# Invariant Extended Kalman Filter

Kalin Norman, Easton Potokar

## 1 Introduction

The Extended Kalman Filter (EKF) has been a staple in industry for decades with a wide range of applications including finance, medicine, robotics, etc. It builds upon the Kalman Filter by linearizing the nonlinear state dynamics. This can introduce significant error, particularly if the system has extreme nonlinearities. Further, when dealing with state vectors that include rotations, generally done in Euler angles, singularities arise and normal distributions struggle to capture the uncertainty inherit in rotations.

We seek to implement the Invariant Extended Kalman Filter (IEKF), which leverages matrix Lie Group and Algebra structure inherit in rotations and poses to linearize error dynamics without any approximations. This leads to a much more robust filter with relatively simple implementation and application to robotics. We also release our implementation and simulators, done in python [3].

## 2 Related Work

Common approximations of nonlinear systems include the use of Extended Kalman Filters, which approximate a nonlinear system to the first order. The EKF is efficient but can struggle to converge to the global solution of the system when the initial input to the system is inaccurate. Invariant Extended Kalman Filters are able to overcome the limitations of the EKF and capture all of the nonlinearities in the system, resulting in a robust solution that will converge to the global solution of the problem, even with large errors present upon initialization.

Alternatively, there exists the Unscented Kalman Filter that seeks to track the mean and covariance through use of specially chosen sample points. This leads to less approximation error than the EKF (correct to the second order), but still has trouble tracking the non-gaussian distributions that may arise from propagating through a nonlinear system.

## 3 Problem Formulation

The problem posed is to correctly estimate the output of a nonlinear system with noise present in the inputs and measurements. The EKF introduces errors through assumption of a gaussian output of the nonlinear system and linearization. The goal is to avoid introducing these errors by doing a change of coordinates to a Lie Group. We will do this to two systems; one in a plane, SE(2), and one in 3D space, $\mathrm{SE}_2(3)$. These models will be as follows.

### 3.1 SE(2) - Unicycle

The Unicycle Model is given by

$$\dot{\theta} = \phi + w_\phi \qquad \dot{x} = (v + w_v)\cos(\theta) \qquad \dot{y} = (v + w_v)\sin(\theta)$$
$$z = \begin{bmatrix} x \\ y \end{bmatrix} + w_z$$

where the controls are given by $u = \begin{bmatrix} \phi & v \end{bmatrix}^T$ and are corrupted by $\begin{bmatrix} w_\phi & w_v \end{bmatrix}^T \sim \mathcal{N}(0, Q)$ and measurements $z$ are corrupted by $w_z \sim \mathcal{N}(0, R)$

### 3.2 $SE_2(3)$ - IMU Measurements

For the $SE_2(3)$ model, we use IMU measurements as our controls, and velocity measurements as measurements. This is given by

$$\dot{R} = R(\omega + w_\omega)^\wedge \qquad \dot{v} = R(a + w_a) + g \qquad \dot{p} = v$$
$$z = v + w_z$$

where the controls are given by acceleration and velocity from the IMU, $u = \begin{bmatrix}\omega & a\end{bmatrix}^T$ and are corrupted by $\begin{bmatrix}w_\omega & w_a\end{bmatrix}^T \sim \mathcal{N}(0, Q)$ and measurements $z$ are corrupted by $w_z \sim \mathcal{N}(0, R)$

## 4 Proposed Solution

### 4.1 Matrix Lie Groups

Most robot poses are SO(n) or SE(n), both of which are matrix Lie groups, and which we denote arbitrarily with $G$. The tangent space of these Lie groups are Lie algebras, denoted with $\mathfrak{g}$. Note there exists an invertible map $^\wedge : \mathbb{R}^d \to \mathfrak{g}$, where $d$ is the dimension of the Lie algebra (and consequently, group), as well as a local bijective map $\exp_m : \mathfrak{g} \to G$. We will often compound these using $\exp : \mathbb{R}^d \to G$.

### 4.2 Invariant Kalman Filter on SE(2)

The first step is transforming our model into elements of SE(2). Note the following is equivalent to the above.

$$\chi = \begin{bmatrix}\cos\theta & -\sin\theta & x \\ \sin\theta & \cos\theta & y \\ 0 & 0 & 1\end{bmatrix} \implies \frac{d}{dt}\chi = \begin{bmatrix}\cos\theta & -\sin\theta & x \\ \sin\theta & \cos\theta & y \\ 0 & 0 & 1\end{bmatrix} \begin{bmatrix}0 & -(\phi + w_\phi) & v + w_v \\ \phi + w_\phi & 0 & 0 \\ 0 & 0 & 0\end{bmatrix} = \chi(u + w)^\wedge$$
$$z = \chi \begin{bmatrix}0 \\ 0 \\ 1\end{bmatrix} + w_z = \chi b + w_z$$

where $u = \begin{bmatrix}v & 0 & \theta\end{bmatrix}^T$, $w = \begin{bmatrix}w_v & 0 & w_\phi\end{bmatrix}^T$, and $w_z$ has been augmented with a 0 as it's last element. We can solve this differential equation explicitly using the matrix exponential to use as a discrete system as follows

$$\chi = \chi_0 \exp_m((u + w)^\wedge t) \implies \chi_{n+1} = \chi_n \exp((u_n + w_n)\Delta t) \approx \chi_n \exp(u_n \Delta t)) \exp(w_n \Delta t))$$

where the final approximation comes from dropping higher order terms on the Baker-Campbell-Hausdorff (BCH) formula, which should be relatively small since our noise is generally relatively small. We will implement the Left Invariant Kalman Filter [2], and thus use the Left Invariant error given by $exp(\xi_{n|n-1}) = \chi_n^{-1}\hat{\chi}_{n|n-1}$. By stepping backward, letting $U = \exp(u\Delta t)$, and using various properties of the matrix exponential, we get

$$\begin{aligned}
\exp(\xi_{n|n-1}) &= \chi_n^{-1}\hat{\chi}_{n|n-1} \\
&= (\chi_{n-1} U \exp(w_n \Delta t))^{-1}\hat{\chi}_{n-1|n-1}U \\
&= \exp(-w_n\Delta t)U^{-1}\chi_{n-1}^{-1}\hat{\chi}_{n-1|n-1}U \\
&= \exp(-w_n\Delta t)U^{-1}\exp(\xi_{n-1|n-1})U \\
&= \exp(-w_n\Delta t)\exp_m(U^{-1}\xi_{n-1|n-1}{}^\wedge U) \\
&= \exp(-w_n\Delta t)\exp(Ad_{U^{-1}}\xi_{n-1|n-1}) \\
\implies \xi_{n|n-1} &= Ad_{U^{-1}}\xi_{n-1|n-1} - w_n\Delta t
\end{aligned}$$

which gives us a linear equation of our error, and thus makes our prediction step for $\Sigma_{n|n-1}$ the following

$$\Sigma_{n|n-1} = Ad_{U^{-1}}\Sigma_{n-1|n-1}Ad_{U^{-1}}^T + (\Delta t)^2 Q$$

while the prediction step for $\hat{\chi}_{n|n-1}$ will use the discretization above. Using an alternate innovation, we linearize similarly by using the first two terms of matrix exponential summation

$$\begin{aligned}
v &= \hat{\chi}_{n|n-1}^{-1}(z_n - \hat{z}_{n|n-1}) = \hat{\chi}_{n|n-1}^{-1}(\chi_n b + w_z - \hat{\chi}_{n|n-1}b) \\
&= \exp(-\xi_{n|n-1})b - b + \hat{\chi}_{n|n-1}^{-1}w_z \approx (I_3 - \xi_{n|n-1}\hat{})b - b + \hat{\chi}_{n|n-1}^{-1}w_z \\
&= -\xi_{n|n-1}\hat{}\, b + \hat{\chi}_{n|n-1}^{-1}w_z = -\begin{bmatrix} 0 & -\xi_3 & \xi_1 \\ \xi_3 & 0 & \xi_2 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \hat{\chi}_{n|n-1}^{-1}w_z \\
&= -\begin{bmatrix} \xi_1 \\ \xi_2 \\ 0 \end{bmatrix} + \hat{\chi}_{n|n-1}^{-1}w_z = -\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\xi_{n|n-1} + \hat{\chi}_{n|n-1}^{-1}w_z \triangleq -H\xi_{n|n-1} + \hat{\chi}_{n|n-1}^{-1}w_z
\end{aligned}$$

thus resulting in our innovation covariance being $S_n = H\Sigma_{n|n-1}H^T + \hat{\chi}_{n|n-1}^{-1}R\hat{\chi}_{n|n-1}^{-T}$. The rest of the update step proceeds as per the vanilla Kalman Filter. Also note, as the last element of the innovation is unneeded, in practice it and the last row of H are truncated.

### 4.3 Invariant Kalman Filter on $SE_2(3)$

The derivation here is nearly identical to the SE(2) case. WE follow a similar system as [4].

$$\begin{aligned}
\chi = \begin{bmatrix} R & v & p \\ 0_{1x3} & 1 & 0 \\ 0_{1x3} & 0 & 1 \end{bmatrix} \implies \frac{d}{dt}\chi &= \begin{bmatrix} R(\omega + w_\omega)\hat{} & R(a + w_a) + g & v \\ 0_{1x3} & 0 & 0 \\ 0_{1x3} & 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} R & v & p \\ 0_{1x3} & 1 & 0 \\ 0_{1x3} & 0 & 1 \end{bmatrix}\begin{bmatrix} (\omega + w_\omega)\hat{} & a + w_a + R^{-1}g & R^{-1}v \\ 0_{1x3} & 0 & 0 \\ 0_{1x3} & 0 & 0 \end{bmatrix} = \chi(u + w)\hat{} \\
z = \chi\begin{bmatrix} 0_3 \\ 1 \\ 0 \end{bmatrix} + w_z &= \chi b + w_z
\end{aligned}$$

where $u = \begin{bmatrix} a + R^{-1}g & R^{-1}v & \omega \end{bmatrix}^T$ are the controls with feedback linearization, $w = \begin{bmatrix} w_a & 0 & w_\omega \end{bmatrix}^T$, and $w_z$ has been augmented with two 0s as its final elements. Note this system is identical to the SE(2) case, so the IEKF derivation is identical, with a few minor changes such as $\hat{}$ map being evaluated differently in $SE_2(3)$ than in SE(2).

The largest change is that H is different due to b being different. Following an identical derivation for the innovation above, we arrive at

$$-H\xi_{n|n-1} + \chi_{n|n-1}^{-1}w_z \triangleq -\begin{bmatrix} 0_{3x3} & I_3 & 0_{3x3} \\ 0_{2x3} & 0_{2x3} & 0_{2x3} \end{bmatrix}\xi_{n|n-1} + \chi_{n|n-1}^{-1}w_z$$

where in practice, the last rows of 0s is again truncated.

## 5 Evaluation

The IEKF has been shown to be asymptotically stable under relatively mild conditions [1](Proposition 2), implying that the IEKF will converge to the true trajectory for nearly any starting state, an assurance the EKF lacks. We show this for both of the systems described in Section 4.

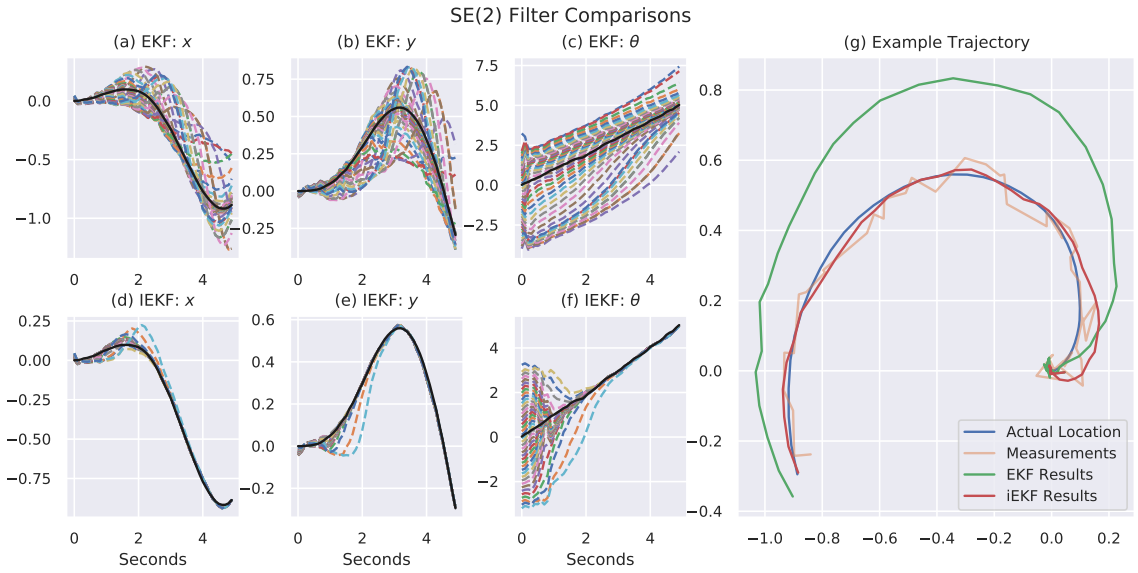


Figure 1: Comparison of the performance of the EKF and IEKF on SE(2) system. The simulation was run a total of 41 times for both filters, with the same system path and measurements, but with varied initial estimates of the system's starting pose. Plots (a), (b), and (c) show the pose estimates, broken down into $x$, $y$, and $\theta$ respectively, of each of the EKF simulations. The estimates are represented by dashed lines, while the true pose of the system is represented by a solid black line. Plots (d), (e), and (f) are constructed in the same fashion as those above, but represent the results of the IEKF. Lastly, plot (g) shows the path from the system, the measurements, and the EKF and IEKF estimates of the system's path. The same initial estimate was used for both the EKF and IEKF results shown in plot (g).

## 5.1 Performance of the EKF and IEKF on SE(2)

Using the unicycle model, defined in section 3.1, both the EKF and IEKF were implemented on the system. In order to verify that the IEKF outperforms the EKF, the unicycle path and measurements were generated and then each of the filters were run a total of 41 times. The unicycle's starting pose, represented as $(x, y, \theta)$, was $(0, 0, 0)$ and the estimated starting poses of the filters ranged from $(-20, -20, -\pi)$ to $(20, 20, \pi)$ with a step size of 1 for the $x$ and $y$ positions, and a step size of $\frac{2\pi}{40}$ for $\theta$. The simulation results are found in Figure 1 and demonstrate how the IEKF is able to converge very quickly to the model's true position, while the EKF converges much more slowly.

## 5.2 IEKF Implemented on $SE_2(3)$

The unicycle system is very simple, a more complex system presents a greater challenge for implementing the EKF, and greater non-linearities have a negative impact on its performance. The IEKF is also more difficult to implement on a more complex system, but much less so than the EKF. Section 4 provides a greater evaluation on the specifics of the IEKF implementations for each system.

A quad-rotor simulation was used to generate data for $SE_2(3)$ with a simulated IMU for the measurements. The data included additive noise in the controls, as shown in Section 4. A plot of the quad-rotor's true location, the pathway traced by the measurements, and the pathway that resulted from the IEKF is found in Figure 2. It is evident that the filter continues to perform exceptionally well, even in situations when the predicted path begins turns in the opposite direction

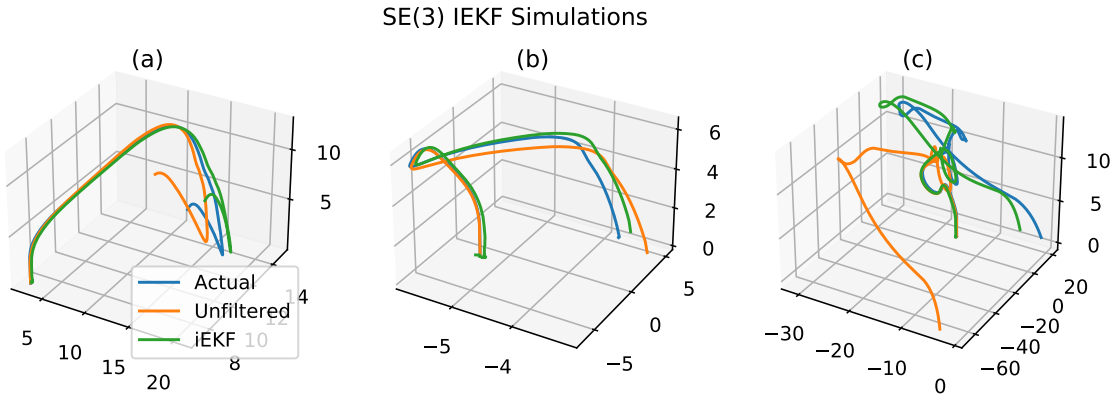


Figure 2: Three separate data sets of a quad-rotor flying, with the true path, predicted path (only using the measurements from the IMU), and the path from the IEKF for each data set. The quad-rotor was free to move about in three dimensional space, so all three plots show the position in coordinates of the form $(x, y, z)$. Plot (a) corresponds to the first set of data, plot (b) to the second, and plot (c) to the third.

of the truth, as is the case in plot (c) of Fig. 2. Note there is still some drift, but this is to be expected since no corrective measures were taken on the position, only on the velocity.

# 6 Conclusion

Ultimately, the EKF, though widely used, is an approximation that is unable to fully characterize nonlinearities in a system. With the use of an IEKF, those nonlinearities are more accurately characterized by the filter, resulting in an estimate that converges to the truth much more quickly than the EKF, and that overall remains more accurate in tracking the true pose. Additionally, the IEKF is much easier to implement for complex systems, so long as the system is found within a Lie Group. As with the EKF, there are circumstances where an IEKF is not able to converge due to the sensors or measurements used, but it will continue to outperform the EKF so long as there are nonlinearities present in the system.

# References

[1] A. Barrau and S. Bonnabel. "The Invariant Extended Kalman Filter as a Stable Observer". In: *IEEE Transactions on Automatic Control* 62.4 (2017), pp. 1797–1812. DOI: 10.1109/TAC.2016.2594085.

[2] Axel Barrau and Silvère Bonnabel. "Invariant Kalman Filtering". In: *Annual Review of Control, Robotics, and Autonomous Systems* 1.1 (2018), pp. 237–257. DOI: 10.1146/annurev-control-060117-105010. URL: https://doi.org/10.1146/annurev-control-060117-105010.

[3] Kalin Norman Easton Potokar. *Invariant Kalman Filter*. 2020. URL: https://github.com/contagon/iekf.

[4] Ross Hartley et al. "Contact-aided invariant extended Kalman filtering for robot state estimation". In: *The International Journal of Robotics Research* 39.4 (2020), pp. 402–430. DOI: 10.1177/0278364919894385. URL: https://doi.org/10.1177/0278364919894385.