Metropolis-Adjusted Langevin Sequential Markov Chain Monte Carlo methods on Matrix Lie Groups

Enzo LOPEZ*†, Karim DAHIA*, Nicolas MERLINGE*,
Benedicte WINTER-BONNET†, Alain MASCHIELLA† and Christian MUSSO*
*DTIS, ONERA, Université Paris-Saclay, 91120, Palaiseau, France
(enzo.lopez / karim.dahia / nicolas.merlinge / christian.musso)@onera.fr
†MBDA, Plessis-Robinson, France
(enzo.lopez / benedicte.winter-bonnet / alain.maschiella)@mbda-systems.com

***Abstract*—Particle filters on Lie groups enable nonlinear filtering by randomly generating particles that preserve the structure of rotation matrices. Despite their potential, they suffer from high computational costs, weight degeneration, and dimensionality issues. Sequential Markov Chain Monte Carlo (SMCMC) methods mitigate these challenges by iteratively refining state estimates. Among them, the Metropolis-Adjusted Langevin Algorithm (MALA) leverages gradient information to improve sampling efficiency. This paper extends MALA to matrix Lie groups, introducing the LG-MALA algorithm, which reduces computational demands while maintaining accuracy. Its effectiveness is demonstrated in a challenging UAV navigation scenario based on data fusion between inertial measurements and landmark-based angular measurements.**

I. Introduction

State estimation for nonlinear systems in robotics and aerospace often involves high-dimensional and non-Gaussian noise distributions. Traditional particle filters, though widely used, face limitations in such scenarios, mainly due to particle impoverishment and high computational costs. Sequential Markov Chain Monte Carlo (SMCMC) methods have emerged as alternatives [1] [2], offering improved sampling efficiency through Markov chain transitions.

Recent research works showed that SMCMC on manifolds outperform their Euclidean counterparts in terms of robustness, accuracy, and computational efficiency [3] [4]. Among these, filters on Lie groups stand out as a particularly effective algebraic framework, especially when dealing with rotational motion. An example includes the Lie Group Sequential Markov Chain Monte Carlo filter (LG-SMCMC) [3], which introduced an alternative to non-linear filtering.

Langevin dynamics provides a framework for modeling the behavior of systems contingent on both deterministic and stochastic forces. It plays a significant role in the field of estimation, particularly in the context of sampling algorithms. One key application is the Metropolis-Adjusted Langevin Algorithm (MALA) [5], which combines Langevin dynamics with the Metropolis-Hastings algorithm to efficiently sample from complex posterior distributions in Bayesian inference problems.

This paper extends the framework of SMCMC to the Lie Group Metropolis-Adjusted Langevin Algorithm (LG-MALA). By leveraging Langevin dynamics guided by gradient information, the LG-MALA filter refines the exploration of posterior distributions. Incorporating a preconditioning matrix based on Fisher information on Lie groups ensures efficient sampling by adapting step sizes to local geometrical properties. We evaluate the LG-MALA filter through simulations in a UAV navigation scenario, demonstrating its ability to address the limitations of traditional particle filters.

The paper is structured as follows: Section II presents the estimation problem and introduces the paper key concepts. Section III outlines the framework for applying Langevin diffusion to Lie groups, which constitutes the main contribution of this work. In Section IV, a simulated UAV navigation problem is presented, demonstrating the effectiveness of the proposed method. Finally, Section V wraps up the paper with concluding remarks.

II. Problem Statement

A. Introduction to Lie groups

1) Basic Concepts: A Lie group $(\mathcal{G}, \cdot)$ refers to a manifold that is endowed with a group structure [6]. This group is characterized by a tangent space at the identity element, commonly referred to as the *Lie algebra*, and denoted as $\mathfrak{g}$. Near the identity element I, there exists a local bijection between the Lie group and the Lie algebra, represented by the *group exponential map* $\exp_{\mathcal{G}} : \mathfrak{g} \to \mathcal{G}$ and the *group logarithm map* $\log_{\mathcal{G}} : \mathcal{G} \to \mathfrak{g}$. In the case of matrix Lie groups, these maps can be expressed using matrix series [6]:

$$\exp_{\mathcal{G}}(X) = \sum_{k=0}^{\infty} \frac{X^k}{k!} \ ; \ \log_{\mathcal{G}}(X) = \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k}(X - I)^k. \tag{1}$$

For a Lie algebra $\mathfrak{g}$ of dimension d, we define two isomorphisms with Euclidean space as follows:

$$[\cdot]^\wedge : \mathbb{R}^d \to \mathfrak{g}, \text{ and } \ [\cdot]^\vee : \mathfrak{g} \to \mathbb{R}^d, \tag{2}$$

where their compositions with $\exp_{\mathcal{G}}$ and $\log_{\mathcal{G}}$ are represented by:

$$\exp_{\mathcal{G}}([\cdot]^\wedge) = \exp_{\mathcal{G}}^\wedge(\cdot) \ ; \ [\log_{\mathcal{G}}(\cdot)]^\vee = \log_{\mathcal{G}}^\vee(\cdot). \tag{3}$$

A summary of these transformations is shown in Figure 1. It should be noted that the bijection between $\mathcal{G}$ and $\mathbb{R}^d$

theoretically holds in the neighborhood of the identity element I_d.

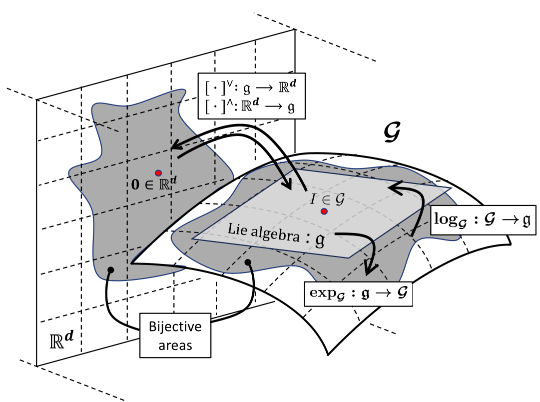

Fig. 1: Graphical illustration of the Lie group structure [7].

2) Group Errors: Consider two state matrices, denoted as $(X, \widehat{X})$, belonging to the group $(\mathcal{G}, \cdot)$. Since the group operation is non-commutative, the group error between X and $\widehat{X}$ can be expressed in two forms: on the *right* as $\eta^R = \widehat{X}X^{-1}$, and on the *left* as $\eta^L = X^{-1}\widehat{X}$. In what follows, only the left form is considered. Additionally, the representation of the group error η in Euclidean space, denoted as ξ, is given by:

$$\xi = \log_\mathcal{G}^\vee(\eta). \tag{4}$$

This formulation is a key element in the performance of Lie group filters, since it maps a matrix error to a vector error. This is of practical interest to maintain accuracy over a wide range of values.

3) Probability Distributions on Lie Groups: Let $X \sim \mathcal{N}_\mathcal{G}(\mu, P)$ represent a random matrix that follows a concentrated Gaussian distribution on $\mathcal{G}$ with mean μ and covariance matrix P. Its probability density function is approximated by [7] [8]:

$$p(X) \approx \frac{1}{\sqrt{(2\pi)^d \det[P]}} e^{-\frac{1}{2}||\xi||_P^2}, \tag{5}$$

where $\xi = \log_\mathcal{G}^\vee(\mu^{-1}X)$ in the left case and $||\cdot||_P$ denotes the Mahalanobis norm with respect to P. This approximation holds when ξ remains within the bijective domain of the group exponential. [8].

B. Sequential Markov Chain Monte Carlo on Lie Groups

Let $\{X_k\}_{k\in\mathbb{N}^*} \in \mathcal{G}$ denote a sequence of discrete-time state matrices and $\{Y_k\}_{k\in\mathbb{N}^*} \in \mathcal{H}$ the set of observations. The system dynamics are described by:

$$\begin{cases} X_k = f(X_{k-1}, n_{k-1}^q), \\ Y_k = h(X_k, n_k^r), \end{cases} \tag{6}$$

where (n_{k-1}^q, n_k^r) are noise terms and (f, h) are non-linear functions that respectively represent the dynamics and observation models. SMCMC methods are based on Monte Carlo simulations and rely on Bayesian theory to estimate the posterior density of the state X_k given the measurements $Y_{1:k}$. The Bayes rule states that:

$$p(X_k|Y_{1:k}) \propto p(Y_k|X_k)p(X_k|Y_{1:k-1}). \tag{7}$$

This equation updates our beliefs about the state based on new measurements. It is shown that the posterior density can be written as a weighted sum of Dirac [9]:

$$p(X_k|Y_{1:k}) \approx \sum_{i=N_b+1}^{N_b+N_p} w_k^i \delta_{X_k^i}(X_k), \tag{8}$$

where N_p is the number of particles i and N_b the burn-in parameter. The weight is computed based on the posterior and the chosen proposal densities explained subsection II-C:

$$w_k^i \propto \frac{p(X_k^i|Y_{1:k})}{q(X_k^i)}. \tag{9}$$

The Metropolis-Hastings (MH) algorithm then generates new samples $\tilde{X}_k$ from the proposal density q and accepts or rejects them based on a defined acceptance probability:

$$A(\tilde{X}_k, X_k^{j-1}) = \min\left(1, \frac{p(\tilde{X}_k|Y_{1:k})}{p(X_k^{j-1}|Y_{1:k})}\frac{q(X_k^{j-1}|\tilde{X}_k)}{q(\tilde{X}_k|X_k^{j-1})}\right), \tag{10}$$

where X_k^{j-1} is the current accepted particle and $q(\tilde{X}_k|X_k^{j-1})$ is the proposal density. The Markov Chain method and the sampling process at step k are detailed in [3].

The LG-SMCMC follows a process similar to that of the traditional SMCMC, but ensures that each particle resides in the Lie group manifold [3]. Initially, particles are sampled from a centered Gaussian distribution in the Euclidean space:

$$\xi^{1:N_p} \sim \mathcal{N}_\mathcal{G}(0, P_0). \tag{11}$$

Each particle is then mapped to the group using the exponential map and adjusted by a left mean μ:

$$X_0^i = \mu \exp_\mathcal{G}^\wedge(\xi^i). \tag{12}$$

Next, each particle propagates according to the system dynamics:

$$X_k^i = f(X_{k-1}^i, n_{k-1}^{q^i}), \tag{13}$$

and the weighting step shown in equation (9) is performed. The state estimate is then computed as a normalized weighted mean of the particles, with the left case mean given by [10]:

$$\hat{X}_k = \underset{\mu_k}{\operatorname{argmin}} \sum_{i=1}^{N_p} w_k^i \log_\mathcal{G}^\vee(\mu_k^{-1}X_k^i). \tag{14}$$

The covariance matrix is then updated using:

$$\hat{P}_k = \sum_{i=1}^{N_p} w_k^i \log_\mathcal{G}^\vee(\hat{X}_k^{-1}X_k^i) \log_\mathcal{G}^\vee(\hat{X}_k^{-1}X_k^i)^\top. \tag{15}$$

C. Metropolis-Adjusted Langevin Algorithm

Langevin dynamics (or Langevin diffusion) is a method used to describe the evolution of a system under the influence of both deterministic forces and random fluctuations. It is often used in statistical mechanics, physics, and machine learning to simulate the motion of particles in a fluid or gas [11].

The Metropolis-Adjusted Langevin Algorithm (MALA) [12] leverages gradients of the log-likelihood function to guide particle updates. Let $\{\mathrm{x}_k\}_{k\in\mathbb{N}^*} \in \mathbb{R}^d$ denote the state vector and $\{\mathrm{y}_k\}_{k\in\mathbb{N}^*} \in \mathbb{R}^m$ the measurement vector in euclidean space, the proposal density $q(\tilde{\mathrm{x}}_k|\mathrm{x}_k^{j-1})$ of the acceptance probability shown equation (10) is taken as:

$$q(\tilde{\mathrm{x}}_k|\mathrm{x}_k^{j-1}) = \mathcal{N}\left(\tilde{\mathrm{x}}_k \middle| \mathrm{x}_k^{j-1} + \frac{\epsilon^2}{2}\Sigma_k \nabla \log p(\mathrm{x}_k^{j-1}|\mathrm{y}_{1:k}), \epsilon^2\Sigma_k\right). \quad (16)$$

The proposal step can then be expressed as:

$$\tilde{\mathrm{x}}_k = \mathrm{x}_k^{j-1} + \frac{\epsilon^2}{2}\Sigma_k \nabla \log p(\mathrm{x}_k^{j-1}|\mathrm{y}_{1:k}) + \epsilon\Sigma_k^{\frac{1}{2}}\eta_k^{j-1}, \quad (17)$$

where:

- ϵ is the step size,
- Σ_k is the preconditioning matrix step k,
- $\eta_k \sim \mathcal{N}(0, I)$ is a standard Gaussian noise vector.

Accepted particles are then used to calculate the weighted mean for state estimation.

III. Sequential Metropolis-Adjusted Langevin Algorithm on Lie Groups

The Sequential Metropolis-adjusted Langevin Algorithm on Lie groups (LG-MALA) detailed in Algorithm 1 iteratively updates system states using sequential gradient-oriented sampling. By deriving analytical formulas specifically tailored to Lie groups, the method achieves new benchmarks in highly non-linear model estimation, such as the attitude-dependent dynamics of the UAV shown in Section IV. To this end, Section A introduces the LG-MALA scheme and its pseudo-code, while Section B details the Fisher-based Langevin preconditioning formulation.

A. MALA formulation on Lie Groups

To ensure that the diffusion remains as faithful as possible to the model, we compute the proposal in the Lie algebra using equation (17) and then map it back to the group using equation (3). The log-posterior distribution is given by:

$$\log p(X_k|Y_{1:k}) = C - \frac{1}{2}\left|\left|\log_{\mathcal{G}}^{\vee}\left(\hat{X}_k^{-1}X_k\right)\right|\right|_{\hat{P}_k}^2 - \frac{1}{2}\left|\left|\log_{\mathcal{G}}^{\vee}\left(h(X_k)^{-1}Y_k\right)\right|\right|_{R_k}^2, \quad (18)$$

where $\nu_k = \log_{\mathcal{G}}^{\vee}\left(h(X_k)^{-1}Y_k\right)$ is the left innovation, R_k is the measurement noise covariance matrix and C is a constant term. From [13] and [7], we can compute the gradient of the log-posterior:

$$\nabla \log p(X_k|Y_{1:k}) = -\varphi_{\mathcal{G}}^T(\xi_k)\,\hat{P}_k^{-1}\xi_k - \varphi_{\mathcal{G}}^T(\nu_k)\,R_k^{-1}\nu_k, \quad (19)$$

where $\varphi_{\mathcal{G}}(.) = \mathrm{ad}_G(.)/\left(e^{\mathrm{ad}_G(.)} - 1\right)$ is the inverse left Jacobian matrix on $\mathcal{G}$ and $\mathrm{ad}_{\mathcal{G}}$ is the adjoint action of the group. The intricacies of this calculus are explained in [7].
The proposed particle, sampled in the Lie algebra, is mapped to the group via the exponential map $\exp_{\mathcal{G}}^{\wedge}(\cdot)$, yielding a Langevin-diffused proposal on Lie groups and evaluated through the Metropolis-Hastings (MH) algorithm. The main contribution over LG-SMCMC [3] is the replacement of a naive MH step with a preconditioned MALA scheme. The next section details the choice of preconditioning matrix Σ_k.

B. Preconditioning with Inverse Fisher Information Matrix

Preconditioning is a crucial enhancement in the MALA to improve sampling efficiency. By incorporating a preconditioning matrix Σ_k, the algorithm adapts the proposal step size to the local geometry of the target distribution [14].

The inverse Fisher information matrix J_k^{-1} provides a more geometrically informed preconditioning. The Fisher information matrix J_k is defined as:

$$J_k = -\mathbb{E}\left[\frac{\partial^2 \log_{\mathcal{G}}^{\vee} p(Y_k|X_k)}{\partial X_k \partial X_k^{\top}}\right]. \quad (20)$$

The inverse of J_k is homogeneous to a covariance matrix and is referred to as the Posterior Cramér-Rao Lower Bound (PCRB). The PCRB adjusts the proposal based on the curvature of the posterior, making it particularly effective in regions of strong anisotropy. In Lie groups the PCRB can be computed using the LG-PCRB scheme [15].

IV. Application to Navigation

We apply the Monte Carlo method from Section III to UAV navigation based on data fusion between an Inertial Measurement Unit (IMU) and the measurements of landmarks Angles of Arrival (AOA). The UAV proceeds to a 75.1 s flight ($\delta t = 0.1$ s) at 160 $\mathrm{m.s}^{-1}$ for which the trajectory is shown Figure 2.

A. Dynamical model

The Earth frame $[e]$ is fixed, while the body frame $[b]$ moves with the UAV. The goal is to estimate state matrix X_k on Lie groups:

$$X_k = \begin{bmatrix} C_b^e & x^e & v^e \\ 0_{2,3} & & I_2 \end{bmatrix}, \quad (21)$$

comprised of position x^e, velocity v^e, and attitude $\theta^e = \log_{\mathcal{G}}^{\vee}(C_b^e)$, using discretized second-order kinematic equations:

$$\begin{cases} C_{b,k+1}^e = C_{b,k}^e \exp_{\mathrm{SO}(3)}^{\wedge}\left(\delta\mathrm{t}\ \omega_{eb}^b + n_{k+1}^{q^{rot}}\right), \\ v_{k+1}^e = v_k^e + \delta\mathrm{t}\left(C_{b,k}^e a_{eb}^b + g^e\right) + n_{k+1}^{q^{vel}}, \\ x_{k+1}^e = x_k^e + \delta\mathrm{t}\ v_k^e + 0.5\ \delta\mathrm{t}^2\left(C_{b,k}^e a_{eb}^b + g^e\right) + n_{k+1}^{q^{pos}}, \end{cases} \quad (22)$$

where $(a_{eb}^b, \omega_{eb}^b)$ are IMU-measured acceleration and rotation rates, g^e is gravity, and n_k^q represents Gaussian noise. IMU characteristics are listed in Table I.

Algorithm 1: (Left) Metropolis-Adjusted Langevin Sequential Markov Chain Monte Carlo on Lie Groups

Initialization

- Initial sample: $\xi^{1:N_p} \sim \mathcal{N}_{\mathbb{E}^d}(0, P_0)$
- Lie group sampling: $X_0^{1:N_p} = \mu_0 \exp_{\mathcal{G}}^{\wedge}(\xi^{1:N_p})$

Time loop: for $k = 2 : T$

1) **Initialize Markov Chain**
   - Propagate: $X_k^{1:N_p} = f(X_{k-1}^{1:N_p}, n_{k-1}^{q^{1:N_p}})$
   - Initialize: $\overline{X}_k^1 = \hat{X}_{k|k-1}$, $\overline{\omega}_k^1 = \frac{1}{N_p}$
2) **Chain loop:** for $j = 2 : (N_b + N_p)$
   - Generate a random variable: $I \sim \mathcal{U}(1, N_p)$
   - Choose particle: $X_k^j = X_k^I$
   - Compute preconditioning matrix: $\Sigma_k^j = J(X_k^j)^{-1}$
   - Compute proposal on Lie algebra from (17),(19)
   - Map it to the group with the exponential map (3)
   - Compute weight $\tilde{\omega}_k$ from equation (9)
   - Generate a random variable: $u \sim \mathcal{U}(0, 1)$
   - If $u \leq A(\tilde{X}_k, \overline{X}_k^{j-1})$ from equation (10)
     - Accept the proposal: $\overline{X}_k^j = \tilde{X}_k$, $\overline{\omega}_k^j = \tilde{\omega}_k$
     - Otherwise, reject it: $\overline{X}_k^j = \overline{X}_k^{j-1}$, $\overline{\omega}_k^j = \overline{\omega}_k^{j-1}$

   End Chain
3) **State estimation**
   - Retain last N_p particles: $X_k^{1:Np} = \overline{X}_k^{N_b+1:N_b+N_p}$
   - Normalize weights: $\omega_k^{1:Np} \propto \overline{\omega}_k^{N_b+1:N_b+N_p}$
   - The mean is given by $\hat{X}_k$ from equation (14)
   - The matrix covariance $\hat{P}_k$ is given equation (15)

End loop

B. Measurement model

The UAV observes thirteen known beacons at fixed positions $\{p_1^e, ..., p_{13}^e\}$, measuring angles of arrival (AOA) (Figure 2):

$$Y_n = \begin{bmatrix} \arctan 2\left(\Delta_{n,y}^b, \Delta_{n,x}^b\right) \\ \arctan 2\left(\Delta_{n,z}^b, \sqrt{(\Delta_{n,x}^b)^2 + (\Delta_{n,y}^b)^2}\right) \end{bmatrix} + n^r, \quad (23)$$

where $\Delta_n^b = C_e^b(p_n^e - x^e)$ is the relative distance in frame $[b]$, and n^r is a Gaussian noise which values are given in Table I. The function $\arctan 2(y, x)$ ensures correct angle computation.

Sensor (IMU)	Noise (1σ)	Rate
Accelerometer	10^{-4} m.s^{-2}	10 Hz
Gyrometer	10^{-5} rad.s^{-1}	10 Hz
Sensor (AOA)	**Noise (1σ)**	**Rate**
AOA	20 mrad	10 Hz

TABLE I: Simulated sensor parameters.

C. Filter Implementation and Comparison

We implement two filters in the (Left) Lie group framework: the LG-SMCMC [3] and the LG-MALA. Their performance is compared using the Root Mean Square Error (RMSE), which measures estimation accuracy [16].

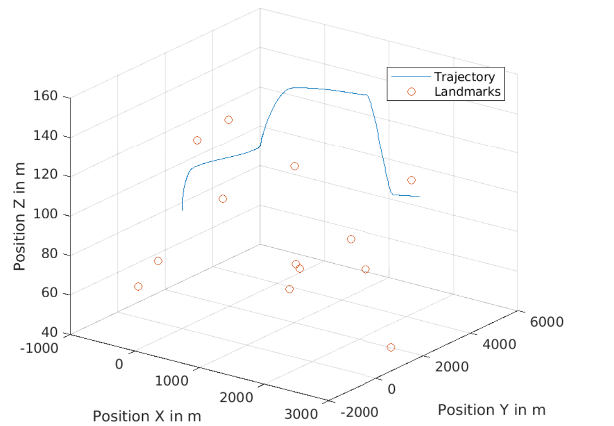

Fig. 2: Simulated trajectory. Landmarks represent environmental features observed by the UAV.

D. Simulations

The measurements and process noise tuning of the filters are outlined in Table III and the initial dispersion of the particles in Table II.

Initial uncertainties	Standard deviation (1σ)
Position	300 m
Velocity	5 m.s^{-1}
Attitude angles	100 mrad

TABLE II: Initial particle dispersion of the filters, denoted as $n_{\text{initial}} \sim \mathcal{N}(0, \sigma)$.

Process noise	Standard deviation (1σ)
Position	5 m
Velocity	0.5 m.s^{-1}
Attitude angles	10 mrad
Measurement noise	**Standard deviation (1σ)**
AOA	20 mrad

TABLE III: Standard deviations of the filters, consistent across all filter types.

Monte Carlo simulations were conducted using empirically chosen parameters: $N_p = 1000$, $N_b = 100$ and $\epsilon = 0.2$. Over the course of 100 iterations, the performance of non-Euclidean and Euclidean methods was compared. Only convergent results are considered in order to exclude statistical outliers. A simulation is classified as non-convergent if, during the final five measurement steps, the state estimate lies out of the confidence ellipsoid defined by the covariance matrix. Table IV presents the proportion of successful runs and the convergence time when the position RMSE reaches 5 m. The RMSE values of the average errors on position, velocity, and the Yaw angle are detailed in Table V. To enhance readability, the table provides the norm instead of their individual components.

As shown in Table V, MALA variants consistently yield lower average errors in position, velocity and attitude by utilizing gradient information. They demonstrate good

Filters	Convergence rates	Convergence time Position RMSE at 5m
SMCMC	98%	30.8 s
MALA	99%	28.5 s
(L)LG-SMCMC	99%	30.8 s
(L)LG-MALA	99%	4.5 s

TABLE IV: Convergence rates and time of tested filters over 100 MC.

Filters	SMCMC	MALA	(L)LG-SMCMC	(L)LG-MALA
Position [m]	10.4	9.13	9.10	7.87
Velocity [m/s]	4.21	3.74	3.99	3.83
Yaw [mrad]	0.48	0.47	0.47	0.46

TABLE V: Average root mean square error norm of the position and velocity, as well as the Yaw angle, for convergent runs.

convergence rates (see Table IV) and higher convergence speeds (see Figure 3). Furthermore, when endowed within the Lie groups structure, the estimation is further improved with a significantly higher convergence speed in position. Thus, (L)LG-MALA emerges as a compelling alternative for highly non-linear state estimation, retaining the advantages of Lie group filters, such as enhanced angle error management. The results highlight its strong potential for industrial applications.

V. Conclusion

This paper introduces a novel formulation of the Sequential Metropolis-Adjusted Langevin filter on Lie groups (LG-MALA). The proposed approach improves accuracy and robustness compared to classic Euclidean methods, particularly in scenarios with challenging noise discrepancies. Future work will explore its application to more complex problems with higher-dimensional state spaces, further leveraging the advantages of gradient-based SMCMC filters on Lie groups.

Acknowledgment

The authors would like to express their gratefulness to the Innovation and Defense Agency (Agence Innovation Defense) of the French Ministry of Defense for their support in this research endeavor.

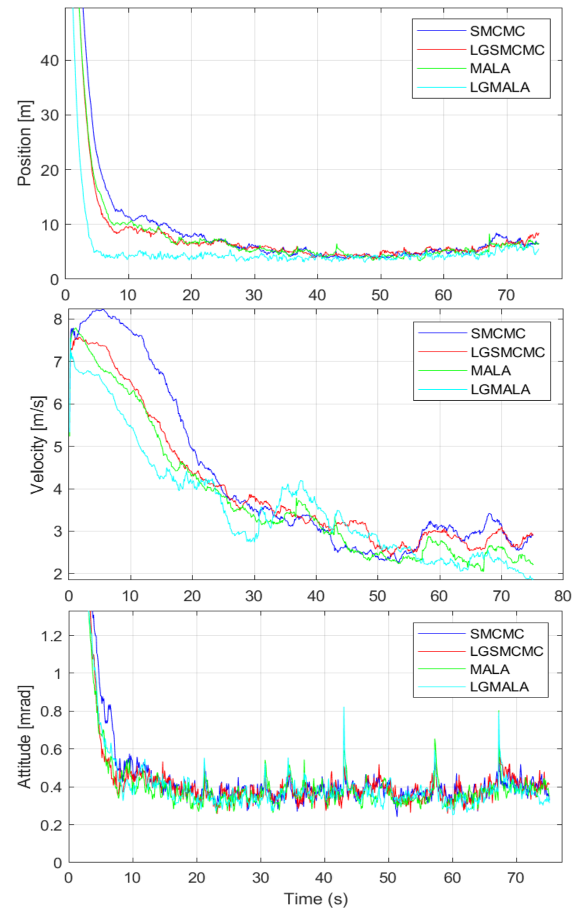

Fig. 3: RMSE of several filters for 100 Monte Carlo runs.

References

[1] A. Endo, E. van Leeuwen, and M. Baguelin, “Introduction to particle markov-chain monte carlo for disease dynamics modellers,” *Epidemics*, vol. 29, p. 100363, 2019.

[2] Y. Li and M. Coates, “Sequential mcmc with invertible particle flow,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE Press, 2017, p. 3844–3848.

[3] E. Lopez, K. Dahia, N. Merlinge, B. Winter-Bonnet, A. Maschiella, and C. Musso, “Sequential markov chain monte carlo methods on matrix lie groups,” in *2024 27th International Conference on Information Fusion (FUSION)*, 2024, pp. 1–7.

[4] M. Girolami and B. Calderhead, “Riemann manifold langevin and hamiltonian monte carlo methods,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 73, no. 2, pp. 123–214, 2011.

[5] F. Septier and G. W. Peters, “Langevin and hamiltonian based sequential mcmc for efficient bayesian filtering in high-dimensional spaces,” *IEEE Journal of Selected Topics in Signal Processing*, 2016.

[6] J. Hilgert and K.-H. Neeb, *Structure and Geometry of Lie Groups*, Springer, Ed. Springer, 2011.

[7] G. Bourmaud, “Estimation de paramètres évoluant sur des groupes de Lie : application à la cartographie et à la localisation d’une caméra monoculaire,” Ph.D. dissertation, Université de Bordeaux, 2015.

[8] G. Chirikjian and M. Kobilarov, “Gaussian approximation of non-linear measurement models on Lie groups,” *53rd IEEE Conference on Decision and Control*, pp. 6401–6406, 2014.

[9] C. Andrieu, A. Doucet, and R. Holenstein, “Particle markov chain monte carlo methods,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, p. 269–342, 2010.

[10] N. Miolane, “Defining a mean on lie group,” *Medical Imaging - https://inria.hal.science/hal-00938320*, 2013.

[11] Y. Demirel and V. Gerbaud, “Chapter 15 - probabilistic approach in thermodynamics,” in *Nonequilibrium Thermodynamics (Fourth Edition)*, fourth edition ed., Y. Demirel and V. Gerbaud, Eds. Elsevier, 2019, pp. 711–791.

[12] M. Mamajiwala, D. Roy, and S. Guillas, “Geometrically adapted langevin dynamics for markov chain monte carlo simulations,” 2022.

[13] K. B. Petersen and M. S. Pedersen, “The Matrix Cookbook,” *http://matrixcookbook.com*, vol. November 15, pp. 8–14, 2012.

[14] S. Livingstone and M. Girolami, “Information-geometric markov chain monte carlo methods using diffusions,” *Entropy*, vol. 16, no. 6, pp. 3074–3102, jun 2014.

[15] C. Chahbazian, K. Dahia, N. Merlinge, B. Winter-Bonnet, A. Blanc, and C. Musso, “Recursive posterior cramér–rao lower bound on lie groups,” *Automatica*, vol. 160, p. 111422, 2024.

[16] Z. Vujovic, “Classification model evaluation metrics,” *International Journal of Advanced Computer Science and Applications*, vol. Volume 12, pp. 599–606, 07 2021.