ESE 650, SPRING 2021

HOMEWORK 2

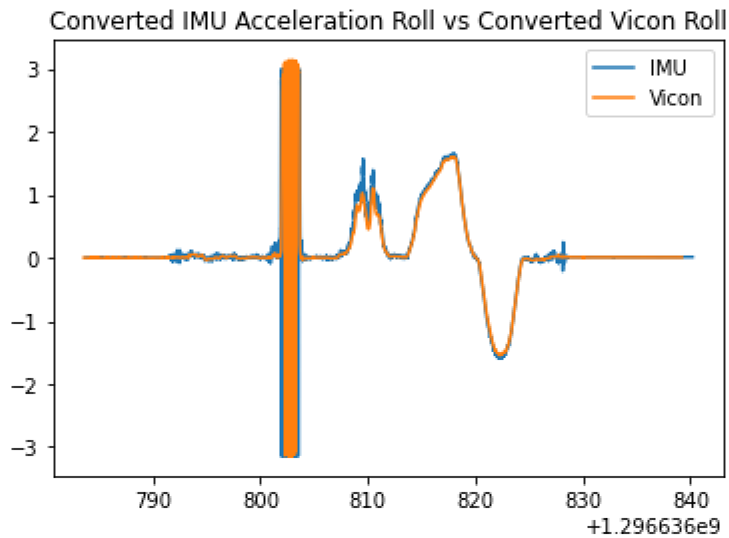
IAN MACDONALD [MACIAN@SEAS.UPENN.EDU],
COLLABORATORS: VASHIST VALSARAJ, ANTHONY NGUYEN

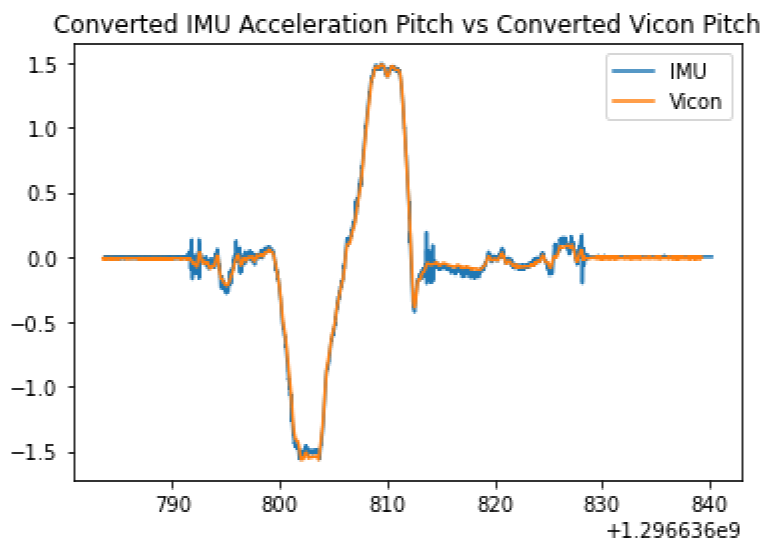
Solution 1 (Time spent: 0 Hours)**.** -

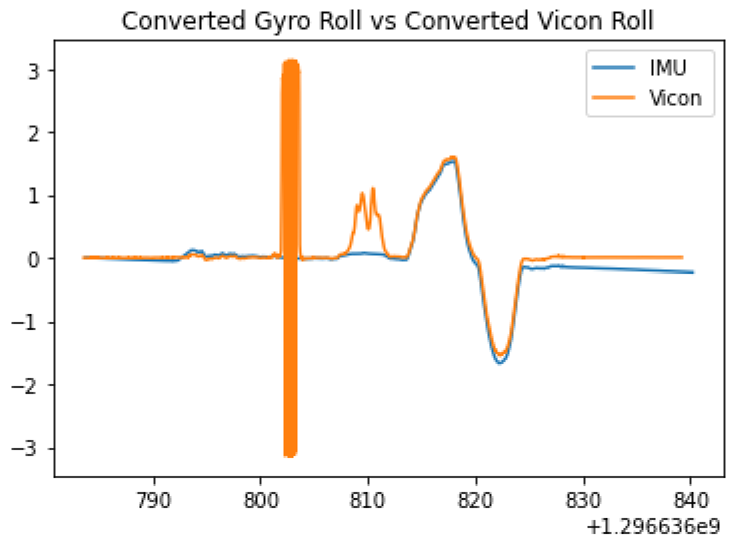
Unfortunately, I did not have enough time to look at problem one. Started with problem 2 since that's where the bulk of the homework was, but ended up spending days on that part and couldn't even try this problem.

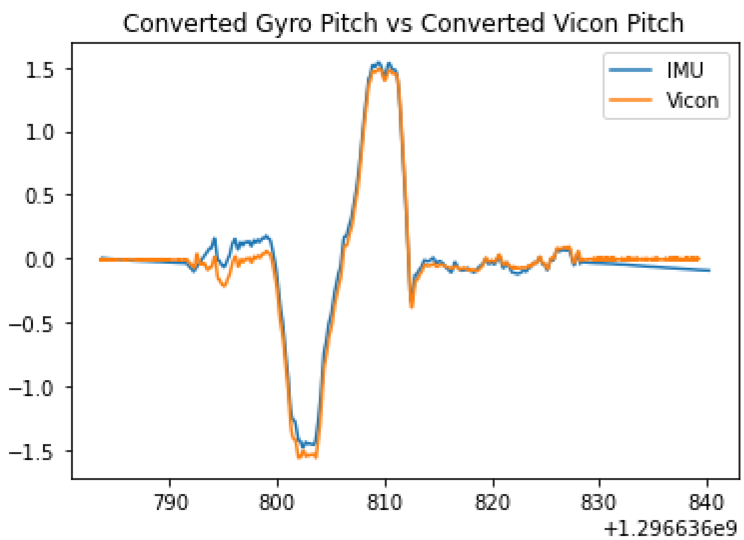
Solution 2 (Time spent: 34 hours)**.** -
b) Sensor Calibration

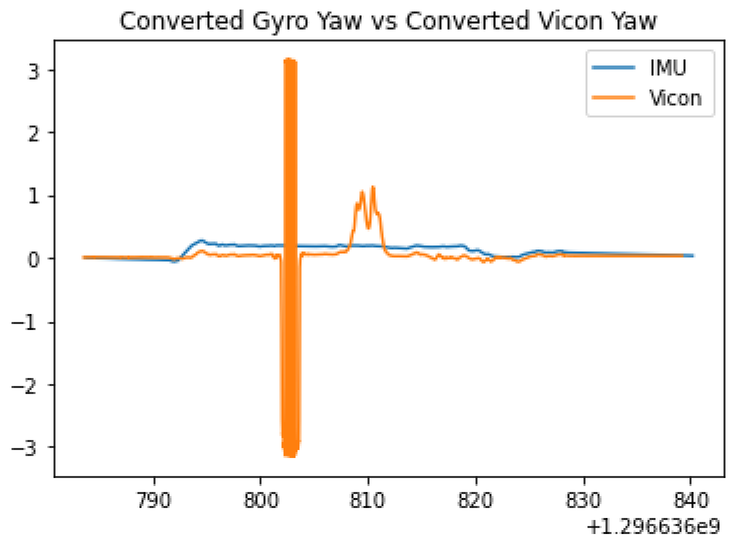
To solve for the bias and sensitivity of the gyroscope and accelerometer, I used the formula $value = (raw - bias)(\frac{3300}{sensitivity})$. Since the gyroscope and accelerometer values do not change (or change very slightly), we can assume the sensor is not moving or is remaining relatively still, and thus remains in it's initial orientation for the first time stamp. Therefore, we can assume that the roll, pitch, and yaw of the sensor should read (0,0,0). We can also assume that the only force acting on this sensor is the gravitational force, and since the orientation remains unchanged the accelerometer should read (0,0,9.81) for the x, y, and z axes respectively.
The estimated biases for the gyroscope data can be found by taking the initial gyroscope readings and plugging them into the above formula as $0 = (raw - bias)(\frac{3300}{sensitivity})$. The only way for this value to equal 0 is when $(raw - bias)$ is equivalent to 0, which allows us to estimate the bias. The estimated sensitivity for the gyroscope is then found by taking another point, plugging in the raw value and bias, and solving it against the given vicon value.
The estimated biases for the accelerometer are slightly more complicated. The same method can be used to solve for the x and y accelerations, but since the z axis acceleration does not read 0 initially, it cannot be solved in this way. To estimate the bias of the z axis, I averaged out the biases of the x and y axes, since the biases all tended to be pretty similar to each other. Then, the corresponding sensitivity was solved for by plugging in the estimated z axis bias into the equation and solving $9.81 = (raw - bias)(\frac{3300}{sensitivity})$ using the initial z acceleration point.
Finally, as mentioned above, all of these are estimates of the true bias/sensitivity. Once I found these estimates, they were fine tuned by adjusting them manually and observing the output plots below, until they approximated the true vicon data with enough accuracy. The final estimates I decided on for the data were:
Gyroscope Biases (Roll, Pitch, Yaw): 511, 501, 506
Gyroscope Sensitivity: 33
Accelerometer Biases (x, y, z): 370, 374, 375.7
Accelerometer Sensitivity: 115
Which yielded the following graphs:


Converted IMU Acceleration Roll vs Converted Vicon Roll
IMU
Vicon
3
2
1
0
−1
−2
−3
790
800
810
820
830
840
+1.296636e9


Converted IMU Acceleration Pitch vs Converted Vicon Pitch
IMU
Vicon
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
790
800
810
820
830
840
+1.296636e9


Converted Gyro Roll vs Converted Vicon Roll
IMU
Vicon
3
2
1
0
−1
−2
−3
790
800
810
820
830
840
+1.296636e9

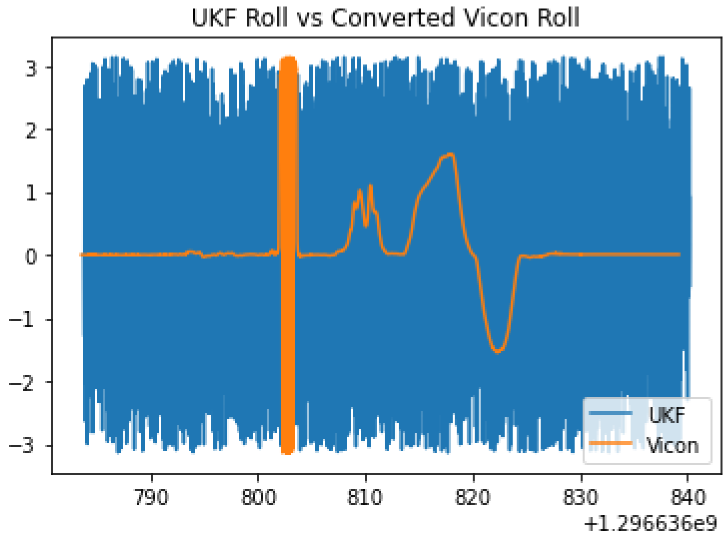

Converted Gyro Pitch vs Converted Vicon Pitch
IMU
Vicon
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
790
800
810
820
830
840
+1.296636e9

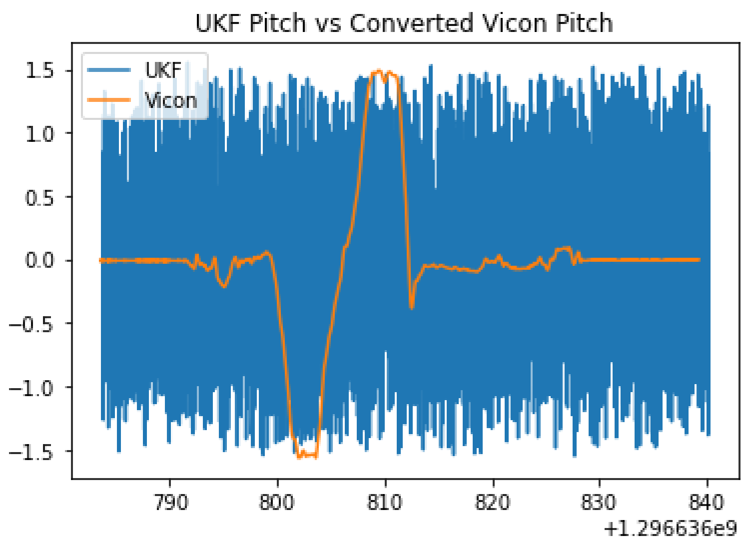


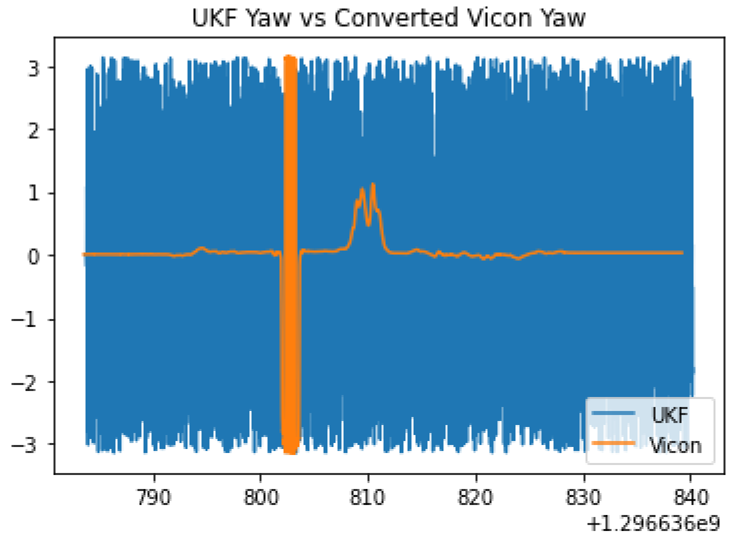
The above graphs looked pretty good, so I proceeded to the filter implementation step.

e) UKF Implementation/Analysis/Debugging

For the UKF Implementation, I tried to follow the paper as closely as I could, but could not get the UKF to work correctly. I tried everything I could think of and spent a lot of time working on it, but even after all the hours I could not figure out what was wrong. I went over the code multiple times, made sure my function implementations matched the calculations performed in the paper, checked the variables over to make sure they were in the correct places, printed out and graphed outputs to see if I could spot anything wrong, and tried a bunch of different P, Q, and R combinations, all to no avail. I even attended office hours and walked through the code step by step with T.A. Yihang, and we could not figure out the problem. I also walked through the code with the collaborators listed above, who were nice enough to give my code a look, also to no avail. The plots below show the final output for my UKF.


UKF Roll vs Converted Vicon Roll
UKF
Vicon
3
2
1
0
−1
−2
−3
790
800
810
820
830
840
+1.296636e9


UKF Pitch vs Converted Vicon Pitch
UKF
Vicon
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
790
800
810
820
830
840
+1.296636e9



Ultimately, it seemed like there was a bug in the filter itself given how much noise there was in the final plot. I put a lot of hours in, but could not figure out how to get the UKF to work well. I submitted the code on Gradescope, so I hope you will be able to take a closer look and maybe see what the problem was, and hopefully I can receive partial credit for my implementation.