Making Sense of Extended Kalman Filters

Tianshu Huang

Fall 2018

1 Background

1.1 Background: Kalman Filters

An agent asynchronously receive observations $\mathbf{z}_k$ at various times. The agent maintains a current state prediction $\hat{\mathbf{x}}_{k|k}$. After receiving an observation $\mathbf{z}_k$, which can be a processed camera input, one-dimensional sensor input such as a distance sensor, or an observation transmitted by another agent, the agent should update its current state estimate:

$$\left.\begin{array}{cc} \text{Previous state prediction} & \hat{\mathbf{x}}_{k-1|k-1} \\ \text{Previous state covariance} & P_{k-1|k-1} \\ \text{Input data} & \hat{\mathbf{z}}_k \end{array}\right\} \longrightarrow \left\{\begin{array}{cc} \text{Updated state prediction} & \hat{\mathbf{x}}_{k|k} \\ \text{Update covariance} & P_{k|k} \end{array}\right.$$

1.2 Background: SLAM

SLAM stands for Simultaneous Localization and Mapping. SLAM has been described as a "chicken and egg" problem: if an agent has a map of its location, it can find out where it is, and if the agent knows where it is, it can draw a map of the things it sees. However, if an agent knows neither, how can it draw a map as it goes along to figure out where it is?

The agent maintains a map M. The agent continuously observes its surroundings, and identifies distinct landmarks $\mathbf{m}_k$. Once a landmark is identified, it must be added to the map if not already in the map; otherwise, the landmark can be used to locate the agent.

2 Kalman Filters

2.1 Notation

Definition 1 *The current state at step* k *is a vector* $\mathbf{x}_k \in \mathbb{R}^n$.

Definition 2 *The state transition matrix is denoted by* F_k. *Without any control inputs or noise, this describes* $\mathbf{x}_k$ *in terms of* $\mathbf{x}_{k-1}$:

$$\mathbf{x}_k = F_k \mathbf{x}_{k-1}$$

Definition 3 *The observation process matrix is denoted by* H_k. *This is a matrix mapping the current state to an observed vector. Therefore, without any noise, any observations* $\mathbf{z}_k$ *can be described as*

$$\mathbf{z}_k = H_k \mathbf{x}_k$$

Definition 4 *The control input model is denoted by a matrix* B_k *for some control vector* $\mathbf{u}_k$ *that maps* $\mathbf{u}_k$ *to* $\Delta \mathbf{x}_k$.

Definition 5 *The additive process noise covariance is denoted by* Q_k. *Additive noise is modeled as a multivariate Gaussian. Denote noise sampled from this noise as a random variable* $\mathbf{w}_k \sim N(0, Q_k)$. *Therefore, state evolution with control input is modelled as*

$$\mathbf{x}_k = F_k \mathbf{x}_{k-1} + B_k \mathbf{u}_k + \mathbf{w}_k$$

Definition 6 *The observation covariance matrix is denoted by* R_k. *Like the state evolution noise, this is modelled by a multivariate gaussian; denote a random variable drawn from this distribution by* $\mathbf{v}_k \sim N(0, R_k)$.

Therefore, observations are modelled by

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

Definition 7 *The state prediction for time* a *given observations up to time* b *is denoted by* $\hat{\mathbf{x}}_{a|b}$.

Definition 8 *The covariance matrix of the state prediction is*

$$P_{a|b} = cov(\mathbf{x}_a - \hat{\mathbf{x}}_{a|b})$$

2.2 A Priori Update

Lemma 1 *The a priori updated covariance for a discrete sample space is given by*

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T$$

Proof For a centered $m \times n$ data matrix D, the covariance is given by $\frac{1}{n} D^T D$. Also, since $D_k^T = F_k D_{k-1}^T$ the post-transition data matrix is given by $D_k = D_{k-1} F_k^T$. Therefore,

$$\begin{aligned}
P_{k|k-1} &= \frac{1}{n} D_k^T D_k \\
&= \frac{1}{n} (D_{k-1} F_k^T)^T (D_{k-1} F_k^T) \\
&= F_k \frac{1}{n} D_{k-1}^T D_{k-1} F_k^T \\
&= F_k P_{k-1} F_k^T
\end{aligned}$$

Theorem 1 *The a priori update covariance for an arbitrary space is*

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T$$

Proof By the definition of covariance, we have $P_{k|k-1} = cov(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})$. Therefore,

$$\begin{aligned}
P_{k|k-1} &= cov(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1}) \\
&= cov\left(F_k \mathbf{x}_{k-1} + B_k u_k + \mathbf{w}_k - (F_k \hat{\mathbf{x}}_{k-1|k-1} + B_k u_k)\right) \\
&= cov(F_k(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1|k-1}) + \mathbf{w}_k) \\
&= E\left[(F_k(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1|k-1}) + \mathbf{w}_k)(F_k(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1|k-1}) + \mathbf{w}_k)^T\right] \\
&= E[F_k(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1|k-1})(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1|k-1})^T F_k^T] \\
&\quad + 2E[\mathbf{w}_k(\mathbf{x}_k - \hat{\mathbf{x}}_{k-1|k-1})^T F_k^T + E[\mathbf{w}_k \mathbf{w}_k^T]
\end{aligned}$$

Since the random variable $\mathbf{x}$ is assumed to be symmetric, $E[\mathbf{x} - \hat{\mathbf{x}}_{k-1|k-1}] = 0$. Also, since $\mathbf{w}_k$ is pulled from a random variable with covariance Q_k, $E[\mathbf{w}_k \mathbf{w}_k^T] = Q_k$. Using the linearity of expectation,

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$$

as desired.

2.3 Innovation

Definition 9 *The innovation, denoted $\tilde{\mathbf{y}}_k$ is the difference between the measurement and the predicted measurement:*

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - H_k \hat{\mathbf{x}}_{k|k-1}$$

Definition 10 *The covariance of the innovation is denoted by S_k.*

Theorem 2 *The covariance matrix of the innovation is given by*

$$S_k = R_k + H_k P_{k|k-1} H_k^T$$

Proof See Lemma 1. Instead of $cov(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k-1})$, we evaluate $cov(\tilde{\mathbf{y}}_k) = cov(\mathbf{z}_k - H_k \hat{\mathbf{x}}_{k|k-1})$.

2.4 A Posteriori Update and Kalman Gain

Definition 11 *The a posteriori state estimate is written as $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k \tilde{\mathbf{y}}_k$ for some matrix K_k, called the optimal Kalman gain.*

The optimal Kalman gain should be the solution of

$$\arg\min_{K_k} \; E\left[||\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}||_2^2\right]$$

Lemma 2 *The posteriori covariance matrix estimate is given by*

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} (I - K_k H_k)^T + K_k R_k K_k^T$$

Proof Starting with the definition of $P_{k|k}$, and substituting $\hat{\mathbf{x}}_{k|k}$, $\tilde{\mathbf{y}}_k$, and finally $\mathbf{z}_k$,

$$\begin{aligned}
P_{k|k} &= cov(\mathbf{x} - \hat{\mathbf{x}}_{k|k}) \\
&= cov(\mathbf{x} - (\hat{\mathbf{x}}_{k|k-1} + K_k\tilde{\mathbf{y}}_k)) \\
&= cov(\mathbf{x} - (\hat{\mathbf{x}}_{k|k-1} + K_k(\mathbf{z}_k - H_k\hat{\mathbf{x}}_{k|k-1}))) \\
&= cov(\mathbf{x} - (\hat{\mathbf{x}}_{k|k-1} + K_k(H_k\mathbf{x}_k + \mathbf{v}_k - H_k\hat{\mathbf{x}}_{k|k-1}))) \\
&= cov(\mathbf{x} - \hat{\mathbf{x}}_{k|k-1} - K_kH_k\mathbf{x}_k - K_kv_k + K_kH_k\hat{\mathbf{x}}_{k|k-1}) \\
&= cov((I - K_kH_k)(\mathbf{x} - \hat{\mathbf{x}}_{k|k-1}) - K_k\mathbf{v}_k) \\
&= E[(I - K_kH_k)(\mathbf{x} - \hat{\mathbf{x}}_{k|k-1})(\mathbf{x} - \hat{\mathbf{x}}_{k|k-1})^T(I - K_kH_k)^T] \\
&\quad - 2E[(I - K_kH_k)(\mathbf{x} - \hat{\mathbf{x}}_{k|k-1})\mathbf{v}_k^TK_k^T] + E[K_k\mathbf{v}_k\mathbf{v}_k^TK_k)]
\end{aligned}$$

Since $\mathbf{x} - \hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{v}_k$ are independent and centered at 0, $E[(\mathbf{x} - \hat{\mathbf{x}}_{k|k-1})^T\mathbf{v}_k] = 0$. Therefore,

$$\begin{aligned}
P_{k|k} &= (I - K_kH_k)E[(\mathbf{x} - \hat{\mathbf{x}}_{k|k-1})(\mathbf{x} - \hat{\mathbf{x}}_{k|k-1})^T](I - K_kH_k)^T + K_kE[\mathbf{v}_k\mathbf{v}_k^T]K_k^T \\
&= (I - K_kH_k)P_{k|k-1}(I - K_kH_k) + K_kR_kK_k^T
\end{aligned}$$

This can be further expanded as

$$\begin{aligned}
P_{k|k} &= P_{k|k-1} - K_kH_kP_{k|k-1} - P_{k|k-1}H_k^TK_k^T + K_kH_kP_{k|k-1}H_k^TK_k^T + K_kR_kK_k^T \\
&= P_{k|k-1} - K_kH_kP_{k|k-1} - P_{k|k-1}K_kH_k + K_k(H_kP_{k|k-1}H_k^T + R_k)K_k^T + K_kR_kK_k^T \\
&= P_{k|k-1} - K_kH_kP_{k|k-1} - P_{k|k-1}K_kH_k + K_kS_kK_k^T
\end{aligned}$$

Theorem 3 *The Kalman gain is given by* $K_k = P_{k|k-1}H_k^TS_k^{-1}$.

Proof As according to the definition above, we seek to minimize

$$\begin{aligned}
E\left[||\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}||_2^2\right] &= E\left[Tr\left((\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})^T\right)\right] \\
&= Tr\left(E[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})^T]\right) \\
&= Tr(P_{k|k})
\end{aligned}$$

In order to minimize $tr(P_{k|k})$ over K_k, set $\frac{\partial}{\partial K_k}tr(P_{k|k}) = 0$. Note that covariance matrices are symmetric, so $S_k = S_k^T$.

$$\begin{aligned}
\frac{\partial}{\partial K_k}tr(P_{k|k}) &= \frac{\partial}{\partial K_k}tr\left((I - K_kH_k)P_{k|k-1}(I - K_kH_k) + K_kR_kK_k^T\right) \\
&= \frac{\partial}{\partial K_k}tr\left(P_{k|k-1} - K_kH_kP_{k|k-1} - P_{k|k-1}K_kH_k + K_kS_kK_k^T\right) \\
&= -\frac{\partial}{\partial K_k}tr\left(K_kH_kP_{k|k-1}\right) - \frac{\partial}{\partial K_k}tr\left(P_{k|k-1}K_kH_k\right) + \frac{\partial}{\partial K_k}tr\left(K_kS_kK_k^T\right) \\
&= -2\frac{\partial}{\partial K_k}tr\left(K_k^TP_{k|k-1}^TH_k^T\right) + \frac{\partial}{\partial K_k}tr\left(K_k^TS_kK_k\right) \\
&= -2P_{k|k-1}^TH_k^T + (S_k + S_k^T)K_k \\
&= -2P_{k|k-1}^TH_k^T + 2S_kK_k
\end{aligned}$$

Therefore, using $P_{k|k-1} = P_{k|k-1}^T$ (by the symmetry of covariance matrices),

$$\begin{aligned} -2P_{k|k-1}^T H_k^T + 2S_k K_k = 0 \\ \Rightarrow K_k S_k = P_{k|k-1}^T H_k^T \\ \Rightarrow K_k = P_{k|k-1}^T H_k^T S_k^{-1} = P_{k|k-1} H_k^T S_k^{-1} \end{aligned}$$

Theorem 4 *When K_k is the optimal kalman gain, $P_{k|k} = (I - K_k H_k) P_{k|k-1}$.*

Proof Using $K_k S_k = P_{k|k-1} H_k^T \Rightarrow K_k S_k K_k^T = P_{k|k-1} H_k^T K_k^T$, we can substitute this back into the final result of Theorem 3:

$$\begin{aligned} P_{k|k} &= P_{k|k-1} - K_k H_k P_{k|k-1} - P_{k|k-1} K_k H_k + K_k S_k K_k^T \\ &= P_{k|k-1} - K_k H_k P_{k|k-1} \\ &= (I - K_k H_k) P_{k|k-1} \end{aligned}$$

2.5 Summary

Algorithm 1 Kalman Filter Algorithm

procedure KalmanFilter(
Previous state estimate $\hat{\mathbf{x}}_{k-1|k-1}$, state transition model F_k,
observation model H_k, process covariance Q_k,
observation noise covariance R_k, control input model B_k,
control vector $\mathbf{u}_K$, measurement $\mathbf{z}_k$)
Prediction Phase:
Update the *a priori* state estimate: $\hat{\mathbf{x}}_{k|k-1} \leftarrow F_k \hat{\mathbf{x}}_{k-1|k-1} + B_k \mathbf{u}_k$
Update the *a priori* error covariance: $P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$
Update Phase:
Compute the innovation: $\tilde{\mathbf{y}} \leftarrow \mathbf{z}_k - H_k \hat{\mathbf{x}}_{k|k-1}$
Compute the innovation covariance: $S_k \leftarrow R_k + H_k P_{k|k-1} H_k^T$
Compute the Optimal Kalman gain: $K_k \leftarrow P_{k|k-1} H_k^T S_k^{-1}$
Update the *a posteriori* state estimate: $\hat{\mathbf{x}}_{k|k} \leftarrow \hat{\mathbf{x}}_{k|k-1} + K_k \tilde{\mathbf{y}}_K$
Update the *a posteriori* estimate covariance: $P_{k|k} \leftarrow (I - K_k H_k) P_{k|k-1}$
Obtain the measurement post-fit residual: $\tilde{\mathbf{y}}_{k|k} \leftarrow \mathbf{z}_k - H_k \hat{\mathbf{x}}_{k|k}$
return $(\hat{\mathbf{x}}_{k|k}, P_{k|k}, \tilde{\mathbf{y}}_{k|k})$
end procedure

3 Extended Kalman Filter: SLAM Algorithm

3.1 Overview

To extend Kalman Filters to SLAM (EKF), after the *a priori* phase, the innovation should be used to extend the map, with landmarks below a certain confidence threshold rejected for ease of computation. If any changes are made, the innovation should be re-computed. Then, to deal with indistinguishable landmarks, an equivalence class is described, and the landmark set is chosen using the MLE (Maximum Likelihood Estimator).

3.2 Definitions

Definition 12 *A map is a collection of landmarks with their associated covariance matrix and confidence value. However, when landmarks are indistinguishable, modifications are necessary in order to create an equivalence class of landmarks. Define a map as a collection* $M = \{L_i\}_i$ *of landmark classes. Then, define a landmark class as*

$$L_i = \{\hat{\mathbf{p}}_i, \Sigma_i, c_i\}_i$$

where $\hat{\mathbf{p}}_i$ *is the state estimate of the landmark* $\mathbf{p}_i$*,* Σ_i *is the covariance matrix of the landmark* $\mathbf{p}_i$*, and* c_i *is the obstacle confidence.*

Definition 13 *The Mahalanobis distance between a point* $\mathbf{p}$ *and a distribution with center* μ *and covariance matrix* Σ_μ *is*

$$D(\mathbf{p}, [\mu, \Sigma]) = \sqrt{(\mathbf{p} - \mu)^T \Sigma_{\mathbf{p}}^{-1} (\mathbf{p} - \mu)}$$

Definition 14 *For observation* $\hat{\mathbf{z}}$ *with covariance* $\Sigma_{\mathbf{z}}$ *and landmark class* L*, we define the norm* $d(\hat{\mathbf{z}}, L)$ *from an observed landmark to a landmark class as the probability that the observed landmark is not a member of the landmark class.*

Since this model assumes that error follows a multivariate normal, the probability that an observation is not a landmark is modelled as[1]

$$\mathbb{P}[\mathbf{z} \neq \mathbf{p}_i] = (1 - c_i) \int_{D_{\mathbf{p}}(\mathbf{t})=0}^{D\mathbf{p}(\mathbf{t})=D(\hat{\mathbf{z}})} N(0, I) d\mathbf{t}$$

where $D_{\mathbf{p}}(\mathbf{t})$ is used as shorthand for $D(\mathbf{t}, [\hat{\mathbf{p}}_i, \Sigma_i + \Sigma_{\mathbf{p}}])$ since the Mahalanobis distance in this application should always be taken in the context of $\mathbf{p}_i$ and $\Sigma_i + \Sigma_{\mathbf{p}}$.

Assuming independence of landmarks,

$$\begin{aligned} d(\hat{\mathbf{z}}, L) &= \mathbb{P}[\mathbf{z} \notin L] \\ &= \prod_{\mathbf{p}_i \in L} \mathbb{P}[\mathbf{z} \neq \mathbf{p}_i] \\ &= \prod_{\{\mathbf{p}_i, \Sigma_i, c_i\} \in L} \left((1 - c_i) \int_{D_{\mathbf{p}}(\mathbf{t})=0}^{D\mathbf{p}(\mathbf{t})=D(\hat{\mathbf{z}})} N(0, I) d\mathbf{t} \right) \end{aligned}$$

[1] This integral can be interpreted as integrating over $\mathbb{R}^n$ over $B_{D(\hat{\mathbf{z}})}(0)$ with the Mahalanobis distance to $\mathbf{p}$ as the norm. This is also the integral of t along the level-set curves of $N(\hat{\mathbf{p}}_i, \Sigma_i + \Sigma_{\mathbf{z}})$ up to $\mathbf{y} - \hat{\mathbf{p}}_i$.

Since the minimum obstacle distance is 6", the integral should be close to 1 for at most all but one landmark. Therefore, we can approximate this as

$$\mathbb{P}[\mathbf{y} \notin L] = \min_i (1 - c_i) \int_{D_{\mathbf{p}}(\mathbf{t})=0}^{D_{\mathbf{p}}(\mathbf{t})=D_{\mathbf{p}}(\hat{\mathbf{z}})} N(0, I) d\mathbf{t}$$

Denote the surface area of a d dimensional ball as v_d. Then, since $N(0, I)$ is symmetric, the integral can be further simplified as

$$\mathbb{P}[\mathbf{y} \notin L] = \min_i (1 - c_i) \int_{t=0}^{D_{\mathbf{p}}(\hat{\mathbf{z}})} v_d t^d \frac{1}{\sqrt{(2\pi)^{d-1}}} e^{-t^2/2} \; dt$$

3.3 New Landmark Creation

After the *a priori* updates, the innovation should be computed, and used to generate new landmarks.

Suppose some $Z = \{\mathbf{z}_i\}_i$ landmarks are observed within a certain landmark equivalence class L. Then, for each observation $\mathbf{z}_i$, compute $\mathbb{P}[\mathbf{y} \notin L]$. This is the confidence that the observed landmark is not a current member of L; the element

$$\{\hat{\mathbf{p}}_i = \mathbf{z}_i, \Sigma_i = \Sigma_{\mathbf{z}}, c_i = \mathbb{P}[\mathbf{y} \notin L]\}$$

should therefore be added to L. These elements can then be appended to the a priori updated map $\hat{\mathbf{x}}_{k|k-1}$, and entries added to the covariance matrix $P_{k|k-1}$.

3.4 Landmark Assignment

In order to match observed landmarks $\{\mathbf{p}_i, \Sigma_i, c_i\}_{i \in A}$ in $\hat{\mathbf{x}}_{k|k-1}$ with observed landmarks $Z = \{\hat{\mathbf{z}}_i\}_{i \in I}$ using a maximum likelihood configuration, we seek to solve the optimization problem

$$\begin{aligned} \underset{\{i_n\}_{n \in B}}{\text{maximize}} \quad & \sum_{n \in I} \mathbb{P}[\mathbf{z}_n = \mathbf{p}_{i_n}] \\ \text{subject to} \quad & \forall n \in B : i_n \in A \\ & \forall m, n \in B, m \neq n : i_n \neq i_m \end{aligned}$$

Examining the expansion

$$\mathbb{P}[\mathbf{z}_n = \mathbf{p}_{i_n}] = c_{i_n} \int_{t=D_{\mathbf{p}}(\mathbf{z})}^{\infty} v_d t^d \frac{1}{\sqrt{(2\pi)^{d-1}}} e^{-t^2/2} \; dt$$

we can see that this problem favors matching observed landmarks to landmarks with a high confidence, even if this results in a higher distance to landmark, as desired.

In order to implement a greedy solution to this problem, assign i_n in order of $P[\mathbf{z}_n = \mathbf{p}_{i_n}]$:

Algorithm 2 Greedy Landmark Matching

procedure MATCH(Map $\{\mathbf{p}_i, \Sigma_i, c_i\}_{i \in A}$, observed landmarks $Z = \{\hat{\mathbf{z}}_i\}_{i \in I}$)
    while Unassigned observations remain **do**
        Find the closest $\mathbf{z}_i$ to remaining each $\mathbf{p}_j$
        Compute the corresponding $\mathbb{P}[\mathbf{z}_i = \mathbf{p}_j]$
        Assign the pair with highest probability and remove the pair
    end while
end procedure

3.5 Algorithm

Adding the previous elements to the base Kalman Filter algorithm, the full algorithm is as follows:

Algorithm 3 Extended Kalman Filter with indistinguishable landmark classes

procedure EKF(
        Previous state estimate $\hat{\mathbf{x}}_{k-1|k-1}$, state transition model F_k,
        observation model H_k, process covariance Q_k,
        observation noise covariance R_k, control input model B_k,
        control vector $\mathbf{u}_K$, measurement $\mathbf{z}_k$,
        Map M (is a subset of $\hat{\mathbf{x}}_{k-1|k-1}$))
    Prediction Phase:
    Update the *a priori* state estimate: $\hat{\mathbf{x}}_{k|k-1} \leftarrow F_k \hat{\mathbf{x}}_{k-1|k-1} + B_k \mathbf{u}_k$
    Update the *a priori* error covariance: $P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$
    Map Update Phase:
    For each observed obstacle in $\mathbf{z}_k$, update the map. Landmarks with a low confidence can be rejected.
    Reflect changes made in the map in $\hat{\mathbf{x}}_{k|k-1}$. Also extend R_k, Q_k, and H_k.
    Rearrange equivalence classes within R_k, H_k, and $\mathbf{z}_k$ to the maximum-likelihood configuration
    Update Phase:
    Compute the innovation: $\tilde{\mathbf{y}} \leftarrow \mathbf{z}_k - H_k \hat{\mathbf{x}}_{k|k-1}$
    Compute the innovation covariance: $S_k \leftarrow R_k + H_k P_{k|k-1} H_k^T$
    Compute the Optimal Kalman gain: $K_k \leftarrow P_{k|k-1} H_k^T S_k^{-1}$
    Update the *a posteriori* state estimate: $\hat{\mathbf{x}}_{k|k} \leftarrow \hat{\mathbf{x}}_{k|k-1} + K_k \tilde{\mathbf{y}}_K$
    Update the *a posteriori* estimate covariance: $P_{k|k} \leftarrow (I - K_k H_k) P_{k|k-1}$
    Obtain the measurement post-fit residual: $\tilde{\mathbf{y}}_{k|k} \leftarrow \mathbf{z}_k - H_k \hat{\mathbf{x}}_{k|k}$
    return $(\hat{\mathbf{x}}_{k|k}, P_{k|k}, \tilde{\mathbf{y}}_{k|k})$
end procedure