Laplace Particle Filter on Lie Groups Applied to Angles-Only Navigation

Clément Chahbazian, Nicolas Merlinge, Karim Dahia, Bénédicte Winter-Bonnet, Julien Marini, Christian Musso

Laplace Particle Filter on Lie Groups Applied to Angles-Only Navigation

Clément CHAHBAZIAN *†, Nicolas MERLINGE*, Karim DAHIA*,
Bénédicte WINTER-BONNET†, Julien MARINI†, Christian MUSSO*
*ONERA, The French Aerospace Lab, Palaiseau, France
{clement.chahbazian ; nicolas.merlinge ; karim.dahia ; christian.musso}@onera.fr
†MBDA France, Le Plessis-Robinson, France
{benedicte.winter-bonnet ; julien.marini}@mbda-systems.com

***Abstract*—Particle Filters are an active field of study to address highly nonlinear estimation problems, especially when coupled with optimisation methods (e.g. with the Laplace Particle Filter). Moreover, recent studies revealed the advantage of Kalman filtering on Lie groups. This paper introduces an improved version of the Laplace Particle Filter by deriving a formulation on Lie Groups. The core idea is to harness probability density functions on Lie groups to cope with the state intrinsic geometric constraints and nonlinear nature. In addition, the Lie Group Laplace Particle Filter (LG-LPF) leverages an optimisation algorithm fully defined on Lie Groups. Numerical results on a nonlinear and angles-only navigation scenario demonstrate a significant enhancement in terms of estimation accuracy and robustness.**

I. Introduction

Nonlinear state estimation is a pervasive problem when seeking to describe the evolution of a stochastic system based on a sequence of random measurements. Particle filters represent a popular approach when robustness to nonlinearities and non-Gaussian distributions is a priority in the algorithm's requirements. They solve the optimal filter problem in its most general formulation which is described by the Chapman-Kolmogorov equation and the Bayes rule.

The principle of particle filters is to sample the state probability density function by a mixture of weighted Dirac distributions centered on each sample, referred to as a particle. At each time step, the particles are propagated according to a stochastic dynamical model of the system. When an observation is available, each particle's weight is updated according to the likelihood of the measurement value. After a few iterations, the total particles weight tends to concentrate on a small number of particles, leading to a poor representation of the state density function. This problem, referred to as the degeneracy phenomenon, is solved by resampling new particles in the neighborhood of the most probable former particles. The particle filter framework paved the way to improved resampling methods [1] and alternative paradigms [2]. In addition, the Laplace particle filter [1] introduces an enhanced resampling step which improves the filter's robustness and accuracy in the challenging case where the likelihood and the prior density have little overlap or are too narrow.

On the other hand, new filtering methods on Lie groups [3], [4] demonstrate improved results compared to usual extended Kalman filters. First, the Lie groups framework allows characterizing a specific class of nonlinear systems showing linear-like properties [5], [6], leading to convergence guarantees for extended Kalman filters. In addition, extended and unscented Kalman filters on Lie groups show improved accuracy when compared to other filters defined on the Euclidean space [3], [7], [8]. These improvements come from algebraic and geometric properties which are specific to Lie groups.

To that extent, combining the Lie group framework with particle filters appears to be a fruitful approach. Some research has already proceeded in this way. In [9] the authors leverage the properties of a restricted class of systems on Lie groups to derive a Rao-Blackwellized particle filter with low computational cost. Besides, [10] shows that embedding the full state estimation in a Lie group particle filter enables to lower the number of particles to achieve similar state output errors compared to traditional particle filters. Nonetheless, both of these filters rely on classic multinomial resampling methods which are not optimal in terms of density representation [1].

The contribution of this paper is to propose a clear and simple framework for the implementation of a Laplace particle filter in which the states fully belong to a finite-dimensional Lie group. Using the results of [1], the paper describes a new resampling method on Lie groups based on the Maximum a Posteriori (MAP) computed from an iterative optimization process. The latter approach is compared to two Euclidean particle filters on a navigation scenario with fixed points angles of arrival as aiding measurements.

After introducing key mathematical notions about Lie groups Section (II), the paper describes the proposed particle filter on Lie groups Section (III). Finally, the filters' results are compared through numerical key indicators Section (IV).

II. Problem statement

This section describes the theoretical framework that underpins the Laplace Particle Filter on Lie Groups (LG-LPF) by first stating the stochastic filtering scheme. Then,

the (Euclidean) Particle Filter is introduced alongside the (Euclidean) Laplace Particle Filter principle. Finally, Lie groups theory is summarized with specific emphasis on its probabilistic framework.

A. State Estimation Problem and Laplace Particle Filter

1) Stochastic Filtering Scheme: Let the discrete-time state process describing the evolution of a sequence of hidden states $\{x_k\}_{k\in\mathbb{N}} \in \mathbb{R}^d$, according to a set of observations $\{y_k\}_{k\in\mathbb{N}} \in \mathbb{R}^m$:

$$\begin{cases} x_{k+1} = f(x_k, n_{q,k}), \\ y_{k+1} = h(x_{k+1}, n_{r,k+1}), \end{cases} \tag{1}$$

where $(n_{q,k}, n_{r,k})$ are centered noise vectors and (f, h) two possibly nonlinear smooth mappings.

The filtering problem lies in the estimation of the posterior density $p(x_k|y_{1:k})$, where $y_{1:k} = [y_1\ , ...\ , y_k]$, under the following classical hypothesis:

- The measurements $y_{1:k}$ are mutually independent given the state;
- The state vectors $x_{1:k}$ describe a Markov process;
- The noise vectors $(n_{q,k}, n_{r,k})$ are independent and identically distributed;
- The initial state probability density function is known.

The estimation process hinges on two main steps. First, the state density is propagated using the Chapman-Kolmogorov equation defined by:

$$p(x_{k+1}|y_{1:k}) = \int p(x_{k+1}|x_k)p(x_k|y_{1:k})dx_k. \tag{2}$$

Then, when a measurement is available, an update step computes the posterior density based on the Bayes rule:

$$p(x_{k+1}|y_{1:k+1}) = \frac{p(y_{k+1}|x_{k+1})p(x_{k+1}|y_{1:k})}{\int p(y_{k+1}|x_{k+1})p(x_{k+1}|y_{1:k})dx_{k+1}}. \tag{3}$$

Equations (2) and (3) are referred to as the *optimal filter*. In the sequel, we denote the measurement likelihood with respect to the predicted distribution by $g(x_{k+1}) = p(y_{k+1}|x_{k+1})$ and the prior density by $q(x_{k+1}) = p(x_{k+1}|y_{1:k})$.

2) Particle Filter: Let the probability density function be approximated by a sample of N_p weighted particles $\{x^i, w^i\}_{i=1:N_p}$ describing a Dirac mixture:

$$\hat{p}(x_k) \approx \sum_{i=1}^{N_p} w_k^i \delta_{x_k^i}(x_k). \tag{4}$$

Then, the filtering problem reduces to a weight estimation on the particles set. The particles are first propagated according to (1):

$$x_{k+1|k}^i = f(x_k^i, n_{q,k}^i)\ ,\ i \in [1, N_p]. \tag{5}$$

When a new measurement is available, the weights are updated using the likelihood g:

$$w_{k+1}^i \propto w_k^i g(x_{k+1|k}^i)\ ,\ i \in [1, N_p]. \tag{6}$$

After a few updates, a large majority of the particles' weights tend to zero while a few tend to unity [11]. Therefore, a resampling step is triggered when degeneracy is about to occur. The particle weights are monitored using the criterion [12]:

$$N_{\mathit{eff}} = \frac{1}{\sum_{i=1}^{N_p}(w^i)^2} < N_{th}. \tag{7}$$

A resampling step occurs when N_{eff} goes below a given threshold $N_{th} = \theta.N_p$ where $\theta \in (0, 1)$ is an a priori defined parameter. Generally, resampling techniques aim at duplicating particles with a high weight and discard low weighted ones.

3) Laplace Particle Filter: The weak point of most resampling strategies lies in the selection of particles which mostly accounts for the prior density. Harnessing the measurement likelihood is expected to improve this step. The Laplace Particle Filter (LPF) [1] describes a resampling method based on an importance function centered on the *Maximum A Posteriori* (MAP), which showed superior accuracy and robustness in scenarios where classic particle filters usually fail.

The LPF core idea comes from the use of accurate importance sampling method when the degeneracy occurs. The importance sampling method provides an estimation of $p(x_{k+1}|y_{1:k+1})$ from the importance density $\tilde{q}$ in the sense of (4) :

$$\hat{p}(x_{k+1}|y_{1:k+1}) \approx \sum_{i=1}^{N_p} w_{k+1}^i \delta_{x_{k+1}^i}(x_{k+1}), \tag{8}$$

where the weights are defined as:

$$w_{k+1}^i \propto \frac{g(x_{k+1}^i)q(x_{k+1}^i)}{\tilde{q}(x_{k+1}^i)}. \tag{9}$$

The precision of the importance sampling method depends on the choice of the importance function $\tilde{q}$. The optimal importance sampling approximation is obtained for $\tilde{q}_{opt} = p(x_{k+1}|y_{1:k+1})$, which is the density to be estimated [13].

To that extent, the core idea of the LPF is to use the conditional expectancy $\mathbb{E}[x_{k+1}|y_{1:k+1}]$ and variance $\mathbb{V}[x_{k+1}|y_{1:k+1}]$ of $p(x_{k+1}|y_{1:k+1})$ as the mean and covariance of $\tilde{q}$, ensuring the latter to be close to p. Accurate approximations of the conditional expectancy and variance are given by the Laplace integration method after which the filter is named. The parameters x^L, P^L are defined as follows [1]:

$$\begin{cases} \mathbb{E}[x_{k+1}|y_{1:k+1}] \approx x^L & = x^* + \beta(x^*, J^*), \\ \mathbb{V}[x_{k+1}|y_{1:k+1}] \approx P^L & = (J^*)^{-1} + \gamma(x^*, J^*), \end{cases} \tag{10}$$

where β, γ are two *exact* high-order corrective terms, x^* denotes the MAP, and J^* the Observed Fisher Information matrix. These elements can be retrieved by solving the following problem:

$$\begin{cases} x^* & = & \max\limits_x g(x)q(x), \\ J^* & = & -\dfrac{\partial^2}{\partial x^2}\log g(x)q(x)\Big|_{x=x^*}. \end{cases} \tag{11}$$

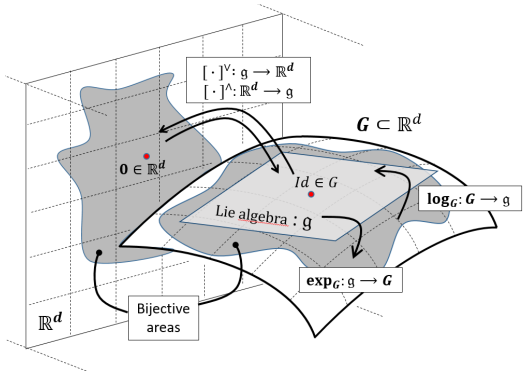


Fig. 1: Illustration of the Lie group structure. The group exponential $\exp_G$ and logarithm $\log_G$ define a bijection of G into $\mathbb{R}^d$.

For most applications, the following approximation yields suitable estimation accuracy:

$$\begin{cases} x^L & \approx & x^*, \\ P^L & \approx & (J^*)^{-1}. \end{cases} \tag{12}$$

Then, a new sample is drawn according to the importance density $\tilde{q}(x; x^L, P^L)$ of mean x^L and covariance P^L, and the weights are updated according to (9). This method provides a near-to-optimal resampling step to the particle filter.

B. Probabilistic estimation on Lie groups

1) Introduction to Lie groups: A Lie group $(G, \cdot)$ is a space having both a *group structure* and a *differential manifold structure* [14]. This enables to define the *Lie algebra*, denoted $\mathfrak{g}$, as the tangent space at I_G, the identity point of G : $\mathfrak{g} = T_{I_G}G$. Assuming G is a finite-dimensional *matrix Lie groups*, the *group exponential* $\exp_G$ and *logarithm maps* $\log_G$, are two bijective mappings between $\mathfrak{g}$ and G in the vicinity of I_G [15]. Furthermore, their expression reduces to the following matrix power series [14]:

$$\exp_G(X) = \sum_{k=0}^{\infty} \frac{X^k}{k!} \; ; \; \log_G(X) = \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k}(X - Id)^k. \tag{13}$$

Depending on the Lie group structure, exact closed forms of these functions exist, thus sparing the computation of high-order power series. Their composition with the algebra isomorphisms $[\cdot]^\wedge : \mathbb{R}^d \to \mathfrak{g}$ and $[\cdot]^\vee : \mathfrak{g} \to \mathbb{R}^d$ are denoted:

$$\exp_G([\cdot]^\wedge) = \exp_G^\wedge(\cdot) \; ; \; \log_G([\cdot]^\vee) = \log_G^\vee(\cdot). \tag{14}$$

The Lie group structure is summarized in Figure 1.

2) Error, Adjoint and group Jacobian: Let $(X, \widehat{X})$ be two elements of a Lie group $(G, \cdot)$. The group law $\cdot$ generally does not commute. Thus, there are two ways to define the error between X and $\widehat{X}$ with respect to $\cdot$: the *right* group error $e_R = \widehat{X} \cdot X^{-1}$ or the *left* group error $e_L = X^{-1} \cdot \widehat{X}$.

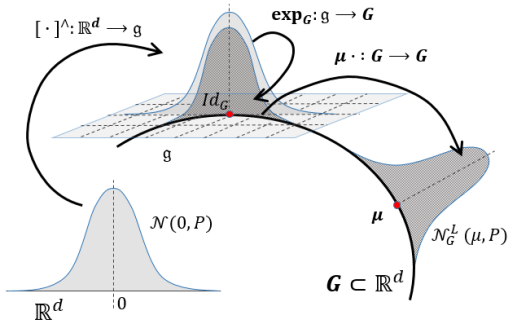


Fig. 2: Construction of a left concentrated Gaussian distribution on G based on the group exponential and left product.

The *group adjoint* [14] is an application $Ad_G : G \to \mathbb{R}^{d \times d}$ behaving like a *commutator* on matrix groups:

$$\forall X \in G, \epsilon \in \mathbb{R}^d : X \cdot \exp_G^\wedge(\epsilon) = \exp_G^\wedge(Ad_G(X)\epsilon) \cdot X. \tag{15}$$

The *algebra adjoint* is an application $ad_G : \mathbb{R} \to \mathbb{R}^{d \times d}$ such that:

$$\forall a \in \mathbb{R}, ad_G(a) = \log_m(Ad_G(\exp_G^\wedge(a))). \tag{16}$$

Based on the latter, the so-called *Lie group Jacobian* $\Phi_G : \mathbb{R}^d \to \mathbb{R}^{d \times d}$ is defined as [16]:

$$\forall \epsilon \in \mathbb{R}^d, \Phi_G(\epsilon) = \frac{\exp(ad_G(\epsilon)) - Id}{ad_G(\epsilon)}. \tag{17}$$

C. Uncertainties on Lie groups

Let $\mu \in G$, $\epsilon \sim \mathcal{N}(0, P)$ be a centered Gaussian random vector, and P a covariance matrix. A random matrix on $X \in G$ follows a *left* (resp. *right*) concentrated Gaussian distribution on G if:

$$\begin{array}{ll} \text{Left case:} & X \sim \mathcal{N}_G^L(X; \mu, P) \; ; \; X = \mu \cdot \exp_G^\wedge(\epsilon). \\ \text{Right case:} & X \sim \mathcal{N}_G^R(X; \mu, P) \; ; \; X = \exp_G^\wedge(\epsilon) \cdot \mu. \end{array} \tag{18}$$

This process is illustrated in Figure 2. The definition of the Gaussian density on Lie groups holds when the density is *concentrated* around its mean, that is to say, all eigenvalues of P are small enough [17]. This point is important to notice For the sake of brevity, this section only describes the *left* distribution case. Adaptation to the *right* distribution case is possible with only minor adjustments. Let $X \sim \mathcal{N}_G^L(\mu, P)$ a *left* random variable on a Lie group, and $\|\cdot\|_P$ the Mahalanobis norm with respect to P. The probability density function of a concentrated Gaussian on a Lie group writes: [16]

$$p_G(X) = \alpha(\epsilon)^{-1} e^{-\frac{1}{2}||\epsilon||_P^2}, \tag{19}$$

where $\epsilon = \log_G^\vee(\mu^{-1}X)$, and α is given by:

$$\alpha(\epsilon) = \sqrt{(2\pi)^d \det\left[\Phi_G(-\epsilon) P \Phi_G(-\epsilon)^T\right]}. \tag{20}$$

Note that alternative approaches [17] suggest an approximation of (20) when the singular values of P are small :

$$\alpha(\epsilon) \approx \sqrt{(2\pi)^d \det\left[P\right]}. \tag{21}$$

III. The Laplace particle filter on Lie groups

The main contributions of this paper are introduced in this section. It presents the key concepts of Laplace Particle Filter on Lie groups (LG-LPF).

A. Lie group Particle Filter Algorithm

Let $X \in G$ and $Y \in G'$ be two random variables on Lie groups admitting the discrete state-space model:

$$\begin{cases} X_{k+1} &= f(X_k, n_{q,k}), \\ Y_{k+1} &= h(X_{k+1}, n_{r,k}), \end{cases} \tag{22}$$

where $n_{q,k}$ and $n_{r,k}$ are centered noise vectors, $f : (G \times \mathbb{R}^d) \to G$ and $h : (G \times \mathbb{R}^d) \to G'$ are two smooth maps. The sequel aims to compute $p_G(X_k|Y_{1:k})$ using the approximation defined by (4). The initial particles are sampled according to (18). Then, they are propagated in line with (22):

$$X^i_{k+1|k} = f(X^i_k, n^i_{q,k}) \ , \ i \in [1, N_p], \tag{23}$$

where $n^i_{q,k} \sim \mathcal{N}(0, Q_k)$, Q_k is the process noise matrix. The propagated mean is the solution of:

$$\widehat{X}_k = \arg\min_{\mu} \sum_{i=1}^{N_p} w_i \log^{\vee}_G(\mu^{-1} \cdot X^i_k). \tag{24}$$

This problem is solved using Algorithm 1 [18]. The covariance

Algorithm 1: (Left) Sample Mean on Lie Groups

Result: μ
Initialization: μ^0, τ ,$\{X^i, w^i\}_{i\in[1,N_p]}$;
while $||\delta|| \geq \tau$ **do**
$\quad \delta^l = \sum_{i=1}^{N_p} w^i \log^{\vee}_G((\mu^l)^{-1} X^i)$
$\quad \mu^{l+1} = \mu^l \cdot \exp^{\wedge}_G(\delta^l)$
end

of the propagated sample is computed in the left case as:

$$\widehat{P}_{k+1|k} = \sum_{i=1}^{N_p} w_i \log^{\vee}_G(\widehat{X}^{-1}_{k+1|k} \cdot X^i_{k+1}) \log^{\vee}_G(\widehat{X}^{-1}_{k+1|k} \cdot X^i_{k+1})^T. \tag{25}$$

The update step is completed according to the likelihood in the group $g_{G'}(X_{k+1|k}) = p_{G'}(Y_{k+1}|X_{k+1|k})$:

$$w^i_{k+1} \propto w^i_k g_{G'}(X^i_{k+1|k}). \tag{26}$$

B. Laplace Particle Filter on Lie Groups (LG-LPF)

The core idea of LG-LPF is to fit a Gaussian importance function to the posterior density using a Gauss-Newton optimization algorithm on the Lie group. As discussed Section II a good importance function is given by:

$$\tilde{q}_G(X_{k+1}) \approx \mathcal{N}^L_G(X_{k+1}; X^L, P^L). \tag{27}$$

A first-order approximation gives $X^L \approx X^*$ and $P^L \approx (J^*)^{-1}$, where (X^*, J^*) are the MAP and the posterior information matrix on G. Hence, the goal of the sequel is to develop a method for computing X^* and J^* inspired from iterative Kalman filters on Lie groups [19].
First, the posterior density on G can be factorized as:

$$p_G(X_{k+1}|Y_{1:k+1}) \propto g_{G'}(X_{k+1}) q_G(X_{k+1}). \tag{28}$$

Under the concentrated Gaussian approximation, $g_{G'}$ and q_G are given by:

$$\begin{cases} q_G(X_{k+1}) \approx \mathcal{N}^L_G(X_{k+1}; \widehat{X}_{k+1|k}, \widehat{P}_{k+1|k}), \\ g_{G'}(X_{k+1}) \approx \mathcal{N}^L_{G'}(h(X_{k+1}); y_{k+1}, R_{k+1}). \end{cases} \tag{29}$$

The mean of the fitted Gaussian (i.e the MAP) can be found by minimizing the negative log-likelihood of (28) using the approximation (21):

$$X^* = \min_{X\in G} || \log^{\vee}_{G'}(y^{-1}_{k+1} h(X))||^2_{R_{k+1}} + || \log^{\vee}_G(\widehat{X}^{-1}_{k+1|k} X))||^2_{\widehat{P}_{k+1|k}}. \tag{30}$$

The posterior information matrix J^* can be obtained using a Lie Group Extended Kalman Filter (LG-EKF) whose linearization point is at the MAP. This process, described in Algorithm 2, is analogous to Lie Group Iterative Extended Kalman Filter (LG-IEKF) update step [19]. Then, a new set

Algorithm 2: (Left) Gauss-Newton algorithm on Lie groups

Result: X^*, J^* at convergence
Inputs: μ, X^0, δ^0, τ, P ;
while $||\delta^l - \delta^{l-1}|| \geq \tau$ **do**
$$H_l = -\frac{\log_{G'}\left(\partial h(X^l \exp^{\wedge}_G(\epsilon))^{-1} y_{k+1}\right)}{\partial \epsilon}\Big|_{\epsilon=0}$$
$$K_l = P\phi^T_G(\delta^l) H^T_l \left[H_l \phi_G(\delta^l) P \phi^T_G(\delta^l) H^T_l + R\right]^{-1}$$
$$\delta^{l+1} = K_l \left(\log^{\vee}_{G'}(y^{-1}_{k+1} h(X^l)) + H_l \delta^l\right)$$
$$X^{l+1} = \mu \exp^{\wedge}_G(\delta^{l+1})$$
end
At convergence:
$X^* = X^l$
$J^* = \left[\phi_G(\delta^l)(Id - K_l H_l \phi_G(\delta^l)) P \phi_G(\delta^l)\right]^{-1}$

of particles $\{X^i_L\}_{i\in[1:N_p]}$ is drawn according to $\tilde{q}_G(X) = \tilde{q}_G(X; X^L, P^L) \approx \mathcal{N}^L_G(X; X^*, (J^*)^{-1})$. The weights are computed through the importance sampling process:

$$w^i_{k+1} = \frac{g_{G'}(X^i_L) q_G(X^i_L)}{\tilde{q}_G(X^i_L)}. \tag{31}$$

Finally, the weights are normalized and the updated mean and covariance are computed from Algorithm 1 and (25).

C. Discussion About the Method

The main interest of Lie groups is to provide a natural framework for angular variables (e.g. quaternions, rotation matrices...), as arbitrarily large errors are always exactly defined, unlike Euclidean filters which often use small rotation errors as an approximation. Moreover, an apt choice of the group involves natural constraints on the state variables, leading to a better covering of the state-space by the particles.

Algorithm 3: The (left) Lie Groups Laplace Particle Filter

Result: $\widehat{X}_{0:T}$ and $\widehat{P}_{0:T}$
Initialization: sample $X_0^i \sim \mathcal{N}(X; \widehat{X}_0, \widehat{P}_0)$ (18);
Prediction: $X_{k+1|k}^i = f(X_k^i, n_{q,k}^i)$, $i \in [1, N_p]$
Update: $w_{k+1}^i \propto w_k^i \cdot g_{G'}(X_{k+1|k}^i)$
if $N_{eff} < N_{th}$ **then**
  Prior mean: $\widehat{X}_{k+1|k}$ from Algorithm 1
  Prior covariance: $\widehat{P}_{k+1|k}$ from (25)
  Compute: X_{k+1}^*, J_{k+1}^* from Algorithm 2
  Compute: $X_{k+1}^L \approx X_{k+1}^*$ and $P_{k+1}^L \approx (J_{k+1}^*)^{-1}$
  Drawn: $X_L^i \sim \tilde{q}_G(X; X_{k+1}^L, P_{k+1}^L)$, $i \in [1, N_p]$
  Update: $w_{k+1}^i = \dfrac{g_{G'}(X_L^i) q_G(X_L^i)}{\tilde{q}_G(X_L^i)}$
  Normalize: $w_{k+1}^i = \dfrac{w_{k+1}^i}{\sum_{i=1}^{N_p} w_{k+1}^i}$
end
Mean: $\widehat{X}_{k+1}$ from Algorithm 1
Covariance: $\widehat{P}_{k+1}$ from (25)

Without loss of generality, the presented method is developed for Gaussian models and provides a framework that suits most applications. For severely non-Gaussian scenarios, a general LPF framework [13] can be applied to Lie Groups to derive the method for any density.

IV. Application and Numerical Results

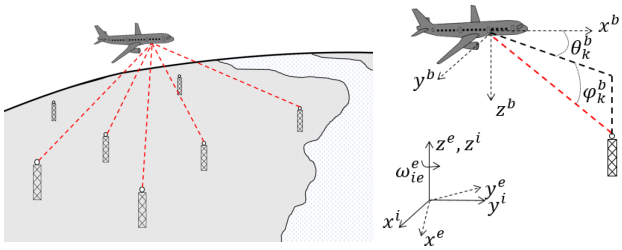
A. Principles of radio navigation

Many aerial positioning systems rely on GPS, which may provide faulty information or undergo outages. The use of a data fusion algorithm for long-range navigation lessens the adverse impact of a sensor failure and improves the performance of positioning in nominal conditions. Radio Direction Finders (RDF) are radio navigation systems that consist of short-medium-range VHF beacons, each having a specific frequency and identification code. An embedded device provides the angle of arrival for each beacon within the range of the system. In the sequel, the beacons are represented by a set of fixed landmarks whose positions are known and denoted $p_{eb,k}^e$ for the k^{th} landmark. The antenna on the aircraft measures the azimuth and the elevation of the aircraft to the beacon line of sight. The situation is illustrated in Figure 3 and the angular measurements equations are:

$$\begin{cases} \theta_k^b &= \arctan 2\left(\Delta_{y,k}^b, \Delta_{x,k}^b\right), \\ \varphi_k^b &= \arctan 2\left(-\Delta_{k,z}^b, \sqrt{(\Delta_{x,k}^b)^2 + (\Delta_{y,k}^b)^2}\right), \end{cases} \tag{32}$$

where $\Delta^b = C_e^b(p_{eb,k}^e - x_{eb}^e)$ is the relative distance between a landmark and the aircraft resolved in the aircraft frame **[b]** defined in Figure 3b, and $\arctan 2(y, x)$ is such that $\forall (x, y) \neq (0, 0)$:

$$\arctan 2(y,x) = \begin{cases} \text{sign}(y) \arctan\left|\frac{y}{x}\right| & x > 0, \\ \text{sign}(y)\frac{\pi}{2} & x = 0, \\ \text{sign}(y)(\pi - \arctan\left|\frac{y}{x}\right|) & x < 0. \end{cases} \tag{33}$$



(a) Global scheme (b) Frames and parameters

Fig. 3: Illustration of the angles of arrival measurements from fixed beacons*(a)*. The parametrization is given in *(b)*.

B. Long-range navigation

Long-range navigation must account for Coriolis forces and ellipsoidal Earth effects. To that extent, the kinematics of the aircraft is described using the WGS84 model resolved in the Earth Centered Earth Fixed (ECEF) frame [20]:

$$\begin{cases} \dot{C}_b^e = C_b^e \Omega_{ib}^b - \Omega_{ie}^e C_b^e, \\ \dot{v}_{eb}^e = C_b^e f_{ib}^b + g^e(x_{eb}^e) - 2\Omega_{ie}^e v_{eb}^e, \\ \dot{x}_{eb}^e = v_{eb}^e, \end{cases} \tag{34}$$

where C_b^e denotes the rotation matrix from the aircraft frame **[b]** to the ECEF frame **[e]**, v_{eb}^e, x_{eb}^e are respectively the velocity and position of **[b]** with respect to **[e]** resolved in **[e]**. Ω_{ie}^e and Ω_{ib}^b are the skew-symmetric matrices of the Earth rotation rate and the gyroscopes output rotation rate, f_{ib}^b is the specific acceleration measured by the accelerometers and $g^e(x_{eb}^e)$ is the estimated local gravity which is the sum of the gravitational force due to Earth gravity and the centrifugal force due to Earth rotation [20].
The embedded navigation system has a high-grade Inertial Measurement Unit (IMU) running at 50Hz, which noise levels are described in Table I.

C. Practical Implementation of LG-LPF

1) Choice of group and probability densities: The first step to implement the filter is to select a proper Lie group in which the state matrix can be defined. An interesting choice is the Special Euclidean group $SE_n(3), n \in \mathbb{N}$ such that the particles are defined as follows:

$$\forall i \in [1, N_p], k > 0 : X_k^i = \begin{pmatrix} C_{b,k}^{e,i} & v_{be,k}^{e,i} & x_{be,k}^{e,i} \\ 0_{1,3} & 1 & 0 \\ 0_{1,3} & 0 & 1 \end{pmatrix} \in SE_2(3). \tag{35}$$

This choice allows the couplings between the attitude matrix C_b^e and the other variables, which is expected to bring more

consistency during the updates. Hence, all the filter particles are $SE_2(3)$ matrices. For the sake of simplicity the probability densities are modelled as concentrated Gaussians on G. The measurements belong to $G' = \mathbb{R}^2$, which is a Lie group such that $\log_{G'} = \exp_{G'} = I_2$. Thus the likelihood of the angle of arrival measurement model writes:

$$p_{G'}(y_{k+1}|X^i_{k+1}) = \frac{\exp\left(-\frac{1}{2}||y_{k+1} - h(X^i_{k+1})\,||^2_{R_{k+1}}\right)}{\sqrt{(2\pi)^9 \det[R_{k+1}]}}, \tag{36}$$

where $h(X) = (\theta^b, \varphi^b)$ is computed for each landmark with (32) and R is the measurement noise matrix.

2) Propagation: The propagation of the particles is obtained thanks to a time discretization with time step dt in (34). The exponential map on $SO(3)$ enables an accurate integration of the attitude rotation matrix. $\forall k > 0, \forall i \in [1, N_p], \epsilon^i_k = \begin{pmatrix}\epsilon^i_{r,k} & \epsilon^i_{v,k} & \epsilon^i_{x,k}\end{pmatrix} \sim \mathcal{N}(0, Q_k)$:

$$\begin{cases} C^{e,i}_{b,k+1|k} = C^{e,i}_{b,k} \exp_{SO(3)}\left(\mathrm{dt}(\Omega^b_{ib} - C^i_k \Omega^e_{ie} C^i_{k'}) + \epsilon^i_{r,k}\right), \\ v^{e,i}_{eb,k+1|k} = v^{e,i}_{eb,k} + \mathrm{dt}\left(C^i_k f^b_{ib} + g^e(x^i_k) - 2\Omega^e_{ie} v^i_k\right) + \epsilon^i_{v,k}, \\ x^{e,i}_{eb,k+1|k} = x^{e,i}_{eb,k} + \mathrm{dt} v^i_k + \epsilon^i_{x,k}. \end{cases} \tag{37}$$

3) Update: When a measurement is available, the particle weights are updated using (36):

$$\forall i \in [1, N_p] : w^i_{k+1} \propto w^i_k \exp(-\frac{1}{2}||y_{k+1} - h(X^i_{k+1|k})||^2_{R_{k+1}}) \tag{38}$$

When the resampling criterion (7) goes below the threshold $N_{\text{th}} = \theta N_p$, where $\theta = 0.6$ in this paper, the particles are about to degenerate and LG-LPF triggers Algorithm 3. In the case of a left distribution, the measurement Jacobian on $SE_2(3)$ writes:

$$H = - \left.\frac{\partial \log_{\mathbb{R}^2}\left(h(X \exp_{SE_2(3)}(\epsilon))^{-1} y\right)}{\partial \epsilon}\right|_{\epsilon=0}. \tag{39}$$

It is worth noting that H represents the linearization of the measurement function at X and is different from the group Jacobian (17). Let $\Delta^b : G \to \mathbb{R}^3$ and $\eta : \mathbb{R}^3 \to \mathbb{R}^2$ such that $h = \eta \circ \Delta^b$, the derivation chain rule gives:

$$H = \left.\frac{\partial h(\Delta)}{\partial \Delta}\right|_{\Delta=\Delta^b(X)} \left.\frac{\partial \Delta^b(X \exp_{SE_2(3)}(\epsilon))}{\partial \epsilon}\right|_{\epsilon=0}. \tag{40}$$

Then, taking $\Delta^b = \begin{pmatrix}\Delta^b_x \\ \Delta^b_y \\ \Delta^b_z\end{pmatrix}$ and $\rho = \sqrt{(\Delta^b_y)^2 + (\Delta^b_x)^2}$:

$$\left.\frac{\partial h(\Delta)}{\partial \Delta}\right|_{\Delta=\Delta^b(X)} = \begin{pmatrix} -\frac{\Delta^b_y}{\rho^2} & \frac{\Delta^b_x}{\rho^2} & 0 \\ \frac{\Delta^b_x \Delta^b_z}{\rho ||\Delta^b||^2} & \frac{\Delta^b_y \Delta^b_z}{\rho ||\Delta^b||^2} & \frac{-\rho}{||\Delta^b||^2} \end{pmatrix}, \tag{41}$$

$$\left.\frac{\partial \Delta^b(X \exp_{SE_2(3)}(\epsilon))}{\partial \epsilon}\right|_{\epsilon=0} = \begin{pmatrix}[\Delta^b(X)]_\times & 0_{3,3} & -I_3\end{pmatrix}, \tag{42}$$

where $[u]_\times$ is the skew-symmetric matrix of the vector u.

D. Testing Framework

This section aims to compare LG-LPF with Euclidean Regularized Particle Filter (RPF) and (Euclidean) Laplace Particle Filter (LPF). They are tested on a straight trajectory in ECEF Figure 4 with angles-only measurements. To test

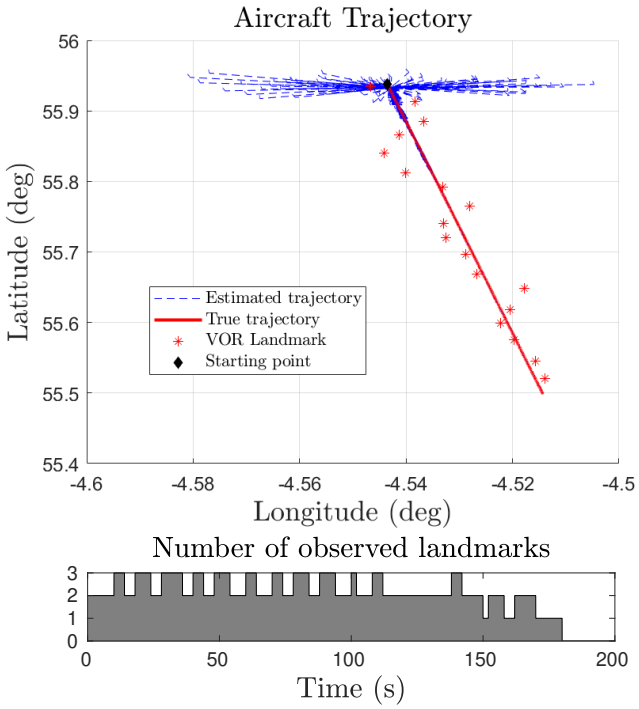


Fig. 4: View of the estimated trajectories of the aircraft for LG-LPF (100 particles) for 20 Monte Carlo runs with different initial errors compared to the true trajectory. The number of observed beacons varies along the trajectory.

the algorithms, fictional beacons are generated all along the ground path of the aircraft. They are represented in Figure 3. Also, the sensor range is limited in this testing framework. The number of visible beacons at each time of the trajectory is displayed in Figure 3.

1) Performance criteria: The filters are evaluated using several criteria. The percentage of convergent Monte Carlo runs is relevant to assess the robustness of the filter. A run is considered convergent if the mean position of the state is contained inside the confidence ellipsoid Γ_k computed by the Posterior Cramer-Rao Bound (PCRB) [21], for the last five measurement iterations as follows:

$$\Gamma_k = \left\{x^e_{be,k} | (x^e_{be,k} - \widehat{x}^e_{be,k})^T \text{PCRB}^{-1}_k (x^e_{be,k} - \widehat{x}^e_{be,k}) \leq \kappa\right\} \tag{43}$$

where the threshold κ is chosen from the test $p(\chi^2(d) \leq \kappa^2) = 0.99$ with $d = 9$ the dimension of the state vector.

The Average Root Mean Square Error (ARMSE) assesses the accuracy of the filter over a period of time. In this paper, the ARMSE values displayed are averaged over the last minute

Sensor Parameters			
Sensor rates (Hz)	IMU: 50Hz	RDF: 1Hz	
IMU noise (1σ)	Gyro: $2\,\mathrm{deg}/h$	Acc: $10^{-3} m/s^2$	
RDF noise (1σ)	Azimuth: $0.6°$	Elevation: $0.6°$	
Filter parameters			
Initial uncertainties	Attitude	Velocity	Position
Nominal (1σ)	$0.115°$	$10ms^{-1}$	$1km$
Poor (1σ)	$11.50°$	$50ms^{-1}$	$10km$
Process noise (1σ)	Attitude	Velocity	Position
	$20°$	$10^{-2} m/s$	$10^{-2} m$
Update noise	Azimuth: $2.8°$	Elevation: $2.8°$	
Resampling threshold		$N_{\text{th}} = 0.6N_p$	

TABLE I
Simulation and filters parameters for the two scenarios.

of the trajectory. The RMSE at time k is computed from convergent runs only based on the criterion (43) as:

$$\text{RMSE}_e(k) = \sqrt{\frac{1}{N_{\text{conv}}} \sum_{m=1}^{N_{\text{conv}}} ||e(k)||_2^2}, \tag{44}$$

where N_{conv} represents the number of convergent runs, and the error vector $e(k)$ is either defined as:

$$e(k) = \begin{cases} \log^{\vee}_{SO(3)} \left((C^e_{b,k})^T \widehat{C}^e_{b,k,m} \right) \\ v^e_{be,k} - \widehat{v}^e_{be,k,m} \\ x^e_{be,k} - \widehat{x}^e_{be,k,m} \end{cases} \tag{45}$$

The ARMSE is computed from (44) over the period T of N_T steps ranging from k_{ini} to k_{end}:

$$\text{ARMSE}_T = \frac{1}{N_T} \sum_{k=k_{\text{ini}}}^{k_{\text{end}}} \text{RMSE}_e(k). \tag{46}$$

In this paper the results are displayed for the last $T = 60s$ (i.e. $N_T = 600$), and $N_{\text{mc}} = 100$.

2) Simulation scenarios: Navigation systems are initialized with the best-known information at the beginning of their mission. This process, called alignment, is often challenging and can have a significant impact on the performance of the system. The filters are tested for two different initial errors detailed in Table I. The nominal scenario intends to compare the three filters with usual initialization errors. The poor alignment scenario enables to assess the ability of the LG-LPF to withstand large initialization errors. This case can occur after a long measurement outage and in a situation where the filter cannot be properly aligned or when using lower grade sensors.

E. Numerical Results

1) Nominal initialization: The simulation results for the nominal scenario are displayed in Table II. The percentage of convergent runs for RPF shows low robustness to the scenario while LPF and LG-LPF are performing well. Furthermore, LG-LPF shows improved performances on every state variable especially regarding the attitude estimation compared to LPF. The attitude ARMSE of RPF (* in Table II) have to be viewed critically. Given the low convergence rate of the filter, they are likely to reflect the survivor bias whereby only the runs with very low attitude error converged. This emphasises the fact that in this case, RPF is not robust to attitude errors while LPF and LG-LPF bring robustness.

$N_p = 500$	**RPF**	**LPF**	**LG-LPF**
Convergent runs	47%	100%	100%
Position (m)	66.0	12.6	7.89
Velocity (m/s)	2.21	0.95	0.69
Yaw (°)	0.048*	0.299	0.092
Pitch (°)	0.032*	0.142	0.088
Roll (°)	0.026*	0.101	0.072

TABLE II: Comparison of the ARMSE of the RPF, LPF and LG-LPF with a nominal initialization.

2) Poor initialization: The simulation results for the poorly initialized scenario are displayed in Table III and the RMSE are displayed in Figure (5). In this case, the focus is on LG-LPF as the initial errors are far beyond the working zone of the Euclidean filters. The simulation shows that LG-LPF is performing well for a very little number of particles. Adding particles to the filter improves its robustness and accuracy.

N_p	**100**	**500**	**1000**
Convergent runs	78%	84%	82%
Position (m)	17.4	8.86	8.78
Velocity (m/s)	2.1	0.85	0.71
Yaw (°)	0.311	0.082	0.041
Pitch (°)	0.248	0.099	0.079
Roll (°)	0.282	0.092	0.073

TABLE III: LG-LPF tested in the poor initialization case for different numbers of particles.

3) Results discussion: The performance of RPF and LPF in the nominal initialization scenario Table II shows that the Laplace method greatly improves robustness and accuracy. Such difference directly comes from the choice of the importance function discussed in Section II. Indeed, RPF would require a lot more particles to match the robustness results of LPF. Then, LG-LPF shows improved results compared to LPF on each variable of the state vector. This behavior was anticipated, as discussed in Section II-B, the particles and the stochastic processes defined on Lie groups fully embrace the nonlinear nature of the state vector model. Thus, the curved geometry on the Lie group behaves as a natural constrain. This prevents the particles from spreading to non-attainable states and minimizes the intrinsic errors of the estimation process. Besides, the particles are directly resampled at the neighborhood of the MAP with a close-to-optimal importance function as Algorithm 2 enables a full state optimization with improved accuracy and stability

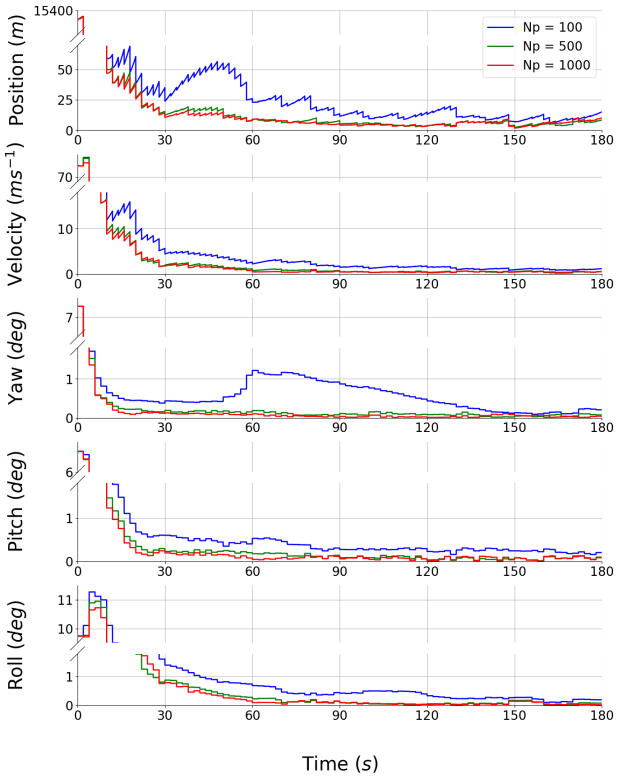


Fig. 5: LG-LPF RMSE for several numbers of particles N_p in the case of the poorly initialized scenario (large initial errors). The results are displayed for 100 Monte Carlo runs.

compared to LPF. The coupling between the variables involved in the matrix error defined on $SE_k(3)$ enhances the estimation of velocity and position. The three filters show similar execution times for the same number of particles.

Considering results in Table III, LG-LPF performs well on a very difficult scenario where LPF and RPF failed to run even for a very small number of particles. Moreover, LG-LPF still achieves better accuracy than LPF and RPF on the nominal scenario. This significant improvement in robustness comes from the exact definition of the group error.

Thus, LG-LPF shows an improved behavior compared to state-of-the-art particle filters. It can handle very high initialization errors, which is desirable when the alignment is not accurate. Finally, LG-LPF runs properly even on a very small amount of particles. This could greatly reduce the computation cost, compared to Euclidean filters, when resources are limited.

V. Conclusion

This paper introduces a general framework for particle filters on Lie groups and describes a new resampling strategy at the MAP. This formulation enables Laplace Particle Filter on Lie groups (LG-LPF) to produce improved results for nonlinear navigation problems (e.g. angle of arrival measurements). Furthermore, LG-LPF performs well, even with a limited amount of particles, and is robust to large estimation errors. Thus, LG-LPF addresses the main drawbacks of particles filters (e.g. computational cost, angles error management) with promising results. Future work will focus on testing the proposed algorithm on more scenarios and make further comparisons with currently state-of-the-art algorithms.

Acknowledgment

The authors are thankful to the Innovation and Defense Agency (Agence Innovation Défense) from the French Ministry of the Armies for their support on this research work.

References

[1] P. B. Quang, C. Musso, and F. Gland, "Particle filtering and the Laplace method for target tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, pp. 350–366, 2016.

[2] N. Merlinge, K. Dahia, H. Piet-Lahanier, J. Brusey, and N. Horri, "A box regularized particle filter for state estimation with severely ambiguous and non-linear measurements," *Autom.*, vol. 104, pp. 102–110, 2019.

[3] G. Bourmaud, R. Mégret, M. Arnaudon, and A. Giremus, "Continuous-discrete extended Kalman filter on matrix Lie groups using concentrated Gaussian distributions," *Journal of Mathematical Imaging and Vision*, vol. 51, pp. 209–228, 2014.

[4] A. Barrau, "Non-linear State Error Based Extended Kalman Filters with Applications to Navigation," Ph.D. dissertation, Ecole Nationale Supérieure des Mines de Paris, 2015.

[5] A. Barrau and S. Bonnabel, "The invariant extended Kalman filter as a stable observer," *IEEE Transactions on Automatic Control*, vol. 62, pp. 1797–1812, 2017.

[6] ——, "Linear observed systems on groups," *Syst. Control. Lett.*, vol. 129, pp. 36–42, 2019.

[7] G. Magalhães, Y. Caceres, J. B. R. D. Val, and R. S. Mendes, "UKF on Lie groups for radar tracking using polar and doppler measurements," 2018.

[8] M. Brossard, S. Bonnabel, and J.-P. Condomines, "Unscented Kalman filtering on Lie groups," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2485–2491, 2017.

[9] A. Barrau and S. Bonnabel, "Invariant particle filtering with application to localization," *53rd IEEE Conference on Decision and Control*, pp. 5599–5605, 2014.

[10] H.-L. Ge, Z. Zhu, and K. Lou, "Tracking video target via particle filtering on manifold," *Information Technology and Control*, vol. 48, pp. 538–544, 2019.

[11] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, pp. 174–188, 2002.

[12] A. Kong, J. S. Liu, and W. Wong, "Sequential imputations and Bayesian missing data problems," *Journal of the American Statistical Association*, vol. 89, pp. 278–288, 1994.

[13] C. Musso, P. B. Quang, and F. Gland, "Introducing the Laplace approximation in particle filtering," *14th International Conference on Information Fusion*, pp. 1–8, 2011.

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

[15] B. Hall, "Lie groups, lie algebras, and representations: An elementary introduction," 2004.

[16] G. Bourmaud, "Estimation de pramètres évoluant sur des groupes de Lie : application la cartographie et la loalisation d'une caméra monoculaire," Ph.D. dissertation, Université de Bordeaux, 2015.

[17] 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.

[18] N. Miolane, "Defining a mean on Lie groups," *Imperial College London*, 2013.

[19] G. Bourmaud, R. Mégret, A. Giremus, and Y. Berthoumieu, "From intrinsic optimization to iterated extended Kalman filtering on Lie groups," *Journal of Mathematical Imaging and Vision*, vol. 55, pp. 284–303, 2015.

[20] P. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Second Edition*, 2013.

[21] P. Tichavský, C. Muravchik, and A. Nehorai, "Posterior Cramer-Rao bounds for discrete-time nonlinear filtering," *IEEE Trans. Signal Process.*, vol. 46, pp. 1386–1396, 1998.