Khalil Fadhel
Udacity- Self Driving Car Course
Term 1

15 October 2017

Unscented Kalman Filter

In this project we aim to implement the unscented Kaman Filter on C++. We are provided a dataset of position points of a car moving lineally detected using a Laser and Radar sensors. In this project, we use the CTRV motion model instead of the constant velocity model. We test our filter at the end using a simulator and we end up by evaluate our filter's results.

Implementing the Kalman Filter

We start by initialising our state. To do so, we setup the mean with the current measurement and we set the covariance matrix in a way that it illustrates high uncertainty. We need to project our sensor measurements from the measurement to the state space. We also initialise the Laser and Radar covariance matrixes.

Once our filter is initialised we start the prediction step. The prediction step is run semioeungsly with Radar or Laser measurements.

Prediction Step

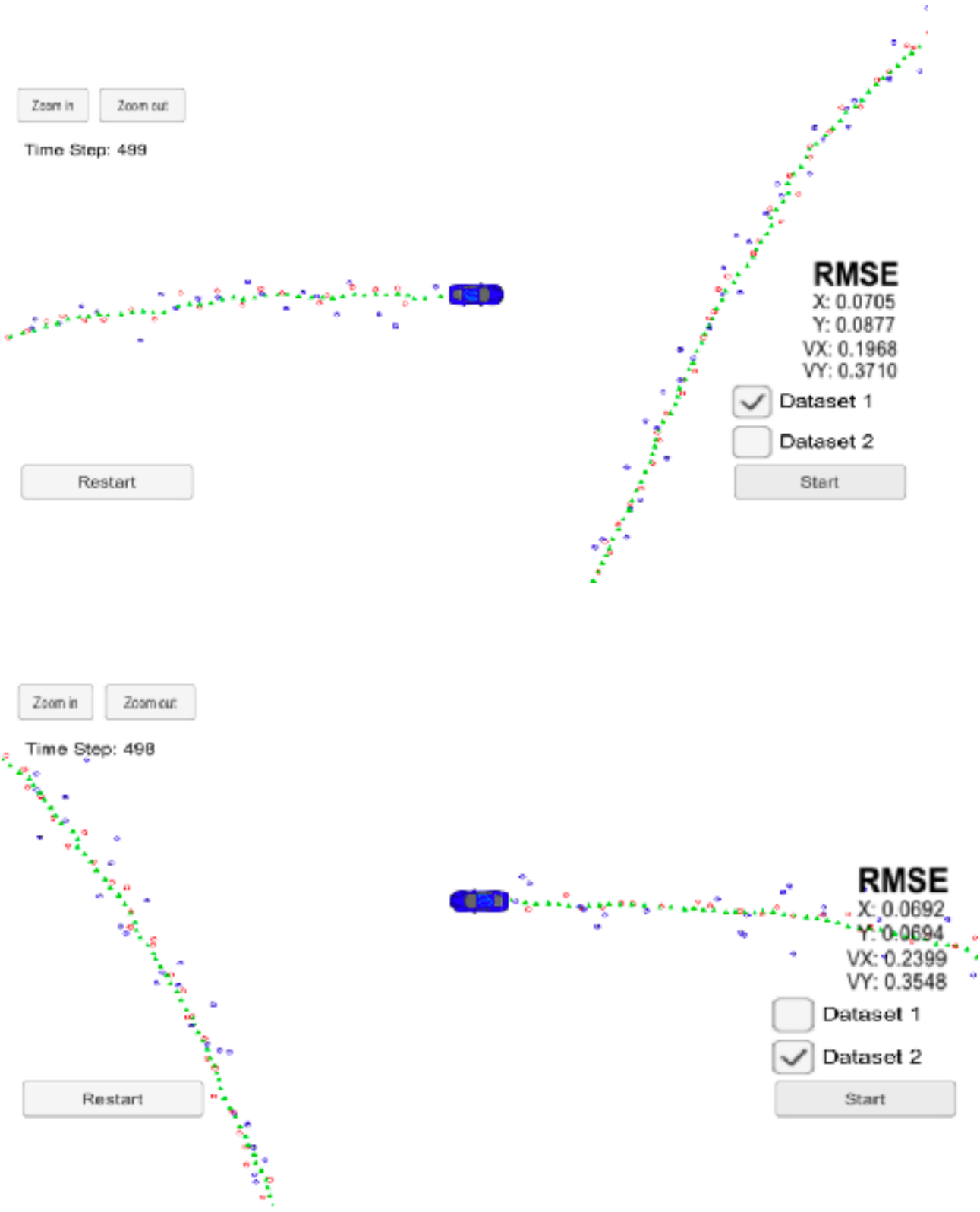
As our prediction function is no longer linear, we need to use sigma points to predict our state after dt. We start by augmenting our state vector with the noise vector and our covariance matrix with the noise covariance matrix. We generate the sigma points, we predict the sigma points then we get back the mean and covariance matrix of the predicted state.

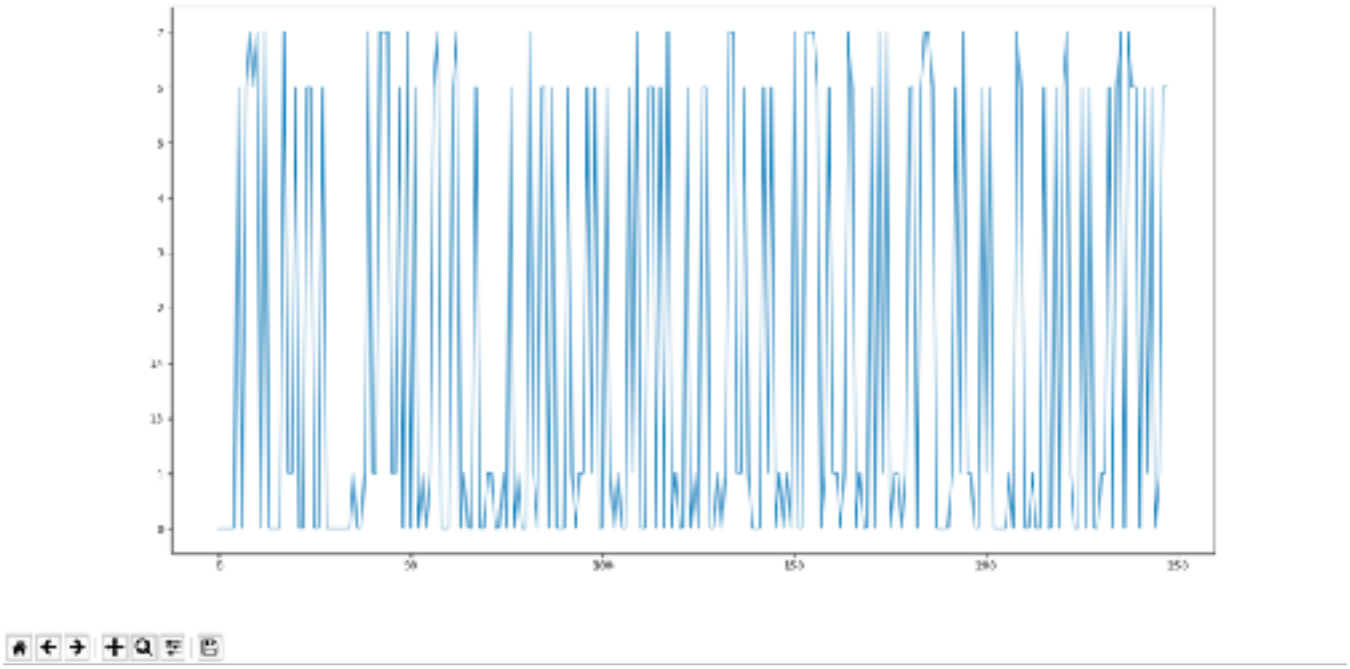
Update Step

As we can transform prediction state to the laser measurement space literally, the laser update step is similar the one for the extended Kaman filter. For the radar update step. We transform the predicted sigma points to the radar measurement space and we get back the mean state and covariance matrix.

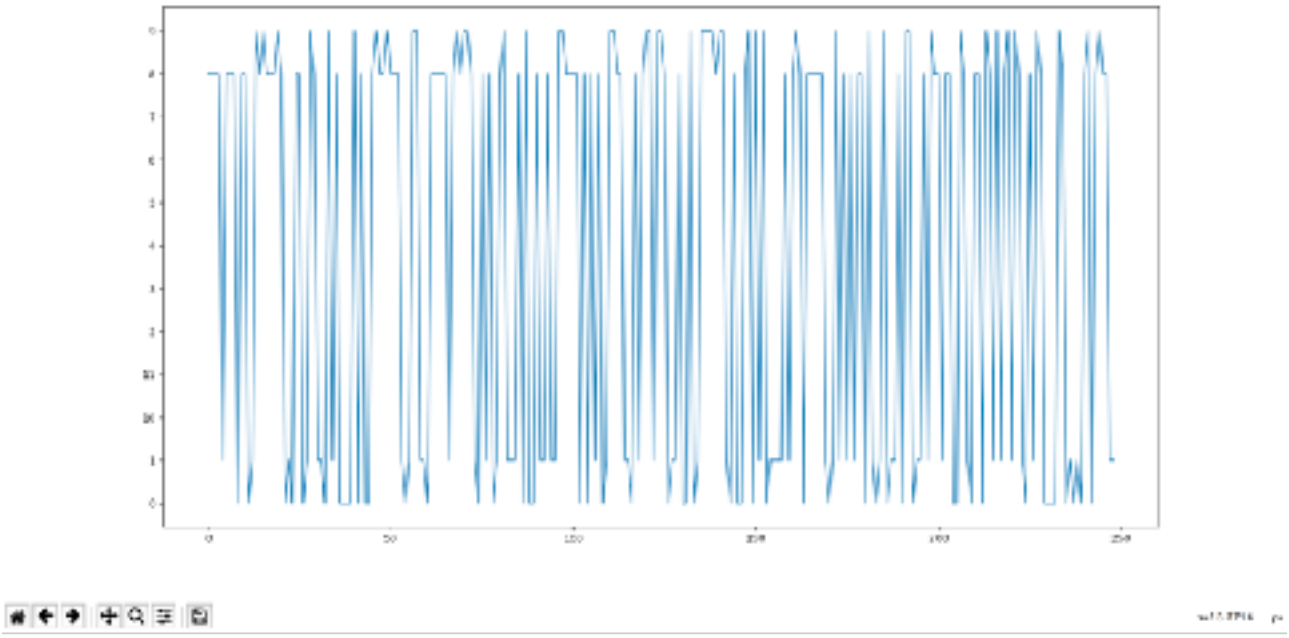
Evaluation

To evaluate our Kalman Filter implementation we calculate the RMSE using the provided ground truth values provided by the simulator. We test our results on the simulator and we succeed to improve the tracking on the start by tweaking the process noise. We also calculate the NIS. Find below the final results.


Zoom in
Zoom out
Time Step: 499
RMSE
X: 0.0705
Y: 0.0877
VX: 0.1968
VY: 0.3710
Dataset 1
Dataset 2
Restart
Start
Zoom in
Zoom out
Time Step: 498
RMSE
X: 0.0692
Y: 0.0694
VX: 0.2399
VY: 0.3548
Dataset 1
Dataset 2
Restart
Start



Laser measurements NIS



Radar measurements NIS

Conclusion

In this project, we succeed to track a vehicle moving lineally by implementing a Kaman filter. We used radar and laser sensors as input and our file was able to disregard noise and correct wrong detection by using both sensors data and predicted states.