Design and Implementation of an Autonomous Aerial Vehicle for Information Gathering in a Simulated Autonomous Environment

Nathanael B. Edwards, Cynthia H.T. Edwards, Bradley J. Nelson, Joseph B. Tomlinson
Oregon State University College of Engineering
Aaron O. Moore
Revolution Robotics, Inc.

ABSTRACT

An autonomous quad-rotor aerial robot was developed for the purpose of indoor surveillance where human intervention may not be possible. The robot is comprised of four outrunner motors mounted directly to two pairs of counter rotating propellers, a high-resolution IMU (inertial measurement unit), an ARM processor running at 600 MHz with 256 MB RAM, a carbon fiber chassis, and a lithium polymer power source. The robot is capable of completely autonomous operation, with missions that can be changed and redefined by simple reconfiguration. The vehicle is sized so that it can navigate most human navigable construction, and can assist in catastrophe recovery and investigation for a wide range of disasters, both natural and man made. This vehicle is designed for entry in the 2009 International Aerial Robotics Competition.

1. INTRODUCTION

1.1 Problem Statement

Hazardous environments resulting from industrial accidents can be inaccessible to humans and prevent information gathering for long periods of time. If more information could be gathered remotely, disaster management could be improved. Using unmanned robotic systems capable of autonomous search in unknown environments is one possible solution to this problem.

This paper describes the design and implementation of an unmanned aerial vehicle capable of autonomous information gathering and navigation without GPS in a simulated nuclear power plant that has gone out of control. Specific requirements considered in design included: the ability to launch from a "mother ship" outside the building, find a 1 meter square window marked by a symbol to be used for entering the building, navigate indoors, build a map of the robot's location in relation to its launch point, find a "control panel" marked by a solid blue LED, and relay video and images back to a base station some distance from the building.

1.2 Conceptual Approach

Students in Oregon State University Robotics Club (OSURC) developed the autonomous aerial vehicle design described herein. The conceptual approach of this team of students was to meet design requirements in the simplest way possible, while leaving room for complex improvements in the future. A quad-rotor vehicle was designed for stability utilizing carbon fiber for weight reduction. Navigation and autonomous search for information are currently accomplished using visual search algorithms, a number of ultrasonic sensors, data from an Inertial Measurement Unit (IMU), statistical techniques, and a node-based recursive path-finding algorithm. Data transmission uses standard 802.11n WLAN and Joint Architecture for Unmanned Systems (JAUS) communication protocols. These relatively simple modules work together to form a complex system capable of meeting the design requirements. Figure 1 in the Appendix shows the system architecture.

1.3 Yearly Project Milestones

The initial goal of the project is to fly to the building, find the window, and navigate through the window into the building. Once this goal is complete and the robot is inside the building, the robot will search for a detectable sound, and a solid blue LED on the control panel of interest. If the correct control panel is found, a data transmission will be made back to the "mother ship." Ideally, these events should occur within a ten-minute period.

The current design implementation is capable of accomplishing these tasks, but may require debugging and optimization in the future. The design includes upgrading from ultrasonic sensor-based navigation to a more reliable LiDAR (Light Detection And Ranging) system combined with a FastSLAM (Fast Simultaneous Localization and Mapping) algorithm for mapping [2]. Implementation of a probabilistic approach to search based on secondary sensor information, such as sound, may help optimize for speed by increasing the ability of the autonomous system to prioritize effectively. Software is continuously updated to streamline for memory usage and effective use of shared resources. Finally, future plans include decreasing power usage by hardware in order to increase battery life and implementation of additional failsafe mechanisms to increase the robustness of the system.

The lift system for our quadrotor consists of four Hacker A20-22l three phase brushless DC motors attached to 10 inch counter-rotating propellers. Three phase brushless motor controllers using an I2C interface allow for a refresh rate of up to 500 Hz. The motors are powered by a 12V 5AH lithium polymer battery, which can provide approximately 20 minutes of flight time.

2. AIR VEHICLE

2.1 Propulsion and Lift System

A quad rotor design was chosen because of the mechanical simplicity of the design, the high maneuverability of platform and the wealth of existing design resources available. Fixed wing aircraft was ruled out early because of low maneuverability. Tri-rotor and helicopter designs where considered but were ruled out due to complexity of design, robustness, maneuverability, and vehicle size requirements.

The lift system for our quadrotor consists of four Hacker A20-22l three phase brushless DC motors attached to 10 inch counter-rotating propellers. Three phase brushless motor controllers using an I2C interface allow for a refresh rate of up to 500 Hz. The motors are powered by a 12V 5AH lithium polymer battery, which can provide approximately 20 minutes of flight time.

2.2 Guidance, Navigation, and Control

The platform uses a Microstrain inertial measurement unit (IMU) for attitude control. Low level feedback control runs on our Gumstix processor which has direct control over motor speeds. The system includes cameras and a LiDAR sensor which can be used with computer vision and fast SLAM to navigate through the hallways of the competition arena and locate the control panel. Fast SLAM uses raw LiDAR data along with data output from computer vision software to map the vehicle's surroundings and track the position of the vehicle with respect to those surroundings. The OpenCV computer vision library is implemented on the platform for computer vision tasks including IARC emblem detection and blue led detection.

2.2.1 Stability Augmentation System

The 3DM-GX2 Inertial Measurement unit is made by Microstrain, Inc. and has integrated Kalman filtering for sensor data fusion and noise filtering. The IMU contains triple axis accelerometers, gyros and magneto sensors. The inertial data collected by the imu is used by a pid control algorithm to stabilize our aircraft and control its flight.

Initially, the feedback system was tested on a single pivoting arm platform using a PWM controlled hobby brushless motor controllers with an update rate of 20hz. Stable control was not possible with this low of a system update rate. To remedy this, a brushless motor controller design from the open source MicroKopter project was adapted for use on the vehicle. This new motor controller system has an I2C interface for motor speed control, allowing motor speeds to be updated at up to 500hz. The initial tests using the MicroKopter controllers used a 300hz control system update rate. With this system, vehicle attitude control functioned well. In order to read accelerometer data as well as orientation data from the IMU, we had to lower the system update rate to 100hz. We saw slightly slower reaction times after lowering the system update rate, but attitude control still functioned adequately.

For the majority of the initial testing, the propellers were positioned approximately 35 cm from the center of the vehicle. In order to reduce the weight of the vehicle, this distance was later reduce to approximately 22 cm. The decreased moment of inertia reduced the physical damping in the system and led to an increase in oscillation frequency and magnitude. After retuning the PID control constants we were able to achieve a stability level similar to the larger platform. The new platform also had noticeably faster response time to control inputs.

2.2.2 Navigation

The current navigational algorithm used starts with the robot mapping the main hallways of the building and noting doorways as nodes for further exploration. After the hall has been mapped, the robot will explore the room corresponding to the nearest node. The robot will move from nearest node to node until the control panel is found. In future implementations the intensity of the audible tone emitted from the control panel will be factored into decision making on which node to explore next.

A lateral control PID system using IMU accelerometer data was tested on the platform. Acceleration due to gravity was compensated for by using IMU orientation data. After integration, the accelerometer data drift caused the vehicle position control to fail. These tests showed that data from absolute position sensors such as ultrasonic rangers or LiDAR would be necessary to control the vehicle. Figure 1 in the Appendix shows the control system diagram.

2.2.3 Flight Termination System

The failsafe flight termination system consists of a “kill switch” and two wireless Zigbit900 modules following the 802.15 wireless protocol. When toggled, a signal is transmitted to the Zigbit receiver located on the aerial vehicle. The signal causes the reset lines of the motor controllers to a low state, effectively stopping the vehicle.

3. PAYLOAD

3.1 Sensor Suite

3.1.1 GNC Sensors

The guidance, navigation, and control for this system utilizes a combination of a LiDAR range finder and a webcam for vision processing.

At this time the approach is to utilize LiDAR data for mapping the flight area while utilizing vision to identify key features including the IARC emblem and the blue LED on the control panel to direct the vehicle though the mission.

The vision system is based around a Logitech Quickcam. This camera utilizes a rolling shutter, which means that the image is not captured in one time instant. As a result, video performance is poor during flight. Future improvements will include purchasing a camera with a global shutter.

3.1.2 Mission Sensors

The webcam is utilized as the primary sensor to detect both the IARC emblem and blue led. The on-board controller also supports a microphone allowing for frequency based detection of the objective control panel.

3.1.3 Target Identification

In order to accurately gather information and navigate, the aerial vehicle must be able to identify targets. The targets of interest in this case (a window entrance to the building and a specific control panel) can be identified by the International Aerial Robotics Competition (IARC) symbol that will be located above both the window and the control panel. Additionally, the control panel will be near a source of emission of a loud sound, and on the control panel there is a single solid blue LED (blinking blue LEDs may be found throughout the building).

3.1.3.1 Identifying the Location of the Sound

The location of the sound will be determined by traveling to the area of the greatest intensity. As the vehicle moves through the building it will compare the current amplitude of the alarm tone (assumed to be between 2-4 KHz) to the previously measured amplitude. The highest relative amplitude will be considered the best possible direction for travel.

3.1.3.2 Identifying the IARC Symbol

The current method used to find the International Aerial Robotics Competition (IARC) symbol that will be located above the window entrance to the building involves use of a Speeded-Up Robust Features (SURF) algorithm [1]. SURF uses statistical comparison between a training image and an image for search to locate conserved features. Areas of images with a high density of conserved features can therefore be "recognized." The algorithm was implemented by modifying object recognition sample code and functions from OpenCV, an open source computer vision library. Two examples showing the conserved points between the training image and the search image (which can be blurred and distorted with relatively little effect on recognition) are shown in Figure 2 in the Appendix.

After detecting the sign, it is useful to know the coordinates of the center of the sign in the global reference frame for navigation through the window. The center of the sign is currently found by simply averaging the x- and y- coordinates of the conserved features found on the search image.

3.1.3.3 Identifying the Solid Blue LED

The current method used to locate the blue LED located on the control panel is to use OpenCV to locate pure blue light. Figure 4 shows two different filters that have been implemented. The image in the upper right shows only pixels where the blue component of the RGB pixel is larger than the red or green component. The brightness of the pixel shows the intensity of the blue component. The image in the lower right only shows pixels where the blue component of the RGB pixel is larger than the red plus the green component. This has shown to be an effective way to filter out white lights that might have a blueish tint. The last part of the program locates the most intense location of the lower right hand image and returns the row and column of an LED if there is one present.

3.1.4 Threat Avoidance

The vehicle is equipped with computer vision and LiDAR sensors, which can be used to detect and avoid obstacles. Software will be implemented which places obstacles on the map that is created from the SLAM algorithm so they can be avoided for the duration of the flight.

Redundant methods of sensing obstacles, with ultrasonic sensors and cameras, will allow the robot to avoid different types of obstacles under a variety of conditions.

3.2 Communications

The system utilizes an 802.11b/g/n wifi solution to communicate between the base station and flight system. This communication system allows all flight data, and sensory data to be transmitted to the base station. The base station is able to send commands and send control data over this system though standard socket connections. A secondary 900Mhz Zigbee based solution is used to transmit the kill signal.

3.3 Power Management System

The flight system uses a 3-cell (11.1V) 5000mAh Lithium Polymer battery pack to power the motors, and a single cell (3.7V) 2000mAh battery pack to power the onboard electronics. The power management system utilizes a pair of step-up regulators to provide two 5V rails for the various peripherals. In addition, a 3.3V and a 1.8V linear regulator provides power to the rest of the electronics. The flight system utilizes an ideal-diode circuit on the system input to allow for a redundant battery supply. This allows the packs to be swapped without having to shutdown the on-board computer.

Both the battery packs are monitored onboard, allowing remote battery voltage detection. A pair of onboard LEDs also indicate the current charge state of these packs. This allows for a quick visual inspection of the on-board batteries. The battery packs were chosen to satisfy a 20-minute flight time.

4. OPERATIONS

4.1 Flight Preparations

Our operations procedure consists of a sequence of pre-flight checks and tests that check the integrity and operational functionality of our flight platform. These checks include motor and prop alignment and orientation tests, wifi connectivity and data transmission tests, sensory system checks, and software user interface functionality tests. These pre-flight checks/tests have been established over time to insure that each component of the system is up and running properly before operation commences.

4.1.1 Wifi Connectivity & Data Transmission

After the initial boot up of our onboard Gumstix computer, the wifi interface needs to be checked to make sure that is properly negotiates with the router. Upon successful connection, a blue LED lights up on the Gumstix to indicate that a connection is established. If the blue light fails to come online, a script can be run through the serial control terminal to bring up the wifi interface and have it try to reconnect to the router. Once the connection is established, the platform software can be started in order to begin receiving data back from the platform to the graphical user interface (GUI).

4.1.2 Sensory System

With the GUI up and running, sensors can be checked by interacting with the vehicle (rotating the platform, placing objects in line with the LiDAR/sonar sensors, etc). Motors can also be tested at this stage.

4.1.3 Motor/Prop Alignment and Orientation

During initial startup after motor configuration modifications, the motors are set to idle at a very low speed. This is important for seeing if they are turning the correct direction and if the appropriate propellers are installed since a set of counter-rotating propellers are used on the vehicle.

By setting up a rigorous, linear sequence of tests, issues that have arisen during the startup sequence can be caught. In addition, any member of the team can operate the platform by following the startup procedure. Once these tests have passed, our system is ready to be placed into manual or autonomous flight modes and can be controlled through our graphical user interface.

5.2 Man/Machine Interface

The operator has two interfaces with the vehicle during normal operation: through a power connection to the vehicle and through a wirelessly connected computer.

To apply power, there are two battery connections that must be made. Because of the sensitivity of low power integrated circuits to noise, a smaller battery provides separate voltage for all processors and microcontrollers and therefore has a separate connection. Once the smaller battery is attached via the polarized JST connector on the processor PCB, the main 5Ah battery can be connected. By default, the motors are physically disabled and do not start until the computer interface initializes the software.

The computer interface provides detailed debugging information, as well as the ability to modify settings. Once the vehicle is powered and has established WiFi connections, the program may be started by running the "run" script from the "/software/host" folder. If manual operation or debugging is desired, it may be done here.

To run the mission script for an IARC Mission 5 scenario, the "iarc5.mission" file must be loaded. After loading, press the "Run Mission" button in the "Mission" tab. Details regarding mission progress are listed on the screen as they occur. Pressing the "Emergency Landing" button causes the vehicle to end the mission and land immediately, or pressing the "Emergency Kill" button will immediately stop the motors, and render the vehicle ballistic.

Some additional things to add to the GUI in the future might be preflight battery checks (check battery voltage), mechanical system checks (check to see things are bolted in and connected properly), and to verify that we are in a safe area to operate our platform.

5. RISK REDUCTION

5.1 Vehicle Status

The vehicle has the ability to be flown autonomously or through manual control. When flown autonomously all navigation is based on the ultrasonic sensors and LiDAR output, while in manual mode these sensors are not enabled. In both modes the IMU is used to ensure stable flight and to control balance through the roll, pitch, and yaw axes.

5.1.1 Shock/Vibration Isolation

The use of small, efficient outrunner electric motors means that our platform has very little vibration from the motors. The inherent structure of the quad-rotor and its way in which rotational torques cancel each other means that the design is stable with neutral response. The center of gravity is set low to enhance the pendulum-like stable characteristics of the quad-rotor platform. The outrunner motors used mean smoother operation at lower RPMs and are an integral part of vibration reduction.

5.1.2 EMI/RFI Solutions

Vulnerable motor controller I2C signals have been completely shielded to reduce risk of motor malfunction. Important to the communication system, all the wires on the antenna were shielded and electrically isolated to avoid potential interference. All wires are run through foam within the carbon fiber booms to provided further shielding from electrical interference.

5.2 Safety

Tethers connected the vehicle to plywood to ensure controlled flight during initial stages. During later testing, removable handles were added to give extra distance between team members and the propellers. In addition, during the testing phase foam golf balls were placed on each corner of the vehicle to aid in landing and to make sure no damage was done to the vehicle or adjacent team members. A battery monitoring system is used to make sure the motors do not reach a low voltage state to prevent damage to motors and batteries. The vehicle has been shrouded to help keep observers safe while the vehicle is in flight. The kill switch for the vehicle is on a completely separate receiver that directly removes power from the motors when the kill signal is received with no external processing or circuitry required. Isolating this switch from the rest of the system allows for shutdown even in the case of wifi failure.

5.3 Modeling and Simulation

Upon deciding to build a quad rotor system, a Blender simulation in Python language was created to show various aspects of the quad-rotor system. Later, ultrasonic obstacle avoidance and mapping algorithms were added to the system to test initial hypothesis about the performance of those systems. The mechanical design was modeled using CAD in SolidWorks to ensure space for components, optimal size of the propellers and to evaluate weight, wiring and flight dynamics.

5.4 Testing

The platform was tethered during initial tests to reduce the risk of breaking the system during a crash. Initially, a dual-rotor balancing arm was used as an initial test for the IMU. Upon successful completion of the dual-rotor balancing arm tests the number of rotors was doubled to implement the same control algorithms to a quad-rotor balancing arm system. After the carbon fiber chassis was constructed, tests were conducted using removable handles that extended from each arm. These allowed for easier PID tuning. All testing was incremental in order to ensure continual progress to the goal of complete, autonomous, un-tethered flight.

6. CONCLUSION

The quad-rotor vehicle is an optimal choice for payload and maneuvering capabilities. The vehicle is capable for many types of missions and applications.

7. REFERENCES

[1]Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, "SURF: Speeded Up Robust Features", *Computer Vision and Image Understanding (CVIU)*, Vol. 110, No. 3, pp. 346--359, 2008. Available: ftp://ftp.vision.ee.ethz.ch/publications/articles/eth_biwi_00517.pdf

[2]Cyrill Stachniss, Udo Frese, Giorgio Grisetti, “OpenSLAM” <www.openslam.org> Date Accessed: July 20, 2009.

APPENDIX. Figures

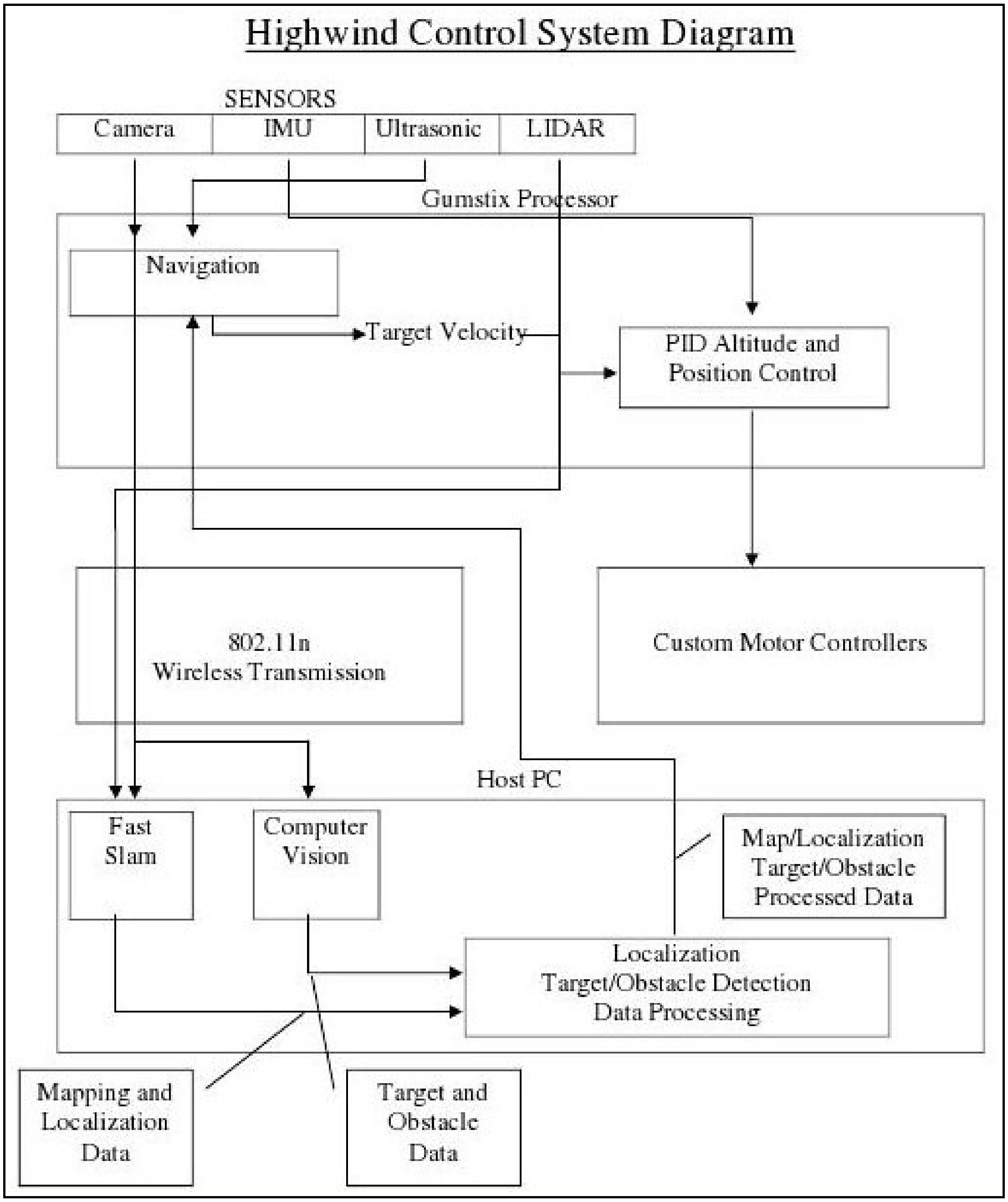

Highwind Control System Diagram
SENSORS
Camera
IMU
Ultrasonic
LIDAR
Gumstix Processor
Navigation
Target Velocity
PID Altitude and Position Control
802.11n Wireless Transmission
Custom Motor Controllers
Host PC
Fast Slam
Computer Vision
Map/Localization Target/Obstacle Processed Data
Localization Target/Obstacle Detection Data Processing
Mapping and Localization Data
Target and Obstacle Data

Figure 1: Control system diagram

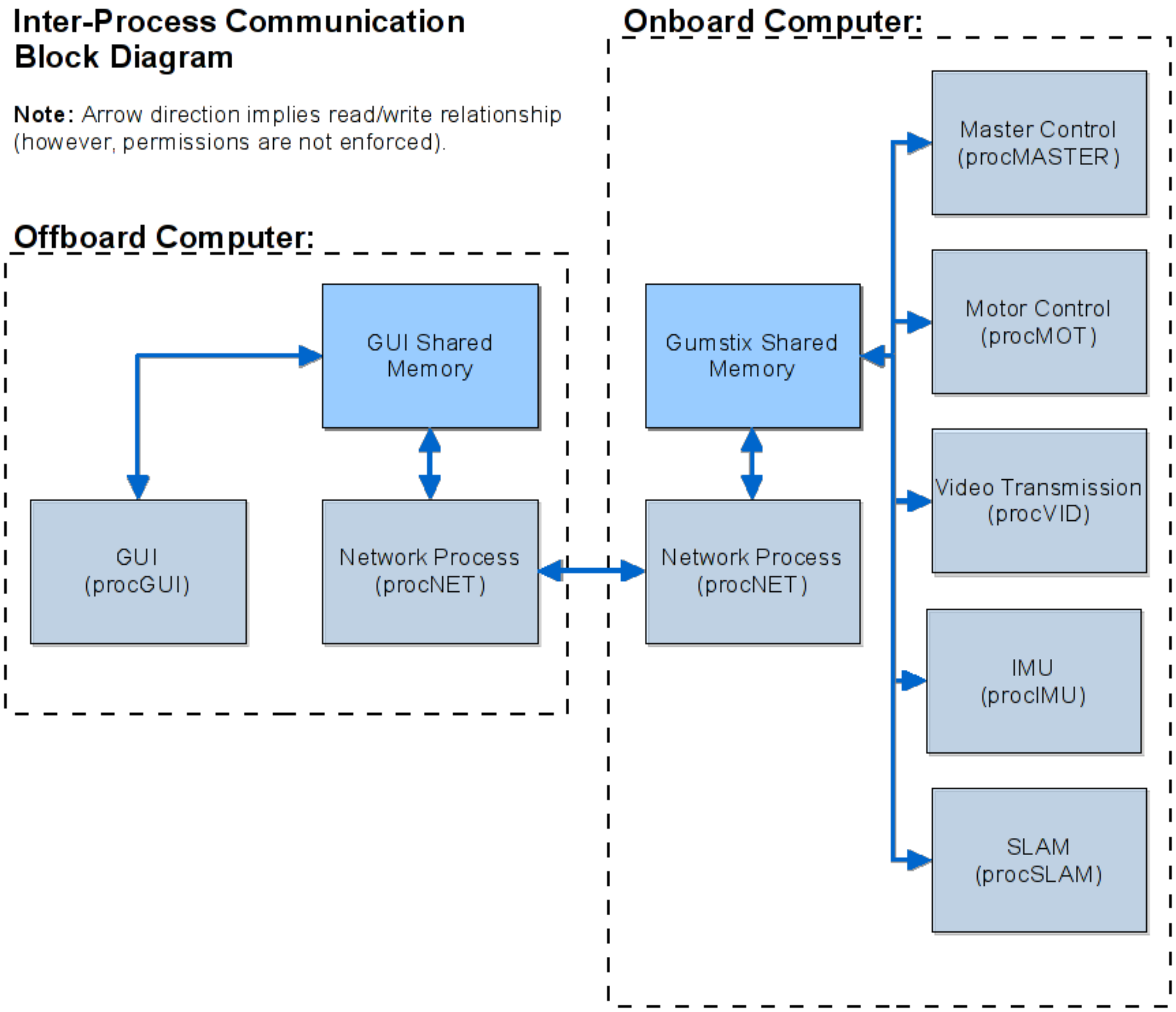

Inter-Process Communication Block Diagram
Note: Arrow direction implies read/write relationship (however, permissions are not enforced).
Offboard Computer:
GUI Shared Memory
GUI (procGUI)
Network Process (procNET)
Onboard Computer:
Gumstix Shared Memory
Network Process (procNET)
Master Control (procMASTER)
Motor Control (procMOT)
Video Transmission (procVID)
IMU (procIMU)
SLAM (procSLAM)

Figure 2: Diagram describing shared memory system

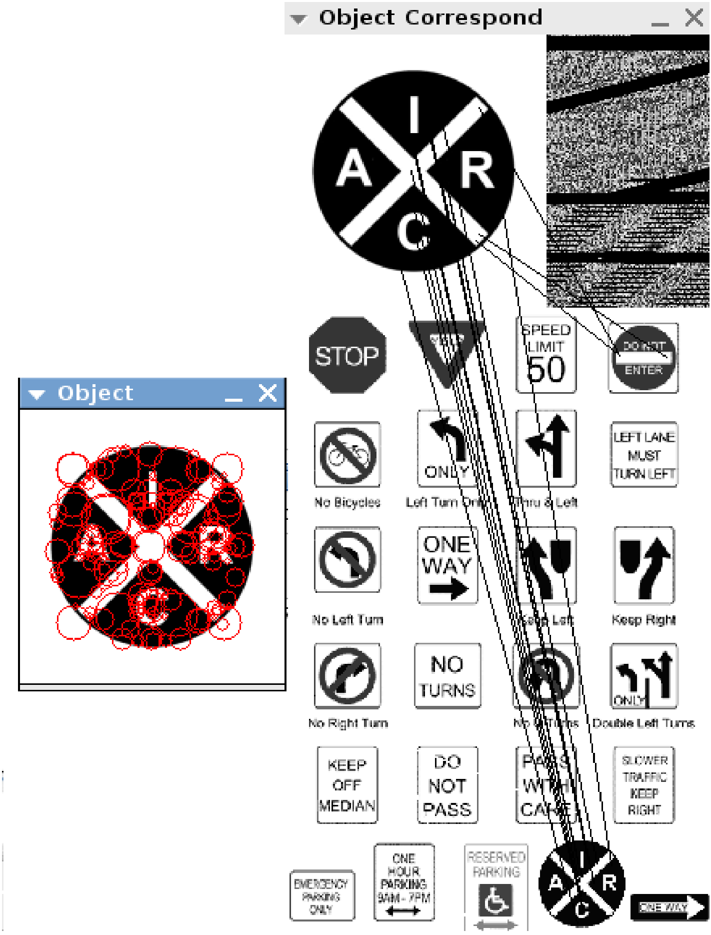


Figure 3: Example of SURF algorithm identifying IARC emblem

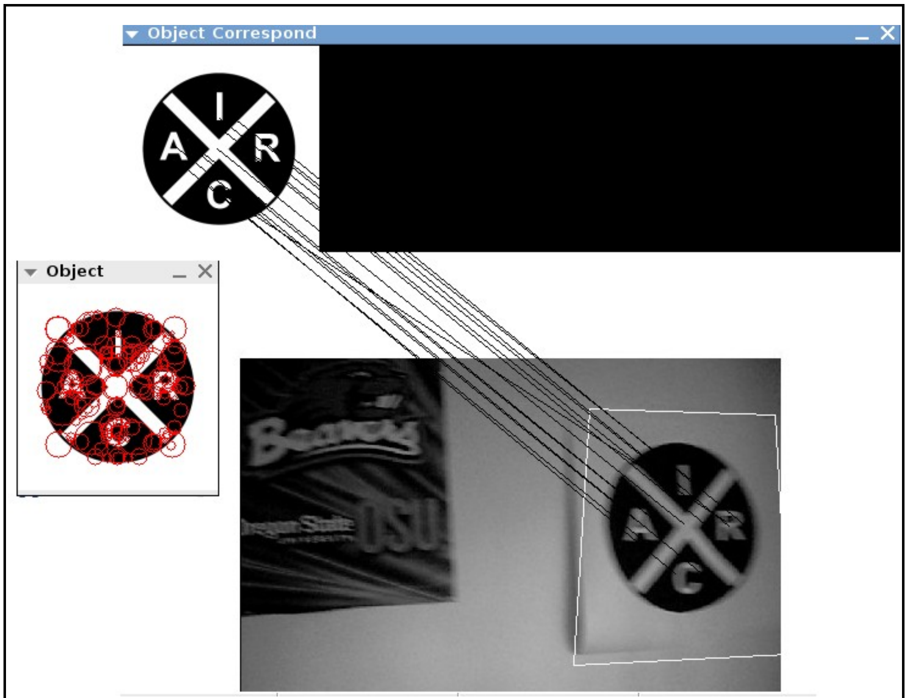
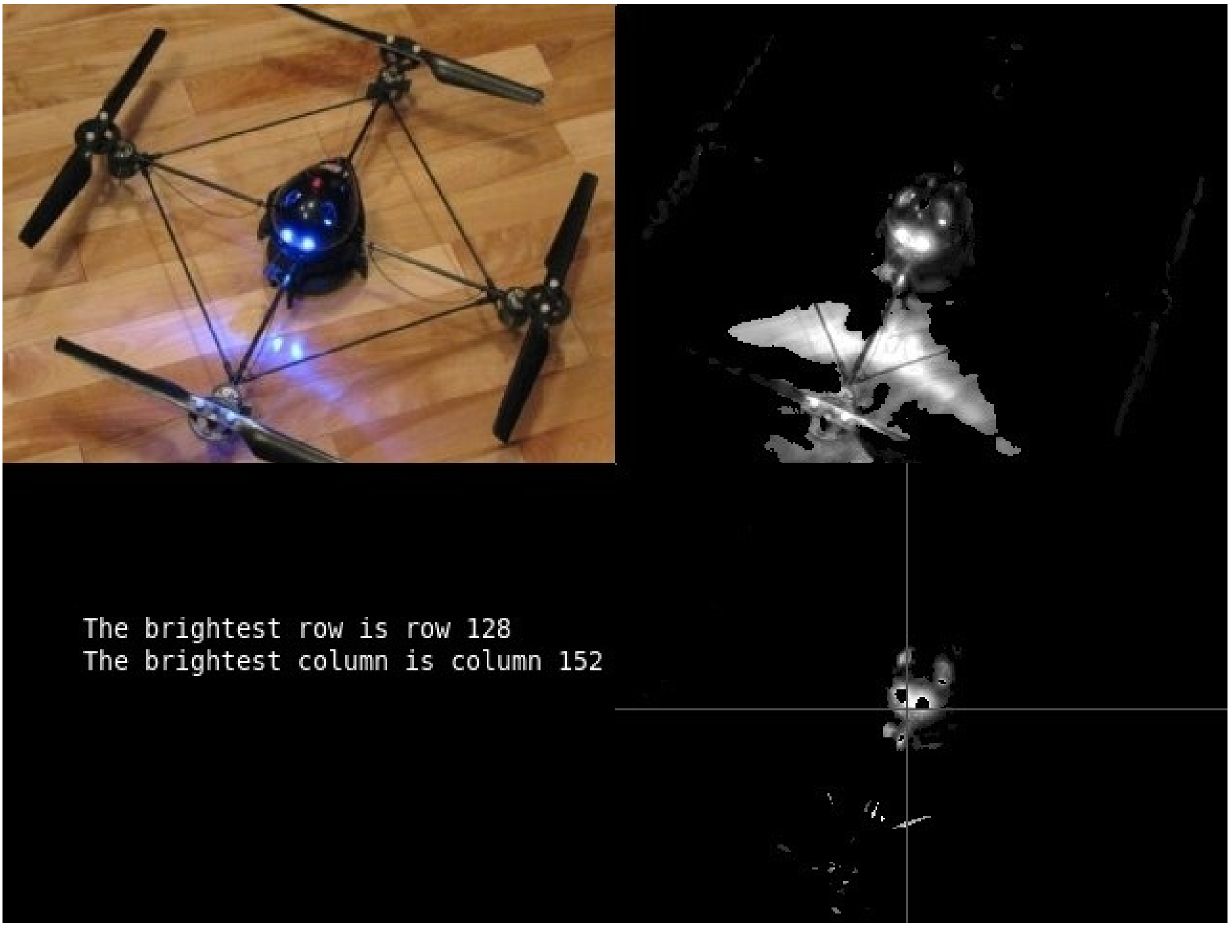
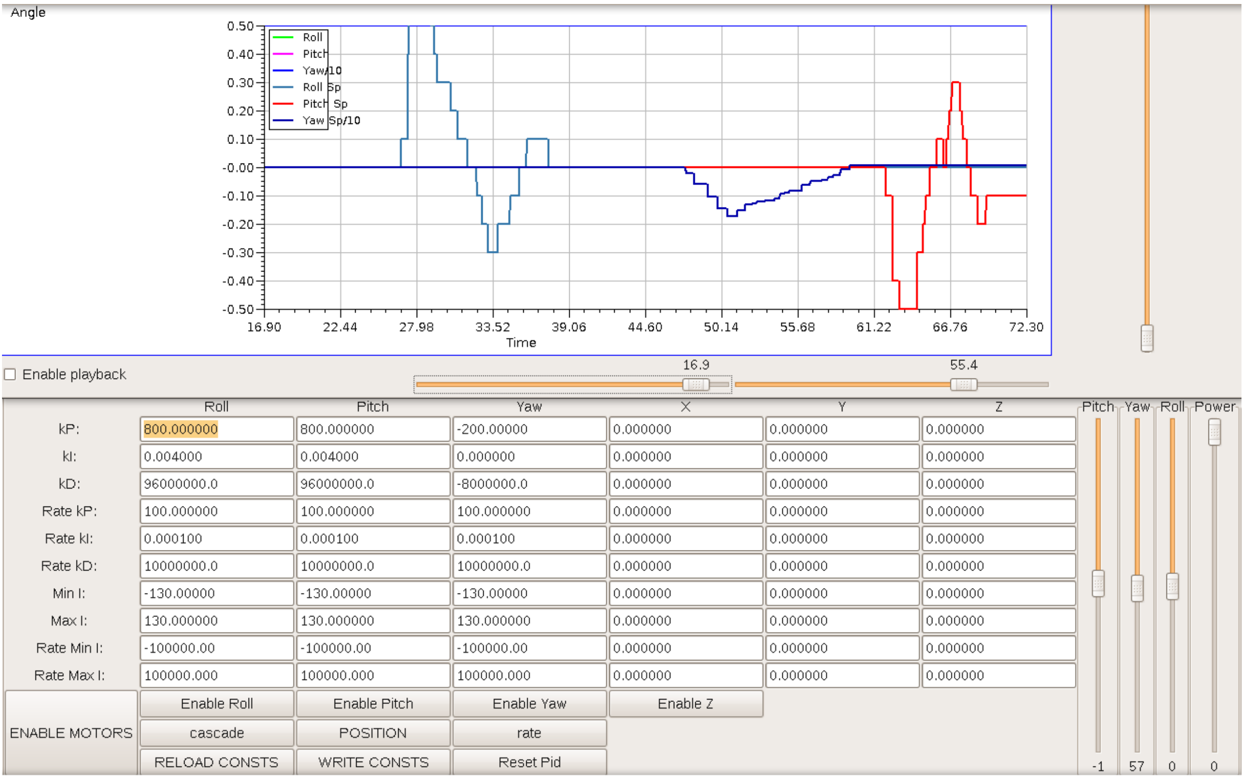


Figure 4: Example of SURF algorithm identifying a blurry and slanted image of the IARC emblem



```
The brightest row is row 128
The brightest column is column 152
```

Figure 5: Example of Blue LED detection algorithm, written in OpenCV.



□ Enable playback

	Roll	Pitch	Yaw	X	Y	Z
kP:	800.000000	800.000000	-200.00000	0.000000	0.000000	0.000000
kI:	0.004000	0.004000	0.000000	0.000000	0.000000	0.000000
kD:	96000000.0	96000000.0	-8000000.0	0.000000	0.000000	0.000000
Rate kP:	100.000000	100.000000	100.000000	0.000000	0.000000	0.000000
Rate kI:	0.000100	0.000100	0.000100	0.000000	0.000000	0.000000
Rate kD:	10000000.0	10000000.0	10000000.0	0.000000	0.000000	0.000000
Min I:	-130.00000	-130.00000	-130.00000	0.000000	0.000000	0.000000
Max I:	130.000000	130.000000	130.000000	0.000000	0.000000	0.000000
Rate Min I:	-100000.00	-100000.00	-100000.00	0.000000	0.000000	0.000000
Rate Max I:	100000.000	100000.000	100000.000	0.000000	0.000000	0.000000

Figure 6: Graphical user interface used for low level control and control loop tuning.