>
</tr>
<tr>
<td>P2 - P3</td>
<td>0.0030</td>
<td>2.9766</td>
</tr>
<tr>
<td>P3 - P4</td>
<td>2.9565</td>
<td>0.0605</td>
</tr>
<tr>
<td>P4 - P1</td>
<td>0.0064</td>
<td>2.9698</td>
</tr>
</tbody>
</table>

Analyzing the data in table 2, with the X-direction compared to Latitude there is an error of $\pm 0.1014$ metres and in the Y-direction compared to Longitude an error of $\pm 0.6218$ metres. This indicates that there is either an error in one of the coordinate translation formulas or a cumulative effect with rounding.

Final Project Report 2017, UNSW Canberra at ADFA
B. Qualitative Results

Throughout the project several approaches utilized which allows for ruling out methods for future work.

For each approach the required PPRZ directories and files were copied into a completely separate directory so that additional PPRZ files did not interef with the process and there was confidence in how the code was operating. When the desired UDP connection and data transformation were functioning correctly the files were refitted to the PPRZ code and then uploaded to the drone using the PPRZ center.

Following the approach to establish a UDP network socket between the drone airborne segment and Vicon system, this method can be discarded as after discussions with PPRZ developers as the AR Drone 2.0 on-board processor is not capable of processing the coordinate transformations at a rate which would enable the drone to know its position in real time. Also, as stated in the section 4 the present version of PPRZ v5.12 does not support the airborne segment to run the module ‘gps_udp’.

Using XBee Wi-Fi modems to establish communications between the ground segment and airborne segment was unsuccessful due to the USB port not providing the required 3.5 volts after PPRZ software is uploaded to the drone. Further investigation into where in the PPRZ source code the voltage is regulated to the USB port is required. The use of XBee devices is a promising path with swarming where they can be utilized to establish their own network easily which would reduce the bandwidth requirements over the Vicon modem Wi-Fi network.

Using on-board sensors to provide guidance is another viable option, during this project the 3-axis accelerometer and 3-axis magnetometer option was investigated. This method had been proven by Santana et al [15] for outdoor flight, but it proved unsuccessful indoors in this project. The reasons for this is believed to be the fact that the Vicon system is located inside a building with a steel frame and large metal objects which can disrupt the local magnetic field. Other on-board sensors worth investigating is the forward or downward looking camera using the optic flow method. However, using on-board sensors may not be as an accurate guidance method as using the Vicon system.

Finally, establishing a UDP network connection between the ground segment and the Vicon system proved to be successful in the manner that the Vicon Link module that was created did perform its intended task to establish a UDP connection with the Vicon system, process coordinate transformations (see table 1) and then transmit the remote GPS message to the drone via the Ivy bus link, see figure 4 for a flow of the position data.

![Figure 4. Communication and Position Data Flow.](image)

*Figure 4. Communication and Position Data Flow.* In the Vicon arena, the Vicon cameras capture the drone and passes this to the ground segment as a cartesian x, y, z position. The ground segment then transforms the x, y, z to ENU coordinates using the Vicon Link module. The ENU coordinates are then used to form the ‘remote GPS message‘ which is transmitted to the drone via the Ivy bus. The drone recognizes the ‘remote GPS message’ as a method to obtain an indoor 3-D fix, then the drone transmits to the ground segment via the telemetry link. The ground segment then displays a 3-D fix and the user can select a way-point or flight path for the drone to fly. This is then transmitted to the drone via the datalink which results in the drone moving to the desired location.
However, due to the extremely long delay in processing there is no real-time position data being sent to the drone and when an attempt to launch the drone into a hover the drone initially took off but landed within 1 second as it was not receiving any position or speed updates. To overcome the extremely long delay in processing the position data initially the transmit rate from the Vicon system was reduced from 100 Hz to 30 Hz (the minimum Vicon setting) which reduced the processing time from approximately 10 minutes to approximately 7 minutes, this is still far too long to be acceptable. To increase the processing rate, it is expected that a computer with a faster processor is to be used, currently the ground station laptop has a 2 GHz processor. Another option is to investigate further the XBee Wi-Fi module solution and send the data via a socket connection or to set up another UDP connection between the drone and ground control station.

VI. Project Management

By following the procedure outlined in ISO/IEC 12207 – Software Life Cycle Processes [16], the software development lifecycle model selected to implement in this project is a waterfall model. This model was chosen as each stage flows onto the next, i.e. the work product deliverables flow down the step wise development path. Following this software development cycle, it can be expected that in the first release a core functionality will achieved, i.e. automatic way-point flight plan being followed by the UAV. Another key step is to identify the individual activities and tasks to be performed, a work break down structure has been created along with a Gantt chart to ensure the project remains on track to achieve key milestones in developing the software.

There are four key phases to the project as prescribed by ISO/IEC 12207: 1) requirements analysis, 2) design, 3) implementation and integration, 4) verification and validation testing.

The requirements analysis of this project is essential to understand properly what the project is to deliver, and this leads to a more effective approach to planning and executing work.

The Functional System Requirements are:
- The GPS position system must provide accurate data to the UAV and GCS;
- There should be an autonomous functionality to the UAV

Non-Functional Requirements
- Safety boundaries should be established regarding fly and no-fly zones
- Flight characteristic data such as roll/pitch/yaw rate should be maintained

Constraints
- The UAV must be an AR Drone 2 quadcopter
- The autopilot software must be Paparazzi UAV
- The flight environment is the Vicon motion capture system arena

The system design is highly restricted to the software architecture of the paparazzi UAV software. This does not mean that no changes will be implemented, only that these changes will have to follow the structure of the PPRZ software.

Implementation and Integration will be the most intensive phase with the integration of the Vicon position data into GPS position data for UAV navigation. Another key area identified will be the integration of the UAV and GCS onto the same Wi-Fi network as the Vicon.

Verification and validation testing will be conducted after each implementation phase to ensure that the product produced meets the desired requirements to be integrated into the system.

Major risks identified to the project are: GPS navigation system unable to be integrated or poor accuracy, this poses the greatest risk to the project as without this functionality the UAV cannot be flown autonomously. The possible alternative to this issue is to fly the UAV outdoors, there is a reduction in accuracy, but the GPS position is still present and the project can continue. Another risk is the sophisticated coding of the PPRZ software, and the limited prior exposure to this level of coding in multiple programming languages. Lastly, real-world environment interactions between several independent systems with multiple interfaces pose a risk, as some interfaces such as the drone hardware to software is black box.
VII. Engineering Lessons Learned

Starting with establishing an understanding of the engineering problem overarching the project and then planning stages to break down the big picture into manageable with tasks to achieve a final product. By identifying key milestone events and having developed a schedule on a Gantt chart meant the project could remain focused and manageable, even when tasks failed to realize their intended goal. Also, being flexible to identify several approaches to realize how a milestone is to be achieved allowed the project to continue when dead ends were encountered.

Due to this project comprising of three major systems that were proven ready for individual use but integrating these together had not been yet achieved. So, the interfaces between the systems had to be identified and understood to maximize their potential for integrating together. Some connections between interfaces had to be taken for granted that would operate without development such as Wi-Fi and the software – hardware interactions on the drone.

This project had several constraints such as the operating system required, Linux with PPRZ also the AR Drone 2.0 and Windows with the Vicon system which required making many decisions as to what version of Linux was appropriate to the software, as it was discovered that some versions of Ubuntu were not compatible with the PPRZ software. Also, identifying what integrated development environment could work with multiple languages and operate on the Ubuntu operating system. These decisions may not seem arduous to experienced engineers but to an inexperienced junior engineer having to go through this process was a valuable experience leading to a greater understanding of the groundwork required to provide an appropriate base to begin work.

This project also developed another fundamental skill for engineers, networking with other researchers and peers. During the project, using research papers, several key relationships were built with Paparazzi developers in France which furthered the knowledge and understanding of the project in a more rapid manner than by working alone. This meant planning working hours that were friendly to their time zone so on-line chat was effective, this was essential as waiting for e-mail responses proved frustrating and time consuming as questions and answers could lead to lengthy delays rather than fast conversations in chat groups. Locally, developing relationships with peers working with UAV projects in the Vicon system increased software development work rate in integrating systems over the Wi-Fi network, this also reduced some work load regarding how the Vicon system detects objects and passes the position to the ground station.

VIII. Conclusion

In order to ensure accurate and reliable development of UAV computer based simulations into a real-world environment, the Vicon system has to be incorporated with an autonomous software such as Paparazzi UAV. Initial stages of integration have been achieved with a module developed which allows for communication between the Vicon system to the ground segment of the Paparazzi software, then processing the remote GPS message and finally this is communicated over the Ivy bus to the airborne segment of the Paparazzi software. While further work is required to increase processing time to real-time rates and further improve accuracy of reported position, a clear path has been established for further work to continue.

IX. Future Work Recommendations

Future work should begin with improving the processing time of the coordinate transformations in the module ‘Vicon Link’. This could be firstly approached by assessing the Paparazzi math methods used to carry out the transformations, it has been noted that at times the PPRZ source code developed can be unnecessarily complicated. Secondly to improve the processing rate using a laptop with a faster processor.

Another improvement on the project will be to include a remote-control device that can allow the user to switch from autonomous flight to guided flight. This is will be a safety device that should prevent unnecessary damage to the drone. Currently, the safety measure is a kill button on the ground control station interface that is slow to react due to having to select the button then acknowledge the desire to kill the drone motors, generally by then the drone has crashed in an unsafe manner. Practical applications should lead to swarming multiple drones which will require collision avoidance software to be developed. Currently, Paparazzi software has a module that
could do this function, however, experience has led to believe developing a module based on the current module is a better option. Continuing with the swarming aspect, using the Ivy bus to multicast to the swarm either via XBee devices on their own Wi-Fi network or via the Vicon Wi-Fi network. Also, developing remote internet control of drones via the Ivy Bus and VPNs [12], this can lead to controlling single or multiple drones remotely from distant locations.

References


