Sensor Fusion with Kalman Filter

Jun Zhu

September 19, 2019

1 Basic Kalman Filter

The Kalman filter estimates a process by using a feedback control. The filter predicts the system state at some time and then obtains feedback based on measurement. The time update (prediction) function of the basic Kalman filter is governed by the linear stochastic equation

$$\hat{\mathbf{x}}_k = \mathbf{F}\hat{\mathbf{x}}_{k-1} + \mathbf{w}, \tag{1}$$

where $\hat{\mathbf{x}}_k \in \Re^n$ is the estimated state vector for $\mathbf{x_k}$ and $\mathbf{w} \sim N(0, \mathbf{Q})$ is the process noise with $\mathbf{Q}$ the covariance matrix. The measurement function is written as:

$$\mathbf{z_k} = \mathbf{H}\hat{\mathbf{x}}_k + \mathbf{v}, \tag{2}$$

where $\mathbf{z}_k \in \Re^m$ is the measurement vector and $\mathbf{v} \sim N(0, \mathbf{R})$ is the process noise with $\mathbf{R}$ the covariance matrix.

The equations used in the Kalman filter are listed in Table 1 [1]. Here $\mathbf{P}_k^-$ and $\mathbf{P}_k$ are the priori and posteriori error covariance matrix, respectively, $\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-$ is called the ***measurement innovation***, and $\mathbf{K}_k$, the so-called ***Kalman gain***, is chosen to minimize the posteriori error covariance.

Table 1: Discrete Kalman filter time and measurement update equations.

Time update (predict)	Measurement update (correct)
$\hat{\mathbf{x}}_k^- = \mathbf{F}\hat{\mathbf{x}}_{k-1}$ $\mathbf{P}_k^- = \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^\mathbf{T} + \mathbf{Q}$	$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}^\mathbf{T}(\mathbf{H}\mathbf{P}_k^-\mathbf{H}^\mathbf{T}+\mathbf{R})^{-1}$ $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-)$ $\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^-$

In the LIDAR measurement, the state vector $\mathbf{x}$ is given by

$$\mathbf{x} = \begin{bmatrix} p_x & p_y & \dot{p}_x & \dot{p}_y \end{bmatrix}^T. \tag{3}$$

Assuming a linear motion, we have

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4}$$

where Δt is the time step. The measurement vector is $\mathbf{z}_k = \begin{bmatrix} p_x & p_y \end{bmatrix}^T$, and the measurement matrix is simply

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{5}$$

2 Extended Kalman filter

The extended Kalman filter is used to solve the problem in which the state function

$$\hat{\mathbf{x}}_k = \mathbf{f}(\hat{\mathbf{x}}_{k-1},\ \mathbf{w}_{k-1}) \tag{6}$$

and/or the measurement function

$$\mathbf{z}_k = \mathbf{h}(\hat{\mathbf{x}}_k, \mathbf{v}_k). \tag{7}$$

are nonlinear.

The equations used in the extended Kalman filter are summarized in Table 2 [1]. Here, $\mathbf{F}$ is the Jacobian matrix of partial derivatives of $\mathbf{f}$ with respect to $\mathbf{x}$, which gives

$$F_{i,j} = \frac{\partial f_i}{\partial x_j}. \tag{8}$$

$\mathbf{W}$ is the Jacobian matrix of partial derivatives of $\mathbf{f}$ with respect to $\mathbf{w}$, which gives

$$W_{i,j} = \frac{\partial f_i}{\partial w_j}. \tag{9}$$

$\mathbf{H}$ is the Jacobian matrix of partial derivatives of $\mathbf{h}$ with respect to $\mathbf{x}$, that is

$$H_{i,j} = \frac{\partial h_i}{\partial x_j}. \tag{10}$$

$\mathbf{V}$ is the Jacobian matrix of partial derivatives of $\mathbf{h}$ with respect to $\mathbf{v}$, that is

$$V_{i,j} = \frac{\partial h_i}{\partial v_j}. \tag{11}$$

It must be noted that in the extended Kalman filter the noise is no longer normal after the nonlinear transformation.

Table 2: Extended Kalman filter time and measurement update equations.

Time update (predict)	Measurement update (correct)
$\hat{\mathbf{x}}_k^- = \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{0})$ $\mathbf{P}_k^- = \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^\mathbf{T} + \mathbf{W_k}\mathbf{Q}_{k-1}\mathbf{W}_k^T$	$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}^\mathbf{T}(\mathbf{H}\mathbf{P}_k^-\mathbf{H}^\mathbf{T} + \mathbf{V}_k\mathbf{R}_k\mathbf{V}_k^T)^{-1}$ $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-, \mathbf{0}))$ $\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^-$

The RADAR measurement returns

$$\mathbf{z_k} = \begin{bmatrix} \rho_k & \varphi_k & \dot{\rho}_k \end{bmatrix}^T \tag{12}$$

in the polar coordination system, which means the mapping from the state vector $\mathbf{x_k}$ to the measurement vector $\mathbf{z_k}$ is no longer linear. The transform equations from the Cartesian coordinate system to the polar coordinate system ($\mathbf{h}$) are given by

$$\rho = \sqrt{p_x^2 + p_y^2},\ \varphi = \arctan(\frac{p_y}{p_x}),\ \dot{\rho} = \frac{p_x\dot{p}_x + p_y\dot{p}_y}{\sqrt{p_x^2 + p_y^2}}. \tag{13}$$

The Jacobian matrix of $\mathbf{h}$ is

$$\mathbf{H} = \begin{bmatrix} \frac{p_x}{\sqrt{p_x^2+p_y^2}} & \frac{p_y}{\sqrt{p_x^2+p_y^2}} & 0 & 0 \\ -\frac{p_y}{p_x^2+p_y^2} & \frac{p_x}{p_x^2+p_y^2} & 0 & 0 \\ \frac{p_y(p_y\dot{p}_x - p_x\dot{p}_y)}{(p_x^2+p_y^2)^{3/2}} & \frac{p_x(p_x\dot{p}_y - p_y\dot{p}_x)}{(p_x^2+p_y^2)^{3/2}} & \frac{p_x}{\sqrt{p_x^2+p_y^2}} & \frac{p_y}{\sqrt{p_x^2+p_y^2}} \end{bmatrix}. \tag{14}$$

Since the linear motion assumption still holds, $\mathbf{F}$ is given by equation (4). Furthermore, we assume that the process and measurement noises are both static. Therefore, $\mathbf{W}$ and $\mathbf{V}$ are both identity matrices.

3 Uncented Kalman filter

3.1 Uncented transform

The uncented transform is a method for calculating the statistics of a random variable which undergoes a nonlinear transformation. Considering propagating a 1D state vector $\mathbf{x}$ through a nonlinear function $\mathbf{y} = \mathbf{f}(\mathbf{x})$. To calculate the statistics of $\mathbf{f}$, we form a matrix $\mathbf{\Sigma}$ consisting of $2L+1$ sigma vectors [2]

$$\mathbf{\Sigma_0} = \bar{\mathbf{x}} \tag{15}$$

$$\mathbf{\Sigma_i} = \bar{\mathbf{x}} + \sqrt{(\lambda + L)\mathbf{P_x}}, i = 1, ..., L \tag{16}$$

$$\mathbf{\Sigma_i} = \bar{\mathbf{x}} - \sqrt{(\lambda + L)\mathbf{P_x}}, i = L+1, ..., 2L \tag{17}$$

where $\lambda = \alpha^2(L+\kappa) - L$ is a scaling parameter. The constant $\alpha \in (0, 1]$ determines the spread of the sigma points around $\bar{\mathbf{x}}$. Another constant κ is usually set to either 0 or $3 - L$.

These sigma vectors are propagated through the same nonlinear function

$$\mathbf{\Gamma_i} = \mathbf{f}(\mathbf{\Sigma_i}), i = 0, ..., 2L \tag{18}$$

The mean and covariance of $\mathbf{\Gamma}$ are given by

$$\bar{\mathbf{\Gamma}} = \sum_{i=0}^{2L} W_i^{(m)} \mathbf{\Gamma_i}, \tag{19}$$

and

$$\mathbf{P_\Gamma} = \sum_{i=0}^{2L} W_i^{(c)} (\mathbf{\Gamma_i} - \bar{\mathbf{\Gamma}})(\mathbf{\Gamma_i} - \bar{\mathbf{\Gamma}})^T \tag{20}$$

where the weights W_i is given by

$$\begin{cases} W_i^{(m)} & = \lambda/(L+\lambda) \\ W_i^{(c)} & = \lambda/(L+\lambda) + 1 - \alpha^2 + \beta \\ W_i^{(m)} & = W_i^{(c)} = 0.5/(L+\lambda), i = 1, ..., 2L \end{cases}, \tag{21}$$

where β is related to the distribution of $\mathbf{x}$ and $\beta = 2$ for Gaussian distribution.

3.2 Uncented Kalman filter equations

The uncented Kalman filter (UKF) is a derivative-free alternative to the extended Kalman filter (EKF). In UKF, the augmented state vector is initialized as

$$\mathbf{x}_0^a = [\mathbf{x}_0^T, \mathbf{0}, \mathbf{0}]^T, \tag{22}$$

and the augmented covariance matrix is initialized as

$$\mathbf{P}_0 = \begin{bmatrix} \mathbf{P}_0^a & 0 & 0 \\ 0 & \mathbf{Q} & 0 \\ 0 & 0 & \mathbf{R} \end{bmatrix} \tag{23}$$

The sigma matrix and augmented sigma matrix at step k are denoted as $\mathbf{\Sigma}_k^x$ and $\mathbf{\Sigma}_k^a$, respectively.
- Time update (predict)

$$\mathbf{\Sigma}_k^{x-} = \mathbf{f}(\mathbf{\Sigma}_{k-1}^x, \mathbf{\Sigma}_{k-1}^{\mathrm{w}}) \tag{24}$$

$$\mathbf{x}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \mathbf{\Sigma}_{k,i}^{x-} \tag{25}$$

$$\mathbf{P}_k^- = \sum_{i=0}^{2L} W_i^{(c)} (\mathbf{\Sigma}_{k,i}^{x-} - \mathbf{x}_k^-)(\mathbf{\Sigma}_{k,i}^{x-} - \mathbf{x}_k^-)^T \tag{26}$$

$$\mathbf{\Gamma}_k^- = \mathbf{h}(\mathbf{\Sigma}_k^{x-}, \mathbf{\Sigma}_{k-1}^{\mathrm{v}}) \tag{27}$$

$$\mathbf{z}_k^- = \sum_{i=0}^{2L} W_i^{(m)} \mathbf{\Gamma}_{k,i}^- \tag{28}$$

- Measurement update (correct)

$$\mathbf{P}_{\mathbf{z}_k \mathbf{z}_k} = \sum_{i=0}^{2L} W_i^{(c)} (\mathbf{\Gamma}_{k,i}^- - \mathbf{z}_k^-)(\mathbf{\Gamma}_{k,i}^- - \mathbf{z}_k^-)^T \tag{29}$$

$$\mathbf{P}_{\mathbf{x}_k \mathbf{z}_k} = \sum_{i=0}^{2L} W_i^{(c)} (\mathbf{\Gamma}_{k,i}^- - \mathbf{z}_k^-)(\mathbf{\Sigma}_{k,i}^{x-} - \mathbf{x}_k^-)^T \tag{30}$$

$$\mathbf{K_k} = \mathbf{P}_{\mathbf{x}_k \mathbf{z}_k} \mathbf{P}_{\mathbf{z}_k \mathbf{z}_k}^{-1} \tag{31}$$

$$\mathbf{x}_k = \mathbf{x}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{z}_k^-) \tag{32}$$

$$\mathbf{P}_k = \mathbf{P}_{k-1} - K_k \mathbf{P}_{\mathbf{z}_k \mathbf{z}_k} K_k^T \tag{33}$$

In the case where the process noise are purely addictive, the process noise $\mathbf{Q}$ should be removed from the augmentation equation 23, and equation 24 and 26 become

$$\mathbf{\Sigma}_k^{x-} = \mathbf{f}(\mathbf{\Sigma}_{k-1}^x) \tag{34}$$

and

$$\mathbf{P}_k^- = \sum_{i=0}^{2L} W_i^{(m)} (\mathbf{\Sigma}_{k,i}^{x-} - \mathbf{x}_k^-)(\mathbf{\Sigma}_{k,i}^{x-} - \mathbf{x}_k^-)^T + \mathbf{Q}, \tag{35}$$

respectively.

In the case where the measurement noise are purely addictive, the measurement noise $\mathbf{Q}$ should be removed from the augmentation equation 23, and equation 27 and 29 become

$$\mathbf{\Gamma}_k^- = \mathbf{h}(\mathbf{\Sigma}_k^{x-}) \tag{36}$$

and

$$\mathbf{P}_{\mathbf{z}_k \mathbf{z}_k} = \sum_{i=0}^{2L} W_i^{(c)} (\mathbf{\Gamma}_{k,i}^- - \mathbf{z}_k^-)(\mathbf{\Gamma}_{k,i}^- - \mathbf{z}_k^-)^T + \mathbf{R}, \tag{37}$$

respectively.

3.3 UKF in CTRV model

3.3.1 The CTRV (Constant Turn Rate and Velocity) model

The state vector and the velocity vector are given by

$$\mathbf{x} = \begin{bmatrix} p_x & p_y & v & \psi & \dot{\psi} \end{bmatrix}^T \tag{38}$$

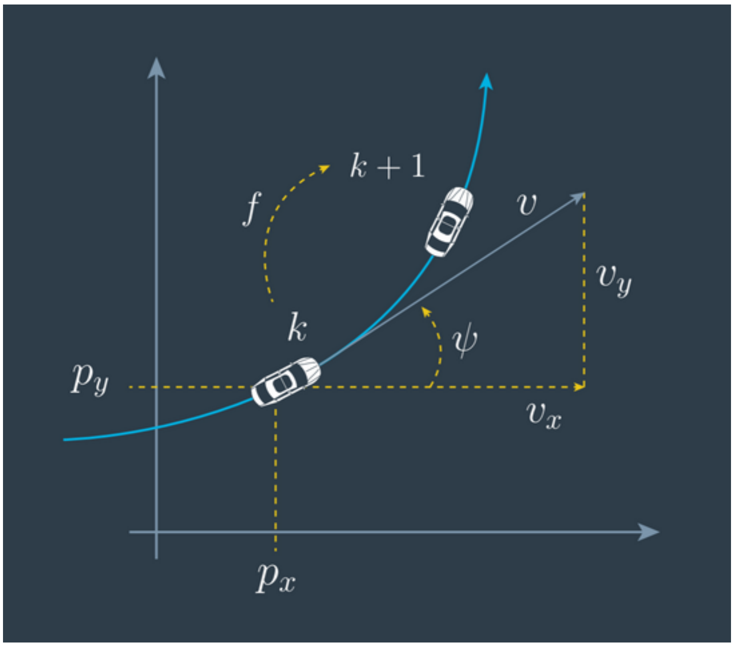

Figure 1: Illustration of the CTRV model

and

$$\mathbf{v} = \begin{bmatrix} \dot{p}_x & \dot{p}_y & \dot{v} & \dot{\psi} & \ddot{\psi} \end{bmatrix}^T, \tag{39}$$

respectively. Since the CTRV model is use, we have $\dot{v} = 0$ and $\ddot{\psi} = 0$.

The time update function can be written as

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \int_{t_{k-1}}^{t_k} \mathbf{v_{k-1}} dt = \mathbf{x}_{k-1} + \begin{bmatrix} v_{k-1} \int_{t_{k-1}}^{t_k} cos(\psi_{k-1}(t))dt \\ v_{k-1} \int_{t_{k-1}}^{t_k} sin(\psi_{k-1}(t))dt \\ 0 \\ \dot{\psi}\Delta t \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{1}{2} w_a \Delta t^2 cos(\psi_{k-1}(t)) \\ \frac{1}{2} w_a \Delta t^2 sin(\psi_{k-1}(t)) \\ \mathrm{w}_a \Delta t \\ \mathrm{w}_{\ddot{\phi}} \Delta t \\ 0 \end{bmatrix} \tag{40}$$

where

$$v_{k-1} \int_{t_{k-1}}^{t_k} cos(\psi_{k-1}(t))dt = \begin{cases} v_{k-1}(sin(\psi_k) - sin(\psi_{k-1}))/\dot{\psi}_{k-1}, & \psi_{k-1} \neq 0 \\ v_{k-1} cos(\psi_{k-1})\Delta t, & \psi_{k-1} = 0 \end{cases} \tag{41}$$

and

$$v_{k-1} \int_{t_{k-1}}^{t_k} sin(\psi_{k-1}(t))dt = \begin{cases} v_{k-1}(cos(\psi_{k-1}) - cos(\psi_k))/\dot{\psi}_{k-1}, & \psi_{k-1} \neq 0 \\ v_{k-1} sin(\psi_{k-1})\Delta t, & \psi_{k-1} = 0 \end{cases}. \tag{42}$$

During the measurement update, equations 36 and 37 can be used.

4 Real-time consistency tests

The estimation errors of a state estimator (filter) based on a finite number of samples (measurements) should be consistent with their theoretical statistical properties: [3]

1. Have mean zero (i.e., the estimates are unbiased).
2. Have covariance matrix as calculated by the filter.

As a result, the consistency criteria of a filter are as follows:

(a) The state errors should be acceptable as zero mean and have magnitude commensurate with the state covariance as yielded by the filter.

(b) **The innovations should also have the same property**.

(c) **The innovations should be acceptable as white**.

The last two criteria are the only ones that can be tested in applications with real data. The first criterion, which is really the most important one, can be tested only in simulations with the ***normalized estimation error squared (NEES)***

$$\epsilon_{x,k} = (\mathbf{x}_k - \hat{\mathbf{x}}_k)^T \mathbf{P}_k^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_k). \tag{43}$$

where $\mathbf{P}_k$ is the posteriori error covariance matrix. Under the hypothesis that the filter is consistent, $\epsilon_{x,k}$ should have a χ^2 distribution with n (n_x) degrees of freedom.

Criterion (b) can be tested with the ***time-average normalized innovation squared (NIS)*** statistic. For the basic Kalman filter, it is given by

$$\epsilon_{z,k} = (\mathbf{z}_k - \mathbf{H}\mathbf{x}_k^-)^{\mathbf{T}} S_k^{-1} (\mathbf{z}_k - \mathbf{H}\mathbf{x}_k^-). \tag{44}$$

where

$$S_k = \mathbf{H}\mathbf{P}_k^- \mathbf{H}^{\mathbf{T}} + \mathbf{R} \tag{45}$$

Note that S_k also appears on the right-hand side of the equation of calculating the Kalman gain. Under the hypothesis that the filter is consistent, $\epsilon_{z,k}$ should have a χ^2 distribution with m (n_z) degrees of freedom.

5 More readings

http://home.wlu.edu/~levys/kalman_tutorial/
http://biorobotics.ri.cmu.edu/papers/sbp_papers/integrated3/kleeman_kalman_basics.pdf

References

[1] Greg Welch and Gary Bishop. *An introduction to the Kalman filter*. ACM, Inc. 2001.

[2] Eric A. Wan and Rudolph van der Merwe, *Kalman filtering and neural networks*. p.221, John Wiley & Sons, Inc., 2002.

[3] Yaakov Bar-Shalom, X. Rong Li, Thiagalingam Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. John Wiley & Sons, Inc., 2001.