The purpose of this assignment is to become comfortable collecting and processing signals from the UAV. The exercise is mostly academic: the measurements we are recording aren't calibrated, and the states we recreate from the signals are unstable approximations of the actual states.

1 Assumptions and constants

For this assignment, we are not dealing with the dynamics of the UAV beyond which states are coupled or decoupled, so our set of assumptions will be different for the assignment. We don't make any assumptions up front, but there will be assumptions made during analysis.

2 Collecting Data

To collect data from the UAV we use the sample Simulink model provided by the manufacturer and connect the sensor information to a To Workspace block. This will create a .mat file that can be unpacked for the sensor measurements.

We can filter and process the signals in the Simulink model, or later in Matlab. I chose to do it in Simulink because it will eventually need to be there for the UAV to run. The first step is to convert the data signals to doubles from singles because that is what is required for filter and integrator blocks. This is shown in Figure 1, with data lines labeled for each measurement.

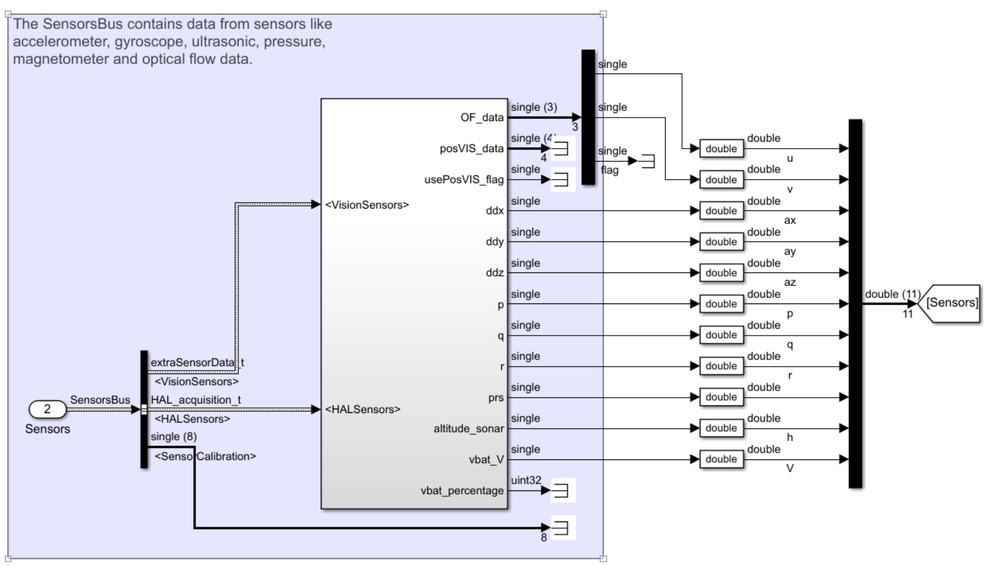

Figure 1: UAV Sensor Measurements Conversion

Next, we want to recreate the states not measured. Those states are $X, Y, w, \phi, \theta, \psi$. For each of those states, we have a direct measurement that is a derivative of it. Therefore, we can just integrate to get a reasonable, if unstable, approximation of the state. For example:

$$\dot{\phi} = p \tag{1}$$

$$\phi = \int_0^t p \; d\tau \tag{2}$$

In Simulink, this is achieved with an integrator block. In Matlab, we can use the trapz command instead for a discrete integration. All state outputs together are shown in Figure 2. The figure also shows all the outputs to the workspace from the UAV.

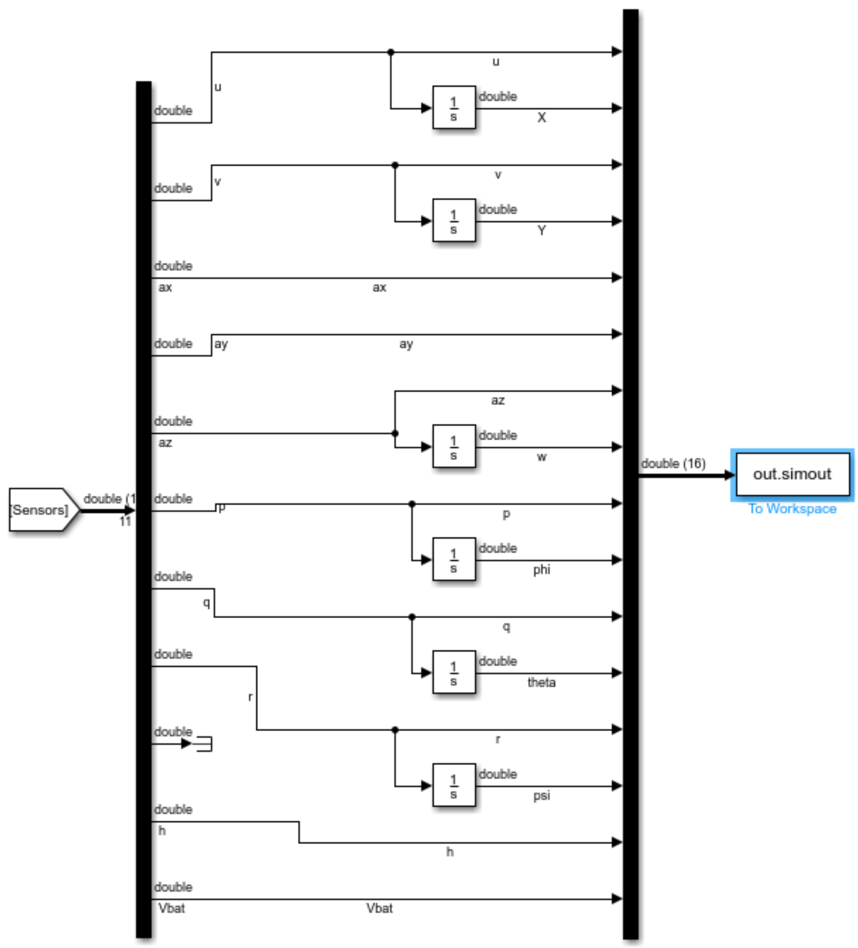

Figure 2: UAV State Reconstruction

Next, we want to filter the data with a low pass filter. This is for three reasons: first, the signals are digital, and therefore will have high frequencies in them because of the steps;

second, there will be noise in the measurements from uncertainty in the sensor; third, our sensors only read at 200 Hz, so any frequencies larger than 100 Hz will become distorted and aliased. Because of this, I decided to make my low pass filter have a cutoff frequency of 100.

$$T = \frac{100}{s + 100} \tag{3}$$

I implemented the filter on all my outputs, shown in Figure 3.

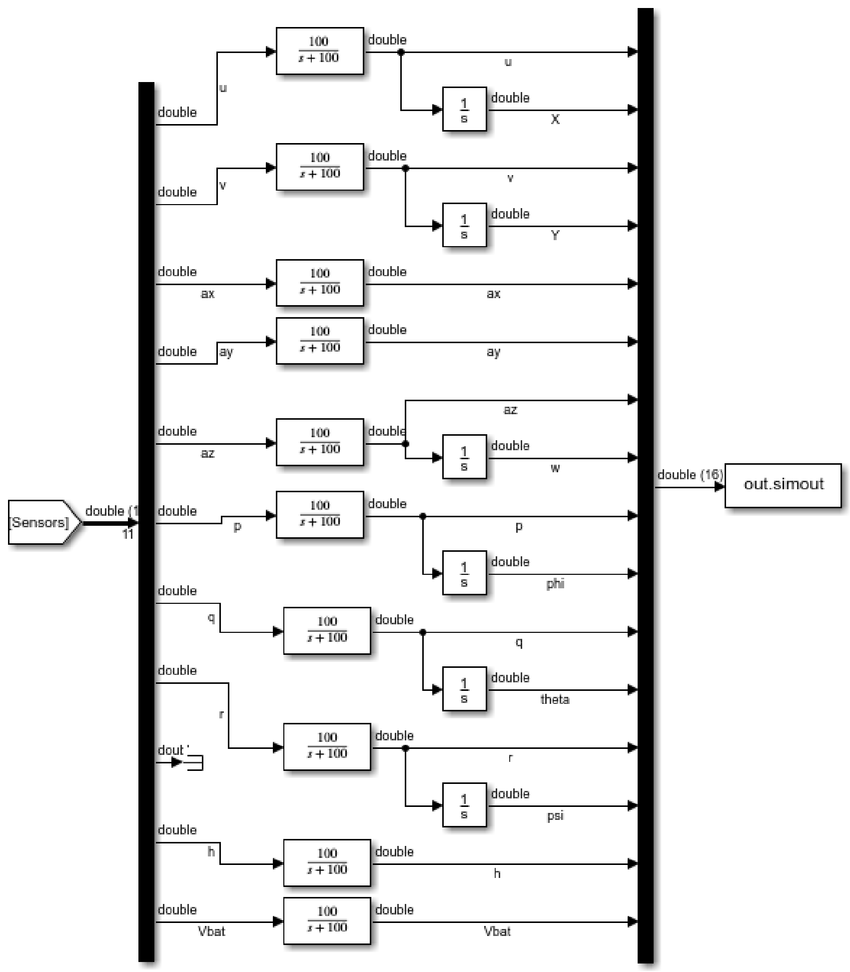

Figure 3: UAV Filtered Sensors and States

For comparison, I had the UAV output both the filtered and unfiltered states. To do this, I muxed together the two sets of data.

3 Analysis

For the analysis of the data, we will be looking at four visualizations: for the data itself, the filtered and unfiltered data in time domain and in frequency domain; and for the filter, comparing the Bode and Nyquist plots of the filter with no filter, as well as with the integrator-filter. The signals I will be analyzing were created by waving the UAV rapidly in small motions, like a personal fan.

3.1 Time Domain

I compared the filtered and time-domain signals visually. There were three types of signals I noticed from this. First, the signals from Optical Flow and Sonar appeared to have a much lower sampling frequency, less than 100 Hz, compared to the sampling frequency of the rest of the signals of 200 Hz. These signals also had much less noise compared to other measurements. An example is shown in Figure 4. The filter simply increased how long it took to reach the peaks, create an appearance similar to an RC circuit.

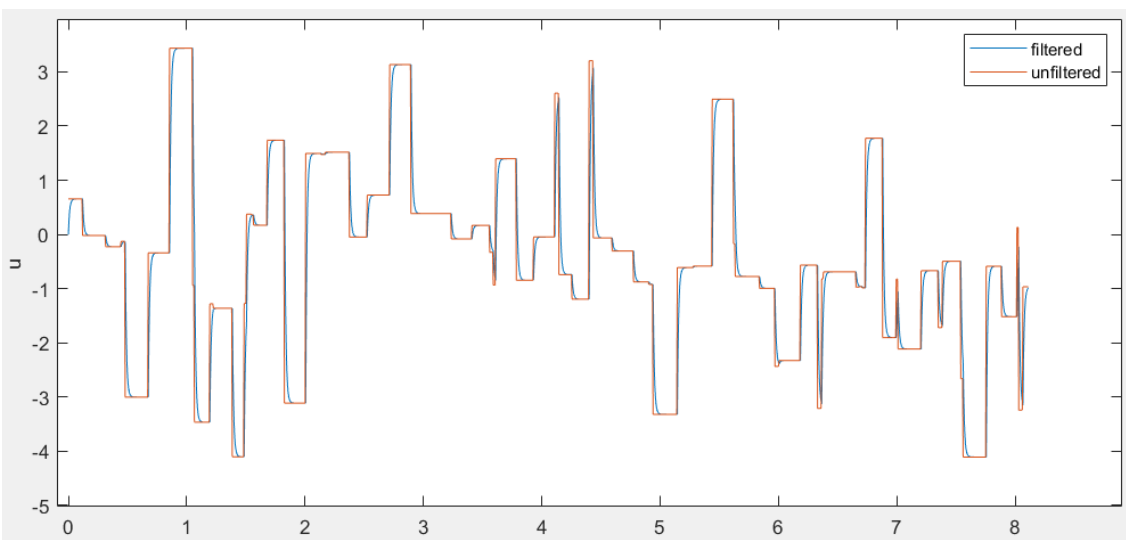

Figure 4: Optical Flow u signal, Filtered and Unfiltered

The next kind was the signals from the IMU, angular velocity and acceleration. These signals were slightly noisy, with some sharp peaks. The filter just reduced the size of the sharp peaks and rounded them down, but otherwise looked largely the same.

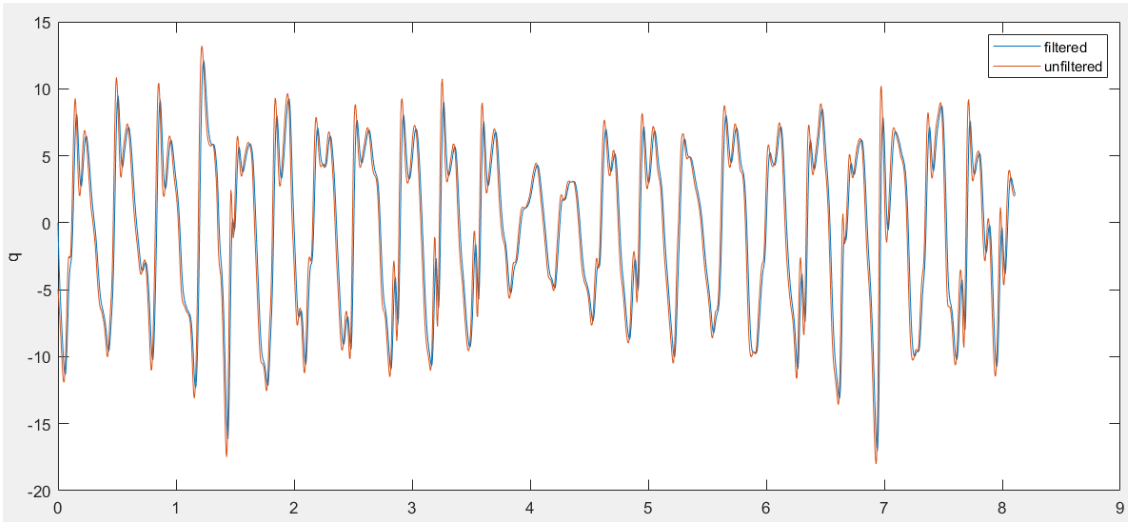

Figure 5: IMU signal, q, Filtered and Unfiltered

Lastly, there were the integrated signals. These looked very smooth. This makes sense since there is is an s^2 on the bottom of the transfer function making high frequencies very damped. An example is shown in Figure 6.

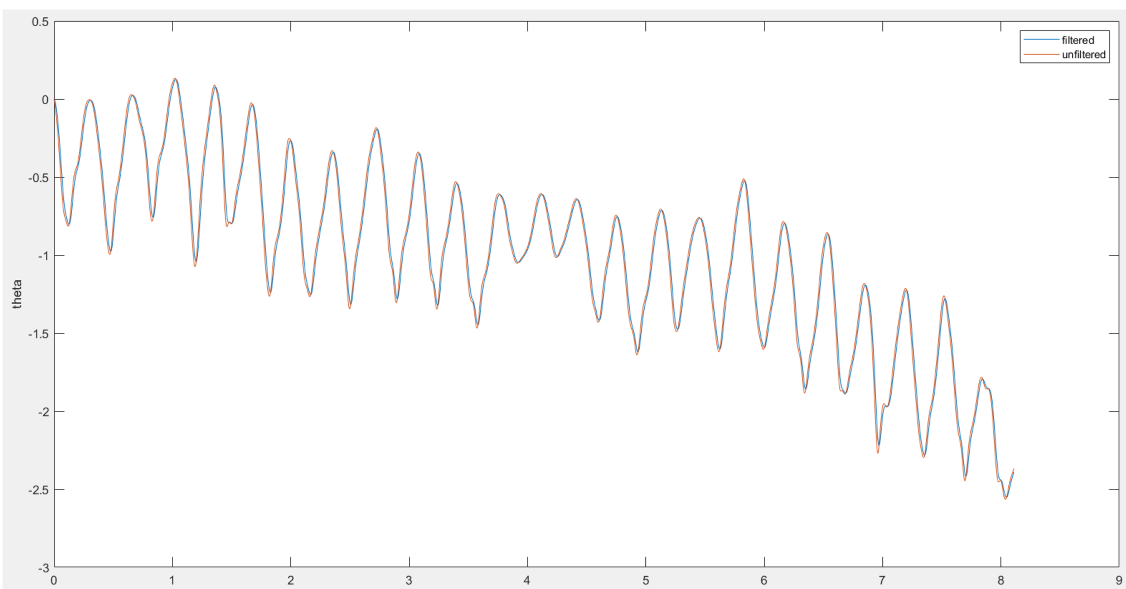

Figure 6: Integrated Signal, θ, Filtered and Unfiltered

These signals also exhibited drift, since the IMU signals aren't calibrated, so there is a bias which adds up in integration.

3.2 Frequency Domain

As expected, the Fourier plots of the signals looked very similar filtered and unfiltered. There started to be a slight damping of the filtered signals at around 15 Hz. There was not very much high frequency noise though, so both plots matched at high frequencies as well. This shows that a filter was not necessary for the data analyzed with an FFT, at least for with this set of data. An example is shown in Figure 7.

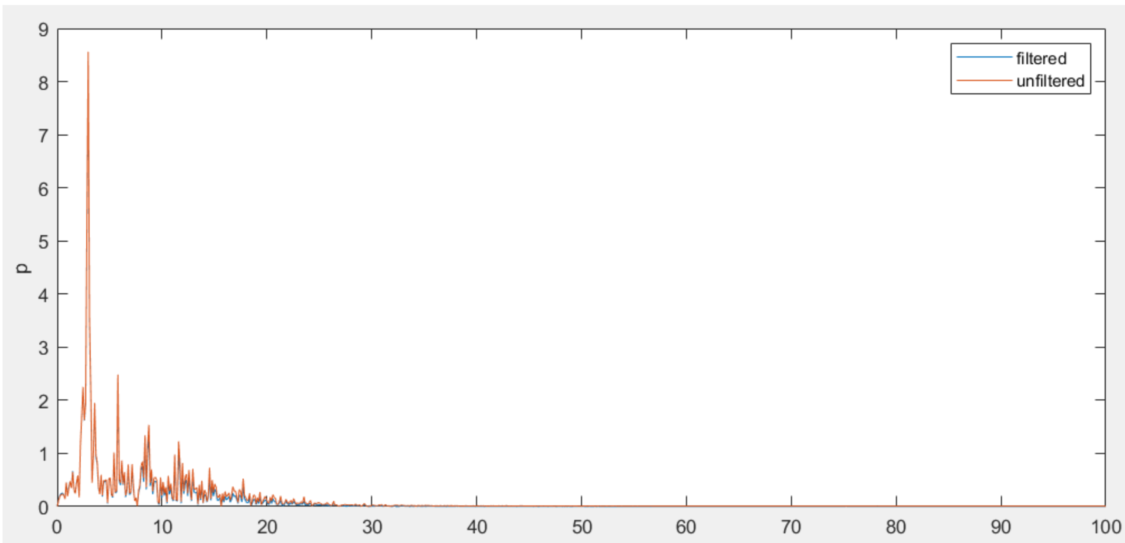

Figure 7: p, Frequency Domain, Filtered and Unfiltered

Additionally, we can see from the plot that the rate at which I was shaking the UAV is approximately 3 Hz.

3.3 Bode Plot

The bode plot of the lack of filter should be just a flat line at 0 dB with no phase delay. The filter should start decreasing at 20 dB per decade at 100 Hz, and the integrated+filtered signal should decrease at 20 dB per decade starting at 1Hz, and after 100 Hz, at 40 dB per decade with a 180^o phase shift. This is what happened, shown in figure 8.

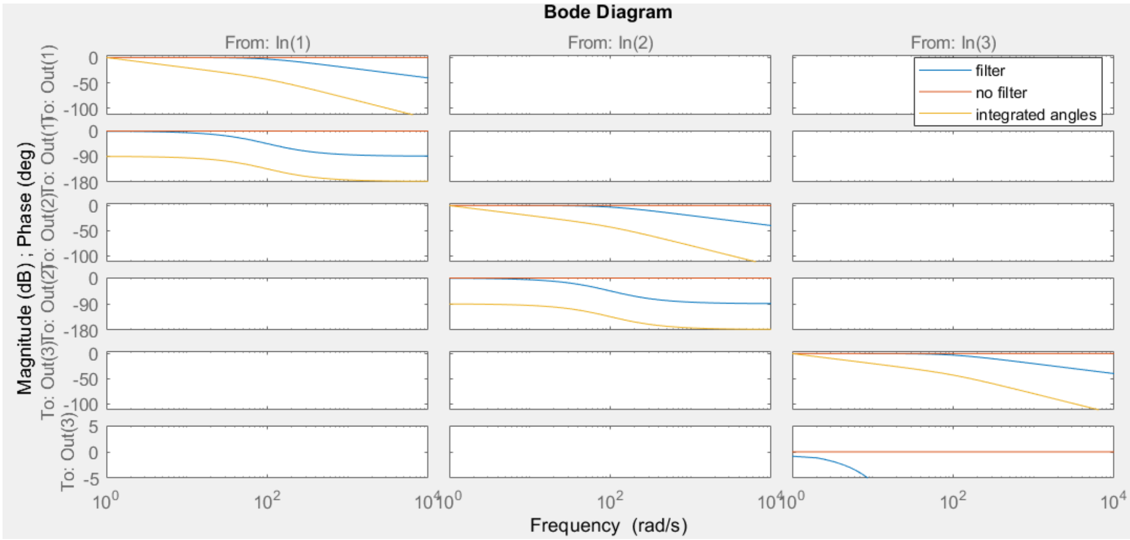

Figure 8: Bode Plot of Filters

3.4 Nyquist Plot

Lastly, we look at the filtered and unfiltered Nyquist Plots. We expect the no-filter to be constant at 1 with no phase regardless of frequency, just a dot. The low-pass filter should be a semi circle from 1 to 0 with -90 phase; and the integrator+filter should start at -90 phase with infinite gain at 0 frequency and 0 gain at infinite frequency. The plots matched this, shown in Figure 9.

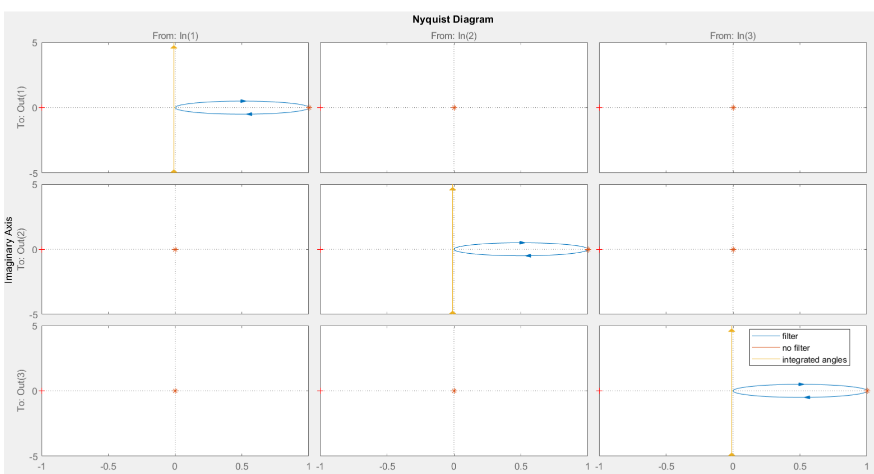

Figure 9: Nyquist Plot of filters

4 Conclusion

The signals on their own do not have large noise, relative to the measured values, and filtering for most states is largely unnecessary. However, we do need to adjust for the bias in the signals by calibrating.