Project 5: Extended Kalman Filter

Kalman filter is a technique for tracking a moving object w.r.t to the autonomous vehicle. It is an iterative method with prediction and update steps. In the prediction step, using a physics model, the position of the object is predicted with the current knowledge. In the update step, the measurements from the sensors on the car are used to update the position of the object.

Sensor fusion: There are 2 sensors used for this project: Lidar and Radar; In contrast to the Camera as the sensor in the prior projects. A Lidar uses infrared rays and Radar uses sound waves. Measurements from Lidar are of higher resolution, whereas the Radar can be used in all weather conditions and can measure velocity of the tracking object in addition to its position. A comparison of the three sensors is shown in the table below.

	Camera	**Lidar**	**Radar**
Cost	low		
Resolution	high	high	
Noise	low		
Velocity measurement			yes
Weather condition			all
Objects not in line of sight			yes
Small/ static objects	yes	no	yes

Table 1: Comparison of different sensors mounted on autonomous car

Extended Kalman filter: Kalman filter works on a linear model. The physics model used in the prediction step is a linear model, however, the radar measurements returned by the sensor in polar coordinates need a nonlinear model to convert to cartesian coordinates. To continue using the Kalman Filter method, the polar to cartesian conversion is linearized (approximately) using a Jacobian. Due to this generalization, the current implementation is called an Extended Kalman Filter.

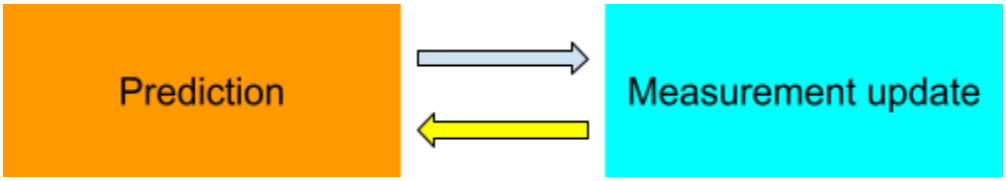

Fig 1: Kalman Filter is an iterative technique with Prediction and Update steps

Equations for prediction step

$$X = F*X$$
$$P = F*P*F' + Q$$

X = mean state vector
F = State transition matrix
P = variance in the state estimation
Q = Process noise

Equations for measurement update step

$$y = z - H*X$$
$$S = H*P*H' + R$$
$$K = P*H'*S^{-1}$$

$$X = X + K*y$$
$$P = (I - K*H)*P$$

z = sensor measurement
H = Measurement function - converts state to measurement space
R = measurement noise
K = Kalman gain
I = Identity matrix

Noises in measurement and process: The noises in the sensor measurements are captured in the R matrix and the noises in the state estimation are captured in the Q covariance matrix. Measurement noise matrix can be populated from the specification of the manufacturer; For the current project, this is given to the students. Process covariance matrix is updated based on the time interval between the measurements and uncertainty in estimation.

Performance measure: The performance of the Extended Kalman Filter is verified by a RMSE (Root Mean Square Error) measure between the model outcomes and the ground truth values provided.

Software environment: The code for object tracking is developed in C++. The open source Eigen library is used for matrix and vector manipulations. A simulator provided by the Udacity team is used to visually observe the outputs from the model and the sensor measurements.

Project steps:

1. Set up the C++ environment with uWebSocket to communicate between the Extended Kalman Filter and the Simulator
2. Develop code for Prediction step
3. Develop code for Radar measurement update
4. Develop code for Lidar measurement update

Note: The phi value in error vector ‘y’ has to be adjusted to be with [-pi,pi]

5. Develop code for Jacobian and RMSE measure

Screenshots from the simulator with the radar and lidar sensor measurements and the model outputs for the two datasets are shown below.

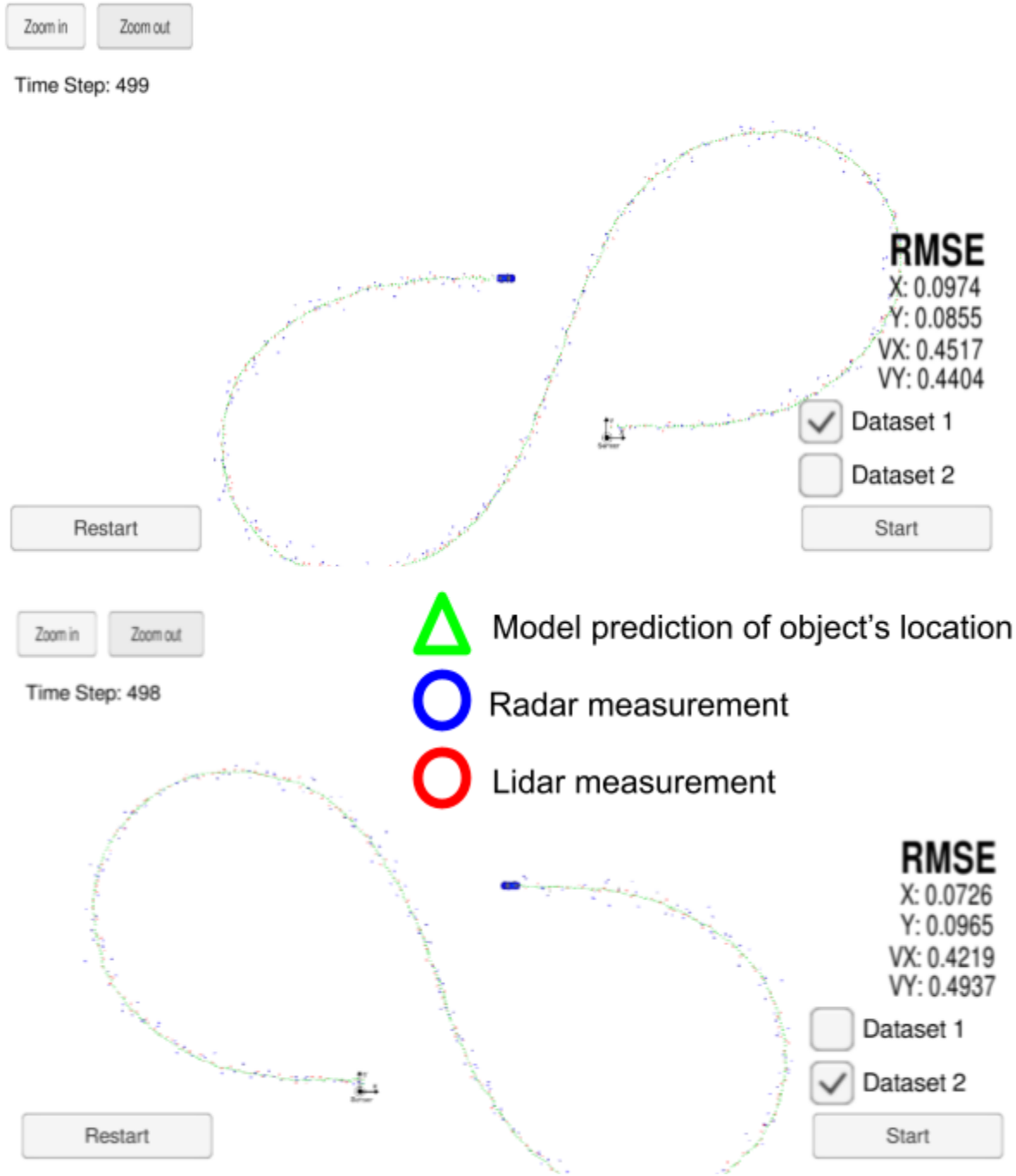

Fig 2: Simulator output with sensor measurements and model estimations for Dataset 1(Top) and Dataset 2(Bottom)

The RMSE values are within the desired limit [0,11,0,11,0,52,0,52]