Notes

Invariant Filtering

Amro Al Baali

May 9, 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

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

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

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

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

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

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

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

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

where

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

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

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

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

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

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

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.

Let the $SE(n)$ process model be given by

$$\begin{aligned}
\underline{\mathbf{X}}_k &= \mathbf{F}\left(\underline{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1}, \underline{\mathbf{w}}_{k-1}\right) && (38)\\
&= \underline{\mathbf{X}}_{k-1}\operatorname{Exp}\left(T_{k-1}\mathbf{u}_{k-1}\right)\operatorname{Exp}(\underline{\mathbf{w}}_{k-1}) && (39)\\
&= \underline{\mathbf{X}}_{k-1}\boldsymbol{\Xi}_{k-1}\operatorname{Exp}(\underline{\mathbf{w}}_{k-1}), && (40)
\end{aligned}$$

$T_{k-1} := t_k - t_{k-1}$ is the sampling period.

the left-invariant measurement model is given by

$$\underline{\mathbf{y}}_k = \underline{\mathbf{X}}_k\mathbf{b} + \underline{\mathbf{n}}_k, \qquad (41)$$

and the right-invariant measurement model is given by

$$\underline{\mathbf{y}}_k = \underline{\mathbf{X}}_k^{-1}\mathbf{b} + \underline{\mathbf{n}}_k, \qquad (42)$$

where $\mathbf{b}$ is some known constant column matrix.

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

$$\underline{\mathbf{X}} = \bar{\mathbf{X}}\delta\underline{\mathbf{X}} \tag{43}$$
$$= \bar{\mathbf{X}} \exp\left(\underline{\boldsymbol{\xi}}^{\wedge}\right) \tag{44}$$
$$= \bar{\mathbf{X}} \overset{\mathrm{R}}{\oplus} \underline{\boldsymbol{\xi}}, \tag{45}$$

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

$$\underline{\boldsymbol{\xi}} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}\right). \tag{46}$$

[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

$$\mathbf{X} \overset{\mathrm{LI}}{\oplus} \boldsymbol{\xi} := \mathbf{X} \exp\left(-\boldsymbol{\xi}^{\wedge}\right) \tag{47}$$
$$= \mathbf{X}\,\mathrm{Exp}\left(-\boldsymbol{\xi}\right), \tag{48}$$

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

$$\mathbf{X}_2 \ominus \mathbf{X}_1 := \log\left(\mathbf{X}_2^{-1}\mathbf{X}_1\right)^{\vee} \tag{49}$$
$$= \mathrm{Log}\left(\mathbf{X}_2^{-1}\mathbf{X}_1\right). \tag{50}$$

4.3 The invariant extended Kalman filter

Without deep derivation, let's take the extended Kalman filter equations (13) and expand them to states that live on smooth manifolds. Let's simply replace the Euclidean $+$ and $-$ operators with $\oplus$ and $\ominus$, respectively[3].

[3] There are different 'flavors' as discussed above.

The prediction equations will then be

$$\check{\mathbf{X}}_k = \mathbf{F}\left(\hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0}\right), \tag{51}$$
$$\check{\mathbf{P}}_k = \mathbf{A}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{A}_{k-1} + \mathbf{L}_{k-1}\mathbf{Q}_{k-1}\mathbf{L}_{k-1}^{\mathsf{T}}. \tag{52}$$

But what are $\check{\mathbf{P}}_k$, $\mathbf{A}_{k-1}$, and $\mathbf{L}_{k-1}$ exactly? Remember, we had to define an error (left invariant, right invariant, etc.).

Similar to the estimator prediction error (21) and correction error (22) of the extended Kalman filter, define the estimator prediction and correction left-invariant errors as [4] [5]

$$\delta\underline{\check{\boldsymbol{\xi}}}_k := \mathbf{X}_k \overset{\mathrm{LI}}{\ominus} \underline{\check{\mathbf{X}}}_k, \tag{53}$$
$$\delta\underline{\hat{\boldsymbol{\xi}}}_k := \mathbf{X}_k \overset{\mathrm{LI}}{\ominus} \underline{\hat{\mathbf{X}}}_k, \tag{54}$$

[4] Note that $\mathbf{X}_k$ is not a random variable in the error definition.

[5] The definition $\delta\underline{\check{\boldsymbol{\xi}}} := \underline{\check{\mathbf{X}}} \overset{\mathrm{LI}}{\ominus} \mathbf{X}$ can also be used. However, the KF equations would look slightly different but the results would still hold.

respectively.

Plugging (38) and (51) into (53) results in

$$
\begin{aligned}
\delta\check{\underline{\boldsymbol{\xi}}}_k &= \mathbf{X}_k \overset{\mathrm{LI}}{\ominus} \check{\underline{\mathbf{X}}}_k && (55)\\
&= \mathbf{F}\left(\underline{\hat{\mathbf{X}}}_{k-1}, \mathbf{u}_{k-1}, \underline{\mathbf{w}}_{k-1}\right) \overset{\mathrm{LI}}{\ominus} \mathbf{F}\left(\hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0}\right) && (56)\\
&= \operatorname{Log}\left(\left(\mathbf{X}_{k-1}\boldsymbol{\Xi}_{k-1}\operatorname{Exp}(\underline{\mathbf{w}}_{k-1})\right)^{-1}\hat{\underline{\mathbf{X}}}_{k-1}\boldsymbol{\Xi}_{k-1}\right) && (57)\\
&= \operatorname{Log}\left(\operatorname{Exp}(-\underline{\mathbf{w}}_{k-1})\boldsymbol{\Xi}_{k-1}^{-1}\mathbf{X}_{k-1}^{-1}\hat{\underline{\mathbf{X}}}_{k-1}\boldsymbol{\Xi}_{k-1}\right) && (58)\\
&= \operatorname{Log}\left(\operatorname{Exp}(-\underline{\mathbf{w}}_{k-1})\boldsymbol{\Xi}_{k-1}^{-1}\operatorname{Exp}(\hat{\underline{\boldsymbol{\xi}}}_{k-1})\boldsymbol{\Xi}_{k-1}\right) && (59)\\
&= \operatorname{Log}\left(\operatorname{Exp}(-\underline{\mathbf{w}}_{k-1})\operatorname{Exp}\left(\operatorname{Adj}_{\boldsymbol{\Xi}_{k-1}^{-1}}\hat{\underline{\boldsymbol{\xi}}}_{k-1}\right)\right) && (60)\\
&\approx \underbrace{\operatorname{Adj}_{\boldsymbol{\Xi}_{k-1}^{-1}}}_{\mathbf{A}_{k-1}}\hat{\underline{\boldsymbol{\xi}}}_{k-1} - \underline{\mathbf{w}}_{k-1}, && (61)
\end{aligned}
$$

Note the usage of $\operatorname{Exp}(\hat{\underline{\boldsymbol{\xi}}}_{k-1}) = \mathbf{X}_{k-1}^{-1}\hat{\underline{\mathbf{X}}}_{k-1}$.

Note $\mathbf{X}\operatorname{Exp}(\boldsymbol{\xi})\mathbf{X}^{-1} = \operatorname{Exp}(\operatorname{Adj}_{\mathbf{X}}\boldsymbol{\xi})$.

where the last equation is an approximation from the BCH formula [2] and $\mathbf{L}_{k-1} = -\mathbf{1}$.

The covariance on the prediction error is then given by

$$
\begin{aligned}
\check{\mathbf{P}}_k &:= \operatorname{Cov}\left[\check{\underline{\boldsymbol{\xi}}}_k\right] && (62)\\
&\approx \mathbf{A}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{A}_{k-1}^{\mathsf{T}} + \mathbf{L}_{k-1}\mathbf{Q}_{k-1}\mathbf{L}_{k-1}^{\mathsf{T}}. && (63)
\end{aligned}
$$

The correction equations can be generalized from (13d) by using '$\overset{\mathrm{LI}}{\oplus}$' in place of '+'. Specifically,

$$
\hat{\underline{\mathbf{X}}}_k = \check{\underline{\mathbf{X}}}_k \overset{\mathrm{LI}}{\oplus} \left(\mathbf{K}_k(\mathbf{y}_k - \mathbf{g}(\check{\underline{\mathbf{X}}}_k, \mathbf{0}))\right). \quad (64)
$$

What's the covariance on $\hat{\underline{\mathbf{X}}}_k$? [6] To answer the question, lets' stick with the left-invariant measurement function (41) and plug in the appropriate variables into (54).

[6] Actually, covariance is on $\hat{\underline{\boldsymbol{\xi}}}_k$.

$$
\begin{aligned}
\hat{\underline{\boldsymbol{\xi}}}_k &= \mathbf{X}_k \overset{\mathrm{LI}}{\ominus} \hat{\underline{\mathbf{X}}}_k && (65)\\
&= \mathbf{X}_k \overset{\mathrm{LI}}{\ominus} \left(\check{\underline{\mathbf{X}}}_k \overset{\mathrm{LI}}{\oplus} (\mathbf{K}_k\underline{\mathbf{z}}_k)\right) && (66)\\
&= \operatorname{Log}\left(\mathbf{X}_k^{-1}\check{\underline{\mathbf{X}}}_k\operatorname{Exp}(-\mathbf{K}_k\underline{\mathbf{z}}_k)\right) && (67)\\
&= \operatorname{Log}(\operatorname{Exp}(\check{\underline{\boldsymbol{\xi}}}_k)\operatorname{Exp}(-\mathbf{K}_k\underline{\mathbf{z}}_k)) && (68)\\
&\approx \check{\underline{\boldsymbol{\xi}}}_k - \mathbf{K}_k\underline{\mathbf{z}}_k. && (69)
\end{aligned}
$$

What's the innovation $\underline{\mathbf{z}}_k$? That's where invariant filtering comes in. Define the left-invariant innovation[7] by

[7] The left-invariant innovation is used since we assumed we have a left-invariant measurement function.

$$
\begin{aligned}
\underline{\mathbf{z}}_k &:= \underline{\check{\mathbf{X}}}_k^{-1} \left(\underline{\mathbf{y}}_k - \mathbf{g}(\underline{\check{\mathbf{X}}}_k, \mathbf{0}) \right) && (70)\\
&= \underline{\check{\mathbf{X}}}_k^{-1} \left(\mathbf{X}_k \mathbf{b} + \underline{\mathbf{n}}_k - \check{\mathbf{X}}_k \mathbf{b} \right) && (71)\\
&= \underline{\check{\mathbf{X}}}_k^{-1} \mathbf{X}_k \mathbf{b} + \underline{\check{\mathbf{X}}}_k^{-1} \underline{\mathbf{n}}_k - \mathbf{b} && (72)\\
&= \operatorname{Exp}(-\underline{\check{\boldsymbol{\xi}}}_k) \mathbf{b} - \mathbf{b} \underline{\check{\mathbf{X}}}_k^{-1} \underline{\mathbf{n}}_k && (73)\\
&\approx \left(\mathbf{1} - \underline{\check{\boldsymbol{\xi}}}_k^{\wedge} \right) \mathbf{b} - \mathbf{b} + \underline{\check{\mathbf{X}}}_k^{-1} \underline{\mathbf{n}}_k && (74)\\
&= -\underline{\check{\boldsymbol{\xi}}}_k^{\wedge} \mathbf{b} + \underline{\check{\mathbf{X}}}_k^{-1} \underline{\mathbf{n}}_k && (75)\\
&= \underbrace{-\mathbf{b}^{\odot}}_{\mathbf{H}_k} \underline{\check{\boldsymbol{\xi}}}_k + \underbrace{\underline{\check{\mathbf{X}}}_k^{-1}}_{\mathbf{M}_k} \underline{\mathbf{n}}_k, && (76)
\end{aligned}
$$

Note that the Jacobian w.r.t. the state $\mathbf{H}_k$ is state-independent. That is, it does not depend on the state $\mathbf{X}_k$.

where

$$
\boldsymbol{\xi}^{\wedge} \mathbf{b} := \mathbf{b}^{\odot} \boldsymbol{\xi} \qquad (77)
$$

$$
(78)
$$

is defined

$$
\begin{bmatrix} \mathbf{b}_{1:3} \\ b_4 \end{bmatrix}^{\odot} = \begin{bmatrix} -\mathbf{b}_{1:3}^{\times} & b_4 \mathbf{1} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \qquad (79)
$$

for $SE(3)$ and

$$
\begin{aligned}
\begin{bmatrix} \mathbf{b}_{1:2} \\ b_3 \end{bmatrix}^{\odot} &= \begin{bmatrix} (1)^{\times} \mathbf{b}_{1:2} & b_3 \mathbf{1} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} && (80)\\
&= \begin{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \mathbf{b}_{1:2} & b_3 \mathbf{1} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} && (81)\\
& && (82)
\end{aligned}
$$

for $SE(2)$.

Plugging the above results into (69) gives the approximation

$$
\begin{aligned}
\underline{\hat{\boldsymbol{\xi}}}_k &= \underline{\check{\boldsymbol{\xi}}}_k - \mathbf{K}_k \underline{\mathbf{z}}_k && (83)\\
&\approx \underline{\check{\boldsymbol{\xi}}}_k - \mathbf{K}_k \left(\mathbf{H}_k \underline{\check{\boldsymbol{\xi}}}_k + \mathbf{M}_k \underline{\mathbf{n}}_k \right) && (84)\\
&= \left(\mathbf{1} - \mathbf{K}_k \mathbf{H}_k \right) \underline{\check{\boldsymbol{\xi}}}_k - \mathbf{K}_k \mathbf{M}_k \underline{\mathbf{n}}_k, && (85)
\end{aligned}
$$

which looks very similar to the Kalman filter correction equation with a difference of sign on $\underline{\mathbf{n}}_k$. The covariance on the correction error is then given by

$$
\begin{aligned}
\hat{\mathbf{P}}_k &:= \operatorname{Cov}\left[\underline{\hat{\boldsymbol{\xi}}}_k \right] && (86)\\
&\approx \left(\mathbf{1} - \mathbf{K}_k \mathbf{H}_k \right) \check{\mathbf{P}}_k \left(\mathbf{1} - \mathbf{K}_k \mathbf{H}_k \right)^{\mathsf{T}} + \mathbf{K}_k \mathbf{M}_k \mathbf{R}_k \mathbf{M}_k^{\mathsf{T}} \mathbf{K}_k^{\mathsf{T}}, && (87)
\end{aligned}
$$

which exactly matches the EKF correction covariance (13e).

5 Example: Left-invariant extended Kalman filter

In this section, an example of the left-invariant extended Kalman filter (L-InEKF) is presented. The example is applied on a robot in 3D space. That is, the states $\mathbf{X}$ live in the 3D special Euclidean Lie group $SE(3)$. A C++ implementation of this example is provided in the repository.

The state is given by

$$\mathbf{X}_k = \begin{bmatrix} \mathbf{C}_{ab_k} & \mathbf{r}_a^{b_k a} \\ \mathbf{0} & 1 \end{bmatrix} \tag{88}$$

$$= \operatorname{Exp}(\boldsymbol{\xi}_k), \tag{89}$$

where $\mathbf{C}_{ab} \in SO(3)$ is the attitude[8] and $\mathbf{r}_a^{b_k a}$ is the displacement of point b_k (robot body) relative to some arbitrary point a resolved in the (world) frame $\mathcal{F}_a$.[9] Furthermore, the Lie algebra coordinates are given by

$$\boldsymbol{\xi}_k = \begin{bmatrix} \boldsymbol{\xi}^{\phi} \\ \boldsymbol{\xi}^{\mathrm{r}} \end{bmatrix}, \tag{90}$$

where $\boldsymbol{\xi}^{\phi}$ are coordinates associated with attitude, and $\boldsymbol{\xi}^{\mathrm{r}}$ are the generalized position[10]. Note that some authors use different ordering for $\boldsymbol{\xi}$. For example, in [2], the ordering of $\boldsymbol{\xi}^{\phi}$ and $\boldsymbol{\xi}^{\mathrm{r}}$ is flipped. This has an effect on the Adjoint representation and the Jacobians, so care must be taken.

[8] Specifically, it's the DCM of frame $\mathcal{F}_a$ relative to frame $\mathcal{F}_b$.

[9] I realize that it might be confusing to use the same letter to denote a point and a frame.

[10] Note that $\boldsymbol{\xi}^{\mathrm{r}} \neq \mathbf{r}_a^{b_k a}$ in general.

5.1 Process model

The process model is given by (40) and the Jacobians are given by (61). Specifically, the Jacobian of the process model w.r.t. $\boldsymbol{\xi}$ is given by

$$\mathbf{A}_{k-1} = \operatorname{Adj}_{\boldsymbol{\Xi}_{k-1}^{-1}}, \tag{91}$$

where

$$\operatorname{Adj}_{\mathbf{X}} = \begin{bmatrix} \mathbf{C}_{ab} & \mathbf{0} \\ \mathbf{r}_a^{ba\times}\mathbf{C}_{ab} & \mathbf{C}_{ab} \end{bmatrix}, \tag{92}$$

$$\operatorname{Adj}_{\mathbf{X}^{-1}} = \begin{bmatrix} \mathbf{C}_{ab}^{\mathsf{T}} & \mathbf{0} \\ -\mathbf{C}_{ab}^{\mathsf{T}}\mathbf{r}_a^{ba\times} & \mathbf{C}_{ab}^{\mathsf{T}} \end{bmatrix}. \tag{93}$$

Note that the Adjoint representation depends on the ordering of $\boldsymbol{\xi}$ defined in (90).

5.2 Measurement model: GPS

Let the measurement model be a position sensor given by

$$\underline{\mathbf{y}}_k = \mathbf{r}_a^{b_k a} + \underline{\mathbf{n}}_k \tag{94}$$

$$= \begin{bmatrix} \mathbf{1} & \mathbf{0} \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{C}_{ab_k} & \mathbf{r}_a^{b_k a} \\ \mathbf{0} & 1 \end{bmatrix}}_{\mathbf{X}_k} \underbrace{\begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}}_{\mathbf{b}} + \underline{\mathbf{n}}_k. \tag{95}$$

To make things easier, let

$$\underbrace{\begin{bmatrix} \underline{\mathbf{y}}_k \\ 1 \end{bmatrix}}_{\tilde{\underline{\mathbf{y}}}_k} = \mathbf{X}_k \mathbf{b} + \underbrace{\begin{bmatrix} \underline{\mathbf{n}}_k \\ 0 \end{bmatrix}}_{\tilde{\underline{\mathbf{n}}}_k}. \tag{96}$$

The innovation (76) (using $\tilde{\underline{\mathbf{y}}}_k$) is then given by

$$\tilde{\underline{\mathbf{z}}}_k \approx -\mathbf{b}^{\odot} \check{\underline{\boldsymbol{\xi}}}_k + \check{\underline{\mathbf{X}}}_k^{-1} \tilde{\underline{\mathbf{n}}}_k \tag{97}$$

$$= -\begin{bmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \check{\underline{\boldsymbol{\xi}}}_k + \begin{bmatrix} \check{\underline{\mathbf{C}}}_{ab_k}^{\mathsf{T}} & -\check{\underline{\mathbf{C}}}_{ab_k}^{\mathsf{T}} \check{\underline{\mathbf{r}}}_a^{b_k a} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \underline{\mathbf{n}}_k \\ 0 \end{bmatrix} \tag{98}$$

$$= \begin{bmatrix} -\begin{bmatrix} \mathbf{0} & \mathbf{1} \end{bmatrix} \check{\underline{\boldsymbol{\xi}}}_k + \check{\underline{\mathbf{C}}}_{ab_k}^{\mathsf{T}} \underline{\mathbf{n}}_k \\ 0 \end{bmatrix} \tag{99}$$

$$= \begin{bmatrix} \underline{\mathbf{z}}_k \\ 0 \end{bmatrix}. \tag{100}$$

From [2], the $(\cdot)^{\odot}$ operator for $SE(3)$ is given by

$$\begin{bmatrix} \mathbf{b}_{1:3} \\ b_4 \end{bmatrix}^{\odot} = \begin{bmatrix} -\mathbf{b}_{1:3}^{\times} & b_4 \mathbf{1} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{101}$$

Ignoring the last row of the equation above gives the (modified) innovation

$$\underline{\mathbf{z}}_k = \underbrace{-\begin{bmatrix} \mathbf{0} & \mathbf{1} \end{bmatrix}}_{\mathbf{H}_k} \check{\underline{\boldsymbol{\xi}}}_k + \underbrace{\check{\underline{\mathbf{C}}}_{ab_k}^{\mathsf{T}}}_{\mathbf{M}_k} \underline{\mathbf{n}}_k. \tag{102}$$

Thus,

$$\mathbf{H}_k = -\begin{bmatrix} \mathbf{0} & \mathbf{1} \end{bmatrix}, \tag{103}$$

$$\mathbf{M}_k = \check{\underline{\mathbf{C}}}_{ab_k}^{\mathsf{T}}. \tag{104}$$

Note that the Jacobian of the innovation w.r.t. to $\check{\boldsymbol{\xi}}_k$ is state-independent.

5.3 Results

The results of using the $SE(2)$ C++ filter are presented in Figure 1-2.

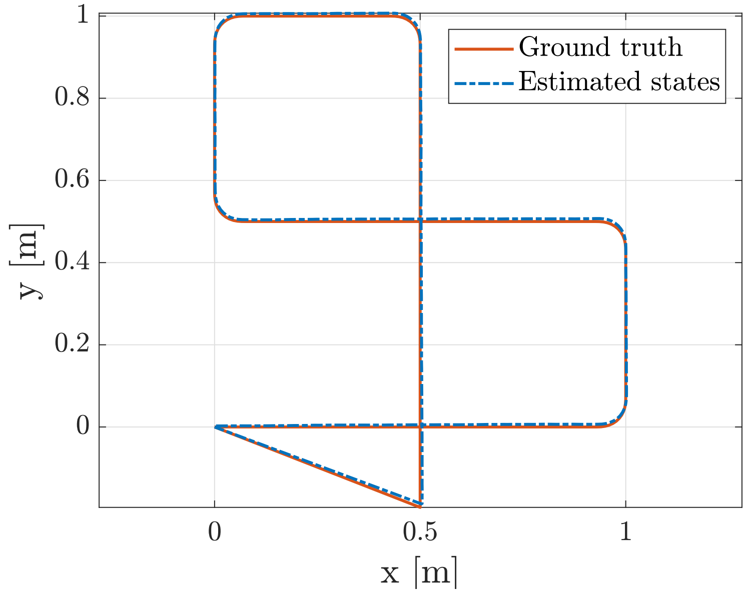

Figure 1: Ground truth and estimated trajectories from the $SE(2)$ example.

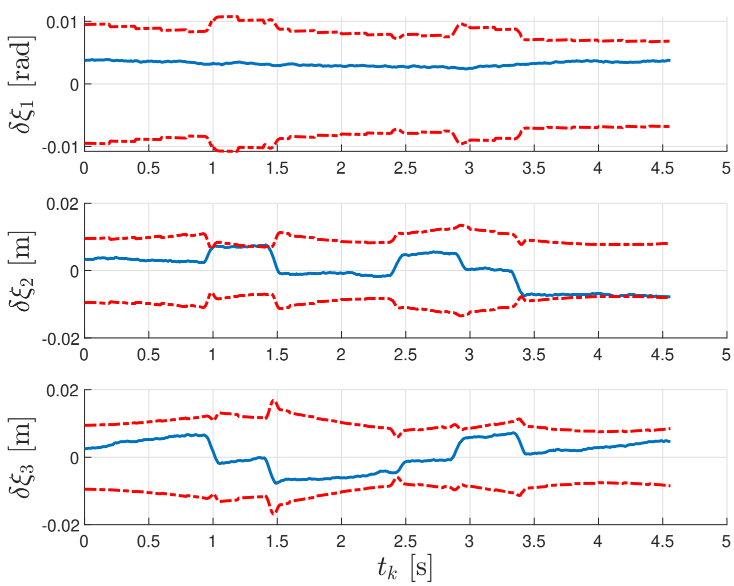

Figure 2: Error plots from the $SE(2)$ example.
Note that $\boldsymbol{\xi}_k = \mathrm{Log}(\mathbf{X}_k)$, where $\mathbf{X} \in SE(2)$ is the $SE(2)$ pose at time t_k.

Appendices

A Deriving the invariant extended Kalman filter

The filtering problem is a MAP problem given by

$$\hat{\mathbf{X}}_k = \underset{\mathbf{X}_k \in G}{\arg\max}\, p\left(\mathbf{X}_{k+1} \mid \hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{y}_k\right) \tag{105}$$

$$= \underset{\mathbf{X}_{k-1} \in G}{\arg\max}\, p\left(\mathbf{y}_k \mid \mathbf{X}_k\right) p\left(\mathbf{X}_k \mid \hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1}\right), \tag{106}$$

where $\hat{\mathbf{X}}_{k-1}$ is the InEKF estimate of $\mathbf{X}_{k-1}$.

As briefly discussed in Section 4.1, random variables on Lie groups can be described by

$$p\left(\mathbf{X}; \bar{\mathbf{X}}, \boldsymbol{\Sigma}\right) = \mathcal{N}\left(\mathbf{X} \ominus \bar{\mathbf{X}}, \boldsymbol{\Sigma}\right) \tag{107}$$

$$= \eta \exp\left(-\frac{1}{2}\left\|\mathbf{X} \ominus \bar{\mathbf{X}}\right\|^2_{\boldsymbol{\Sigma}^{-1}}\right). \tag{108}$$

Let

$$\check{\mathbf{X}}_k := \left(\hat{\mathbf{X}}_{k-1} \overset{\mathrm{R}}{\oplus} \mathbf{u}_{k-1}\right) \oplus \underline{\mathbf{w}}_{k-1}, \tag{109}$$

and let the measurement function be given by

$$\mathbf{y}_k = \mathbf{X}_k \mathbf{b} + \underline{\mathbf{n}}_k, \tag{110}$$

where $\mathbf{b}$ is a know column matrix.

Taking the above assumptions into consideration, and plugging (108) into the MAP objective function (106) and taking the negative log, results in

$$\hat{\mathbf{X}}_k = \underset{\mathbf{X}_k \in G}{\arg\max}\, p\left(\mathbf{y}_k \mid \mathbf{X}_k\right) p\left(\mathbf{X}_k \mid \hat{\mathbf{X}}_{k-1}, \mathbf{u}_{k-1}\right) \tag{111}$$

$$= \underset{\mathbf{X}_k \in G}{\arg\min}\, \frac{1}{2}\left\|\mathbf{y}_k - \mathbf{X}_k \mathbf{b}\right\|^2_{\mathbf{R}^{-1}} + \frac{1}{2}\left\|\mathbf{X}_k \ominus \check{\mathbf{X}}_k\right\|^2_{\check{\mathbf{P}}_k^{-1}}, \tag{112}$$

which is a (nonlinear) least squares problem.

The optimization problem can be rewritten as

$$\hat{\mathbf{X}}_k = \underset{\mathbf{X}_k \in G}{\arg\min}\, \frac{1}{2}\mathbf{e}(\mathbf{X}_k)^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} \mathbf{e}(\mathbf{X}_k), \tag{113}$$

where

$$\mathbf{e}: G \to \mathbb{R}^m, \tag{114}$$

$$\mathbf{e}(\mathbf{X}_k) = \begin{bmatrix} \mathbf{X}_k \ominus \check{\mathbf{X}}_k \\ \mathbf{X}_k \mathbf{b} - \mathbf{y}_k \end{bmatrix}, \tag{115}$$

$$\mathbf{\Sigma} = \begin{bmatrix} \check{\mathbf{P}}_k & \\ & \mathbf{R}_k \end{bmatrix}. \tag{116}$$

Remark A.1. Note that the error function $\mathbf{e}$ is a mapping from the (Lie) group space G to a the Lie algebra vector space $\mathbb{R}^m$. That's the nice thing about Lie algebra; we can use the standard optimization tools on the Lie algebra Euclidean space.

The optimization problem (113) can be solved using Gauss-Newton algorithm [5, 2, 6].

First, get an affine approximation of the error function (114) linearized around the operating point $\bar{\mathbf{X}}_k$. To do so, define the perturbation

$$\mathbf{d}_k := \mathbf{X}_k \ominus \bar{\mathbf{X}}_k, \tag{117}$$

$$\mathbf{X}_k = \bar{\mathbf{X}}_k \oplus \mathbf{d}_k, \tag{118}$$

according to the chosen error definition[11].

The variable name $\mathbf{d}_k \in \mathbb{R}^{n_x}$ is chosen because it'll be used as a search direction in the optimization.

[11] That is, replace '$\ominus$' and '$\oplus$' with the appropriate definition such as '$\overset{\text{LI}}{\oplus}$'.

Then, the affine approximation can be written as

$$\begin{aligned} \mathbf{e}(\mathbf{X}_k) &= \mathbf{e}(\bar{\mathbf{X}}_k \oplus \mathbf{d}_k) && (119) \\ &\approx \mathbf{e}\left(\bar{\mathbf{X}}_k\right) + \mathbf{J}\mathbf{d}_k, && (120) \end{aligned}$$

where $\mathbf{J} \in \mathbb{R}^{m \times n}$ is the Jacobian of the error function with respect to the Lie algebra coordinates. [12]

[12] Basically,

$$\mathbf{J} = \frac{\partial \mathbf{e}(\bar{\mathbf{X}}_k \oplus \boldsymbol{\xi}_k)}{\partial \boldsymbol{\xi}_k}, \tag{121}$$

where $\boldsymbol{\xi}_k \in \mathbb{R}^{n_k}$ is the coordinates of the Lie algebra components.

[13] Quadratic in $\mathbf{d}_k$.

Plugging the error affine approximation (119) into the objective function gives the quadratic[13] approximation

$$\begin{aligned} \tilde{J}(\mathbf{d}_k) &:= J(\bar{\mathbf{X}}_k \oplus \mathbf{d}_k) && (122) \\ &= \frac{1}{2}\mathbf{e}(\bar{\mathbf{X}}_k \oplus \mathbf{d}_k)^\mathsf{T} \mathbf{\Sigma}^{-1} \mathbf{e}(\bar{\mathbf{X}}_k \oplus \mathbf{d}_k) && (123) \\ &\approx \frac{1}{2}\left(\mathbf{e}\left(\bar{\mathbf{X}}_k\right) + \mathbf{J}\mathbf{d}_k\right)^\mathsf{T} \mathbf{\Sigma}^{-1} \left(\mathbf{e}\left(\bar{\mathbf{X}}_k\right) + \mathbf{J}\mathbf{d}_k\right) && (124) \\ &= \frac{1}{2}\mathbf{d}_k^\mathsf{T} \mathbf{J}^\mathsf{T} \mathbf{\Sigma}^{-1} \mathbf{J}\mathbf{d}_k + \mathbf{e}\left(\bar{\mathbf{X}}_k\right)^\mathsf{T} \mathbf{\Sigma}^{-1} \mathbf{J}\mathbf{d}_k + \frac{1}{2}\mathbf{e}\left(\bar{\mathbf{X}}_k\right)^\mathsf{T} \mathbf{\Sigma}^{-1} \mathbf{e}\left(\bar{\mathbf{X}}_k\right). && (125) \end{aligned}$$

If $\mathbf{J}$ is full column rank, then the quadratic function $\tilde{J}$ is strongly convex and has a unique minimizer. The minimizer is given by solving

$$\mathbf{d}_k^\star = -\left(\mathbf{J}^\mathsf{T} \mathbf{\Sigma}^{-1} \mathbf{J}\right)^{-1} \mathbf{J}^\mathsf{T} \mathbf{\Sigma}^{-1} \mathbf{e}(\bar{\mathbf{X}}_k), \tag{126}$$

where

$$\hat{\mathbf{P}}_k = \left(\mathbf{J}^\mathsf{T}\mathbf{\Sigma}^{-1}\mathbf{J}\right)^{-1} \tag{127}$$

at convergence [2].

We can try to solve for $\mathbf{d}_k$ analytically. Say the error function Jacobian is given by

$$\mathbf{J} = \begin{bmatrix} \gamma\mathbf{1} \\ \mathbf{C}_k \end{bmatrix}, \tag{128}$$

where $\gamma = \pm 1$ depends on the choice of the error definition.

A.1 The Kalman gain

Plugging (128) into (127) gives

$$\begin{aligned} \hat{\mathbf{P}}_k^{-1} &= \left(\mathbf{J}^\mathsf{T}\mathbf{\Sigma}^{-1}\mathbf{J}\right) && (129)\\ &= \check{\mathbf{P}}_k^{-1} + \mathbf{C}_k^\mathsf{T}\mathbf{R}_k^{-1}\mathbf{C}_k. && (130) \end{aligned}$$

Pre-multiplying by $\hat{\mathbf{P}}_k$ gives

$$\begin{aligned} \mathbf{1} &= \hat{\mathbf{P}}_k\check{\mathbf{P}}_k^{-1} + \underbrace{\hat{\mathbf{P}}_k\mathbf{C}_k^\mathsf{T}\mathbf{R}_k^{-1}}_{=:\mathbf{K}_k}\mathbf{C}_k && (131)\\ &= \hat{\mathbf{P}}_k\check{\mathbf{P}}_k^{-1} + \mathbf{K}_k\mathbf{C}_k, && (132) \end{aligned}$$

where $\mathbf{K}_k$ is the *Kalman gain*. Solving for $\hat{\mathbf{P}}_k$ results in

$$\hat{\mathbf{P}}_k = \left(\mathbf{1} - \mathbf{K}_k\mathbf{C}_k\right)\check{\mathbf{P}}_k. \tag{133}$$

Post-multiply by $\mathbf{C}_k^\mathsf{T}\mathbf{R}_k^{-1}$ gives

$$\begin{aligned} \underbrace{\hat{\mathbf{P}}_k\mathbf{C}_k^\mathsf{T}\mathbf{R}_k^{-1}}_{\mathbf{K}_k} &= \left(\mathbf{1} - \mathbf{K}_k\mathbf{C}_k\right)\check{\mathbf{P}}_k\mathbf{C}_k^\mathsf{T}\mathbf{R}_k^{-1} && (134)\\ \mathbf{K}_k &= \check{\mathbf{P}}_k\mathbf{C}_k^\mathsf{T}\mathbf{R}_k^{-1} - \mathbf{K}_k\mathbf{C}_k\check{\mathbf{P}}_k\mathbf{C}_k^\mathsf{T}\mathbf{R}_k^{-1} && (135)\\ \mathbf{K}_k\left(\mathbf{1} + \mathbf{C}_k\check{\mathbf{P}}_k\mathbf{C}_k^\mathsf{T}\mathbf{R}_k^{-1}\right) &= \check{\mathbf{P}}_k\mathbf{C}_k^\mathsf{T}\mathbf{R}_k^{-1} && (136) \end{aligned}$$

Post-multiply by $\mathbf{R}_k$

$$\mathbf{K}_k\left(\mathbf{R}_k + \mathbf{C}_k\check{\mathbf{P}}_k\mathbf{C}_k^\mathsf{T}\right) = \check{\mathbf{P}}_k\mathbf{C}_k^\mathsf{T}. \tag{137}$$

Finally, the Kalman gain is given by

$$\mathbf{K}_k = \check{\mathbf{P}}_k\mathbf{C}_k^\mathsf{T}\underbrace{\left(\mathbf{R} + \mathbf{C}_k\check{\mathbf{P}}_k\mathbf{C}_k^\mathsf{T}\right)^{-1}}_{\mathbf{S}_k^{-1}}. \tag{138}$$

A.2 The search direction

Plugging the posterior covariance (127) into the optimal search direction (126) results in

$$\begin{aligned}
\mathbf{d}_k^\star &= -\hat{\mathbf{P}}_k \mathbf{J}^\top \boldsymbol{\Sigma}^{-1} \mathbf{e}(\bar{\mathbf{X}}_k) && (139)\\
&= -\hat{\mathbf{P}}_k \begin{bmatrix} \gamma \mathbf{1} & \mathbf{C}_k^\top \end{bmatrix} \begin{bmatrix} \check{\mathbf{P}}_k^{-1} & \\ & \mathbf{R}_k^{-1} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{X}}_k \ominus \check{\mathbf{X}}_k \\ \bar{\mathbf{X}}_k \mathbf{b} - \mathbf{y}_k \end{bmatrix} && (140)\\
&= -\hat{\mathbf{P}}_k \left(\gamma \check{\mathbf{P}}_k^{-1} \left(\bar{\mathbf{X}}_k \ominus \check{\mathbf{X}}_k \right) + \mathbf{C}_k^\top \mathbf{R}_k^{-1} (\bar{\mathbf{X}}_k \mathbf{b} - \mathbf{y}_k) \right) && (141)\\
&= -\gamma \hat{\mathbf{P}}_k \check{\mathbf{P}}_k^{-1} \left(\bar{\mathbf{X}}_k \ominus \check{\mathbf{X}}_k \right) - \underbrace{\hat{\mathbf{P}}_k \mathbf{C}_k^\top \mathbf{R}_k^{-1}}_{\mathbf{K}_k} (\bar{\mathbf{X}}_k \mathbf{b} - \mathbf{y}_k) && (142)\\
&\overset{\star}{=} -\gamma \left(\mathbf{1} - \mathbf{K}_k \mathbf{C}_k \right) \left(\bar{\mathbf{X}}_k \ominus \check{\mathbf{X}}_k \right) - \underbrace{\hat{\mathbf{P}}_k \mathbf{C}_k^\top \mathbf{R}_k^{-1}}_{\mathbf{K}_k} (\bar{\mathbf{X}}_k \mathbf{b} - \mathbf{y}_k) && (143)\\
&= -\gamma \left(\bar{\mathbf{X}}_k \ominus \check{\mathbf{X}}_k \right) + \gamma \mathbf{K}_k \mathbf{C}_k \left(\bar{\mathbf{X}}_k \ominus \check{\mathbf{X}}_k \right) - \mathbf{K}_k \left(\bar{\mathbf{X}}_k \mathbf{b} - \mathbf{y}_k \right) && (144)\\
&= -\gamma \left(\bar{\mathbf{X}}_k \ominus \check{\mathbf{X}}_k \right) + \mathbf{K}_k \left(\gamma \mathbf{C}_k \left(\bar{\mathbf{X}}_k \ominus \check{\mathbf{X}}_k \right) + (\mathbf{y}_k - \bar{\mathbf{X}}_k \mathbf{b}) \right) && (145)\\
&= -\gamma \left(\bar{\mathbf{X}}_k \ominus \check{\mathbf{X}}_k \right) + \mathbf{K}_k \left(\gamma \mathbf{C}_k \left(\bar{\mathbf{X}}_k \ominus \check{\mathbf{X}}_k \right) + (\mathbf{y}_k - \bar{\mathbf{X}}_k \mathbf{b}) \right) && (146)\\
&= -\gamma \left(\bar{\mathbf{X}}_k \ominus \check{\mathbf{X}}_k \right) + \mathbf{K}_k \mathbf{z}_k (\bar{\mathbf{X}}_k), && (147)
\end{aligned}$$

⋆ From (132),

$$\hat{\mathbf{P}}_k \check{\mathbf{P}}_k^{-1} = \mathbf{1} - \mathbf{K}_k \mathbf{C}_k. \tag{148}$$

where

$$\mathbf{z}_k(\bar{\mathbf{X}}_k) = \gamma \mathbf{C}_k \left(\bar{\mathbf{X}}_k \ominus \check{\mathbf{X}}_k \right) + \mathbf{y}_k - \bar{\mathbf{X}}_k \mathbf{b} \tag{149}$$

is the innovation.

Finally, using (118), the posterior estimate is

$$\begin{aligned}
\hat{\mathbf{X}}_k &= \bar{\mathbf{X}} \oplus \mathbf{d}_k^\star && (150)\\
&= \bar{\mathbf{X}} \oplus \left(-\gamma \left(\bar{\mathbf{X}}_k \ominus \check{\mathbf{X}}_k \right) + \mathbf{K}_k \mathbf{z}_k (\bar{\mathbf{X}}_k) \right). && (151)
\end{aligned}$$

Note that if the linearization point is defined as

$$\bar{\mathbf{X}}_k \coloneqq \check{\mathbf{X}}_k, \tag{152}$$

then the innovation (149) simplifies to

$$\mathbf{z}_k(\check{\mathbf{X}}_k) = \mathbf{y}_k - \bar{\mathbf{X}}_k \mathbf{b}, \tag{153}$$

and the update equation (151) becomes

$$\begin{aligned}
\bar{\mathbf{X}}_k &= \check{\mathbf{X}}_k \oplus \left(\mathbf{K}_k \mathbf{z}_k (\check{\mathbf{X}}_k) \right) && (154)\\
&= \check{\mathbf{X}}_k \oplus \left(\mathbf{K}_k (\mathbf{y}_k - \check{\mathbf{X}}_k \mathbf{b}) \right). && (155)
\end{aligned}$$

This is the invariant extended Kalman filter correction.

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

[5] J. Nocedal and S. J. Wright, *Numerical optimization*, 2nd ed., ser. Springer series in operations research. New York: Springer, 2006, oCLC: ocm68629100.

[6] F. Dellaert and M. Kaess, “Factor Graphs for Robot Perception,” *Foundations and Trends in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017. [Online]. Available: http://www.nowpublishers.com/article/Details/ROB-043