Report - CH5120 Course Project 1

EXTENDED KALMAN FILTER

Ridhi Puppala - ME15B133 Akash - ME15B005

Introduction

Kalman Filter is very useful in the case of model mismatch and noisy measurements or data. It is the main tool for state estimation in the above mentioned cases. It is based on state space representation for mathematical model of real world system.

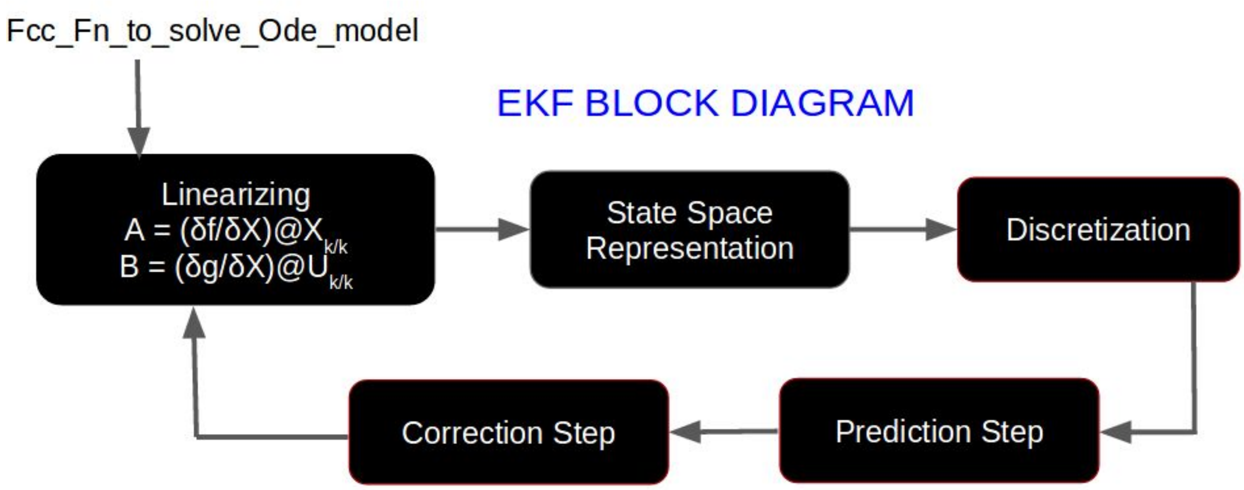
In estimation theory, the **extended Kalman filter (EKF)** is the nonlinear version of the Kalman filter which linearizes about an estimate of the current mean and covariance. In the case of well defined transition models, the EKF has been considered.

Formulation

Consider the following nonlinear system, described by the difference equation and the observation
model with additive noise:

$$X_k = f(x_{k-1}) + g(u_{k-1}) + w_{k-1} \qquad w_{k-1}\ \varepsilon\ N[0,\sigma]$$
$$Y_k = h(x_k) + v_k \qquad v_k\ \varepsilon\ N[0,\sigma]$$

The stochastic noise in state estimation(w_k) and measurement(v_k) are approximated to lie on Gaussian distribution curves with a zero mean and a variance σ.
The initial state x_0 is a random vector with known mean $\mu_0 = E[x_0]$ and covariance $P_0 = E[(x_0 - \mu_0)(x_0 - \mu_0)^T]$.



Prediction and Correction

After linearizing of the continuous and time invariant FCC model given, we discretize to implement prediction and correction steps through the following equations:

Prediction Step:

$$\mathbf{X}_{k+1/k} = A\mathbf{X}_{k/k} + B\mathbf{U}_k$$
$$P_{k+1/k} = AP_{k/k}A^T + Q$$

Where $A = (\delta f/\delta X)@X_{k/k}$ $B = (\delta g/\delta X)@U_{k/k}$ and Q is covariance matrix of w_k

Correction Step:

$$\mathbf{X}_{k+1/k+1} = \mathbf{X}_{k+1/k} + KG(\mathbf{Y}_{k+1} - \mathbf{C}\mathbf{X}_{k+1/k})$$
$$P_{k+1/k+1} = (I - KG(C))P_{k+1/k}$$

Where Kalman Gain $KG = P_{k+1/k}C^T(CP_{k+1/k}C^T + R)^{-1}$ and R is covariance matrix of v_k

In the given model, measurement values are provided, so C is taken as [0 0 1] for case 1, with only T_{rg} measured and [1 0 0;0 0 1] for case 2 with C_{rc} and T_{rg} both measured.

Tuning - KF PARAMETERS

Selection of Q and R : Q is the noise covariance matrix and R is the measurement noise covariance matrix. Kalman filter is sensitive to errors in Q and R and its output can be unacceptable if errors are large. Matrix R is much easy to ascertain, because the measurement equipment often has some error characteristics.The measurement noise covariance **R** is estimated from knowledge of predicted observation errors. Here we chose R matrix to be a diagonal matrix containing the variance of the measurements as the entries. We chose it to be a diagonal matrix to achieve decoupling of error contributions. The entries were fixed on a trial and error basis was chosen to have a higher value compared to Q because we depend more on the predictions.

Selection of P Matrix: P_0 is your initial **state** covariance. That expresses how much you know about the initial estimate of the state x_0 .If we have no idea it is initially set to a high value. Since we are depending more on our predictions rather than the measurements generated from the model. We chose it to have a small value.

MATLAB Tools Used

jacobianest (@fun,var) : this function was used to calculate A_C and B_C at each timestep.It takes the function handle and the variable as the input and then it will return the jacobian matrix along with the error.

System name = ss(A, B, C, D) : This function defines a LTI system with the parameters A,B,C and D.

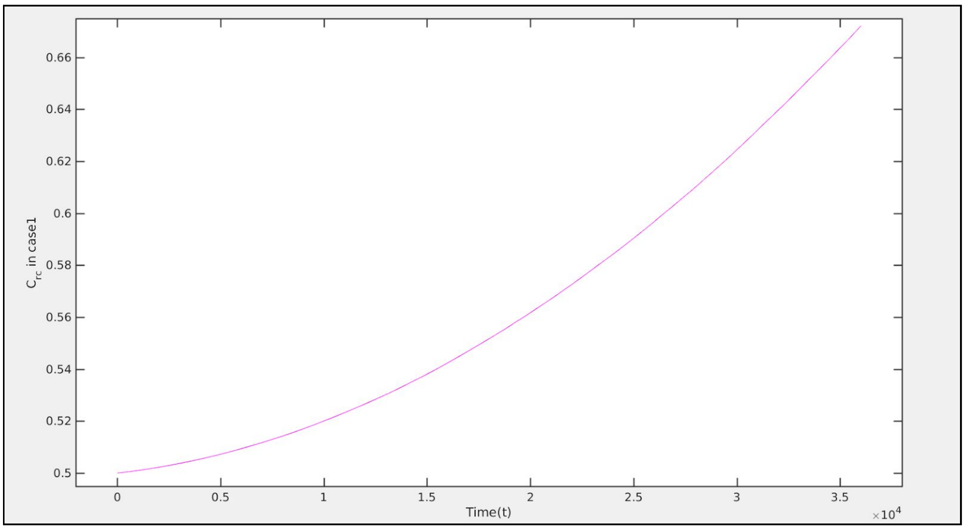
c2d(sys, Ts) : This function is used to discretize a continuous model to a discrete model with a given sample time of Ts.

ssdata(sys) : This function is used to extract the model parameters from a system.In our code it was used to extract the model parameters after it was discretized

Case 1 Results

Case 1: When only **one state** is measured (**T_rg**)

Fig1 : C_{rc} vs Time



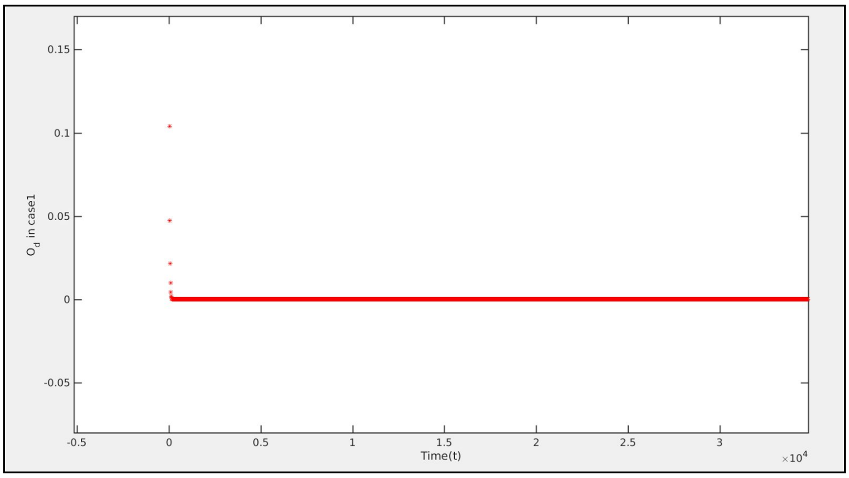
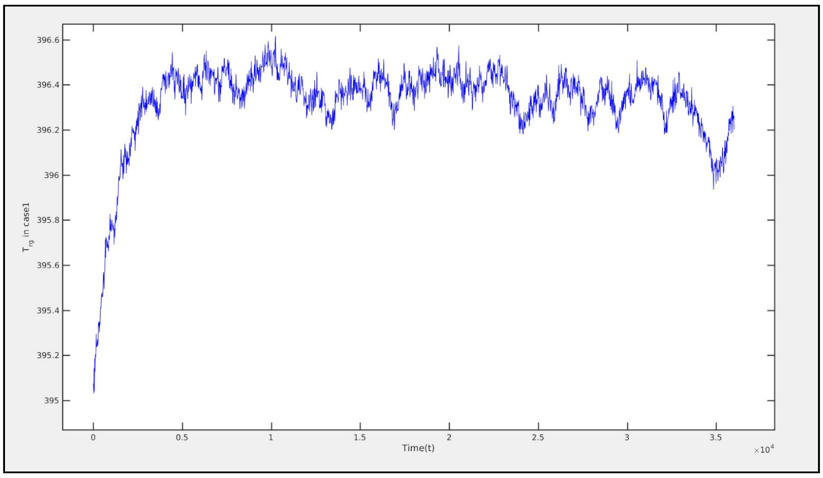


Fig2 : O_d vs Time

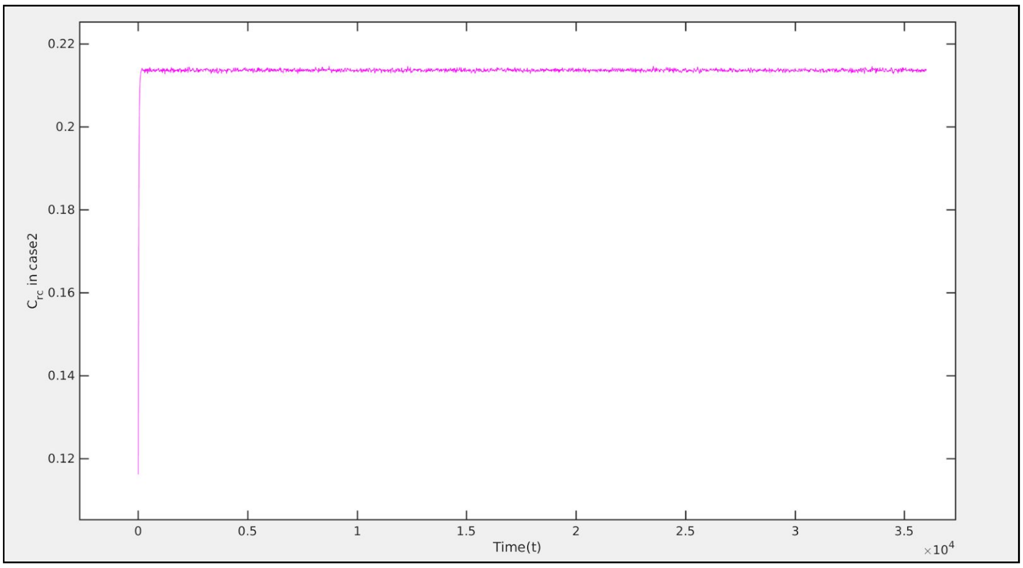
Fig3 : T_{rg} vs Time



Case2 Results

Case 2: When two states are measured (**T_rg and C_{rc}**)

Fig4: C_{rc} Vs Time



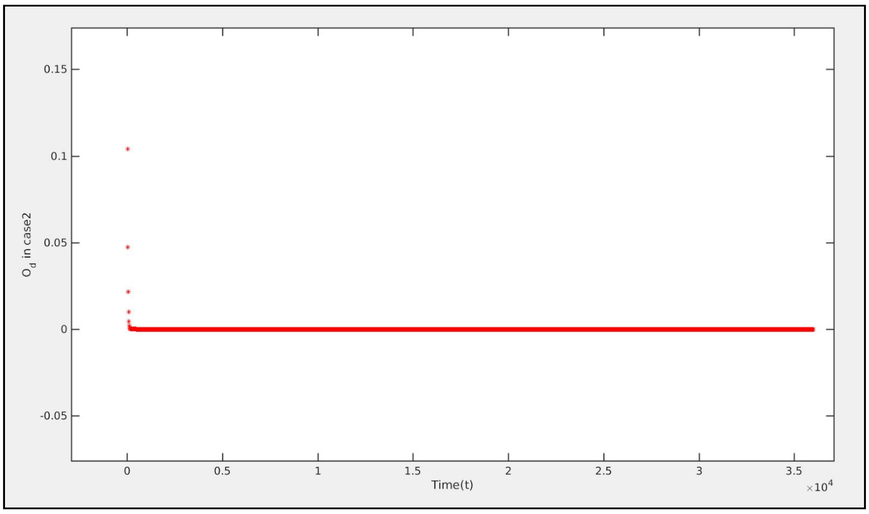
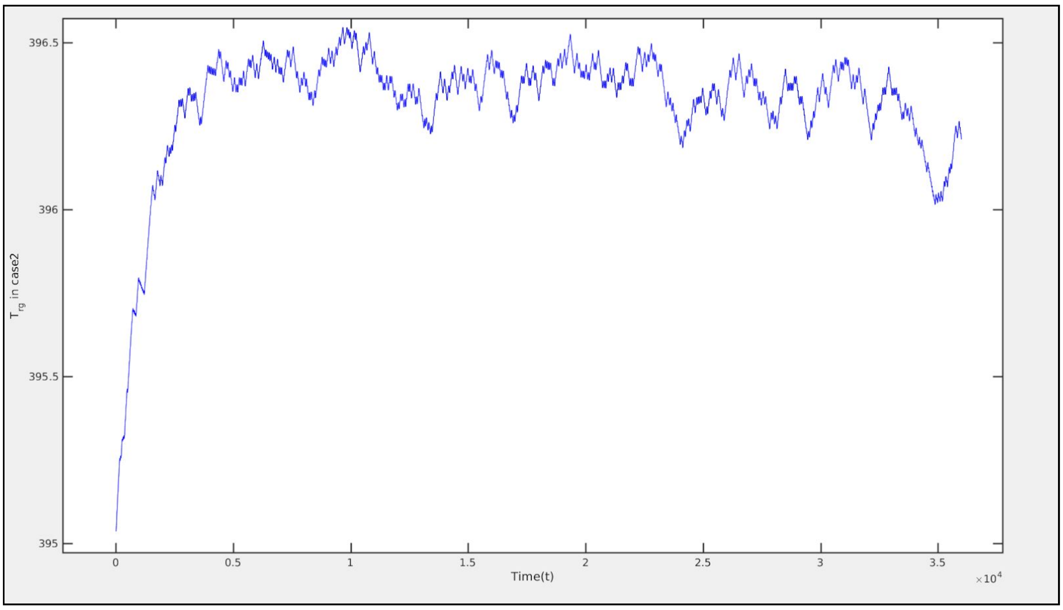


Fig5: O_d Vs Time

Fig6: T_{rg} Vs Time



EXTENDED KALMAN FILTER

General Observations

EKF and Calculating the Jacobian Matrix : Since the FCC model is a non-linear. We will have to linearize it at each instant around the given state. The A_C and B_C was calculated using the Jacobianest function. The Extended Kalman Filter is a Local Asymptotic Observer for Nonlinear Discrete-Time Systems. As a result the model is controllable at each time step.
But the A_C and B_C matrix usually had non real components. This was because the because of the improper selection of the Q matrix. Therefore by iteratively changing the values of Q a proper a real A_C and B_C matrix was attained.

State Space Models from Matlab: After calculating the A_C and B_C matrix, the analysis of this model was done using MATLAB control systems toolbox. After feeding these data into the ss (A,B,C,D) function. It was then converted into discrete model by using the c2d conversion function. A common error that frequently occurred was ‘The value of the "a" property must be a numeric array without any Inf's or NaN's. ‘.This was due to the improper selection of Q matrix due to which some of the values returned from the Jacobian matrix were Inf or NaN.
Another error that was frequently obtained was ‘The "c2d" command failed because some unstable dynamics with large positive real parts caused overflow. Try reducing the sampling period for discretization.’ This error was observed when very small or very high values of the Q matrix was used. Since reducing the sampling time was not an option this error was overcome by using different values of Q

Observation from results

- Even if Q was changed to large values, there was no effect on T_{rg}, only change observed was frequency of noise of T_{rg} varied with diagonal entries of Q matrix.
- While C_{rc} and O_d was largely dependent on entries in Q matrix because the order of magnitude of these stats were negligible compared to that of T_{rg}
- When $P_{0/0}$ was zero for some values of Q, the ss() function gave error due to imaginary values of state space parameters
- X0 effect was negligible, as by the end of first 5 iterations, it would reach the same value irrespective of other parameters and tuning
- T_{rg} is subject to a lot of noise while estimation. Frequency of noise and estimated value changes with Tuning. Based on this optimal tuning values were selected as O_d and C_{rc} remained unaffected most of the time, showing similar characteristic
- O_d in both cases approaches zero in the estimation. This behavior shows that is less observable compared to other states. Even while tuning it was found that O_d converges to zero within no time. Estimation might be inaccurate.

Note: Figures and Code for both cases can be found in respective folders