MicroCART 2013-2014

Project Plan

Team Members:

Luke Mulkey

Bill Franey

Kevin Engel

Kelsey Moore

Aaron Peterson

Mike Johnson

Nathan Ferris

Faculty Advisers:

Dr. Nicola Elia

Dr. Phillip Jones

Table of Contents

Page

1. Problem Statement.............................3
2. Solution/Procedure Statement...........3
3. Concept Sketch.................................. 5
4. System Block Diagram....................... 6
5. System Description............................ 7
6. Operating Environment..................... 7
7. UI Description...................................... 7
8. Functional Requirements.................. 8
9. Non-Functional Requirements.......... 8
10. Deliverables..9
11. Work Structure...................................9
12. Resources.. 11
13. Timeline...12
14. Risks..14
15. Conclusion.. 15

Problem/Mission Statement

Currently, the system is trapped indoors and is heavily reliant on an expensive high-speed camera system. This severely limits mobility and reduces potential for real-world applications. The current mission is to convert the system from being completely controlled by an external computer and camera system, to one that is almost entirely on-board the vehicle. This will require using bluetooth to communicate between a laptop base station and the on-board sensors and FPGA. These changes will allow us to fly outside and demonstrate some autonomous functionality.

Solution/Procedure Statement

To achieve the goal of outdoor flight and the demonstration of an autonomous task, there are a series of steps designed to help us achieve them in a timely and logical fashion.

Stage 1: Handoff

- Become familiar with necessary tools and environments
  - Quartus
  - QT
  - Kalman Filters
- Learn how to run the old system
  - Ensure the quad-helicopter has a functional backup system at any given time that is capable of autonomous flight and basic functionality
  - Understand how tools operate together to create a functional system

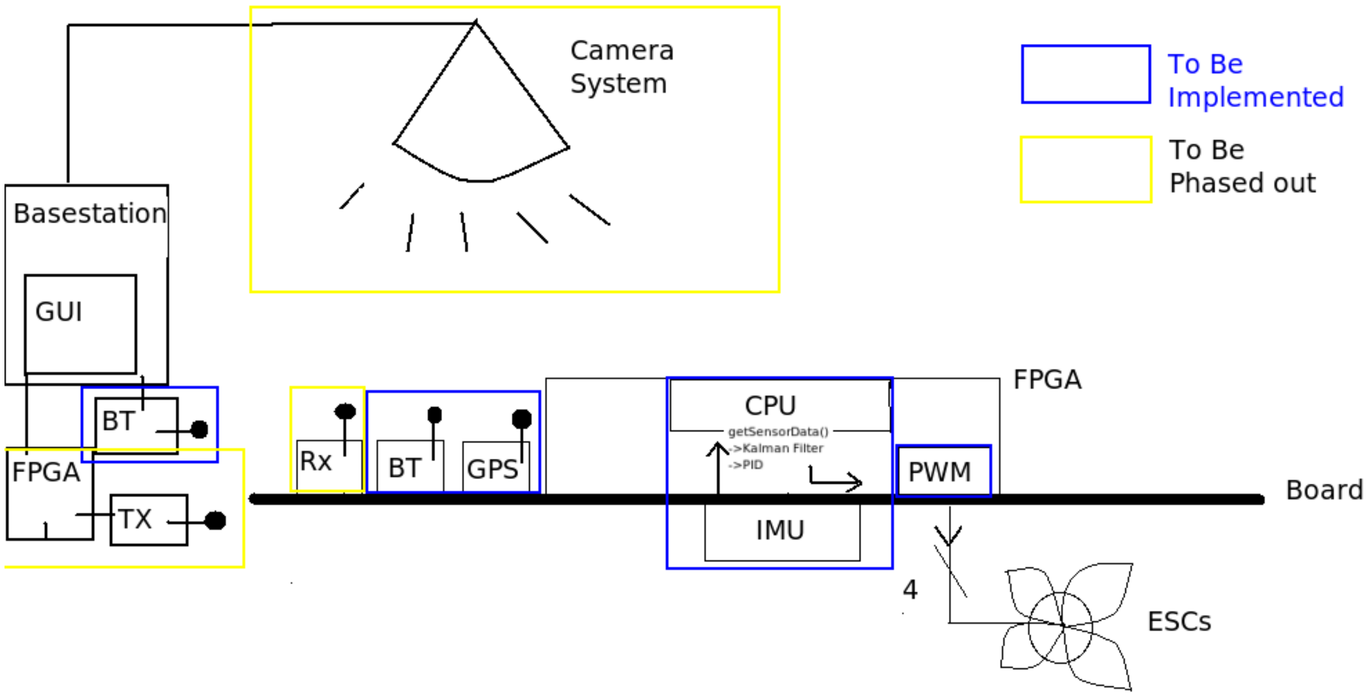
Stage 2: Component Implementation and Testing

- Establish communication between the board and the base station via bluetooth
  - Ability to transfer flight data for Kalman filter testing
  - Ability to read sensor data from the IMU
- Evaluate functionality and accuracy of Kalman filter using real data
- Create new GUI using refactored base station code
  - Test refactored code's functionality
  - Recreate last year's GUI, referencing the refactored base station code
- Evaluate PID controls and integration

Stage 3: System Integration and Execution

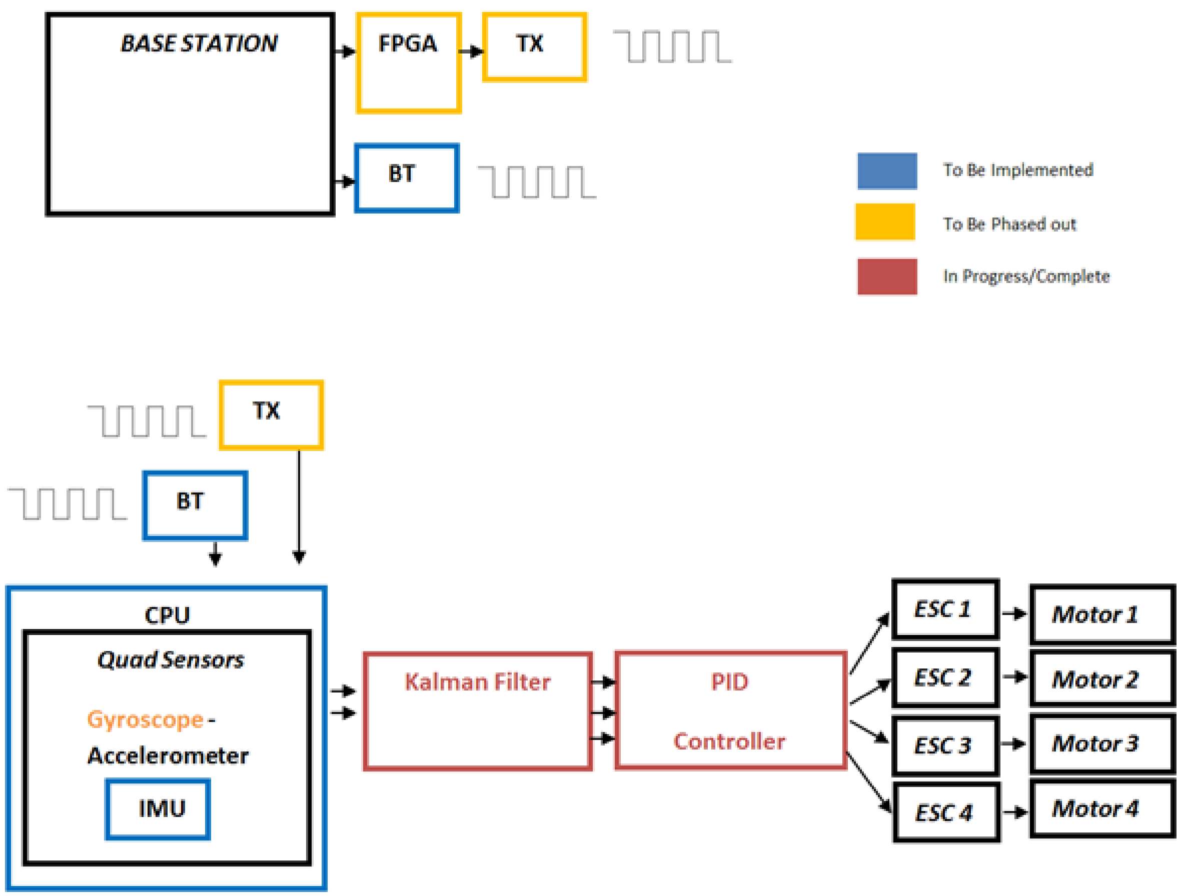
- Test functionality of internal sensors using pseudo-GPS (i.e. the intentionally slowed down camera system)
- Test quad-helicopter outdoors using the GPS while tethered

Concept Sketch



This conceptual sketch is a high-level mixture of the current layout and the envisioned layout for components going forward. Currently, the camera system keeps track of the quad-helicopter’s whereabouts and reports back to the base station. The basestation then performs calculations and kicks controller data through an FPGA/transmitter system to the receiver on the quad-helicopter. This process sends that data to a stock “black-box” gyro system(not pictured), before finally sending its generated PWMs to the ESCs. This system and all of its components (boxed in yellow) will all eventually be phased out. The new system will have an onboard IMU containing gyroscopes and accelerometers which will return data to the onboard FPGA in real time. This data will be smoothed using a Kalman filter, and then it will be interpreted by a PID controller before being converted to PWMs and sent to the ESCs. The important components of this new system are outlined in blue. This system is superior for a few reasons. First, it does not rely on a spatially constrained camera system. Second, the quad-helicopter will no longer have "mystery" commands by removing the stock black box GU-344 and introduce new custom IMU and PWM signals. This will allow for total control of the quad-helicopter both indoors and out.

System Block Diagram



System Description

The system starts with a start signal moving from the base station over the bluetooth to the FPGA board. The Kalman filter smooths the data to be sent to the PID controls for the motors, and once received, the motors act accordingly. Sensors send the data to the PID controller as the motors are functioning. The FPGA takes this data and moves the PID controller accordingly.

Operation Environment

The goal for the project is to enable the quad-helicopter to fly autonomously outdoors in an environment that is very different from where the quad-helicopter currently flies for all of its testing. The test environment will be located indoors where wind is negligible and the climate is controlled. Tests will take place in a three dimensional grid defined by the IR camera system that is about 27 m^3 , with a thin carpet landing surface.

After indoor testing has been completed, the operating environment will move outdoors. Here, the quad-helicopter will be intended to operate in an environment that is very similar to the indoor one. While restricting the limitations of the current hardware, the plan is to operate in winds no greater than 10 MPH and temperatures between 40°F to 90°F; the outdoor operating environment will contain no obstacles.

UI Description

The design is intended to be used for demonstrations of embedded systems and controls integration on a autonomous flying vehicle. It also adds to the continued research and development of autonomous flight vehicles. For this reason, the intended users are those trained for the system. The reason for this is that the interface requires a set up for the quad-helicopter by programming the FPGA board, debugging and fixing the quad-helicopter when something doesn't go accordingly, and a list of commands currently for flight. Once completed, the average American population should be able to use the GUI to control the quad-helicopter after the initial setup.

Functional Requirements

The quad-helicopter will need certain requirements to function properly. The quad-helicopter first needs to be able to fly autonomously inside on the regular camera system. It then needs to be able to fly on a slowed camera system to simulate GPS data. From this, the quad-helicopter should only accept the correct input commands, and the Kalman filter and PID controllers should be properly functioning to stabilize the quad-helicopter. Afterwards, the task of taking the quad-helicopter outside will be one of the two final requirements.

Non-Functional Requirements

As with any creation, non-functional requirements are important to consider for its implementation. For the quad-helicopter, and important non-functional requirement is safety for those observing and flying the quad-helicopter. Another non-functional requirement is to have an easy to use GUI. The final requirement is to be able to perform a specific task that has been received from the GUI to do. The task will require the vehicle to locate objects around it to avoid collisions in an indoor demonstration.

Deliverables

The deliverables will include the transition from the old system (base-station, IR camera system) to the new system (on-board processing, additional sensors, bluetooth communication). With the new system, we will then prepare the quad-helicopter to fly outside. From there, the final deliverable is to perform a successful automated “live-scenario” such as a rescue mission outside. If completed early, additional tasks will be created such as modifying the GUI to increase performance.

Work Structure

I. Overall Design

1. Stage 1 - The handoff process from last year’s team
    a. The team will formulate ideas and goals to implement on top of last year’s project.
    b. The team will understand how to use the camera system, fly the quad-helicopter, and use development tools like QT and Quartus.
    c. The team will work with previous team members to ensure understanding of all code and systems that will be used.
2. Stage 2 - Component Implementation and Testing
    a. Establish communication between the board and the base station over a bluetooth link in order to retrieve data from the quad-helicopter.
    b. Recreate last year’s GUI using the refactored base station code.
    c. Evaluate Kalman filter and PID controls using received data.
3. Stage 3 - System Integration and Execution

a. Test system-wide functionality using the current camera system

b. Test functionality using a modified version of the current camera system by slowing the data down to spoof GPS behavior.

c. Test the completed system with all components outside.

II. Communication Design

1. Stage 1 - Implement bluetooth communication on FPGA

    a. Program the FPGA to communicate via bluetooth to the base station computer

    b. Setup the base station computer to listen for data sent over bluetooth

2. Stage 2 - Ensure sensor data quality over bluetooth

    a. After establishing communication the team needs to verify sensor data accuracy.

III. Sensor Filter and PID Controller

1. Stage 1 - Comparing and modifying PID controller and Kalman filter

    a. Compare past files for design and accuracy

2. Stage 2 - Choose and implement well working model

    a. Compare different models to ensure best implementation

    b. Test the PID controller and Kalman filter to ensure the desired results are being produced.

VI. User Interface Design

1. Stage 1 - Learning and understanding last year's design

    a. Understand how to use the QT development tools and the C++ language

b. Reading documentation to get familiar with the code structure

2. Stage 2 - Integrating the refactored base station code with last year's GUI

    a. Strip all the old code out of the GUI to get a view with no logic.

    b. Integrate the new and refactored code in the old GUI

3. Stage 3 - Integration Testing

    a. Ensure the old GUI system works with the refactored code by testing it on the base station

    b. Ensure the entire system behaves as intended by ensuring data coming to and from the base station is as expected

Resources

Already acquired resources:

Amount	Part
1	Initial Quad-helicopter
2	Training Quad-helicopter
3	Three-cell Lithium Batteries
1	FPGA Board
2	Bluetooth Devices
1	Basestation Transmitter

To be acquired:

Amount	Part
1	Improved FPGA Board

1	SD Card
1	Camera

Estimated Resource Amount:
$500

Timeline

October

1. Communication with the quad-helicopter
    a. Bluetooth communication commands understood
    b. Bluetooth commands tested
    c. Camera system information being sent to base station
2. Testing Kalman Filter
    a. Receive information from camera system to test IMU data
    b. Check if Kalman filter data matches the IR readings
    c. Replace current time constant with the difference in time stamps
3. New refracted code
    a. Test if quad-helicopter supports new code
    b. Integrate code into existing GUI
    c. Test that quad-helicopter still operates as expected

November

1. Improved PID
    a. Work with information sent from quad-helicopter to determine PID controls
    b. Test PID controls

2. Improved Kalman filter
    a. Use information from last month to integrate Kalman filter into hardware
    b. Test to see how close the information is to the IR camera
3. Test overall system flight with Kalman filter and PID controls

<u>December</u>

With a small gap between breaks the only goal is to test the quad-helicopters flight inside using pseudo GPS (slowed IR camera system).

<u>January</u>

1. Continued work with flight using pseudo GPS
2. Test flight outside in a safe-zone area
    a. Test wind conditions
    b. Test temperature

<u>February</u>

1. New “on-board” FPGA-board
    a. Transition to new board
    b. Test to see if all systems are still working correctly inside
    c. Test to see if all systems are still working outside
2. Continue outside flight testing
    a. Continued information gathering
    b. Finish GPS integration

March

1. Add new sensors
    a. Implement camera system for altitude testing
    b. Attach SONAR for altitude testing
    c. Determine which system should be easier to use
2. Implement PMODS that may be used for indoor task

April

1. Continue testing sensors and finish ability of quad-helicopter to fly outside
2. Continue work on indoor task

May

Plan on having the quad-helicopter flying stably outdoors and able to successfully and autonomously complete the task.

Risks

The only major risk to this project is personal injury. If the quad-helicopter were to lose control during flight testing, serious injuries could occur. To ensure safety, two major safety measures have been added. One of the features is that at any time a switch can turn the quad-helicopter over to manual flight. This manual flight will always have the throttle set to zero and the quad-helicopter's motors will stop, preventing damage to people, objects, and the quad-helicopter itself. The other safety measure is to have the quad-helicopter tethered to the ground, or place a net surrounding the quad-helicopter as it is in flight to keep it from hitting bystanders.

Conclusion

The overarching goal for the MicroCART project is to no longer use the twelve-camera system that has been used for the last two years so the quad-helicopter can fly outside. It will instead use an onboard GPS system, PID controller, and bluetooth integration for onboard commands. Using this idea, we will also implement a command task that will take in the objects around it and perform a task (e.g. putting out a simulated fire). This could lead to easier usage of the quad-helicopter and potential for future MicroCART students to implement better task integration and have a more open area to work in.