Clay Freeman & Jacob Lee

ME457 : Mechatronics

Dr. Fields – Fall 2018

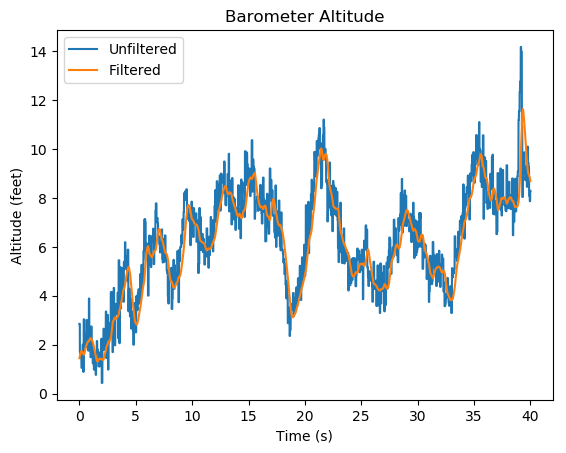
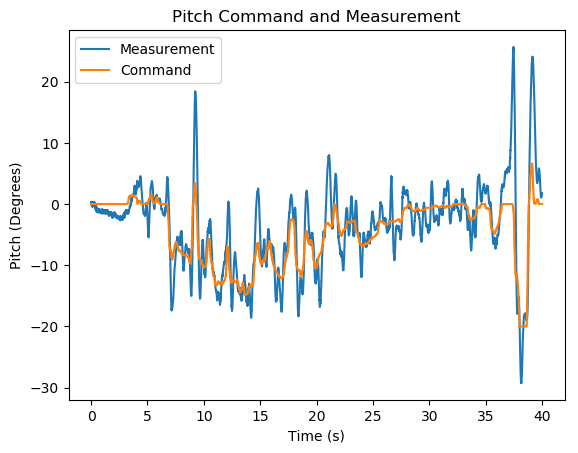
Checkpoint 4

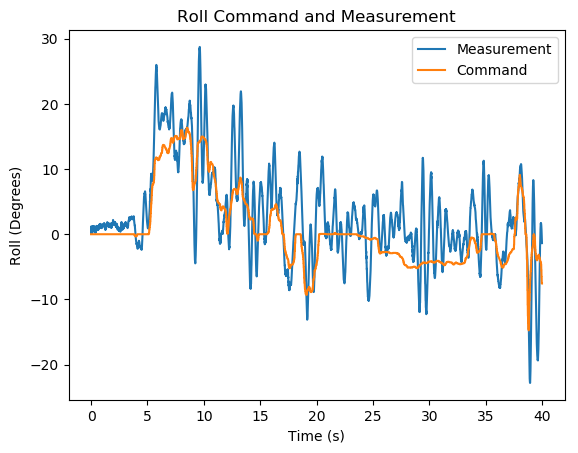
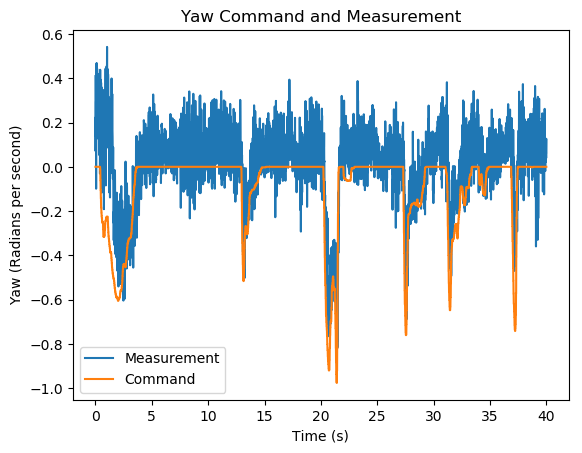
9 November 2018

# Checkpoint 4: Free Flight

## Section 1: Data Analysis

The figures below show the filtered and unfiltered altitude estimation from the barometer. Pitch, yaw, and roll measurements are plotted with the commands. High frequency noise was filtered using a 2nd order lowpass Butterworth filter with a sampling frequency of 100 Hz and a cutoff frequency of 1Hz.





During the first five seconds of the plotted flight time, the motors were spinning but the quad was still hovering along the ground as adjustments were made in the yaw to get an optimal takeoff orientation. Takeoff occurred at the 5 second mark where a majority of the control is focused on the pitch and roll. Using this controller configuration, both pitch and roll had significant overshoot, but this did not play as big a role as disturbance rejection in the maneuverability of the vehicle. Yaw is difficult to adjust in free flight because it is attached to the same control stick as the throttle which is very sensitive and critical to the success of the flight. The yaw has a slight drift over time but increasing integral gain to the yaw controller makes the system unstable.

The controller is barely flyable. The gains that meet the overshoot and settling time requirements from the previous two checkpoints did not translate into free flight well. The stand used to calibrate the controller contributed an input to the system that could not be easily separated from the underlying gains necessary for free flight and it was difficult to diagnose controller problems when the quad crashed before providing useful feedback. Another factor that played a big role in the overall success of the drone is that Quad 3 consistently had problems with the IMU and ESCs which made gain tuning impossible. The actual flight test for this checkpoint was performed using a different quad which still needed more adjustments and lots of practice to just keep the drone airborne.



## Section 2: Final Gains

At this point, the PID gains are receiving regular adjustments to try to achieve a more stable system. Because pitch and roll operate on the same principle, these controls can use the same gains in the code. At the time of this checkpoint, Kp = 0.0038, Ki = 0, and Kd = 0.00073. Integral gain in the pitch and roll controls never created a desirable outcome in free flight. The added instability would always overwhelm the dampening contribution from the proportional and derivative components of the controller. So far, PD control has been the most successful controlling method in terms of fly-ability.

A yaw controller is working to balance a system that is naturally stable. These systems are known to have a better response to a PI controller than the PID control since the derivative slows the rise time and does not help with steady state error. In practice, the residual steady state error in this parameter was minimized around 0.1 rad/s using Kp\_yaw = 0.26, Ki\_yaw = 0.0032, and Kd\_yaw = 0. Increased gains in the integral component would cause a fight with the pitch and roll control systems which could sometimes create a runaway system mid-flight. A balance needed to be achieved between these gain values and the gains that drive the pitch and roll, and it is an ongoing process to get this parameter to work well with the other two.

## Section 3: Lessons Learned

The drone testing lab has been very challenging over the course of this semester. The raspberry pi was a new toy to me at the beginning of the summer, so I was just beginning to learn about using SSH to communicate with the device over the WIFI network. Preparing for this class was one of the reasons I began to pursue learning to program in python, so there was a small language barrier at the beginning. -Clay

Hardware is one of the areas that we do not get enough experience with in school. The motors and speed controllers were completely new, and it was a great experience to get to work with the small electronics. The lab helped us get a better sense for how some of these drones work and to get some hands-on experience working with hardware. This class was very useful with bridging the gap between the theory and a real application. The most beneficial lessons in the course came from the troubleshooting that we learned as a part of the regular struggles during lab time. Quad 3 had extra problems beyond the scope of the class that made it even more difficult to properly tune the controller with the learning curve of trying to pilot a drone.

Electronics and hardware is only useful if it produces meaningful information for the controller. Filtering has a variety of applications and can be a very useful tool for interfacing different devices and smoothing a data set. While we learned a variety of numeric integration or differentiation equations from numerical methods last semester, each method has an appropriate application. Maximizing accuracy and minimizing computation error is not always the best option. For the state estimation techniques, the controller used simpler methods which yield a sufficient solution that is cheap on computational resources. The flight controller is performing these computations at 100Hz so any calculation error is minor compared to the rest of the sources of error in the system.

Another useful technique we learned this semester is sensor fusion. Using a single instrument is not always sufficient to get an accurate estimate of a physical parameter. Combining and filtering multiple sensor inputs can generate more accurate information than a single sensor alone. This was useful for the accelerometer and gyroscope data.