

The Integrated Probabilistic Data Association Filter Adapted to Lie Groups

Mark E. Petersen
Brigham Young University, Provo, UT

Randal W. Beard, Fellow, IEEE
Brigham Young University, Provo, UT

Abstract— The Integrated Probabilistic Data Association Filter (IPDAF) is a target tracking algorithm based on the Probabilistic Data Association Filter that calculates a statistical measure that indicates if an estimated representation of the target properly represents the target or is generated from non-target-originated measurements. The main contribution of this paper is to adapt the IPDAF to constant velocity target models that evolve on connected, unimodular Lie groups, and where the measurements are also defined on a Lie group. We present an example where the methods developed in the paper are applied to the problem of tracking a ground vehicle on the special Euclidean group $SE(2)$.

Index Terms— Tracking, Estimation, Integrated Probabilistic Data Association (IPDAF), Lie Group, Multiple Target Tracking

I. Introduction

When tracking a single target whose initial position is unknown, all real-world sensors produce non-target-originated measurements (e.g. false measurements or clutter). In the presence of dense clutter, it is a challenge to locate and track the target since it is difficult to distinguish between target-originated measurements and false measurements. The typical approach is to use new measurements to either improve the estimate of an existing track (a track is a representation of the target which consists of at least the state estimate) or initialize new tracks. If the clutter density is high, numerous tracks that do not represent the target are initialized from clutter.

An example scenario illustrating the challenge of tracking in clutter is depicted in Fig. 1, where the black dots represent measurements, the green car represents the true target, and the blue cars represent the tracks currently in memory. The left image represents a time step when measurements are received for the first time.

This work has been funded by the Center for Unmanned Aircraft Systems (C-UAS), a National Science Foundation Industry/University Cooperative Research Center (I/UCRC) under NSF award No. IIP-1650547, along with significant contributions from C-UAS industry members. (Corresponding author: R. Beard, email: beard@byu.edu).

The right image represents a subsequent time step when previous measurements are used to initialize new tracks and additional measurements are received. The challenge is to identify which track represents the target.

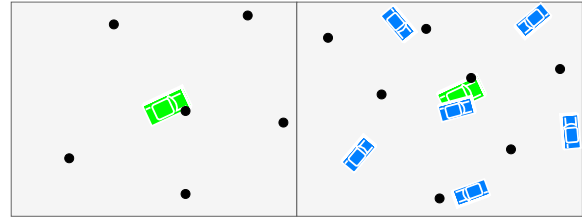


Fig. 1: A depiction of the challenge of identifying which track represents the target. The black dots represent measurements, the green car represents the target, and the blue cars represent tracks. The left image represents the first time step that the measurements are received. The right image represents the next time step when previous measurements are used to initialize new tracks and additional measurements are received.

Identifying which track best represents the target requires an additional estimate called the *track likelihood*. Tracks with a low track likelihood are rejected and pruned from memory, while tracks with a high track likelihood are confirmed as good tracks. The confirmed track with the highest track likelihood can be used as the best estimate of the target.

Different approaches to calculating the track likelihood depend on the data association algorithm. Data association is the process of assigning new measurements to existing tracks so that the associated measurement can be used to improve the estimates of the tracks. There are two types of data association: hard data association and soft data association. Hard data association assigns at most one new measurement to each track, and soft data association can assign multiple new measurements to each track.

Tracking algorithms that use hard data association, such as the Nearest Neighbor filter (NNF) [1], the Global Nearest Neighbor filter (GNNF) [2, 3] and track splitting algorithms, commonly use the likelihood function or the negative log likelihood function (NLLF) to determine if a track should be rejected [4]. The likelihood function measures the joint probability density of all of the measurements associated with a track. If the track's likelihood function falls below some threshold, the track is rejected. This approach does not indicate which track best represents the target, only which tracks should be removed. However, it is common to suppose that the track with the highest track likelihood best represents the target.

Another approach to quantify the track likelihood when using hard data association is based on the sequential probability ratio test (SPRT) [5]. The SPRT uses a sequence of data to either confirm a null hypothesis or reject the alternate hypothesis by analyzing the probability ratio of the two hypotheses. In terms of tracking,

the SPRT calculates the joint probability density of the measurements associated to a track for the hypothesis that all the measurements are target originated and for the hypothesis that all the measurements are false. It then takes the ratio of the two probability densities and either rejects the track if the ratio is below a threshold, confirms the track if the ratio is above a threshold or continues to gather more information as new measurements are received until the track can be confirmed or rejected [6]. The SPRT is used in a variety of tracking algorithms including the Multiple Hypothesis Tracker (MHT) [7].

Many of the common hard data association algorithms are either computationally inexpensive and not robust to clutter like the NNF and the GNNF or very computationally expensive and robust like the MHT and other track splitting methods. On the other hand, soft data associations algorithms, like the probabilistic data association filter (PDAF) [8, 9], offer a good balance between robustness and computational expense. Similar to a track splitting method, when multiple measurements are associated to a track the PDAF makes a copy of the track for every associated measurement, and then the copy is updated with one of the associated measurements resulting in the track splitting for every measurement. The PDAF differs from the track splitting method in that after track splitting, the split tracks are fused together into a single track according to the likelihood of each split track representing the target.

A nice feature of the PDAF is that it can be used with many different types of dynamic models including dynamic models evolving on Lie groups. Using Lie groups to model rigid body dynamics is a recent approach [10, 11, 12, 13, 14, 15]. Lie groups are beneficial since they naturally model the target's pose and motion without suffering from the singularities inherent in other attitude representations like Euler angles, axis-angle, etc. In addition, Lie group models more realistically model the uncertainty that exists with physical systems. Because of these features, Lie groups have shown increased accuracy in estimation [16]. Many physical systems are more naturally represented using Lie groups including satellite attitude dynamics, fixed-wing unmanned aircraft systems, multirotors and other flying objects, cars on road networks, ground robots and walking pedestrians.

The two aforementioned methods of quantifying the track likelihood do not work for soft data association algorithms. For this reason, the PDAF was extended in [17] to calculate the track likelihood using a novel approach called the Integrated Probabilistic Data Association Filter (IPDAF). To our knowledge, the IPDAF has not been adapted to Lie groups. However, the joint integrated probabilistic data association filter (JIPDAF) was adapted to the Lie group $SE(2)$ in [18]. The JIPDAF is the adaptation of the IPDAF to tracking multiple targets. When tracking only a single target, the JIPDAF reduces to the IPDAF. However, the reduction would require understanding the more complicated JIPDAF instead of the simpler IPDAF. Since the main focus of [18] was not

the adaptation of the JIPDAF to general Lie groups, the algorithm was not derived and explained in detail.

The purpose of this paper is to present the IPDAF adapted to connected, unimodular Lie groups in a tutorial manner by providing enough detail to make it clear how it can be implemented. We refer to the resulting algorithm as the Lie group integrated probabilistic data association filter (LG-IPDAF). Our focus in this paper will be on nearly constant velocity models, where the velocity is an element of the Lie algebra. We will primarily consider measurement models where the measurement is an element of the Lie group, e.g., position and orientation for $SE(2)$. In addition, we make several unique contributions by explaining the validation region for Lie groups, showing how the indirect Kalman filter is used with Lie groups and by presenting the system model more generally than [18] by representing the target's pose as an element of a Lie group and its velocity as an element of the associated Lie algebra instead of representing the velocity using a constrained element of the Lie group. The modeling approach used in [18] can be problematic when angular velocities are sufficiently high since the mapping from the Lie algebra to the Lie group is surjective and not bijective. Our last contribution is to present the material generically so that it can be easily applied to any Lie group, whereas in [18] the algorithm is specifically presented for $SE(2)$.

The rest of this paper is outlined as follows. In Section II we review basic concepts of Lie group theory to establish notation and to enable those not familiar with Lie groups to follow the subsequent development. In Section III we present an overview of the LG-IPDAF algorithm. In Section IV we present the system model used by the LG-IPDAF. In Sections V we give detailed derivations of the key elements of the LG-IPDA Filter including the prediction, data association, and update steps as well as the track initialization scheme. In Section VI we present a simple simulation example and conclude in Section VII. Detailed proofs of some of the results are given in the Appendices to facilitate the flow of the paper.

II. Lie Group Theory Review

Lie group theory is an extensive topic that we cannot completely cover in this paper, but excellent tutorials for robotic applications are given in [13, 19]. We also recommend [12, 20, 21, 22, 23, 24] for a more rigorous treatment of Lie group theory. In this paper, we restrict our discussion to connected, unimodular Lie groups, an assumption that is required by our model of the uncertainty on the Lie group as will be explained later. The objective of this section is to provide a brief review and to establish notation. Even though we illustrate our notation using the Lie group $SE(2)$, the provided definitions are sufficiently generic to apply to every connected, unimodular Lie group.

A. Lie Group and Lie Algebra

Let G denote a Lie group and \mathfrak{g} denote its corresponding Lie algebra. We identify the Lie algebra with the tangent space of G at the identity element. For example, the special Euclidean group $SE(2)$ is a matrix Lie group used to model rigid body motion in two dimensions. It is isomorphic to the set

$$SE(2) \cong \left\{ \begin{bmatrix} R & p \\ 0_{1 \times 2} & 1 \end{bmatrix} \middle| R \in SO(2), p \in \mathbb{R}^2 \right\}, \quad (1)$$

with the group operator being matrix multiplication, R denoting a 2-dimensional rotation matrix that represents the attitude of the rigid body, and p denoting the position of the rigid body. In this work we will use R to denote the rotation from the body frame to the inertial frame, and p to denote the position of the body with respect to the inertial frame expressed in the inertial frame.

The Lie algebra of $SE(2)$ is denoted $\mathfrak{se}(2)$ and is isomorphic to the set

$$\mathfrak{se}(2) \cong \left\{ \begin{bmatrix} [\omega]_{\times} & \rho \\ 0_{1 \times 2} & 0 \end{bmatrix} \middle| \omega \in \mathbb{R}, \rho \in \mathbb{R}^2 \right\}, \quad (2)$$

with ρ denoting the translational velocity, ω denoting the angular velocity, and $[\cdot]_{\times}$ being the skew symmetric operator defined as

$$[\omega]_{\times} = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix}. \quad (3)$$

In this work we will use ρ to denote the translational velocity of the body with respect to the inertial frame expressed in the body frame, and ω to denote the rotational velocity of the body with respect to the inertial frame expressed in the body frame.

The Lie algebra can take on various representations. However, by taking advantage of its algebraic structure, elements of the Lie algebra can be expressed as the linear combination of orthonormal basis elements $\{e_i\} \subset \mathfrak{g}$. For example, let $\mathbf{v} \in \mathfrak{se}(2)$, then $\mathbf{v} = \sum_{i=1}^3 a_i e_i$ with a_i denoting the coefficient associated with e_i and where

$$e_1 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad e_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad e_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

The coefficients form an algebraic space isomorphic to the Lie algebra that we will refer to as the Cartesian algebraic space denoted \mathbf{R}_G , where the subscript indicates the corresponding Lie group. Elements in the Cartesian algebraic space can be represented using matrix notation as $v = [a_1, a_2, \dots]^T$. We distinguish elements of the Lie algebra from elements of the Cartesian algebraic space using bold font notation for elements of the Lie algebra.

The wedge, \wedge , and vee, \vee , functions are linear isomorphisms used to map between the Lie algebra and the Cartesian algebraic space. We denote these functions respectively as

$$\begin{aligned} \wedge : \mathbf{R}_G &\rightarrow \mathfrak{g}; & (v) &\mapsto \mathbf{v} \\ \vee : \mathfrak{g} &\rightarrow \mathbf{R}_G; & (\mathbf{v}) &\mapsto v. \end{aligned}$$

For $\mathfrak{se}(2)$ the vee map is defined as

$$\left(\begin{bmatrix} [\omega]_{\times} & \rho \\ 0_{1 \times 2} & 0 \end{bmatrix} \right)^{\vee} = \begin{bmatrix} \rho \\ \omega \end{bmatrix} = v, \quad (5)$$

and the wedge map is the inverse.

B. Exponential Map

For Riemannian manifolds, a geodesic is the shortest path between two points. The exponential function on the Lie group G , denoted $\text{Exp}^G : G \times \mathbf{R}_G \rightarrow G$, is a geodesic that starts at a point $g_1 \in G$ and travels in the direction of a tangent vector $v \in \mathbf{R}_G$ for unit time to the point $g_2 \in G$ as stated in Proposition 2.7 of [25]. We denote the exponential function and its inverse, Log^G , as

$$\text{Exp}^G : G \times \mathbf{R}_G \rightarrow G; (g_1, v) \mapsto g_2$$

$$\text{Log}^G : G \times G \rightarrow \mathbf{R}_G; (g_2, g_1) \mapsto v.$$

When working with Lie groups, it is common to restrict the definition of the exponential map to the identity element of the group; we denote this restriction as

$$\text{Exp}_I^G : \mathbf{R}_G \rightarrow G; (v) \mapsto g_3$$

$$\text{Log}_I^G : G \rightarrow \mathbf{R}_G; (g_3) \mapsto v.$$

Lie groups allow the restricted exponential map to be moved to another element of the group by applying the left or right group action. Using left-trivialization, in other words the left group action, we define the relation

$$\text{Exp}^G(g_1, v) = g_1 \bullet \text{Exp}_I^G(v) = g_2 \quad (8a)$$

$$\text{Log}^G(g_2, g_1) = \text{Log}_I^G(g_1^{-1} \bullet g_2) = v \quad (8b)$$

with \bullet denoting the group operator that we omit in the future. A depiction of the exponential map is given in Fig. 2 where the sphere represents the manifold, the plane represents the tangent space that extends to infinity, the arrow in the tangent space represents v , and the arrow on the manifold represents the geodesic from g_1 to g_2 in the direction of v . For matrix Lie groups, the exponential

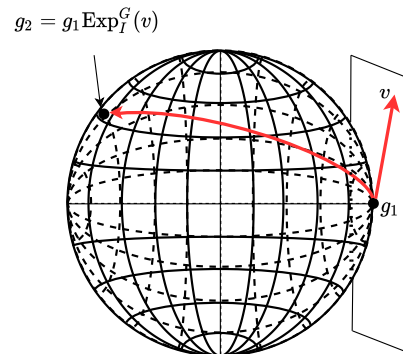


Fig. 2: A depiction of a geodesic starting at g_1 , moving in the direction of v and ending at g_2 .

and logarithm maps at the identity element of the group are the matrix exponential and matrix logarithm. These maps operate on the Lie algebra, but their definitions are

extended to the Cartesian algebraic space using the wedge and vee operations. Unfortunately, not every Lie group is isomorphic to a matrix Lie group, but fortunately most of Lie groups that appear in robotics and control applications are [21]. Thus, the matrix exponential can serve as the exponential map for the majority of the interesting Lie groups.

Let $v = [\rho^\top, \omega]^\top$ and $g = \begin{bmatrix} R & p \\ 0 & 0 \end{bmatrix}$. For SE(2) the matrix logarithm and exponential mappings have closed-form expressions given by

$$\begin{aligned} \text{Exp}_I^{\text{SE}(2)}(v) &= \begin{bmatrix} \text{expm}([\omega]_\times) & D(\omega)\rho \\ 0_{1 \times 2} & 1 \end{bmatrix} \\ \text{Log}_I^{\text{SE}(2)}(g) &= \begin{bmatrix} \text{logm}(R) & D^{-1}(\text{logm}(R))p \\ 0_{1 \times 2} & 0 \end{bmatrix}, \end{aligned}$$

where

$$\text{expm}([\omega]_\times) = \begin{bmatrix} \cos \omega & -\sin \omega \\ \sin \omega & \cos \omega \end{bmatrix} \quad (9)$$

$$\text{logm}(R) = \arctan2(R_{21}, R_{11}) = \theta \quad (10)$$

$$D(\omega) = \frac{\sin \omega}{\omega} I + \frac{1 - \cos \omega}{\omega} [1]_\times \quad (11)$$

$$D^{-1}(\theta) = \frac{\theta \sin \theta}{2(1 - \cos \theta)} I - \frac{\theta}{2} [1]_\times, \quad (12)$$

and where we note that $D(0)$ and $D^{-1}(0)$ are well defined since $\lim_{\omega \rightarrow 0} \sin \omega / \omega = 1$, $\lim_{\omega \rightarrow 0} (1 - \cos \omega) / \omega = 0$, and $\lim_{\theta \rightarrow 0} \theta \sin \theta / (2(1 - \cos \theta)) = 1$. We also note that when ω is small, $D(\omega) \approx I + \frac{1}{2} [\omega]_\times$ and $D^{-1}(\theta) \approx I - \frac{1}{2} [\theta]_\times$.

C. Adjoint

The matrix adjoint of $g \in G$ is a representation of g that acts on \mathbf{R}_G and is generically defined as

$$\text{Ad}_g^G : \mathbf{R}_G \rightarrow \mathbf{R}_G; \quad (v) \mapsto \text{Ad}_g^G v,$$

and represents, for example, a change of coordinates from one location on the manifold to another. A useful property of the adjoint is

$$g \text{Exp}_I^G(v) = \text{Exp}_I^G(\text{Ad}_g^G v) g. \quad (13)$$

For SE(2), the matrix adjoint of $g \in \text{SE}(2)$ is

$$\text{Ad}_g^{\text{SE}(2)} = \begin{bmatrix} R & -[1]_\times p \\ 0_{1 \times 2} & 1 \end{bmatrix}. \quad (14)$$

The matrix adjoint of $v_1 \in \mathbf{R}_G$ is a representation of \mathbf{R}_G that acts on $v_2 \in \mathbf{R}_G$ generically defined as

$$\text{ad}_{v_1}^G : \mathbf{R}_G \rightarrow \mathbf{R}_G; \quad (v_2) \mapsto \text{ad}_{v_1}^G v_2.$$

For $\mathfrak{se}(2)$, the matrix representation of the adjoint is

$$\text{ad}_v^{\text{SE}(2)} = \begin{bmatrix} [w]_\times & -[1]_\times \rho \\ 0_{1 \times 2} & 0 \end{bmatrix}.$$

D. Jacobian of the Matrix Exponential

When working with Lie groups, we need the differential of the exponential and logarithm functions. These differentials are commonly called the right and left Jacobians. The right and left Jacobians and their inverses

are defined to map elements of \mathbf{R}_G to the general linear group (set of invertible matrices) that acts on \mathbf{R}_G . For matrix Lie groups, they are defined as

$$J_r^G(v) = \sum_{n=0}^{\infty} \frac{(-\text{ad}_v^G)^n}{(n+1)!}, \quad J_l^G(v) = \sum_{n=0}^{\infty} \frac{(\text{ad}_v^G)^n}{(n+1)!},$$

$$J_r^{G^{-1}}(v) = \sum_{n=0}^{\infty} \frac{B_n(-\text{ad}_v^G)^n}{n!}, \quad J_l^{G^{-1}}(v) = \sum_{n=0}^{\infty} \frac{B_n(\text{ad}_v^G)^n}{n!},$$

where B_n are the Bernoulli numbers, the subscripts r/l indicate the right and left Jacobian, and the superscript indicates the corresponding Lie group. The derivation of the left and right Jacobians stems from the Baker-Campbell-Hausdorff formula [21, 12]. The right Jacobian has the properties that for any $v \in \mathbf{R}_G$ and any small $\tilde{v} \in \mathbf{R}_G$,

$$\text{Exp}_I^G(v + \tilde{v}) \approx \text{Exp}_I^G(v) \text{Exp}_I^G(J_r^G(v) \tilde{v}) \quad (15a)$$

$$\text{Exp}_I^G(v) \text{Exp}_I^G(\tilde{v}) \approx \text{Exp}_I^G\left(v + J_r^{G^{-1}}(v) \tilde{v}\right). \quad (15b)$$

Similarly for the left Jacobian,

$$\text{Exp}_I^G(v + \tilde{v}) \approx \text{Exp}_I^G(J_l^G(v) \tilde{v}) \text{Exp}_I^G(v)$$

$$\text{Exp}_I^G(\tilde{v}) \text{Exp}_I^G(v) \approx \text{Exp}_I^G\left(v + J_l^{G^{-1}}(v) \tilde{v}\right).$$

For SE(2), the Jacobians have closed form solutions, and the right Jacobian for SE(2) is

$$J_r^{\text{SE}(2)}(v) = \begin{bmatrix} W_r(\omega) & D_r(\omega)\rho \\ 0_{1 \times 2} & 1 \end{bmatrix} \quad (16)$$

where

$$\begin{aligned} W_r(\omega) &= \frac{\cos \omega - 1}{\omega} [1]_\times + \frac{\sin \omega}{\omega} I \\ D_r(\omega) &= \frac{1 - \cos \omega}{\omega^2} [1]_\times + \frac{\omega - \sin \omega}{\omega^2} I, \end{aligned}$$

and where we again note that $W_r(0)$ and $D_r(0)$ are well defined and that for small ω , $W_r(\omega) \approx I - \frac{1}{2} [\omega]_\times$ and $D_r(\omega) \approx \frac{1}{2} [1]_\times$.

E. Direct Product Group

In this paper, we assume that the target has nearly constant velocity, implying that the target's state is modeled as an element of the Lie group formed from the direct product of G and \mathbf{R}_G , denoted $x = (g, v) \in G_x \doteq G \times \mathbf{R}_G$, where the target's pose is expressed as an element of G and its velocity is expressed as an element of \mathbf{R}_G . The operator of this Lie group is inherited from its subgroups G and \mathbf{R}_G . Since \mathbf{R}_G is an algebraic space, it has an Abelian group structure with the group operator being addition. Thus, the group operator and inverse of G_x are

$$\begin{aligned} x_1 \bullet x_2 &= (g_1, v_1) \bullet (g_2, v_2) = (g_1 g_2, v_1 + v_2) \\ x_1^{-1} &= (g_1, v_1)^{-1} = (g_1^{-1}, -v_1). \end{aligned}$$

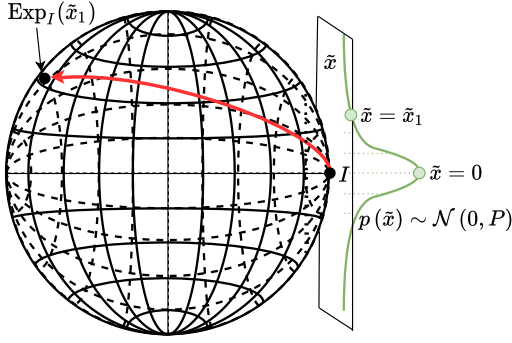


Fig. 3: A depiction of the concentrated Gaussian distribution.

The corresponding Cartesian algebraic space of G_x is $\mathbf{R}_x \doteq \mathbf{R}_G \times \mathbf{R}_G$, and its exponential and logarithmic maps are

$$\text{Exp}_I^{G_x}(u) = \text{Exp}_I^{G_x}((u^g, u^v)) = (\text{Exp}_I^G(u^g), u^v) \quad (17a)$$

$$\text{Log}_I^{G_x}(x) = \text{Log}_I^{G_x}((g, v)) = (\text{Log}_I^G(g), v), \quad (17b)$$

where $u = (u^g, u^v) \in \mathbf{R}_x$. The right Jacobian in matrix notation of G_x is

$$\mathbf{J}_r^{G_x}(u) = \mathbf{J}_r^{G_x}((u^g, u^v)) = \begin{bmatrix} \mathbf{J}_r^G(u^g) & 0 \\ 0 & I \end{bmatrix}. \quad (18)$$

For more information about the direct product group see [26].

F. Uncertainty

In this paper we use Gaussian distributions to model the uncertainty in the sensor, state estimate, and system dynamics. As an example, let $\tilde{x} \in \mathbf{R}_x$ be a zero-mean, Gaussian random variable with covariance P such that $\tilde{x} \sim \mathcal{N}(\mu = 0, P)$; i.e.

$$p(\tilde{x}) = \eta \exp\left(-\frac{1}{2}\tilde{x}^\top P^{-1}\tilde{x}\right) \quad (19)$$

with η denoting the normalizing coefficient. Gaussian distributions are defined on vector spaces. Since not every Lie group has a vector space structure (e.g. SE(2) is not a vector space since scalar multiplication is not defined on the set) Gaussian distributions cannot be defined directly on every Lie group. However, they can be defined on the Cartesian algebraic space at the identity element of the Lie group and extended to the Lie group using the exponential map. Thus, the probability of an element of the Lie group is the probability of the corresponding element of the Cartesian algebraic space. For example, let $x = \text{Exp}_I^{G_x}(\tilde{x})$. The probability of x is determined from the probability of \tilde{x} as depicted in Fig. 3.

In order for the uncertainty to be indirectly defined over the entire Lie group, the Lie group is required to be connected (i.e. there exists a geodesic between any two elements). In other words, the exponential map at identity is surjective allowing the Gaussian distribution to extend to every element of the Lie group.

Depending on the connected Lie group, the exponential map may not be injective, which means that possibly an infinite number of elements of the Cartesian algebraic space will map to the same element of the Lie group. In this case, we require the uncertainty to have a concentrated Gaussian density (CGD) [27]. The CGD is a zero mean Gaussian distribution that is tightly focused around the origin of the Cartesian algebraic space, where by tightly focused we mean that the majority of the probability mass is in a subset $U \subseteq \mathbf{R}_x$ centered around the origin where the exponential mapping from U to G_x is injective, and that the probability of an element not being in U is negligible. This property allows us to ignore the probability of an element being outside of U .

The CGD can be centered at an element other than the identity element of the group using the group action provided that the Lie group is unimodular implying that the determinant of the group action is one. The unimodular property combined with the connected property allows the CGD to be mapped to the Lie group and moved to any element in the group via the group action without changing the probability mass density. For example, the target's state may be represented as

$$x = \hat{x} \text{Exp}_I^{G_x}(\tilde{x}), \quad (20)$$

where $\hat{x} \in G_x$ is the target's state estimate and \tilde{x} is the error state whose probability density function is defined in equation (19). The exponential function at identity maps the random variable \tilde{x} to the Lie group, and the state estimate moves the uncertainty to the target's state without changing the mass density of the uncertainty. These properties allow the probability of the state x to be related to the corresponding probability of the error state \tilde{x} . Thus, the uncertainty distribution of x is defined by the distribution of \tilde{x} as

$$p(x) \triangleq \eta \exp\left(-\frac{1}{2}\text{Log}_I^{G_x}(\hat{x}^{-1}x)^\top P^{-1}\text{Log}_I^{G_x}(\hat{x}^{-1}x)\right) \quad (21a)$$

$$= \eta \exp\left(-\frac{1}{2}\tilde{x}^\top P^{-1}\tilde{x}\right) = p(\tilde{x}). \quad (21b)$$

With a slight abuse of notation, we denote the probability density function (PDF) of the state x as $x \sim \mathcal{N}(\hat{x}, P)$ where \hat{x} is the state estimate and P is the error covariance of the error state \tilde{x} .

An advantage to representing the uncertainty in the Cartesian algebraic space is having a minimum representation of the uncertainty. For example, an element of the matrix group SE(2) has three dimensions but is represented by a 3×3 matrix with nine elements. Representing the uncertainty directly on the set of 3×3 matrices with nine elements would require the covariance to be 9×9 , whereas the corresponding Cartesian algebraic space only has three components and the corresponding covariance matrix will be 3×3 .

For more information about representing uncertainty on Lie groups see [28, 16, 29, 30].

G. First Order Taylor Series and Partial Derivatives

Let $g, \hat{g} \in G$, and let $\tilde{g} \in \mathbf{R}_G$ be a small perturbation from the origin with the relation $g = \hat{g} \text{Exp}_I^G(\tilde{g})$. Also let $f : G \rightarrow G$ be an arbitrary function. The first order Taylor series of f evaluated at \hat{g} is

$$f(g) \approx f(\hat{g}) \text{Exp}_I^G\left(\frac{\partial f}{\partial g}(\hat{g}) \tilde{g}\right),$$

where $\frac{\partial f}{\partial g}(\hat{g})$ is the partial derivative of f with respect to g evaluated at $g = \hat{g}$. Using the definition and notation shown in [13] and the relation defined in (8), the partial derivative of f with respect to g is defined as

$$\frac{\partial f}{\partial g} = \lim_{v \rightarrow 0} \frac{\text{Log}_I^G\left(f(g)^{-1} f(g \text{Exp}_I^G(v))\right)}{v}, \quad (22)$$

where $v \in \mathbf{R}_G$. In Equations (22), we abuse notation by denoting the vector of the numerator divided by each element of v , as the numerator divided by the vector v . Note that the limit in equation (22) is taken in the Cartesian algebraic space instead of the Lie group, since the Cartesian algebraic space is a vector space where derivatives are well defined.

III. Overview

The Lie group integrated probabilistic data association filter (LG-IPDAF) is designed to track a single dynamic target using a single sensor that observes the target. The target is modeled using a constant-velocity, white-noise-driven system model defined on Lie groups. The system model is defined in Section IV.

We call the act of the sensor observing and producing measurements from the measurement space at a given instant of time a *sensor scan*. The sensor is assumed to detect the target with probability $P_D \in [0, 1]$ where $P_D = 0$ means that the target is not in the sensor's field of view. It is assumed that a sensor produces at most one target originated measurement, called a true measurement. We also assume that every sensor scan and all other measurements are non-target-originated measurements, called false measurements or clutter. The false measurements are assumed to be independent identically distributed (iid) with uniform spatial distribution, where the number of false measurements per sensor scan is modeled by the Poisson distribution

$$\mu_F(\phi) = \exp(\lambda \mathcal{V}) \frac{(\lambda \mathcal{V})^\phi}{\phi!}, \quad (23)$$

where λ is the spatial density of false measurements, \mathcal{V} is the volume of the sensor's field of view, and ϕ is the number of false measurements.

The LG-IPDAF represents the target mathematically using tracks. A track is a tuple $\mathcal{T} = (\hat{x}, P, \mathcal{CS}, \epsilon, L)$, where \hat{x} is the state estimate of a target, P is the corresponding error covariance, \mathcal{CS} is the set of measurements associated to the track called the consensus set, ϵ denotes the probability that the track represents a target and is called

the track likelihood, and L is the track label, a unique numerical label used to identify confirmed tracks.

The LG-IPDAF does not assume that the target's initial state is known and relies on a track initialization and confirmation scheme to locate the target. Tracks are initialized from measurements. Since the sensor produces both target-originated measurements and false measurements, tracks generated by the LG-IPDAF can be created from true measurements and/or false measurements. Therefore, an initialized track can either represent a target or clutter. To identify the track that represents the target, the LG-IPDAF calculates the track likelihood. A track with a high track likelihood is confirmed to be a good representation of the target, and the confirmed track with the highest track likelihood is assumed to represent the target.

As time progresses, the target moves and the sensor produces measurements. When new measurements are received, the LG-IPDAF algorithm performs four steps in order: (1) the prediction step, (2) the data association step, (3) the update step, and (4) the track initialization step.

Let t_k denote the current time and t_{k-} denote the time at the previous iteration. At the beginning of the prediction step, the track's state estimate, error covariance and track likelihood are at time t_{k-} and conditioned on the measurements up to time t_{k-} . We denote these values respectively as $\hat{x}_{k-|k-}$, $P_{k-|k-}$, and $\epsilon_{k-|k-}$. During the prediction step the track's state estimate, error covariance and track likelihood are propagated forward in time using the system model. The propagated state estimate, error covariance and track likelihood at time t_k , conditioned on the measurement up to time t_{k-} , are denoted $\hat{x}_{k|k-}$, $P_{k|k-}$, and $\epsilon_{k|k-}$ respectively. The prediction step is discussed in detail in Section V.A.

During the data association step, the new measurements are associated to an existing track or given to a database that contains non-track-associated measurements. The data association algorithm is designed so that a target-originated measurement is associated to the track that represents the target with probability $P_G \in [0, 1]$ provided that the track exists. This probability is used to construct a volume in the measurement space centered around the track's estimated measurement called the validation region. Any measurement that falls within a track's validation region is associated to the track. We denote the set of measurements associated to a track at time t_k as $Z_k = \{z_{k,j}\}_{j=1}^{m_k}$ where $z_{k,j}$ denotes the j^{th} measurement at time t_k associated with the track and m_k denotes the number of measurements associated with the track at time t_k . The data association step is discussed in detail in Section V.B.

During the update step, the new measurements associated with a track are used to update the track. For every associated measurement $\{z_{k,j}\}_{j=1}^{m_k}$, the track's state estimate $\hat{x}_{k|k-}$ is copied and updated using a distinct associated measurement to produced what we define to

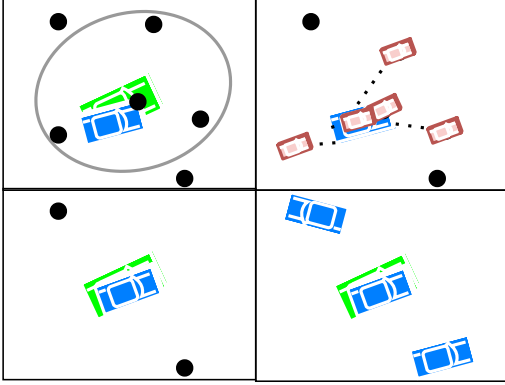


Fig. 4: A depiction of a single iteration of the LG-IPDAF. The large green car represents the target, the smaller blue cars represent the tracks, the black dots represent measurements, the gray ellipse represents the validation region and the small red cars represent the split state estimates. The top left image depicts the prediction and data association steps during which new measurements are received, the track is propagated in time, the validation region is constructed, and four measurements are associated to the track. The top right image shows the first part of the update step where the state estimate is split into five states: one for each associated measurement, and one for the null hypothesis that none of the measurements are target originated. The bottom left shows the rest of the update step where the split state estimates are fused together to form a single state estimate. The bottom right image shows the last step which initializes new tracks from non-track-associated measurements.

be the *split state estimates* $\{\hat{x}_{k|k,j}\}_{j=0}^{m_k}$ where $\hat{x}_{k|k,j}$ for $j > 0$ is the state estimate after being updated with measurement $z_{k,j}$ and $\hat{x}_{k|k,0} = \hat{x}_{k|k-}$ is the null hypothesis that none of the measurements originated from the target. Then the split state estimates $\{\hat{x}_{k|k,j}\}_{j=0}^{m_k}$ are fused together according to their probability of being the correct state estimate. In this way, the LG-IPDAF is never totally correct but never completely wrong as in the case of hard data association algorithms. After the update step, the propagated state estimate, error covariance and track likelihood at time t_k , conditioned on the measurements up to time t_k , are denoted $\hat{x}_{k|k}$, $P_{k|k}$, and $\epsilon_{k|k}$ respectively. The update step is discussed in detail in Section V.C.

During the track initialization step, non-track-associated measurements are used to initialize new tracks. This step is discussed in detail in Section V.D. A depiction of a single iteration of the LG-IPDAF is shown in Fig. 4.

A. Assumptions

Our derivation of the LG-IPDAF uses the following assumptions:

- 1) There exists a single target that can be observed by a single sensor and modeled by a constant-velocity, white-noise-driven system model defined in equation (24).
- 2) A sensor scan occurs whenever the sensor observes the measurement space. At every sensor scan there are m_k validated measurements denoted $\{z_{k,j}\}_{j=1}^{m_k} = Z_k$.
- 3) At every scan there is at most one target originated measurement and all others are false (i.e. non target-originated measurements).
- 4) The sensor detects the target with probability $P_D \in [0, 1]$.
- 5) The target originated measurement falls within the track's validation region with probability P_G provided that the track represents the target. The probability P_G is discussed in Section V.B.
- 6) The false measurements are independently identically distributed (iid) with uniform spatial density λ .
- 7) The expected number of false measurements per sensor scan is modeled using the density function μ_F . In this paper, μ_F denotes a Poisson distribution defined in equation (23).
- 8) The past information about a track is summarized as $(\hat{x}_{k-|k-}, P_{k-|k-}, \epsilon_{k-|k-})$ where $\hat{x}_{k-|k-}$, $P_{k-|k-}$, and $\epsilon_{k-|k-}$ denote the track's state estimate, error covariance and track likelihood at the previous time and conditioned on the previous track-associated measurements.

The parameters P_D and λ can be statistically calculated from sensor data, and the parameter P_G is selected by the user and discussed in Section V.B.

IV. System Model

In this section we present the system model used in the LG-IPDAF and also derive the affinization of the system model that will be necessary when approximating Gaussian distributions.

A. System Model

Let $x_k = (g_k, v_k) \in G_x \doteq G \times \mathbf{R}_G$ denote the target's state at time t_k , t_{k-} denote the time at the previous iteration, $t_\Delta = t_k - t_{k-}$ denote the time interval from the previous iteration to the current time, $\mathbf{R}_x \doteq \mathbf{R}_G \times \mathbf{R}_G \ni q_\Delta = (q_\Delta^g, q_\Delta^v) \sim \mathcal{N}(0, Q(t_\Delta))$ denote the process noise modeled as a Wiener process [31], $z_k \in G_s$ denote the measurement at time t_k , $\mathbf{R}_s \ni r_k \sim \mathcal{N}(0, R)$ denote the measurement noise, where G_s is the Lie group for the measurement space and \mathbf{R}_s is the corresponding Cartesian algebraic space. The proposed discrete, time-invariant model is

$$x_k = f(x_{k-}, q_\Delta, t_\Delta) \quad (24a)$$

$$z_k = h(x_k, r_k), \quad (24b)$$

where f is the state transition function defined as

$$f(x_{k-}, q_{\Delta}, t_{\Delta}) \triangleq (g_{k-}, v_{k-}) \text{Exp}_I^{G_x}(t_{\Delta} v_{k-} + q_{\Delta}^g, q_{\Delta}^v) \quad (25a)$$

$$= (g_{k-} \text{Exp}_I^G(t_{\Delta} v_{k-} + q_{\Delta}^g), v_{k-} + q_{\Delta}^v). \quad (25b)$$

This form is similar to the system model defined in [32].

The definition of the observation function h is dependent on the application and further generalization is not needed. In Section VI, we show how h is defined when the target of interest is a car restricted to a plane, and where its pose (position and orientation) is observed. In that case, the state is the Lie group $\text{SE}(2) \times \mathbf{R}_{\text{SE}(2)}$ and the measurement Lie group is $\text{SE}(2)$, where the observation function is

$$h(x_k, r_k) = g_k \text{Exp}_I^{G_{\text{SE}(2)}}(r_k), \quad (26)$$

and where the noise satisfies $r_k \sim \mathcal{N}(0, R) \in \mathbf{R}_{\text{SE}(2)}$.

B. System Affinization

A Gaussian random variable applied to an affine function remains Gaussian while a Gaussian random variable applied to a nonaffine function is no longer Gaussian. Depending on the Lie group, the system model can be affine or nonaffine. In the nonaffine case and under the assumption that the signal-to-noise ratio (SNR) is high, the Gaussian structure of the system model uncertainties can be well preserved with little loss of information by affinizing the system model when propagating and updating the uncertainty as is commonly done with the extended Kalman filter.

The system model is approximated as affine by computing its first order Taylor series at the points

$$\zeta_{f_{\Delta}} \doteq (x_{k-} = \hat{x}_{k-}, q_{\Delta} = 0, t_{\Delta}) \quad (27)$$

$$\zeta_{h_k} \doteq (x_k = \hat{x}_k, r_k = 0), \quad (28)$$

according to Subsection II.G. What we mean by affine, is that the propagation of the uncertainty is affine. This computation requires the calculation of the state transition function Jacobians and the observation function Jacobians.

The Jacobians for the state transition function can be defined generically, and the Jacobians for the observation function are application dependent. However, we will define the observation function Jacobians for our running example.

LEMMA 1. *Given the discrete time-invariant model in Equations (24), (25) and (26) the Jacobians evaluated*

at $\zeta_{f_{\Delta}}$ and ζ_{h_k} are

$$F_{\Delta} = \left. \frac{\partial f}{\partial x} \right|_{\zeta_{f_{\Delta}}} = \begin{bmatrix} \text{Ad}_{\text{Exp}_I^G(t_{\Delta} \hat{v}_{k-})}^{-1} & \text{J}_r^G(t_{\Delta} \hat{v}_{k-}) t_{\Delta} \\ 0_{n \times n} & I_{n \times n} \end{bmatrix} \quad (29a)$$

$$G_{\Delta} \doteq \left. \frac{\partial f}{\partial q} \right|_{\zeta_{f_{\Delta}}} = \begin{bmatrix} \text{J}_r^G(t_{\Delta} \hat{v}_i) & 0_{n \times n} \\ 0_{n \times n} & I_{n \times n} \end{bmatrix} \quad (29b)$$

$$H_k = \left. \frac{\partial h}{\partial x} \right|_{\zeta_{h_k}} = \begin{bmatrix} I_{n \times n} & 0_{n \times n} \end{bmatrix} \quad (29c)$$

$$V_k = \left. \frac{\partial h}{\partial r} \right|_{\zeta_{h_k}} = I_{n \times n}, \quad (29d)$$

where n is the dimension of the target's pose, $0_{n \times n}$ is the $n \times n$ zero matrix, and $I_{n \times n}$ is the $n \times n$ identity matrix.

Consequently, if $\hat{x}_{k-} \in G_x$ is a state estimate that is close to x_{k-} , and if the error state between x_{k-} and \hat{x}_{k-} is defined as $\tilde{x}_{k-} \doteq \text{Log}_I^{G_x}(\hat{x}_{k-}^{-1} x_{k-}) \in \mathbf{R}_x$, then the evolution of the system can be described by the ‘‘affinized system’’

$$x_k \approx f(\hat{x}_{k-}, 0, t_{\Delta}) \text{Exp}_I^{G_x}(F_{\Delta} \tilde{x}_{k-} + G_{\Delta} q_{\Delta}) \quad (30a)$$

$$z_k \approx h(\hat{x}_k, 0) \text{Exp}_I^{G_s}(H_k \tilde{x}_k + V_k r_k). \quad (30b)$$

Proof:

We will prove the expression for F_{Δ} , and leave the derivation of the other Jacobians, which are similar, to the reader. Let $\tau = (\tau^g, \tau^v) \in \mathbf{R}_x$ denote the perturbation of the state. Then, using the definition of the derivative in equation (22)

$$\left. \frac{\partial f}{\partial x_{k-}} \right|_{\tau \rightarrow 0} = \lim_{\tau \rightarrow 0} \frac{\text{Log}_I^{G_x}(f(x_{k-}, 0, t_{\Delta})^{-1} f(x_{k-} \text{Exp}_I^{G_x}(\tau), 0, t_{\Delta}))}{\tau}.$$

Substituting the definition of the state transition function in equation (25) yields

$$\begin{aligned} \left. \frac{\partial f}{\partial x_{k-}} \right|_{\tau \rightarrow 0} &= \lim_{\tau \rightarrow 0} \frac{1}{\tau} \text{Log}_I^{G_x} \left[(g_{k-} \text{Exp}_I^G(t_{\Delta} v_{k-}), v_{k-})^{-1} \right. \\ &\quad \left. (g_{k-} \text{Exp}_I^G(\tau^g) \text{Exp}_I^G(t_{\Delta} v_{k-} + t_{\Delta} \tau^v), v_{k-} + \tau^v) \right] \\ &= \lim_{\tau \rightarrow 0} \frac{1}{\tau} \text{Log}_I^{G_x} \left[(\text{Exp}_I^G(t_{\Delta} v_{k-})^{-1} g_{k-}^{-1} g_{k-} \text{Exp}_I^G(\tau^g) \right. \\ &\quad \left. \text{Exp}_I^G(t_{\Delta} v_{k-} + t_{\Delta} \tau^v), v_{k-} + \tau^v - v_{k-}) \right] \\ &= \lim_{\tau \rightarrow 0} \frac{1}{\tau} \text{Log}_I^{G_x} \left[(\text{Exp}_I^G(t_{\Delta} v_{k-})^{-1} \text{Exp}_I^G(\tau^g) \right. \\ &\quad \left. \text{Exp}_I^G(t_{\Delta} v_{k-} + t_{\Delta} \tau^v), \tau^v) \right] \end{aligned}$$

Using the property of the adjoint in equation (13) and the property of the right Jacobian in equation (15) gives

$$\begin{aligned} \left. \frac{\partial f}{\partial x_{k-}} \right|_{\tau \rightarrow 0} &= \lim_{\tau \rightarrow 0} \frac{1}{\tau} \text{Log}_I^{G_x} \left[(\text{Exp}_I^G(t_{\Delta} v_{k-})^{-1} \text{Exp}_I^G(\tau^g) \right. \\ &\quad \left. \text{Exp}_I^G(t_{\Delta} v_{k-}) \text{Exp}_I^G(\text{J}_r^G(t_{\Delta} v_{k-}) t_{\Delta} \tau^v), \tau^v) \right] \\ &= \lim_{\tau \rightarrow 0} \frac{1}{\tau} \text{Log}_I^{G_x} \left[\left(\text{Exp}_I^G(\text{Ad}_{\text{Exp}_I^G(t_{\Delta} v_{k-})}^{-1} \tau^g) \right. \right. \\ &\quad \left. \left. \text{Exp}_I^G(\text{J}_r^G(t_{\Delta} v_{k-}) t_{\Delta} \tau^v), \tau^v \right) \right] \\ &= \lim_{\tau \rightarrow 0} \frac{1}{\tau} \left(\text{Log}_I^{G_x} \left(\text{Exp}_I^G(\text{Ad}_{\text{Exp}_I^G(t_{\Delta} v_{k-})}^{-1} \tau^g) \right. \right. \\ &\quad \left. \left. \text{Exp}_I^G(\text{J}_r^G(t_{\Delta} v_{k-}) t_{\Delta} \tau^v), \tau^v \right) \right) \end{aligned}$$

The portion of the derivative corresponding to τ^g is computed as

$$\begin{aligned} \frac{\partial f}{\partial g_{k^-}} &= \lim_{\tau^g \rightarrow 0} \frac{\left(\text{Log}_I^G \left(\text{Exp}_I^G \left(\text{Ad}_{\text{Exp}_I^G(t_\Delta v_{k^-})}^{-1} \tau^g \right) \right), 0 \right)}{\tau^g} \\ &= \lim_{\tau^g \rightarrow 0} \frac{\left(\text{Ad}_{\text{Exp}_I^G(t_\Delta v_{k^-})}^{-1} \tau^g, 0 \right)}{\tau^g} = \begin{bmatrix} \text{Ad}_{\text{Exp}_I^G(t_\Delta v_{k^-})}^{-1} \\ 0_{n \times n} \end{bmatrix}, \end{aligned} \quad (31)$$

and the portion of the derivative corresponding to τ^v is computed as

$$\begin{aligned} \frac{\partial f}{\partial v_{k^-}} &= \lim_{\tau^v \rightarrow 0} \frac{\left(\text{Log}_I^G \left(\text{Exp}_I^G \left(J_r^G(t_\Delta v_{k^-}) t_\Delta \tau^v \right) \right), \tau^v \right)}{\tau^v} \\ &= \lim_{\tau^v \rightarrow 0} \frac{\left(J_r^G(t_\Delta v_{k^-}) t_\Delta \tau^v, \tau^v \right)}{\tau^v} = \begin{bmatrix} J_r^G(t_\Delta v_{k^-}) t_\Delta \\ I_{n \times n} \end{bmatrix}. \end{aligned} \quad (32)$$

Combining equations (31) and (32) yields (29a). ■

V. The Lie Group Integrated Probabilistic Data Associate Filter

This section gives detailed derivation of the key elements of the LG-IPDA filter. In particular, the prediction step in Section V.A, the data association step in Section V.B, the measurement update step in Section V.C, and new track initialization in Section V.D.

A. Prediction Step

The prediction step of the LG-IPADF is similar to the prediction step of the indirect Kalman filter with the addition of propagating the track likelihood [33, 32]. We begin by introducing additional notation. Let $Z_{0:i}$ denote the set of track associated measurements from the initial time to time t_i . Let ϵ denote a Bernoulli random variable that represents the probability that the track represents the target. The probability that the track represents a target conditioned on $Z_{0:k^-}$ is the track likelihood and is denoted at the previous time step as $\epsilon_{k^-|k^-} \doteq p(\epsilon_{k^-} | Z_{0:k^-})$. Conversely, we denote the probability that the track does not represent a target conditioned on $Z_{0:k^-}$ as $p(\epsilon_{k^-} = F | Z_{0:k^-}) = 1 - p(\epsilon_{k^-} | Z_{0:k^-})$.

Define the probability of the track's previous state conditioned on the previous track-associated measurements and it representing a target as

$$p(x_{k^-} | \epsilon_{k^-}, Z_{0:k^-}) \doteq \eta \exp \left(-\frac{1}{2} \tilde{x}_{k^-|k^-}^\top P_{k^-|k^-}^{-1} \tilde{x}_{k^-|k^-} \right), \quad (33)$$

where $\tilde{x}_{k^-|k^-} = \text{Log}_I^{G_x} \left(\hat{x}_{k^-|k^-}^{-1} x_{k^-} \right)$ is the error state, $\hat{x}_{k^-|k^-}$ is the state estimate and $P_{k^-|k^-}$ is the error covariance at time t_{k^-} conditioned on the measurements $Z_{0:k^-}$. The terms $\hat{x}_{k^-|k^-}$, $P_{k^-|k^-}$, and $\epsilon_{k^-|k^-}$ are the track's state estimate, error covariance and track likelihood at the beginning of the propagation step.

To derive the prediction step, we need to construct the Gaussian approximation of the probability of the track's current state conditioned on the track's previous state, the track representing the target, and the previous measurements denoted $p(x_k | x_{k^-}, \epsilon_{k^-}, Z_{0:k^-})$.

LEMMA 2. *Given that Assumptions 1 and 8 hold, then the Gaussian approximation of $p(x_k | x_{k^-}, \epsilon_{k^-}, Z_{0:k^-})$ is*

$$\begin{aligned} &p(x_k | x_{k^-}, \epsilon_{k^-}, Z_{0:k^-}) \\ &\approx \eta \exp \left(-\frac{1}{2} \left(\tilde{x}_{k|k^-} - F_\Delta \tilde{x}_{k^-|k^-} \right)^\top \bar{Q}_\Delta \left(\tilde{x}_{k|k^-} - F_\Delta \tilde{x}_{k^-|k^-} \right) \right), \end{aligned} \quad (34)$$

where

$$\begin{aligned} \tilde{x}_{k|k^-} &= \text{Log}_I^{G_x} \left(\hat{x}_{k|k^-}^{-1} x_k \right) \\ \hat{x}_{k|k^-} &= f \left(\hat{x}_{k^-|k^-}, 0, t_\Delta \right) \\ \bar{Q}_\Delta &= G_\Delta Q(t_\Delta) G_\Delta^\top, \end{aligned} \quad (35)$$

and where f is the state transition function defined in equation (25), $Q(t_\Delta)$ is the process noise covariance, and the Jacobians F_Δ and G_Δ are defined in equation (29) and evaluated at the point $\zeta_{f_\Delta} = (\hat{x}_{k^-|k^-}, 0, t_\Delta)$.

Proof:

The probability $p(x_k | x_{k^-}, \epsilon_{k^-}, Z_{0:k^-})$ is approximated as Gaussian by computing the first and second moments of the propagated error state, using the affinized system defined in (30). From Equation (30) the affinized state transition function is

$$x_k \approx \underbrace{f \left(\hat{x}_{k^-|k^-}, 0, t_\Delta \right)}_{\hat{x}_{k|k^-}} \text{Exp}_I^{G_x} \left(\underbrace{F_\Delta \tilde{x}_{k^-|k^-} + G_\Delta q_\Delta}_{\tilde{x}_{k|k^-}} \right), \quad (36)$$

where x_k , $\hat{x}_{k|k^-} = f(\hat{x}_{k^-|k^-}, 0, t_\Delta)$, $\tilde{x}_{k|k^-} = F_\Delta \tilde{x}_{k^-|k^-} + G_\Delta q_k$ are respectively the predicted state, the predicted state estimate, and the predicted error state all conditioned on the track's previous state, the track representing the target, and the previous measurements.

Since the probability $p(x_k | x_{k^-}, \epsilon_{k^-}, Z_{0:k^-})$ is conditioned on a specific value of the previous state, the previous error state $\tilde{x}_{k^-|k^-}$ is not a random variable. Thus, the first and second moments of the propagated error state are

$$\begin{aligned} \text{E}[\tilde{x}_{k|k^-}] &= \text{E}[F_\Delta \tilde{x}_{k^-|k^-} + G_\Delta q_\Delta] \\ &= F_\Delta \tilde{x}_{k^-|k^-} \\ \text{cov}[\tilde{x}_{k|k^-}] &= G_\Delta Q(t_\Delta) G_\Delta^\top = \bar{Q}_\Delta, \end{aligned}$$

and the approximate Gaussian PDF is given in equation (34). ■

LEMMA 3. *Suppose that Assumptions 1 and 8 hold, then the propagation of the track's state estimate and error covariance are*

$$\hat{x}_{k|k^-} = f \left(\hat{x}_{k^-|k^-}, 0, t_\Delta \right), \quad (37a)$$

$$P_{k|k^-} = F_\Delta P_{k^-|k^-} F_\Delta^\top + G_\Delta Q(t_\Delta) G_\Delta^\top \quad (37b)$$

and the probability of the track's current state conditioned on the track representing the target and the previous measurements is

$$p(x_k | \epsilon_{k-}, Z_{0:k-}) = \eta \exp\left(-\frac{1}{2}\tilde{x}_{k|k-}^\top P_{k|k-}^{-1}\tilde{x}_{k|k-}\right) \quad (38)$$

where $\tilde{x}_{k|k-} = \text{Log}_I^{G_x}(\hat{x}_{k|k-}^{-1}x_k)$.

Proof: See Appendix A.

The track likelihood is modeled using a Markov process as described in [17], and its propagation is

$$\epsilon_{k|k-} \doteq p(\epsilon_k | Z_{0:k-}) = \sigma(t_\Delta) p(\epsilon_{k-} | Z_{0:k-}),$$

where the probability $\sigma(t_\Delta) \in [0, 1]$ is chosen to represent how the track likelihood can change in time due the target being occluded, leaving the sensor surveillance region, etc.

Therefore, at the end of the prediction step, we know the state estimate $\hat{x}_{k|k-}$ and error covariance and $P_{k|k-}$, and the probabilities $p(x_k | \epsilon_{k-}, Z_{0:k-})$ and $p(\epsilon_k | Z_{0:k-})$.

B. Data Association

In this section we derive the data association algorithm that associates new measurements to tracks. Let ψ denote a Bernoulli random variable that indicates that the measurement originated from the target. The estimated measurement of a track with state estimate $\hat{x}_{k|k-}$ is

$$\hat{z}_k \triangleq h(\hat{x}_{k|k-}, 0), \quad (39)$$

where h is the generic observation function defined in equation (24).

The validation region is a volume in measurement space centered around the track's estimated measurement. The volume is selected such that a target-originated measurement has probability P_G to fall within the track's validation region provided that the track represents the target. A measurement that falls within the validation region of a track is called a validated measurement and is associated to the track. Otherwise, the measurement is given to a database that stores non-track-associated measurements. Computation of the validation region is complicated by the fact that the measurements and the target's state are elements of Lie groups and do not have a vector space structure.

LEMMA 4. *Suppose that Assumptions 1 and 8 hold, then the probability of measurement z_k conditioned on it being target-originated, is given by*

$$p(z_k | \psi, \epsilon_k, Z_{0:k-}) \approx \eta \exp\left(-\frac{1}{2}\tilde{z}_k^\top S_k^{-1}\tilde{z}_k\right), \quad (40)$$

where the innovation \tilde{z}_k and the innovation covariance S_k are given by

$$\hat{z}_k = h(\hat{x}_{k|k-}, 0) \quad (41a)$$

$$\tilde{z}_k = \text{Log}_I^{G_z}(\hat{z}_k^{-1}z_k) \quad (41b)$$

$$S_k = V_k R V_k^\top + H_k P_{k|k-} H_k^\top, \quad (41c)$$

and where the Jacobians H_k and V_k are defined in (29) and evaluated at the point $\zeta_{h_k} = (\hat{x}_{k|k-}, 0)$.

Proof: Similar to the proof of Lemma 3 in Appendix A.

Using equations (40) and (41), we define the metric $d_V : G_s \times G_s \rightarrow \mathbb{R}$ as

$$d_V(z_k, \hat{z}_k) = \nu_k^\top S_k^{-1} \nu_k, \quad (42)$$

where the innovation covariance is used to normalize the metric. Thus, the metric d_V is the sum of m squared Gaussian random variables where m is the dimension of the measurement space, and the values of the metric are distributed according to a chi-square distribution with m degrees of freedom.

The validation region is defined as the set

$$\text{val}(\hat{z}_k, \tau_G) \doteq \{z \in G_s \mid d_V(z, \hat{z}_k) \leq \tau_G\},$$

where the parameter τ_G is called the gate threshold. A measurement that falls within this validation region is associated to the track whose estimated measurement is \hat{z}_k . This implies that a validation region is constructed for each track.

The volume of the validation region is defined as

$$\mathcal{V}_k = c_m |\tau_G S_k|^{1/2}, \quad (43)$$

where c_m is the volume of the unit hypersphere of dimension m calculated as

$$c_m = \frac{\pi^{m/2}}{\Gamma(m/2 + 1)},$$

with Γ denoting the gamma function [4]. It is worth noting that the volume of the validation region is dependent on the error covariance through the innovation covariance S_k . Therefore, the validation region contains information on the quality of the state estimate. This concept will be used in Section V.C.

The gate probability is

$$P_G = \int_{\mathcal{V}_k} p(z | \psi, \epsilon_k, Z_{0:k-}) dz, \quad (44)$$

and is the value of the chi-square cumulative distribution function (CDF) with parameter τ_G [4]. Let θ_k denote a Bernoulli random variable that the measurement is target originated and inside the validation region. Using (40) and (44), the probability of a measurement conditioned on the measurement being inside the validation region, being target-originated, the track representing the target and the previous measurements is

$$p(z_k | \theta_k, \epsilon_k, Z_{0:k-}) = P_G^{-1} p(z_k | \psi, \epsilon_k, Z_{0:k-}). \quad (45)$$

The data association step of the LG-IPDAF can be extended to multiple targets and tracks by assigning a measurement to every track whose validation region the measurement falls in without taking into account joint associations. This is done by copying the measurement for every track it is associated with and giving a copy to each track the measurement is associated with and treating track associated measurements of one track independently of track associated measurements of another track.

At the end of the data association step, we have the track-associated measurements $Z_k = \{z_{k,j}\}_{j=1}^{m_k}$ for each track.

C. Measurement Update Step

According to Assumption 3, at most one measurement originates from the target every sensor scan. Thus, given the set of new validated measurements $Z_k = \{z_{k,j}\}_{j=1}^{m_k}$, either one of the measurements is target originated or none of them are. This leads to different possibilities. As an example, all of the measurements could be false, or the measurement $z_{k,1}$ could be the target-originated measurement and all others false, or $z_{k,2}$ could be the target-originated measurement and all others false, etc. These different possibilities are referred to as association events denoted $\theta_{k,j}$ where the subscript $j > 0$ means that the j^{th} validated measurement is target originated and all others are false, and where $j = 0$ means that all of the validated measurements are false. Hence, there are a total of $m_k + 1$ association events.

LEMMA 5. *Suppose that Assumptions 1-8 hold, and that $Z_k = \{z_{k,j}\}_{j=1}^{m_k}$ is the set of validated measurements at time t_k , and let P_D be the probability of detection, P_G the gate probability, λ the spatial density of false measurements, $p(z_{k,j} | \psi, \epsilon_k, Z_{0:k-})$ be defined by Equation (40), and let*

$$\mathcal{L}_{k,j} \doteq \frac{P_D}{\lambda} p(z_{k,j} | \psi, \epsilon_k, Z_{0:k-}), \quad (46)$$

then the probability $\beta_{k,j} \doteq p(\theta_{k,j} | \epsilon_k, Z_{0:k})$ of an association event $\theta_{k,j}$ conditioned on the measurements $Z_{0:k}$, and the track representing a target, is given by

$$\beta_{k,j} = \begin{cases} \frac{(1-P_D P_G)}{1-P_D P_G + \sum_{j=1}^{m_k} \mathcal{L}_{k,j}} & j = 0, \\ \frac{\mathcal{L}_{k,j}}{1-P_D P_G + \sum_{i=1}^{m_k} \mathcal{L}_{k,i}} & j = 1, \dots, m_k. \end{cases} \quad (47)$$

Proof: See Appendix B.

Using the association events, we can update the track's state estimate $\hat{x}_{k|k-}$ and error covariance $P_{k|k-}$ conditioned on each association event in order to generate the split tracks. This update step is similar to the Kalman filter's update step. The standard Kalman filter algorithm updates the state estimate directly using vector space arithmetic. However, since not every Lie group has a vector space structure, we have to approach the update step differently.

Recall that $p(x_k | \epsilon_k, Z_{0:k-}) = p(\tilde{x}_k | \epsilon_k, Z_{0:k-})$ with the relation $x_k = \hat{x}_{k|k-} \text{Exp}_I^{G_x}(\tilde{x}_{k|k-})$. The error state $\tilde{x}_{k|k-} \sim \mathcal{N}(\mu_{k|k-}, P_{k|k-})$ has a vector space structure. This allows us to update the error state using a method similar to the Kalman filter. We denote the updated error state and its corresponding mean and error covariance conditioned on $\theta_{k,j}$, the track representing the target, and all track-associated measurements as $\tilde{x}_{k|k,j}^- \sim \mathcal{N}(\mu_{k|k,j}^-, P_{k|k,j}^-)$. Using the updated error state, the probability of the track's state conditioned on the association event $\theta_{k,j}$ and the track representing the target and all the track associated measurements, is $p(x_k | \theta_{k,j}, \epsilon_k, Z_{0:k}) = p(\tilde{x}_k^- | \theta_{k,j}, \epsilon_k, Z_{0:k})$ where $x_k = \hat{x}_{k|k-} \text{Exp}_I^{G_x}(\tilde{x}_{k|k}^-)$.

The updated error state may have a non-zero mean. To reset the error state's mean to zero we add $\mu_{k|k,j}^-$ to the state estimate $\hat{x}_{k|k-}$ via the exponential map, and the error covariance is modified accordingly. We denote the updated and reset error state and its corresponding mean and error covariance conditioned on $\theta_{k,j}$ as $\tilde{x}_{k|k,j} \sim \mathcal{N}(\mu_{k|k,j} = 0, P_{k|k,j}^c)$. Using the reset error state, we have $p(x_k | \theta_{k,j}, \epsilon_k, Z_{0:k}) = p(\tilde{x}_k | \theta_{k,j}, \epsilon_k, Z_{0:k})$ with the relation $x_k = \hat{x}_{k|k} \text{Exp}_I^{G_x}(\tilde{x}_{k|k})$. In summary, we update the mean of the error state, and then reset it to zero by adding it onto the state estimate. This process is the update step of the indirect Kalman filter and is presented in the following lemma.

LEMMA 6. *Suppose that Assumptions 1, 3, and 8 hold, and define $Z_k = \{z_{k,j}\}_{j=1}^{m_k}$ as the set of validated measurements at time k , then the Gaussian approximation of the probability of the split track $x_{k,j}$ is*

$$p(x_{k,j} | \theta_{k,j}, \epsilon_k, Z_{0:k}) = \eta \exp\left(\tilde{x}_{k|k,j}^\top \left(P_{k|k,j}^c\right)^{-1} \tilde{x}_{k|k,j}\right), \quad (48)$$

where

$$\tilde{x}_{k|k,j} = \text{Log}_I^{G_x}\left(\hat{x}_{k|k,j}^{-1} x_{k,j}\right) \quad (49)$$

$$\hat{x}_{k|k,j} = \begin{cases} \hat{x}_{k|k-} \text{Exp}_I^{G_x}\left(\mu_{k|k,j}^-\right), & j \neq 0 \\ \hat{x}_{k|k-}, & j = 0 \end{cases} \quad (50)$$

$$P_{k|k,j}^c = \begin{cases} J_r^{G_x}\left(\mu_{k|k,j}^-\right) P_{k|k}^- J_r^{G_x}\left(\mu_{k|k,j}^-\right)^\top, & j \neq 0 \\ P_{k|k-}, & j = 0 \end{cases} \quad (51)$$

$$\mu_{k|k,j}^- = K_k \nu_{k,j} \quad (52)$$

$$\nu_{k,j} = \text{Log}_I^{G_s}\left(h\left(\hat{x}_{k|k-}, 0\right)^{-1} z_{k,j}\right) \quad (53)$$

$$K_k = P_{k|k-} H_k^\top S_k^{-1} \quad (54)$$

$$S_k = H_k P_{k|k-} H_k^\top + V_k R V_k^\top \quad (55)$$

$$P_{k|k}^- = (I - K_k H_k) P_{k|k-}, \quad (56)$$

and where R is the measurement noise covariance, and the Jacobians H_k and V_k are defined in (29) and evaluated at the point $\zeta_{h_k} = (\hat{x}_{k|k-}, 0)$.

Proof: See Appendix C.

Equation (52) is the updated mean of the error state before being reset to zero, Equation (53) is the innovation term, Equation (54) is the Kalman gain, Equation (55) is the innovation covariance, and Equation (56) is the updated error covariance before the updated mean is reset. Equation (50) adds the mean of the error state to the state estimate essentially resetting the error state's mean to zero, and Equation (51) is the covariance update as a consequence of resetting the error state's mean to zero.

Using the probabilities of the split tracks defined in equation (48) and the probabilities of each association event defined in equation (47), the probability of the track's state conditioned on it representing the target and all track-associated measurements is calculated using the

theorem of total probability:

$$p(x_k | \epsilon_k, Z_{0:k}) = \sum_{j=0}^{m_k} p(x_{k,j} | \theta_{k,j}, \epsilon_k, Z_{0:k}) \beta_{k,j}. \quad (57)$$

In essence, the probability of the track's state is the weighted average of the split tracks' state probabilities where the weight is the probability of the corresponding association event. The process of splitting tracks could be repeated every time new measurements are received which would lead to an exponential growth of split tracks. This process would quickly become computationally and memory expensive. To keep the problem manageable, the split tracks are fused together using the smoothing property of conditional expectations discussed in [34].

Normally if the track's state estimate is expressed in Euclidean space, the smoothing property of conditional expectations would indicate that the state estimates should be fused together according to the equation

$$\hat{x}_{k|k} = \sum_{j=0}^{m_k} \hat{x}_{k|k,j} \beta_{k,j}, \quad (58)$$

as in the original probabilistic data association filter. However, this approach does not work with arbitrary Lie groups since not every Lie group has a vector space structure. Instead we have to solve equation (57) indirectly by posing the problem in the Cartesian algebraic space. This is accomplished by using the relation in equation (21) to note that

$$p(x_{k,j} | \epsilon_k, Z_{0:k}) = p(\tilde{x}_{k,j}^- | \epsilon_k, Z_{0:k}), \quad (59)$$

where $\tilde{x}_{k,j}^-$ is the error state conditioned on $\theta_{k,j}$ after update but before its mean is reset to zero as discussed in Lemma 6. We use this version of the error state since we can fuse the means $\mu_{k|k,j}^-$, defined in equation (52), together in a similar way as in equation (58).

Using the relation in equation (59), an equivalent expression to equation (57) is

$$p(\tilde{x}_{k|k}^- | \epsilon_k, Z_{0:k}) = \sum_{j=0}^{m_k} p(\tilde{x}_{k|k,j}^- | \theta_{k,j}, \epsilon_k, Z_{0:k}) \beta_{k,j}, \quad (60)$$

with the relations $x_k = \hat{x}_{k|k-} \text{Exp}_I^{G_x}(\tilde{x}_{k|k}^-)$ and $x_{k,j} = \hat{x}_{k|k-,j} \text{Exp}_I^{G_x}(\tilde{x}_{k|k,j}^-)$.

In essence, the validated measurements are used to update the mean of the error state for each split track $\mu_{k|k,j}^-$ according to Lemma 6. Since the means are elements of the Cartesian algebraic space (in the same tangent space), they can be added together using a weighted average to form a single mean $\mu_{k|k}^-$. The mean of the error state is then reset to zero by adding $\mu_{k|k}^-$ onto the state estimate $\hat{x}_{k|k-}$ using the exponential map. This process is depicted in Fig. 5.

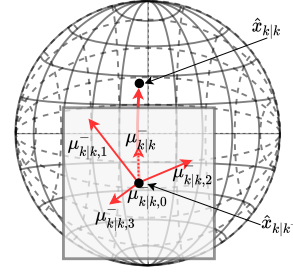


Fig. 5: The error state means for each split track before reset $\mu_{k|k,j}^-$ are fused together using the association event probabilities $\beta_{k,j}$ in the tangent space to form the error state mean $\mu_{k|k}^-$. The mean is then used to update the track's state estimate from $\hat{x}_{k|k-}$ to $\hat{x}_{k|k}$.

Using the smoothing property of conditional expectations, the expected value of the error state $\tilde{x}_{k|k}^-$ is

$$\mathbb{E}[\tilde{x}_{k|k}^-] = \mathbb{E}[\mathbb{E}[\tilde{x}_{k|k,j}^- | \theta_{k,j}, \epsilon_k]] \quad (61a)$$

$$= \sum_{j=0}^{m_k} \mathbb{E}[\tilde{x}_{k|k,j}^- | \theta_{k,j}, \epsilon_k] \beta_{k,j} \quad (61b)$$

$$= \sum_{j=0}^{m_k} \mu_{k|k,j}^- \beta_{k,j} \quad (61c)$$

$$= \mu_{k|k}^- \quad (61d)$$

where $\beta_{k,j}$ is defined in Lemma 5 and $\mu_{k|k,j}^-$ is defined in Lemma 6.

Using the updated mean of the error state, the covariance is

$$\text{cov}[\tilde{x}_{k|k}^-] = \mathbb{E}\left[\left(\tilde{x}_{k|k}^- - \mu_{k|k}^-\right)\left(\tilde{x}_{k|k}^- - \mu_{k|k}^-\right)^\top\right].$$

From this point the derivation follows from [4] and results in

$$P_{k|k}^- = \beta_{k,0} P_{k|k-} + (1 - \beta_{k,0}) P_{k|k}^{c-} + \tilde{P}_{k|k}, \quad (62)$$

where

$$P_{k|k}^{c-} = (I - K_k H_k) P_{k|k-} \quad (63a)$$

$$K_k = P_{k|k-} H_k^\top S_k^{-1} \quad (63b)$$

$$S_k = H_k P_{k|k-} H_k^\top + V_k R V_k^\top \quad (63c)$$

$$\tilde{P}_{k|k} = K_k \left(\sum_{j=1}^{m_k} \beta_{k,j} \nu_{k,j} \nu_{k,j}^\top - \nu_k \nu_k^\top \right) K_k^\top \quad (63d)$$

$$\nu_{k,j} = \text{Log}_I^{G_s} \left(h(\hat{x}_{k|k-}, 0)^{-1} z_{k,j} \right) \quad (63e)$$

$$\nu_k = \sum_{j=1}^{m_k} \beta_{k,j} \nu_{k,j} \quad (63f)$$

$$\mu_{k|k}^- = K_k \nu_k, \quad (63g)$$

and where $P_{k|k-}$ is the error covariance before the update step, $P_{k|k}^{c-}$ is the error covariance of the error state $\tilde{x}_{k|k,j}^-$ derived in Lemma 6, $\tilde{P}_{k|k}$ is the covariance that

captures the "spread of the means", and the Jacobians H_k and V_k are defined in (29) and evaluated at the point $\zeta_{h_k} = (\hat{x}_{k|k-}, 0)$.

To reset the error state's mean to zero, the mean $\mu_{k|k}^-$ is added onto the state estimate $\hat{x}_{k|k-}$ by forming a geodesic from $\hat{x}_{k|k-}$ to $\hat{x}_{k|k}$ in the direction of $\mu_{k|k}^-$ as depicted in Fig 5. To derive this process, let $\tilde{x}_{k|k}^- = \mu_{k|k}^- + a_{k|k}$ where $a_{k|k} \sim \mathcal{N}(0, P_{k|k}^{c-})$ contains the uncertainty in the error state. Then under the assumption that $a_{k|k}$ is small and using the property of the right Jacobian defined in equation (15) we add $\mu_{k|k}^-$ onto $\hat{x}_{k|k-}$ as follows:

$$x_{k|k} = \hat{x}_{k|k-} \text{Exp}_I^{G_x} \left(\mu_{k|k}^- + a_{k|k} \right) \quad (64a)$$

$$\approx \underbrace{\hat{x}_{k|k-} \text{Exp}_I^{G_x} \left(\mu_{k|k}^- \right)}_{\hat{x}_{k|k}} \text{Exp}_I^{G_x} \left(\underbrace{J_r^{G_x} \left(\mu_{k|k}^- \right) a_{k|k}}_{\tilde{x}_{k|k}} \right), \quad (64b)$$

where $\hat{x}_{k|k} = \hat{x}_{k|k-} \text{Exp}_I^{G_x} \left(\mu_{k|k}^- \right)$ is the updated state estimate, and $\tilde{x}_{k|k} = J_r^{G_x} \left(\mu_{k|k}^- \right) a_{k|k}$ is the updated and reset error state. The error covariance of the error state $\tilde{x}_{k|k}$ is

$$\begin{aligned} \text{cov}(\tilde{x}_{k|k}) &= \text{cov} \left(J_r^{G_x} \left(\mu_{k|k}^- \right) a_{k|k} \right) \\ &= J_r^{G_x} \left(\mu_{k|k}^- \right) \text{cov}(a_{k|k}) J_r^{G_x} \left(\mu_{k|k}^- \right)^\top \\ &= J_r^{G_x} \left(\mu_{k|k}^- \right) P_{k|k}^- J_r^{G_x} \left(\mu_{k|k}^- \right)^\top \\ &= P_{k|k}, \end{aligned}$$

and therefore, $\tilde{x}_{k|k} \sim \mathcal{N}(\mu_{k|k} = 0, P_{k|k})$.

LEMMA 7. *Suppose that Assumptions 1-8 hold, and that Z_k is the set of validated measurements, then the update for the track likelihood is*

$$\epsilon_{k|k} \doteq p(\epsilon_k | Z_{0:k}) = \frac{1 - \alpha_k}{1 - \alpha_k p(\epsilon_k | Z_{0:k-})} p(\epsilon_k | Z_{0:k-}), \quad (65)$$

where

$$\alpha_k = \begin{cases} P_D P_G, & \text{if } m_k = 0 \\ P_D P_G - \sum_{j=1}^{m_k} \mathcal{L}_{k,j}, & \text{otherwise} \end{cases}, \quad (66)$$

and $\mathcal{L}_{k,j}$ is defined in equation (46).

Proof: See Appendix D.

Using the track likelihood, a track is either rejected after the update step if the track likelihood is below the threshold τ_{RT} , confirmed to represent a target if the track likelihood is above the threshold τ_{CT} , or neither rejected nor confirmed until more information is gathered with new measurements. In the case of tracking a single target, the confirmed track with the best track likelihood is assumed to represent the target. In the case of tracking multiple targets, every confirmed track is assumed to represent a different target.

Therefore, we have the following main theorem that summarizes the results.

THEOREM 1. *If Assumptions 1-8 hold and $Z_k = \{z_{k,j}\}_{j=1}^{m_k}$ denotes the set of m_k validated measurements at time t_k and $\hat{x}_{k|k-}$, $P_{k|k-}$ and $p(\epsilon_k | Z_{0:k-})$ denote the track's state estimate, error covariance and track likelihood at time t_k and conditioned on the previous measurements, then the track's updated state estimate, error covariance and track likelihood are*

$$\hat{x}_{k|k} = \hat{x}_{k|k-} \text{Exp}_I^{G_x} \left(\mu_{k|k}^- \right) \quad (67)$$

$$P_{k|k} = J_r^{G_x} \left(\mu_{k|k}^- \right) P_{k|k-} J_r^{G_x} \left(\mu_{k|k}^- \right)^\top \quad (68)$$

$$p(\epsilon_k | Z_{0:k}) = \frac{1 - \alpha_k}{1 - \alpha_k p(\epsilon_k | Z_{0:k-})} p(\epsilon_k | Z_{0:k-}) \quad (69)$$

where $\mu_{k|k}^-$ is given in Equation (63), $P_{k|k}^-$ is given in Equation (62), and α_k is defined in equation (66).

Proof: Follows directly from Lemmas 1, 4, 5, 6, and 7.

D. Track Initialization

A track can be initialized in a variety of ways. We refer the reader to [4] for some of the common methods, and we adapt one of the methods to Lie groups in this section. We assume that the system model is as defined by equations (24), (26) and (25).

Let z_k and z_i denote two non-track-associated measurements from distinct times where z_k is a measurement from the current time and z_i is a measurement from a previous time spatially close to the measurement z_k . We select two measurements that are close together since a moving target forms a continuous trajectory and produces measurements spatially close together.

According to the observation function in Equation (26), the target's pose is observed. Therefore, we can estimate that $\hat{g}_k \approx z_k$. According to the state transition function in equation (25), the target is assumed to have constant velocity. We solve for the velocity using the equation

$$v_k \approx \frac{\text{Log}_I^{G_s}(z_i^{-1} z_k)}{t_k - t_i}, \quad (70)$$

where $t_k - t_i$ is the time interval between the measurements. The tracks current state estimate is set to $\hat{x}_{k|k} \approx (\hat{g}_{k|k}, \hat{v}_{k|k})$ where $\hat{g}_{k|k} = z_k$ and $\hat{v}_{k|k}$ is solved for by equation (70).

The initial error covariance is specified by the application. As an example, for the simple example discussed in Section VI, we set the initial error covariance to $P = 5I_{2n \times 2n}$ to reflect the uncertainty in the initial state estimate and the high velocity of the moving car. The initial track likelihood can be set to $\epsilon_{k|k} = 0.5$ to reflect the fact that the new track is just as likely to represent clutter as it is to represent the target.

VI. Example

We demonstrate the LG-IPDAF in simulation by tracking a car that evolves on SE(2) and whose system

model is described in Section IV. The car is given four trajectories: a circular trajectory with a 10 meter radius, a Zamboni-like trajectory consisting of curves and straight lines, a spiral trajectory, and a straight-line trajectory. In the first trajectory the car is given a translational velocity of 10 m/s and an angular velocity of 1 rad/s . In the second trajectory the car is given a translational velocity of 10 m/s and an angular velocity that varies between zero to 1.5 rad/s to create the U-turns. In the third trajectory the target is given an angular velocity of 1 rad/s and a translational velocity that increases from two to 10 m/s . In the fourth trajectory the car is given a translational velocity of 7 m/s . We selected these trajectories to compare how well a constant-velocity model on SE(2) (SE(2)-CV) implemented using the LG-IPDAF versus a constant-velocity, linear model on \mathbb{R}^2 (LTI-CV) implemented using the IPDAF tracks a target that undergoes straight-line and non-straight-line motion. The sensor is modeled as being able to detect the car's pose with measurement noise covariance $R = \text{diag}(10^{-1}, 10^{-1}, 10^{-2})$ and a surveillance region of 140 by 140 meters.

We set the following LG-IPDAF and IPDAF parameters: the spatial density of false measurements to $\lambda = 0.01$ implying that there is an average of 196 false measurements every sensor scan, the car's probability of detection to $P_D = 0.9$, gate probability to $P_G = 0.9$, track likelihood confirmation threshold to $\tau_{CT} = 0.7$, and track likelihood rejection threshold to $\tau_{RT} = 0.1$. We initialize new tracks from two unassociated neighboring measurements from different times with an initial error covariance of $P = 5I_{2n \times 2n}$ and track likelihood to $\epsilon = 0.2$. The process noise of the initialized tracks is $Q = \text{diag}(1, 1, 0.1, 1, 1, 0.1) dt$, where the simulated time step $dt = 0.1$. The initialized process noise is large enough to account for the car's acceleration, but small enough to prevent many false measurements from being associated with the track. We also use a low initial track likelihood value due to the high number of false measurements per sensor scan making it more probable that an initialized track does not represent the target.

For each model (the SE(2)-CV and LTI-CV models) and trajectory, we conduct a Monte-Carlo simulation consisting of 100 iterations, each 30 seconds long, and compute three statistical measures: track probability of detection (TPD), average Euclidean error (AEE) in position, and average confirmation time (ACT). Track probability of detection is the probability that the target is being tracked [35]. The average Euclidean error, is the confirmed track's average error in position. The average confirmation time is the average amount of time until the first track is confirmed.

A depiction of the simulation for the circular, Zamboni, spiral, and straight-line trajectories are shown in Figs. 6, 7, 8 and 9. The target's trajectory is shown as a black line, and the target's final pose is represented as the large black arrowhead. The confirmed tracks' trajectories are shown as piece-wise, green lines and the confirmed track's final pose is represented as a large green

arrowhead. The target's measurements from the initial to the final time are represented as magenta dots. The red arrowheads represent different unconfirmed and non-rejected track's at the final time step of the simulation. The blue asterisks represent the false measurements received at the last time step of the simulation. During non-straight-line motions, the LTI-CV model struggles to track the target. This is shown in two ways: first by there not being a green trajectory (confirmed track's trajectory) during parts of the target's non-straight-line motion, and second the green trajectories drifting from the black trajectory (target's trajectory). The second case is prevalent in the circular trajectory for the LTI model indicating that many LTI tracks were initialized and confirmed, but quickly drifted off the circular trajectory and being pruned once their track likelihood fell below the pruning threshold..

The statistical measures from the experiment are in Table I. As stated in the table, the SE(2)-CV model tracked the target significantly better for the circular, Zamboni and spiral trajectories, and about the same as the LTI-CV model for the straight-line trajectory according to the track probability of detection measure. This is because the LTI-CV model struggles to track non-straight-line motion. In Fig. 7a you can see that during the straight segments, the LTI-CV model tracks the target well, but loses the target as it turns. Table I also shows that the SE(2)-CV had less error than the LTI-CV model. However, on average the LTI-CV model had faster track confirmation time. We believe that this is due to the dimension of the measurement and state space. The smaller the dimension, the smaller the initial volume of the validation region (i.e. the initial overall uncertainty in the state estimate was less). The initial smaller uncertainty allowed the track likelihood for the LTI-CV model to increase faster than for the SE(2)-CV model.

Since the main objective of this paper is deriving and presenting the LG-IPDAF algorithm, we did not do a detailed simulated analysis on the effects of changing the parameters used by the algorithm and in the simulation. The purpose of the experimental result is to show that there are cases where an LTI-CV model (or the original IPDAF algorithm) is insufficient. In essence, the LTI-CV model fails when it is no longer able to properly associate the correct measurement to the track. This occurs frequently when the target's position moves either outside of the LTI-CV model's validation region or if a series of false measurements are deemed more probable than the correct measurement according to the validation region. In the case of the circular trajectory, the target's position is often outside of the LTI-CV model's validation region causing the track to become lost. This is because the LTI-CV model predicts a straight-line motion that is tangent to the target's circular motion, causing the model to slowly drift away from the target's true position.

The disadvantage of the LG-IPDAF is that it is more complex to implement and it is more computationally expensive compared to the IPDAF. To help demonstrate the difference in computational complexity we did an analyt-

ical and experimental analysis of the two algorithms used in our experiment. On average, a single iteration of the IPDAF algorithm using the LTI-CV model took 4.4 seconds, whereas the LG-IPDAF algorithm using the SE(2)-CV model required 13.2 seconds per iteration. For the analytical analysis, let x_d denote the dimension of the state, $g_d = x_d/2$ the dimension of the pose, z_d the dimension of the measurement, a the number of measurements received in a single sensor scan, and m denote the number of associated measurements, then the approximate computational complexity of the IPDAF is $\mathcal{O}(z_d^2(a + 3m) + 5x_d^3)$ and the approximate computational complexity of the LG-IPDAF is $\mathcal{O}(z_d^3 \log(z_d)(a + m) + 2g_d^3 \log(g_d))$.

The main difference in complexity between the two algorithms is the matrix logarithm and exponential functions which have complexity $\mathcal{O}(j^3 \log(j))$ where j^2 is the number of elements in the matrix. In the experiment, the data association step took the longest due to the number of measurements and dominated the computational complexity. If we compare just the computational complexity of the data association step by taking the ratio of the two we get

$$\frac{3^3 \log(3)a}{2^2 a} \approx 3.3,$$

where 3 is the dimension of the measurement space used by the SE(2)-CV model since it includes orientation, and 2 is the dimension of the measurement space used by the LTI-CV model. This ratio is approximately the ratio of the simulated time $13.2/4.4 = 3$ seconds.

This computational complexity analysis is done without any optimization. Some of the more interesting Lie groups such as SE(n) and SO(n) have much more efficient methods to calculate the matrix exponential and logarithm, for example, the Rodriguez formula for SO(3).

Even though the LG-IPDAF is more computationally complex, as shown by the experiments there are occasions when the IPDAF is inadequate and cannot properly track objects that have non-straight-line trajectories. This usually occurs in the presence of dense clutter or when the rate of receiving new measurements is too low to compensate for the target's non-straight-line motion resulting in the target's position drifting outside the track's validation region. It is in these instances that we recommend the use of the LG-IPDAF.

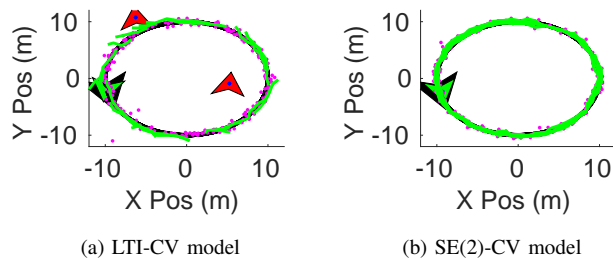


Fig. 6: Plots of the zoomed in circular trajectories for the LTI-CV and SE(2)-CV models.

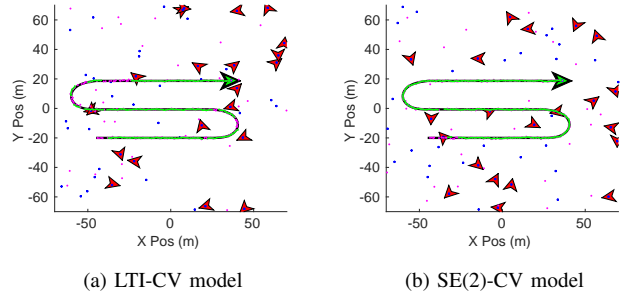


Fig. 7: Plots of the Zamboni-like trajectories for the LTI-CV and SE(2)-CV models.

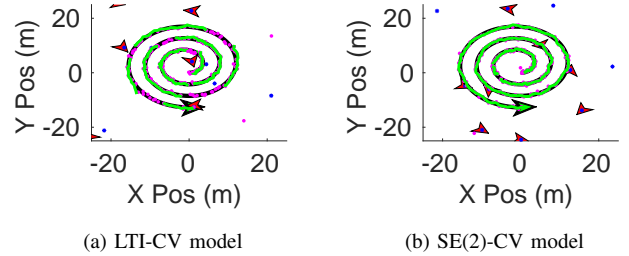


Fig. 8: Plots of the Spiral-like trajectories for the LTI-CV and SE(2)-CV models.

VII. Conclusion

In this paper we have shown how to adapt the IPDA filter to connected, unimodular Lie groups. In our example, we showed that a constant-velocity target model on SE(2) is significantly better able to track non-linear motion in dense clutter than the LTI-CV model. This is because the LTI-CV model expresses only linear motion and cannot predict non-linear motion such as curvy and circular trajectories. However, the SE(2)-CV model ex-

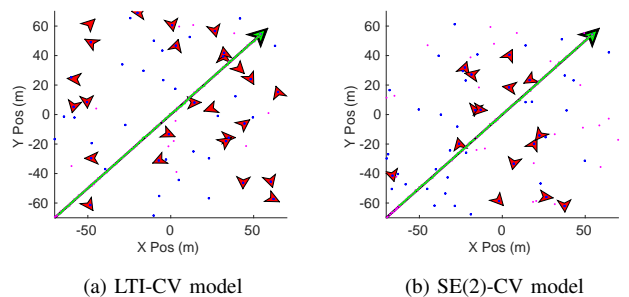


Fig. 9: Plots of straight-line trajectories for the LTI-CV and SE(2)-CV models.

TABLE I: Statistical measures from the experiment

	Circular			Zamboni			Spiral			Straight		
	TPD	AEE	ACT	TPD	AEE	ACT	TPD	AEE	ACT	TPD	AEE	ACT
SE(2)	0.96	0.29	0.90	0.96	0.29	0.87	0.96	0.29	0.83	0.96	0.29	0.74
LTI	0.54	1.32	0.49	0.88	0.45	0.44	0.81	0.39	0.50	0.97	0.34	0.42

presses non-linear and linear motion, and thus it is able to predict both types of motion and track them well. We have also shown that the LG-IPDAF is capable of quickly rejecting and confirming tracks with high fidelity.

Appendix A

Proof of Lemma 3: Prediction Step

Proof:

According to the theorem of total probability, the probability of the track's current state conditioned on the previous measurements and the track representing the target is

$$p(x_k | \epsilon_{k-}, Z_{0:k-}) = \int_{\tilde{x}_{k-|k-}} p(x_k | x_{k-}, \epsilon_{k-}, Z_{0:k-}) p(x_{k-} | \epsilon_{k-}, Z_{0:k-}) d\tilde{x}_{k-|k-}, \quad (71)$$

where we integrate over the error state instead of the state by recalling that $\tilde{x}_{k-|k-} = \text{Log}_I^{G^x}(\hat{x}_{k-|k-}^{-1} x_{k-})$, and using the relation in equation (21).

Using the definitions of $p(x_k | x_{k-}, \epsilon_{k-}, Z_{0:k-})$ and $p(x_{k-} | \epsilon_{k-}, Z_{0:k-})$ in Equations (33) and (34), Equation (71) becomes

$$p(x_k | \epsilon_{k-}, Z_{0:k-}) \approx \eta \int_{\tilde{x}_{k-|k-}} \exp(-L) d\tilde{x}_{k-|k-}, \quad (72)$$

where

$$L = \frac{1}{2} (\tilde{x}_{k|k-} - F_\Delta \tilde{x}_{k-|k-})^\top \bar{Q}_\Delta^{-1} (\tilde{x}_{k|k-} - F_\Delta \tilde{x}_{k-|k-}) + \frac{1}{2} \tilde{x}_{k-|k-}^\top P_{k-|k-}^{-1} \tilde{x}_{k-|k-},$$

and where \bar{Q}_Δ is defined in equation (35). After some algebra we get

$$L = \frac{1}{2} \left\{ \tilde{x}_{k-|k-}^\top (P_{k-|k-}^{-1} + F_\Delta^\top \bar{Q}_\Delta^{-1} F_\Delta) \tilde{x}_{k-|k-} + \tilde{x}_{k|k-}^\top \bar{Q}_\Delta^{-1} \tilde{x}_{k|k-} - \tilde{x}_{k|k-} \bar{Q}_\Delta^{-1} F_\Delta \tilde{x}_{k-|k-} - \tilde{x}_{k-|k-}^\top F_\Delta^\top \bar{Q}_\Delta^{-1} \tilde{x}_{k|k-} \right\}$$

Defining $S \doteq P_{k-|k-}^{-1} + F_\Delta^\top \bar{Q}_\Delta^{-1} F_\Delta$ gives

$$L = \frac{1}{2} (\tilde{x}_{k-|k-} - S^{-1} F_\Delta^\top \bar{Q}_\Delta^{-1} \tilde{x}_{k|k-})^\top S (\tilde{x}_{k-|k-} - S^{-1} F_\Delta^\top \bar{Q}_\Delta^{-1} \tilde{x}_{k|k-}) + \frac{1}{2} \tilde{x}_{k|k-} (\bar{Q}_\Delta^{-1} - \bar{Q}_\Delta^{-1} F_\Delta S^{-1} F_\Delta^\top \bar{Q}_\Delta^{-1}) \tilde{x}_{k|k-}.$$

Using the matrix inversion lemma

$$\bar{Q}_\Delta^{-1} - \bar{Q}_\Delta^{-1} F_\Delta (F_\Delta^\top \bar{Q}_\Delta^{-1} F_\Delta + P_{k-|k-}^{-1})^{-1} F_\Delta^\top \bar{Q}_\Delta^{-1} = (F_\Delta P_{k-|k-} F_\Delta^\top + \bar{Q}_\Delta)^{-1}$$

gives

$$L = L_1(\tilde{x}_{k-|k-}, \tilde{x}_{k|k-}) + \frac{1}{2} \tilde{x}_{k|k-} (F_\Delta P_{k-|k-} F_\Delta^\top + \bar{Q}_\Delta)^{-1} \tilde{x}_{k|k-},$$

and therefore

$$p(x_k | \epsilon_{k-}, Z_{0:k-}) = \eta \exp(-L_2(\tilde{x}_{k|k-})),$$

where the integral of $\exp(-L_1)$ has been absorbed into the normalizing coefficient η .

Let

$$P_{k|k-} = F_\Delta P_{k-|k-} F_\Delta^\top + \bar{Q}_\Delta$$

and recall that $\tilde{x}_{k|k-} = \text{Log}_I^{G^x}(\hat{x}_{k|k-}^{-1} x_k)$, where from Lemma 2 we have that $\hat{x}_{k|k-} = f(\hat{x}_{k-|k-}, 0, t_\Delta)$. Therefore, the Lemma follows by noting that $p(x_k | \epsilon_{k-}, Z_{0:k-}) \approx \mathcal{N}(\hat{x}_{k|k-}, P_{k|k-})$. ■

Appendix B

Proof of Lemma 5: Association Events

Proof:

The probability of an association event conditioned on the measurements $Z_{0:k}$ and that the track represents the target is inferred using the location of the validated measurements with respect to the track's estimated measurement $\hat{z}_k = h(\hat{x}_{k|k-}, 0)$ and the number of validated measurements. The basic idea is that the closer a validated measurement is to the estimated measurement relative to how close the other validated measurements are to the estimated measurement, the more likely the validated measurement is the target-originated measurement, and therefore the more likely its respective association event it.

The probability of an association event conditioned on the measurements and the track representing the target is

$$p(\theta_{k,j} | \epsilon_k, Z_{0:k}) = p(\theta_{k,j} | \epsilon_k, Z_k, Z_{0:k-}, m_k), \quad (73)$$

where we have explicitly written the inference on the number of validated measurements m_k . Using Bayes' rule, the probability in equation (73) is

$$\frac{p(\theta_{k,j} | \epsilon_k, Z_{0:k})}{p(Z_k | \theta_{k,j}, m_k, \epsilon_k, Z_{0:k-})} = \frac{p(\theta_{k,j} | m_k, \epsilon_k, Z_{0:k-})}{p(Z_k | m_k, \epsilon_k, Z_{0:k-})}, \quad (74)$$

where

$$p(Z_k | m_k, \epsilon_k, Z_{0:k-}) = \prod_{j=0}^{m_k} (p(Z_k | \theta_{k,j}, m_k, \epsilon_k, Z_{0:k-}) p(\theta_{k,j} | m_k, \epsilon_k, Z_{0:k-})). \quad (75)$$

Since the validated measurements are independent, the joint density of the validated measurements is

$$p(Z_k | \theta_{k,j}, m_k, \epsilon_k, Z_{0:k-}) = \prod_{\ell=1}^{m_k} p(z_{k,\ell} | \theta_{k,j}, \epsilon_k, Z_{0:k-}). \quad (76)$$

Since the false measurements are assumed uniformly distributed in the validation region with volume \mathcal{V}_k , the probability of a false measurement in the validation

region is \mathcal{V}_k^{-1} . Also, since the target originated measurement is validated with probability P_G , the probability of a target originated measurement being validated is $P_G^{-1}p(z_{k\ell} | \psi, \epsilon_k, Z_{0:k-})$, which is defined in equations (40) (44) and (45). Therefore, the probability of each measurement conditioned on the respective association event is

$$p(z_{k,\ell} | \theta_{k,j}, \epsilon_k, Z_{0:k-}) = \begin{cases} \mathcal{V}_k^{-1} & \text{if } j \neq \ell \\ P_G^{-1}p(z_{k,\ell} | \psi, \epsilon_k, Z_{0:k-}) & \text{if } j = \ell \end{cases} \quad (77)$$

Let $m_k^- \triangleq m_k - 1$, and $\mathbb{M} = \{1, \dots, m_k\}$. Substituting equation (77) into equation (76) yields the joint probability of the measurements

$$p(Z_k | \theta_{k,j}, m_k, \epsilon_k, Z_{0:k-}) = \begin{cases} \mathcal{V}_k^{-m_k} P_G^{-1}p(z_{k,j} | \psi, \epsilon_k, Z_{0:k-}) & \text{if } j \in \mathbb{M} \\ \mathcal{V}_k^{-m_k} & \text{if } j = 0 \end{cases} \quad (78)$$

We now proceed to calculate $p(\theta_{k,j} | m_k, \epsilon_k, Z_{0:k-})$. Let ϕ denote the number of false measurements. Under the assumption that there is at most one target originated measurement, there are two possibilities for the number of false measurements, $\phi = m_k$, denoted ϕ_{m_k} , or $\phi = m_k^-$ denoted $\phi_{m_k^-}$. Therefore, the *a priori* probability of an association event conditioned on the number of measurements and previous measurements is

$$p(\theta_{k,j} | m_k, \epsilon_k, Z_{0:k-}) = p(\theta_{k,j} | \phi_{m_k^-}, m_k, \epsilon_k, Z_{0:k-}) p(\phi_{m_k^-} | m_k, \epsilon_k, Z_{0:k-}) + p(\theta_{k,j} | \phi_{m_k}, m_k, \epsilon_k, Z_{0:k-}) p(\phi_{m_k} | m_k, \epsilon_k, Z_{0:k-}) = \begin{cases} \left(\frac{1}{m_k}\right) p(\phi_{m_k^-} | m_k, \epsilon_k, Z_{0:k-}) & j = 1, \dots, m_k \\ p(\phi_{m_k} | m_k, \epsilon_k, Z_{0:k-}) & j = 0, \end{cases} \quad (79)$$

where the probability $p(\theta_{k,j} | \phi_{m_k^-}, m_k, \epsilon_k, Z_{0:k-}) = \frac{1}{m_k}$ when $j > 0$ since each association event, $\theta_{k,j>0}$, is just as likely to be true with the specified conditions, and $p(\theta_{k,j} | \phi_{m_k^-}, m_k, \epsilon_k, Z_{0:k-}) = 0$ when $j = 0$ since not all of the validated measurements can be false according to the condition $\phi_{m_k^-}$.

Using Bayes' formula, the conditional probabilities of the number of false measurements are

$$p(\phi_{m_k^-} | m_k, \epsilon_k, Z_{0:k-}) = \frac{p(m_k | \phi_{m_k^-}, \epsilon_k, Z_{0:k-}) p(\phi_{m_k^-} | \epsilon_k, Z_{0:k-})}{p(m_k | \epsilon_k, Z_{0:k-})} = \frac{P_G P_D \mu_F(m_k^-)}{p(m_k | \epsilon_k, Z_{0:k-})} \quad (80)$$

and

$$p(\phi_{m_k} | m_k, \epsilon_k, Z_{0:k-}) = \frac{p(m_k | \phi_{m_k}, \epsilon_k, Z_{0:k-}) p(\phi_{m_k} | Z_{0:k-})}{p(m_k | \epsilon_k, Z_{0:k-})} = \frac{(1 - P_G P_D) \mu_F(m_k)}{p(m_k | \epsilon_k, Z_{0:k-})}, \quad (81)$$

where μ_F is the probability density function of the number of false measurements, and $p(m_k | \phi_{m_k^-}, \epsilon_k, Z_{0:k-}) = P_G P_D$ since $p(m_k | \phi_{m_k^-}, \epsilon_k, Z_{0:k-})$ is the probability that the target is detected and the target originated measurement is inside the validation region.

According to the theorem of total probability

$$p(m_k | \epsilon_k, Z_{0:k-}) = p(m_k | \phi_{m_k^-}, \epsilon_k, Z_{0:k-}) p(\phi_{m_k^-} | \epsilon_k, Z_{0:k-}) + p(m_k | \phi_{m_k}, \epsilon_k, Z_{0:k-}) p(\phi_{m_k} | \epsilon_k, Z_{0:k-}) = P_G P_D \mu_F(m_k^-) + (1 - P_G P_D) \mu_F(m_k). \quad (82)$$

Substituting equations (80), (81), and (82) into equation (79) yields

$$p(\theta_{k,j} | m_k, \epsilon_k, Z_{0:k-}) = \begin{cases} \frac{\frac{1}{m_k} P_D P_G \mu_F(m_k^-)}{P_G P_D \mu_F(m_k^-) + (1 - P_G P_D) \mu_F(m_k)} & j \in \mathbb{M} \\ \frac{(1 - P_G P_D) \mu_F(m_k)}{P_G P_D \mu_F(m_k^-) + (1 - P_G P_D) \mu_F(m_k)} & j = 0 \end{cases} \quad (83)$$

Substituting equations (78) and (79), into equation (75) yields

$$p(Z_k | m_k, \epsilon_k, Z_{0:k-}) = \frac{\mathcal{V}_k^{-m_k^-} P_D \mu_F(m_k^-) \sum_{\ell=1}^{m_k} p(z_{k,\ell} | \psi, \epsilon_k, Z_{0:k-})}{P_D P_G \mu_F(m_k^-) + (1 - P_D P_G) \mu_F(m_k)} + \frac{\mathcal{V}_k^{-m_k} (1 - P_D P_G) \mu_F(m_k)}{P_D P_G \mu_F(m_k^-) + (1 - P_D P_G) \mu_F(m_k)}. \quad (84)$$

Substituting equations (78), (79) and (84) into (74) yields

$$p(\theta_{k,j} | \epsilon_k, Z_{0:k-}) = \begin{cases} \frac{P_D p(z_{k,j} | \psi, \epsilon_k, Z_{0:k-})}{P_D \sum_{\ell=1}^{m_k} p(z_{k,\ell} | \psi, \epsilon_k, Z_{0:k-}) + m_k \mathcal{V}_k^{-1} (1 - P_D P_G) \frac{\mu_F(m_k)}{\mu_F(m_k^-)}} & j \in \mathbb{M} \\ \frac{m_k \mathcal{V}_k^{-1} (1 - P_D P_G) \frac{\mu_F(m_k)}{\mu_F(m_k^-)}}{P_D \sum_{\ell=1}^{m_k} p(z_{k,\ell} | \psi, \epsilon_k, Z_{0:k-}) + m_k \mathcal{V}_k^{-1} (1 - P_D P_G) \frac{\mu_F(m_k)}{\mu_F(m_k^-)}} & j = 0 \end{cases} \quad (85)$$

Setting the probability density function of false measurements μ_F in equation (85) to a Poisson density function defined in equation (23) yields equation (47). ■

Appendix C

Proof of Lemma 6: Split Track Update

We begin with the following Lemma.

LEMMA 8. *Suppose that Assumptions 1 and 8 hold, then the probability of measurement z_k conditioned on it being*

target-originated and conditioned on the current state, is given by

$$p(z_k | \psi, x_k, \epsilon_k, Z_{0:k-}) \approx \eta \exp\left(-\frac{1}{2}(\tilde{z}_k - H_k \tilde{x}_{k|k-})^\top \bar{R}_k (\tilde{z}_k - H_k \tilde{x}_{k|k-})\right), \quad (86)$$

where

$$\tilde{z}_k = \text{Log}_I^{G_s}(\hat{z}_k^{-1} z_k) \quad (87)$$

$$\hat{z}_k = h(\hat{x}_{k|k-}, 0) \quad (88)$$

$$\bar{R}_k = V_k R V_k^\top, \quad (89)$$

$$\tilde{x}_{k|k-} = \text{Log}_I^{G_s}(\hat{x}_{k|k-}^{-1} x_k), \quad (90)$$

R is the measurement noise covariance, and the Jacobians H_k and V_k are given in equation (29) and evaluated at the point $\zeta_{k_k} = (\hat{x}_{k|k-}, 0)$.

Proof: Similar to the proof of Lemma 2.

Proof of Lemma 6. Using Bayes rule,

$$p(x_{k,j} | \theta_{k,j}, \epsilon_k, Z_{0:k}) = \frac{p(Z_k | \theta_{k,j}, x_{k,j}, \epsilon_k, Z_{0:k-}) p(x_{k,j} | \epsilon_k, Z_{0:k-})}{p(Z_k | \theta_{k,j}, \epsilon_k, Z_{0:k-})}. \quad (91)$$

We solve for the probability $p(x_{k,j} | \theta_{k,j}, \epsilon_k, Z_{0:k})$ by using the maximum a posterior (MAP) optimization algorithm to find the value of $\tilde{x}_{k|k-}$ and its corresponding error covariance $P_{k|k-}$ that maximizes the right-hand-side of equation (91).

Since the MAP does not depend on $p(Z_k | \theta_{k,j}, \epsilon_k, Z_{0:k-})$, it can be absorbed into the normalizing coefficient simplifying the problem to

$$\max_{\tilde{x}_{k|k-}, P_{k|k-}} \eta p(Z_k | \theta_{k,j}, x_{k,j}, \epsilon_k, Z_{0:k-}) p(x_{k,j} | \epsilon_k, Z_{0:k-}).$$

When $j = 0$, none of the validated measurements are target originated, and thus, $p(Z_k | \theta_{k,j=0}, x_k, \epsilon_k, Z_{0:k-})$ simplifies to $p(Z_k | \theta_{k,j=0}, Z_{0:k-})$ and no longer has any dependency on the track's state. This reduces the MAP optimization problem to

$$\max_{\tilde{x}_{k|k-}, P_{k|k-}} \eta p(x_{k,j} | \epsilon_k, Z_{0:k-}),$$

and hence, the solution is $p(x_{k,j=0} | \theta_{k,j=0}, \epsilon_k, Z_{0:k}) = p(x_k | \epsilon_k, Z_{0:k-})$.

When $j > 0$, only the measurement $z_{k,j}$ has any dependency and influence on the track's state since all other measurements are false. Using this fact with the relation in Equation (77), the optimization problem simplifies to

$$\max_{\tilde{x}_{k|k-}, P_{k|k-}} \eta P_G^{-1} p(z_{k,j} | \psi, x_{k,j}, \epsilon_k, Z_{0:k-}) \cdot p(x_{k,j} | \epsilon_k, Z_{0:k-}).$$

Multiplying together and combining the exponents of the probabilities $p(z_{k,j} | \psi, x_k, \epsilon_k, Z_{0:k-})$ and $p(x_k | \epsilon_k, Z_{0:k-})$, defined in equations (86) and (38), simplifies the MAP optimization problem to

$$\max_{\tilde{x}_{k|k-}, P_{k|k-}} \eta P_G^{-1} \exp(-L),$$

where

$$L = \frac{1}{2} (\nu_{k,j} - H_k \tilde{x}_{k|k-})^\top \bar{R}_k^{-1} (\nu_{k,j} - H_k \tilde{x}_{k|k-}) + \frac{1}{2} \tilde{x}_{k|k-}^\top P_{k|k-}^{-1} \tilde{x}_{k|k-}$$

$$\nu_{k,j} = \text{Log}_I^{G_s}(\hat{z}_k^{-1} z_k)$$

$$\hat{z}_k = h(\hat{x}_{k|k-}, 0)$$

$$\bar{R}_k = V_k R V_k^\top,$$

and where $\tilde{x}_{k|k-} = \text{Log}_I^{G_x}(\hat{x}_{k|k-}^{-1} x_k)$, h is the observation function defined in equation (24), and H_k and V_k are the Jacobians of the observation function defined in equation (29) and evaluated at $\zeta_{h_k} = (\hat{x}_{k|k-}, 0)$.

The MAP is solved by finding the value of $\tilde{x}_{k|k-}$ that minimizes L . Since L is quadratic in $\tilde{x}_{k|k-}$, the value of $\tilde{x}_{k|k-}$ that minimizes L is found by taking the first derivative of L with respect to $\tilde{x}_{k|k-}$, setting it to zero and solving for $\tilde{x}_{k|k-}$. This value becomes the new error state mean $\mu_{k|k}^-$. The corresponding covariance is found by taking the second derivative of L with respect to $\tilde{x}_{k|k-}$ and setting this value to the new covariance $P_{k|k}^{c-}$.

Taking the first and second partial derivatives of L with respect to $\tilde{x}_{k|k-}$ yields

$$\frac{\partial L}{\partial \tilde{x}_{k|k-}} = -(\nu_{k,j} - H_k \tilde{x}_{k|k-})^\top \bar{R}_k^{-1} H_k + \tilde{x}_{k|k-}^\top P_{k|k-}^{-1}$$

$$\frac{\partial^2 L}{\partial \tilde{x}_{k|k-}^2} = H_k^\top \bar{R}_k^{-1} H_k + P_{k|k-}^{-1} = (P_{k|k}^{c-})^{-1}.$$

Setting the first derivative to zero, solving for $\tilde{x}_{k|k-}$ and setting this value to $\mu_{k|k}^-$ gives

$$\mu_{k|k}^- = \left(H_k^\top \bar{R}_k^{-1} H_k + P_{k|k-}^{-1}\right)^{-1} H_k^\top \bar{R}_k^{-1} \nu_{k,j}.$$

With algebraic manipulation the updated error covariance and error state mean are

$$P_{k|k}^{c-} = (I - K_k H_k) P_{k|k-}$$

$$\mu_{k|k,j}^- = K_k \nu_{k,j},$$

where the Kalman gain K_k and innovation term $\nu_{k,j}$ are

$$K_k = P_{k|k-} H_k^\top S_k$$

$$\nu_{k,j} = \text{Log}_I^{G_s}(\hat{z}_k^{-1} z_{k,j})$$

$$\hat{z}_k = h(\hat{x}_{k|k-}, 0)$$

$$S_k = V_k R V_k^\top + H_k P_{k|k-} H_k^\top.$$

Since the error state's mean is no longer zero, the error state no longer has a concentrated Gaussian distribution. In order to reset the mean of the error state to zero, we add $\mu_{k|k,j}^-$ onto the state estimate $\hat{x}_{k|k-}$ and adjust the covariance of the error state. In particular, let the error state after update but before being reset be $\tilde{x}_{k|k,j}^- = \mu_{k|k,j}^- + a_{k|k}$ where $a_{k|k} \sim \mathcal{N}(0, P_{k|k}^{c-})$. Then under the assumption that $a_{k|k}$ is small and using the property of the right Jacobian defined in equation (15) add $\mu_{k|k,j}^-$ onto $\hat{x}_{k|k-}$

to get

$$x_{k|k} = \hat{x}_{k|k} - \text{Exp}_I^{G_x} \left(\mu_{k|k,j}^- + a_{k|k} \right) \quad (92a)$$

$$= \underbrace{\hat{x}_{k|k} - \text{Exp}_I^{G_x} \left(\mu_{k|k,j}^- \right)}_{\hat{x}_{k|k,j}} \text{Exp}_I^{G_x} \left(\underbrace{J_r^{G_x} \left(\mu_{k|k,j}^- \right) a_{k|k}}_{\tilde{x}_{k|k,j}} \right), \quad (92b)$$

where

$$\hat{x}_{k|k,j} = \hat{x}_{k|k} - \text{Exp}_I^{G_x} \left(\mu_{k|k,j}^- \right) \quad (93)$$

is the updated state estimate, and $\tilde{x}_{k|k,j} = J_r^{G_x} \left(\mu_{k|k,j}^- \right) a_{k|k}$ is the updated error state after reset. Equation (93) can be thought of as moving from $\hat{x}_{k|k}$ to $\hat{x}_{k|k,j}$ along the geodesic defined by the tangent vector $\mu_{k|k,j}^-$ as depicted in Fig. 10.

The error covariance of the error state $\tilde{x}_{k|k,j}$ is

$$\begin{aligned} \text{cov} \left(\tilde{x}_{k|k,j} \right) &= \text{cov} \left(J_r^{G_x} \left(\mu_{k|k,j}^- \right) a_{k|k} \right) \\ &= J_r^{G_x} \left(\mu_{k|k,j}^- \right) \text{cov} \left(a_{k|k} \right) J_r^{G_x} \left(\mu_{k|k,j}^- \right)^\top \\ &= J_r^{G_x} \left(\mu_{k|k,j}^- \right) P_{k|k}^c J_r^{G_x} \left(\mu_{k|k,j}^- \right)^\top \\ &= P_{k|k}^c. \end{aligned}$$

Therefore, $\tilde{x}_{k|k,j} \sim \mathcal{N} \left(\mu_{k|k,j} = 0, P_{k|k}^c \right)$.

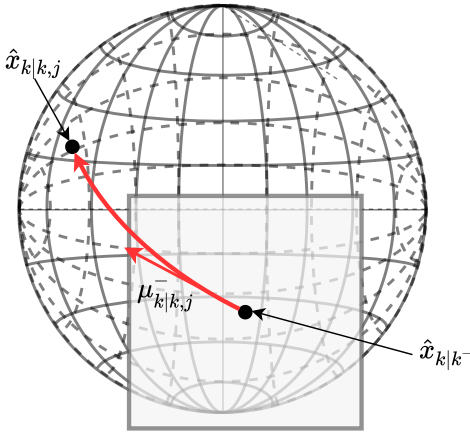


Fig. 10: A depiction of the state estimate update conditioned on $\theta_{k,j}$ by using $\mu_{k|k,j}^-$ to form a geodesic from $\hat{x}_{k|k}$ to $\hat{x}_{k|k,j}$.

Appendix D

Proof of Lemma 7: Track Likelihood

The probability of the track likelihood conditioned on the measurements $Z_{0:k}$ are inferred using the location of the validated measurements with respect to the track's estimated measurement $\hat{z}_k = h \left(\hat{x}_{k|k}, 0 \right)$ and the number of validated measurements. The basic idea is that the more likely at least one of the validated measurements is target originated compared to the likelihood that none of the

validated measurements are target originated, the more likely the track represents a target. Also, if the number of validated measurements is more than the expected number of false measurements inside the validation region, then the more likely one of the validated measurements originated from the target and the track represents the target.

We write the track likelihood conditioned on the measurements as

$$p \left(\epsilon_k \mid Z_{0:k} \right) = p \left(\epsilon_k \mid m_k, Z_k, Z_{0:k-} \right), \quad (94)$$

where we have separated Z_k and $Z_{0:k-}$ from $Z_{0:k}$ and explicitly written the inference on the number of validated measurements m_k . Using Bayes' rule, equation (94) can be written as

$$p \left(\epsilon_k \mid Z_{0:k} \right) = \frac{p \left(Z_k \mid \epsilon_k, m_k, Z_{0:k-} \right) p \left(\epsilon_k \mid m_k, Z_{0:k-} \right)}{p \left(Z_k \mid m_k, Z_{0:k-} \right)}, \quad (95)$$

where

$$\begin{aligned} p \left(Z_k \mid m_k, Z_{0:k-} \right) &= p \left(Z_k \mid \epsilon_k, m_k, Z_{0:k-} \right) p \left(\epsilon_k \mid m_k, Z_{0:k-} \right) \\ &\quad + p \left(Z_k \mid \epsilon_k = F, m_k, Z_{0:k-} \right) p \left(\epsilon_k = F \mid m_k, Z_{0:k-} \right) \end{aligned} \quad (96)$$

and $\epsilon_k = F$ denotes that the track does not represent the target.

The probability $p \left(Z_k \mid m_k, \epsilon_k, Z_{0:k-} \right)$ is derived in Appendix B and defined in equation (84). Accordingly

$$\begin{aligned} p \left(Z_k \mid \epsilon_k = F, m_k, Z_{0:k-} \right) &= \prod_{j=1}^{m_k} p \left(z_{k,j} \mid \epsilon_k = F, m_k, Z_{0:k-} \right) \\ &= \prod_{j=1}^{m_k} \mathcal{V}_k^{-1} = \mathcal{V}_k^{-m_k} \end{aligned} \quad (97)$$

because under the condition that the track does not represent the target, all of the validated measurements must be false, and the validated false measurements are assumed to be independent and uniformly distributed in the validation region with volume \mathcal{V}_k .

Using Bayes' rule we get

$$p \left(\epsilon_k \mid m_k, Z_{0:k-} \right) = \frac{p \left(m_k \mid \epsilon_k, Z_{0:k-} \right) p \left(\epsilon_k \mid Z_{0:k-} \right)}{p \left(m_k \mid Z_{0:k-} \right)} \quad (98)$$

and

$$\begin{aligned} p \left(\epsilon_k = F \mid m_k, Z_{0:k-} \right) &= \frac{p \left(m_k \mid \epsilon_k = F, Z_{0:k-} \right) p \left(\epsilon_k = F \mid Z_{0:k-} \right)}{p \left(m_k \mid Z_{0:k-} \right)}. \end{aligned} \quad (99)$$

The probability $p \left(m_k \mid \epsilon_k, Z_{0:k-} \right)$ is derived in Appendix B and defined in equation (82). The probability $p \left(m_k \mid \epsilon_k = F, Z_{0:k-} \right)$ is

$$p \left(m_k \mid \epsilon_k = F, Z_{0:k-} \right) = \mu_F \left(m_k \right), \quad (100)$$

since all of the measurements are false under the condition that the track does not represent a target.

Using the theorem of total probability with equations (82) and (100), the probability of the number of measurements m_k conditioned on the previous measurements is

$$p(m_k | Z_{0:k-}) = p(m_k | \epsilon_k, Z_{0:k-}) P_T + p(m_k | \epsilon_k = F, Z_{0:k-}) p(\epsilon_k | k-) P_T \quad (101a)$$

$$= P_G P_D \mu_F(m_k^-) P_T + (1 - P_D P_G p(\epsilon_k | k-)) \mu_F(m_k), \quad (101b)$$

where $P_T \doteq p(\epsilon_k | Z_{0:k-})$. Substituting equations (82) and (101) into equation (98) yields

$$p(\epsilon_k | m_k, Z_{0:k-}) = \frac{(P_G P_D \mu_F(m_k^-) + (1 - P_G P_D) \mu_F(m_k)) P_T}{P_G P_D \mu_F(m_k^-) P_T + (1 - P_D P_G P_T) \mu_F(m_k)}. \quad (102)$$

Using the fact that

$$p(\epsilon_k = F | m_k, Z_{0:k-}) = 1 - p(\epsilon_k | m_k, Z_{0:k-}),$$

and substituting the equations (102), (97), and (84) into equation (96) yields the probability of the validated measurements conditioned on the previous measurements

$$p(Z_k | m_k, Z_{0:k-}) = \frac{\mathcal{V}_k^{-m_k^-} P_D \mu_F(m_k^-) \sum_{\ell=1}^{m_k} p(z_{k,j} | \psi, \epsilon_k, Z_{0:k-}) P_T}{P_G P_D \mu_F(m_k^-) P_T + (1 - P_D P_G P_T) \mu_F(m_k)} + \frac{\mathcal{V}_k^{-m_k} (1 - P_D P_G P_T) \mu_F(m_k)}{P_G P_D \mu_F(m_k^-) P_T + (1 - P_D P_G P_T) \mu_F(m_k)}. \quad (103)$$

Substituting in equations (84), (102), and (103) into equation (95) and setting the density of false measurements to the Poisson distribution defined in equation (23) yields the probability of the track likelihood conditioned on the previous measurements given in equations (65) and (66).

References

- [1] Yaakov Bar-Shalom and Fortmaan Thomas E. *Tracking and Data Association*. Academic Press, Inc., 1988.
- [2] Nitin Bhatia and Vandana. “Survey of Nearest Neighbor Techniques”. In: *International Journal of Computer Science and Information Security* 8.2 (2010), pp. 302–305. arXiv: [1007.0085](https://arxiv.org/abs/1007.0085). URL: <http://arxiv.org/abs/1007.0085>.
- [3] P. Konstantinova, A. Udvarev, and T. Semerdjiev. “A Study of a Target Tracking Method Using Global Nearest Neighbor Algorithm”. In: *International Conference on Computer Systems and Technologies* (2003). ISSN: 0042-8469.
- [4] Yaakov Bar-Shalom, Peter Willett, and Xin Tian. *Tracking And Data Fusion: A Handbook of Algorithms*. YBS Publishing, 2011. ISBN: 9780964831278.

- [5] Abraham Wald. *Sequential Analysis*. New York: John Wiley & Sons Inc, 1947.
- [6] X. Rong Li, Ning Li, and Vesselin P. Jilkov. “SPRT-Based Track Confirmation and Rejection”. In: *Proceedings of the 5th International Conference on Information Fusion, FUSION 2002 2* (2002), pp. 951–958. DOI: [10.1109/ICIF.2002.1020914](https://doi.org/10.1109/ICIF.2002.1020914).
- [7] Samuel S. Blackman. “Multiple Hypothesis Tracking for Multiple Target Tracking”. In: *IEEE Aerospace and Electronic Systems Magazine* 19.1 II (2004), pp. 5–18. ISSN: 08858985. DOI: [10.1109/MAES.2004.1263228](https://doi.org/10.1109/MAES.2004.1263228).
- [8] Yaakov Bar-Shalom and Edison Tse. “Tracking in a Cluttered Environment with Probabilistic Data Association”. In: *Automatica* 11.5 (1975), pp. 451–460. ISSN: 00051098. DOI: [10.1016/0005-1098\(75\)90021-7](https://doi.org/10.1016/0005-1098(75)90021-7).
- [9] Yaakov Bar-Shalom, Fred Daum, and Jim Huang. “The Probabilistic Data Association Filter: Estimation in the Presence of Measurement Origin Uncertainty”. In: *IEEE Control Systems* 29.6 (2009), pp. 82–100. ISSN: 1066033X. DOI: [10.1109/MCS.2009.934469](https://doi.org/10.1109/MCS.2009.934469).
- [10] Joshua G. Mangelson et al. “Characterizing the Uncertainty of Jointly Distributed Poses in the Lie Algebra”. In: *arXiv* (2019), pp. 1–19. ISSN: 1552-3098. DOI: [10.1109/tro.2020.2994457](https://doi.org/10.1109/tro.2020.2994457). arXiv: [1906.07795](https://arxiv.org/abs/1906.07795).
- [11] Timothy D. Barfoot and Paul T. Furgale. “Associating Uncertainty with Three-Dimensional Poses for Use in Estimation Problems”. In: *IEEE Transactions on Robotics* 30.3 (2014), pp. 679–693. ISSN: 15523098. DOI: [10.1109/TRO.2014.2298059](https://doi.org/10.1109/TRO.2014.2298059).
- [12] Timothy Barfoot. *State Estimation For Robotics*. Cambridge University Press, 2019.
- [13] Joan Solà, Jeremie Deray, and Dinesh Atchuthan. *A Micro Lie theory for State Estimation in Robotics*. 2018. arXiv: [1812.01537](https://arxiv.org/abs/1812.01537). URL: <http://arxiv.org/abs/1812.01537>.
- [14] F. C. Park, J. E. Bobrow, and S. R. Ploen. “A Lie Group Formulation of Robot Dynamics”. In: *The International Journal of Robotics Research* 14.6 (1995), pp. 609–618. ISSN: 17413176. DOI: [10.1177/027836499501400606](https://doi.org/10.1177/027836499501400606).
- [15] Taeyoung Lee, Melvin Leok, and N Harris McClamroch. “Geometric Tracking Control of a Quadrotor UAV on SE(3)”. In: *49th IEEE Conference on Decision and Control (CDC)* (2010). DOI: [10.1109/CDC.2010.5717652](https://doi.org/10.1109/CDC.2010.5717652).
- [16] Andrew W. Long et al. “The Banana Distribution is Gaussian: A Localization Study with Exponential Coordinates”. In: *Robotics: Science and Systems* 8 (2013), pp. 265–272. ISSN: 2330765X. DOI: [10.15607/rss.2012.viii.034](https://doi.org/10.15607/rss.2012.viii.034).
- [17] Darko Mušicki, R. Evans, and S. Stankovic. “Integrated Probabilistic Data Association”. In: *IEEE Transactions on Automatic Control* 39.6 (1994), pp. 1237–1241.

- [18] Josip Cescic et al. “Radar and Stereo Vision Fusion for Multitarget Tracking on the Special Euclidean Group”. In: *Robotics and Autonomous Systems* 83 (2016), pp. 338–348. ISSN: 09218890. DOI: [10.1016/j.robot.2016.05.001](https://doi.org/10.1016/j.robot.2016.05.001). URL: <http://dx.doi.org/10.1016/j.robot.2016.05.001>.
- [19] Christoph Hertzberg et al. “Integrating Generic Sensor Fusion Algorithms with Sound State Representations through Encapsulation of Manifolds”. In: *Information Fusion* 14.1 (2013), pp. 57–77.
- [20] John Stillwell. *Naive Lie Theory*. Springer Science, 2008.
- [21] Brian C. Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. Springer-Verlag New York, Inc, 2003. ISBN: 0-387-40122-9.
- [22] Francesco Bullo, Andrew D. Lewis, and Bill Goodwine. *Geometric Control of Mechanical Systems*. Vol. 50. 12. New York: Springer Science, 2005, p. 2111. ISBN: 0-387-22195-6. DOI: [10.1109/TAC.2005.860277](https://doi.org/10.1109/TAC.2005.860277).
- [23] John M. Lee. *An Introduction to Smooth Manifolds*. Springer Science, 2013. ISBN: 978-1-4419-9981-8.
- [24] Ralph Abraham, Jerrold Marsden, and Tudor Ratiu. *Manifolds, Tensor Analysis, and Applications*. First. New York: Springer-Verlag, 1998. ISBN: 0-387-96790-7.
- [25] Manfredo Perdigao do Carmo. *Riemannian Geometry*. Birkhauser Boston, 1992. ISBN: 0-8176-3490-8.
- [26] Kenth Engø. “Partitioned Runge-Kutta Methods in Lie-Group Setting”. In: *Reports in Informatics* 43.202 (2000), pp. 21–39. URL: <http://www.ii.uib.no/publikasjoner/texrap/ps/2000-202.ps>.
- [27] Yunfeng Wang and Gregory S. Chirikjian. “Error Propagation on the Euclidean Group with Applications to Manipulator Kinematics”. In: *IEEE Transactions on Robotics* 22.4 (2006), pp. 591–602. ISSN: 15523098. DOI: [10.1109/TRO.2006.878978](https://doi.org/10.1109/TRO.2006.878978).
- [28] Youngji Kim and Ayoung Kim. “On the Uncertainty Propagation: Why Uncertainty on Lie Groups Preserves Monotonicity?” In: *IEEE International Conference on Intelligent Robots and Systems 2017-Septe* (2017), pp. 3425–3432. ISSN: 21530866. DOI: [10.1109/IROS.2017.8206181](https://doi.org/10.1109/IROS.2017.8206181).
- [29] Guillaume Bourmaud et al. “Continuous-Discrete Extended Kalman Filter on Matrix Lie Groups Using Concentrated Gaussian Distributions”. In: *Journal of Mathematical Imaging and Vision* 51.1 (2014), pp. 209–228. ISSN: 09249907. DOI: [10.1007/s10851-014-0517-0](https://doi.org/10.1007/s10851-014-0517-0).
- [30] Guillaume Bourmaud et al. “Discrete Extended Kalman Filter on Lie groups”. In: *European Signal Processing Conference* (2013), pp. 1–5. ISSN: 22195491.
- [31] D. J. Higham. “An Algorithmic Introduction to Numerical Simulation of Stochastic Differential Equations”. In: *SIAM Review* 43.3 (2001), pp. 525–546. ISSN: 00361445. DOI: [10.1137/S0036144500378302](https://doi.org/10.1137/S0036144500378302).
- [32] A. M. Sjøberg and O. Egeland. “An EKF for Lie Groups with Application to Crane Load Dynamics”. In: *Modeling, Identification and Control* 40.2 (2019), pp. 109–124. ISSN: 18901328. DOI: [10.4173/mic.2019.2.3](https://doi.org/10.4173/mic.2019.2.3).
- [33] Joan Solà. “Quaternion Kinematics for the Error-State Kalman Filter”. In: *CoRR* abs/1711.0 (2017). arXiv: [1711.02508](https://arxiv.org/abs/1711.02508). URL: <http://arxiv.org/abs/1711.02508>.
- [34] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation*. Vol. 9. 1216. 2001, p. 584. ISBN: 047141655X.
- [35] A. A. Gorji, R. Tharmarasa, and T. Kirubarajan. “Performance Measures for Multiple Target Tracking Problems”. In: *Fusion 2011 - 14th International Conference on Information Fusion* (2011), pp. 1560–1567.