EE CprE 491 – May 1718
MicroCART Senior Design Team
Week 4 Report

September 26 – October 2
Faculty Advisors: Phillip Jones, Nicola Elia

Team Members:

Brendan Bartels — *Recorder*
Kris Burney — *Ground Station Key Concept Holder*
Joe Bush — *Quadcopter Software Key Concept Holder*
Jake Drahos — *Team Webmaster*
Eric Middleton — *Hardware Maintainer*
Tara Mina — *Team Communications Leader*
Andy Snawerdt — *Control Systems Key Concept Holder*
David Wehr — *Team Leader*

Summary for Progress this Week

This week, we started working more closely with the quadcopter, including actually rotating the motors by providing a PWM signal from a function generator directly to the pins connected to the ESC machine, which gave us a much better understanding of how at the hardware level the motors are being controlled. In addition, we found some issues that we will need to address, and have begun addressing, for taking accurate measurements with ECP machine.

Past Week Accomplishments

- Came up with and ran a setup to measure thrust constant – Tara, Andy, and David
    - With Matt's help, we came up with a setup for doing the following:
        - Measuring thrust constant of the quad
        - Creating a constant torque on the quad to measure moment of inertia with
    - Figured out how the motors rotate with how/where the PWM signal goes to the ESC to control the motor speeds
        - Found and mapped out all of the hardware connections
            - Determined what the wires going to the ESCs were and what they did:
                - 1 ground pin - brown wire
                - 1 high voltage pin - red wire (not connected to anything, there is a different wire providing power to the ESC, which is connected to the battery)
                - 1 signal pin - brownish-yellow wire, where the PWM signal is given to the ESC
            - Found which ribbon cable pins are connected to the PWM signal-providing wires to each of the ESCs
            - Mapped the pins on the ribbon cable to the input pins of the Zybo board

        - Determined the characteristics of the PWM signal and how it relates to motor speeds:
            - Goes from 0 to 3.2-3.3 volts
            - Frequency of 400 Hertz
            - Needs a duty cycle of 40% (1 millisecond pulse width) to initialize the ESCs, after which increasing the pulse width allows the motors to rotate
            - From this point, increasing the duty cycle to about 47% begins to rotate the motors
            - Can increase the duty cycle to about 80% or a bit beyond (did not go beyond this value, because it was starting to go pretty fast)
        - Basic features of our setup
            - Replaced the battery with the older, analog power supply that can give more current
                - To provide a constant voltage source, which is needed to have accurate measurements for testing
                - Measured this with multimeter to mark how the voltage provided decreases slightly as the motor speed increases
            - Put a scale on top of a stand above the group (to reduce ground effect)
            - Put the quadcopter upside down on a roll of duct tape on a scale, and zeroed the scale, so that any additional downward force would represent the thrust of the motors
        - Basic measurement procedure
            - Increased the duty cycle one percent at a time (our independent variable we are controlling)
            - For each change in the duty cycle, measured the following:
                - Voltage provided by the power source
                - Current pulled (or provided by the power source)
                - “Weight” on the scale in grams - which can be converted to a force
                - Revolutions per second of each motor (as measured by a tachometer)
- Determined that the ECP machine is not providing a constant torque - Brendan, Tara, and Andy
    - See the “Pending Issues”
- Wrote a MATLAB script that will process data from the ECP machine - Andy
    - Takes in a text file of data measurements as recorded by the ECP machine
    - Takes in an absolute file path for the data (rotating the quadcopter) and the calibration data (rotating everything, except the quad)
    - Parses in the data columns, saves them as data vectors
    - Determines when the voltage applied goes negative (since the disturbance applied first rotates in one direction, then in the opposite direction), at which point, we disregard the data
    - Uses the following to calculate the moment of inertia:

        - Expressions derived in Matt’s thesis
        - Proportionality constants specific to our setup, including:
            - Relating ticks to degrees/radians rotated
            - Relating volts to Newton-meter torque applied
    - Goes through multiple files within a directory to parse and averaging more data to get one value for I, thus giving us a more accurate measurement of the moment of inertia
- Measured yaw moment of inertia again and pitch moment of inertia - Brendan, Tara, and Andy
    - Ran the same processes as we did before last week to measure the yaw moment of inertia, this time without the battery pack, which will throw off our results
    - Also, ran the same process as we did last week, but with the quad rotated to measure the pitch moment of inertia
- WiFi Work - David
    - Tried setting up linux as SoftAP, but wasn’t successful. Discussion with Eric led to the conclusion that the ESP chip should be the access point.
        - Setting up the WiFi module to be an access point will make it easier to initialize communication between the ground station and the quad. If it is not an access point, then to perform point-to-point communication, the ground station must be configured as an access point. This requires 3rd party tools on linux, and is dependent on the ground station WiFi being supporting SoftAP, which not all modules do.
    - I was able to get it set up as an access point, and run my existing UDP communication test, after some troubleshooting of the ESP8266.
        - In this process, I also learned how to use features that are only in the Espressif SDK, allowing us to use the full capability of the ESP8266 chip.
    - Began writing code so ESP8266 can be drop-in replacement for Bluetooth module.
- Other ground station work
    - Adding features to VRPN
        - FPS calculation
        - Timestamp
    - These are prereqs for working towards latency calculation
        - There needs to be some way to uniquely identify a packet to track latency with the quad

Pending Issues

- ECP machine not giving constant torque – Brendan, Tara, and Andy
    - The ECP machine power source is not providing the voltage we specify for the disturbance when testing
    - We know this because:
        - Our moment of inertia measurements are off when measuring known moments of inertia
        - But, the encoder position measurements are very accurate when we rotated the ECP machine manually, 1 revolution, 2 revolutions, etc.
        - Using the expression given in Matt’s thesis to convert a given voltage disturbance to moment of inertia based on the change in position and time

difference, we applied specific disturbances to measure a known moment of inertia, and solved for the disturbance voltage, and got very different voltage values compared to what we were specifically directing the machine to apply

        - We switched the rotating device with another one in the lab, while keeping the power source box the same and obtained the same faulty results (after calculation, getting different input voltages than what we were applying)
        - We tried switching the power source box with another one in the lab and still got faulty results, but they were different results from the other power source box, even though we were defining the same voltage value when using the software tool to define our disturbance to be applied
    - This is a problem, because it means the following:
        - All of our moment of inertia measurements and estimates are incorrect, so we will have to redo this work
        - We need to come up with a different way of measuring the moment of inertia, using a different constant torque that is dependable
            - We have talked to Ian and Matt about possibilities
            - Have came up with a different way of applying a constant torque, which is based on a very similar setup for measuring the thrust constant, described in our documentation and a bit in this document
- Need to get new motors – All team members
    - One of the motors is not functioning properly and the motors that are on the quadcopter right now are not being made anymore, so we cannot simply replace the broken motor with the same kind of motor, and since we need 4 identical motors, we believe we will need to get 4 new motors, all of the same type
    - Possible ideas:
        - Could take motors from the other quadcopter
        - Could buy new motors
    - Need these as soon as possible, since many sub-teams are needing them for current work, such as the Controls sub-team for measuring thrust constant
- Base station - Jake, Eric, Kris
    - Issues with outdated C compiler on lab machine
        - Rewrite some code for compatibility with older standards, rather than best practices with newest standards
    - Adding logging and latency tracking
        - Timestamp can be used to identify packets, latency calculation can be done based on that
    - FPS calculation is far from ideal
        - It should be a rolling average
    - Code organization
        - Currently a bit scattered throughout git. This should be moved into our main repo ASAP

Individual Contributions

Team Member	Contribution	Weekly Hours	Total Hours

Brendan Bartels	Matt’s Thesis, Data Collection	5	19
Kris Burney	Bluetooth poc, added BEC to quad, explained quad log format to Andy.	15	38
Joe Bush	Made calendar, downloading/reading code	7	25
Jake Drahos	Base station work/VRPN debugging	7	25
Eric Middleton	Base station work, Bluetooth poc	7	36
Tara Mina	ECP data collection, thrust constant setup	11	35
Andy Snawerdt	ECP data collection, Thrust constant measurement procedure document, MATLAB scripts for moment of inertia calculations.	15	40
David Wehr	WiFi work, thrust measurement	9	36

Comments and Extended Discussion

The issues we have run into this week, though they have been slowing down our progress, have helped us gain a better sense of how the system works overall. Because the ECP machine was not giving us accurate results, we had to look closely at each component, try replacing different components with other ones available in lab to see if our results change, in order to determine the source of the issue we are observing. Finally, when we realized that the ECP machine is not properly generating a constant torque which is what we would need for accurate measurements, we had to look into other ways of generating a constant torque, and came up with the idea of using the quadcopter motors themselves to generate a constant torque about the different axes of rotation. This has us look closely into the process for measuring and determining the thrust constant of the quadcopter, a step we would have had to complete at some point in the near future anyways, and determine how we can control the motors from a raw PWM signal chosen and tweaked from a function generator for easy testing. Thus, although the issues we are experiencing with the equipment so far are a bit frustrating, we have learned a lot because of them.

Plans for Coming Week

- Measure the thrust constant, if we can get the new motors - Brendan, Tara, Andy
  - This is a big IF, because we need the new motors before we actually measure the thrust
  - Use the setup we defined under the “Past Week Accomplishments” subtitle
  - Continue to update the documentation on this process
  - Determine how we can use the thrust constant to provide a constant torque for measuring the moment of inertia of the quad about its three axes of rotation
- Modify Bluetooth proof of concept to log VRPN data better - Kris
- Continue mapping IMU dataflow - Joe, Brendan, Tara, Andy, David
  - Determine how the data, once brought in from the sensors, is processed and manipulated in the code
  - Create Powerpoint slides representing this data flow
- Model the mapped IMU dataflow - Brendan, Tara, Andy
  - Use Simulink to model the data flow based on what the software is doing
  - Check if the model makes sense by testing it
  - Also mimic noise of the data

- Begin cleaning up wiring on the quad - Eric, Andy, Tara
- Learn how to download code to the quad - Joe
- Understand why quad doesn't pass logged data back to ground station properly. Connect with Joe Avey on this - Kris
- Generalize Bluetooth proof of concept for use with wifi sockets - Kris
- Eliminate as much cpp as possible from bluetooth/vrpn proof of concepts - Jake, Kris
- Finish making ESP8266 drop-in replacement - David
- Work with controls group to map data flow - David, Joe
- Update ground station utility to work with WiFI - David, Ground Station group

Summary of Weekly Advisor Meeting

In this meeting, we talked about the next steps for our sub-teams, including our current state of progress, the issues we have been having, how we went about fixing them, and our upcoming needs in the course of this project.

- Notes for Controls team:
    - Once we have a working controller:
        - Do not need the ECP to drive the quad anymore
        - Can remove the belt for that
    - When understanding the programming structure:
        - In addition to analyzing the data flow from the bottom up (from the sensors to the software) also go in the opposite direction
        - Continue to make the Powerpoint Diagram
    - Finding scripts to part ECP data
        - Can ping Joe Avey and ask him
        - Look at what Matt gave us
        - Can also reference Matt's thesis for important expressions/derivations
    - Updates from the team:
        - Measured yaw moment of inertia
        - Came up with a way to measure pitch moment of inertia
        - Clarified some points of confusion with Matt's thesis, such as the generic, fundamental, or according to Matt, the "magical" frame of reference
- Notes for the Ground Station team:
    - Dr. Jones wants to see more emails from them
    - Want to prevent the following:
        - A disconnect between Dr. Jones's vision of the direction of the ground station and the ground station team's implementation
        - Realizing much later that we've written bad code
        - Having future teams not be able to understand our code
    - Want to have something scalable for the future, but first need to know the goals and constraints before doing so
- Notes from Team Leader (David)
    - Need to make a project plan for the course assignment:
        - Should include:
            - Everything we plan to get done this semester, in detail
            - Some less-detailed ideas of what we want to get done next semester

            - Our objectives, in short
        - Next steps:
            - Consider meeting up as a team
            - Look at previous year's project plans
            - Main goal is to "get your feet wet"
    - Also need to keep track of these 491 documents/reports that will be due in the future
        - These reports are easy to write the "first 95%" of, it's the last 5% that takes a lot of time
        - Since we will write many drafts/updates of these reports, don't worry about the last 5% for the first few reports, we will make sure the last ones are polished
        - Follow last year's documents as good examples
- Other notes:
    - When bringing in last year's code into our project, just bring over what you need when you need it, though this is probably most of it
    - Need to look into logging things
        - Should be doing that soon
        - Should be well-documented, find this and read it
- Summary of our work this past week
    - Ground Station Sub-team:
        - Worked on Ground Station stuff with Kris and Jake
        - Looked into bluetooth and how it works:
            - Current implementation is "clunky"
            - Using BlueZ, uses a socket interface which is familiar
        - Successfully tested communication link
            - Between laptops
            - Between ground station and quadcopter
            - Library is strictly bluetooth
            - Possible concern brought up by Elia: Bluetooth with TCP might cause problems in the future
    - Controls Sub-team:
        - Understanding the IMU data
            - Considering scaling and other simple manipulations when brought in from the sensors
            - Trying to understand the difference between the raw data and the converted data, and what we need to consider when looking at the data flow, since we are only modeling the real data:
                - Should consider angles as a starting point for our input to our model of the system
                - Though, may also want to model the filter itself as well, and possibly the noise (don't just run the model with perfect data)