

*Department of Mechanical and Mechatronics Engineering*

MTE 546 - Lab 3 Report

**A Report Prepared For:**

Prof. Eugene Li

MTE 546

Multi-Sensor Data Fusion

**A Report Prepared By:**

Kapilan Satkunanathan (20694418)

Kevin Zhang (20659590)

Group #9

**Introduction**

Building upon the results gathered from lab experiments 1 and 2, we will attempt to fuse the data obtained from real and “virtual” sensors to determine the orientation of a phone. For the experiment, 3 different initial trajectories were chosen with two set time points where a 90 degree rotation will be simulated; the first to rotate to the vertical position, the second to return back to its original position. The 3 trajectories in question, based on the x and y axes, are (2, 1), (3, 1) and (10, 1) cm/s. The first and second rotations will occur at 3.5 and 12 seconds into the simulation, respectively. See Figure 1 below for a visual representation of the simulated motion.

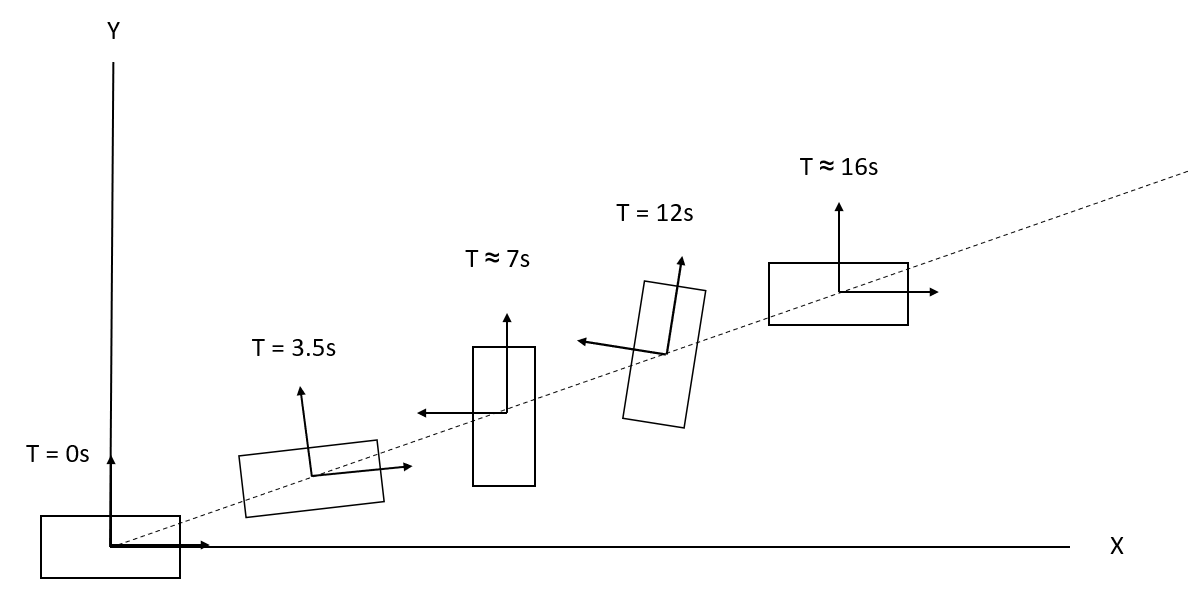


Figure Visual representation of the targeted trajectory, rotation included.

**Simulation**

The simulation consists of 3 main components, the “true” path data simulation, the Extended Kalman Filter, and a Fuzzy Logic algorithm for orientation correction. The first component, the path data simulation, simulates the expected path and rotation of the phone based on derived sensor models. The second component, the EKF, processes the path and sensor data to predict and correct the phone’s position and velocity. The third component, the FLA, applies fuzzy logic based on whether the phone is rotating or not to correct the estimated orientation of the phone.

**Simulated Motion Path**

In simulation, along a specified trajectory, a 90 degree rotation of a phone about the z-axis can be represented as a switch of velocity components in the x-y plane. For example, if the phone was traveling at a velocity of 2 cm/s in the x direction and 1 cm/s in the y direction, a 90 degree turn can be represented by a new velocity of 1cm/s in the x direction and 2cm/s in the y direction. But this instantaneous switch in velocity represents an instantaneous rotation, which in practice is not possible; to smoothen the rotation, trigonometric functions based upon the phone’s current orientation are used to bridge the change in velocity components. In order to complete this simulated smooth rotation, a virtual gyroscope was required in order to maintain a current estimation of the phone orientation.

The first trajectory is (2, 1) cm/s, the x and y axes respectively, the rotation functions for the x and y axes are as follows:

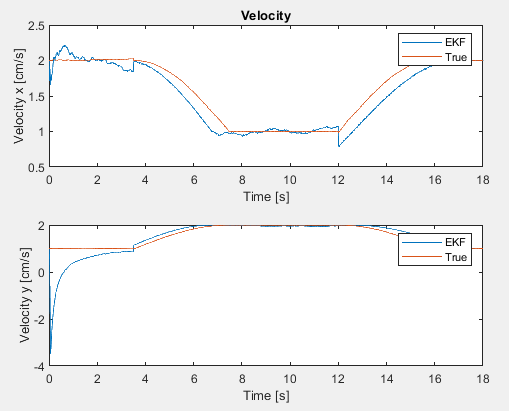
Figure 2 below maps the velocities of the “true” generated path and the corrected velocities generated from the EKF for the first trajectory.

Figure Mapping of the true and EKF corrected velocities for the first trajectory.

The second trajectory is (3, 1) cm/s, x and y axes respectively, the rotation functions for the x and y axes are as follows:

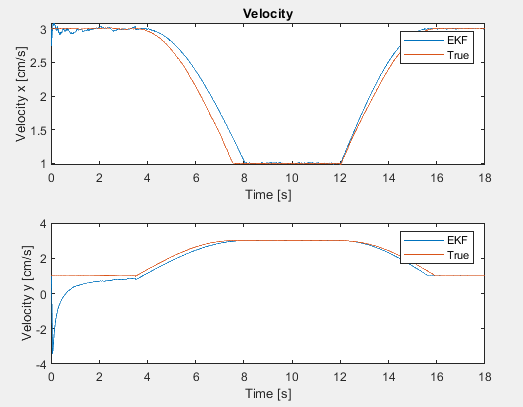
Figure 3 below maps the velocities of the “true” generated path and the corrected velocities generated from the EKF for the second trajectory.

Figure Mapping of the true and EKF corrected velocities for the second trajectory.

The third trajectory is (10, 1) cm/s, x and y axes respectively, the rotation functions for the x and y axes are as follows:

Figure 4 on the following page maps the velocities of the “true” generated path and the corrected velocities generated from the EKF for the third trajectory.

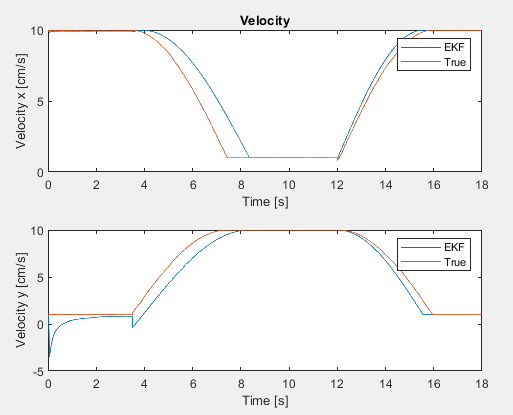
For the above equations, represents the current estimated orientation of the phone base around the z axis, where during the rotation, the gyroscope model developed in Lab 1, is activated and added to the orientation state at each time increment until the phone has reached 90 degrees. Examining the rotation functions, it can be seen that at 0 degrees, the trajectories are at their base speeds, for example the first trajectory is (2, 1) cm/s, while at 90 degrees, the trajectory flips the velocity components, for example the first trajectory becomes (1, 2) cm/s. The trigonometric functions allow for a smooth transition. Based upon the gyroscope model, one 90 degree rotation is completed in approximately 3.5 to 4 seconds.

Figure Mapping of the true and EKF corrected velocities for the second trajectory.

Based upon the designed simulation, the virtual gyroscope will have the greatest impact. Examining the simulated sensor models, the gyroscope within the virtual sensor would be more sensitive to noise as a linear model is used, whereas the accelerometer is using an inverse model. The two models can be seen below respectively.

Furthermore, the simulated rotation relies directly upon the estimated orientation stored by the gyroscope, whereas the accelerometer data is fed into an Extended Kalman Filter which will lower the data’s importance if deemed necessary by the algorithm. As such the data obtained from the gyroscope will be more susceptible to error. Examining the use of the virtual accelerometer and gyroscope, it can be seen that they are used in tandem, but no correlation exists between the two.

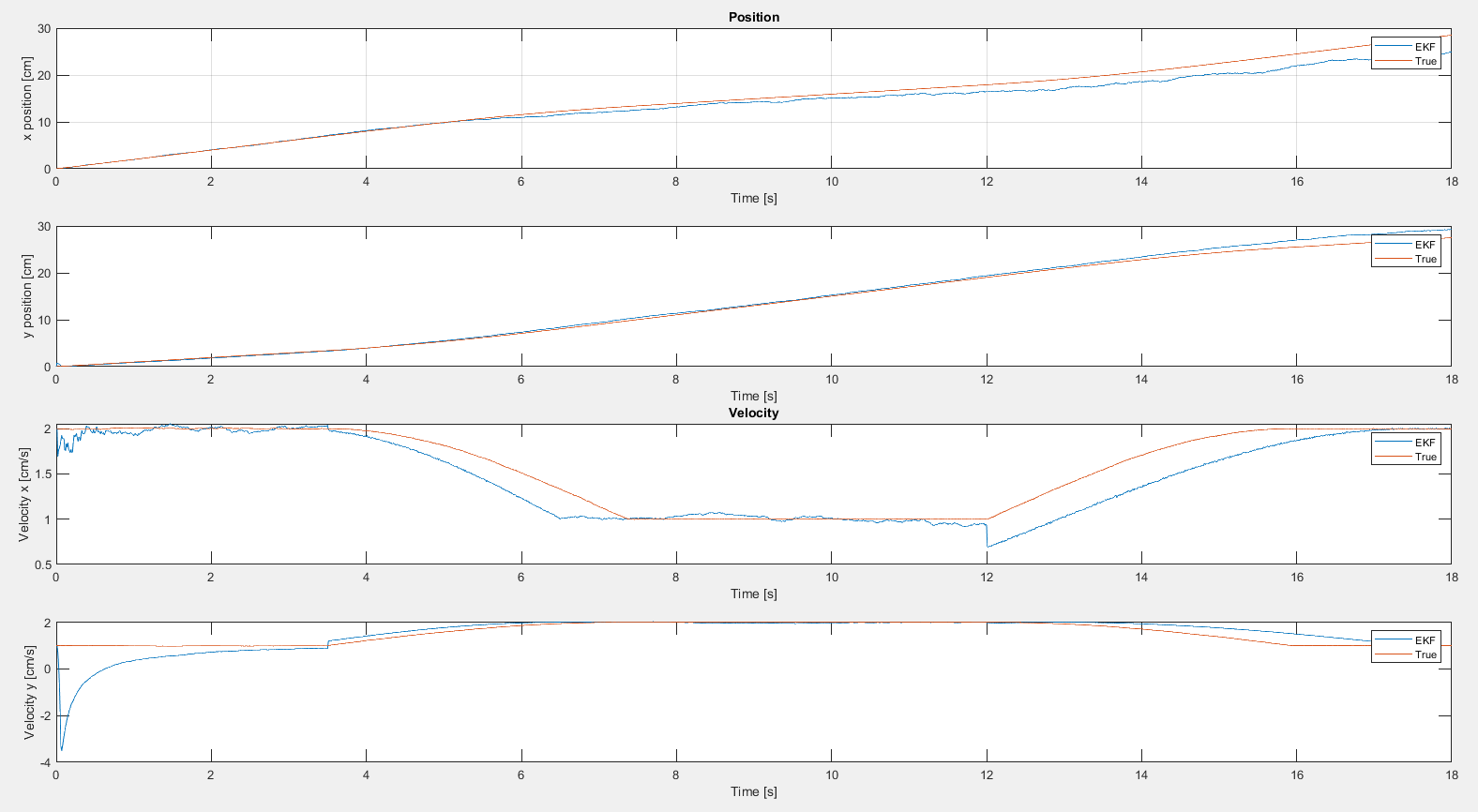
Figure 5 below shows the simulated and EKF processed position and velocity for trajectory 1, (2, 1) cm/s.

Figure Simulated and EKF processed position and velocities for trajectory 1.

Examining Figure 5, we can see that the EKF tracks the position fairly well but struggles to accurately track the velocity.

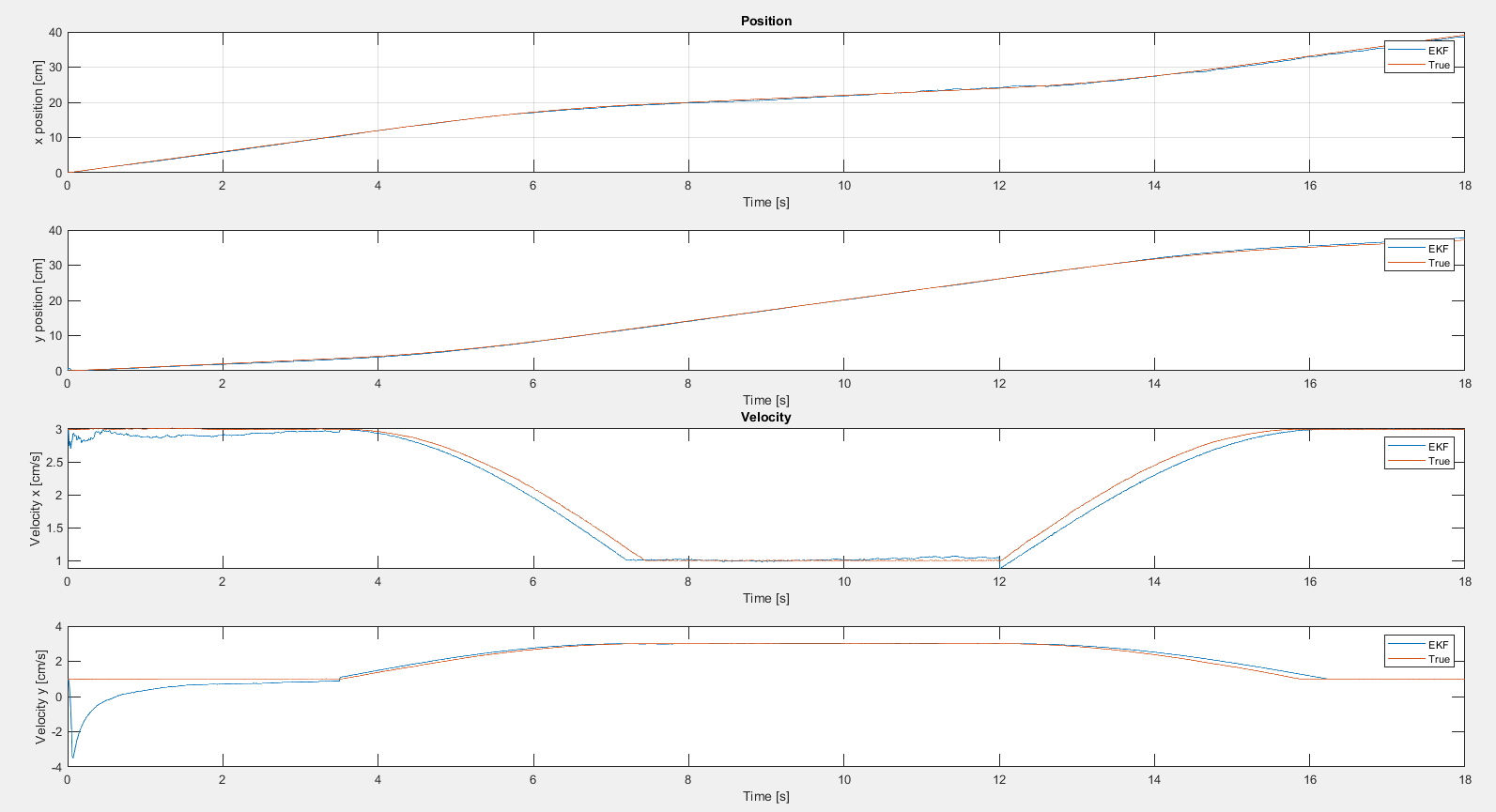
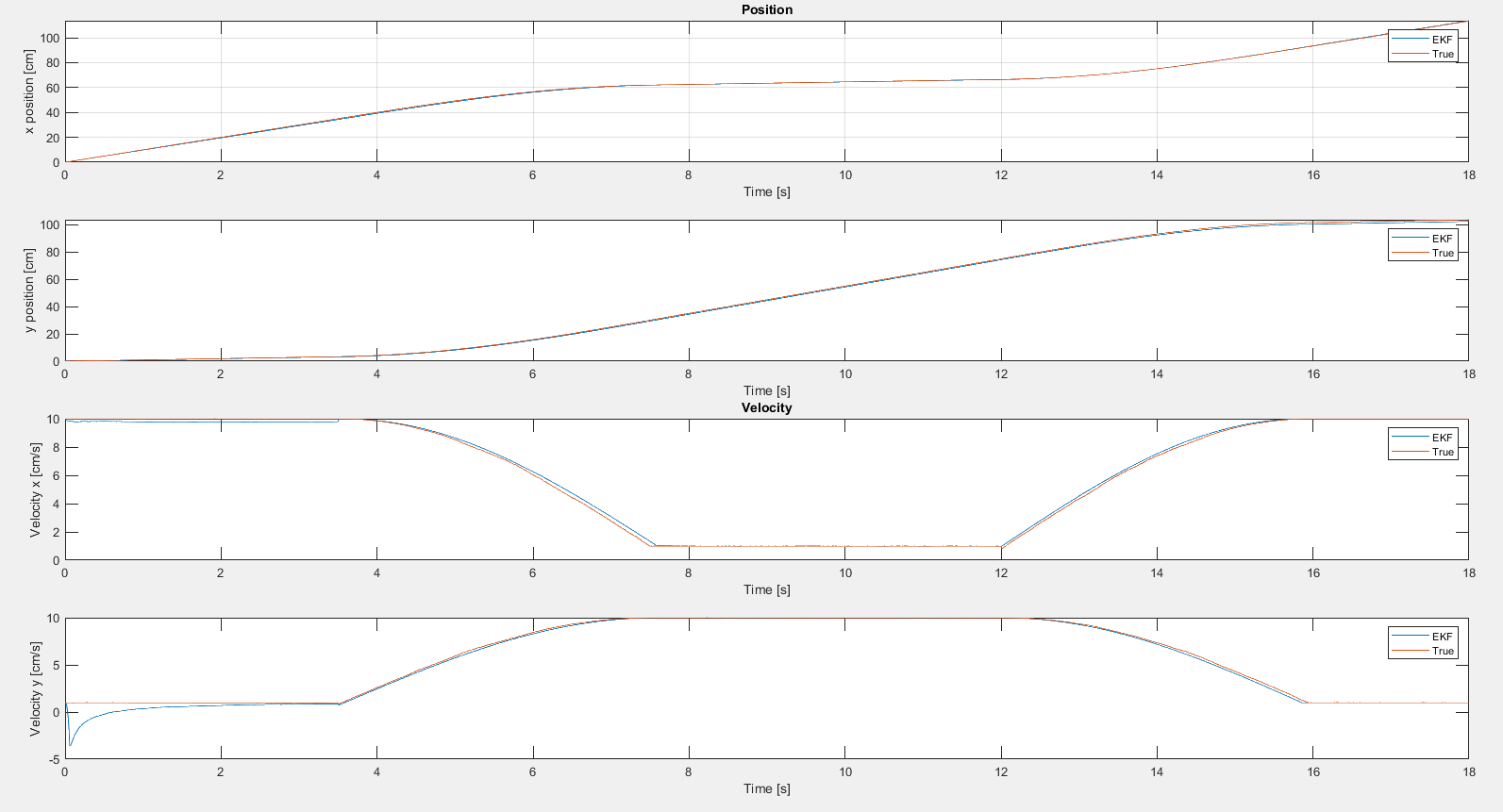
Figure 6 below shows the simulated and EKF processed position and velocity for trajectory 2, (3, 1) cm/s.

Figure Simulated and EKF processed position and velocities for trajectory 2.

Examining Figure 6, we can see that the EKF tracks the position and velocity very well overall but struggles to track the velocity in the first two seconds.

Figure 7 below shows the simulated and EKF processed position and velocity for trajectory 3, (10, 1) cm/s.

Figure

Examining Figure 7, we can see that the EKF tracks the position and velocity very well. Overall, it can be seen that the EKF tracks the positions and velocities of the three trajectories fairly well, but it has the most trouble tracking trajectory 1.

**Virtual Gyroscope**

Using the gyroscope model developed in Lab 1 for rotation about the z-axis, based upon the detected angular velocity, the estimated orientation of the phone is calculated and stored. The gyroscope model used can be seen below. Zero gaussian sensor noise of 0.001 was added to the sensor model.

*when rotation takes place*

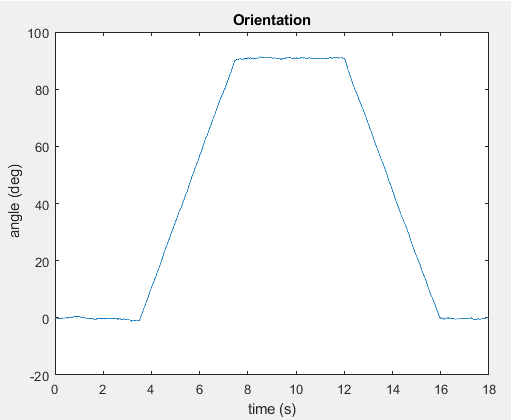
Figure 8 below shows the orientation of the phone along trajectory 1, (2, 1) cm/s.

Figure Orientation of the phone along trajectory 1 taken from stored gyroscope data.

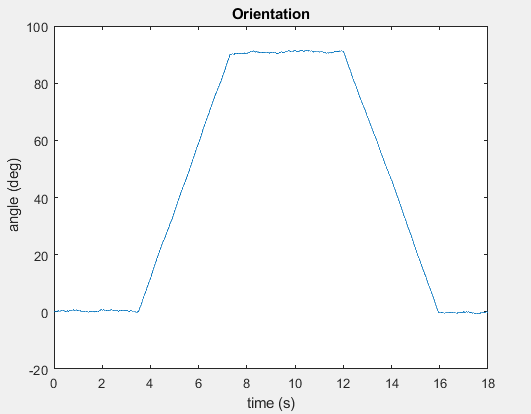
Figure 9 below shows the orientation of the phone along trajectory 2, (3, 1) cm/s.

Figure Orientation of the phone along trajectory 2 taken from stored gyroscope data.

Figure 10 below shows the orientation of the phone along trajectory 3, (10, 1) cm/s.

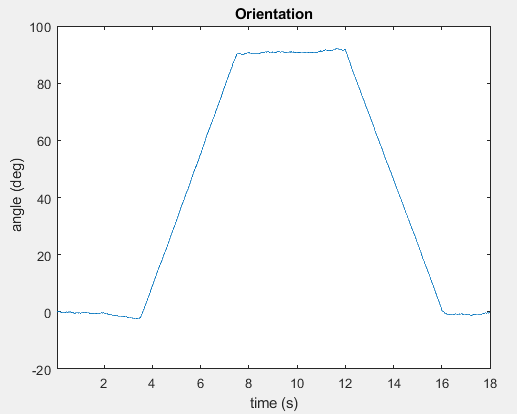
Observing the orientation plots, it can be seen that the virtual gyroscope can track the orientation of the phone quite well, but inaccuracies due to noise and drift still exist.

Figure Orientation of the phone along trajectory 3 taken from stored gyroscope data.

Please note that a linear model was used for the virtual gyroscope, which can be unrealistic when compared to real gyroscopes; but is not too unreasonable for our application where we are simulating a smooth constant rotation. Furthermore, real gyroscopes may not be subjected to zero gaussian noise and may be much noisier when compared to our virtual model. Our virtual model can be seen as idealistic rather than practical.

**Fuzzy Logic Algorithm**

To correct for the orientation of the phone, a soft decision making system utilizing fuzzy logic is required to determine whether the phone is in rotation. First, a weighted average of the phone’s estimated orientation is calculated using the magnitude of Kalman gain, see the equation below.

represents the stored orientation state taken from the gyroscope, and represents a calculated orientation state based on the stored velocity states from the accelerometer, it is calculated based on simple trigonometry. These states are taken at time state i.

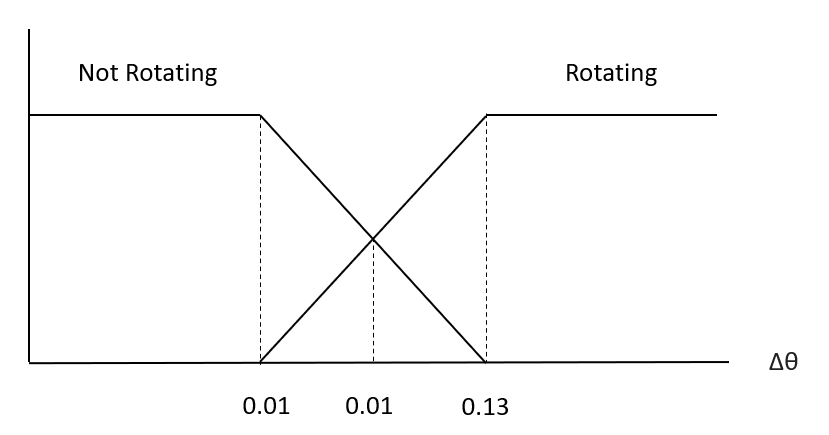
Fuzzy logic is then applied to the absolute difference between time state i and i-1 using the following fuzzy set. If it is determined that the phone is rotating, then the change in orientation is applied, otherwise there is no change.

Figure Image depicting the fuzzy sets used to determine the rotation state.

The membership functions for the rotating and not rotating sets are as follows:

To defuzzify the set, the maximum membership approach is taken, the set with the highest membership is taken as the true value. If it is determined that the phone is rotating, then the change in orientation is applied, otherwise there is no change. Figures 12 to 20 on the following pages illustrate the effects of the fuzzy logic algorithm in orientation correction for trajectories 1, 2 and 3. Examining the figures, it can be seen that the FLA successfully corrects the orientation of the phone from noise and drift and can mostly identify when the phone is going through rotation and not with the exception of a few anomalies, note that the fuzzy logic algorithm had the most trouble dealing with trajectory 1.

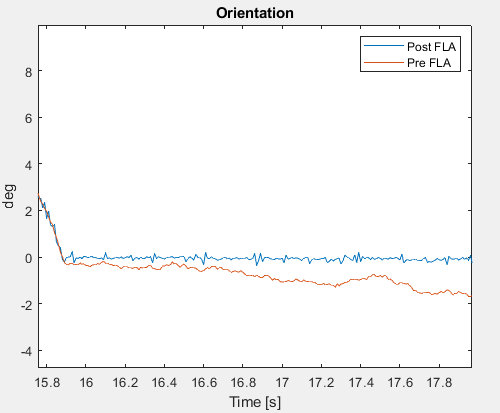
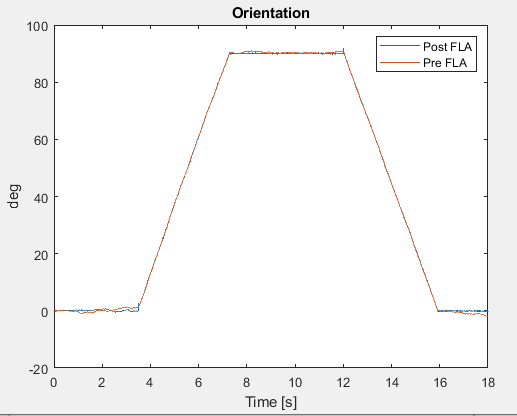


Figure Orientation of the phone before and after fuzzy logic processing along trajectory 1.

Figure Closer look at the effects of the FLA for trajectory 1.

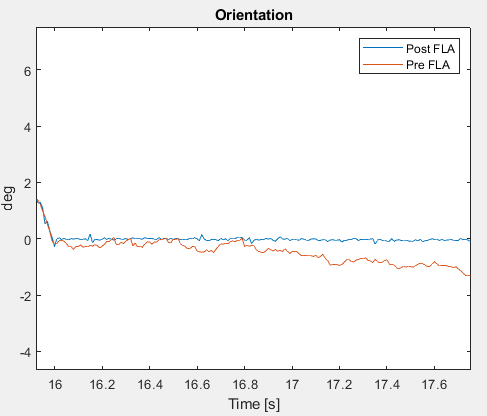
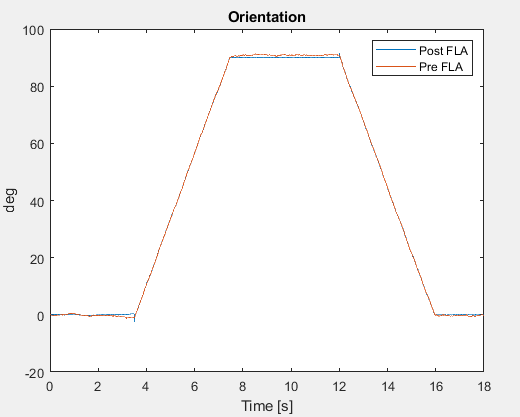


Figure Orientation of the phone before and after fuzzy logic processing along trajectory 2.

Figure Closer look at the effects of the FLA for trajectory 2.

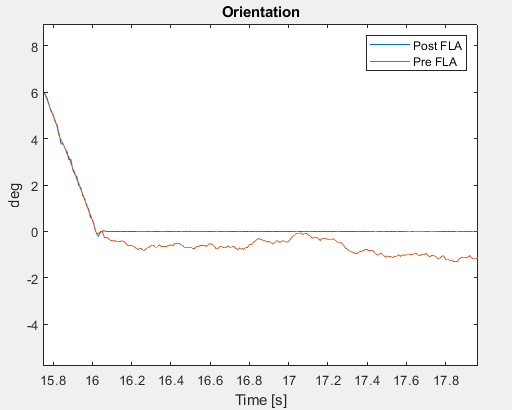
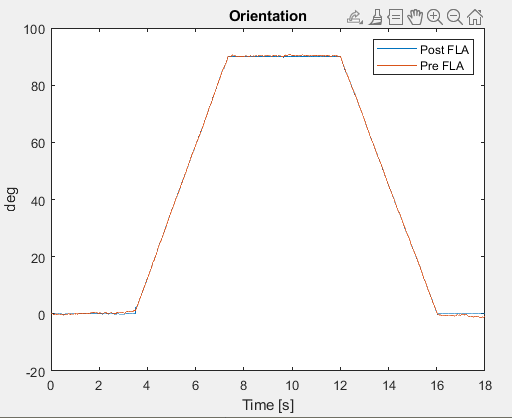


Figure Orientation of the phone before and after fuzzy logic processing along trajectory 3.

Figure Closer look at the effects of the FLA for trajectory 3.

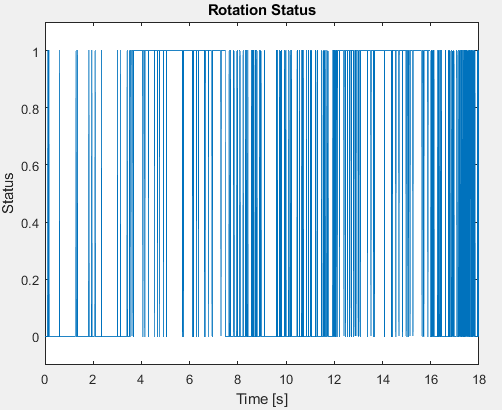
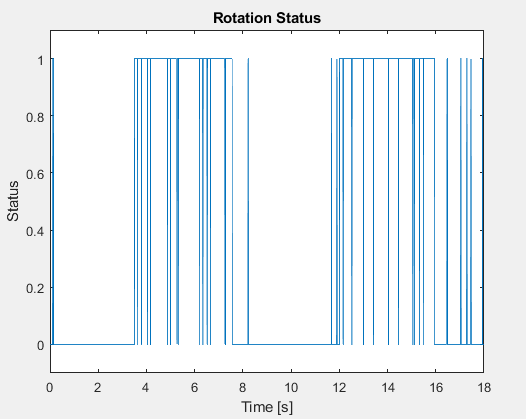
Figures 18 to 20 below reveal the FLA corrected rotation status, whether or not the phone is in rotation, throughout the simulation; a value of zero represents no rotation, whereas a value of 1 represents rotation.

Figure 19 Plot depicting the rotation status of the phone along trajectory 2.

Figure 18 Plot depicting the rotation status of the phone along trajectory 1.

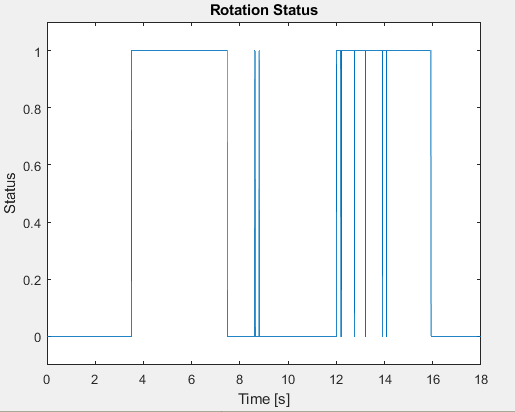


Figure 20 Plot depicting the rotation status of the phone along trajectory 3.

The simulation and testing of all three described trajectories were successfully completed with equal difficulty, but it should be noted that trajectory 1, (2, 1) cm/s, had the “worst” results out of the three simulated trajectories. Compared to trajectories 2 and 3, trajectory 1 had much worse EKF tracking for the position and velocity of the phone, in addition, it provided the most difficulty in determining the rotational status using the FLA.