THE UNIVERSITY OF WESTERN AUSTRALIA

Final Year Project Proposal

Towards Autonomous Search and Discovery in UAVs

Author:
Brett Downing

Supervisors:
Prof. Thomas Braunl
Chris Croft

October 2, 2015

Abstract

Abstract

Contents

1 Introduction

Background, Aims

2 History / prior works / relevance

Chase-cam (Lily cam, ...Dog) Cinematography feature films, Red Bull, Remote Inspection, approached by mining industry technology start-ups

3 Motivation

I really really want one. computer vision assisted control routine Obstacle avoidance Introduction to SLAM systems

4 Project log

Platform Description Frame, Flight controller, processor, gimbal Platform Weaknesses (fitness for purpose etc) NAZA not hackable, deliberately obfuscated comms RC switch did not permit altitude control No telemetry Waypoints handled on the embedded system despite GPS loiter behaviour of the flight computer.

Team Achievements Conversion to Ardupilot 3.3 Mavlink, waypoints, altidude control, failsafes, telemetry Full, and revocable hand-over to external autopilot. Linearised SI units input and output from flight controller Reduction of mass, tidy wiring harness

Web UI and HUD Waypoint modifier using no-fly zones (not integrated with ardupilot exclusion zones) Glyph detection Capture and tagging of images for proprietary offline 3D reconstruction.

5 Object tracker Theory

Position estimation Desired relative pose

Observations and uncertainties Structure from image Computational load Live capture Uncertainties: A good uncertainty model encodes what is known and what is unknown. Good treatment of uncertainties combines knowledge without losing information, or adding assumptions. In the case of structure from image, a system that extracts maximal information from a single camera should be automatically capable of full stereoscopy using only the uncertainty analyses that applied to the monocular case. Monocular Camera uncertainty model Lidar Lite Uncertainty model GPS uncertainty model Time evolution uncertainty models Base Utilities Vector sum combination point sample Kalman style filtering Relative uncertainties Networks of relative measurements Graph traversal Nodes: Objects' time history (including self) Survey Markers Assumptions (GPS objects?) Links Relative Observations Motion Estimates IMU data Kalman style interpolation Current implementation Strengths Encodes appropriate information in a covariance matrix form Cleanly integrates multiple observations into an object model Weaknesses Camera model does not account for uncertainty from angle of pixels, gimbal pose, IMU sample time (angular velocity), GPS drift GPS drift is deliberately ignored in favour of applying a generous velocity uncertainty to the objects. Cannot describe complex distributions (unbounded polynomial order) arbitrary geometry Uncertainty Images from SoggyDog Future Work Velocity interpolation not implemented. Graph traversal (SLAM) not implemented. Describe hyperbolic distribution and flaws Describe Laser beam necking case

6 Test Results

Software tests Gimbal pose, IMU data, GPS location etc Camera uncertainty models combined with assumptions Pretty pictures Multiple observations with time-evolution

Live tests *Able to physically follow one object, while tracking multiple others. Acceleration limits, velocity limits, time-cutoffs. pitch and roll stability Basic colour matching limitations

7 Future Work

Expand uncertainty analysis SLAM, end to end solution SIFT algorithm Optical Flow Recommendations: ROS, ROS, ROS.

8 Conclusions

We have greatly improved the capabilities of the UWA autonomous hexacopter platform, and have brought the code-base up to a level where it is feasible to implement a SLAM process.

Appendix A ArduPilot (Pixhawk) Proposal

ENGINEERING PROPOSAL
Hexacopter flight controller upgrade
24 April 2014

This document outlines the proposal for replacing and upgrading the flight controller used on the hexacopter to a 3DR Pixhawk [1] running the open-source flight control software ArduPilot [2]. This marks a significant departure from the current 'black-box' NAZA flight controller. The primary motivation for replacement is based on the strong concern that the current system is not sustainable: The current system restricts any automation improvements, provides no value to the wider community, and limits opportunities for future students.

The key benefits of switching to ArduPilot are:

- Tighter integration and feedback of the hexacopter state with the automation software;
- Avoiding sensor duplication
- Access to existing ground control and simulation software
- Well documented communication interfaces
- Wide community support
- Already being used in the research field.

ArduPilot was used in the winning entries in the 2012 and 2014 Canberra outback challenges [3, 4] and is the platform of choice in the forum DIYdrones.

The Pixhawk will use the MAVLink communications protocol to communicate with the Raspberry Pi. It is acknowledged that there are risks associated in the switchover, which will be discussed and addressed in this document.

1 Scope

The scope of this project involves:

1. Removing the NAZA-M V1 flight controller, power monitor (PMU) and GPS/Compass module
2. Removing the autonomous/manual switching circuit
3. Installing the 3DR Pixhawk, uBlox Neo-7m GPS/Compass module, power monitor, LED and wireless telemetry kit
4. Reworking the wiring harness between the flight controller and the Raspberry Pi
5. Reconfiguring the automation software to interface with the Pixhawk

2 Feature comparison

Feature	Old (NAZA) system	Proposed (ArduPilot) system
Failsafe (return to base) mode	Yes.	Yes.
Low-power failsafe	Yes, voltage monitor.	Yes, voltage and current monitor.
Auto/Manual switch	Yes (via switching circuit).	Yes, built-in to Pixhawk.
Buzzer	Yes, but not controllable. Pi has separate buzzer.	Yes, controllable. Pi can also have separate buzzer as before.
Fully autonomous (takeoff and waypoint travel)	No.	Yes. Can be programmed for fully automated missions.
IMU (pitch, roll, heading, altitude) and GPS data access	Limited (undocumented, unsupported, deliberately obfuscated interface). Current break-in cable to the GPS is unreliable and may cause damage to the GPS.	Yes. Well documented communications protocol (MAVLink).
Radio telemetry	Possible (not currently installed).	Yes.
Pan/Tilt gimbal for camera	Limited (tilt control only from the Pi).	Yes.
Simulation environment	No.	Yes, both hardware and software (HITL/SITL) with physics simulation [5]
Existing software tools	Flight assistant software performs calibration only.	1. Mission planner - handles system configuration, calibration and programming for autonomous flight [6]. 2. Tower - an Android mobile app for controlling the hexacopter [7]
Modifiable	No, closed system.	Yes. It is possible to modify all aspects of the system (open-source), but not recommended for our application without thorough testing.
Autonomous control from the Raspberry Pi	Yes, via the switching circuit and PWM output.	Yes, digital control via MAVLink (a communication protocol).
Built-in support for the PIKSI GPS	No.	Yes, but it is unclear how mature this support is at this stage.

2.1 Further discussion of features

2.1.1 Failsafe

The ArduPilot configuration has a fail-safe mode equivalent to the NAZA, activated in exactly the same way. Once configured, the flight computer will return-to-land on loss of hand-held transmitter signal. The failsafe can be configured as situations demand.

2.1.2 Modification of the flight controller

While it is possible to modify ArduPilot itself, changes to the low-level controls is not recommended nor required. This option will remain open for future work but will require extensive testing. The ArduPilot's default flight modes are sufficient for the scope of our work.

2.1.3 Camera gimbal

The pan servo cannot be controlled by the Pi due to a limitation of the NAZA. This is an out-of-the-box feature on the ArduPilot.

2.1.4 Sensor availability

The Raspberry Pi cannot access pan or tilt data as collected by the NAZA. Any pan/tilt data received will be different to the NAZA, causing the Pi to fight the NAZA. MAVLink provides access to the GPS and IMU data directly from the flight computer, which allows both the QStarz GPS and the XSens IMU to be removed. MAVLink also reports remaining battery capacity, which will allow us to write software that issues warnings and aborts missions intelligently.

2.1.5 Telemetry downlink

A new redundant data channel will permit monitoring outside range of WiFi, or in the case of server failure, which has caused near misses in the past.

The ArduPilot software supports fully autonomous missions, with instruction sent over this link. However, we expect to maintain the availability of manual override at all times and will configure the fail-safe to reflect this.

2.1.6 Control debugging

Using an external monitor interferes with the current control interface between the NAZA and Raspberry Pi. The proposed modification will save a great deal of time and effort in working around this problem.

The PWM signal from ServoBlaster must be calibrated with the NAZA flight control software. With no hardware PWM outputs, exact calibration with the NAZA flight computer is difficult. This is not an issue using MAVLink as it is a digital protocol.

2.1.7 Additional sensor compatibility

The ArduPilot community has added support for a wide variety of sensors over a wide variety of interfaces. This support includes integration into the navigation Kalman filter where appropriate.

2.1.8 Forward compatibility

DJI has released two new flight computers since the NAZA-M V1. The NAZA-M V1 is still supported, but the latest firmware updates require a CAN bus expander valued at A$80 to make use of any new features. Very few of these new features address our current concerns.

The Pixhawk was designed to replace the APM2.6 as the firmware files became too large for the 8-bit processor it carried. It was designed to be largely future-proof, boasting a generous amount of ram, flash and CPU power. Even if the ArduPilot project out-grows the Pixhawk, it is likely to have continued support and backported features, such as the APM2.6 owners currently enjoy.

2.1.9 Portability to future platforms

The ArduPilot project began with fixed wing aircraft and has since been extended to various configurations of multi-rotor, Helicopters, Rovers and Boats. Any work done building functionality against ArduCopter (the multirotor firmware) is immediately relevant to almost any other autonomous platform.

2.2 Alternative solutions

2.2.1 Upgrade the current flight controller

As a commercial system, there is *no* official interface that allows for automation or access to any of its sensors. In the past, this project achieved automation by using a remotely controlled switching circuit and the Raspberry Pi to emulate the PWM signals of joystick commands (aileron, elevator and rudder controls). This approach also requires the Raspberry Pi to have its own set of sensors, where progress would be bogged down in trying to interface with the low-level components and having to re-invent, test and tune basic control functions.

Although there is some limited upgrade capability (upgrading to the PMU V2), which would provide access to the IMU data, this is only possible through an unofficial, undocumented, reverse-engineered system. Even with this extra information, this does not address the other issues outlined in terms of community support, the camera gimbal, telemetry and the existing level of software available for the ArduPilot platform.

2.2.2 APM 2.6

The APM2.6 is a known-good ArduPilot board. The hardware is cheaper than the Pixhawk, but it is version capped as of last year. This board is not future-proof and will not benefit from new features in ArduPilot.

2.2.3 MultiWii

MultiWii is another general purpose open-source flight controller. Originally developed to use the gyroscope and accelerometer system from the Nintendo Wii game controller, it has developed into a general flight controller that operates on the Arduino platform. Community support for this platform is less than that for the ArduPilot. The platform also has lesser specs than the out-moded APM2.6.

2.2.4 Paparazzi

Paparazzi is another open-source flight controller system, which has been developed since 2003. It is an older project that appears to be extremely versatile, but not very beginner friendly. Their focus appears to be on modularity and wide applications. The community encourages significant modification of the core flight control software, which is likely to remain out of scope for this platform. Many of the demonstrated applications are single-purpose scientific flights.

2.2.5 OpenPilot

The OpenPilot appears to have a strong community backing for FPV and acrobatics, but the ArduPilot appears to have a sample of simpler flight modes and a stronger autonomous focus, and a stronger Australian community. The OpenPilot hardware does support ArduCopter, but does not have the extensive feature set of the Pixhawk, nor its sensor redundancy.

3 Risks

3.1 Risk matrix

	Very unlikely	Unlikely	Possible	Likely	Very Likely
Negligible	Very low	Very low	Very low	Low	Low
Minor	Very low	Very low	Low	Low	Medium
Moderate	Low	Low	Medium	Medium	High
Significant	Medium	Medium	High	High	Very high
Severe	Medium	High	High	Very high	Very high

3.2 Risk register

No.	Description	Probability	Impact	Rating	Mitigation	Contingency
1.1	The UAV community moves to a different platform	Unlikely	Moderate	Low	Many people, including developers have invested in ArduPilot compatible hardware which represents a small community lock-in	Find alternative, compatible flight controller software
1.2	Unforeseen hardware incompatibilities	Possible	Moderate	Medium	Conduct prior investigation and identify any potential hardware issues	Consult community for potential fix. OR replace incompatible hardware
2.2	Late deliveries or manufacturing faults	Possible	Significant	High	Request priority delivery, AND Much of the integration testing can be performed with an existing ArduPilot owned personally by a team member.	Source parts elsewhere

3.3 Further risk comments

Risk 1.1 - The UAV community moves to a different platform

The developer forums are extremely active, and new features are being added regularly. Of the open-source systems discussed, ArduPilot is the most actively developed [8, 9, 10, 11]. If an open-source flight controller is to be used, it makes sense to use the most actively developed platform to avoid project stagnation. Should ArduPilot development cease, there are a number of compelling platforms with a great deal of support - OpenPilot, MultiWii and Paparazzi.

Consequences
External development of the ArduPilot software would halt, but it would still be available in its current state. Being an open source project, many of the alternative drone software projects already support the Pixhawk hardware, just as ArduPilot has been ported to hardware from other projects.

Risk 1.2 - Unforeseen hardware incompatibilities

The Hexacopter has 30A Opto ESCs and a Futaba R7008SB receiver [12]. Support for these components with the ArduPilot has been demonstrated in other UAV projects, so hardware incompatibility is unlikely. The Raspberry Pi has been demonstrated to work with ArduPilot [13].

Consequences
Hardware incompatibilities would result in additional debugging work and lost time. It could also result in the replacement of other hardware, or the purchase of additional interfacing hardware.

Controls
Preliminary research into each subsystem of current hexacopter.

Risk 2.2 - Late deliveries or manufacturing faults

Online purchases carry the risk of components not arriving on time. It may also make it more difficult to return components, should they be faulty.

Consequences
Late deliveries would lead to lost time, potentially putting the project on hold or incurring additional costs.

4 Costs

Component	Cost ($AUD)	Source	Comments
Pixhawk	$256	3DR store	
UBlox GPS + Honeywell Compass	$41	eBay	
Telemetry	$32	eBay	Not strictly necessary, but allows in-flight full recovery after a complete server crash
Indicator LED	$12	eBay	

Total cost: A$341 (24/04/2015)

5 Conclusion

We fully expect the components listed above to serve as a drop-in replacement for the NAZA and other ancillary components.

The removal of the NAZA flight computer and the inclusion of the Pixhawk radically simplifies the wiring harness, removing many of the components and links that the 2014 team complained about being ‘unreliable’ and the cause of multiple crashes.

The move to the Pixhawk will greatly increase the maintainability of the platform and significantly widen the plausible scope of any future work with this hardware.

References

[1] 3D Robotics, "3DR Pixhawk." http://3drobotics.com/pixhawk/, 2015.

[2] DIY Drones team, "ArduPilot." http://ardupilot.com/, 2015.

[3] C. Anderson, "CanberraUAV (ArduPlane) wins Outback Challenge!." http://diydrones.com/profiles/blogs/canberrauav-arduplane-wins-outback-challenge, 2012.

[4] G. Mortimer, "Congratulations Canberra UAV, stunning Outback Challenge win." http://diydrones.com/profiles/blogs/congratulations-canberra-uav-stunning-outback-challenge-win, 2014.

[5] Ardupilot community, "Simulation." http://dev.ardupilot.com/wiki/simulation-2/, 2015.

[6] M. Oborne, "Mission planner home." http://planner.ardupilot.com/, 2015.

[7] 3D Robotics, "Tower (DroidPlanner 3)." https://play.google.com/store/apps/details?id=org.droidplanner.android, 2015.

[8] A. Tridgell, R. Mackay, P. Hickey, *et al.*, "ArduPilot source code." https://github.com/diydrones/ardupilot/, 2015.

[9] OpenPilot Foundation, "OpenPilot source code." https://github.com/openpilot/OpenPilot, 2015.

[10] F. Ruess, G. Hattenberger, C. D. Wagter, *et al.*, "Paparazzi source code." https://github.com/paparazzi/paparazzi, 2015.

[11] MultiWii Authors, "MultiWii source code." https://github.com/multiwii/multiwii-firmware, 2015.

[12] NoLimitLab, "Setting up Futaba 14SG and R7008SB to work with the pixhawk." https://nolimitlab.wordpress.com/resources/setting-up-futaba-14sg-and-r7008sb-to-work-with-the-pixhawk/, 2015.

[13] ArduPilot community, "Communicating with Raspberry Pi via MAVLink." http://dev.ardupilot.com/wiki/companion-computers/raspberry-pi-via-mavlink/, 2015.