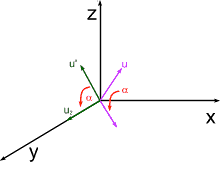
**Data Normalization Procedure:**

Because we did not restrict the orientation of the smartphone during our data collection, we ran into a situation where the data was not normalized between samples. To fix this, we measured the effect of gravity on each of the three axes of the smartphone and, using vector analysis and rotation matrices, modified the collected accelerometer readings into a more standard format.

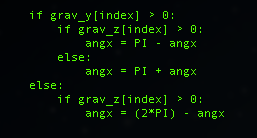
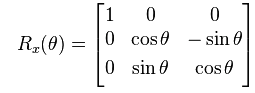


http://www.csee.umbc.edu/~rheingan/435/pages/res/gen-7.3Dtform-single-page-0.html

Our first step was to remove the effect of gravity from one of the smartphone’s axes. By performing a rotation along the x-axis, we could effectively remove the effect of gravity on the smartphone’s z-axis (see figure above). If we ignore the effect of gravity on our axis of rotation, we end up with a projection of gravity on one of the Cartesian planes. In this rotation, we have a projection of gravity on the yz-plane. Using this projection (u’), we can then use an inverse cosine function to calculate the angle between our gravity projection on the yz-plane and the xy-plane.

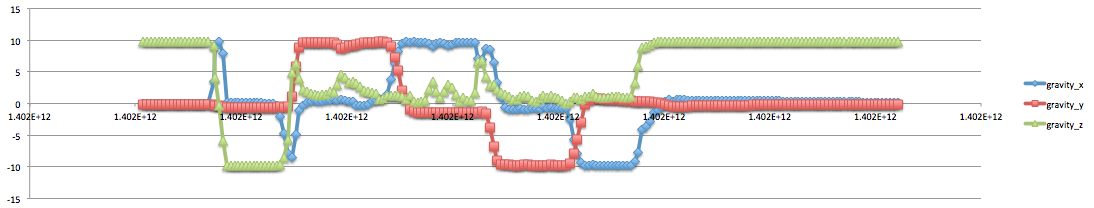
alpha = acos( |gravity on y-axis| / |gravity on yz-plane| )

Depending on which quadrant the gravity vector is located in the yz-plane, we modify this angle by adding or subtracting the calculated angle from PI or 2\*PI. (Code snippet below). Using this calculated angle, we can then build a rotation matrix for a rotation around the x-axis using the formula below:

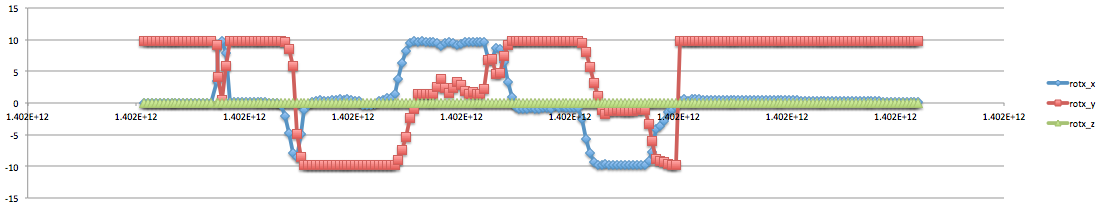
 

If we represent the readings for the x, y, and z axes of the smartphone in a 3x1 matrix, we can perform matrix multiplication to give us our accelerometer and gravity readings with no gravity present in the z-axis. Before and after values for gravity can be seen below.

Original:



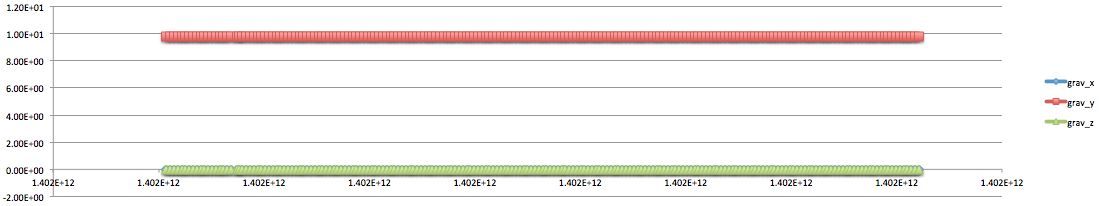
No gravity on z-axis:



Now that we have removed the effect of gravity from the z-axis in our data, we need to align our gravity vector along the phones y-axis. Using the same method as before with the new values calculated, we determine the angle between the gravity vector projection on the xy-plane (Note: after rotation, this is essentially the entire gravity vector since it is now aligned on the xy-plane after the previous rotation) and the y-axis.

beta = acos( |gravity on y-axis| / |gravity on xy-plane| )

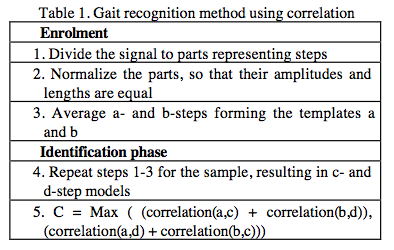
Like before, depending on which quadrant the gravity vector lies in may result in slight additions/subtractions from the calculated angle. This angle is then used to build our z-axis rotation matrix and then multiplied by the accelerometer and gravity readings to give us our accelerometer readings when the phone is in an upright position (ie. y-axis of smartphone is aligned with y-axis of person). Resulting gravity vector readings can be seen below:



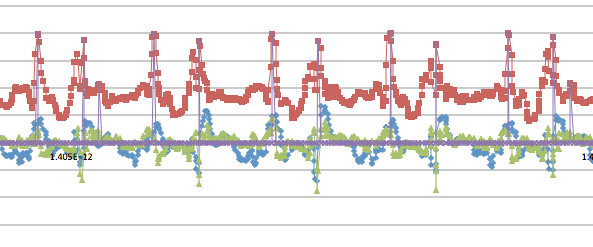
Although the gravity vector is now aligned with the phones negative y-axis, we are unable to align the phones x-axis with the forward movement of the person. We have a theory that principle component analysis of the x- and z- axis data points will give us the forward walking axis of the person but that has yet to be tested and verified.

**Method:**

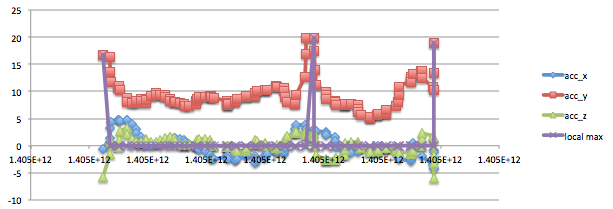
The method we are using to enroll and verify a person’s gait signature is mostly based on the work of Mäntyjärvi (2005). In particular, we look at the correlation method presented in the paper. With this method, we are to divide the signal into steps labeled “A” and “B”. Further details of the process can be seen in Table 1 below.



The steps are divided by identifying the local maximums on the y-axis data points. We find these local maximum points in the signal by identifying the maximum point over a window of 27 points. This means that in order to be identified as a local maximum point, its accelerometer reading must be greater than the previous 13 data points as well as the next 13 data points. This is done across the entire enrollment signal.



Although this method does produce some errors as seen in the sample signal above, the top and bottom 5%-10% of the steps in terms of length will be thrown out from the template creation. These local maximum points allow us to identify the steps and separate them accordingly.



Once the steps have been separated into the “A” and “B” steps and the top and bottom 5% of the signal in terms of length have been discarded, normalization of the signal will need to take place in terms of height and length. Dividing all points by the maximum point value of the step will normalize the height of the signal. This ensures that all points in the step are in the range of 0-1.