Home Work 5
System Identification and Simulation
Kalman Filter, EKF, UKF & Particle Filter Algorithms

Name: Taimoor Shakeel Sheikh
Matlab Version: Matlab 2017a
Operating System: Windows 10

1. **This work is in pairs - choose a partner and write down his name. The rules how to work with lego you should know from previous homework.**

Answer 1:

My partner in this HW5 is Geesara Prathap.

2. **Construct lego robot, set trajectory of moving and collect indirect data from sensors (for example odometry, or you can even add smartphone on the robot and get data from it). In the report you should describe trajectory that you choose for your robot and data that you decided to collect (for sure you should choose such data that can help to identify trajectory, and as you understand it is not enough to have just odometry to make data fusion). Pass sensors data in separate files with appropriate names.Collect at least 300 values for each sensor.**

Answer 2:

The trajectory I used for this task is from the cognitive lab the path as used in my previous assignment and it have sharp turns in the trajectory and it forms the L shape line on which I make my robot move as shown below in figure 1 and the data is collected using two different cellphones placed on the robot and with the help of application named as AndroSensor which I downloaded from the google play store gets the sensors data like accelerometers, linear acceleration, GPS Location, Orientation from two different cellphones sensors in each direction XYZ and saved in excel file named. For each sensor I collected more than 300 values from both sensors.

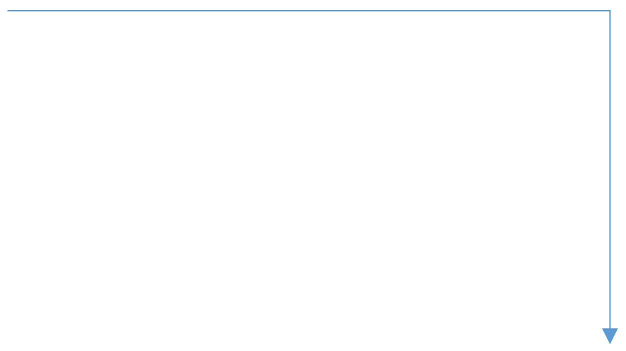


Figure 1: Trajectory Used to Move Lego Robot

3. **Give your sensors data to your partner (do not forget to give description what is the data) and get data from him (just data, not a trajectory).**

Answer 4:

The data file I get from my partner its format is CSV and format of file is shown below. It include the sensors data like accelerometers, GPS Location, orientation, linear acceleration in each direction XYZ and in excel file named as DATANEW, For each sensors data I select more than 300 values from both linear acceleration sensors in X & Y directions.

Format of excel file in which sensors data's (0 and 1) are saved.

Sensor 0

GPS (X, Y, Z) Accelerometer (X, Y, Z) Linear Acceleration (X, Y, Z)

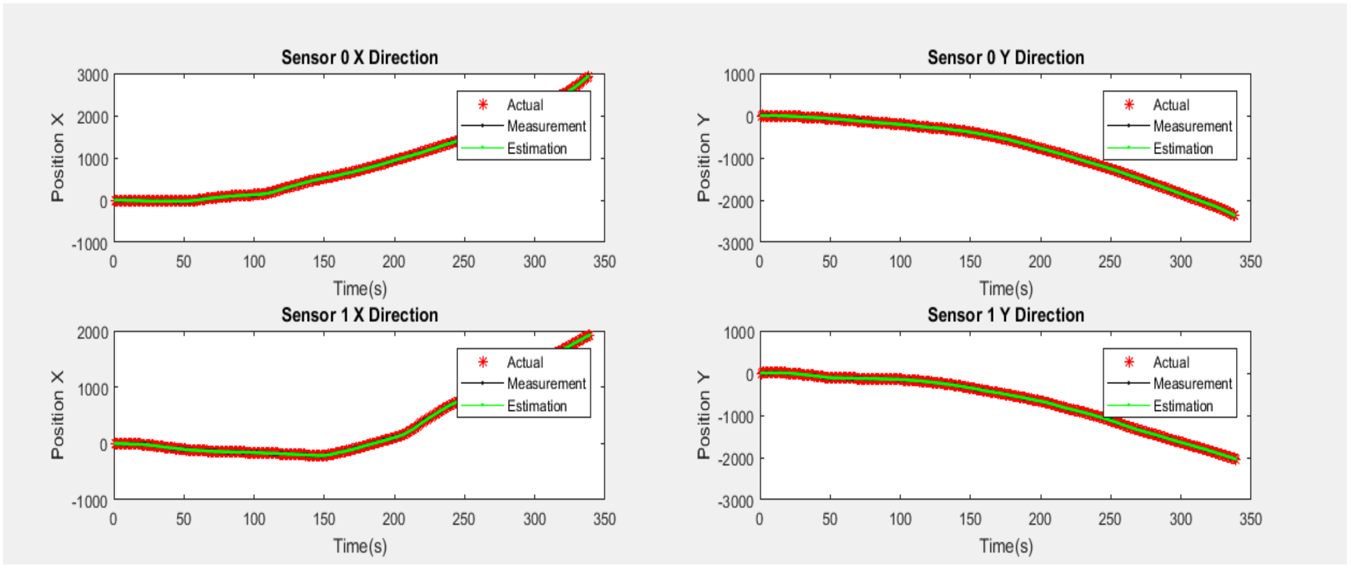
Sensor 1

GPS (X, Y, Z) Accelerometer (X, Y, Z) Linear Acceleration (X, Y, Z)

4. **Try to guess your partner's robot trajectory. For that firstly construct trajectory just by each sensor separately. Then try all algorithms that can be applied in your case (KF, EKF, UKF, and PF) for data fusing. If it cannot be applied - show why. Compare all results and describe why you got such results. Full points for the task you will get only if implement all algorithms manually (but you can use libraries to get some points). And again try to guess trajectory.**

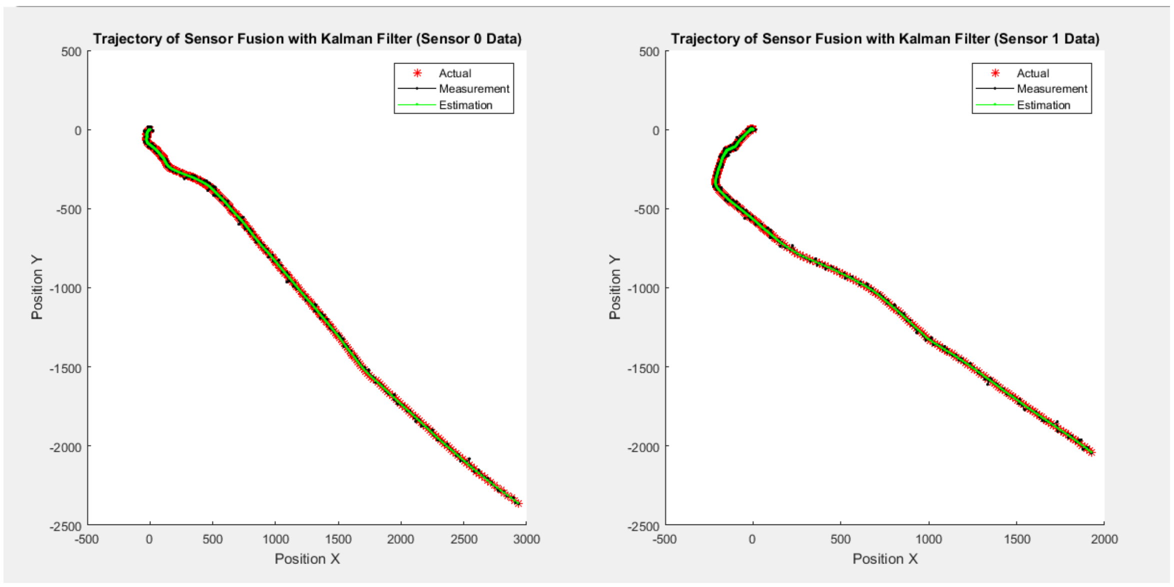
Answer 4:

For the partner's robot trajectory first I construct trajectory by two sensors linear acceleration sensor 0 and linear acceleration sensor 1 in X and Y direction separately. In below figure the Y axis showing the position x and position y with respect to different time instances of both sensors 0 and 1.

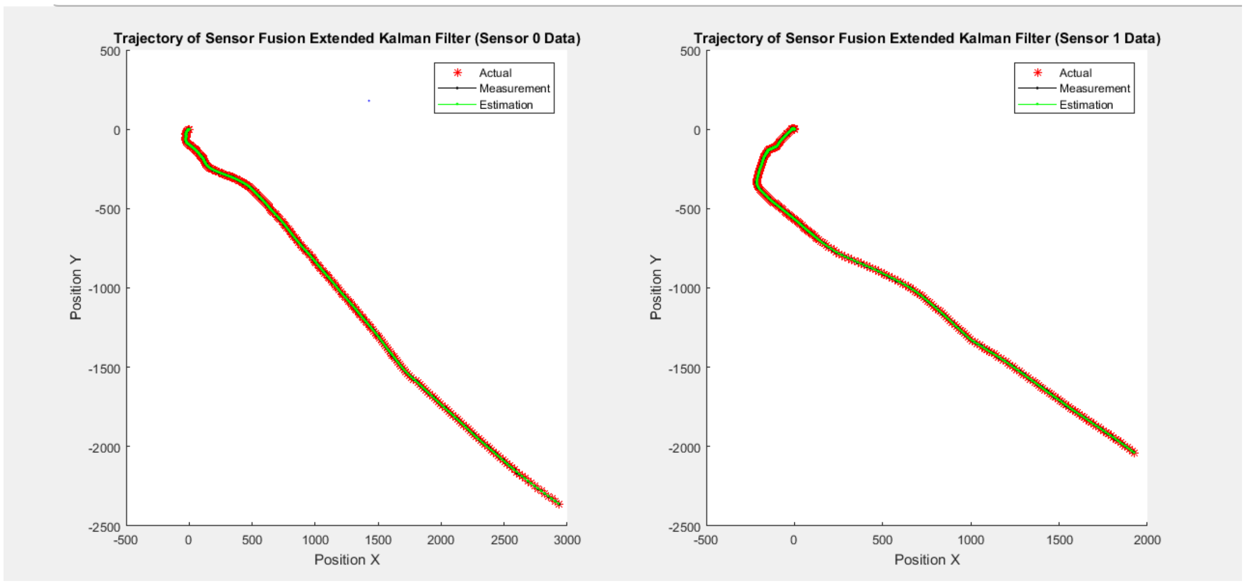


Different filter results using data fusing in my case I was successful in applying all algorithms but in case of particle filter it is not giving good results.

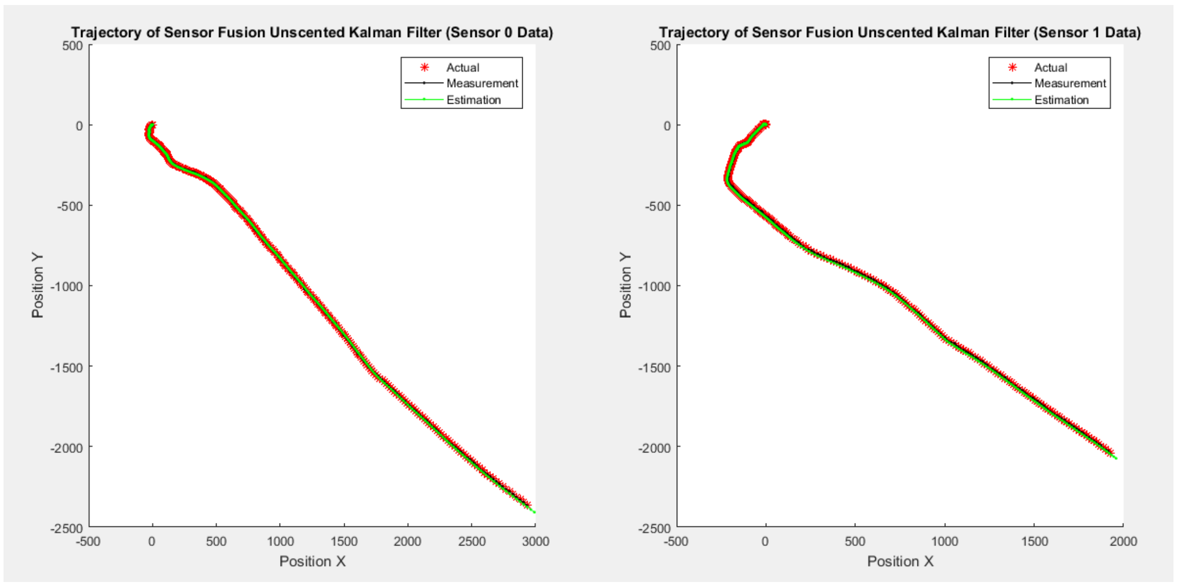
- **Kalman Filter Result:**
  In kalman filter I get the best result of sensor fusion on two sensors dataset because it can be easily formed with the help of linear equations,kalman gain and the measurement value and estimated value is almost same and they are following the exact trajectory as you see in below picture there are three data in each subplot actual value shown in red color star, the measurement value in black value dot and the estimated value of the trajectory is in green(line + dot) and the distribution is bias in this case as compared to other filters. In this model I have also added the Q (covariance of state vector) and R (covariance of measurement vector) as noise to the model. If R approaches to 0 then kalman gain is equal to the Matrix C and the system will give fully observational value, and if Q approaches to 0 the estimated and actual value becomes approximate same.



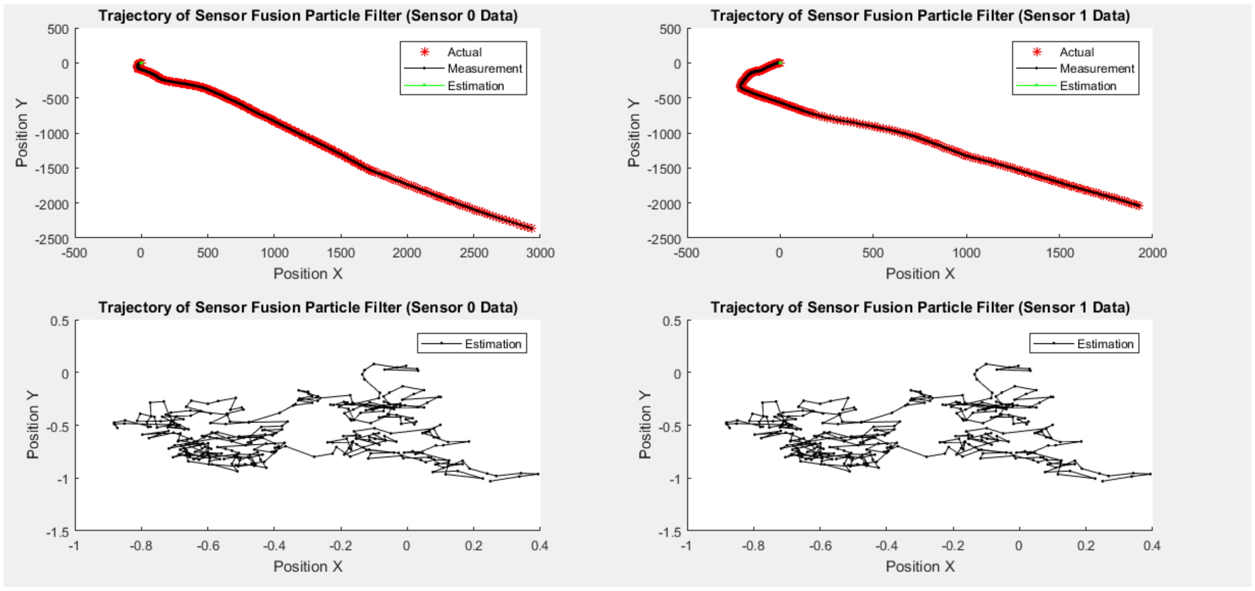
- **EKF Result**
  In extended kalman filter we don't have linear equations so in order to apply this filter we need nonlinear equations, Kalman Gain and jacobians of the measurement and observation model and the results of this filter is my case is equivalent to the kalman filter results and it shows that result of EKF sensor fusion is good because it can be easily formed with the help of nonlinear equations the jacobians of model. The measurement value and the estimated value is very close and it is good approximation of sensor fusion on two sensors dataset as shown in below figure there are three data in each subplot actual value shown in red color star, the measurement value in black value and the estimated value of the trajectory is in green and If R approaches to 0 then extended kalman gain is equal to the Matrix H and the system will relate all observational value, and if Q approaches to 0 the estimated and actual value becomes approximate same so extended kalman filter can be formed from the linear equations if equations can be changed to nonlinear form.



- **UKF Result**
  In unscented kalman filter again we don't have linear equations so in order to apply this filter we need nonlinear equations, sigma points and the Cholesky matrix and there is Wm (Covariance of Sigma Points as weights) and Wc (Covariance of Points as weights) which can be find on the basis of alpha, beta, k where alpha is chosen such a way that it should describe our model by considering the sigma points, n is dependent on the state vector, beta changes with alpha. So by setting these all parameters if we can make linear model from prediction model so it will be consider as UKF. The result of sensor fusion on two sensors dataset is equivalent to the EKF but not better then KF because as we can see in figure the estimation is done but the results are not closely estimated as in previous cases. There are three data in each subplot actual value shown in red color star, the measurement value in black value and the estimated value of the trajectory is in green.



- **PF Result.**
  In particle filter the distribution is modeled with the help of sample as densities and compute weights at different instances so algorithm can be used to estimate the posterior distribution. Normalize and resampling is done in next step to obtain N equal-weight particles. So at each time step we get the estimate distributions and we have Q (Covariance of particles) and R (Covariance of measurement weights). So by setting these all constraints the result I get from particle filter on sensor fusion of two sensors dataset is not so good in my case because it doesn't estimate the results accurately as shown in figure as the estimation is done in previous filter cases. There are three data in each subplot actual value shown in red color star, the measurement value in black value. The 3rd and 4th subplot is showing the estimated value of the trajectory in black color.



- **Guessed Trajectory.**
  The trajectory I get from my partners dataset is shown below it's the almost L shape with not so much turns except 1 turn otherwise it is almost approaching in some straight line.