

# Write Up for P5: Extended Kalman Filter

<b>Compiling</b>	<b>0</b>
<b>Accuracy</b>	<b>1</b>
<b>Project discussion</b>	<b>1</b>
Dealing with first frame	1
Remove radar/laser	2
Going further	2

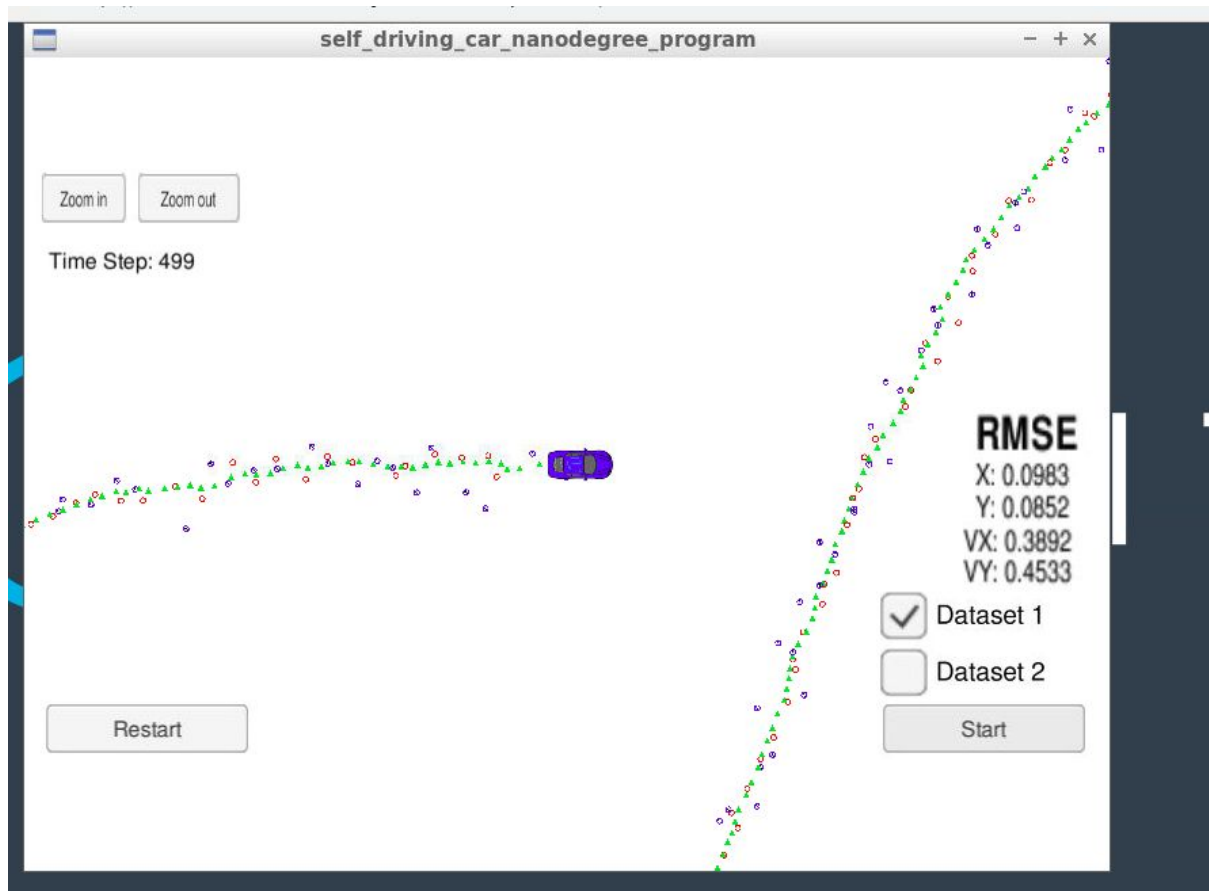
## Compiling

This document is a write up for the 5th project. My submission is included in the git repository: <https://github.com/mmarouen/CarND-Behavioral-Cloning-P3>

The code was run using the workspace settings and is compiled using `cmake` and `make`

# Accuracy

RMSE=[0.0983,0.0852,0.3892,0.4533]



## Project discussion

### Dealing with first frame

The parameters that can be tuned for the first frame are:

Initial state  $\mathbf{x}_0$  + initial covariance  $\mathbf{P}_0$  (other parameters were chosen adequately).

- Initial state  $\mathbf{x}_0$ :

I use the first measurement as initialization for  $\mathbf{p}_x$  &  $\mathbf{p}_y$  and set  $\mathbf{v}_x=\mathbf{v}_y=0$ .

```
x_ <- rho*cos(theta), rho*sin(theta),0.0,0.0;
```

```
x_ <- measurement_pack.raw_measurements_[0], measurement_pack.raw_measurements_[1],0.0,0.0;
```

- Initial covariance  $\mathbf{P}_0$ :

For me this is the trickiest part in all the project!

After several trials, I decided to use the following covariance matrix as initialization:

```
P_ << 200.0, 50.0, 30.0, 0.0,
      50.0, 100.0, 0.0, 15.0,
      30.0, 0.0, 20.0, 5.0,
      0.0, 15.0, 5.0, 10.0;
```

Here are the main reasons for this:

- High covariance for `px` compared to `py` because movement is mostly along x-axis
- Same applied for `vx` and `vy`
- An increase in `px` will affect positively `py`

## Remove radar/laser

### Laser only



**RMSE**  
 X: 0.1477  
 Y: 0.1155  
 VX: 0.7030  
 VY: 0.5356

### Radar only



**RMSE**  
 X: 0.2275  
 Y: 0.3468  
 VX: 0.5731  
 VY: 0.8098

- Laser is better at detecting `px` & `py`
- Radar is better at detecting `vx`
- Laser is better at detecting `vy`

These results are not surprising because radar measurements are more noisy.

On the other hand, radar observes speed, which may explain why radar detect `vx` better.

## Going further

This project was beneficial to get started with c++ and apply my knowledge on Kalman filters. I think the RMSE can be improved by:

- Better initialization of `P_0`
- Better initialization of `x_0`