DISTRIBUTED BAYESIAN TRACKING ON THE SPECIAL EUCLIDEAN GROUP USING LIE ALGEBRA PARAMETRIC APPROXIMATIONS

Claudio J. Bordin Jr.[1], **Caio G. de Figueredo**[2], **and Marcelo G. S. Bruno**[3]

[1]claudio.bordin@ufabc.edu.br [2]caio.figueredo@ifce.edu.br [3]bruno@ita.br

[1]Universidade Federal do ABC, Brazil [2]Instituto Federal do Ceará, Brazil [3]Instituto Tecnológico de Aeronáutica, Brazil

1. Introduction

- The joint rotational and translational state of a rigid body can be parameterized as an element of the Special Euclidean Group $SE(3)$.
- Modern engineering systems involve cooperation between multiple agents on a partially connected network to run a common task, e.g., estimate a hidden state.
- In previous works, we introduced diffusion particle filters (PF) to perform cooperative tracking of states that evolved on the Spherical and the Stiefel manifolds and the Special Orthogonal Group.
- Diffusion PFs include a data assimilation step where agents update their beliefs about the unknown states, assimilating local measurements and measurements from neighboring nodes.
- The local updated beliefs are then exchanged between nodes in a compressed form, using Gaussian parametric approximations on the Lie Algebra associated to $SE(n)$.

2. Special Euclidean Group $SE(n)$

- The Special Euclidean Group SE(n) is a matrix Lie group. An element $\boldsymbol{S}$ of SE(n) is given as

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{\Omega} & \boldsymbol{u} \\ \boldsymbol{0}_{1\times n} & 1 \end{bmatrix},$$

  where $\boldsymbol{\Omega}$ is a member of the Special Orthogonal Group SO(n), $\boldsymbol{u} \in \mathbb{R}^n$, and $\boldsymbol{0}_{1\times n}$ denotes a vector with null entries.
- The group SE(n) has dimension $d \triangleq n(n+1)/2$ and, for $n = 3$, it corresponds to the set of all possible translations and rotations of a 3-dimensional rigid object.
- SE(n) is also a differentiable manifold. Thus, we can define a tangent space $\mathcal{T}_{\boldsymbol{S}}$ at each point $\boldsymbol{S} \in$ SE(n).
- The Lie algebra $\mathfrak{se}(n)$ is, by definition, the tangent space to the identity matrix $\boldsymbol{I}$, i.e., $\mathcal{T}_{\boldsymbol{I}}$.
- A matrix $\boldsymbol{S} \in$ SE(n) can be mapped into a matrix $\boldsymbol{X} \in \mathfrak{se}(n)$ using the logarithmic map Log : SE$(n) \rightarrow \mathfrak{se}(n)$.
- For SE(n), Log$(\boldsymbol{S})$, is the usual matrix logarithmic function, denoted $logm$.
- For a matrix $\boldsymbol{S} \in$ SE(n),

$$\text{Log}(\boldsymbol{S}) = \begin{bmatrix} \text{logm}(\boldsymbol{\Omega}) & \boldsymbol{V}\boldsymbol{u} \\ \boldsymbol{0}_{1\times n} & 0 \end{bmatrix},$$

  where $\boldsymbol{V} = \boldsymbol{I} + \sum_{m=1}^{\infty} \boldsymbol{\Omega}^m/(m+1)!$.
- For $n = 3$, $\boldsymbol{Z} = \text{logm}(\boldsymbol{\Omega})$ has the form

$$\boldsymbol{Z} = \begin{bmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{bmatrix}.$$

- Thus, $\boldsymbol{X} = \text{Log}(\boldsymbol{S}) \in \mathfrak{se}(n)$ is isomorphic to a vector $\boldsymbol{x} \in \mathbb{R}^d$, and we can define a bijective mapping $\Phi : \mathfrak{se}(n) \rightarrow \mathbb{R}^d$, such that $\Phi(\boldsymbol{X}) = \left[\boldsymbol{z}^T \;\; (\boldsymbol{V}\,\boldsymbol{u})^T\right]^T \in \mathbb{R}^d$ where $\boldsymbol{z}$ is a vector that collects the free-varying entries of $\boldsymbol{Z}$.

3. Problem Setup

- Let $\boldsymbol{S}_k \in$ SE(n) denote an unknown state at time $k \geq 0$ that evolves according to the random walk

$$\boldsymbol{S}_k = \boldsymbol{S}_{k-1}\,\text{Exp}\left(\Phi^{-1}(\boldsymbol{\epsilon}_k)\right), \; k > 0,$$

  with $p(\boldsymbol{S}_0) \propto 1$, where $\{\boldsymbol{\epsilon}_k\}$ is a sequence of i.i.d.Gaussian random vectors in $\mathbb{R}^d$ with zero mean and covariance matrix $\boldsymbol{\Lambda}_k$.
- The nodes record at each instant k the observations

$$\boldsymbol{Y}_{k,r} = \boldsymbol{\mathcal{H}}_r(\boldsymbol{\Pi}\boldsymbol{S}_k) + \boldsymbol{W}_{k,r}, \quad k > 0,$$

  where $r \in \{1, ..., R\}$ denotes the r-th node in the network, $\boldsymbol{\mathcal{H}}_r(\cdot)$: $\mathbb{R}^{n\times(n+1)} \mapsto \mathbb{R}^{n\times(n+1)}$ is a general function, $\boldsymbol{\Pi} \in \mathbb{R}^{n\times(n+1)}$ is an $(n+1)\times(n+1)$ identity matrix without its bottommost row, and $\{\boldsymbol{W}_{k,r}\}$ is a sequence of i.i.d.samples of a Matrix Gaussian p.d.f. $\mathcal{N}_{n,n+1}(\boldsymbol{W}_{k,r}|\boldsymbol{0}_{n,n+1}, \boldsymbol{\Psi}_r, \boldsymbol{\Gamma}_r)$.
- Given the observations $\{\boldsymbol{Y}_{l,r}\}$, $0 \leq l \leq k$, $0 \leq r \leq R$, our goal is to recursively estimate $\boldsymbol{S}_k$ in a distributed fashion.

4. RndEx Diffusion PF

- The RndEx Algorithm has two steps: Random Exchange and Data Assimilation.
- In the Random Exchange step, a node l exchanges with another randomly chosen node r its posterior p.d.f. $p(\boldsymbol{S}_{k-1}|\tilde{\boldsymbol{Y}}_{1:k-1,l})$, in which $\tilde{\boldsymbol{Y}}_{1:k-1,l}$ denotes all observations assimilated up to instant k.
- Suppose that the posterior p.d.f. is approximated by the weighted particle set $\{w^{(q)}_{k-1,l}, \boldsymbol{S}^{(q)}_{k-1,l}\}$, $q = 1, ..., Q$, $Q \gg 1$.
- Before the exchange, node l compresses the representation using a Gaussian parametric approximation as follows.

1) Given $\{w^{(q)}_{k-1,l}, \boldsymbol{S}^{(q)}_{k-1,l}\}$ compute its centroid $\hat{\boldsymbol{S}}_{k-1,r}$ as

$$\hat{\boldsymbol{S}}^{<i+1>}_{k-1,l} = \hat{\boldsymbol{S}}^{<i>}_{k-1,l} \cdot \text{expm}\left[\sum_{q=1}^{Q} w^{(q)}_{k-1,l}\,\text{logm}\left(\left(\hat{\boldsymbol{S}}^{<i>}_{k-1,l}\right)^{-1} \boldsymbol{S}^{(q)}_{k-1,l}\right)\right],$$

where $\hat{\boldsymbol{S}}^{<i>}_{k-1,l}$ denotes the i-th estimate of the weighted average, with $\hat{\boldsymbol{S}}^{<0>}_{k-1,l}$ chosen as a random element of the particle set.

2) Then, we evaluate $\boxed{\boldsymbol{x}^{(q)}_{k-1,l} = \Phi\left(\text{logm}\left(\hat{\boldsymbol{S}}^{-1}_{k-1,l}\boldsymbol{S}^{(q)}_{k-1,l}\right)\right) \in \mathbb{R}^d.}$

3) The weighted particle set $\{w^{(q)}_{k-1,l}, \boldsymbol{x}^{(q)}_{k-1,l}\}$ is then approximated by a Gaussian p.d.f.via moment matching, with sample moments $\bar{\boldsymbol{x}}_{k-1,l}$ and $\boldsymbol{\Sigma}_{k-1,l}$.

- At the end of the Random Exchange Step, node r receives $\{\bar{\boldsymbol{x}}_{k-1,l}; \boldsymbol{\Sigma}_{k-1,l}; \hat{\boldsymbol{S}}_{k-1,l}\}$ and rebuilds the particle set as

$$\begin{aligned} \tilde{\boldsymbol{x}}^{(q)}_{k-1,l} &\sim \mathcal{N}\left(\boldsymbol{x}_{k-1}|\bar{\boldsymbol{x}}_{k-1,l}; \boldsymbol{\Sigma}_{k-1,l}\right), \\ \tilde{\boldsymbol{S}}^{(q)}_{k-1,l} &= \hat{\boldsymbol{S}}_{k-1,l}\,\text{expm}\left(\Phi^{-1}(\tilde{\boldsymbol{x}}^{(q)}_{k-1,l})\right) \\ \tilde{w}^{(q)}_{k-1,l} &= \frac{1}{Q}, \end{aligned}$$

  where $q \in \{1, ..., Q\}$ and $\mathcal{N}(\boldsymbol{s}|\boldsymbol{\mu}; \boldsymbol{\Theta})$ denotes a (vector) multivariate Gaussian p.d.f.with argument $\boldsymbol{s}$, parameterized by the mean vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Theta}$.
- In the Data Assimilation step, node r samples new particles from the prior importance function

$$\boldsymbol{S}^{(q)}_{k,r} \sim p(\boldsymbol{S}_k|\boldsymbol{S}^{(q)}_{k-1,l}),$$

  and updates the particles' weights as

$$w^{(q)}_{k,r} \propto w^{(q)}_{k-1,l}\left[\prod_{u\in\tilde{N}(r)} p(Y_{k,u}|\boldsymbol{S}^{(q)}_{k,r})\right],$$

  in which $\tilde{N}(r)$ denotes the $r-$th node closed neighborhood.
- The updated set $\{w^{(q)}_{k,r}, \boldsymbol{S}^{(q)}_{k,r}\}$ is then a Monte Carlo representation of the posterior p.d.f. $p(\boldsymbol{S}_k|\,\tilde{\boldsymbol{Y}}_{k,r}, \tilde{\boldsymbol{Y}}_{1:k-1,l}) \triangleq p(\boldsymbol{S}_k|\,\tilde{\boldsymbol{Y}}_{1:k,r})$.
- State estimates can be computed as the centroid of this particle set.

5. Alternative Formulation

- Alternatively, the state components $\boldsymbol{\Omega}^{(q)}_{k-1,l}$ and $\boldsymbol{u}^{(q)}_{k-1,l}$ can be separately compressed:

1) Given $\{w^{(q)}_{k-1,l}, \boldsymbol{S}^{(q)}_{k-1,l}\}$ compute the centroids

$$\hat{\boldsymbol{\Omega}}^{<i+1>}_{k-1,l} = \hat{\boldsymbol{\Omega}}^{<i>}_{k-1,l}\text{expm}\left[\sum_{q=1}^{Q} w^{(q)}_{k-1,l}\,\text{logm}\left(\left(\hat{\boldsymbol{\Omega}}^{<i>}_{k-1,l}\right)^T \boldsymbol{\Omega}^{(q)}_{k-1,l}\right)\right],$$

$$\hat{\boldsymbol{u}}_{k-1,l} = \sum_{q=1}^{Q} w^{(q)}_{k-1,l}\boldsymbol{u}^{(q)}_{k-1,l},$$

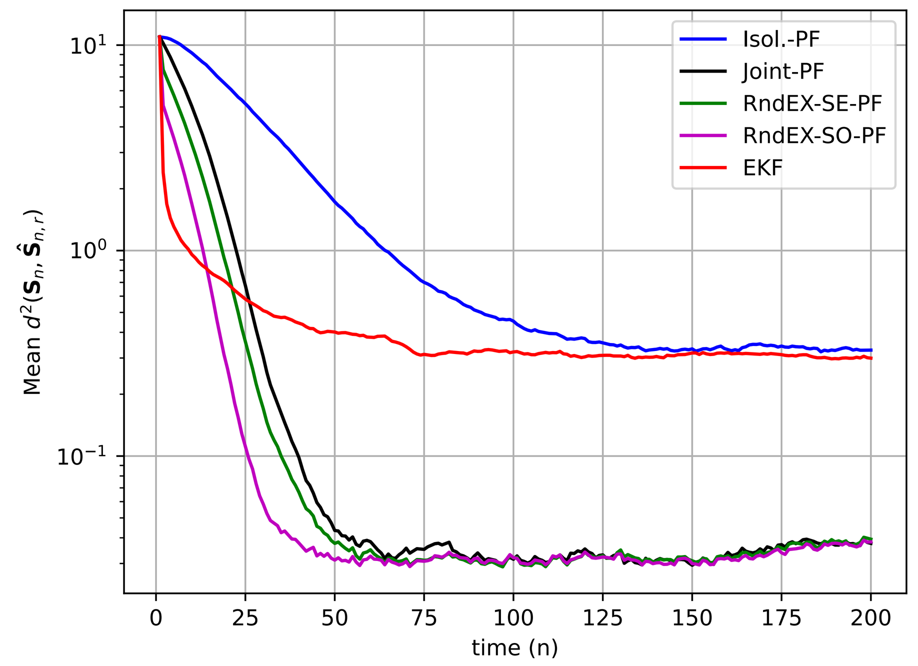
2) Then, evaluate $\boxed{\boldsymbol{x}^{(q)}_{k-1,l} = \Phi\left(\text{logm}\left(\hat{\boldsymbol{\Omega}}^{T}_{k-1,l}\boldsymbol{\Omega}^{(q)}_{k-1,l}\right)\right) \in \mathbb{R}^d.}$

3) The weighted particle set $\{w^{(q)}_{k-1,l}, \boldsymbol{x}^{(q)}_{k-1,l}, \boldsymbol{u}^{(q)}_{k-1,l}\}$ is then approximated by a Gaussian p.d.f.via moment matching.

- At the end of the Random Exchange Step, node r receives $\{\bar{\boldsymbol{x}}_{k-1,l}; \bar{\boldsymbol{u}}_{k-1,l}; \boldsymbol{\Sigma}_{k-1,l}; \hat{\boldsymbol{\Omega}}_{k-1,l}\}$ and rebuilds the particle set as previously.

6. Simulation Results

- We performed a numerical simulation with 300 independent trials.
- We set $n = 3$ and $Q = 300$. In each trial, 200 synthetic data samples were generated.
- We used a network with five nodes: nodes 1 to 4 are on the vertices of a square and node 5 is at its center and is connected to all other nodes.
- The noise covariance matrices were set to $\boldsymbol{\Psi}_r = \boldsymbol{I}$ and $\boldsymbol{\Gamma}_r = \boldsymbol{I} \cdot 10^{-\alpha_r/10}$, with α_r equal to 3, 6, 10, 13 and 20 dB for $r = 1, \ldots, 5$, respectively. The driving noise covariance matrix $\boldsymbol{\Lambda}_k$ was set to $0.05\,\boldsymbol{I}$.
- For comparison, we ran in the same setup three competing algorithms: i) bootstrap PFs operating isolatedly in each node (Isol.-PF), ii) a bootstrap PF with access to all observations (Joint-PF), and ii) a SE(n)-constrained EKF.
- We assumed that $[\boldsymbol{\mathcal{H}}_r(\boldsymbol{X})]_{i,j} = h([\boldsymbol{X}]_{i,j})$, and set $h(x) = \tanh(x/\beta)/\beta$ with $\beta = 1.7$.
- The algorithm's performance was evaluated in terms of the squared geodesic distance in SE(n), i.e., $d^2(\boldsymbol{S}_k, \hat{\boldsymbol{S}}_k) = \|\text{logm}(\boldsymbol{\Omega}_k^T\hat{\boldsymbol{\Omega}}_k)\|_F^2 + \|\boldsymbol{u}_k - \hat{\boldsymbol{u}}_k\|^2$.



7. Conclusions

- This paper described two novel RndEx particle filters for tracking the state of a dynamic system that evolves on the Special Euclidean Group.
- The first algorithm exchanges Gaussian parametric approximations built on a vector space isomorphic to $\mathfrak{se}(n)$.
- The second algorithm directly computes approximations for the translational information $\boldsymbol{u}^{(q)}_{k-1,l}$, performing computations on the Lie algebra of $\mathfrak{so}(n)$ only for rotation matrices $\boldsymbol{\Omega}^{(q)}_{k-1,l}$.
- Experimental results show that the proposed methods perform similarly to a centralized PF-based estimator, greatly outperforming PFs operating in isolation and an extended Kalman filter, at increased computational cost.

IEEE ICASSP 2023, Rhodes Island, Greece.