Low Cost Drone for Autonomous GPS Denied Navigation, Awareness, and Interaction

John Marcus
Embry-Riddle Aeronautical university B.S. Aerospace Engineering Student

ABSTRACT

The Robotics Association of Embry-Riddle has continued development on a low cost, GPS denied multirotor system. This system is capable of interacting with ground robots and avoiding dynamic obstacles. GPS denied systems are complicated due to the complexity of navigation without fixed reference points. Autonomy is achieved using two computers, a Pixracer with PX4 firmware, and an Odroid XU4, which communicate and mesh information from a combination of instruments, mainly an optical flow sensor, USB camera, and LIDAR. The optical flow camera is used in a feedback loop in order for the system to get velocity feedback, which is then meshed with other information on the onboard computers to achieve stable flight. The system then uses the USB camera to identify and interact with its objectives. The total cost for autonomy for this system is less than $600, this only includes the price of computers, sensors and wiring.

INTRODUCTION

The Robotics Association at Embry-Riddle (RAER) has continued work on the International Aerial Robotics Competition mission seven since its commencement years ago. Every year, they have built on the previous year, seeking to build a low cost, completely autonomous system. The goal of this mission is to “move the state-of-the-art in aerial robotics forward” [1], and because of the lack of GPS denied systems available, this is where this competition concentrates.

Statement of the Problem

Autonomous aircraft in a GPS denied environment are not commonly used in commercial, academic and military applications due to the difficulty of developing a reliable navigation system in such an environment. The lack of reference point for the location of aircraft creates a situation where a system must be aware of its location as well as environment without an absolute reference. The sensors and programming required to do this may vary greatly from environment to environment, and so with the International Aerial Robotics Competition (IARC) environment in mind, a system must be developed for navigation and completing the task at hand.

This specific challenge has ten ground robots moving randomly in a twenty meter by twenty-meter environment. The goal is to redirect all ground robots to one side of the arena via a top mounted switch, which when triggered turns the ground robots 45°. Several other obstacles stand in the way of completing this task. There are no defined walls for this competition, which could otherwise be used for localization. There are four ground robots that act as obstacles to the aerial platform. They have PVC tubes on top ranging up to two meters tall, and they move about randomly. Additionally, a system should also be able to fly simultaneously alongside another aerial platform.

Conceptual Solution

This is a challenge that must be solved using an aerial vehicle. It must be an agile, yet stable platform that can sense, avoid, interact with ground robots, and be fully autonomous. For this the system selected was a multi-rotor.

The figure above depicts the main concept behind solving this solution. The sensors carried onboard needs to be able to track the objective: the ground robots. Their position must be known relative to the system so that they system can move to and interact with them. The system must also have sensors capable of detecting any obstacles that move into its flightpath with sufficient time to be able to avoid them. Altitude, and orientation must also be known so that the system can correct its flight trajectories to stay within the bounds of the competition. Mechanically, the system must be capable of interacting with the ground robots without

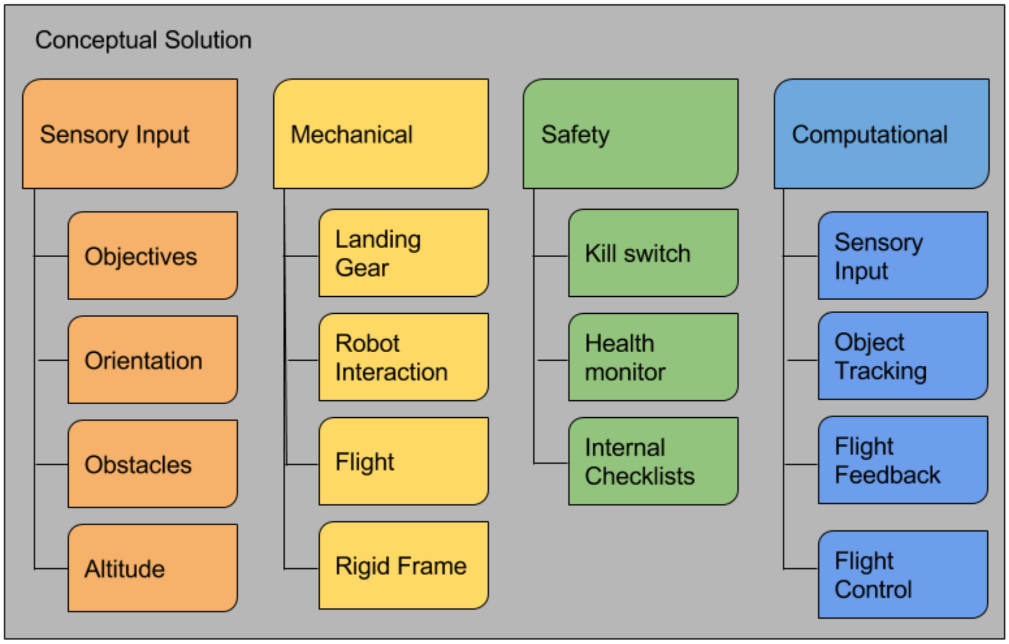

Figure 1. Conceptual solution overview

jeopardizing itself. The frame must be rigid and able to withstand potentially uneven or hard landings. For safety, there will always need to be an emergency kill switch, and health monitor for the system. Battery voltage, ability to fly, computational ability, and structure must all be monitored in case of failure, so the system can tell if it is safe to fly. Finally, the onboard computers must be able to process the sensory input, track objects even after they are detected and predict their location, and receive corrective feedback from the sensors to improve its own flight characteristics.

Yearly Milestones

Due to the uncertain nature of the environment, as well as the difficulty that comes with navigation, the team's goal is to solve the simplest problem and build an increasingly complex system; starting from a simple autonomous control and eventually leading to completion, efficiency, adaptability, and eventually cost effectiveness.

Last year, the main goal was to build different necessary parts that would later be integrated. This included programing reliable vision processing, manual flight, and practicing necessary technical skills. This year, the team shifted focus on to basic autonomy and system integration. The autonomous system could detect ground robots during flight, as well as autonomously take off and perform basic maneuvers. Next year, this platform will continue integrating sensor data with flight controls to achieve better flight performance as well as the ability to redirect ground robots and improving flight characteristics.

AIR VEHICLE

This competition demands an agile, stable, and low endurance aircraft. Fixed wing aircraft or lighter than air aircraft are not capable of the maneuvers required, such as landing on individual robots or remaining within a twenty by twenty-meter area. In addition to this, with a time of only ten minutes, the endurance required is relatively low on the scale of most aircraft. For these reasons, rotor aircraft is best suited for this type of environment and competition. The team has the most experience with multi-rotor aircraft, and so this is the basis for the mechanical system.

As can be seen in Figure 2, the main sensor package for identifying the ground robots would be a USB camera, and this could also be used for localization utilizing the grid that is placed on the ground of the competition. For detecting obstacles, a LIDAR would be used, and for velocity feedback, in place of a GPS, an optical flow sensor would be used. On the Odroid XU4, the microcomputer used for programming the multi-rotor, software processes sensors and mission information to output velocity vectors to the multi-rotor's autopilot, the Pixracer. The multi-rotor would take off, and fly above the possible obstacles, until it detects a ground robot, when it will redirect and follow the ground robot until it is successfully brought to the terminating edge of the competition.

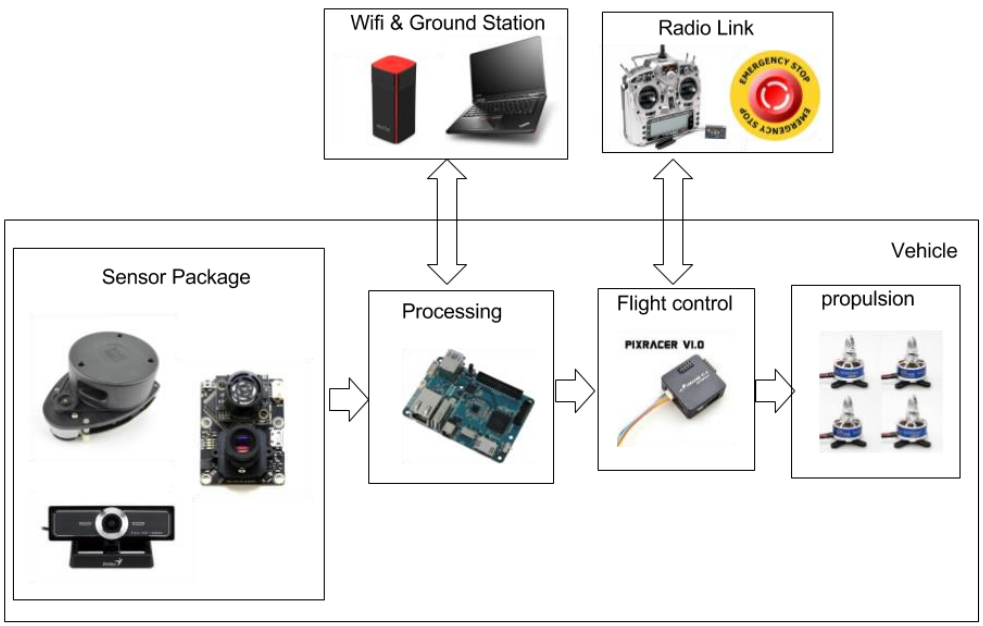

Figure 2. System Architecture

Propulsion and Lift System

Our system makes use of many commercially available multi-rotor parts, such as its Tarot FY-650 frame. This is not to say this system has been neglected, as each component was carefully selected to ensure flight performance is tuned to the time and weight requirements of the system. For propulsion, there are four Dynam Tomcat M35083 motors, each with a 12”x 4.5” APC propellers. Every motor is powered by a 30 Amp ESC, for a total of 120 Amps. The whole system is powered using 4 cell Li-Po batteries, which range from 14.5V to 16.5V.

Flight Performance

The aircraft is capable of a 9 minute 30 second flight with a gross weight of 6.6 pounds. This is the result of a manual test flight performed while also running all computers on board. During this test, the system did not perform maneuvers, but maintained position, so practical endurance may be shorter.

Navigation and Control

To fly indoors autonomously, a system of navigation must be developed, which is done using a variety of sensors and computers. This system has two main computers and three main sensors. The computers used for control and navigation include the Pix Racer and Odroid XU4, while the sensors are the USB camera and the optical flow sensor.

Computers

All the programming that this team performs needs to be able to ultimately control the motion of the aircraft. The two computers that are responsible for this are the Pixracer and Odroid XU4. The Pixracer is a small autopilot computer used for many different types of drones. This autopilot comes with several sensors, including a telemetry module, accelerator, gyro, magnetometer and barometer. While the magnetometer and barometer cannot be relied upon, due to the indoor usage of this system, the other instruments can be used. The telemetry modules and accelerometers are used as a stability-augmenting system during flight. The Odroid is a microcomputer with a 2 GHz Samsung Cortex processor.

At the time of purchase, in 2015, this was one of the most powerful single board computers. The Odroid is nearly three times as powerful as the more popular Raspberry pi 3, according to benchmarks performed by CNXSoftware in 2016, and still outperformed every other single board computer tested on six out of seven benchmarks [2]. However, there are more powerful computers now, such as the NVidia Jetson. However, the Odroid XU4 has proven more than capable of running all of the software needed for this competition.

Software

To create the autonomous system, the Pixracer communicates with the Odroid. The Pixracer has the PX4 firmware installed, which is designed to run in conjunction with an off-board computer natively. The Odroid runs the Robotic Operating System, ROS, to create a communication network. A module for ROS, Mavros is specifically designed for communicating with the PX4 Firmware. This allows the system to send velocity vectors to the Pixracer autopilot, which in turn controls the speed of each motor, the architecture for this can be seen in Figure 3.

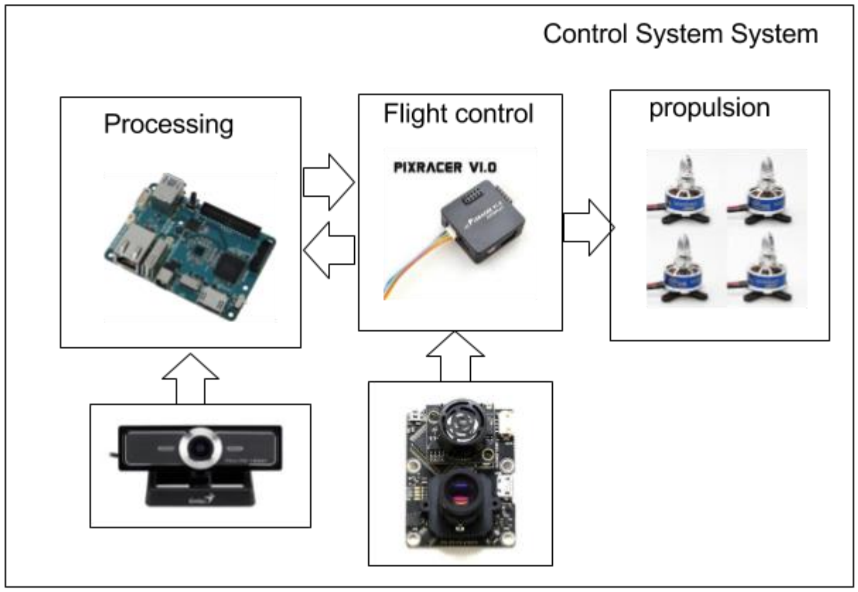

Figure 3. Control System Architecture

Feedback is important for controlling the system. To achieve this the system makes use of an optical flow sensor, specifically the PX4 Flow smart camera sensor. This is a high frame rate, low resolution camera. In addition to this, there is a sonar sensor mounted on the board for determining the altitude of the aircraft. An on-board Cortex processor uses these sensors to determine the ground velocity of the aircraft. This information is then sent to the velocity control software on the Odroid. A feedback loop is then run in conjunction with the desired velocity to ensure proper flight velocity.

In addition to this, The USB camera can be used to track the grid on the ground of the area for a less fine but more absolute position tracking. This method would have a resolution of one meter; the resolution of the grid, however would allow for absolute positioning and edge detection. This works well with this competition, as the grid on the ground is continuous and constant, and therefore can be reliably tracked. Navigation is achieved using a combination of velocity control as well as ground tracking. The system will keep track of its velocity over time, and this will allow the aircraft know where it's positioned.

Flight Termination System

To safely operate an unmanned system, there should always be a manual kill switch for the motors of the system. This kill switch is remotely operated and immediately cuts the power to all systems. Per the competition requirements, there will be a separate switch that is able to kill the system. For this purpose, the hosts of the IARC competition have created a common kill switch schematic, which has been adapted for this use. The common kill switch was designed for a 13.5 Volt, 35 Amp circuit [3]. This design was modified to fit the systems voltage and amperage draw.

PAYLOAD

The current system is designed to fly with a five-pound gross weight for a minimum of ten minutes. While the system is currently heavier than five pounds gross, the endurance is hardly affected. The payload consists of three main sensors: USB camera mounted on a gimbal, Flow sensor, and RP LIDAR. This sensor package costs a total of $390 and is entirely commercially available. The most expensive component is the RP LIDAR, used for threat identification.

Target Identification

To identify ground targets, a USB camera is used in conjunction with the Open Computer Vision, Python library. The camera is mounted on a gimbal to ensure proper orientation during flight. This is necessary because tracking ground robots while maneuvering above them changes the orientation of the platform. Figure 4 shows a frame taken from the vision processing software and displays a before and after image. The after image shows an identified target, as well as a line, for use with tracking the grid of the competition. The ground robots are identified using a combination of circle detection and filters, including one based off of altitude, which was found experimentally using MATLAB.

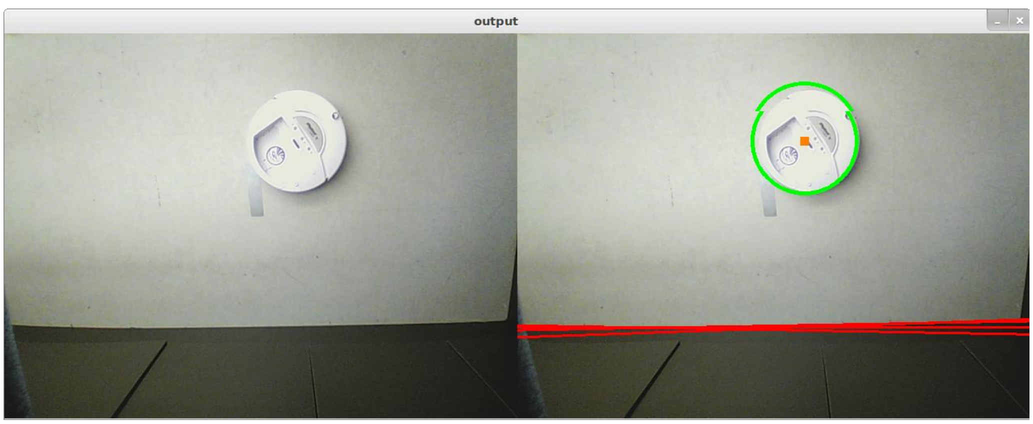

Figure 4: Example frame from vision processing. This shows proper ground robot detection as well as edge detection.

Threat Identification and Avoidance

The RP LIDAR devolved by Slamtec is this systems method of threat identification. This LIDAR has a view of 360 degrees and angle increments of less than 1 degree. This LIDAR can scan its view at 5.5 Hz, and has a maximum range of 6 meters [4]. This sensor can detect the moving ground robot obstacles during the competition. Per the rules of the competition, they move at a velocity of 1/3 meter per second, and so with a range of 6 meters, the system will be aware of any obstacles nearly 5 seconds prior to collision, assuming the system is moving towards the obstacles, a worst case. When in flight with another system, some rules of flight must be established, as it is possible that the other system exceeds the capabilities of this object detection.

The threat avoidance system comes directly from the identification system. All threats within scanning radius will be avoided by flying perpendicular to their trajectory or above them, at an altitude between 2 meters and 3 meters, which is the maximum height of the obstacles and the maximum allowed altitude respectively.

Communications

To communicate with the ground system, a Wi-Fi antenna is fitted to the Odroid. The ground station connects with an SSH through port 22 to launch the necessary software, once the system launches, it is completely independent due to all computation and software running on board. The manual flight radio can however resume control of the flight through the Pixracer at any time should autonomy fail.

Power Management System

The system must be powered only electrically and able to fly for the duration of the competition, ten minutes. In addition to this, various components of the system have dramatically different power requirements, so a form of power distribution must be present. The system's energy is stored by four cell Li-Po batteries. Due to the battery mounts on either side of the multi-rotor, the aircraft have a capacity of ten amp hours' of battery life. However the team usually opt for the eight necessary to reach ten minutes. The power from the batteries is run through two different commercially available BEC's that power the Odroid and Pixracer before reaching the Electronic Speed Controls (ESCs), that power the motors.

OPERATIONS

A standard procedure has been established for the usage of the system, as well as a preflight check list.

Preflight Checklist

Autonomous flight can be dangerous if proper precautions are not taken. As a result, an established preflight check list is built into the software that includes both system and human factors such as sensor calibration or proper safety procedure. The preflight checklist is included below. Some of these checks are questions, and this is due to the preflight check both prevent the software from running and recording answers as they are entered. This allows the preflight information to be checked in the event of a failure.

Preflight Check list:

1: Area check … is it safe to fly?
2: Safety Glasses on?
3: Is Vision properly running?
4: Wiring check … I2C
5: Wiring check … battery
6: Wiring check … USB
7: What is preflight Battery Voltage?

8: Ensure motor mount rigidity
9: What is the Flow Sensor XY Magnitude?
10: Is PID properly running?

User Interface

As previously mentioned in the communications section above, the ground station user interacts with the system via a remote terminal. While this does not have any graphics, and can be clunky to use at first, it is simple to develop and provides the user with operational freedom, and full control and view of the programs immediately prior to flight. In addition to this, large amounts of information can be sent to the ground station during flight without the worry about bandwidth due to the low demand of terminal scripts.

RISK REDUCTION

Flying a quadrotor requires a certain amount of hazard assessment and avoidance. The platform itself must be constructed and designed with these considerations.

Vibration Isolation

In flight, multirotor aircraft typically experience a lot of vibration, which must be damped to ensure the system does not shake apart during flight or with prolonged usage. For this, on the underside of the multirotor there are two vibration damping rails on which all the sensors are mounted. The optical flow and RP LIDAR are mounted together with 3D printed material. On the opposite side, the USB camera is mounted on a gimbal, which has an additional set of vibration damping rubber connectors.

Shock Absorption

Quadrotors are notorious for having bad crash performance, and rightfully so. Without their motors, they have no active system to perform a soft landing. For this, the system has a several mechanisms designed to protect expensive sensors, computers, and batteries in the case of a failure or freefall.

Battery Protection

Lithium Polymer batteries are volatile when ruptured, and so the battery mounts serve to protect them. The battery mounts used on the system were designed by the team and 3D printed. The mount is shaped as an open top rectangle where the batteries are attached to the inside of the channel. They mount between two of the motor arms. The outside and bottom of these mounts serve as a protective layer to the batteries.

Landing Gear

The landing gear for the system has three protections from a freefall landing. The gear is protected with elastic foam. This helps nondestructively absorb energy and keep the system upright if there is an uneven landing. Inside the foam, there is a carbon fiber shell with a wooden core. Because the carbon fiber has a higher yield strength then the wood, in the event of an asymmetric landing, the wood will snap, absorbing energy. This is a design choice, because wood is relatively cheap and readily available. Should this not absorb enough energy, the legs can also pivot by snapping out of position. These can be snapped back together relatively quickly.

Testing

Testing of the platform is performed with several safety checks in place. A qualified safety pilot must always be on the manual flight controller should autonomy not perform as expected. In addition to this, the area must be clear and safe for testing. Observers to the test must have a barrier preventing the system from potentially hitting them. For testing the platform, due to limited space and time requirements, Embry-Riddle has one ground robot to perform tests with.

CONCLUTION

To build an autonomous system to work in a GPS denied environment, this platform serves as basic solution, capable of basic navigation and interaction with ground robots. The cost of recreating the autonomous system is less than $600, as can be seen in Figure 4, a price which does not include the mechanical frame, propulsion, or batteries. Navigation within a GPS denied environment for low cost is feasible, and can be applied to several applications, such as herding ground robots. In coming years, this system will continue to improve efficiency and capabilities.

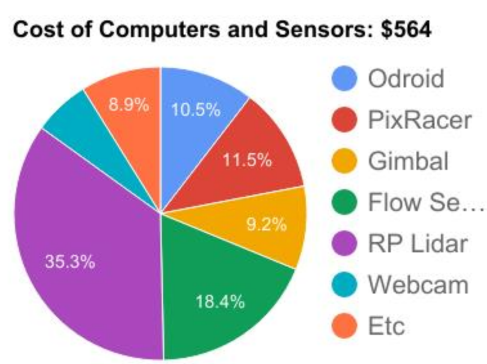

Figure 4: the cost for this system is primarily in the sensors

REFRENCES

[1] Michelson, R., Official Rules for the International Aerial Robotics Competition, Mission 7, http://www.aerialroboticscompetition.org/downloads/mission7rules_013017.pdf

[2] CNXSoftware, Low Cost Development Boards Linux Benchmarks – Raspberry Pi vs Banana Pi vs Orange Pi vs ODROID. http://www.cnx-software.com/2016/04/02/low-cost-development-boards-linux-benchmarks-raspberry-pi-vs-banana-pi-vs-orange-pi-vs-odroid/

[3] Kahn, A., IARC Common Kill Switch,
http://www.aerialroboticscompetition.org/resources.php

[4] RPLIDAR A1Low Cost 360 Degree Laser Range Scanner Introduction and Datasheet, Slamtec,
http://bucket.download.slamtec.com/e9e096e9d9f30205d665260abe2cfb0c2dd62efa/LD108_
SLAMTEC_rplidar_datasheet_A1M8_v1.0_en.pdf