Autonomous Quadcopter Docking System

Low-cost implementation of small UAV platforms using consumer electronics

Abstract

The goal of this project was to design the systems and algorithms necessary to allow a quadcopter to autonomous locate and land on a ground station. The purpose of this system was to outline the framework for a quadcopter based data collection or surveillance system that copes with the relatively short battery life of these highly mobile devices by consistently landing the UAV safely in a designated location to be recharged.

The 3D Robotics ArduCopter was chosen as the quadcopter platform since it is durable, capable of autonomously hovering in place, and is capable of carrying a payload, such as the camera used to determine the location of the dock. A system was then devised such that the quadcopter can correctly determine the location of a target ground station while hovering and then land when on the target. Only commercially available components and free software was used so that the entire docking system is easily accessible to future researchers and UAV enthusiasts.

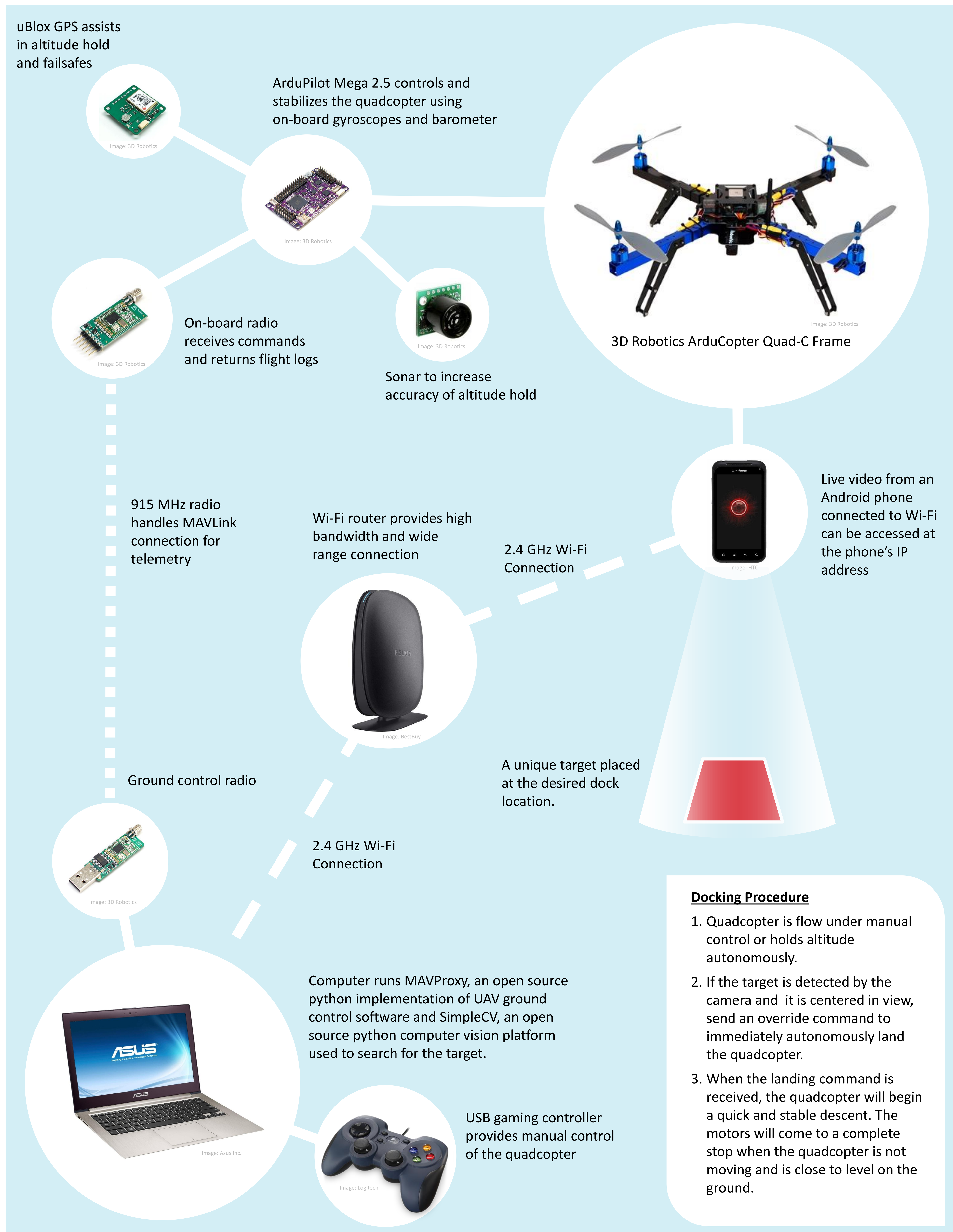
Implementation

Left: Assembled Quadcopter with propeller protectors. Center: the target. Left: A thresholded image detecting the largest red object.

The quadcopter is controlled via MAVProxy, an open source command line ground station over, using MAVLink, on open source micro aerial vehicle marshaling and commination library written in python. MAVProxy can initiate a variety of autonomous behaviors, such as altitude hold or position hold. Using a customized add on module, the quadcopter can flown manually using a gaming controller.

To locate and land on designated dock, a new module of MAVProxy was implemented to receive the video stream from an Android smart phone over Wi-Fi and search for a target – a 22cmx22cm red square. Using the SimpleCV python library, the video stream is thresholded by distance from the target color. The image is then searched for connected components. If the largest connected component exceeds a threshold, it is determined to target. When the target is within the center potion of the image, an override command initiates an autonomous landing over MAVLink by altering the MAVProxy state. The thresholded video stream is displayed to a show the quadcopters view of the target. This entire module runs in a separate thread from the MAVProxy/MAVLink connection to decrease latency.

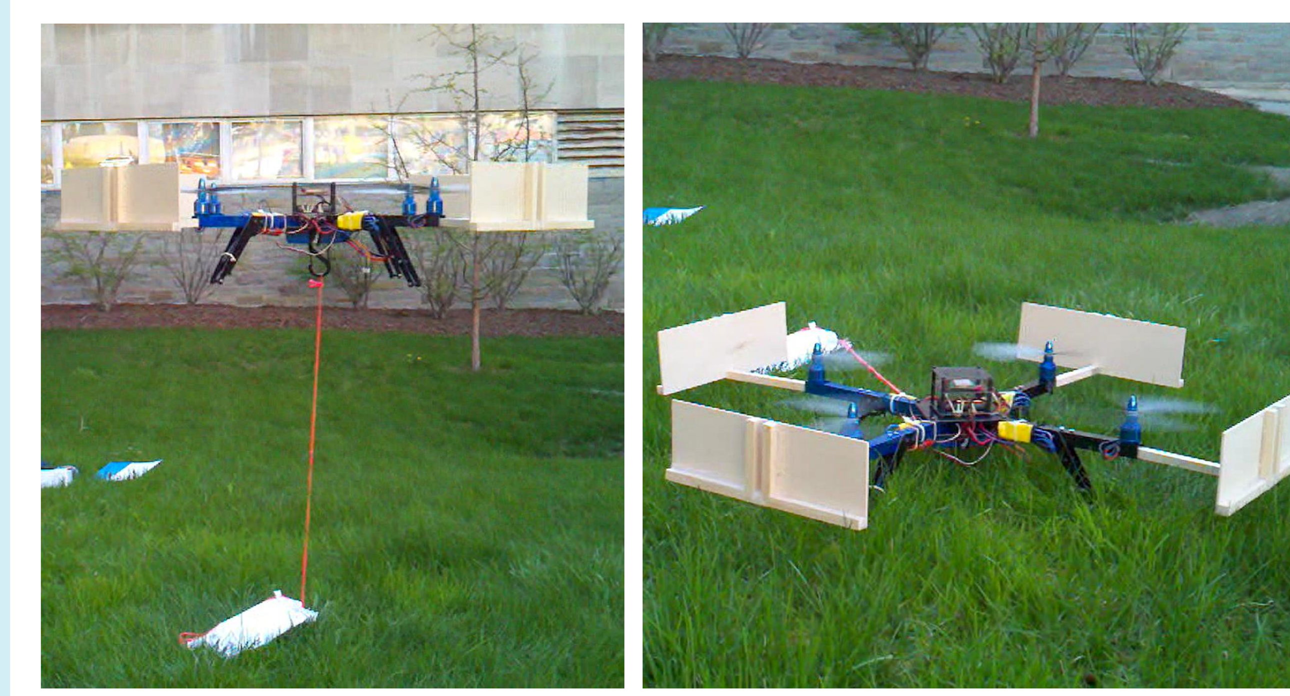
Final Design



Results

Using this system, the quadcopter has successfully landed using visual targets. The quadcopter is able to land from hovering 60 cm in the air in about 0.6 seconds. The type of target used can greatly increase the time required to recognize it, thus large red squares proved to be unique object in both indoor and outdoor locations. But, as with any computer vision task, the lighting conditions can significantly change the appearance of the targets. While the Android smart phone's camera is able actively adjust, it does so much more slowly than the control loop's execution, causing the quadcopter to sometimes miss the target. The camera showed best performance indoors when indirect natural light or in complete darkness using the built in back LEDS for illumination.

One of the greatest limiting factors of the system is the update rate of the video as the video stream accumulates lag as it continues to film. This can be mitigated by having a local dedicated Wi-Fi router and by removing applications running in the background of the Android phone.



Left: Quadcopter hovering on tether outdoors. Right: Quadcopter landing after seeing the target.

While much autonomous robotics research is conducted using complex and prohibitively expensive motion capture systems, this docking system provides proof that using off the shelf parts, an old smart phone, and free open source software can accomplished complex computer vision and autonomous vehicle control tasks that student, researches and hobbyist can benefit from.

Acknowledgements

I would like to thank my advisor Bruce Land for his advice, encouragement, and continued support of this project.

Contact

Sima Mitra
Cornell University School of Electrical and Computer Engineering
https://www.simamitra.com