Extended Kalman filter in $SE(2)$

Thomas Hitchcox
Department of Mechanical Engineering, McGill University
817 Sherbrooke Street West, Montreal QC H3A 0C3

May 18, 2020

1 Problem setup

In this problem, the position and attitude of a cart on the plane are estimated using an extended Kalman filter (EKF). The state vector is $\mathbf{x} = \begin{bmatrix} \theta_{bt} & \mathbf{r}_t^{zw\mathsf{T}} \end{bmatrix}^\mathsf{T}$. The vehicle is equipped with a wheel odometry sensor, which measures $v_{b^1}^{zw/t}$, and a gyroscope, which measures ω_b^{bt}. Gyro measurements are available at $50\,\mathrm{Hz}$, while wheel odometry measurements are available at $25\,\mathrm{Hz}$. Position (GPS) measurements are available at $5\,\mathrm{Hz}$. The layout is shown in Figure 1.

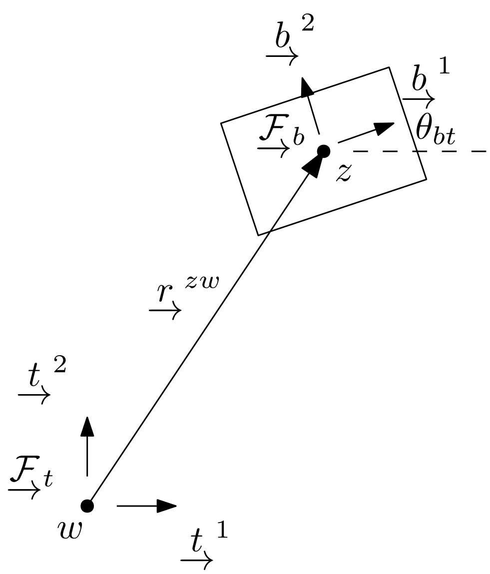

Figure 1: Laying out the $SE(2)$estimation problem

2 Data generation

This problem requires corrupted velocity and angular velocity resolved in body frame $\mathcal{F}_b$, and corrupted position resolved in tangent frame $\mathcal{F}_t$. For the purposes of generating gyro data, the tangent frame was treated as the inertial frame.

3 Deriving the process and measurement models

The continuous-time kinematics are

$$\begin{aligned}\dot{\mathbf{C}}_{bt} &= -\omega_b^{bt\wedge}\mathbf{C}_{bt},\\ \dot{\mathbf{r}}_t^{zw} &= \mathbf{C}_{bt}^{\mathsf{T}}\mathbf{v}_b^{zw/t}.\end{aligned} \tag{1}$$

Note that (1) may be simplified to

$$\dot{\theta}_{bt} = \omega_b^{bt}. \tag{2}$$

The interoceptive measurements are the rate gyro,

$$u_b^g = \omega_b^{bt} + w_b^g, \tag{3}$$

and the wheel odometry measurements,

$$u_b^w = v_{b^1}^{zw/t} + w_b^w, \tag{4}$$

where $w^g \sim \mathcal{N}(0, \sigma_g^2)$ and $w^w \sim \mathcal{N}(0, \sigma_w^2)$. Resolving all quantities in $\mathcal{F}_t$ yields

$$\begin{aligned}\omega_t^{bt} &= u_t^g - w_t^g,\\ \mathbf{v}_t^{zw/t} &= \mathbf{C}_{tb}^{\mathsf{T}}\big(\mathbf{u}_b^w - \mathbf{w}_b^w\big),\end{aligned} \tag{5}$$

where $\mathbf{u}_b^w = \begin{bmatrix} u_b^w & 0\end{bmatrix}^{\mathsf{T}}$, and similarly $\mathbf{w}_b^w = \begin{bmatrix} w_b^w & 0\end{bmatrix}^{\mathsf{T}}$. Discretizing (5) using a forward-Euler scheme and evaluating at $(\bar{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0})$ yields an expression for the discrete-time kinematic equations,

$$\begin{aligned}\theta_k &= \theta_{k-1} + T u_{k-1}^g,\\ \mathbf{r}_k &= \mathbf{r}_{k-1} + T\mathbf{C}_{k-1}^{\mathsf{T}}\mathbf{u}_{k-1}^w,\end{aligned} \tag{6}$$

To propagate covariance in the EKF framework, an expression is needed for the discrete-time error kinematics. To obtain this, first linearize (5) about an operating point, then discretize. Linearizing (5) by letting $w^g = \bar{w}^g - \delta w^g$, $\mathbf{C}_{tb}^{\mathsf{T}} = \bar{\mathbf{C}}_{bt}^{\mathsf{T}}(\mathbf{1} + \delta\theta^\wedge)$, and $\mathbf{w}^w = \bar{\mathbf{w}}^w - \delta\mathbf{w}^w$, subtracting the nominal solution, and ignoring higher-order terms yields

$$\begin{aligned}\dot{\bar{\theta}} + \delta\dot{\theta} &= u^g - \bar{w}^g + \delta w^g,\\ \delta\dot{\theta} &= \delta w^g,\end{aligned} \tag{7}$$

and

$$\begin{aligned}\dot{\bar{\mathbf{r}}} + \delta\dot{\mathbf{r}} &= \bar{\mathbf{C}}^{\mathsf{T}}\big(\mathbf{1} + \delta\theta^\wedge\big)\big(\mathbf{u}^w - \bar{\mathbf{w}}^w + \delta\mathbf{w}^w\big),\\ \delta\dot{\mathbf{r}} &= \bar{\mathbf{C}}^{\mathsf{T}}\delta\theta^\wedge\mathbf{u}^w + \bar{\mathbf{C}}^{\mathsf{T}}\delta\mathbf{w}^w,\\ &= \bar{\mathbf{C}}^{\mathsf{T}}\boldsymbol{\Omega}\,\mathbf{u}^w\delta\theta + \bar{\mathbf{C}}^{\mathsf{T}}\delta\mathbf{w}^w,\end{aligned} \tag{8}$$

where

$$\boldsymbol{\Omega} = \begin{bmatrix} 0 & -1\\ 1 & 0\end{bmatrix}. \tag{9}$$

Discretize (7) and (8) using a forward-Euler scheme,

$$\begin{aligned}\delta\theta_k &= \delta\theta_{k-1} + T\delta w_{k-1}^g,\\ \delta\mathbf{r}_k &= \delta\mathbf{r}_{k-1} + T\big(\mathbf{C}_{k-1}^{\mathsf{T}}\boldsymbol{\Omega}\,\mathbf{u}_{k-1}^w\delta\theta_{k-1} + \mathbf{C}_{k-1}^{\mathsf{T}}\delta\mathbf{w}_{k-1}^w\big).\end{aligned} \tag{10}$$

Finally, the discrete-time kinematics of (10) may be collected as

$$\begin{bmatrix}\delta\theta_k\\ \delta r_k^1\\ \delta r_k^2\end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0\\ -Tu_{k-1}^w\sin\theta_{k-1} & 1 & 0\\ Tu_{k-1}^w\cos\theta_{k-1} & 0 & 1\end{bmatrix}}_{\mathbf{A}_{k-1}}\begin{bmatrix}\delta\theta_{k-1}\\ \delta r_{k-1}^1\\ \delta r_{k-1}^2\end{bmatrix} + \underbrace{\begin{bmatrix} T & 0 & 0\\ 0 & T\cos\theta_{k-1} & 0\\ 0 & T\sin\theta_{k-1} & 0\end{bmatrix}}_{\mathbf{L}_{k-1}}\begin{bmatrix}\delta w_{k-1}^g\\ \delta w_{k-1}^w\\ 0\end{bmatrix} \tag{11}$$

The discrete-time measurement equation is simply

$$\mathbf{y}_k = \mathbf{r}_k^{zw} + \mathbf{w}_k^p, \tag{12}$$

therefore $\mathbf{C}_k = \begin{bmatrix}\mathbf{0} & \mathbf{1}\end{bmatrix}$, and $\mathbf{M}_k = \mathbf{1}$.

4 Implementation

Setting up and running the 2D EKF in MATLAB yields the results shown in Figures 2 and 3. For implementation details see `main_ekf_SE2.m`.

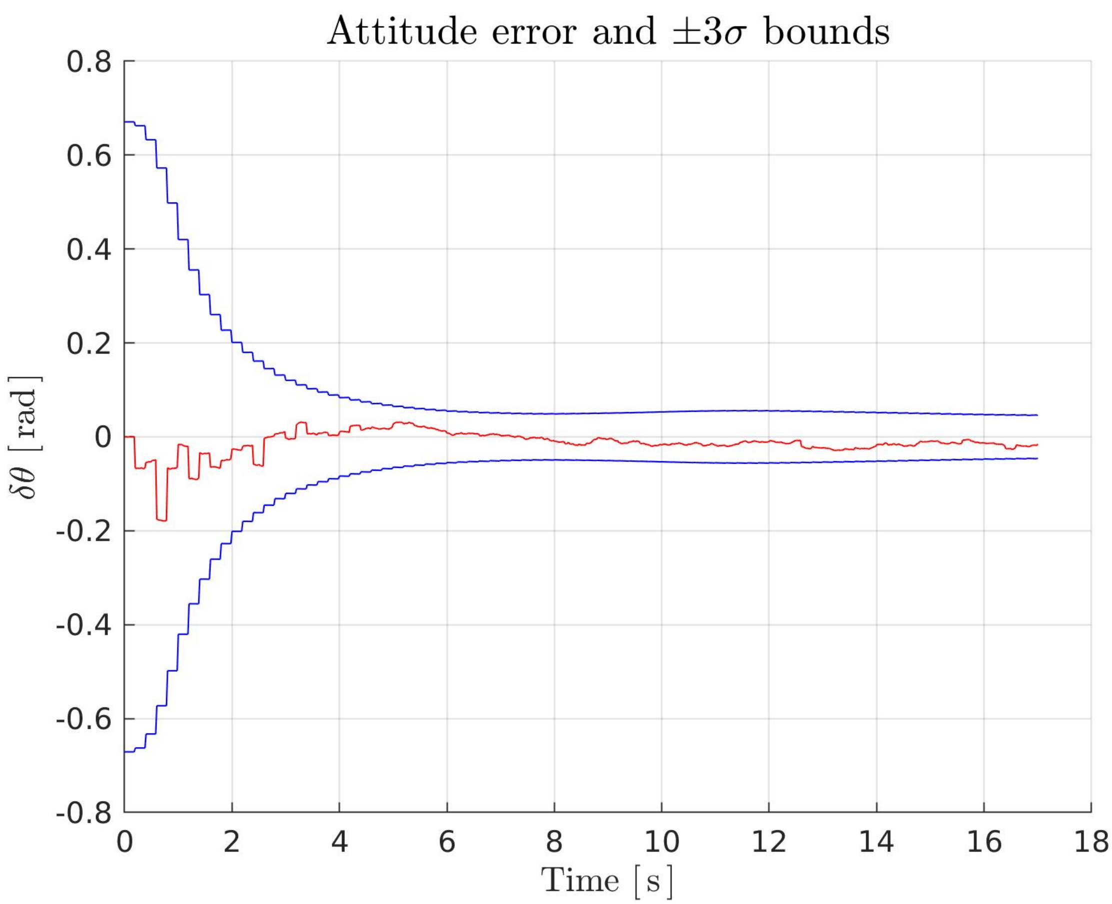

Figure 2: Filter results for attitude. Mean error is shown in red, $\pm 3\sigma$ bounds are shown in blue.

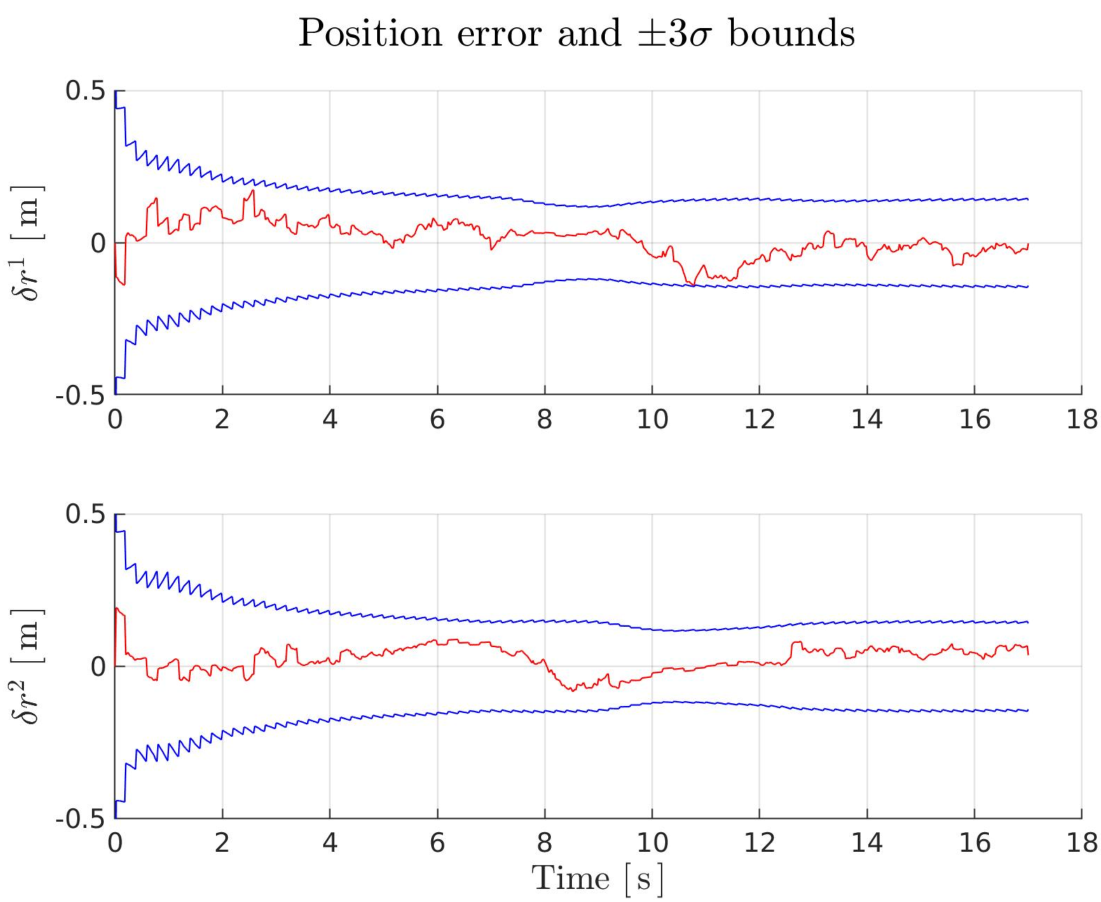

Figure 3: Filter results for position. Mean error is shown in red, $\pm 3\sigma$ bounds are shown in blue.