On wrapping the Kalman filter and estimating with the SO(2) group

Ivan Marković, Josip Ćesić, Ivan Petrović
University of Zagreb, Faculty of Electrical Engineering and Computing, Unska 3, 10000 Zagreb, Croatia
Email: ivan.markovic@fer.hr, josip.cesic@fer.hr, ivan.petrovic@fer.hr

***Abstract*—This paper analyzes directional tracking in 2D with the extended Kalman filter on Lie groups (LG-EKF). The study stems from the problem of tracking objects moving in 2D Euclidean space, with the observer measuring direction only, thus rendering the measurement space and object position on the circle—a non-Euclidean geometry. The problem is further inconvenienced if we need to include higher-order dynamics in the state space, like angular velocity which is a Euclidean variables. The LG-EKF offers a solution to this issue by modeling the state space as a Lie group or combination thereof, e.g., SO(2) or its combinations with $\mathbb{R}^n$. In the present paper, we first derive the LG-EKF on SO(2) and subsequently show that this derivation, based on the mathematically grounded framework of filtering on Lie groups, yields the same result as heuristically wrapping the angular variable within the EKF framework. This result *applies only* to the SO(2) and SO$(2) \times \mathbb{R}^n$ LG-EKFs and is not intended to be extended to other Lie groups or combinations thereof. In the end, we showcase the SO$(2) \times \mathbb{R}^2$ LG-EKF, as an example of a constant angular acceleration model, on the problem of speaker tracking with a microphone array for which real-world experiments are conducted and accuracy is evaluated with ground truth data obtained by a motion capture system.**

I. Introduction

In moving object tracking, it is not uncommon to work with sensors that can provide only direction to the object in question. The measurement and estimation state space have a specific geometry of their own, which is different from the geometry of the true trajectory space. The problem is challenging, because, although the motion of the object resides either in 3D or 2D Euclidean space, corresponding measurements reside either on the sphere or the circle, respectively. Namely, if we are measuring and estimating only the direction to the object in 2D, i.e., the azimuth, the state and measurements will bear the non-Euclidean properties of angles. However, if we are to extend the state space so that it includes both the angular velocity and acceleration (which are Euclidean variables), so that we can apply a higher-order dynamic motion model, we are faced with constructing a 'hybrid' state space consisting of both the non-Euclidean and Euclidean variables.

There exist Bayesian methods based on the principle of assumed density filtering with directional distributions on the circle, namely the von Mises distribution, the wrapped Gaussian distribution and the Bingham distribution (which actually models variables with 180° symmetry), that capture intrinsically the non-Euclidean nature of angular random variables [1]–[6]. The benefit of these approaches is that they take globally into account the geometry of the state space. For example, in the case of the von Mises distribution it has been shown that the filter outperforms the naive Kalman filter, which treats angles like regular Euclidean variables, and the modified Kalman filter, which takes into account the nature of angles by wrapping them on the circle [4], [7]. However, extending the state space with additional variables of different geometry, e.g., to analytically model the azimuth with the von Mises distribution and the range or the angular velocity with the Gaussian distribution and capture correctly the cross-correlations, remains a challenge.

The SO(2) group is a set of orthogonal matrices with determinant one, whose elements geometrically represent rotations. This makes it an interesting candidate for estimation with angular variables. Furthermore, a filter could be derived not just for SO(2), but also for combinations of SO(2) with $\mathbb{R}$. This would enable us to create the aforementioned 'hybrid' state vector that would join both non-Euclidean and Euclidean variables within the same filter and enable a seamless utilization of higher-order system models with constant angular velocity or acceleration. An extended Kalman filter on matrix Lie groups was recently proposed in [8]. It provides us with a mathematical framework for solving the 'hybrid' state space problem. Indeed, the filter can be applied directly for any state that is a combination of Lie groups, since a Cartesian product of Lie groups is a Lie group [8]. However, it should be noted that the LG-EKF is a local approach, in the sense that it does not take globally the geometry of the state space into account, but locally captures the geometry of state space via exponential mapping. Another approach would be to model the whole state space as a Euclidean vector within the 'classical' Kalman filter framework, and wrap the operations involving angular variables. Indeed, this was performed in [4] to modify the unscented Kalman filter for angular state estimation, in [9] to take idiosyncrasies of directional statistics when using polar or spherical coordinates in the cubature Kalman filter, and in [7] to modify the Gaussian mixture probabilistic hypothesis density filter for multitarget tracking on a circle.

In this paper we propose to analyze the LG-EKF for directional tracking of moving objects in 2D. First, we look into deriving the LG-EKF on the SO(2), which also serves as a gentle introduction to the subject matter since the LG-EKF introduces non-trivial notation. Second, we model the directional moving object tracking in 2D as an estimation problem on the Lie group composed of the direct product SO$(2) \times \mathbb{R}^2$, i.e., a group that represents the moving object azimuth, angular velocity and angular acceleration. For the motion model, we use the constant angular acceleration model. In the end, we show that the $\mathrm{SO}(2)$ LG-EKF filter derivation based on the mathematically grounded framework of filtering on Lie groups yields the same result as heuristically wrapping the angular variables within the extended Kalman filter (EKF) framework. Since for the case of $\mathbb{R}^n$ the LG-EKF evaluates to EKF [8], this results also extends to $\mathrm{SO}(2) \times \mathbb{R}^2$ LG-EKF and an $\mathbb{R}^3$ EKF when wrapping the angular component. Please note that this result *applies only* to the $\mathrm{SO}(2)$ filter and is not intended to be extended to other Lie groups or combinations thereof. Indeed, given that $\mathrm{SO}(2)$ is abelian, i.e., commutative, the result does not seem unexpected, but we assert that it gives interesting theoretical perspective on estimation and tracking with the heuristically modified EKF. Before we proceed with the filter derivation, we introduce some necessary formal definitions and operators for working with matrix Lie groups.

II. Mathematical background

A. Wrapping the Kalman filter

In this section we shall assume that wrapping operation amounts to enforcing the angular variable to be in the $[-\pi, \pi]$ interval, and we designate this operation as follows

$$\mathrm{w}^{\pi}(x) = \mathrm{mod}(x + \pi, 2\pi) - \pi. \tag{1}$$

Note that when computing the difference between two angular variables, the wrapping effect of the circle should be taken into account, e.g., the difference between 178° and −178° should evaluate to 4°. This is also achieved by (1) when the difference is given as the argument, i.e., difference between two angles x and y is computed as $\mathrm{w}^{\pi}(x - y)$.

Let us assume the following system model

$$x_{k+1|k} = f_k(x_k, u_k) + n_k, \quad n_k \sim \mathcal{N}(0, Q) \tag{2}$$

where x_k is the system state, u_k is the control input, n_k is process noise, and $f_k(\,\cdot\,)$ is the non-linear system state equation. In the EKF the idiosyncrasies of angular data appear most prominently in the correction step when calculating the innovation, which should be computed as

$$x_{k+1} = x_{k+1|k} + K_k \,\mathrm{w}^{\pi}(z_k - h_k(x_{k+1|k})), \tag{3}$$

where K_k is the Kalman gain, z_k is the measurement, and $h_k(\,\cdot\,)$ is the non-linear measurement equation.

To demonstrate this, let us take a simple example of having an identity measurement equation, $x_{k+1|k} = 358°$, $z_k = 2°$ and $K_k = 0.5$. If we would not wrap the innovation, the updated state would yield a clearly incorrect result of $x_{k+1} = 180°$ inlieu of $x_{k+1} = 360°$. For practical purposes, after correction and prediction the system state can be checked to the required interval by computing $x_{k+1} \leftarrow \mathrm{w}^{\pi}(x_{k+1})$. In the sequel when we refer to the modified Kalman filter, it entails treating angular variables with the previously introduced operation. Furthermore, we assume that the reader is familiar with EKF equations, which we will not present or derive explicitly in order to keep the brevity of the paper.

B. Lie Groups

A Lie group G is a group which is also a smooth manifold and the group composition and inverse are smooth functions on the manifold G. A manifold is an object that looks locally like a piece of $\mathbb{R}^n$ and G is 'smooth' in the sense that is has a tangent space, of the appropriate dimension, at each point. Take for example the circle, a curve in $\mathbb{R}^2$ which looks locally (but not globally) like $\mathbb{R}^1$. For a matrix Lie group the composition and inverse are simply matrix multiplication and inversion, with the identity element $\mathbf{I}^{n\times n}$ [10].

A Lie Algebra $\mathfrak{g}$ is an open neighborhood of $\mathbf{0}^{n\times n}$ in the tangent space of G at the identity $\mathbf{I}^{n\times n}$. The matrix exponential $\exp_{\mathrm{G}}$ and matrix logarithm $\log_{\mathrm{G}}$ establish a local diffeomorphism between Lie groups and Lie algebras [8]

$$\exp_{\mathrm{G}} : \mathfrak{g} \to \mathrm{G}, \quad \log_{\mathrm{G}} : \mathrm{G} \to \mathfrak{g}. \tag{4}$$

The Lie Algebra $\mathfrak{g}$ associated to a p-dimensional matrix Lie group $\mathrm{G} \subset \mathbb{R}^{n\times n}$ is a p-dimensional vector space [10]. A linear isomorphism between $\mathfrak{g}$ and $\mathbb{R}^p$ is given by

$$[\cdot]^{\vee}_{\mathrm{G}} : \mathfrak{g} \to \mathbb{R}^p, \quad [\cdot]^{\wedge}_{\mathrm{G}} : \mathbb{R}^p \to \mathfrak{g}. \tag{5}$$

Lie Groups are not necessarily commutative. The following two operators capture this property

- the adjoint representation of G on $\mathbb{R}^p$

$$\mathrm{Ad}_{\mathrm{G}} : \mathrm{Ad}_{\mathrm{G}}(X)x = \left[X[x]^{\wedge}_{\mathrm{G}} X^{-1}\right]^{\vee}_{\mathrm{G}} \tag{6}$$

- the adjoint representation of $\mathbb{R}^p$ on $\mathbb{R}^p$

$$\mathrm{ad}_{\mathrm{G}} : \mathrm{ad}_{\mathrm{G}}(x)y = [[x]^{\wedge}_{\mathrm{G}}[y]^{\wedge}_{\mathrm{G}} - [y]^{\wedge}_{\mathrm{G}}[x]^{\wedge}_{\mathrm{G}}]^{\vee}_{\mathrm{G}} \tag{7}$$

where $x, y \in \mathbb{R}^p$. In the sequel, these operators, the exponential and logarithmic mapping are given concrete form for the pertinent Lie groups.

C. The $SO(2)$ group

In this example our system state (azimuth of the tracked object) is modeled as the group $G = \mathrm{SO}(2)$, i.e., as the rotation matrix $X_k = R_{\theta_k}$

$$R_{\theta_k} = \begin{bmatrix} \cos\theta_k & -\sin\theta_k \\ \sin\theta_k & \cos\theta_k \end{bmatrix}. \tag{8}$$

The composition and inverse in $\mathrm{SO}(2)$ are simply evaluated as $X_1X_2 = R_1R_2, X^{-1} = R^{\mathrm{T}}$. For this case the associated Lie algebra which bridges $X_k \in \mathrm{G}$ and $x_k = \theta_k \in \mathbb{R}^1$ is $\mathfrak{g} = \mathfrak{so}(2)$, and the following holds

$$[\theta_k]^{\wedge}_{\mathrm{G}} = \begin{bmatrix} 0 & -\theta_k \\ \theta_k & 0 \end{bmatrix}. \tag{9}$$

The link between $\mathrm{SO}(2)$ and $\mathfrak{so}(2)$ is given by the exponential and logarithmic mapping

$$\exp_{\mathrm{G}}([\theta_k]^{\wedge}_{\mathrm{G}}) = R_{\theta_k} \;:\; \mathfrak{so}(2) \to \mathrm{SO}(2), \tag{10}$$

$$\log_{\mathrm{G}}(R_{\theta_k}) = [\theta_k]^{\wedge}_{\mathrm{G}} \;:\; \mathrm{SO}(2) \to \mathfrak{so}(2). \tag{11}$$

Due to the commutativity of $\mathrm{SO}(2)$, the adjoint operators are

$$\mathrm{ad}_{\mathrm{G}}(\theta_k) = 0, \quad \mathrm{Ad}_{\mathrm{G}}\left(\exp_{\mathrm{G}}\left([\theta_k]^{\wedge}_{\mathrm{G}}\right)\right) = 1. \tag{12}$$

These properties greatly simplify the LG-EKG formulae for the $\mathrm{SO}(2)$ group which will become evident in the sequel.

D. The $SO(2) \times \mathbb{R}^2$ group

In this section we propose to model the system state as the Cartesian product of groups $\mathrm{G} = \mathrm{SO}(2) \times \mathbb{R}^2$. This is a slight abuse of notation intended for clarity, since when talking about $\mathbb{R}$ within the group or algebra, we are actually referring to the group of algebra representation of $\mathbb{R}$, for which the explicit representation is given further in the paper. The moving object state X_k will represent the azimuth of the target as a rotation matrix $R_{\theta_k} \in \mathrm{SO}(2)$, angular velocity as a real number $\omega_k \in \mathbb{R}$, and angular acceleration also as a real number $\alpha_k \in \mathbb{R}$. The system state X_k can be symbolically represented as

$$X_k = \begin{bmatrix} R_k & & \\ & \begin{bmatrix} 1 & \omega_k \\ 0 & 1 \end{bmatrix} & \\ & & \begin{bmatrix} 1 & \alpha_k \\ 0 & 1 \end{bmatrix} \end{bmatrix} = \begin{pmatrix} R_k \\ \omega_k \\ \alpha_k \end{pmatrix}_{\mathrm{G}} . \qquad (13)$$

Note that composition and inverse on such a group is evaluated as follows

$$X_1 X_2 = \begin{pmatrix} R_1 R_2 \\ \omega_1 + \omega_2 \\ \alpha_1 + \alpha_2 \end{pmatrix}_{\mathrm{G}} , \quad X^{-1} = \begin{pmatrix} R^{\mathrm{T}} \\ -\omega \\ -\alpha \end{pmatrix}_{\mathrm{G}} . \qquad (14)$$

The associated Lie algebra is $\mathfrak{g} = \mathfrak{so}(2) \times \mathbb{R}^2$ which bridges the state on the Lie group $X_k \in \mathrm{G}$ with the vector $x_k = [\theta_k \ \omega_k \ \alpha_k]^{\mathrm{T}} \in \mathbb{R}^3$, and the following holds

$$[x_k]^{\wedge}_{\mathrm{G}} = \begin{bmatrix} [\theta_k]^{\wedge}_{\mathrm{SO}(2)} & & \\ & [\omega_k]^{\wedge}_{\mathbb{R}} & \\ & & [\alpha_k]^{\wedge}_{\mathbb{R}} \end{bmatrix} = \begin{pmatrix} [\theta_k]^{\wedge}_{\mathrm{SO}(2)} \\ \omega_k \\ \alpha_k \end{pmatrix}_{\mathfrak{g}} , \qquad (15)$$

where $[\theta_k]^{\wedge}_{\mathrm{SO}(2)}$ is given by (9), while

$$[\omega_k]^{\wedge}_{\mathbb{R}} = \begin{bmatrix} 0 & \omega_k \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad [\alpha_k]^{\wedge}_{\mathbb{R}} = \begin{bmatrix} 0 & \alpha_k \\ 0 & 0 \end{bmatrix} . \qquad (16)$$

The link between the group G and the associated algebra $\mathfrak{g}$ is defined by the exponential mapping

$$\exp_{\mathrm{G}}\left([x_k]^{\wedge}_{\mathrm{G}}\right) = \begin{pmatrix} \exp_{\mathrm{SO}(2)}\left([\theta_k]^{\wedge}_{\mathrm{SO}(2)}\right) \\ \omega_k \\ \alpha_k \end{pmatrix}_{\mathrm{G}} = \begin{pmatrix} R_k \\ \omega_k \\ \alpha_k \end{pmatrix}_{\mathrm{G}} , \qquad (17)$$

and logarithmic mapping

$$\log_{\mathrm{G}}(X_k) = \begin{pmatrix} \log_{\mathrm{SO}(2)}(R_k) \\ \omega_k \\ \alpha_k \end{pmatrix}_{\mathfrak{g}} = \begin{pmatrix} [\theta_k]^{\wedge}_{\mathrm{SO}(2)} \\ \omega_k \\ \alpha_k \end{pmatrix}_{\mathfrak{g}} . \qquad (18)$$

Furthermore, since $\mathrm{SO}(2)$ and $\mathbb{R}$ are abelian and the Cartesian product of abelian groups is abelian, the adjoint operators are again trivial

$$\mathrm{ad}_{\mathrm{G}}(x_k) = \mathbf{0}^{3\times 3}, \quad \mathrm{Ad}_{\mathrm{G}}\left(\exp_{\mathrm{G}}\left([x_k]^{\wedge}_{\mathrm{G}}\right)\right) = \mathbf{I}^{3\times 3}. \qquad (19)$$

III. The EKF on Matrix Lie Groups

As in the case of classical Kalman filtering, we need to begin by defining a motion model by which we will calculate the prediction. For general filtering on matrix Lie groups, the system model is defined by the following equation [8]

$$X_{k+1} = f(X_k, u_k, n_k) = X_k \exp_{\mathrm{G}}\left([\hat{\Omega}_k + n_k]^{\wedge}_{\mathrm{G}}\right) , \qquad (20)$$

where $X_k \in \mathrm{G}$ is the system state at time k, G is a p-dimensional Lie Group, $n_k \sim \mathcal{N}_{\mathbb{R}^p}(\mathbf{0}^{p\times 1}, Q_k)$ is white Gaussian noise and $\hat{\Omega}_k = \Omega(X_k, u_k) : \mathrm{G} \times \mathbb{R}^w \to \mathbb{R}^p$ is the system state equation which describes how the model acts on the state and control input in order to calculate the displacement.

Note that the function of $\hat{\Omega}_k$ is to take the system state which resides on G and the control input which resides on $\mathbb{R}^w$, calculate the displacement by applying the system model, and then transfer the displacement to the vector space $\mathbb{R}^p$ where additive noise is added. This displacement is then transferred to the associated Lie algebra by the $[\,.\,]^{\wedge}_{\mathrm{G}}$ operator and then exponentially mapped back to the Lie group to be added by way of composition to the system state X_k. Given that, a question arises how to implement a specific system model, since in LG-EKF it operates through a displacement? That is, how to construct $\hat{\Omega}_k$ from $f_k(x_k, u_k)$? The first step would be to write the system equation as $f_k(x_k, u_k) = x_k + \hat{f}_k(x_k, u_k)$ which can then be practically 'translated' to appropriate $\hat{\Omega}_k$. Note that generality is not lost here since $-x_k$ can be included within $\hat{f}_k(x_k, u_k)$,

The prediction step of the LG-EKF is governed by the following formulae [8]

$$\mu_{k+1|k} = \mu_k \exp_{\mathrm{G}}\left([\hat{\Omega}_k]^{\wedge}_{\mathrm{G}}\right) \qquad (21)$$

$$P_{k+1|k} = \mathcal{F}_k P_k \mathcal{F}_k^T + \Phi_{\mathrm{G}}(\hat{\Omega}_k) Q_k \Phi_{\mathrm{G}}(\hat{\Omega}_k)^T , \qquad (22)$$

where $\mu_k \in \mathrm{G}$ is the estimated mean value of the system state X_k, $P_k \in \mathbb{R}^{p\times p}$ is the estimated covariance matrix, while other terms are non-trivially calculated matrices

$$\mathcal{F}_k = \mathrm{Ad}_{\mathrm{G}}\left(\exp_{\mathrm{G}}\left([-\hat{\Omega}_k]^{\wedge}_{\mathrm{G}}\right)\right) + \Phi_{\mathrm{G}}(\hat{\Omega}_k)\mathscr{C}_k, \qquad (23)$$

$$\Phi_{\mathrm{G}}(\nu) = \sum_{m=0}^{\infty} \frac{(-1)^m}{(m+1)!} \mathrm{ad}_{\mathrm{G}}(\nu)^m, \quad \nu \in \mathbb{R}^p, \qquad (24)$$

$$\mathscr{C}_k = \frac{\partial}{\partial \epsilon} \Omega\left(\mu_k \exp_{\mathrm{G}}\left([\epsilon]^{\wedge}_{\mathrm{G}}\right), u_{k-1}\right)_{|\epsilon=0} . \qquad (25)$$

The parameter $\epsilon \in \mathbb{R}^p$ can be seen as a *Lie algebraic error* which is approximated as being distributed according to a classical Euclidean Gaussian distribution $\epsilon \sim \mathcal{N}_{\mathbb{R}^p}(\mathbf{0}^{p\times 1}, P_k)$. It is interesting to note that the mean value μ_k resides on the Lie group G, while the covariance matrix P_k describes uncertainty in $\mathbb{R}^p$. Although at first this appears peculiar, it is a consequence of modeling the uncertainty of states on Lie groups by the assumption of the concentrated Gaussian distribution $X_k \sim \mathcal{G}(\mu_k, P_k)$. In essence, the state resides on the group, but its uncertainty resides on the tangential vector space. For a more formal introduction of this concept, please confer [8].

The discrete measurement model on the matrix Lie Group is given as follows

$$z_{k+1} = h(X_{k+1}) \exp_{\mathrm{G}'}\left([m_{k+1}]^{\wedge}_{\mathrm{G}'}\right) , \tag{26}$$

where $z_{k+1} \in \mathrm{G}'$, $h : \mathrm{G} \to \mathrm{G}'$, and $m_{k+1} \sim \mathcal{N}_{\mathbb{R}^q}(\mathbf{0}^{q\times 1}, R_k)$ is white Gaussian noise. Note that here a different group G' is used since the system state and measurements might belong to different groups. Having the measurement model defined, we can proceed now to the update step which will first constitute the calculation of the Kalman gain

$$K_{k+1} = P_{k+1|k}\mathcal{H}^T_{k+1}\left(\mathcal{H}_{k+1}P_{k+1|k}\mathcal{H}^T_{k+1} + R_{k+1}\right)^{-1} , \tag{27}$$

where the measurement matrix $\mathcal{H}_{k+1}$ is calculated via

$$\begin{aligned}\mathcal{H}_{k+1} = \frac{\partial}{\partial\epsilon}\Big[\log_{\mathrm{G}'}\Big(h(\mu_{k+1|k})^{-1} \\ h\left(\mu_{k+1|k}\exp_{\mathrm{G}}\left([\epsilon]^{\wedge}_{\mathrm{G}}\right)\right)\Big)\Big]^{\vee}_{\mathrm{G}\ |\epsilon=0} .\end{aligned} \tag{28}$$

Furthermore, the innovation vector multiplied by Kalman gain is computed as

$$\nu_{k+1} = K_{k+1}\left[\log_{\mathrm{G}'}\left(h(\mu_{k+1|k})^{-1}z_{k+1}\right)\right]^{\vee}_{\mathrm{G}'} . \tag{29}$$

Finally, the update of the system state and covariance matrix can be evaluated as [8]

$$\mu_{k+1} = \mu_{k+1|k}\exp_{\mathrm{G}}\left([\nu_{k+1}]^{\wedge}_{\mathrm{G}}\right) \tag{30}$$

$$P_{k+1} = \Phi_{\mathrm{G}}(\nu_{k+1})\left(\mathbf{I}^{p\times p} - K_{k+1}\mathcal{H}_{k+1}\right)P_{k+1|k}\Phi_{\mathrm{G}}(\nu_{k+1})^T . \tag{31}$$

We can notice similarities between the LG-EKF and EKF equations and, indeed, when G and G′ are Euclidean spaces the LG-EKF reduces to EKF [8]. Furthermore, due to the results (12) and (19), matrices $\mathcal{F}_k$ and $\Phi_{\mathrm{G}}(\nu)$ for both SO(2) and $\mathrm{SO}(2)\times\mathbb{R}^2$ evaluate to

$$\Phi_{\mathrm{G}}(\nu) = \mathbf{I}, \quad \mathcal{F}_k = \mathbf{I} + \mathscr{C}_k. \tag{32}$$

In the sequel we derive the LG-EKF for the groups which we propose to utilize for tracking of moving objects with angular measurements and show that in this special case the LG-EKF reduces to the heuristically modified EKF.

A. LG-EKF on SO(2)

In this section we derive the LG-EKF filter for state estimation on $\mathrm{G} = \mathrm{SO}(2)$. For this group, mathematically dense LG-EKF equations are simplified and serve well to intuitively grasp the mechanics of the filter.

1) Prediction: Let us take two examples of system models. In the first we assume a stationary process, i.e., in the prediction the mean value will remain unchanged except for the uncertainty that is added through the process noise (this is similar to the von Mises filter [1])

$$x_{k+1|k} = x_k + n_k, \quad n_k \sim \mathcal{N}_{\mathbb{R}^1}(0, \sigma^2_Q). \tag{33}$$

This yields the LG-EKF system model $\Omega(X_k) = 0$ with the same process noise n_k, which when inserted in (21) will evaluate through the exponential as an identity matrix, thus leaving the mean value unperturbed.

In order to compute the prediction of the covariance matrix via (22), given the result in (32), we only need to determine $\mathscr{C}_k$. In this case the Lie algebraic error is $\epsilon \in \mathbb{R}^1$ and due to the system model the matrix $\mathscr{C}_k$ evaluates to zero, thus leaving $\mathcal{F}_k = 1$, and the prediction equations are

$$\mu_{k+1|k} = \mu_k, \quad P_{k+1|k} = P_k + Q_k. \tag{34}$$

As we can see, these are the same formulae that an EKF prediction would yield with (33) as the system model.

As the second example, we take the non-linear system [4] where the robot rotary joint angle was estimated

$$x_{k+1|k} = x_k + c_1\sin(x_k) + c_2 + n_k, \tag{35}$$

where second and third term account for gravity and velocity, while the final term is again one-dimensional white Gaussian noise. This yields the following LG-EKF system model

$$\Omega(X_k) = c_1\sin([\log(X_k)]^{\vee}_{\mathrm{G}}) + c_2. \tag{36}$$

Note that $[\log(X_k)]^{\vee}_{\mathrm{G}}$ is necessary to bring the rotation matrix with parameter μ_k to a scalar angle in $\mathbb{R}^1$. The Lie algebraic error is again $\epsilon \in \mathbb{R}^1$ and given the system model (36) matrix $\mathscr{C}_k$ evaluates to

$$\begin{aligned}\mathscr{C}_k &= \frac{\partial}{\partial\epsilon}\Omega\left(\begin{bmatrix}\cos\mu_k & -\sin\mu_k \\ \sin\mu_k & \cos\mu_k\end{bmatrix}\begin{bmatrix}\cos\epsilon & -\sin\epsilon \\ \sin\epsilon & \cos\epsilon\end{bmatrix}\right)_{|\epsilon=0} \\ &= \frac{\partial}{\partial\epsilon}\Omega\left(\begin{bmatrix}\cos(\mu_k+\epsilon) & -\sin(\mu_k+\epsilon) \\ \sin(\mu_k+\epsilon) & \cos(\mu_k+\epsilon)\end{bmatrix}\right)_{|\epsilon=0} \\ &= \frac{\partial}{\partial\epsilon}\left(c_1\sin(\mu_k+\epsilon)+c_2\right)_{|\epsilon=0} = c_1\cos\mu_k\end{aligned} \tag{37}$$

This means that $\mathcal{F}_k = 1 + c_1\cos\mu_k$, and that the LG-EKF prediction equations are

$$\begin{aligned}\mu_{k+1|k} &= \mu_k\exp_{\mathrm{G}}\left([c_1\sin([\log(X_k)]^{\vee}_{\mathrm{G}}) + c_2]^{\wedge}_{\mathrm{G}}\right) \\ P_{k+1|k} &= P_k(1 + c_1\cos\mu_k)^2 + Q_k.\end{aligned} \tag{38}$$

We can see that the covariance prediction formula is identical to the EKF covariance prediction.

More generally, to demonstrate the equivalence of the modified EKF and SO(2) LG-EKF prediction steps we need to show that $\mathcal{F}_k = 1 + \mathscr{C}_k$ is equal to

$$F_k = \frac{\partial f_k(x_k, u_k)}{\partial x_k}_{|x_k=\mu_k}, \tag{39}$$

where F_k is the state transition matrix, i.e., the EKF system state Jacobian of (2). By inspecting (37) we can notice that for SO(2) the argument within Ω will always be the sum of the mean value and the Lie algebraic error $\mu_k + \epsilon$. This gives

$$\begin{aligned}\mathcal{F}_k &= 1 + \frac{\partial}{\partial\epsilon}\Omega(\exp_{\mathrm{G}}([\mu_k+\epsilon]^{\wedge}_{\mathrm{G}}), u_k)_{|\epsilon=0} \\ &= 1 + \frac{\partial}{\partial\epsilon}\hat{f}_k(\mu_k+\epsilon, u_k)_{|\epsilon=0} \\ &= 1 + \frac{\partial}{\partial\epsilon}(f_k(\mu_k+\epsilon, u_k) - (\mu_k+\epsilon_\theta))_{|\epsilon=0} \\ &= \frac{\partial f_k(\mu_k+\epsilon, u_k)}{\partial\epsilon}_{|\epsilon=0} = \frac{\partial f_k(\xi_k, u_k)}{\partial\xi_k}_{|\xi_k=\mu_k},\end{aligned} \tag{40}$$

where variable substitution was performed in the last step: $\xi_k \leftarrow \mu_k + \epsilon, \partial\xi_k \leftarrow \partial\epsilon$. In the end $\mathcal{F}_k$ evaluates to the EKF Jacobian F_k when the underlying group is $\mathrm{SO}(2)$.

2) Correction: Since we are measuring angles, we define the measurement Lie group as $\mathrm{G}' = \mathrm{SO}(2)$ and the measurement function $h : \mathrm{SO}(2) \to \mathrm{SO}(2)$

$$h(X_{k+1}) = R_{k+1}, \quad m_{k+1} \sim \mathcal{N}_{\mathbb{R}^1}(0, \sigma_R^2), \tag{41}$$

which is trivial since the measurement and state group are the same, while the measurement noise is a one-dimensional white Gaussian noise. As in the prediction step, the associated Lie algebra is $\mathfrak{g}' = \mathfrak{so}(2)$.

To compute the correction step, we need to evaluate (28) for the LG-EKF on $\mathrm{SO}(2)$. The composition of the predicted mean $\mu_{k+1|k}$ and the Lie algebraic error yields

$$h(\mu_{k+1|k} \exp_{\mathrm{SO}(2)}([\epsilon]^\wedge_{\mathrm{SO}(2)})) = \begin{bmatrix} \cos(\mu_{k+1|k} + \epsilon) & -\sin(\mu_{k+1|k} + \epsilon) \\ \sin(\mu_{k+1|k} + \epsilon) & \cos(\mu_{k+1|k} + \epsilon) \end{bmatrix}. \tag{42}$$

Since $h(\mu_{k+1|k})^{-1}$ is the transpose of the corresponding rotation matrix, by inserting these results in (28) we can calculate the measurement matrix

$$\mathcal{H}_{k+1} = \frac{\partial}{\partial\epsilon}\left(\left[\log_{\mathrm{G}}\left(\begin{bmatrix} \cos\epsilon & -\sin\epsilon \\ \sin\epsilon & \cos\epsilon \end{bmatrix}\right)\right]^\vee_{\mathrm{G}}\right)_{|\epsilon=0} = 1 \tag{43}$$

Given this results it is straightforward to see that Kalman gain and covariance update equation of $\mathrm{SO}(2)$ LG-EKF are equal to the EKF equations. The state correction equations yield the same result, except that the LG-EKF takes wrapping into account by composition of rotation matrices, while the modified EKF computes everything in $\mathbb{R}^1$ and would need to wrap the corrected state with $\mathrm{w}^\pi(\,\cdot\,)$.

B. LG-EKF on SO$(2) \times \mathbb{R}^2$

In this section we derive the LG-EKF filter for state estimation on $\mathrm{G} = \mathrm{SO}(2) \times \mathbb{R}^2$. Given the demonstrated equality of the $\mathrm{SO}(2)$ LG-EKF and the modified EKF and that LG-EKF reduces to EKF for Euclidean spaces, it is intuitive to expect that this result would extend to groups derived by composing $\mathrm{SO}(2)$ with Euclidean spaces. In the sequel we illustrate this property by deriving a constant angular acceleration model for tracking with angle-only measurements.

1) Prediction: Given the state representation, we can now define the system model. For this purpose, we use the constant angular acceleration model $\hat{\Omega}_k = \Omega(X_k) : G \to \mathbb{R}^3$

$$\Omega(X_k) = \begin{bmatrix} T\omega_k + \frac{1}{2}T^2\alpha_k \\ T\alpha_k \\ 0 \end{bmatrix}, \quad n_k \sim \mathcal{N}_{\mathbb{R}^3}(0, Q). \tag{44}$$

Note that the displacement due to motion is calculated first in $\mathbb{R}^3$ and then according to (21) transferred to Lie algebra $\mathfrak{g}$, exponentially mapped to the group G and then by way of composition added to the system state X_k.

In this case, the Lie algebraic error is $\epsilon = [\epsilon_\theta\ \epsilon_\omega\ \epsilon_\alpha]^{\mathrm{T}} \in \mathbb{R}^3$, hence the composition of the mean value μ_k and ϵ yields

$$\mu_k \exp_{\mathrm{G}}([\epsilon]^\wedge_{\mathrm{G}}) = \begin{pmatrix} R_k R_\epsilon \\ \omega_k + \epsilon_\omega \\ \alpha_k + \epsilon_\alpha \end{pmatrix}_{\mathrm{G}}, \tag{45}$$

with $R_k R_\epsilon$ has the same for as the matrix product in (42). By applying the motion model (44) on this results we get

$$\Omega(\mu_k \exp_{\mathrm{G}}([\epsilon]^\wedge_{\mathrm{G}})) = \begin{bmatrix} T(\omega + \epsilon_\omega) + \frac{1}{2}T^2(\alpha + \epsilon_\alpha) \\ T(\alpha + \epsilon_\alpha) \\ 0 \end{bmatrix}. \tag{46}$$

To compute the prediction step for the covariance matrix, we need to calculate matrix $\mathcal{F}_k$. Since adjoint operators are trivial, using (46) we calculate

$$F_k = \mathbf{I} + \mathscr{C}_k = \begin{bmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}. \tag{47}$$

We can see that the matrix $\mathcal{F}_k$ evaluates to the well known transition matrix of the classical EKF constant acceleration motion model.

2) Correction: Since here we track moving objects by measuring angles, we define the measurement Lie group $\mathrm{G}' = \mathrm{SO}(2)$ and the measurement function $h : \mathrm{SO}(2) \times \mathbb{R}^2 \to \mathrm{SO}(2)$

$$h(X_{k+1}) = R_{k+1}, \quad m_{k+1} \sim \mathcal{N}_{\mathbb{R}^1}(0, \sigma_R^2) \tag{48}$$

which in this case simply extracts the rotation matrix R_{k+1} from X_{k+1}. Calculation of the matrix $\mathcal{H}_{k+1}$ is the same as in (28), except that ϵ is now a vector

$$\mathcal{H}_{k+1} = \frac{\partial}{\partial\epsilon}\left(\epsilon_\theta\right)_{|\epsilon=0} = [1\ 0\ 0]. \tag{49}$$

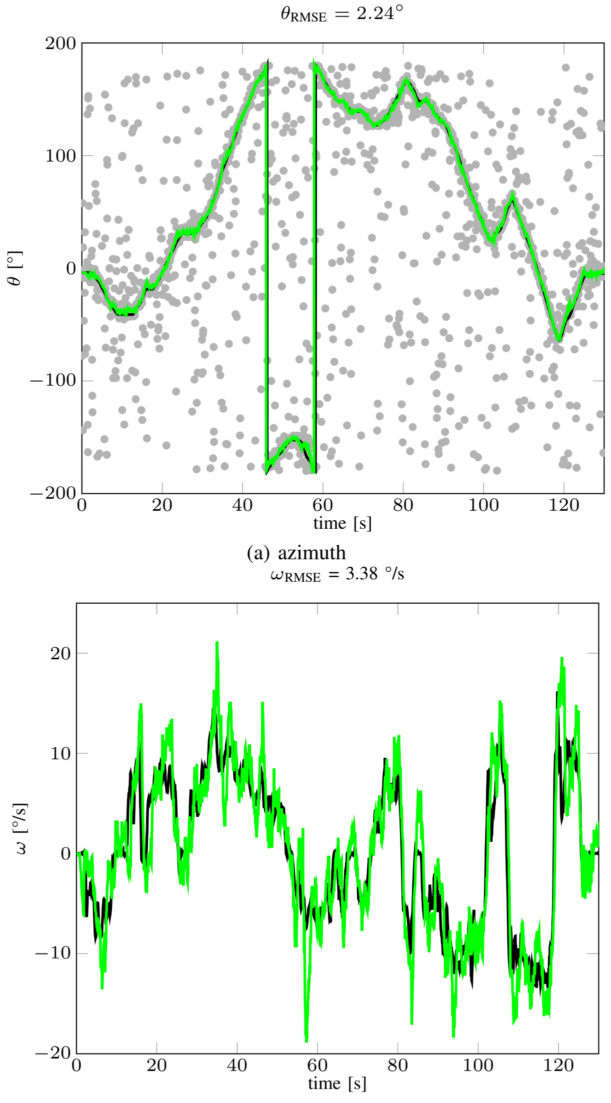
Again, the same result as we would expect for the EKF measurement matrix.

IV. Experiments

As a practical example of an application of the studied filter we apply this on the problem of speaker tracking with a microphone array and in the present paper we test the $\mathrm{SO}(2) \times \mathbb{R}^2$ LG-EKF on real-world data. For the sound acquisition we used the ManyEars framework consisting of an 8-channel USB sound card [11], while for obtaining measurements we used the beamforming algorithm for speaker localization [12] implemented under the Robot Operating Systems [13] within the same framework. The maximum of the beamforming energy was picked as the speaker measurement.

The experiments were conducted in a $120\,\mathrm{m}^2$ room with parquet wooden flooring and one side covered with windows. The speaker was simulated by a loudspeaker playing an excerpt from Nature's podcast *Audiophile* in English. The area in which the loudspeaker moved was covered by a motion capture system, which was used to generate ground truth data. In order to handle outliers, we used validation gating; namely, the innovation matrix $S_{k+1} = \mathcal{H}_{k+1}P_{k+1|k}\mathcal{H}^{\mathrm{T}}_{k+1} + R_{k+1}$ was calculated and we applied the standard χ^2–test

$$\nu_{k+1}S^{-1}_{k+1}\nu^{\mathrm{T}}_{k+1} < \gamma, \tag{50}$$



(a) azimuth

(b) angular velocity

Fig. 1: Performance of the LG-EKF on $\mathrm{SO}(2) \times \mathbb{R}^2$ when tracking a moving speaker. The solid black line is the ground truth as given by the motion capture system, the green solid line is the estimated state of the speaker, while the gray circles represent measurements, i.e. outputs of the beamformer. State RMSE is given in the title of each of the subfigures.

where the threshold γ was determined from the inverse χ^2_p cumulative distribution at a significance level $a = 0.95$ and p degrees-of-freedom. Figure 1 shows the experiment results and corroborates that the filter successfully manages to track the moving speaker in spite of the number of outliers. Note that the modified EKF would yield the same results, except that in the case of the LG-EKF the system state was defined on $\mathrm{SO}(2) \times \mathbb{R}^2$ and the idiosyncrasies of angular data were intrinsically taken care of.

V. Conclusion

In this paper we have studied directional moving object tracking in 2D based on the extended Kalman filter on matrix Lie groups. First, we have proposed to analyze this estimation problem by modeling the state to reside on the $\mathrm{SO}(2)$ group. Subsequently, we have shown that the $\mathrm{SO}(2)$ filter derivation based on the mathematically grounded framework of filtering on Lie groups yields evaluates to heuristically wrapping the extended Kalman filter. We emphasize that this result *applies only* to the $\mathrm{SO}(2)$ filter and is not intended to be extended to other Lie groups or combinations thereof. Second, we have derived the constant angular velocity $\mathrm{SO}(2) \times \mathbb{R}^2$ filter, where the system state consisted of azimuth, angular velocity and angular acceleration. For this filter we showcased a real-world experiment of a speaker tracking problem with a microphone array by assessing the accuracy using the ground truth obtained by a motion capture system.

Acknowledgments

This work has been supported from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688117 (SafeLog) and the Unity Through Knowledge Fund under the project Cooperative Cloud based Simultaneous Localization and Mapping in Dynamic Environments.

References

[1] M. Azmani, S. Reboul, J.-B. Choquel, and M. Benjelloun, "A recursive fusion filter for angular data," in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, dec 2009, pp. 882–887.

[2] I. Marković and I. Petrović, "Speaker localization and tracking with a microphone array on a mobile robot using von Mises distribution and particle filtering," *Robotics and Autonomous Systems*, vol. 58, no. 11, pp. 1185–1196, 2010.

[3] ——, "Bearing-only tracking with a mixture of von Mises distributions," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 707–712.

[4] G. Kurz, I. Gilitschenski, and U. D. Hanebeck, "Recursive monlinear filtering for angular data based on circular distributions," in *American Control Conference (ACC)*, 2013, pp. 5439–5445.

[5] G. Kurz, I. Gilitschenski, S. Julier, and U. D. Hanebeck, "Recursive estimation of orientation based on the Bingham distribution," in *International Conference on Information Fusion (FUSION)*, 2013, pp. 1–16.

[6] G. Stienne, S. Reboul, M. Azmani, J. Choquel, and M. Benjelloun, "A multi-temporal multi-sensor circular fusion filter," *Information Fusion*, vol. 18, pp. 86–100, jul 2014.

[7] I. Marković, J. Ćesić, and I. Petrović, "Von Mises mixture PHD filter," *IEEE Signal Processing Letters*, vol. 22, no. 12, pp. 2229–2233, 2015.

[8] G. Bourmaud, R. Mégret, M. Arnaudon, and A. Giremus, "Continuous-discrete extended Kalman filter on matrix Lie groups using concentrated Gaussian distributions," *Journal of Mathematical Imaging and Vision*, vol. 51, no. 1, pp. 209–228, 2015.

[9] D. F. Crouse, "Cubature / Unscented / Sigma Point Kalman Filtering with Angular Measurement Models," in *International Conference on Information Fusion*, 2015, pp. 1550–1557.

[10] G. S. Chirikjian, *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*. Springer, 2012.

[11] F. Grondin, D. Létourneau, F. Ferland, V. Rousseau, and F. Michaud, "The ManyEars open framework," *Autonomous Robots*, vol. 34, no. 3, pp. 217–232, 2013.

[12] J.-M. Valin, F. Michaud, and J. Rouat, "Robust localization and tracking of simultaneous moving sound sources using beamforming and particle filtering," *Robotics and Autonomous Systems*, vol. 55, no. 3, pp. 216–228, 2007.

[13] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS : an open-source Robot Operating System," *IEEE International Conference on Robotics and Automation (ICRA), Workshop on Open Source Software*, 2009.