

CONSTRUCTION/OPERATION OF TWO HEXACOPTERS FOR AUTONOMOUS LANDING

Joshua Springer

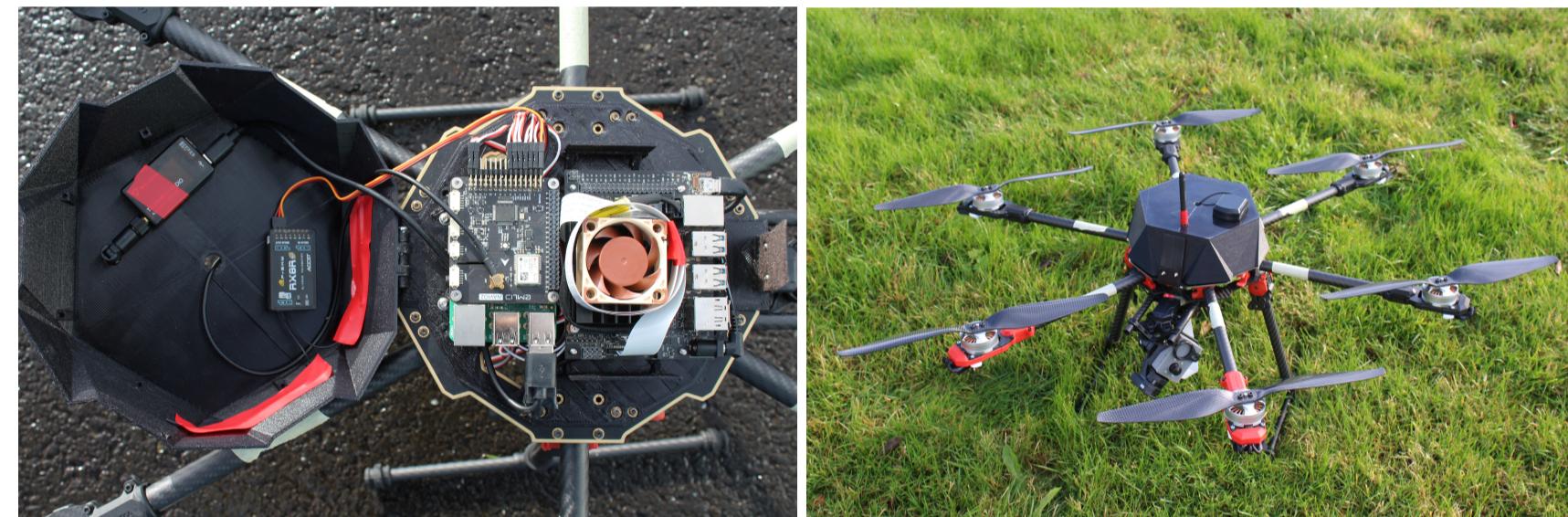


Introduction

Two Tarot 680 hexacopters allow real world testing of autonomous navigation techniques. This project tests an autonomous landing algorithm based on computer vision and fiducial markers.^[1] A landing pad is marked with fiducial markers which are recognized by the drone's onboard camera. The gimbal automatically aims the camera at the landing pad to track it during descent. The landing software then directs the drone to descend towards the landing pad.

Components

All drone electronics (e.g. motors, speed controllers, propellers, power electronics, RC Receiver, telemetry radio) are mounted on the Tarot 680 Pro hexacopter body. A Raspberry Pi 3 B+ runs ArduPilot and generates software-level control commands. It has a Navio2 hat which provides IMU data, sends low-level control signals to the speed controllers and the camera's gimbal, and receives manual control signals from the RC system. A companion board handles image analysis and estimates the pose of the drone relative to the landing pad. A radio telemetry system provides real time data to a ground control station.



The Raspberry Pi runs a version of Raspbian Linux that is adapted for the Navio 2 hat. Together these create a flight controller, which communicates with the companion board via ROS using Ethernet over USB. The companion board analyzes video from the camera to detect the landing pad, aims the camera at the detected marker(s), estimates the pose of the drone relative to the landing pad, and then sends a target position to the flight controller. In autonomous flight mode, the flight controller flies the drone towards the target position in order to safely approach the landing pad. The two drones differ in their companion boards: one has an NVIDIA Jetson Nano, and the other has a Google Coral Dev.

3D-Printed Components

A canopy from Thingiverse was 3D printed to cover the electronics. Component mounting plates were designed in OpenSCAD and provide a flat, secure surface to mount the electronics to the drone, as well as a hinge for the canopy. The camera cases also provide a way to mount the camera modules in the gimbal.

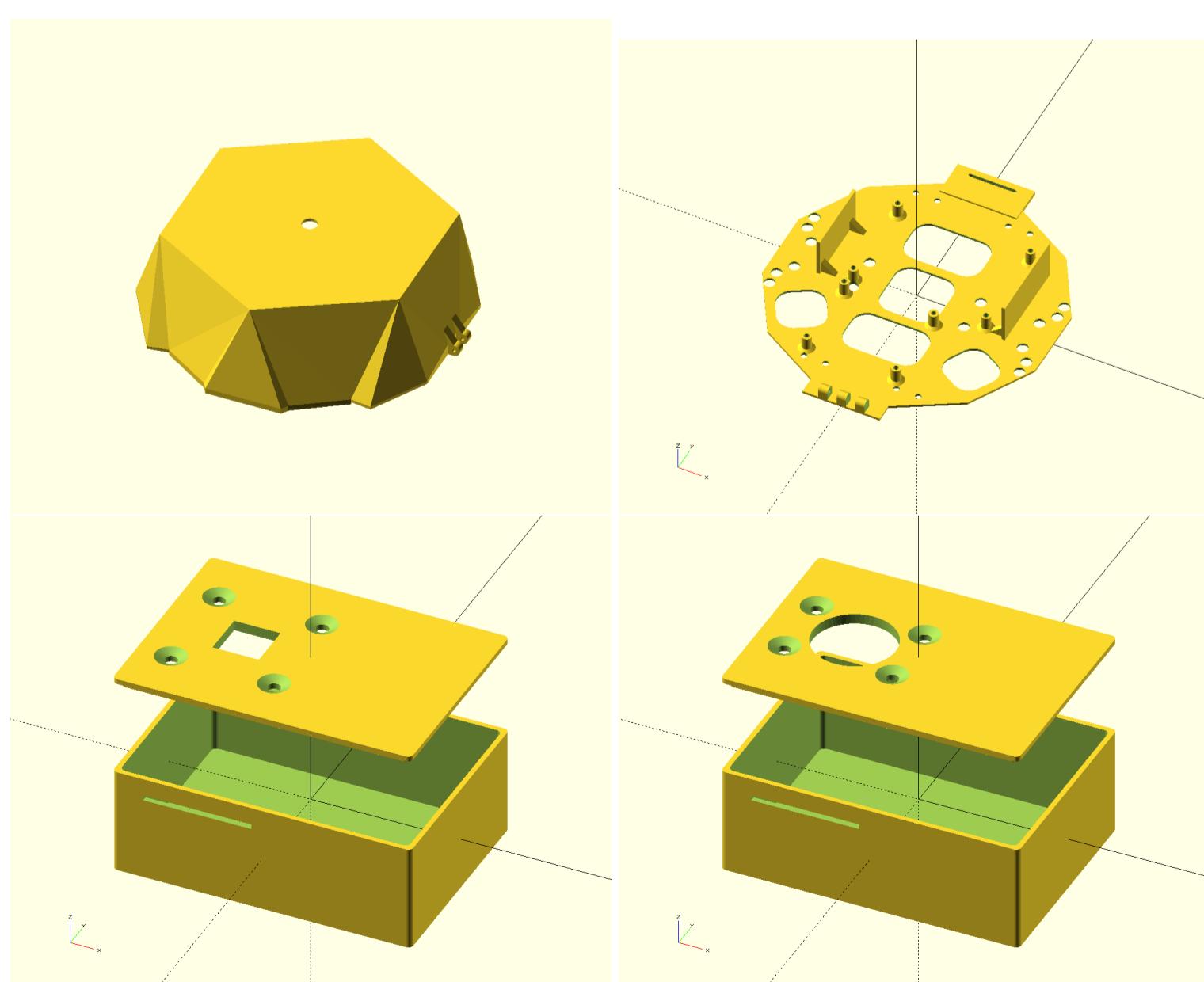
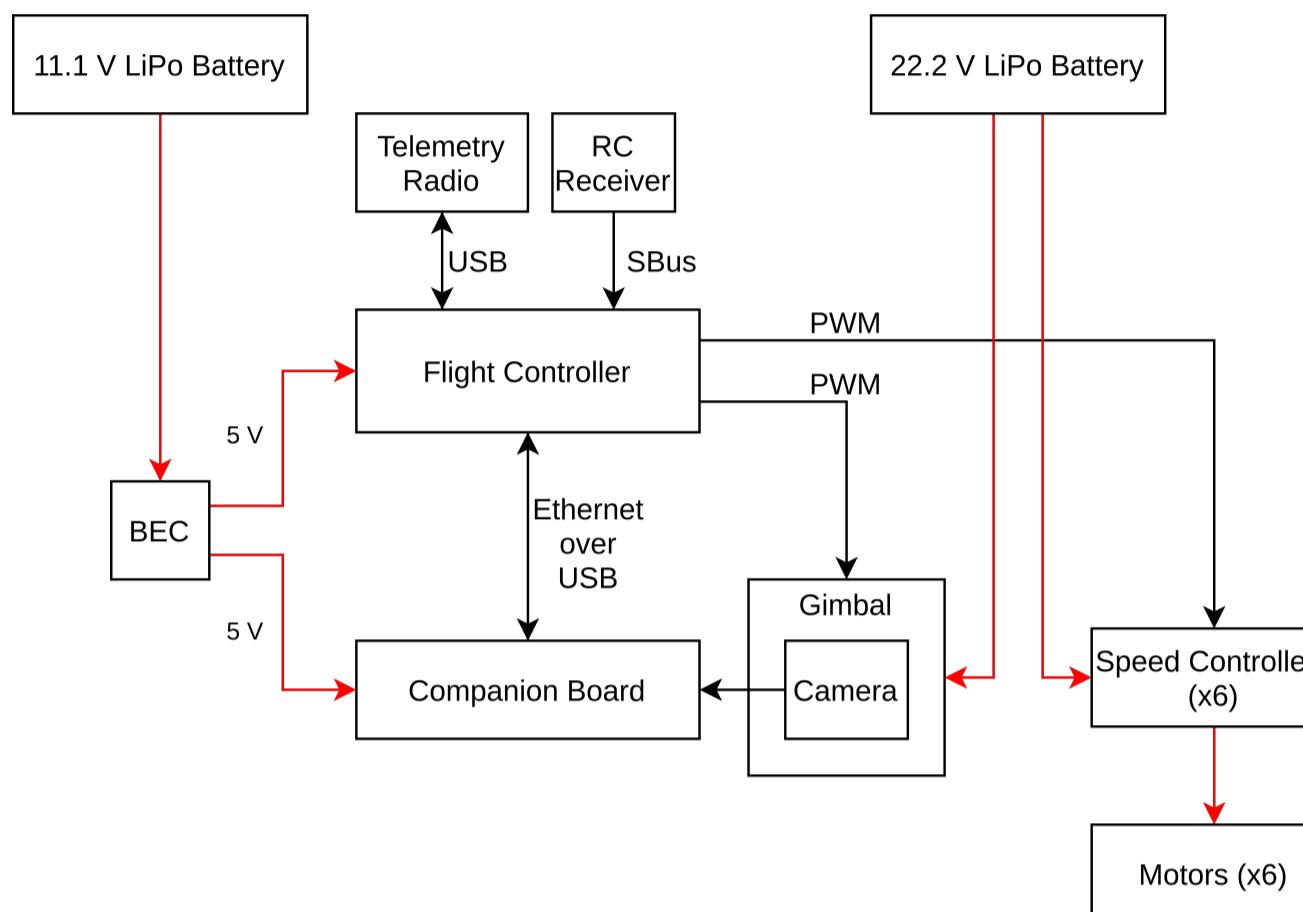


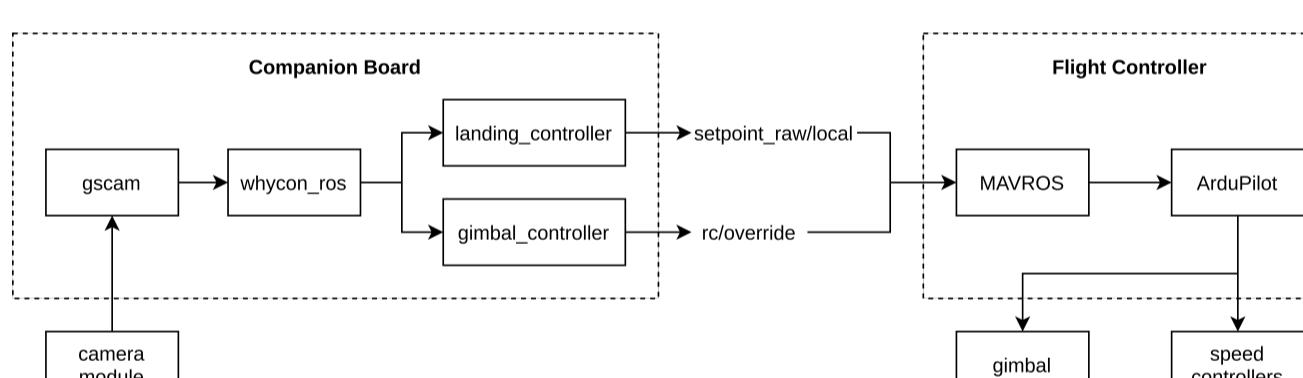
Fig. 2: Top right to bottom left - canopy, component mounting plate, Coral camera case, Jetson Nano camera case.

Power and Data Flow

The drones have two isolated power systems - an 11.1V system (regulated to 5V by the BEC) for the electronics, and a 22.2V system for the motors and gimbal. This is because the electronics require a very stable power source to function reliably, and the motors can suddenly draw large currents, reducing the apparent voltage unpredictably.



Data flow for the autonomous landing task begins at the camera module, from which high-resolution video is fed into the companion board. A program `gscam` resizes the video to a lower resolution to reduce computational requirements, and pipes the resized video to the `whycon_ros` module for analysis. The `landing_controller` module performs coordinate system transforms to generate a target position for the drone based on the detected position of the landing pad. The `gimbal_controller` module sends control signals (through the flight controller via the `rc override` ROS topic) to the gimbal in order to center the landing pad in the camera's field of view. When no landing pad is detected, the gimbal is controlled manually by the operator. The MAVROS module receives information from the landing and gimbal controllers, translates it into lower-level commands, and passes those commands to the ArduPilot software. ArduPilot sends control signals to the gimbal and speed controllers to make the drone approach and track the landing pad.



Landing Pad Design and Detection Performance

Fiducial markers (top: April Tag^[3], bottom: WhyCon^[2]) allow the drone to determine its position relative to the landing pad using only a normal RGB camera. The larger WhyCon marker is easier for the system to recognize at long range, while the April Tag marker is positioned in front of the WhyCon marker in order to keep it in the field of view of the camera at close range. The drone touches down in the center of the WhyCon marker. After initial testing, the April Tag system required substantial computational resources on the companion board and was therefore removed. The final landing pad design includes only a WhyCon marker. Fig. 6 shows the performance of the gimbal controller in aiming the camera at the landing pad in a 640x480 frame. These graphs show that the camera is successfully aimed at the landing pad (with some typical PID oscillation) in most cases. Cases where the landing pad is not fully centered are those where the gimbal reaches its rotational limits. The oscillation is a result of high PID gains (for quick correction) but does not adversely affect the detection performance.

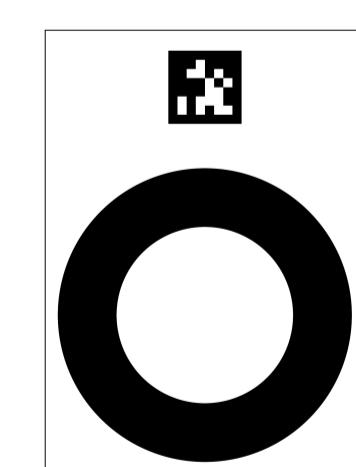
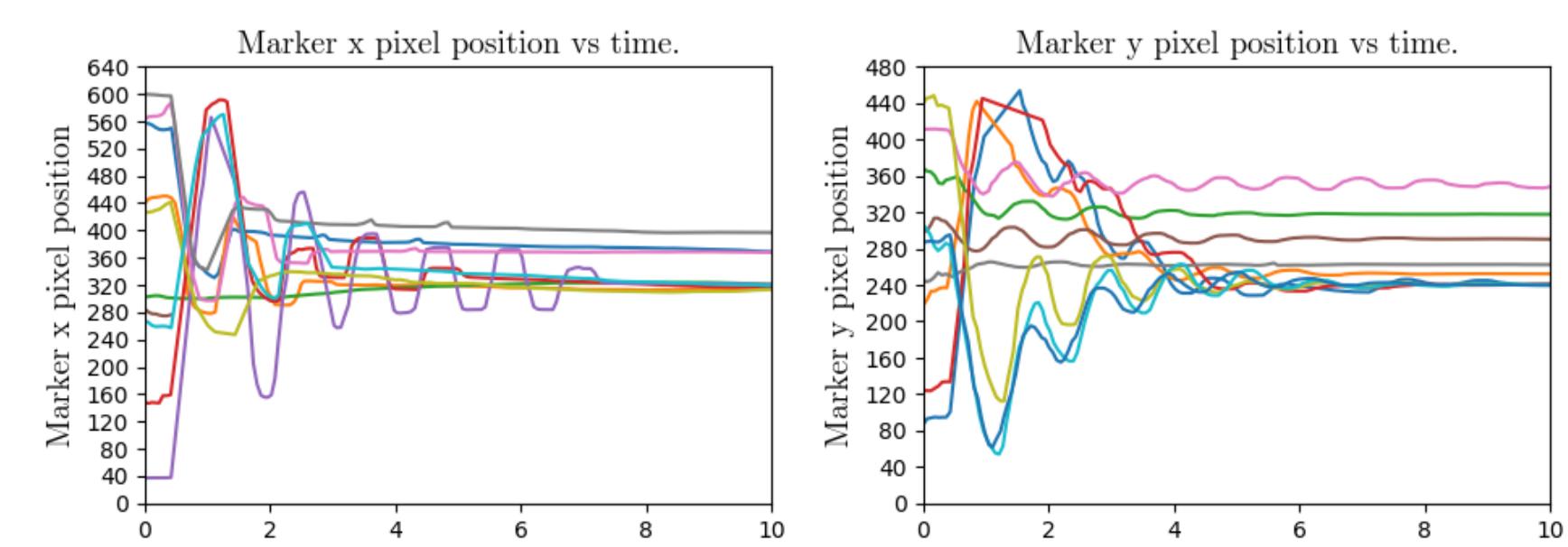


Fig. 5: Initial Landing Pad Design



Flight and Landing Performance

Both drones have stable and reliable performance even in high wind, with safe flight times of roughly 15 minutes. They have enough power to lift additional sensors/electronics. The cameras are able to aim at the landing pad reliably in a laboratory setting, however the Jetson Nano had power issues in field testing. The Google Coral drone was able to reliably aim the camera, generate position targets and approach the landing pad autonomously. Because of an issue with the WhyCon marker system, the pose estimation was inaccurate at close range and therefore landing was aborted at the last minute. However, this issue is not major and will be addressed in future work. Fig. 7 visualizes the pitch, roll, and altitude performance of the drone during a hover of about 10 seconds. The pitch and roll vary by about 2.5 degrees (as normal) because of hardware performance and environmental conditions. The altitude varies by about 2 meters as a result of manual control. Fig. 8 shows the drone in flight.

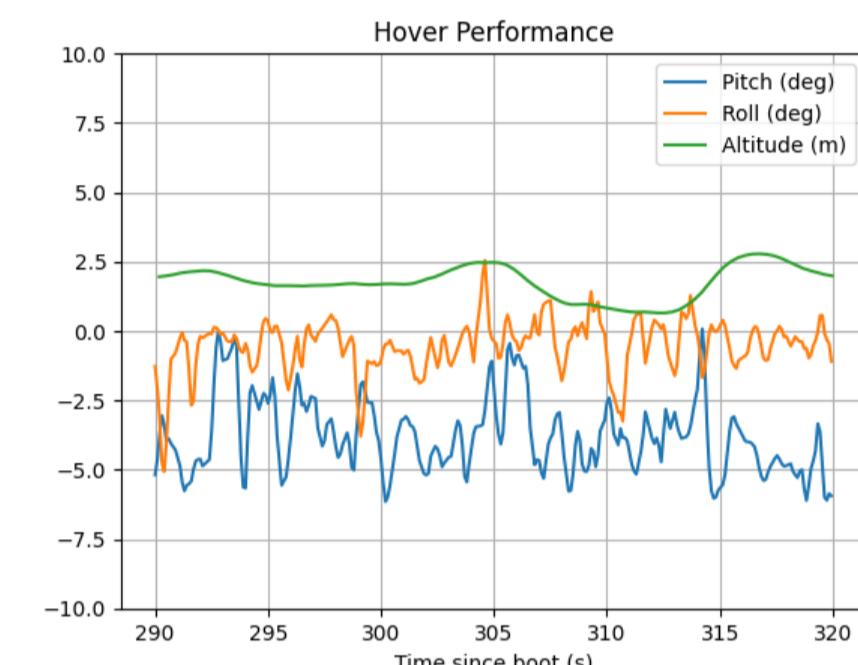


Fig. 7: Pitch, roll, and altitude during hover.



Fig. 8: Drone in Flight (Scan QR code in top right for video!)

Future Work

The pose estimation system will be changed to fix its issues that occur when the drone is at close range to the landing pad. One possible solution is to attempt to fix the WhyCon system itself. Another solution is to use existing deep learning models (ex.- EfficientPose or PoseCNN) to create a new pose estimation system, as the companion boards are designed for embedded deep learning tasks.

The drones will be tested in different autonomous scenarios, such as more complex point-to-point missions. They will serve as flexible platforms for more specific tasks such as terrain mapping, object tracking, etc.

References

- [1] Joshua Springer. "Autonomous Landing of a Multicopter Using Computer Vision". MA thesis. Reykjavik University, Department of Computer Science, 2020.
- [2] Matias Nitsche et al. "WhyCon: An Efficient, Marker-based Localization System". In: *IROS Workshop on Open Source Aerial Robotics*. 2015.
- [3] E. Olson. "AprilTag: A robust and flexible visual fiducial system". In: *2011 IEEE International Conference on Robotics and Automation*. 2011, pp. 3400–3407.