Kalman Filtering: Case Study 2
(3D Location Tracking)

1 Problem Statement

Our goal is to track the location (and velocity) of a moving object, e.g. a ball, in a 3-dimensional space. We will allow gravity to act on the ball and the initial position and velocities are assumed to be known. We will be using noisy location estimates using a (simulated) sensor. The objective is to estimate the true location (and velocity) of the ball in 3D space.

2 State Space

We keep track of the following:

1. x_1: location in the x_1 direction.

2. x_2: location in the x_2 direction.

3. x_3: location in the x_3 direction.

4. v_1: velocity in the v_1 direction.

5. v_2: velocity in the v_2 direction.

6. v_3: velocity in the v_3 direction.

7. a_1: velocity in the a_1 direction.

8. a_2: velocity in the a_2 direction.

9. a_3: velocity in the a_3 direction.

Therefore, let the state vector be represented as:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ v_1 \\ v_2 \\ v_3 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

3 Kalman Equations and Dynamic Model

3.1 Kalman Equations

Reproduced from Wikipedia (below) we have the systems of equations (in matrix format) that need to be solved as part of the Kalman Filtering algorithm. Our challenge is to adapt the problem setting to result in the model that satisfies the form of these equations which will allow us to use Kalman Filtering to track the ball's position and velocity.

The Kalman filter model assumes the true state at time k is evolved from the state at (k - 1) according to:

$$\mathbf{x}_k = \mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k$$

where,

- A_k is the state transition model which is applied to the previous state X_{k-1};

- B_k is the control-input model which is applied to the control vector u_k (which is not applicable in our setting);

- w_k is the process noise which is assumed to be drawn from a zero mean multivariate normal distribution with covariance Q_k:

$$\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}_k) \mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}_k)$$

At time k an observation (or measurement) z_k of the true state x_k is made according to:

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k$$

where H_k is the observation model which maps the true state space into the observed space and v_k is the observation noise which is assumed to be zero mean Gaussian white noise with covariance R_k.

$$\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k)$$

The initial state, and the noise vectors at each step $\{x_0, w_1, \ldots, w_k, v_1 \ldots v_k\}$ are all assumed to be mutually independent.

3.2 Filtering

The state of the Kalman filter is represented by two variables:

$\hat{\mathbf{x}}_{k|k}$, the a posteriori state estimate at time k given observations up to and including at time k;

$\mathbf{P}_{k|k}$, the a posteriori error covariance matrix (a measure of the estimated accuracy of the state estimate).

The Kalman filter can be written as a single equation, however it is most often conceptualized as two distinct phases: "Predict" and "Update". The predict phase uses the state estimate from the previous timestep to produce an estimate of the state at the current timestep. This predicted state estimate is also known as the a priori state estimate because, although it is an estimate of the state at the current timestep, it does not include observation information from the current timestep. In the update phase, the current a priori prediction is combined with current observation information to refine the state estimate. This improved estimate is termed the a posteriori state estimate.

Typically, the two phases alternate, with the prediction advancing the state until the next scheduled observation, and the update incorporating the observation.

3.2.1 Predict

1. Predicted (a priori) state estimate: $\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k$

2. Predicted (a priori) estimate covariance: $\mathbf{P}_{k|k-1} = \mathbf{A}_k \mathbf{P}_{k-1|k-1} \mathbf{A}_k^{\mathrm{T}} + \mathbf{Q}_k$

3.2.2 Update

1. Innovation or measurement residual: $\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}$

2. Innovation (or residual) covariance: $\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k$

3. Optimal Kalman gain: $\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1}$

4. Updated (a posteriori) state estimate: $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k$

5. Updated (a posteriori) estimate covariance: $\mathbf{P}_{k|k} = (I - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}$

3.3 3D Motion: Model

In our setting, we have the following:

3.3.1 Dynamic Matrix, A_k

The Dynamic Matrix, A_k, is calculated from the dynamics of the motion:

$$x_{1,k+1} = x_{1,k} + v_{1,k} \cdot \Delta t + \frac{1}{2} a_{1,k} \cdot \Delta t^2$$

$$x_{2,k+1} = x_{2,k} + v_{2,k} \cdot \Delta t + \frac{1}{2} a_{2,k} \cdot \Delta t^2$$

$$x_{3,k+1} = x_{3,k} + v_{3,k} \cdot \Delta t + \frac{1}{2} a_{3,k} \cdot \Delta t^2$$

$$v_{1,k+1} = v_{1,k} + a_{1,k} \cdot \Delta t$$

$$v_{2,k+1} = v_{2,k} + a_{2,k} \cdot \Delta t$$

$$v_{3,k+1} = v_{3,k} + a_{3,k} \cdot \Delta t$$

$$a_{1,k+1} = a_{1,k}$$

$$a_{2,k+1} = a_{2,k}$$

$$a_{3,k+1} = a_{3,k}$$

Hence,

$$A_k = \begin{pmatrix}
1.0 & 0.0 & 0.0 & \Delta t & 0.0 & 0.0 & \frac{1}{2}\Delta t^2 & 0.0 & 0.0 \\
0.0 & 1.0 & 0.0 & 0.0 & \Delta t & 0.0 & 0.0 & \frac{1}{2}\Delta t^2 & 0.0 \\
0.0 & 0.0 & 1.0 & 0.0 & 0.0 & \Delta t & 0.0 & 0.0 & \frac{1}{2}\Delta t^2 \\
0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & \Delta t & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & \Delta t & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 & \Delta t \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0
\end{pmatrix}$$

Δt specifies the time between consecutive steps.

3.3.2 Matrix B_k and vector u_k

In this constant veocity model we have $B_k = 0$ and $u_k = 0$.

3.3.3 Observation Matrix, H_k

We always observe the location, x_1, x_2 and x_3, which results in the following simple model for H_k:

$$H_k = \begin{pmatrix}
1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\
0.0 & 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0
\end{pmatrix}$$

3.3.4   Measurement Noise Covariance, R_k

We have the following model for measurement noise covariance matrix:

$$R_k = \begin{pmatrix} r & 0.0 & 0.0 \\ 0.0 & r & 0.0 \\ 0.0 & 0.0 & r \end{pmatrix}$$

The constant r denotes the belief about the variance of the measurement noise.

3.3.5   Process Noise Covariance, Q_k

Process noise refers to the modeling of forces that could influence our model by creating noise in the form of acceleration disturbance. According to Wikipedia, for this constant velocity model we have:

$$Q_k = G \cdot G^T \cdot \sigma_v^2$$

where,

$$G = \begin{bmatrix} 0.5\Delta t^2 & 0.5\Delta t^2 & 0.5\Delta t^2 & \Delta t & \Delta t & \Delta t & 1.0 & 1.0 & 1.0 \end{bmatrix}$$

and we can assume the acceleration process noise $\sigma_v^2 = 8.8m/s^2$, according to Schubert, R., Adam, C., Obst, M., Mattern, N., Leonhardt, V., Wanielik, G. (2011). Empirical evaluation of vehicular models for ego motion estimation. 2011 IEEE Intelligent Vehicles Symposium (IV), 534–539.

4   Source

Material for this case study was inspired by the following:
https://balzer82.github.io/Kalman/

For more details on Kalman Filtering equations used, please check:
https://en.wikipedia.org/wiki/Kalman_filter