Lie-Theoretic Multi-Robot Localization

1. Abstract

This repository contains the code for *Lie-Theoretic Multi-Robot Localization* [5], an extension of *The Banana Distribution Is Gaussian: A Localization Study with Exponential Coordinates* [3], where data from two robots are fused. Notably, we consider 2-dimensional straight-line motion. The project is implemented in *MATLAB*.

2. Problem Setup

A team of two-wheeled differential-drive robots moves in the field, following a straight-line path based on the given inputs. However, due to the system's stochastic nature, errors accumulate over time, making odometry or dynamics alone insufficient for accurate pose estimation. Thus, incorporating relative pose measurements from neighboring robots helps update the robots' pose estimates.

3. Preliminaries

Considering a two-wheeled robot, we define its position as $p = [p_1, p_2]^{\mathrm{T}} \in \mathbb{R}^2$ and its orientation as $\theta \in S^1$. The orientation can also be represented by a rotation matrix $R \in SO(2)$, where the special orthogonal group $SO(2)$ is defined as

$$SO(2) = \{R \in \mathbb{R}^{2\times 2} \mid R^{\mathrm{T}}R = I, \det R = 1\}.$$

Specifically, the rotation matrix $R \in SO(2)$ is represented as

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

Instead, we can express the system using the homogeneous representation (known as the Special Euclidean Group $SE(2)$) as follows:

$$g(p_1, p_2, \theta) = \left\{ \begin{bmatrix} R & p \\ 0_{1\times 2} & 1 \end{bmatrix} \in \mathbb{R}^{3\times 3} \mid R \in SO(2),\ p \in \mathbb{R}^2 \right\}.$$

We will also make use of the Lie algebra $\mathfrak{se}(2)$ associated with $SE(2)$. For a vector $x = [v_1, v_2, \alpha]^{\mathrm{T}} \in \mathbb{R}^3$, an element X of the Lie algebra $\mathfrak{se}(2)$ can then be expressed as

$$X = \hat{x} = \begin{bmatrix} 0 & -\alpha & v_1 \\ \alpha & 0 & v_2 \\ 0 & 0 & 0 \end{bmatrix},$$

and

$$X^{\vee} = x,$$

where the $\wedge$ and $\vee$ operators allow us to map between $\mathbb{R}^3$ and $\mathfrak{se}(2)$. The exponential map $\exp(\cdot)$ provides an exact means of mapping elements from the Lie algebra to the corresponding elements in the Lie group. For elements in $SE(2)$, the exponential map $\exp(\cdot)$ and its inverse map, the logarithmic map $\log(\cdot)$, can be expressed as follows:

$$\begin{aligned} \exp(\cdot) : \mathfrak{se}(2) \to SE(2), &\quad g(p_1, p_2, \theta) = \exp(X), \\ \log(\cdot) : SE(2) \to \mathfrak{se}(2), &\quad X = \log(g(p_1, p_2, \theta)). \end{aligned}$$

For a comprehensive introduction to Lie groups, the interested reader is referred to [2].

3.1. Stochastic Differential Equations

The kinematics of a two-wheeled mobile robot can be characterized by a Stratonovich's stochastic differential equation [3],

$$dx = \begin{bmatrix} \frac{r}{2}(\omega_1 + \omega_2)\cos(\theta) \\ \frac{r}{2}(\omega_1 + \omega_2)\sin(\theta) \\ \frac{r}{l}(\omega_1 - \omega_2) \end{bmatrix} dt + \sqrt{D} \begin{bmatrix} \frac{r}{2}\cos(\theta) & \frac{r}{2}\cos(\theta) \\ \frac{r}{2}\sin(\theta) & \frac{r}{2}\sin(\theta) \\ \frac{r}{l} & -\frac{r}{l} \end{bmatrix} \begin{bmatrix} dW_1 \\ dW_2 \end{bmatrix}. \tag{1}$$

Here, the state of the robot is given by $x = [p_1, p_2, \theta]^{\mathrm{T}} \in \mathbb{R}^3$, $r \in \mathbb{R}$ denotes the wheel radius, $D \in \mathbb{R}$ represents the noise coefficient, and $l \in \mathbb{R}$ is the axis length. The angular velocities of the wheels are denoted by ω_i for $i \in \{1, 2\}$, while dW_i are unit-strength Wiener processes (See Fig 2). The time variable is represented by t.

The stochastic differential equation (1) can be reformulated in a simpler form [3]

$$\begin{aligned} (g^{-1}dg)^{\vee} &= \underbrace{\begin{bmatrix} \frac{r}{2}(\omega_1 + \omega_2) \\ 0 \\ \frac{r}{l}(\omega_1 - \omega_2) \end{bmatrix}}_{h} dt + \sqrt{D} \underbrace{\begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ 0 & 0 \\ \frac{r}{l} & -\frac{r}{l} \end{bmatrix}}_{H} \underbrace{\begin{bmatrix} dW_1 \\ dW_2 \end{bmatrix}}_{dW}, \\ &= h dt + H dW, \end{aligned} \tag{2}$$

which can be sampled using Euler–Maruyama method,

$$g_{(t+dt)} = g_{(t)} \exp\left((h\, dt + H\, dW)^{\wedge}\right). \tag{3}$$

Importantly, the drift term h and the diffusion matrix H in (2) are state-independent so that we can utilize the uncertainty propagation method developed in [1], while the drift term and diffusion matrix in (1) are state-dependent. Also note that the system (2) is *group-affine* [4], whose error dynamics has a nice property.

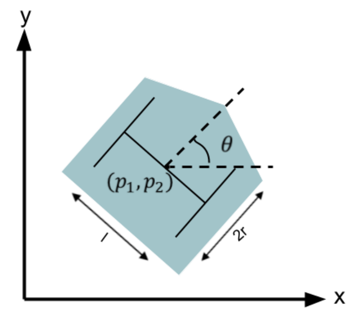

Figure 1. Model of the two-wheeled differential drive robot.

3.2. Mean and Covariance Propagation

The function `mu_Sigma_prediction` computes the mean and covariance propagation in SE(2). When the initial pose of a robot is $g_0 = \mathbf{I}$, if it performs a straight-line motion, i.e. $\omega_1 = \omega_2$, the mean and covariance can be evaluated analytically as (Eq. (42) in [5])

$$\hat{\mu}_k^{-}(t) = \begin{bmatrix} 1 & 0 & r\omega t \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{4}$$

and

$$\hat{\Sigma}_k^{-}(t) = \begin{bmatrix} \frac{1}{2}Dr^2 t & 0 & 0 \\ 0 & \frac{2D\omega^2 r^4 t^3}{3l^2} & \frac{D\omega r^3 t^2}{l^2} \\ 0 & \frac{D\omega r^3 t^2}{l^2} & \frac{2Dr^2 t}{l^2} \end{bmatrix}. \tag{5}$$

If the initial pose is given by $g_0 = a_i$, then the propagated mean is expressed as $a_i \hat{\mu}_k^-(t)$[1].

For general scenarios, where the motion is not restricted to a straight line, the mean and covariance are derived using Eqs. (30) and (31) from [3].

3.3. Measurement

The sensor measurement of the relative pose between the i-th agent and its neighbors, including noise, is modeled by the following expression:

$$m_{ik} = g_i^{-1} g_k \exp(\hat{\eta}_{ik}), \qquad \eta_{ik} \sim \mathcal{N}(\mathbf{0}, \Sigma_m), \tag{6}$$

where g_i and g_k denote the ground-truth poses of the i-th agent and the k-th neighbor, respectively, and η_{ik} is the sensor noise. The function `compute_relative_pose` computes the relative position between the agents and its neighbors.

3.4. Mean and Covariance Update

Now, for the i-th robot, we use relative measurements m_{ik} between itself and its neighboring robots $1, \ldots, k, \ldots, n$ and the measurement model (6) to update its state. The procedure follows Eqs. (36–38) in [5] is introduced in Algorithm (1).

Algorithm 1 Update Procedure

1: **Define** $q_k = m_{ik} (\hat{\mu}_k^-)^{-1} a_k^{-1} a_i \hat{\mu}_i^-$

2: **Compute** $\exp(\hat{x}_k) = q_k$

3: **Define** $\Gamma_k = \left(I + \frac{1}{2}\mathrm{ad}(\hat{x}_k)\right)$

4: **Compute** $S_k = \Gamma_k^{\mathrm{T}} (\mathrm{Ad}(m_{ik}))^{-T} \Sigma_{koik}^{-1} (\mathrm{Ad}(m_{ik}))^{-1} \Gamma_k$

5: **Define** $S_i = \Gamma_i^{\mathrm{T}} (\hat{\Sigma}_k^-)^{-1} \Gamma_i$

6: **Compute the sum** $\bar{S}' = S_i + \sum_{k=1}^n S_k$

7: **Compute** $\bar{x}' = \bar{S}'^{-1} \sum_{k=1}^n S_k (\hat{x}_k)^\vee$

8: **Compute** $\bar{\Gamma}' = I + \sum_{k=1}^n \Gamma_k$

In Algorithm (1), Σ_{ik} denotes the measurement noise covariance, and $\Sigma_{koik} = A_{ik} + B_{ik} + F(A_{ik}, B_{ik})$, with the following definitions:

$$A_{ik} = \mathrm{Ad}(m_{ik}^{-1}) \Sigma_k \mathrm{Ad}(m_{ik}^{-1})^{\mathrm{T}}, \qquad B_{ik} = \Sigma_{ik},$$

$$A''_{ik_{ij}} = \mathrm{ad}(E_i)\mathrm{ad}(E_j) A_{ik_{ij}}, \qquad B''_{ik_{ij}} = \mathrm{ad}(E_i)\mathrm{ad}(E_j) B_{ik\,ij}$$

$$F_i(A_i, B_i) = \frac{1}{4} \sum_{i,j=1}^{d} \mathrm{ad}(E_i) B_i \; \mathrm{ad}(E_j)^{\mathrm{T}} A_{i_{ij}}$$
$$+ \frac{1}{12} \left(\left\{ \sum_{i,j=1}^{d} A''_{i_{ij}} \right\} B_i + B_i^{\mathrm{T}} \left[\sum_{i,j=1}^{d} A''_{i_{ij}} \right]^{\mathrm{T}} \right)$$
$$+ \frac{1}{12} \left\{ \left[\sum_{i,j=1}^{d} B''_{i_{ij}} \right] A_i + A_i^{\mathrm{T}} \left[\sum_{i,j=1}^{d} B''_{i_{ij}} \right]^{\mathrm{T}} \right\}.$$

Subsequently, the mean and covariance of the i-th agent are updated as follows

$$\hat{\Sigma}_k^+ = \bar{\Gamma}' \bar{S}'^{-1} \bar{\Gamma}'^{\mathrm{T}}, \qquad \hat{\mu}_k^+ = a_i \hat{\mu}_i^-(t) \exp(-\hat{\bar{x}}').$$

The operators Ad and ad, along with the basis sets E_i, are defined in [3], [5]. The implementation of this subsection is provided in function `mu_Sigma_update`.

[1] It is important to note that in [5], $\hat{\mu}_k^-(t)$ represents the propagated *relative* mean (4) for the i-th robot, rather than the absolute propagated mean $a_i \hat{\mu}_k^-(t)$.

4. Simulation Result

This example simulates the localization of a robot team moving in a straight line. The true robot motions are simulated 1000 times, and the end positions of each trial are plotted in Fig. 2. The results show that the distribution of this stochastic differential system (SDE) forms a banana-shaped pattern, as discussed in [3].

In this simulation, all three robots are commanded to move in a straight line for 1.5 seconds at a speed of 0.5 m/s. The goal is to estimate the true pose of the middle robot (blue), referred to as robot i, using relative measurements obtained from its neighboring robots (yellow and red). Following the procedure outlined in Subsections (3.2–3.4), the position of robot i is estimated. The simulation results, shown in Fig. 2, indicate a more accurate mean position (black triangle), demonstrating higher confidence in the estimate.

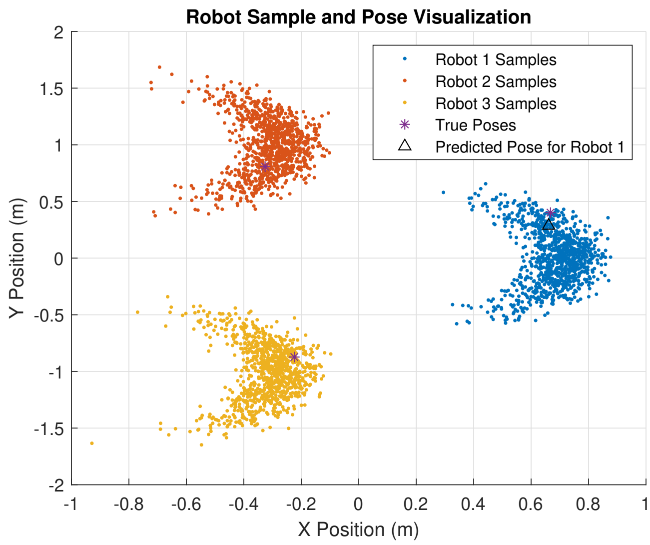

Figure 2. Simulation result.

Should you have any questions, you are welcome to contact me at

✉ maria.khodaverdian@csie.am

References

[1] Y. Wang and G. S. Chirikjian, "Nonparametric second-order theory of error propagation on motion groups", *The International journal of robotics research*, vol. 27, no. 11-12, pp. 1258–1273, 2008.

[2] G. S. Chirikjian, *Stochastic models, information theory, and Lie groups, volume 2: Analytic methods and modern applications*. Springer Science & Business Media, 2011, vol. 2.

[3] A. W. Long, K. C. Wolfe, M. J. Mashner, G. S. Chirikjian, *et al.*, "The banana distribution is gaussian: A localization study with exponential coordinates", *Robotics: Science and Systems VIII*, vol. 265, no. 1, 2013.

[4] A. Barrau and S. Bonnabel, "The invariant extended kalman filter as a stable observer", *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1797–1812, 2016.

[5] X. Li and G. S. Chirikjian, "Lie-theoretic multi-robot localization", in *Riemannian Computing in Computer Vision*, P. K. Turaga and A. Srivastava, Eds. Cham: Springer International Publishing, 2016, pp. 165–186, ISBN: 978-3-319-22957-7. DOI: 10.1007/978-3-319-22957-7_8. [Online]. Available: https://doi.org/10.1007/978-3-319-22957-7_8.