Real Time, Onboard-only Landing Site Evaluation for Autonomous Drones

by

Joshua Springer

Thesis proposal submitted to the School of Computer Science
at Reykjavík University in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

September 2021

Thesis Committee:

Marcel Kyas, Supervisor
Professor, Reykjavík University, Iceland

Gylfi Þór Guðmundsson, Supervisor
Adjunct Professor, Reykjavík University, Iceland

Joseph Foley, Advisor
Professor, Reykjavík University, Iceland

External Person, Examiner
Role, University, Country

Real Time, Onboard-only Landing Site Evaluation for Autonomous Drones

Joshua Springer

September 2021

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Ég veit ekki hvernig á að tala islensku

Joshua Springer

September 2021

Útdráttur

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetuer a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetuer. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetuer tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

Acknowledgements

I would like to thank the Flying Spaghetii Monster and his noodly appendage.

Contents

Chapter 1

Introduction

1.1 Problem Statement and Motivation

The goal of the proposed research is to explore the topic of autonomous, unstructured drone landing. Current autonomous landing methods have at least one of the following disadvantages: they are blind to obstacles, they require previously *known* landing sites, they depend on sophisticated ground control stations for offloading of expensive computation. This proposed research targets a gap in current autonomous landing methods. Specifically, we aim to develop a method for quickly analyzing terrain and identifying safe landing sites using only embedded computational hardware and a minimal set of sensors.

Landing is a particularly difficult aspect of drone flight, owing mainly to its risky nature and required precision. As a result, most drone landings are carried out by (or under the supervision of) a human operator, inherently limiting the applicability of autonomous drones. Some autopilot software includes an Application Programming Interface (API) for *precision landing*, which allows a drone to localize and direct itself with respect to a landing pad during an autonomous landing, according to data provided by external sensors and programs. However, there is no particular method of autonomous landing in widespread use. As autonomous and semi-autonomous drones are not able to reliably handle landings on rough terrain or in non-ideal conditions, human operators often disable autonomous control during landing (opting for full manual control), or abuse/hack the landing system by descending to a low altitude, grabbing hold of the drone, and disabling the motors, as shown in Figure 1.1. Aside from potentially exposing users to dangerous rotors, this landing technique showcases the limitations induced by a lack of autonomous landing method.

In sufficiently flat, large areas, fully autonomous drone missions can end with a GPS-based autonomous landing which is blind to obstacles in the environment. However, intuitively and demonstrably, this can lead to crash-landings at landing sites that have obstacles within the error radius of the GPS, which can be anywhere from a few centimeters to a few meters. In the available open source autopilot softwares, obstacles are simply not handled, and drones will continue their landing attempts even if fatally obstructed.

1.2 Background

1.2.1 Autopilot Software/Hardware

The most prominent, open source drone autopilot software packages are ArduPilot [1] and PX4 [4], which can integrate easily with many additional/custom software packages. DJI drones, while the most commonly used consumer-grade drones, use proprietary, closed source autopilot software that has a limited API for interacting with external software. Thus, ArduPilot and PX4 and custom drones are typically used for research on drones themselves, while DJI software

Figure 1.1: Non-ideal, human-assisted landing in the absence of an autonomous, safe landing method that considers the surrounding environment.

and drones are typically used for consumer/commercial tasks. ArduPilot and PX4 communicate using the same open source, customizable protocol - MAVLink [2] - which has APIs in many different programming languages as well as with Robot Operating System (ROS).

1.2.2 Simulation Software

1.2.3 Robotics Software

1.2.4 Fiducial Markers

1.3 Related Work

Chapter 2

Current Progress

2.1 Initial Hexacopters

After finishing a master thesis [5] wherein I developed an algorithm for autonomously landing a drone using fiducial markers in simulation, the next step was to test this method on physical platforms. The algorithm required identifying fiducial markers through image analysis, tracking the markers via a gimbal-mounted camera, calculating position targets in order to direct the drone towards the landing pad, and communicating those position targets to the flight control software. The base frame for the drones are the Tarot 680 hexacopter kit, which provides a good thrust-to-weight ratio, good flight stability, and space for mounting multiple computational components. A combination of Raspberry Pi and Navio2 [3] shield serve as a flight controller which can communicate with a companion board. The companion boards (a Google Coral Dev board and an NVIDIA Jetson Nano) communicate via a USB network to the flight controller and perform all heavy computations involving image analysis, coordinate system transforms, PID control, and position target generation.

An overview the components is as follows:

- **11.1 V LiPo Battery:** this battery provides power to a battery eliminator circuit (BEC) for isolation of the power system for the computational electronics (the flight controller and companion board).
- **BEC (Battery Eliminator Circuit):** the BEC transforms 11.1V power to 5V power for the flight controller and companion board. The flight controller and companion board each have their own 4A channel to meet their given power requirements.
- **Flight Controller:** this combination of a Raspberry Pi 3 B+ and Navio2 shield runs the ArduPilot software to control the drone, and communicates with the companion board to control the gimbal.
- **Telemetry Radio:** the telemetry radio provides two-way communication between the flight controller and a ground control station that is also fitted with its own telemetry radio. It is connected to the flight controller via USB. The software on the ground control station provides an interface for real-time status messages and sending high-level commands.
- **RC Receiver:** the RC receiver provides a one-way radio link between the pilot's transmitter and the flight controller, allowing the pilot to manually control the drone. It is connected to the flight controller via SBus which provides an 8-channel multiplexed PWM signal to reduce the needed wires and space. This provides an interface for control by a human pilot, which is often used in testing but will eventually be mostly unused.
- **22.2 V LiPo Battery:** this battery provides power to the speed controllers and gimbal.

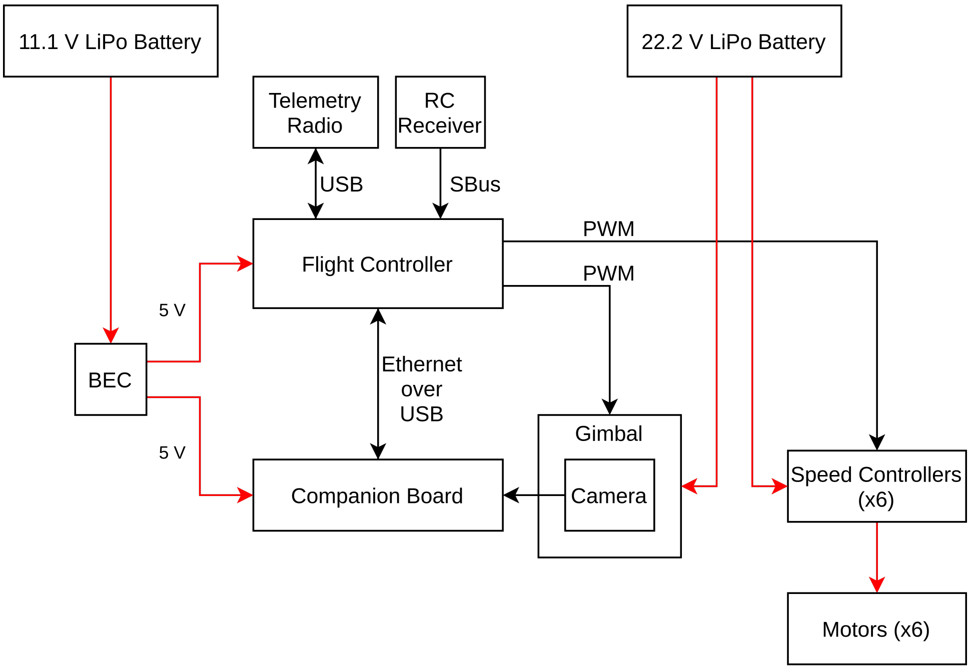


Figure 2.1: Hardware Setup

- **Speed Controllers:** the speed controllers receive a PWM signal from the flight controller which indicates a throttle value. They then provide corresponding power signals to the motors.
- **Motors:** the motors spin propellers to provide thrust in order to control the drone's position in the air.
- **Gimbal:** the gimbal controls the orientation of the camera based on PWM signals from the flight controller which indicate target angles. Its onboard IMU and driver filter the motion of the camera in order to provide a smooth camera image.
- **Companion Board:** the companion board reads an image from the camera and calculates the position of the landing pad relative to the drone. It then communicates this information to the flight controller via an Ethernet over USB connection using ROS.

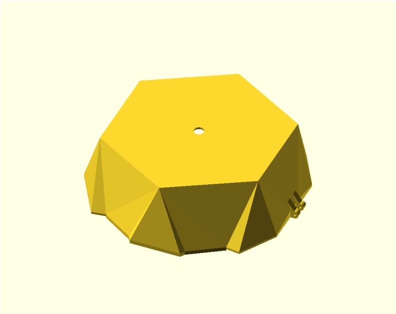
The computational components require some protection from the harsh Icelandic weather, and we therefore designed and 3D-printed a component mounting plate with a connector for a canopy. We also designed and printed cases to protect camera modules and allow them to be mounted in a gimbal with a GoPro form factor. The final versions of these components (after several iterations) can be seen in Figure 2.2. The fully assembled hexacopters are shown in Figure 2.3

2.2 WhyCode Modifications

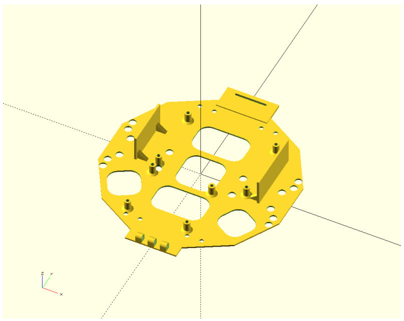
2.3 April Tag Modifications

2.4 Infrared Camera, Heavy-Lift Drone

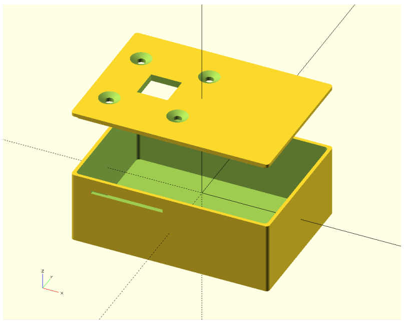
2.5 Experiments with AirSim



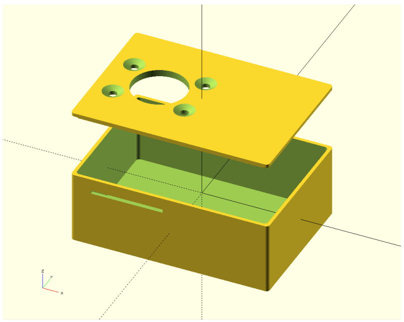
Protective Canopy.



Component mounting plate.



Google Coral camera case.

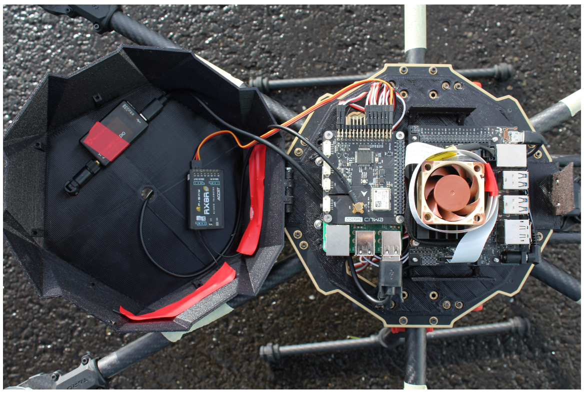


Jetson Nano camera case.

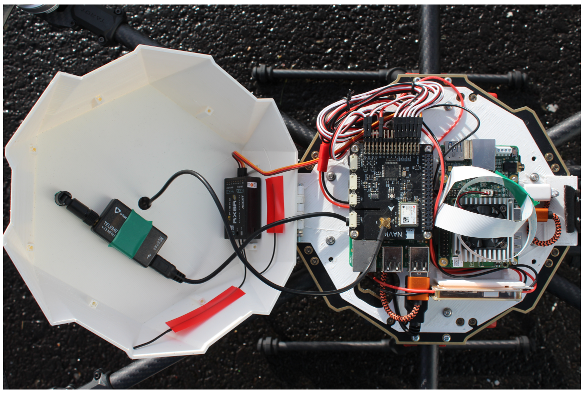
Figure 2.2: 3D printed parts for the Tarot hexacopters.

(a) The Jetson drone.

(b) The Coral drone.



(c) The Jetson drone's electronics compartment.



(d) The Coral drone's electronics compartment.

Figure 2.3: The assembled drones and their electronics compartments.

Chapter 3

Research Plan

3.1 Data Set Generation

3.2 Terrain Classifier Creation

3.3 Testing in Simulation

3.4 Testing on Physical Hardware

3.5 Risk Analysis

Bibliography

[1] Ardupilot.org.

[2] Dronecode. MAVLink Micro Air Vehicle Communication Protocol. (accessed: 2021.5.19).

[3] Emlid. Navio2. (accessed: 2020.6.5).

[4] Lorenz Meier. Pixhawk. (accessed: 2020.6.5).

[5] Joshua Springer. Autonomous Landing of a Multicopter Using Computer Vision. Master's thesis, Reykjavík University, 2020.