Experiment No. 7

AIM: Design a Kalman filter for object tracking (navigation).

Software to be used: PYTHON

Theory:

Introduction to Kalman Filter: In 1960, R.E. Kalman published his famous paper describing a recursive solution to the discrete-data linear filtering problem. The Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) solution of the least-squares method. The filter is very powerful in several aspects: it supports estimations of *past*, *present*, and even *future* states (data/signal sample), and it can do so even when the *precise nature of the modelled system is unknown*.

The Kalman filter addresses the general problem of trying to estimate the state $x \in \Re^n$ of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$x_{k+1} = A_k x_k + Bu_k + w_k, \tag{1}$$

with the measurement $z \in \Re^m$ that is

$$z_k = H_k x_k + v_k. \tag{2}$$

The matrix A_k in the difference equation (1) relates the state x at time step k to the state at step $k+1$, in the absence of either a driving function or process noise. The matrix B relates the control input to the state x. The matrix H_k in the measurement equation (2) relates the state x_k to the measurement z_k. The random variables w_k and v_k represent the process and measurement noise (respectively) characterized as

$$p(w) \sim N\left(0, Q\right), \text{ and } p(v) \sim N\left(0, R\right),$$

Here, we shall make the following assumptions:

i) $\{w_k\}$ and $\{v_k\}$ are individually white stochastic processes (uncorrelated from instance to instance, as being also stationary, and usually having zero mean), and independent to each other.

$$E\left[v_k v_l^*\right] = \begin{cases} E\left[v_k\right] E\left[v_l^*\right], & \text{by the whiteness assumption} \\ 0, & \text{by the zero mean assumption} \end{cases}$$

ii) $\{w_k\}$ and $\{v_k\}$ are individually zero mean, Gaussian processes with known covariances. Here, it is to be noted that the probability density of a random variable v is entirely determined by the mean $m = E\left[v\right]$ and covariance $R = E\left[\left(v-m\right)\left(v-m\right)^*\right]$. When has dimension n and R is non-singular, the probability density function is

$$p_V(v) = \frac{1}{(2\pi)^{n/2} \left|R\right|^{1/2}} \mathrm{e}^{-\frac{1}{2}\left[(v-m)^{\mathrm{H}} R^{-1} (v-m)\right]}.$$

The Kalman filter estimates a process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of (noisy) measurements. As such, the equations for the Kalman filter fall into two groups: time update equations and measurement update equations. The time update equations are responsible for projecting forward (in time) the current state and error covariance estimates to obtain the a priori estimates for the next time step. The measurement update equations are responsible for the feedback—i.e. for incorporating a new measurement into the a priori estimate to obtain an improved a posteriori estimate.

The time update equations can also be thought of as predictor equations, while the measurement update equations can be thought of as corrector equations. Indeed, the final estimation algorithm resembles that of a predictor-corrector algorithm for solving numerical problems as shown below in Fig. 1.

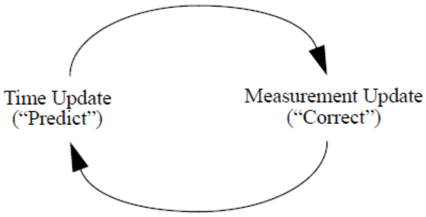

Fig. 1 The discrete Kalman filter cycle.

We define $\hat{x}_k^-$ (note the "super minus") to be our *a priori* state estimate (prediction) at step k given knowledge of the process prior to step k, and $\hat{x}_k \in \Re^n$ to be our *a posteriori* state estimate at step k given measurement z_k. We can then define *a priori* and *a posteriori* estimate errors as

$$e_k^- = x_k - \hat{x}_k^-, \text{ and}$$

$$e_k = x_k - \hat{x}_k.$$

The *a priori* estimate error covariance is then

$$P_k^- = E\left[e_k^- e_k^{-T}\right],$$

and the *a posteriori* estimate error covariance is

$$P_k = E\left[e_k e_k^T\right].$$

The specific equations for the time and measurement updates (to perform Kalman filtering) are given below.

Discrete Kalman filter time update equations:

$$\hat{x}_{k+1}^- = A_k x_k + Bu_k \qquad (3)$$

$$\mathrm{P}_k^- = A_k P_k A_k^T + Q_k \qquad (4)$$

Discrete Kalman filter measurement update equations:

$$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k\right)^{-1} \qquad (5)$$

$$\hat{x}_k = \hat{x}_k^- + K\left(z_k - H_k \hat{x}_k^-\right) \qquad (6)$$

$$P_k = \left(I - K_k H_k\right) P_k^- \qquad (7)$$

After each time and measurement update pair, the process is repeated with the previous a posteriori estimates used to project or predict the new a priori estimates. This recursive nature is one of the very appealing features of the Kalman filter—it makes practical implementations much more feasible than (for example) an implementation of a Weiner filter which is designed to operate on all of the data directly for each estimate.

In the actual implementation of the filter, each of the measurement error covariance matrix R_k and the process noise Q_k might be measured prior to operation of the filter. In the case of the measurement error covariance R_k , we should generally be able to take some off-line sample measurements in order to determine the variance of the measurement error. In the case of Q_k , often times the choice is less deterministic. For example, this noise source is often used to represent the uncertainty in the process model (1). Sometimes a very poor model can be used simply by "injecting" enough uncertainty via the selection of Q_k . Often times, superior filter performance (statistically speaking) can be obtained by "tuning" the filter parameters Q_k and R_k . The tuning is usually performed off-line. In closing, we note that under conditions where R_k and Q_k are constant, both the estimation error covariance P_k and the Kalman gain K_k will stabilize quickly and then remain constant.

A large value of Q_k : More uncertainty in prediction of the state

A large value of R_k : Noisy measurement and the *a posteriori* estimation would depends less on measurements.

Example 1: A Two-State Filter: Friedland's Model for tracking in 1-D

Consider an aircraft or similar space vehicle with constant velocity perturbed by a zero mean random acceleration. The position of the vehicle is assumed to be measured by a track-while-scan radar sensor at uniform sampling intervals of time T seconds and all measurements are noisy. The observation errors have zero mean and are uncorrelated. The problem is to obtain the optimum estimates of position x and velocity v of the vehicle. For each coordinate, the vehicle dynamics (process) is assumed to be described by

Dynamic Model

$$x_{k+1} = x_k + v_k T + \frac{1}{2} a_k T^2$$

$$v_{k+1} = v_k + a_k T,$$

where, a_k denotes the acceleration acting on vehicle or aircraft at scan k . In that given model, the acceleration a_k is assumed to be a random constant between successive observations with zero mean and uncorrelated with its values at other intervals, i.e.,

$$E\left[a_k\right] = 0,$$
$$E\left[a_k^2\right] = \sigma_a^2 = const.$$

In the vector-matrix form, the vehicle dynamics may be written as

$$X_{k+1} = AX_k + Bu_k + W_k$$

with

$$X_k = \begin{bmatrix} x_k \\ v_k \end{bmatrix}, \; u_k = \left[a_k\right], \qquad A = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \text{ and } B = \begin{bmatrix} \dfrac{T^2}{2} \\ T \end{bmatrix}.$$

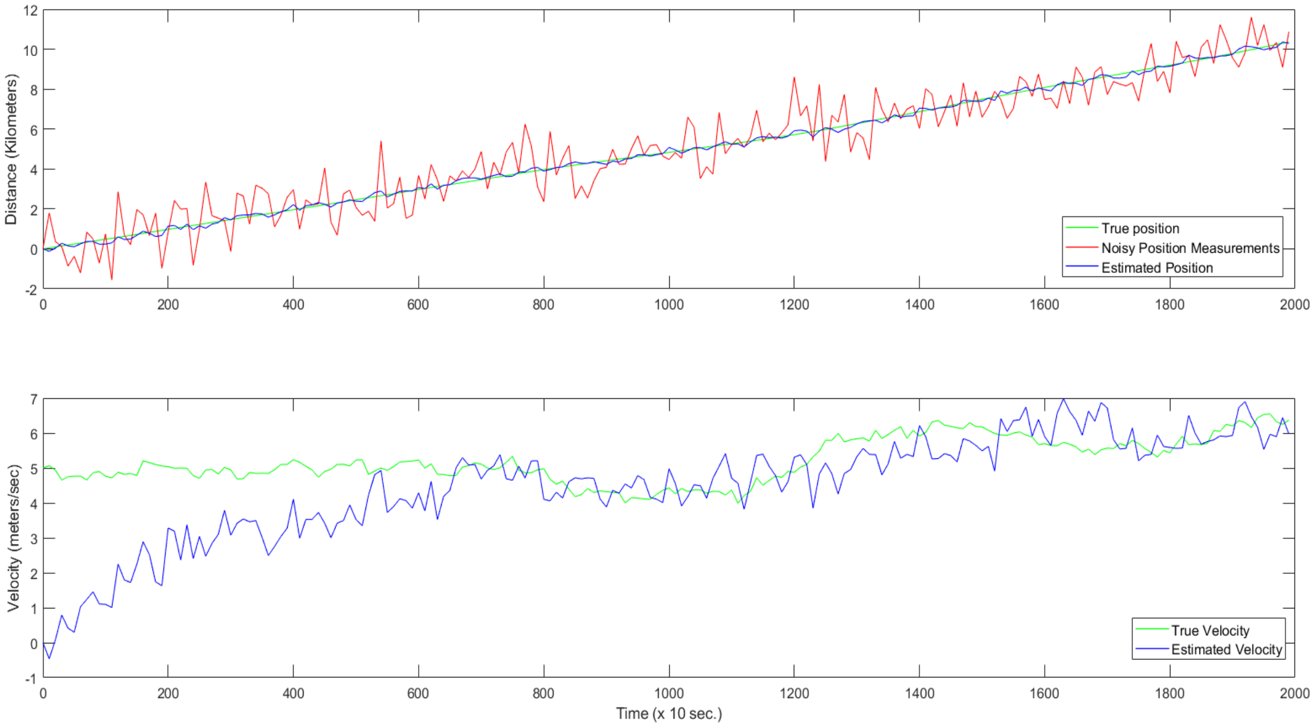

Fig. 2 Illustration of an example for two-state Kalman filter

Measurement Model:

The position of the vehicle is assumed to be measured by a radar at uniform intervals of time T seconds and each observation z_k is noisy. The measurement equation is given by

$$z_k = HX_k + v_k, \quad \text{with } H = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

where x_k is the true position at scan k, and v_k is random noise corrupting the measurement at scan k.

Procedure:

1. Define the initial position x_0, the initial velocity v_0, and the scan time T.
2. Now compute the true values x_k of position and velocity v_k using the expression mentioned in the dynamic model. In the dynamic model, a_k is assumed to be a random variable (with Gaussian distribution), therefore for every time step k, generate a_k using *randn(1)* function that will be used to generate true position and true velocity values. These true values are to be used for comparison purpose and to simulate measurements.
3. Now, define measurement noise variance as σ_x^2. Simulate the noisy measurements z_k by adding Gaussian noise of variance σ_x^2 in the true positions x_k. *Importantly, this variance is generally defined by sensor manufacturer or can be estimated by experiments in known environment.*
4. The above-mentioned steps 1 – 4 are basically for simulating true and noisy sample values of velocity and position. Now the Kalman filtering starts!!!!!
5. Define the process noise covariance Q_k. For example,

$$Q_k = \begin{bmatrix} \sigma_{x_k}^2 & 0 \\ 0 & \sigma_{v_k}^2 \end{bmatrix}$$

A lower value of Q_k (e.g., 0.0001) *signifies that the prediction error is very small, and a higher value of* Q_k (e.g., 1.5) *signifies high uncertainty in prediction. Its exact value can't be obtained; thus, it is selected empirically.*

6. Define the measurement noise covariance $R_k = \left[\sigma^2_{\dot{x}_k} \right]$. The diagonal matrix R_k shall be of size $m \times m$, where m is the number of observations.
   A lower value of R_k (e.g., 0.0001) *signifies that the measurement contains less noise, and therefore, the Kalman filter will rely on the measurement received for estimation, and a higher value of* R_k (e.g., 1.5) *signifies large error in measurements, and therefore, the Kalman filter will ignore measuremenst mostly to estimate the state. It can be selected based on information about sensor performance and can also be selected empirically based on environment knowledge.*
7. Initialize the *a posteriori* estimate error covariance P_k (of size equal to Q_k) with either zeros values (most preferred) or with random values.
8. Now, implement equations (3) – (7) sequentially, where the estimated state variable can be obtained from equation (6).

Example 2: A Three-State Filter: Friedland's Model for tracking in 1-D

Each position coordinate of the moving vehicle is assumed to be described by the following equations of motion:

Dynamic Model

$$X_{k+1} = AX_k + W_k,$$

where

$$A = \begin{bmatrix} 1 & T^2 & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \text{ and } X_k = \begin{bmatrix} x_k \\ v_k \\ a_k \end{bmatrix}.$$

w_k is a stationary white noise process with covariance matrix Q_k given by

$$\begin{aligned} Q_k &= E\left[W_k W_k^T \right] \\ &= qT \begin{bmatrix} T^4/20 & T^3/8 & T^2/6 \\ T^3/8 & T^2/3 & T/2 \\ T^2/6 & T/2 & 1 \end{bmatrix}, \end{aligned}$$

where q is the spectral density of the continuous white noise change in acceleration process and is given by $q = \sigma_a^2 T,$

where, σ_a^2 is the variance of the change in acceleration noise.

Measurement Model:

The measurement model is assumed to be described by

$$Z_k = HX_k + V_k,$$

where

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \text{and } Z_k = \begin{bmatrix} x_k^{'} \\ v_k^{'} \end{bmatrix},$$

with $x_k^{'}$ and $v_k^{'}$ are the measured position and velocity at scan k. The variable V_k is the stationary white noise process with covariance matrix R_k given by

$$\begin{aligned} R_k &= E\left[V_k V_k^T\right] \\ &= \begin{bmatrix} \sigma_{x'}^2 & 0 \\ 0 & \sigma_{v'}^2 \end{bmatrix}. \end{aligned}$$

Procedure:

1. Follow the above-mentioned steps 1 – 4 in example-1 to simulate signals. Additionally, generate measurements for velocity by adding Gaussian noise of variance $\sigma_{v'}^2$ in the true velocity values v_k.
2. Then, define Q_k as described above, initialize P_k, and then perform the filtering.

Example 3: A 3-D Model for tracking

In three dimensions (3-D), the vehicle dynamics may be represented by the state vector is given by

$$X_k = \begin{bmatrix} x_k & y_k & z_k & v_k^x & v_k^y & v_k^z \end{bmatrix}^T$$

The target dynamical model can be represented in the Cartesian coordinate as

$$X_{k+1} = A_k X_k + Bu_k + W_k$$

where,

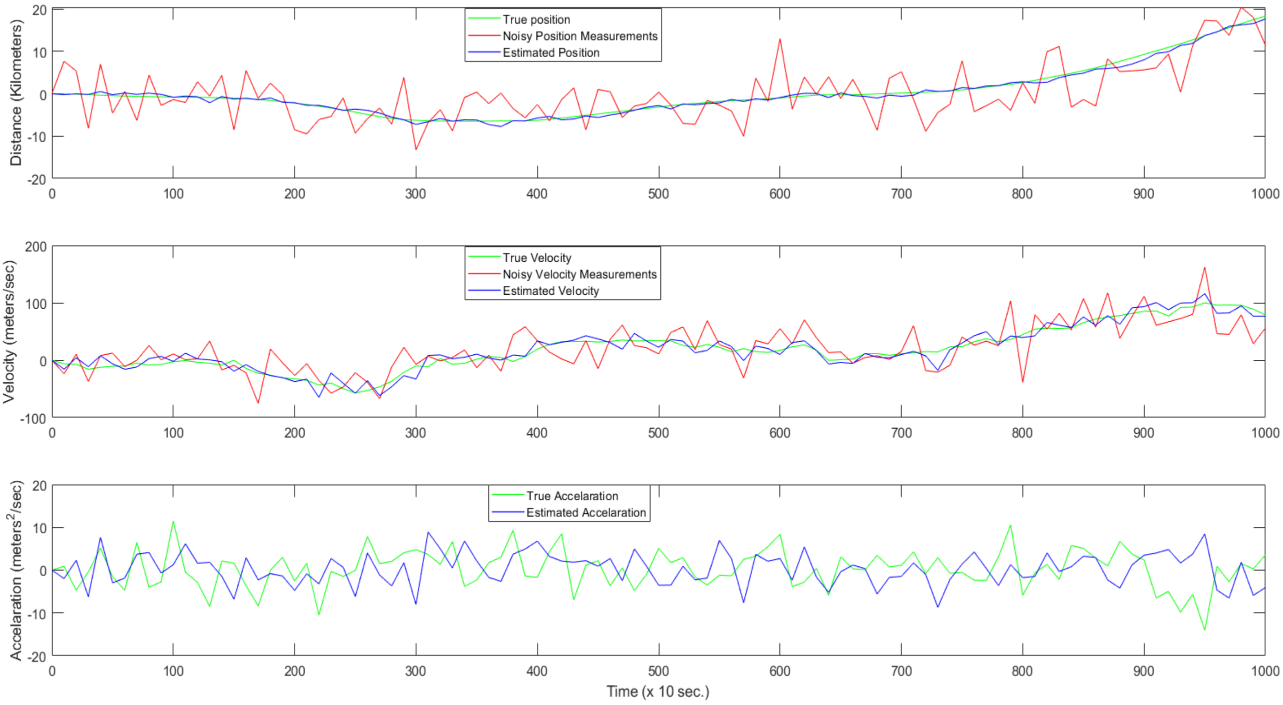

Fig. 3 An example depicting three-states estimation using a Kalman filter

$$A_k = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} T^2/2 & 0 & 0 \\ 0 & T^2/2 & 0 \\ 0 & 0 & T^2/2 \\ T & 0 & 0 \\ 0 & T & 0 \\ 0 & 0 & T \end{bmatrix}, \ u_k = \begin{bmatrix} a_k^x \\ a_k^y \\ a_k^z \end{bmatrix},$$

$$Q_k = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_z^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{v_x}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{v_y}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{v_z}^2 \end{bmatrix}.$$

Besides, the Q matrix can also be defined as per the model characteristics. The measurement equation is given by

$$Z_k = H_k X_k + V_k,$$

where,

$$Z_k = \begin{bmatrix} x_k^{'} \\ y_k^{'} \\ z_k^{'} \end{bmatrix}, \ H_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \ R_k = \begin{bmatrix} \sigma_{x'}^2 & 0 & 0 \\ 0 & \sigma_{y'}^2 & 0 \\ 0 & 0 & \sigma_{z'}^2 \end{bmatrix}.$$

with $x_k^{'}$, $y_k^{'}$, and $z_k^{'}$ are the measured positions in x, y, and z directions, respectively, at scan k. The variable V_k is the stationary white noise process with covariance matrix R_k.

Procedure:

1. Define the initial positions $[x_0, y_0, z_0]^T$ and the initial velocities $\left[v_0^x, v_0^y, v_0^z\right]^T$ for the three dimensions x, y, and z, and the scan time T.
2. Now calculate the true values $[x_k, y_k, z_k]^T$ of positions and velocities $\left[v_k^x, v_k^y, v_k^z\right]^T$ using the expressions mentioned in the dynamic model. In the dynamic model, $a_k = \left[a_k^x, a_k^y, a_k^z\right]$ are assumed to be a random variables (with Gaussian distribution), therefore for every time step k, generate a_k using *randn(3,1)* function that will be used to generate true position and true velocity values. These true values are to be used for comparison purpose and to simulate measurements as well.
3. Now, define measurement noise covariance matrix R_k. Simulate the noisy measurements z_k by adding Gaussian noise (of some defined variances σ_x^2, σ_y^2, and σ_z^2) in the true positions $[x_k, y_k, z_k]^T$.
4. Define the process noise covariance Q_k.
5. Now, implement equations (3) – (7) sequentially, where the estimated state variable can be obtained from equation (6).

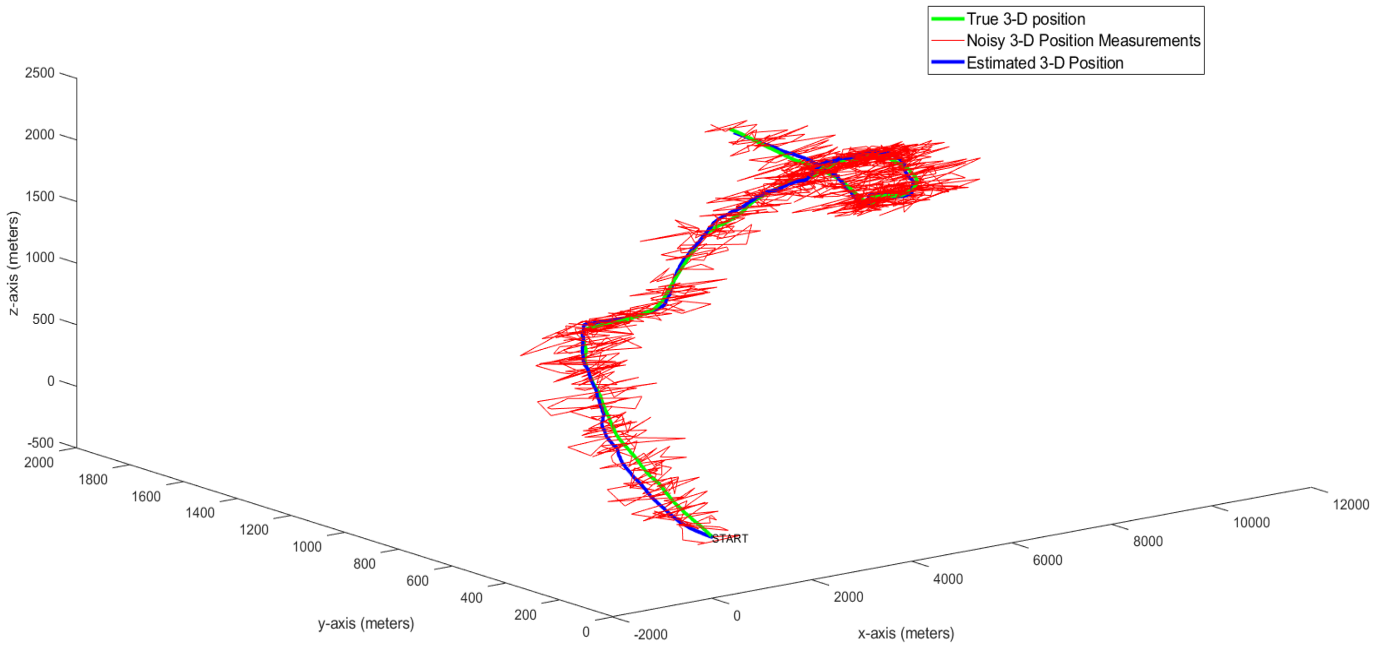

Fig. 4 An example illustrating 3-D position estimation using a Kalman filter