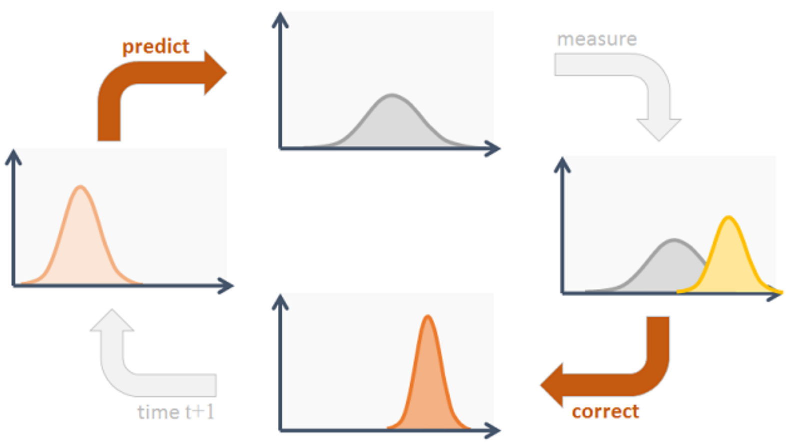


APPROACH TO KALMAN FILTER

Space Systems

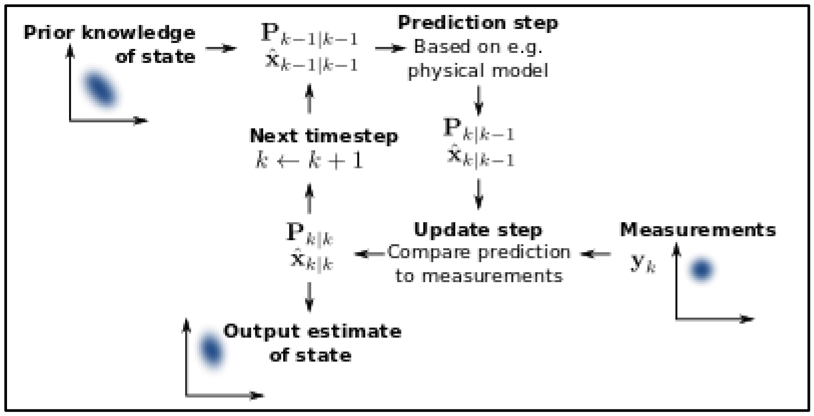
LOTI.00.005

Aditya Savio Paul
MS : Robotics and Computer Engineering
University of Tartu
Estonia
E: Aditya.savio23@gmail.com

Contents

1. Concept

Kalman filtering, also known, is an algorithm that uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone.

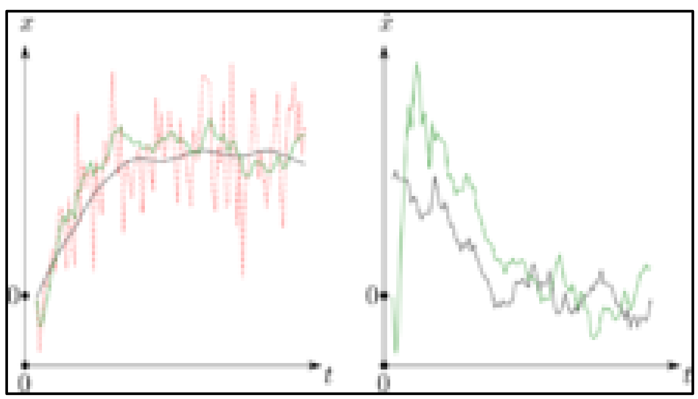


1. Kalman Filter: An Approach

Also known as linear quadratic estimation (LQE) it works by estimating a joint probability distribution over the variables for each timeframe.

A Kalman Filter finds heavy application in vehicular navigation and coordination, for semi-autonomous and autonomous control along with robotic operations and trajectory control. The Kalman algorithm is recursive and can run in real time, using only the present input measurements and the previously calculated state and its uncertainty matrix without the knowledge of past information.

It takes into accont the current and previouls values, along with the uncertainty matrix that produces the output results as apredicted variant which can be plotted over the true values and variations can be visually anaysed, accomodated or corrected, in the control algorthm filter.These variations are helpful in trajectory control and optimisation for the vehicle or robotic system; as it correlates the noise and covariance along with the uncertainty matrix giving high precision for the data as a data set as well as for the entire subsystem working simulateneously with the other subsystems as well. This bring in an efficient work coordination between different subsystem or the entire system as a whole to be able to execute a trajectory and optimise it real time.



2. True, Predicted and Observational Values

2. AIM

Implement a Kalman Filter on an existing set of data values and study variations in noise observation and covariance matrices.

3. Objectives

a. Obtain predefined environment for true position / values
b. Input Kalman Filter equations for predicted values
c. Plot True Values, Predicted Values and Observations
d. Study different Covariance, Observation and Noise matrices

4. Resources

a. Hardware
   Processor : Intel Core i5-8250U CPU @1 .60GHz/1.80GHz
   RAM : 6 Gb, 64 Bit x64 Based Processor

b. Software
   Windows 10 Operating System
   Anaconda Spyder : Python IDE

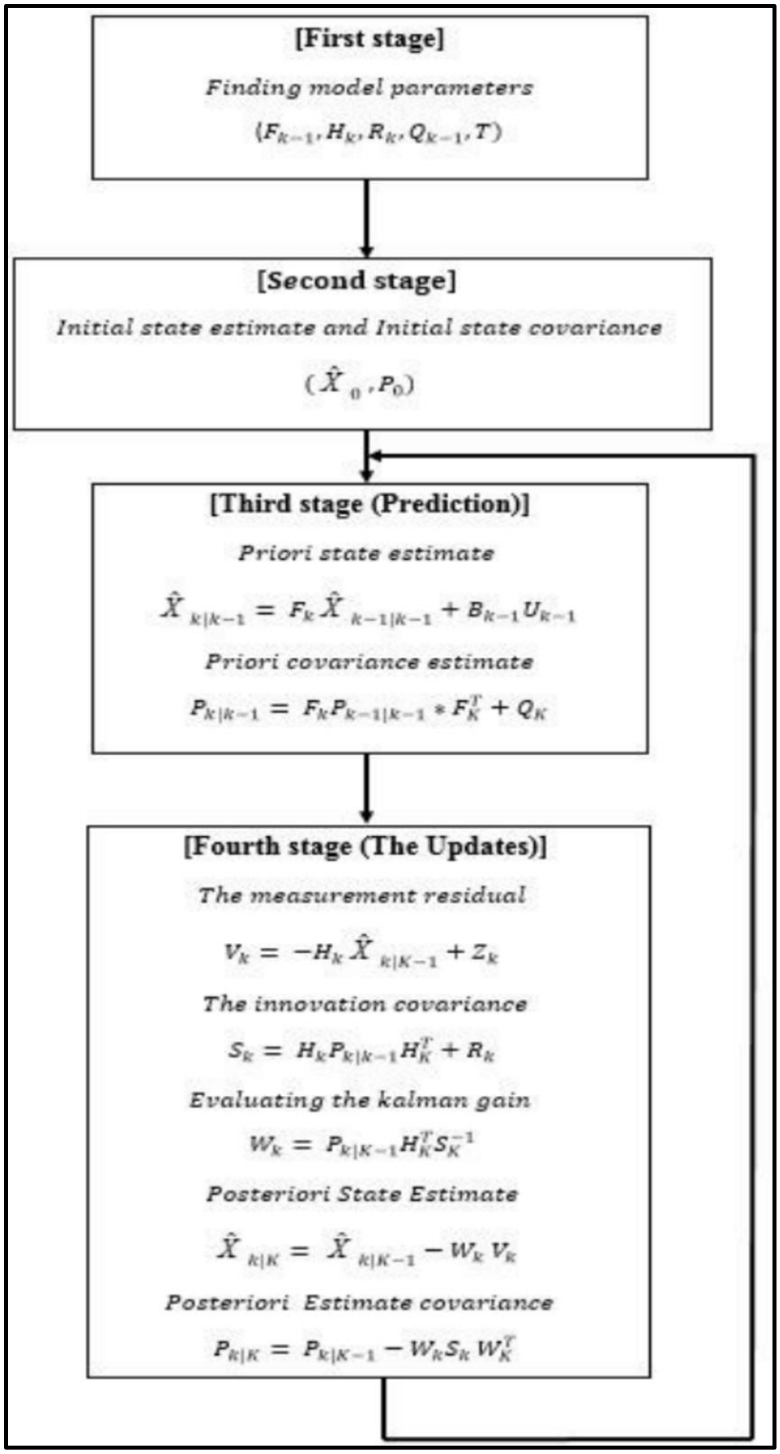
5. Approach

The approach of the Problem statement to implement a Kalman filter by supplying equations in an iterative function that define premises of the Kalman Filter.
The following variables were considered during the implementation

- Predicted State Estimate
- Predicted Error Covariance
- Measurement Pre Fit Residual
- Pre Fir Residual Covariance
- Optimal Kalman Gain
- Updated State Estimate
- Updated Estimated Covariance
- Measurement Post Fit Residual

The above mentioned variables were iterated in the Kalman Function and the output was plotted on different subplots subsequently

- True Values
- Prediction Values
- Observations

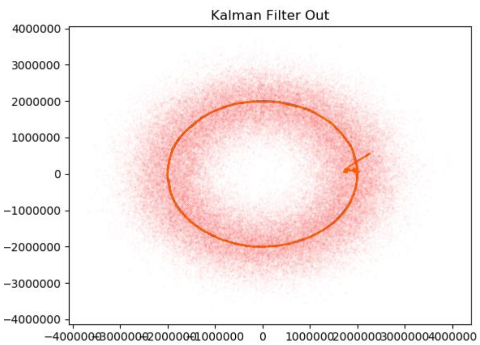


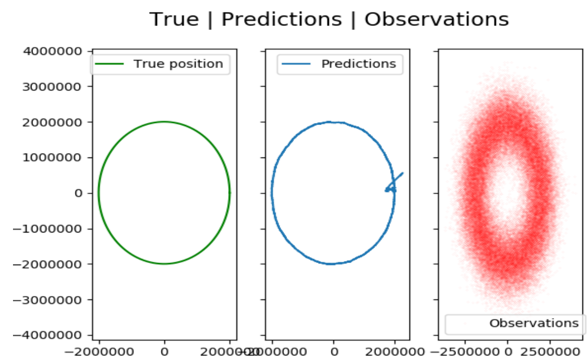
3. Kalman Filter Flowchart

6. Observations

CASE 1

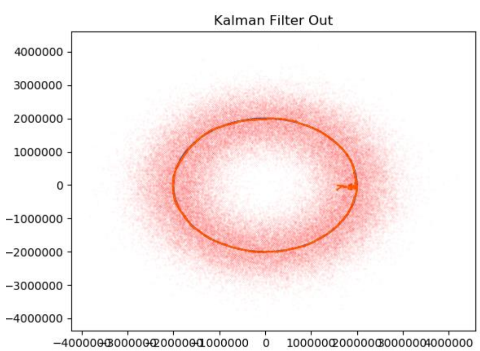
Rationale	Noise (Centre Distribution)	Covariance Diagonal Values
PosStd	loc = 0.0	Power = 2
VelStd	loc = 0.0	Power = 2

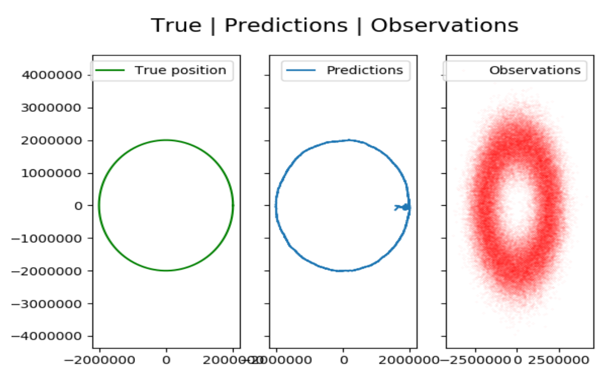




CASE2

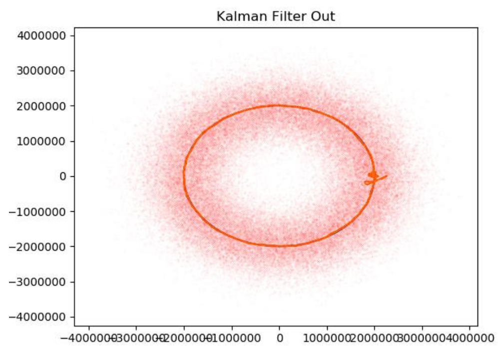
Rationale	Noise (Centre Distribution)	Covariance Diagonal Values
PosStd	loc = 1.0	Power = 2
VelStd	loc = 1.0	Power = 2

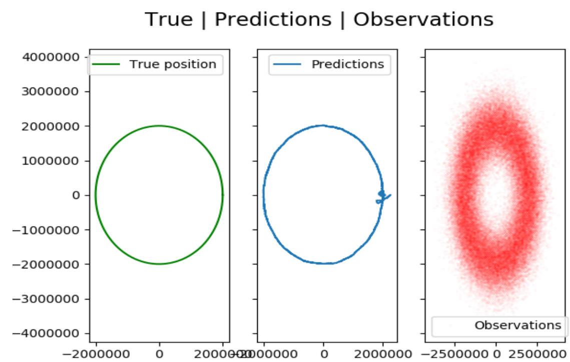




CASE3

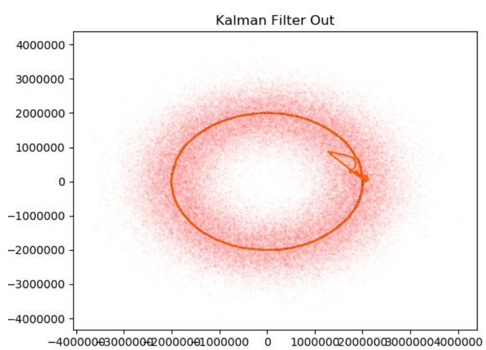
Rationale	Noise (Centre Distribution)	Covariance	Observation Model
PosStd	loc = 2.0	Power = 2	Power = 2/4000000
VelStd	loc = 2.0	Power = 2	Power = 2/4000000

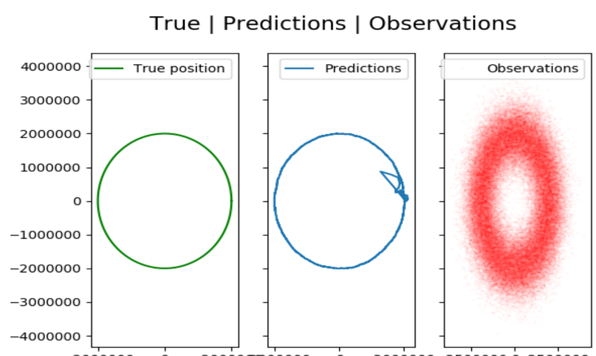




CASE4

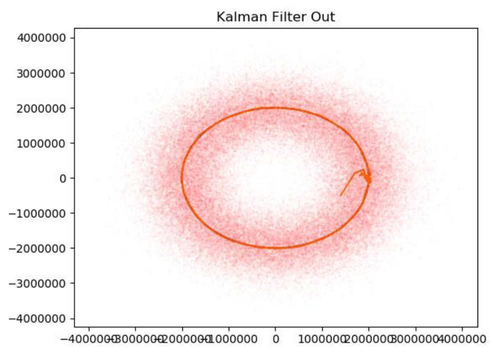
Rationale	Noise (Centre Distribution)	Covariance	Observation Model
PosStd	loc = 0.0	Power = 3	Power = 2/4000000
VelStd	loc = 0.0	Power =3	Power = 2/4000000

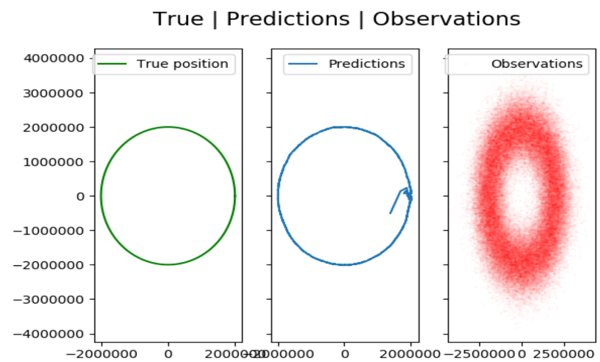




CASE 5

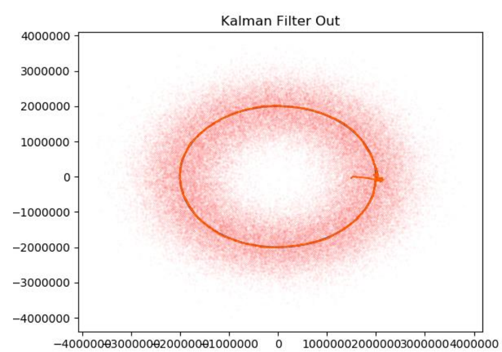
Rationale	Noise (Centre Distribution)	Covariance	Observation Model
PosStd	loc = 0.0	Power = 2	Power = 2/6000000
VelStd	loc = 0.0	Power =2	Power = 2/6000000

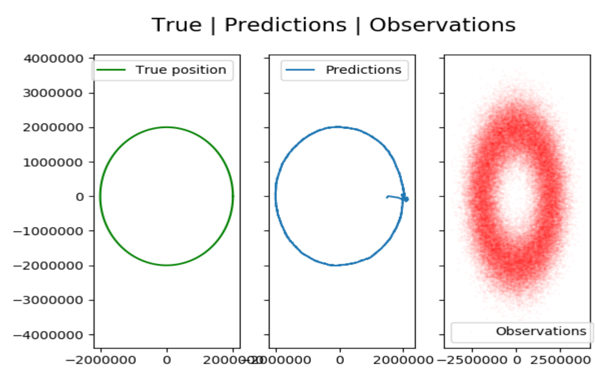




CASE 6

Rationale	Noise (Centre Distribution)	Covariance	Observation Model
PosStd	loc = 0.0	Power = 2	Power = 2/8000000
VelStd	loc = 0.0	Power =2	Power = 2/8000000





7. Discussions

6 case studies were recorded and plotted for different values of noise, covariance matrix and Observation matrices.

It can be seen that the prediction plots change considerably as compared to true values when we change the different parameters of iterations. This shows the sensitivity of the Kalman filter towards minor changes in the parameters and how does it vary with the considerations we have with the prediction values.

While certain values tend to have a positive shift towards the trues values, the others values seems to stray away. These trends could be noted further with more case studies to analyze the accuracy and precision of the implemented Kalman Filter to have a more efficient and better designed and modelled system that uses the Linear Quadratic Estimation for an iterative approach to solve the task. The visualized plots definitely help in estimating the error and sight procedures to rectify and optimize the best solution towards the problem statement.

8. Annexure

9. References

- https://en.wikipedia.org/wiki/Kalman_filter
- https://www.youtube.com/watch?v=mwn8xhgNpFY
- https://www.youtube.com/watch?v=CaCcOwJPytQ
- https://stackoverflow.com/questions/16798771/numpy-dot-product

10. Acknowledgement

I extend my heartfelt gratitude to Mr. Indrek Sünter for hosting the course Space Systems and providing vital resources in the field of pattern recognition and machine learning to be able to compile this report. It has been a fruitful research to advance on the ways of learning methods in machine learning to understand the underlying concepts of the discipline. Special mention to Robert Mark for the assignment to better understand the working principle of Kalman Filters.