

Kalman Filter

Kalman Filter is implemented as a class. It accepts matrixes A , H , Q , R , and P in the constructor. It has also a method to initialize state $x(0)$. The estimation is done in the *update* method.

What could be done better:

- Allow the update method to accept a new matrix A , when it changes with time
- Allow the class to accept the B and D matrixes and u vector
- Keep track of time (summing dt at every step)

Car Track

The implementation is similar to the .m file that was provided with the homework. In the beginning a KalmanFilter object is created and initialized, then data regarding the car is generated, finally the KalmanFilter is run to estimate the car's position and velocity.

In the end the RMSE on such two metrics is computed.

Tests

There are two tests.

One makes sure that the KalmanFilter object throws an exception when the *update* method is called without initialization.

The other makes sure that the KalmanFilter object makes a good prediction with a single variable system, where the output is a constant with some noise added or subtracted. A good prediction is: the difference between the estimation and the actual measure is less than one.

Note: I am not completely sure that this is the right way to test a kalman filter. Regardless of how I did on the assignments, I would like to know how you would do the testing.