A Quick Guide for the Iterated Extended Kalman Filter on Manifolds

Jianzhu Huai[1], Xiang Gao[2]

***Abstract*— The extended Kalman filter (EKF) is a common state estimation method for discrete nonlinear systems. It recursively executes the propagation step as time goes by and the update step when a set of measurements arrives. In the update step, the EKF linearizes the measurement function only once. In contrast, the iterated EKF (IEKF) refines the state in the update step by iteratively solving a least squares problem. The IEKF has been extended to work with state variables on manifolds which have differentiable $\boxplus$ and $\boxminus$ operators, including Lie groups. However, existing descriptions are often long, deep, and even with errors. This note provides a quick reference for the IEKF on manifolds, using freshman-level matrix calculus. Besides the bare-bone equations, we highlight the key steps in deriving them.**

I. Problem and Assumptions

The problem is to estimate the state vector $\mathbf{x}$ and its covariance $\mathbf{P}$ given its dynamic equation $\mathbf{f}()$ and the external observations $\mathbf{z}$. We assume that the state vector $\mathbf{x}$ is an element of a differentiable manifold with a boxplus operator $\boxplus$ and a boxminus operator $\boxminus$, defined as

$$\mathbf{x} \boxplus \boldsymbol{\delta} = \mathbf{y}, \tag{1}$$
$$\mathbf{y} \boxminus \mathbf{x} = \boldsymbol{\delta}. \tag{2}$$

where $\mathbf{y}$ is another manifold element close to $\mathbf{x}$ with a distance vector $\boldsymbol{\delta}$. For an Euclidean space, $\boxplus$ is the common plus(+) and $\boxminus$ is the common minus (-). For elements of SO3, example definitions of $\boxplus$ and $\boxminus$ are given in (23) and (28). In the state estimation problem, we usually denote the estimate of $\mathbf{x}$ by $\widehat{\mathbf{x}}$, which is usually defined relative to $\mathbf{x}$ by

$$\widehat{\mathbf{x}} \boxplus \boldsymbol{\delta} = \mathbf{x}, \tag{3}$$
$$\mathbf{x} \boxminus \widehat{\mathbf{x}} = \boldsymbol{\delta}. \tag{4}$$

An alternative definition is $\mathbf{x} \boxplus \boldsymbol{\delta} = \widehat{\mathbf{x}}$, *e.g.*, in [1]. This definition leads to a less-intuitive Kalman update step and is uncommonly used.

For the extended Kalman filter (EKF), we use the dynamic equation discretized at time steps $t_1, \cdots, t_k$. Time steps are usually chosen according to the stamp of the measurement vector $\mathbf{z}$, *e.g.*, $\mathbf{z}_k = \mathbf{z}(t_k)$ is acquired at time t_k. Let's consider one EKF time step from t_{k-1} until t_k, which involves propagation with the discrete dynamic equation and update with the measurement $\mathbf{z}_k$. The discrete dynamic equation from t_{k-1} to t_k is given by

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}), \tag{5}$$

where $\mathbf{w}_{k-1}$ is the discrete noise at t_{k-1} with a Gaussian distribution $N(\mathbf{0}, \mathbf{Q}_{k-1})$. The observation equation at t_k is

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{n}_k, \tag{6}$$

where $\mathbf{n}_k$ is the discrete noise at t_k with a Gaussian distribution $N(\mathbf{0}, \mathbf{R}_k)$. For simplicity, we assume that the observation noise is additive and $\mathbf{z}_k$ is in a vector space. Indeed, many state estimation problems meet this assumption.

II. Iterated Extended Kalman Filter

Now we describe the estimation process of the iterated EKF (IEKF). Denote the a-posteriori state vector estimate at t_{k-1} by $\mathbf{x}^+_{k-1}$ and its covariance by $\mathbf{P}^+_{k-1}$. The propagation step propagates the state estimate to t_k by

$$\mathbf{x}^-_k = \mathbf{f}(\mathbf{x}^+_{k-1}, \mathbf{u}_{k-1}, \mathbf{0}), \tag{7}$$
$$\mathbf{P}^-_k = \mathbf{F}_{k-1}\mathbf{P}^+_{k-1}\mathbf{F}^\intercal_{k-1} + \mathbf{G}_{k-1}\mathbf{Q}_{k-1}\mathbf{G}^\intercal_{k-1}, \tag{8}$$

where the propagation Jacobians are

$$\mathbf{F}_{k-1} = \lim_{\boldsymbol{\epsilon}\to\mathbf{0}} \frac{\mathbf{f}(\mathbf{x}^+_{k-1} \boxplus \boldsymbol{\epsilon}, \mathbf{u}, \mathbf{w}) \boxminus \mathbf{f}(\mathbf{x}^+_{k-1}, \mathbf{u}, \mathbf{w})}{\boldsymbol{\epsilon}} \tag{9}$$
$$\mathbf{G}_{k-1} = \frac{\partial \mathbf{f}(\mathbf{x}^+_{k-1}, \mathbf{u}, \mathbf{w})}{\partial \mathbf{w}}\Big|_{\mathbf{w}=\mathbf{0}} \tag{10}$$

With the measurement $\mathbf{z}_k$, the EKF solves for the state vector $\mathbf{x}^+_k$ that minimizes the weighted square sum of the deviation from the predicted state vector $\mathbf{x}^-_k$ and the innovation term $\mathbf{r}_k = \mathbf{z}_k - \mathbf{h}(\mathbf{x}^+_k)$, *i.e.*,

$$\begin{aligned}\min_{\mathbf{x}^+_k} &(\mathbf{x}^+_k \boxminus \mathbf{x}^-_k)^\intercal \mathbf{P}^{-\,-1}_k (\mathbf{x}^+_k \boxminus \mathbf{x}^-_k) + \\ &(\mathbf{z}_k - \mathbf{h}(\mathbf{x}^+_k))^\intercal \mathbf{R}^{-1}_k (\mathbf{z}_k - \mathbf{h}(\mathbf{x}^+_k)).\end{aligned} \tag{11}$$

In contrast, the IEKF iteratively updates the state vector starting from $\mathbf{x}^+_{k,0} = \mathbf{x}^-_k$. In iteration j, it solves for an increment $\boldsymbol{\delta}_{k,j}$ to the current state vector estimate $\mathbf{x}^+_{k,j}$ that minimizes the weighted square sum of the deviation from the prediction $\mathbf{x}^-_k$ and the innovation $\mathbf{r}_{k,j} = \mathbf{z}_k - \mathbf{h}(\mathbf{x}^+_{k,j})$,

$$\begin{aligned}\min_{\boldsymbol{\delta}_{k,j}} &\left\| \mathbf{x}^+_{k,j} \boxminus \mathbf{x}^-_k + \mathbf{J}_{k,j}\boldsymbol{\delta}_{k,j} \right\|_{\mathbf{P}^{-\,-1}_k} + \\ &\left\| \mathbf{z}_k - \mathbf{h}(\mathbf{x}^+_{k,j}) - \mathbf{H}_{k,j}\boldsymbol{\delta}_{k,j} \right\|_{\mathbf{R}^{-1}_k}\end{aligned} \tag{12}$$

where the Jacobian matrices are defined as

$$\mathbf{H}_{k,j} = \lim_{\boldsymbol{\epsilon}\to\mathbf{0}} \frac{\mathbf{h}(\mathbf{x}^+_{k,j} \boxplus \boldsymbol{\epsilon}) - \mathbf{h}(\mathbf{x}^+_{k,j})}{\boldsymbol{\epsilon}} \tag{13}$$
$$\mathbf{J}_{k,j} = \lim_{\boldsymbol{\epsilon}\to\mathbf{0}} \frac{(\mathbf{x}^+_{k,j} \boxplus \boldsymbol{\epsilon} \boxminus \mathbf{x}^-_k) - (\mathbf{x}^+_{k,j} \boxminus \mathbf{x}^-_k)}{\boldsymbol{\epsilon}} \tag{14}$$

$\mathbf{J}_{k,j}$ is a square identity matrix if $\mathbf{x}$ is in a vector space. Otherwise, it can be well approximated by the square identity

[1]Jianzhu Huai is with the State Key Lab of Surveying, Mapping, and Remote Sensing, Wuhan University
[2]Xiang Gao was with the Technical University of Munich.

matrix. Setting the derivative of (12) to zero, we can get the solution for $\boldsymbol{\delta}_{k,j}$, which are used for the iterative update[1],

$$\begin{aligned}
\mathbf{S}_{k,j} &= \mathbf{H}_{k,j}\mathbf{L}_{k,j}\mathbf{P}_k^-\mathbf{L}_{k,j}^\intercal\mathbf{H}_{k,j}^\intercal + \mathbf{R}_k, &(15)\\
\mathbf{K}_{k,j} &= \mathbf{L}_{k,j}\mathbf{P}_k^-\mathbf{L}_{k,j}^\intercal\mathbf{H}_{k,j}^\intercal\mathbf{S}_{k,j}^{-1}, &(16)\\
\boldsymbol{\delta}_{k,j} &= \mathbf{K}_{k,j}\big[\mathbf{H}_{k,j}\mathbf{L}_{k,j}(\mathbf{x}_{k,j}^+ \boxminus \mathbf{x}_k^-)+ \\
&\quad \mathbf{z}_k - \mathbf{h}(\mathbf{x}_{k,j}^+)\big] - \mathbf{L}_{k,j}(\mathbf{x}_{k,j}^+ \boxminus \mathbf{x}_k^-), &(17)\\
\mathbf{x}_{k,j+1}^+ &= \mathbf{x}_{k,j}^+ \boxplus \boldsymbol{\delta}_{k,j}. &(18)
\end{aligned}$$

where $\mathbf{L}_{k,j} = \mathbf{J}_{k,j}^{-1}$. The covariance will be updated once at the end of all iterations,

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_{k,n}\mathbf{H}_{k,n})\mathbf{L}_{k,n}\mathbf{P}_k^-\mathbf{L}_{k,j}^\intercal \tag{19}$$

In summary, the IEKF algorithm is given in Algorithm 1.

Algorithm 1 The iterated extended Kalman filter algorithm

Input at time t_{k-1}: $\mathbf{x}_{k-1}^+$, $\mathbf{P}_{k-1}^+$, $\mathbf{u}_{k-1}$, $\mathbf{z}_k$, max iterations n, termination threshold ϵ
Output at time t_k: $\mathbf{x}_k^+$, $\mathbf{P}_k^+$
Propagate state and covariance with (7)(8) to $\mathbf{x}_k^-$, $\mathbf{P}_k^-$
$j \leftarrow 0$
$\mathbf{x}_{k,0}^+ \leftarrow \mathbf{x}_k^-$
while $j \neq n$ **do**
  update the state vector estimate with (15)-(18)
  $j \leftarrow j+1$
  if $\|\boldsymbol{\delta}_{k,j}\| < \epsilon$ **then**
    break
  end if
end while
$\mathbf{x}_k^+ = \mathbf{x}_{k,n}^+$
update the covariance with (19)

When the dimension of observations is high, the inversion in (16) can be intensive. There are two approaches to deal with this issue. One is to perform QR decomposition of $\mathbf{H}_{k,j}$ as done in MSCKF [2, (4.61)],

$$\begin{aligned}
\mathbf{H}_{k,j} &= \begin{bmatrix}\mathbf{Q}_1 & \mathbf{Q}_2\end{bmatrix}\begin{bmatrix}\mathbf{T}_H\\ \mathbf{0}\end{bmatrix}, &(20)\\
\underbrace{\mathbf{Q}_1^\intercal(\mathbf{z} - \mathbf{h}(\mathbf{x}_{k,j}^+))}_{\mathbf{r}_q} &= \mathbf{T}_H\boldsymbol{\delta}_{k,j} + \underbrace{\mathbf{Q}_1^\intercal\mathbf{n}_k}_{\mathbf{n}_q}, &(21)
\end{aligned}$$

where $\mathbf{n}_q$ is a Gaussian noise, $N(\mathbf{0}, \mathbf{R}_q = \mathbf{Q}_1^\intercal\mathbf{R}_k\mathbf{Q}_1)$. Then, substituting $\mathbf{r}_q$, $\mathbf{T}_H$, and $\mathbf{R}_q$ for $\mathbf{r}_{k,j}$, $\mathbf{H}_{k,j}$, and $\mathbf{R}_k$, the IEKF update proceeds as in (15)-(19). The other approach is to rewrite $\mathbf{K}$ as done in FAST-LIO [3], *i.e.*,

$$\mathbf{K}_{k,j} = [\mathbf{H}_{k,j}^\intercal\mathbf{R}_k^{-1}\mathbf{H}_{k,j} + (\mathbf{L}_{k,j}\mathbf{P}_k^-\mathbf{L}_{k,j}^\intercal)^{-1}]^{-1}\mathbf{H}_{k,j}^\intercal\mathbf{R}_k^{-1}. \tag{22}$$

The expression is equivalent to (16), but needs an extra covariance inversion.

[1]Thanks to Yarong Luo from WHU for checking the update equations.

III. Example Manifold SO3

As an example, we compute $\mathbf{F}_{k-1}$ and $\mathbf{J}_{k,j}$ when $\mathbf{x}$ is an SO3 element. Denote the orientation of $\{B\}$ frame relative to the $\{W\}$ frame by $\mathbf{R}_{WB}$, and its estimate by $\widehat{\mathbf{R}}_{WB}$. If we define $\boxplus$ and $\boxminus$ like

$$\begin{aligned}
\widehat{\mathbf{R}}_{WB} \boxplus \delta\theta_{WB} &= \widehat{\mathbf{R}}_{WB}\mathrm{Exp}(\delta\theta_{WB}) = \mathbf{R}_{WB}, &(23)\\
\mathbf{R}_{WB} \boxminus \widehat{\mathbf{R}}_{WB} &= \mathrm{Log}(\widehat{\mathbf{R}}_{WB}^\intercal\mathbf{R}_{WB}) = \delta\theta_{WB}, &(24)
\end{aligned}$$

as done in FAST-LIO [3], then $\mathbf{F}_{k-1}$ and $\mathbf{J}_{k,j}$ are

$$\begin{aligned}
\mathbf{F}_{k-1} &= \lim_{\boldsymbol{\epsilon}\to\mathbf{0}} \frac{1}{\boldsymbol{\epsilon}}\mathrm{Log}\Big[(\mathbf{R}_{WB_{k-1}}\mathrm{Exp}(\int_{t_{k-1}}^{t_k}\boldsymbol{\omega}_{WB}^B dt))^\intercal\cdot\\
&\quad \mathbf{R}_{WB_{k-1}}\mathrm{Exp}(\boldsymbol{\epsilon})\mathrm{Exp}(\int_{t_{k-1}}^{t_k}\boldsymbol{\omega}_{WB}^B dt)\Big] &(25)\\
&= \mathrm{Exp}(-\int_{t_{k-1}}^{t_k}\boldsymbol{\omega}_{WB}^B dt),\\
\mathbf{J}_{k,j} &= \lim_{\boldsymbol{\epsilon}\to\mathbf{0}} \frac{\mathrm{Log}(\mathbf{x}_k^{-\intercal}\mathbf{x}_{k,j}^+\mathrm{Exp}(\boldsymbol{\epsilon})) - \mathrm{Log}(\mathbf{x}_k^{-\intercal}\mathbf{x}_{k,j}^+)}{\boldsymbol{\epsilon}} &(26)\\
&= \mathbf{J}_r^{-1}(\delta\phi),\\
\delta\phi_j &= \mathbf{x}_{k,j}^+ \boxminus \mathbf{x}_k^-, &(27)
\end{aligned}$$

where we use the angular velocity $\boldsymbol{\omega}_{WB}^B$ relationship $\mathbf{R}_{WB_k} = \mathbf{R}_{WB_{k-1}}\mathrm{Exp}(\int_{t_{k-1}}^{t_k}\boldsymbol{\omega}_{WB}^B dt)$, and $\mathbf{x}$ denotes the rotation matrix $\mathbf{R}_{WB}$ for clarity. The above results agree with FAST-LIO equations.

Otherwise, if we define $\boxplus$ and $\boxminus$ like

$$\begin{aligned}
\widehat{\mathbf{R}}_{WB} \boxplus \delta\theta_{WB} &= \mathrm{Exp}(\delta\theta_{WB})\widehat{\mathbf{R}}_{WB} = \mathbf{R}_{WB} &(28)\\
\mathbf{R}_{WB} \boxminus \widehat{\mathbf{R}}_{WB} &= \mathrm{Log}(\mathbf{R}_{WB}\widehat{\mathbf{R}}_{WB}^\intercal) &(29)
\end{aligned}$$

as in [4], then $\mathbf{F}_{k-1}$ and $\mathbf{J}_{k,j}$ will be

$$\begin{aligned}
\mathbf{F}_{k-1} &= \mathbf{I}_3, &(30)\\
\mathbf{J}_{k,j} &= \lim_{\boldsymbol{\epsilon}\to\mathbf{0}} \frac{\mathrm{Log}(\mathrm{Exp}(\boldsymbol{\epsilon})\mathbf{x}_{k,j}^+\mathbf{x}_k^{-\intercal}) - \mathrm{Log}(\mathbf{x}_{k,j}^+\mathbf{x}_k^{-\intercal})}{\boldsymbol{\epsilon}} &(31)\\
&= \mathbf{J}_l^{-1}(\delta\phi),\\
\delta\phi_j &= \mathbf{x}_{k,j}^+ \boxminus \mathbf{x}_k^-. &(32)
\end{aligned}$$

IV. Related Work

The IEKF for the vector space had been discussed in *e.g.*, [5]. Later, it was extended to the state vector on differentiable manifolds, *e.g.*, [6]. However, the definition of the boxplus operator Jacobian $\mathbf{L}$ [6, (49)] is vague, and $\mathbf{L}$ was erroneously transposed in [6, (50-52)]. In fact, $\mathbf{L}$ relates the multiplicative increment on the manifold to the additive increment in the tangent space (see also (35)), *i.e.*,

$$\mathbf{L}(\phi) = \lim_{\boldsymbol{\epsilon}\to\mathbf{0}} \frac{\mathrm{Exp}(\phi + \boldsymbol{\epsilon}) \boxminus \mathrm{Exp}(\phi)}{\boldsymbol{\epsilon}} \tag{33}$$

where Exp is the exponential map [7] for the differentiable manifold. For vectors, $\mathrm{Exp}(\mathbf{x}) = \mathbf{x}$. For SO3 elements, $\mathrm{Exp}(\phi) = \mathbf{I}_3 + \sum_{k=1}^{\infty}\frac{1}{k!}\lfloor\phi\rfloor_\times^k$. Authors of FAST-LIO [8] also formulate the IEKF on manifolds, but at great length with many new notations. The lack of a quick-start reference for the IEKF on manifolds motivates this note. Note that

X. Gao's recent book "SLAM technology in autonomous driving and robotics" also presents a simplified IEKF for lidar-inertial odometry in chapter 8.3.

APPENDIX

Here are a few helpful equations for the above derivations.

$$\frac{\partial(\mathbf{a}+\mathbf{B}\mathbf{x})^{\mathsf{T}}\mathbf{P}^{-1}(\mathbf{a}+\mathbf{B}\mathbf{x})}{\partial\mathbf{x}} = 2(\mathbf{a}+\mathbf{B}\mathbf{x})^{\mathsf{T}}\mathbf{P}^{-1}\mathbf{B} \tag{34}$$

Handy Jacobians for SO3 elements are copied from [9],

$$\mathrm{Exp}(\phi+\delta\phi) \approx \mathrm{Exp}(\phi)\mathrm{Exp}(\mathbf{J}_r(\phi)\delta\phi) \tag{35}$$

$$\mathbf{J}_r(\phi) = \mathbf{I}_3 - \frac{1-\cos\theta}{\theta^2}\lfloor\phi\rfloor_\times + \frac{\theta-\sin\theta}{\theta^3}\lfloor\phi\rfloor_\times^2 \tag{36}$$

$$\mathrm{Exp}(\phi+\delta\phi) \approx \mathrm{Exp}(\mathbf{J}_l(\phi)\delta\phi)\mathrm{Exp}(\phi) \tag{37}$$

$$\mathbf{J}_r^{-1}(\phi) = \mathbf{I}_3 + \frac{1}{2}\lfloor\phi\rfloor_\times + \left(\frac{1}{\theta^2} - \frac{1+\cos\theta}{2\theta\sin\theta}\right)\lfloor\phi\rfloor_\times^2 \tag{38}$$

$$\mathbf{J}_r^{\mathsf{T}}(\phi) = \mathbf{J}_l(\phi) = \mathbf{J}_r(-\phi) \tag{39}$$

$$\mathbf{J}_r^{-\mathsf{T}}(\phi) = \mathbf{J}_l^{-1}(\phi) = \mathbf{J}_r^{-1}(-\phi) \tag{40}$$

REFERENCES

[1] C. Jekeli, *Inertial Navigation Systems with Geodetic Applications*. de Gruyter. [Online]. Available: https://www.degruyter.com/view/books/9783110800234/9783110800234/9783110800234.xml 1

[2] J. Huai, "Collaborative SLAM with crowdsourced data." [Online]. Available: https://etd.ohiolink.edu/pg_10?0::NO:10:P10_ACCESSION_NUM:osu1483669256597152 2

[3] W. Xu and F. Zhang, "FAST-LIO: A fast, robust LiDAR-inertial odometry package by tightly-coupled iterated Kalman filter," vol. 6, no. 2, pp. 3317–3324. [Online]. Available: https://ieeexplore.ieee.org/document/9372856/ 2

[4] J. Huai, Y. Lin, Y. Zhuang, C. Toth, and D. Chen, "Observability analysis and keyframe-based filtering for visual inertial odometry with full self-calibration," vol. 38, no. 5, pp. 3219–3237. [Online]. Available: http://arxiv.org/abs/2201.04989 2

[5] J. Havlík and O. Straka, "Performance evaluation of iterated extended Kalman filter with variable step-length," vol. 659, no. 1, p. 012022, publisher: IOP Publishing. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/659/1/012022 2

[6] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback," vol. 36, no. 10, pp. 1053–1072. [Online]. Available: https://doi.org/10.1177/0278364917728574 2

[7] Manfredo P. do Carmo, *Differential geometry of curves and surfaces*, 2nd ed. Dover Publications, Inc. 2

[8] D. He, W. Xu, and F. Zhang, "Kalman filters on differentiable manifolds." [Online]. Available: http://arxiv.org/abs/2102.03804 2

[9] M. R. Hartley, "Contact-aided state estimation on Lie groups for legged robot mapping and control," rosshart@umich.edu. [Online]. Available: https://deepblue.lib.umich.edu/handle/2027.42/151657 3