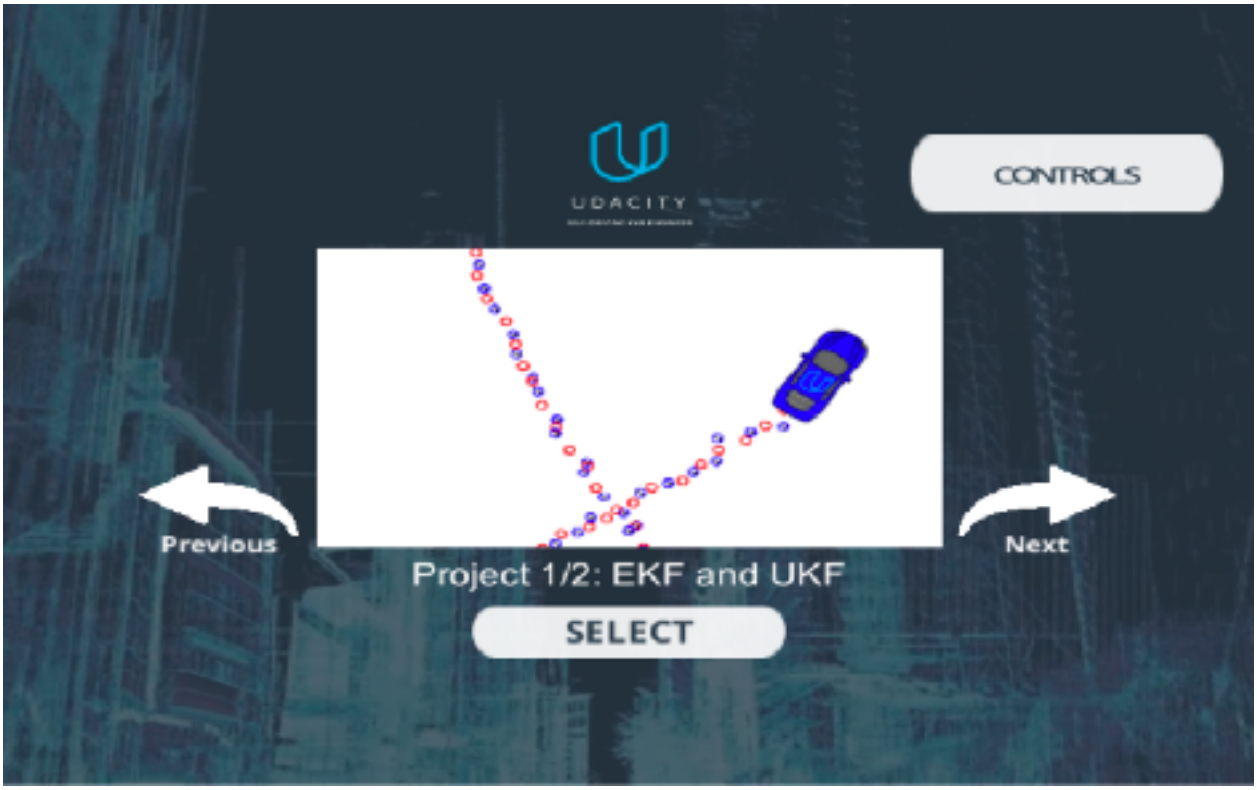
Khalil Fadhel
Udacity- Self Driving Car Course
Term 1

15 October 2017

Extended Kalman Filter



In this project we aim to implement the extended Kaman Filter on C++. We are provided a dataset of position points of a car moving lineally detected using a Laser and Radar sensors. 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. For the radar measurements we need to project the state from the Polar to the Cartesian space. We also initialise the Laser and Radar covariance matrixes.

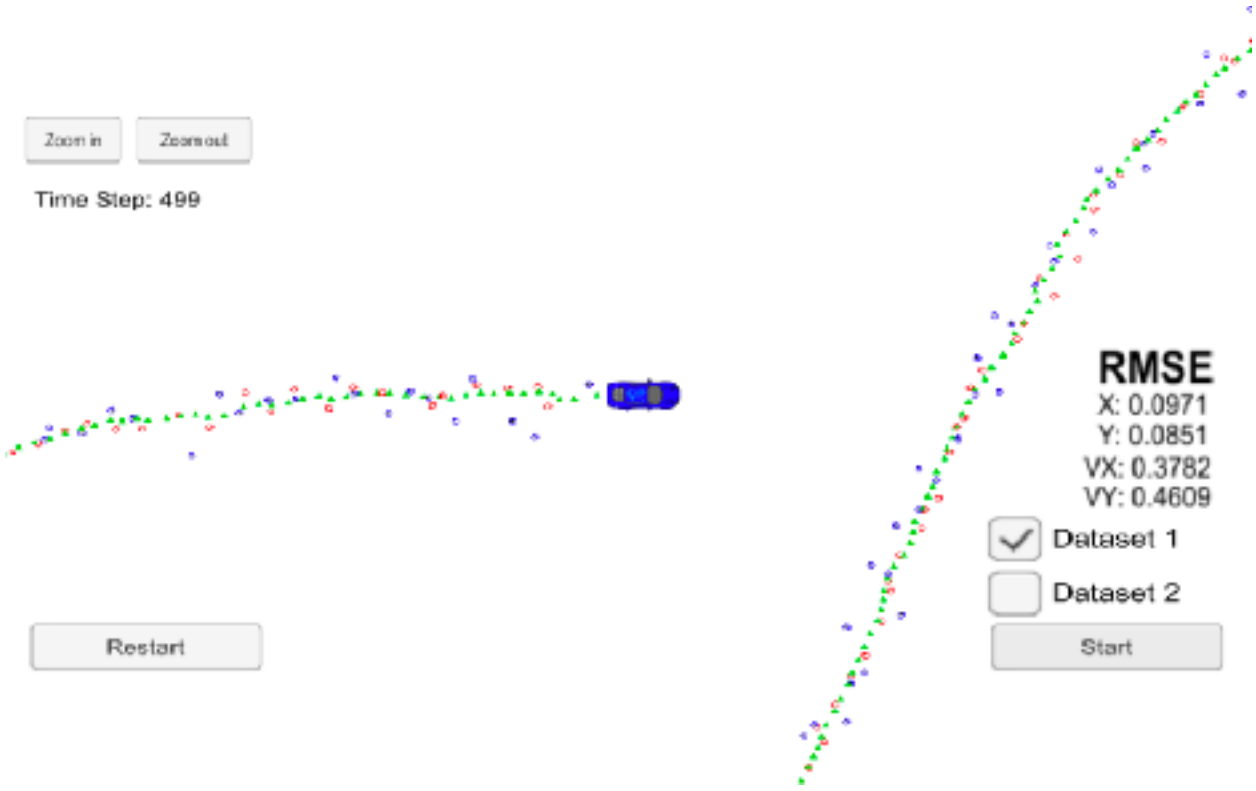
Once our filter is initialised we start the prediction step. The prediction step is run the same way with Radar or Laser measurements. We start by updating our transition and noise matrixes with the new elapsed time (dt). Then we calculate the predicted state and the corresponding covariance matrix.

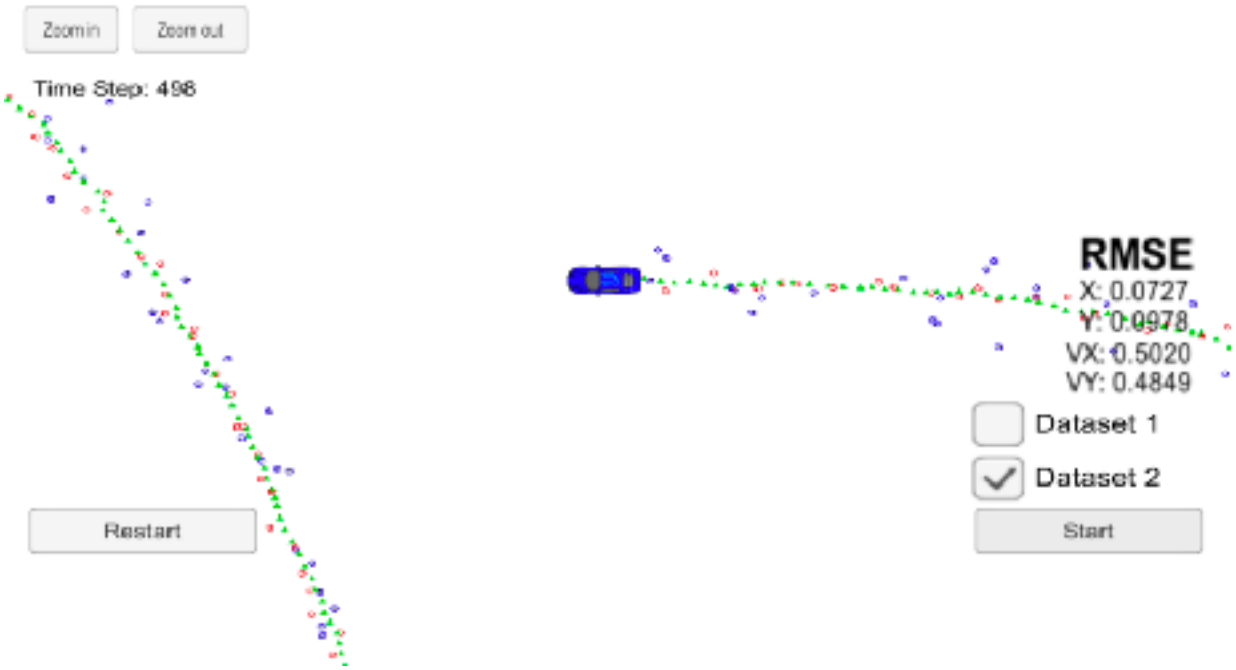
We finalise our filter implementation with the update step. For the update step we start by checking the source of our measurement. If it's from a laser sensor we use the predicted state

and the measurement state to calculate the state and the covariance matrix. If the measurement is from a radar sensor, we map the predicted state to the poll state using the h function, then we calculated the Jacobean and finalise by updating the 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 covariance matrix initialisation. Find below the final results.





Conclusion

In this project, we succeed to track a vehicle by implementing a Kaman filter. We used radar and laser sensors as input and our filter was able to disregard noise and correct wrong detection by using both sensors data and predicted states. An Improvement of our solution will be improving the detection on the turns.