

ESE 650, Learning in Robotics, Spring 2016: Project2

Yu-Cheng Lin

Introduction

This document describes my approaches to project 2: attitude estimation with accelerometer and gyroscope with Kalman filter. In all the plots of this document, red lines are always ground truth and blue lines are measured/estimated.

Calibration and obtaining ground truth

Before calibration, I must obtain ground truth from the vicon data. The orientation was easy to recover since the vicon data conveniently provided rotation matrices. Note that the rotation matrices actually describes the rotation from the body frame to inertial frame. Therefore, I must rotate the gravity vector $[009.81]^T$ with the transpose of the given vicon data.

$$\vec{g}_b = R_{vicon}^T \vec{g}_w$$

where subscript w denotes world frame and b denotes body frame.

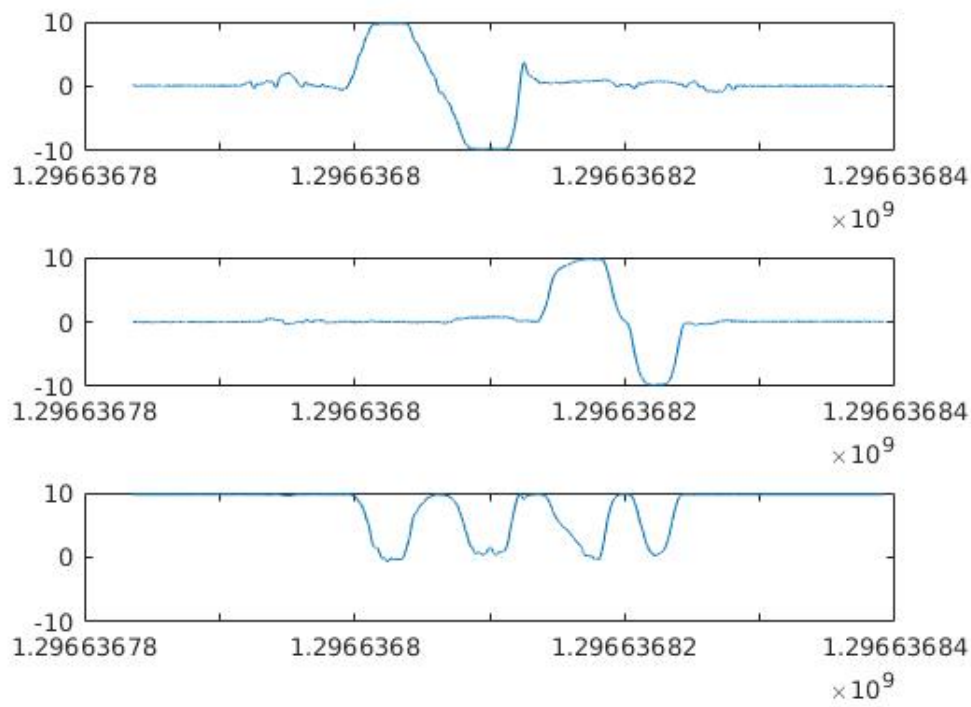
The angular velocity can be obtained by the numerical derivative of the orientation. However, I should not directly take the numerical derivative of a rotation matrix. Instead, I can describe a change in rotation R_e so that the rotation at time step t R_t can be followed by R_e to obtain the rotation at time step t+1.

$$R_t R_e = R_{t+1}$$

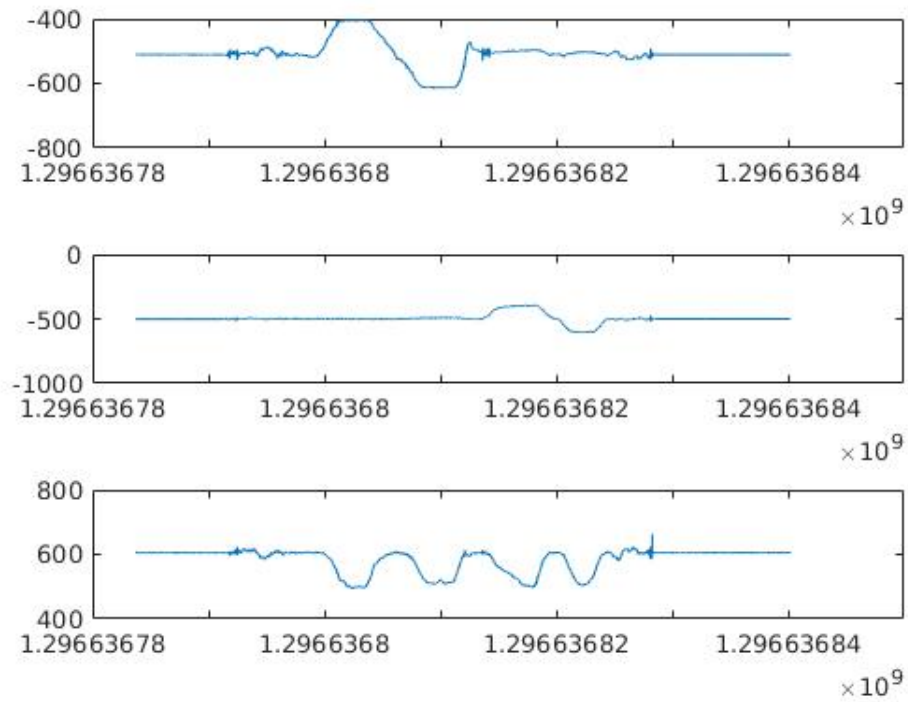
$$R_e = R_t^T R_{t+1}$$

Then, I can take this rotation R_e and convert it to a rotation vector $\vec{\theta}$ such that $e^{\vec{\theta}} = R_e$. Take $\vec{\theta}$ and then divide by the elapsed time. I can get an estimate of the angular velocity. However, such reading is very noisy. To compensate for the noise, I ran the numerical derivative through a 6th order Butterworth filter to obtain smooth results.

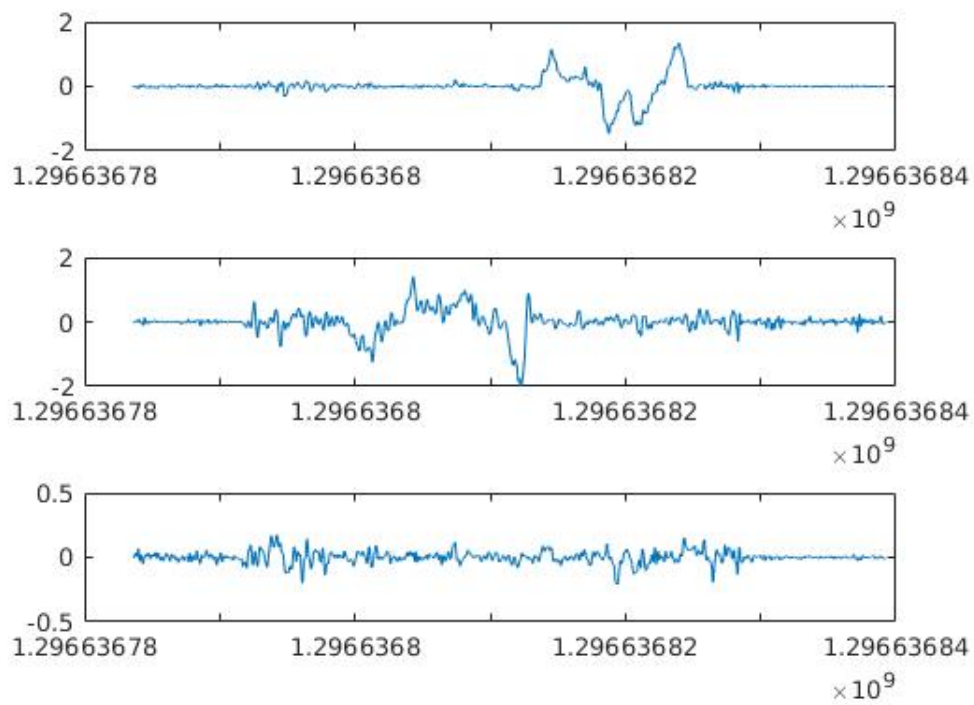
The calibration for the sensors were done by inspection instead of linear regression. This works better because matching time stamps of vicon and the Imu is difficult (some interpolation will be needed), and linear regression is susceptible to noise. Perhaps a L-2 penalized linear regression with the values by inspection as prior can work very well, but it is not the focus of the project. Following plots are the results I obtained from training set 1.



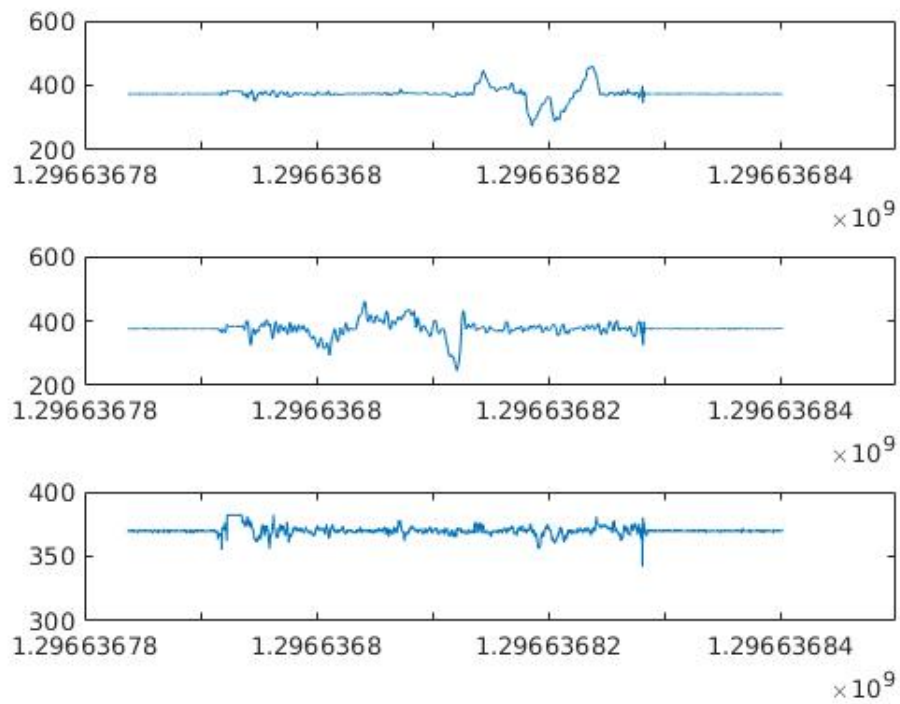
Expected gravity measurement given vicon data



measured gravity by IMU



Expected angular velocity measurement given vicon data



measured angular velocity by IMU

Obtaining Orientation with accelerometer

It is mathematically impossible to obtain the full orientation with only one vector measurement. However, from this vector measurement, I can deduct two degrees of freedom in a rotation. Let's work under the framework of euler angles to get more intuition.

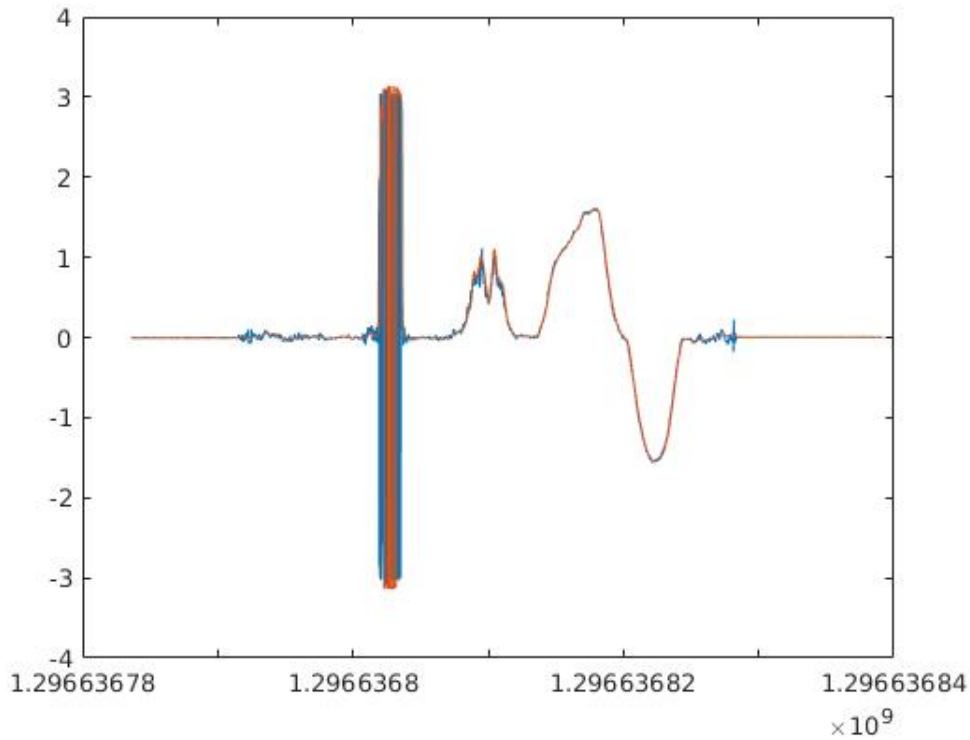
First, given a normalized gravity vector \vec{g} in the world frame with direction in $[001]^T$. and a rotation matrix R from body frame to world frame. We can see that $R^T \vec{g}$ picks out the third row of the rotation matrix R and set them equal to the measured gravity.

$$R_3 = [g_x \quad g_y \quad g_z]$$

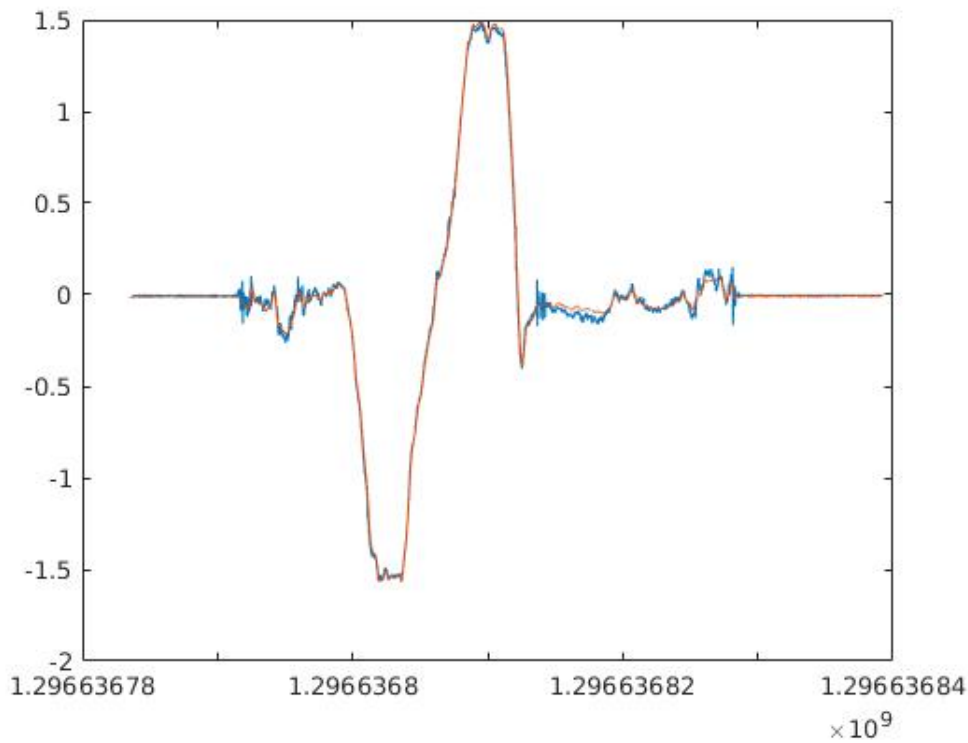
Now, let's write R in terms of euler angle with convention ZYX.

$$R_3 = \vec{g}^T = [\sin \theta \quad \cos \theta \sin \phi \quad \cos \theta \cos \phi]$$

Here, we have two equations and two unknowns. Note that the gravity vector only gives me two degrees of freedom because it describes a direction (The vector has unit norm). The acceleration alone does not give me any information about the Yaw.



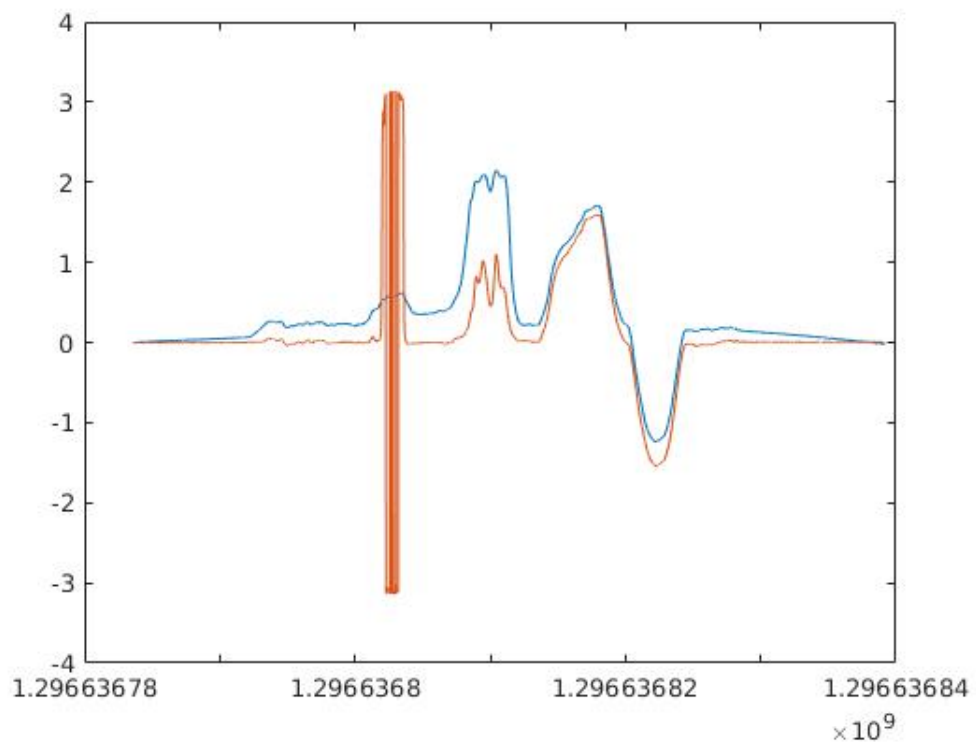
measured roll angle by accelerometer



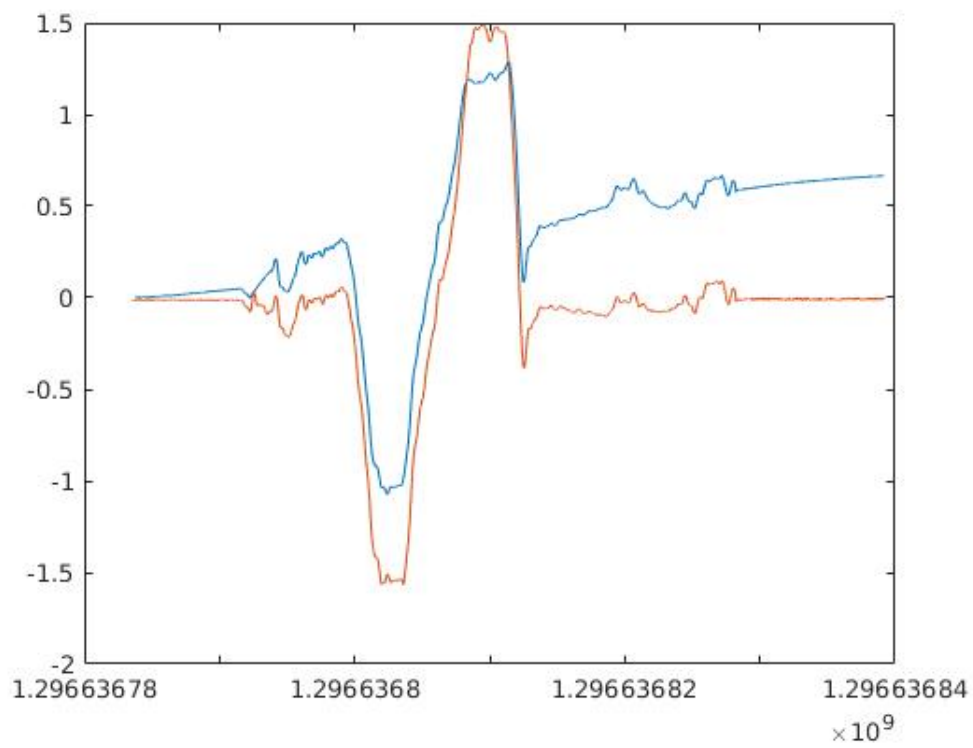
measured pitch angle by accelerometer

Obtaining Orientation with gyroscope

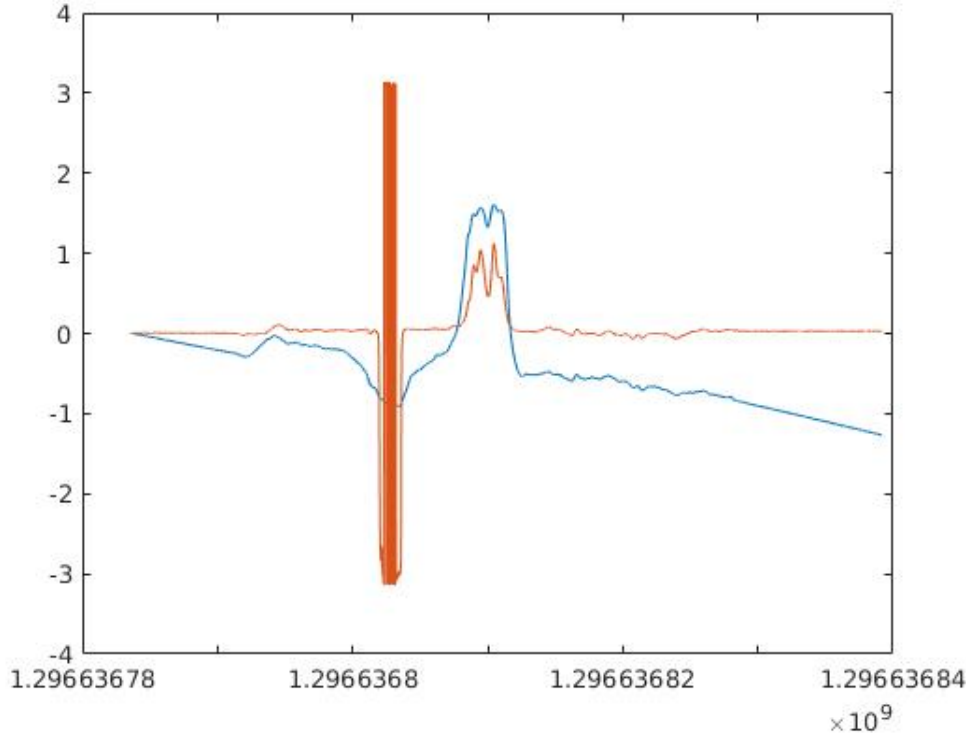
The gyroscope can be numerically integrated to get the change in orientation. We can write the change of rotation as $\omega\Delta t$ so it's a rotation vector $\vec{\theta}$. We can take the exponential of the anti-symmetric matrix form of this vector to obtain a differential rotation matrix ΔR . Post multiply this matrix to the current rotation matrix will give me the new rotation matrix. However, without knowledge of initial condition, this method can be very unreliable. For the purposes of this project, I will assume that the initial condition is at $R = I$.



measured roll angle by gyroscope



measured pitch angle by gyroscope



measured yaw angle by gyroscope

Unscented Kalman Filter

From the two plots above, it is clear that we have to somehow merge the data from the two sensors to obtain a better estimate. The accelerometer gives us very good orientation information except that the yaw is missing. The gyroscope can estimate all three angles, but it needs a known initial condition. In addition, drift of bias is a big problem for the gyroscope.

Some kind of probabilistic sensor fusion should be employed for this attitude estimation problem. Given the nonlinear nature of rotation, Unscented Kalman filter naturally suits this attitude estimation problem very well because of its ability to do nonlinear prediction and measurement. I followed the Kraft paper mainly but added in some small twists to improve my results.

Setup

As illustrated earlier in this report, the gyroscope is unreliable if my calibration of the bias is off or if the bias drifts. To compensate for this, on top of the 7 states from the Krafts paper, I added three more states $[b_x, b_y, b_z]$ to account for the bias in the gyroscope. This modifies my measurement model

$$H(\omega^-, b^-) = \omega_{ADC}^-$$

where

$$H(\omega^-, b^-) = K_\omega * \omega^- + b^-$$

Although there are changes to the process model and the covariance matrices. They are just dimension changes. Thus, I will skip the discussion of these components.

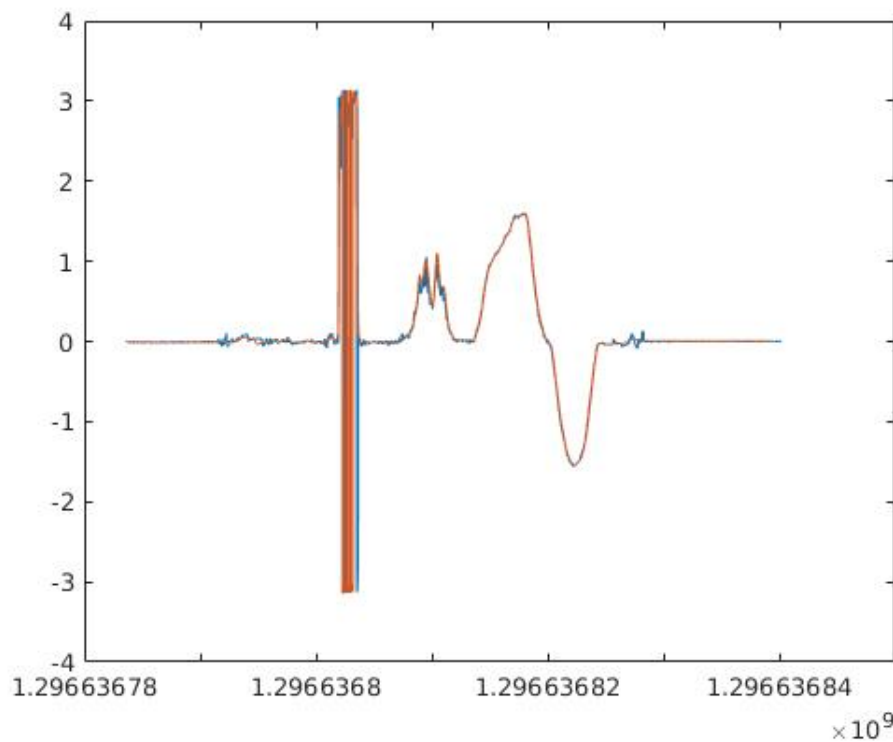
Tuning

Since the bias terms couples extra uncertainties to the angular velocity, blindly tuning R actually works better than measuring the covariance in the sensors. The rule of thumb is that since the data is fairly noisy, we would prefer very low covariance for the Q matrix to dampen the estimates.

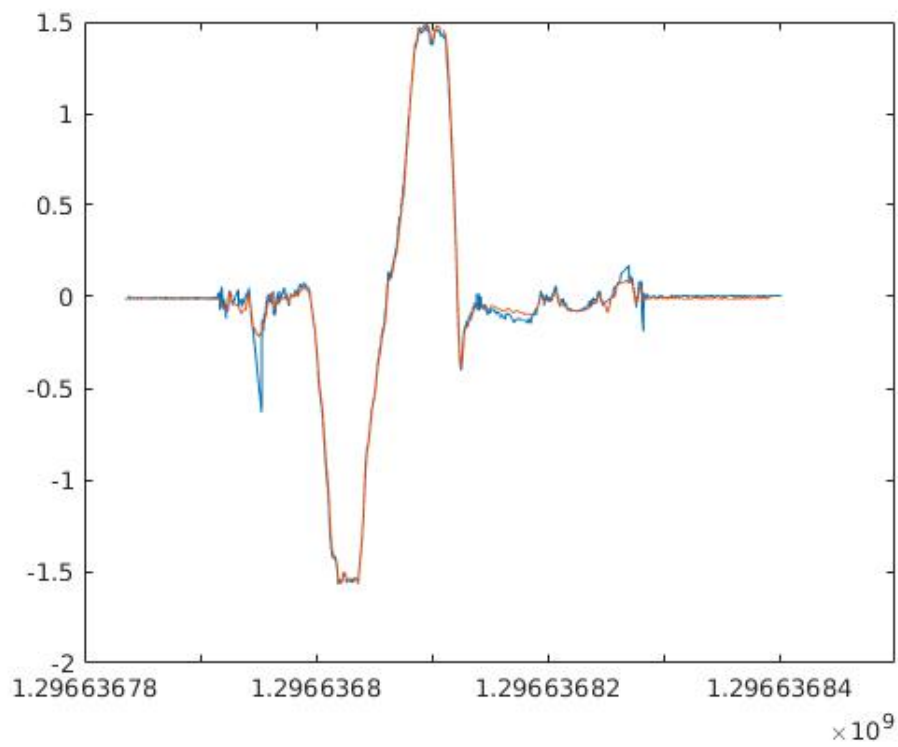
Hacks

Sometimes, undesired acceleration messes up the reading of the accelerometer. This can be detected by calculating the norm of the measured gravity. If the norm is out of a certain range, I will not take in the measurement at that time step and just use the dynamic model. However, the The a priori covariance P^- grows very fast because of the nonlinear mapping of the sigma points. If there are many consecutive points that have bad readings, P^- can grow too fast and make the Kalman filter unstable. In such cases, a simple hack is to return P_{k-1} , the covariance of the last step instead of the projected covariance. It's not mathematically correct, but it works reasonably well for this project.

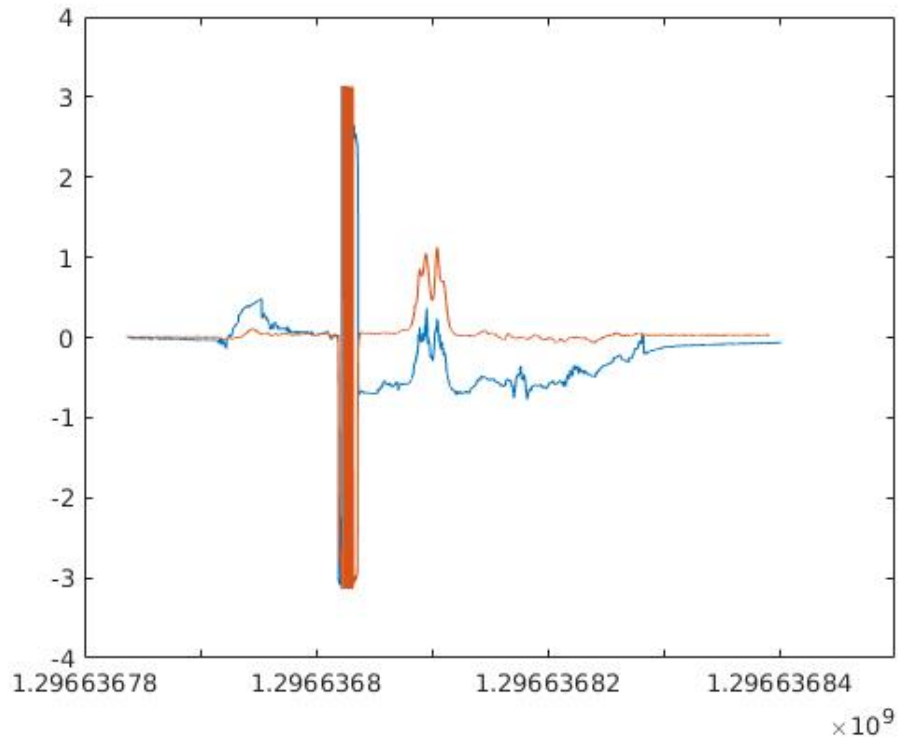
Training Set Results



roll angle of dataset1 estimated by my UKF



pitch angle of dataset1 estimated by my UKF



yaw angle of dataset1 estimated by my UKF

Image Stitching

Since my attitude estimate works well enough, I moved onto image stitching. My approach is simple. Define a camera frame which the x-axis projects forward, and z axis goes up. Associate the z direction with the row pixels and the y axis with column pixels. Define some arbitrary field of view γ and given image height h , width w I can calculate a vector in direction

$$\begin{bmatrix} \frac{h}{\tan \gamma/2} & \frac{w}{2} - p_i & \frac{h}{2} - p_j \end{bmatrix}$$

where p_i and p_j denotes pixel position in the image. Rotate this vector with estimated quaternion, find the corresponding spherical coordinate (omitting radius). Map the spherical angles to the rows and columns of a big image. I will get a panorama image.

