Notes

Filtering in C++

Amro Al Baali

April 18, 2021

Contents

1 Why this document?

This document is provided to explain and clarify the code uploaded with it. The repository includes examples of implementing filters, usually Kalman fitlers, in C++. The filters will be mainly implemented on

1. a linear system, and
2. a non-Euclidean nonlinear system (usually defined on a Lie group).

2 The Kalman filter

2.1 The system

Consider the linear ordinary differential equation (ODE) describing a mass-spring-damper system

$$m\ddot{x}(t) + b\dot{x}(t) + kx(t) = u(t), \tag{1}$$

where m is the mass, b is the damping, k is the spring constant, and $u(t)$ is the forcing function. The system (1) can be written in state space form

$$\begin{aligned}\dot{\mathbf{x}} &= \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u \qquad (2)\\ &= \boldsymbol{A}\mathbf{x} + \boldsymbol{B}u_t, \qquad (3)\end{aligned}$$

where

$$\mathbf{x} = \begin{bmatrix} x & \dot{x} \end{bmatrix}^{\mathsf{T}}, \tag{4}$$

and the time arguments (t) are dropped for brevity.

The disrete-time kinematic model is given by

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}u_{k-1}, \tag{5}$$

where the discrete-time system matrices $\mathbf{A}$ and $\mathbf{B}$ are computed using some discretization scheme. For the linear example above, the $\mathbf{A}$ matrix is given by

$$\begin{aligned}\mathbf{A} &= \exp\left(\boldsymbol{A}T_k\right), \qquad (6)\\ \mathbf{B} &= \int_0^{T_k} \exp(\boldsymbol{A}\alpha)\mathrm{d}\alpha\boldsymbol{B}, \qquad (7)\end{aligned}$$

where T_k is the sampling period [1].

The matrix $\mathbf{B}$ can be approximated using forward Euler to get

$$\mathbf{B} \approx T_k\boldsymbol{B}. \tag{8}$$

2.2 Process model

The discrete-time process model[1] is used in the *prediction* step of the Kalman filter. It is given by

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{L}\mathbf{w}_{k-1}, \tag{9}$$

where $\mathbf{x}_k \in \mathbb{R}^{n_x}$ is the state, $\mathbf{u}_k \in \mathbb{R}^{n_u}$ is the control input, and $\mathbf{w}_k \in \mathbb{R}^{n_w}$ where $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ is the process noise and $\mathbf{Q}_k$ is the process noise covariance.

[1] Also referred to as the kinematic model, motion model, progression model, *e.t.c.*.

2.3 Measurement functions

The correction step requires a measurement model which is given by

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{M}\mathbf{n}_k, \tag{10}$$

where $\mathbf{y}_k \in \mathbb{R}^{n_y}$ and $\mathbf{n}_k \in \mathbb{R}^{n_n}$ where $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ is the measurement noise and $\mathbf{R}_k$ is the measurement noise covariance.

For the example presented, the measurement is a position measurememnt, so the measurement function is given by

$$\begin{aligned} y_k &= \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}_k + n_k \qquad (11)\\ &= \mathbf{C}\mathbf{x}_k + n_k. \qquad (12) \end{aligned}$$

3 The extended Kalman filter

Without getting into the derivation of the equations, the (extended) Kalman filter equations are given by [2, eq. (4.32)]

$$\begin{aligned}
\check{\mathbf{x}}_k &= \mathbf{f}\left(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0}\right), && (13a)\\
\check{\mathbf{P}}_k &= \mathbf{A}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{A}_{k-1}^{\mathsf{T}} + \mathbf{L}_{k-1}\mathbf{Q}_k\mathbf{L}_{k-1}^{\mathsf{T}}, && (13b)\\
\mathbf{K}_k &= \check{\mathbf{P}}_k\mathbf{H}_k\left(\mathbf{H}_k\check{\mathbf{P}}_k\mathbf{H}_k^{\mathsf{T}} + \mathbf{M}_k\mathbf{R}_k\mathbf{M}_k^{\mathsf{T}}\right)^{-1}, && (13c)\\
\hat{\mathbf{x}}_k &= \check{\mathbf{x}}_k + \mathbf{K}_k\left(\mathbf{y}_k + \mathbf{g}_k\left(\check{\mathbf{x}}_k, \mathbf{0}\right)\right), && (13d)\\
\hat{\mathbf{P}}_k &= \left(\mathbf{1} - \mathbf{K}_k\mathbf{H}_k\right)\check{\mathbf{P}}_k\left(\mathbf{1} - \mathbf{K}_k\mathbf{H}_k\right)^{\mathsf{T}} \\
&\quad + \mathbf{K}_k\mathbf{M}_k\mathbf{R}_k\mathbf{M}_k^{\mathsf{T}}\mathbf{K}_k^{\mathsf{T}}, && (13e)
\end{aligned}$$

where

$$\mathbf{A}_{k-1} = \left.\frac{\partial \mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}\right)}{\partial \mathbf{x}_{k-1}}\right|_{\substack{\mathbf{x}_{k-1}=\hat{\mathbf{x}}_{k-1},\\ \mathbf{w}_{k-1}=\mathbf{0}}}, \tag{14}$$

$$\mathbf{L}_{k-1} = \left.\frac{\partial \mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}\right)}{\partial \mathbf{w}_{k-1}}\right|_{\substack{\mathbf{x}_{k-1}=\hat{\mathbf{x}}_{k-1},\\ \mathbf{w}_{k-1}=\mathbf{0}}}, \tag{15}$$

$$\mathbf{H}_{k-1} = \left.\frac{\partial \mathbf{g}\left(\mathbf{x}_k, \mathbf{n}_k\right)}{\partial \mathbf{x}_k}\right|_{\substack{\mathbf{x}_k=\check{\mathbf{x}}_k,\\ \mathbf{n}_k=\mathbf{0}}}, \tag{16}$$

$$\mathbf{M}_{k-1} = \left.\frac{\partial \mathbf{g}\left(\mathbf{x}_k, \mathbf{n}_k\right)}{\partial \mathbf{n}_k}\right|_{\substack{\mathbf{x}_k=\check{\mathbf{x}}_k,\\ \mathbf{n}_k=\mathbf{0}}}. \tag{17}$$

The covariance equations (13b) and (13e) are computed using first-order covariance propagation on (13a) and (13d), respectively. Let's clarify this point as it will be important when discussing the invariant extended Kalman filter.

Remark 3.1. A Gaussian random variable

$$\underline{\mathbf{x}} \sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right) \tag{18}$$

can be written as

$$\underline{\mathbf{x}} = \boldsymbol{\mu} + \delta\underline{\mathbf{x}}, \tag{19}$$

$$\delta\underline{\mathbf{x}} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}\right). \tag{20}$$

Using Remark 3.1, define the (random) variables

$$\delta\check{\underline{\mathbf{x}}}_k \coloneqq \check{\underline{\mathbf{x}}}_k - \mathbf{x}_k, \tag{21}$$

$$\delta\hat{\underline{\mathbf{x}}}_k \coloneqq \hat{\underline{\mathbf{x}}}_k - \mathbf{x}_k, \tag{22}$$

where

$$\delta\check{\underline{\mathbf{x}}}_k \sim \mathcal{N}\left(\mathbf{0}, \check{\mathbf{P}}_k\right), \tag{23}$$

$$\delta\hat{\underline{\mathbf{x}}}_k \sim \mathcal{N}\left(\mathbf{0}, \hat{\mathbf{P}}_k\right). \tag{24}$$

Using Taylor's expansion of (13a), the error dynamics of the (extended) Kalman filter equations are then given by

$$\begin{aligned}
\delta\check{\underline{\mathbf{x}}}_k &= \check{\underline{\mathbf{x}}}_k - \mathbf{x}_k && (25)\\
&= \mathbf{f}\left(\hat{\underline{\mathbf{x}}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0}\right) - \mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \underline{\mathbf{w}}_{k-1}\right) && (26)\\
&\approx \mathbf{f}\left(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0}\right) - \mathbf{f}\left(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0}\right) \\
&\qquad - \mathbf{A}_{k-1}\underbrace{\left(\mathbf{x}_{k-1} - \hat{\underline{\mathbf{x}}}_{k-1}\right)}_{-\delta\hat{\underline{\mathbf{x}}}_{k-1}} - \mathbf{L}_{k-1}\left(\underline{\mathbf{w}}_{k-1} - \mathbf{0}\right) && (27)\\
&= \mathbf{A}_{k-1}\delta\hat{\underline{\mathbf{x}}}_{k-1} - \mathbf{L}_{k-1}\underline{\mathbf{w}}_{k-1}. && (28)
\end{aligned}$$

The covariance on $\delta\check{\mathbf{x}}_k$ is then given by

$$\operatorname{Cov}\left[\delta\check{\underline{\mathbf{x}}}_k\right] = \mathbf{A}_k\hat{\mathbf{P}}_{k-1}\mathbf{A}_{k-1}^{\mathsf{T}} + \mathbf{L}_{k-1}\mathbf{Q}_{k-1}\mathbf{L}_{k-1}^{\mathsf{T}} \tag{29}$$

which is equivalent to (13b).

Applying the same concept on the correction equation gives

$$\begin{aligned}
\delta\hat{\underline{\mathbf{x}}}_k &= \check{\underline{\mathbf{x}}}_k + \mathbf{K}_k\left(\underline{\mathbf{y}}_k - \mathbf{g}\left(\check{\underline{\mathbf{x}}}_k, \mathbf{0}\right)\right) - \mathbf{x}_k && (30)\\
&= \check{\underline{\mathbf{x}}}_k + \mathbf{K}_k\left(\mathbf{g}\left(\mathbf{x}_k, \underline{\mathbf{n}}_k\right) - \mathbf{g}\left(\check{\underline{\mathbf{x}}}_k, \mathbf{0}\right)\right) - \mathbf{x}_k && (31)\\
&= \mathbf{x}_k + \delta\check{\underline{\mathbf{x}}}_k + \mathbf{K}_k\left(\mathbf{g}\left(\mathbf{x}_k, \underline{\mathbf{n}}_k\right) - \mathbf{g}\left(\check{\underline{\mathbf{x}}}_k, \mathbf{0}\right)\right) && (32)\\
&\approx \delta\check{\underline{\mathbf{x}}}_k + \mathbf{K}_k\left(\mathbf{g}\left(\check{\mathbf{x}}_k, \mathbf{0}\right) + \mathbf{H}_k\left(\mathbf{x}_k - \check{\underline{\mathbf{x}}}_k\right)\right. \\
&\qquad \left. + \mathbf{L}_k\left(\underline{\mathbf{n}}_k - \mathbf{0}\right) - \mathbf{g}\left(\check{\mathbf{x}}_k, \mathbf{0}\right)\right) && (33)\\
&= \delta\check{\underline{\mathbf{x}}}_k + \mathbf{K}_k\left(\mathbf{H}_k\left(\mathbf{x}_k - \check{\underline{\mathbf{x}}}_k\right) + \mathbf{L}_k\left(\underline{\mathbf{n}}_k - \mathbf{0}\right)\right) && (34)\\
&= \delta\check{\underline{\mathbf{x}}}_k + \mathbf{K}_k\left(-\mathbf{H}_k\delta\check{\underline{\mathbf{x}}}_k + \mathbf{L}_k\left(\underline{\mathbf{n}}_k - \mathbf{0}\right)\right) && (35)\\
&= \left(\mathbf{1} - \mathbf{K}_k\mathbf{H}_k\right)\delta\check{\underline{\mathbf{x}}}_k + \mathbf{K}_k\mathbf{M}_k\underline{\mathbf{n}}_k. && (36)
\end{aligned}$$

The covariance is then given by

$$\operatorname{Cov}\left[\delta\hat{\underline{\mathbf{x}}}_k\right] = \left(\mathbf{1} - \mathbf{K}_k\mathbf{H}_k\right)\check{\mathbf{P}}_k\left(\mathbf{1} - \mathbf{K}_k\mathbf{H}_k\right)^{\mathsf{T}} + \mathbf{K}_k\mathbf{M}_k\mathbf{R}_k\mathbf{M}_k^{\mathsf{T}}\mathbf{K}_k^{\mathsf{T}} \qquad (37)$$

which is equivalent to (13e).

4 The invariant extended Kalman filter

The invariant filter[3] is applicable to states that live in Lie groups. However, since the filter deals with random variables, it's important to know how to represent random variables living in Lie groups. That is, $\underline{\mathbf{X}} \in G$, where G is some group.

4.1 Random variables on Lie groups

In the Euclidean case, Remark 3.1 can be used to describe a random variable. However, how can we do that in a non-Euclidean case? Let's restrict ourselves with Lie groups.

A random variable (on a Lie group) can be given by

$$\begin{aligned}
\underline{\mathbf{X}} &= \bar{\mathbf{X}}\delta\underline{\mathbf{X}} && (38)\\
&= \bar{\mathbf{X}}\exp\left(\underline{\boldsymbol{\xi}}^{\wedge}\right) && (39)\\
&= \bar{\mathbf{X}} \overset{\mathrm{R}}{\oplus} \underline{\boldsymbol{\xi}}, && (40)
\end{aligned}$$

where $\overset{\mathrm{R}}{\oplus}$ is the 'right perturbation' operator[2] from [4] and

$$\underline{\boldsymbol{\xi}} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}\right). \qquad (41)$$

[2] There are multiple versions of the $\oplus$ operator such as left, left-invariant, right, and right-invariant.

This is the Lie-group version of Remark 3.1. Note that $\bar{\mathbf{X}} \in G$ and $\exp\left(\underline{\boldsymbol{\xi}}^{\wedge}\right) \in G$, thus $\underline{\mathbf{X}} \in G$ since G is a Lie group which is closed under multiplication.

4.2 Left-invariant perturbation

The left-invariant "addition" $\overset{\mathrm{LI}}{\oplus} : G \times \mathbb{R}^n \to G$ is defined by

$$\mathbf{X} \overset{\mathrm{LI}}{\oplus} \boldsymbol{\xi} := \mathbf{X} \exp\left(-\boldsymbol{\xi}^{\wedge}\right) \tag{42}$$
$$= \mathbf{X}\, \mathrm{Exp}\left(-\boldsymbol{\xi}\right), \tag{43}$$

and the left-invariant "subtraction" $\overset{\mathrm{LI}}{\ominus} : G \times G \to \mathbb{R}^n$ is defined by

$$\mathbf{X}_2 \ominus \mathbf{X}_1 := \log\left(\mathbf{X}_2^{-1}\mathbf{X}_1\right)^{\vee} \tag{44}$$
$$= \mathrm{Log}\left(\mathbf{X}_2^{-1}\mathbf{X}_1\right). \tag{45}$$

References

[1] J. Farrell, *Aided navigation: GPS with high rate sensors.* McGraw-Hill, Inc., 2008.

[2] T. D. Barfoot, *State Estimation for Robotics.* Cambridge: Cambridge University Press, 2017. [Online]. Available: http://ebooks.cambridge.org/ref/id/CBO9781316671528

[3] A. Barrau and S. Bonnabel, "Invariant Kalman Filtering," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 237–257, May 2018. [Online]. Available: https://www.annualreviews.org/doi/10.1146/annurev-control-060117-105010

[4] J. Solà, J. Deray, and D. Atchuthan, "A micro Lie theory for state estimation in robotics," *arXiv:1812.01537 [cs]*, Jun. 2019, arXiv: 1812.01537. [Online]. Available: http://arxiv.org/abs/1812.01537