R-RANSAC with Lie Groups

I. Introduction

Optimal algorithms for the multiple target tracking (MTT) problem are not feasible in real time due to their computational complexities; therefore, many sub-optimal approaches have been developed which can be implemented in real time. Some of these common algorithms are the global nearest neighbor (GNN), the joint probabilistic data association filter (JPDAF), the multiple hypothesis tracking (MHT), and the probabilistic multi-hypothesis tracker (PHMT). Each of these algorithms have variants that trade optimality for computational feasibility. There are also many other MTT algorithms which we will not discussed here but we refer the reader to [37] for a survey of the more common algorithms. We briefly present some of these common algorithms in order to compare and contrast them.

The GNN is the simplest MTT method. It is a single-scan tracker that uses hard data association to assign measurements to tracks based on distance without taking into account statistical information such as the innovation term [4]. The GNN filter is a computationally efficient algorithm, but its "greedy" data association causes track divergence and loss in clutter. This method requires an additional scheme to prune, initialize and merge tracks.

The JPDAF is an extension of the probabilistic data association filter (PDAF) to multiple target tracking. It is a single-scan tracker that uses a soft data association technique to weighs new measurements inside a track's validation region based upon the new measurement's joint data association probability [2]. The weights are used to update the tracks. Even though it is a single-scan tracker, calculating the joint probability of all possible data associations per scan makes it more computationally complex and demanding. In addition, this method requires an additional scheme to prune, initialize and merge tracks.

The MHT algorithm is a batch tracker that uses a hard data association technique. It is an optimal algorithm since it considers all possible measurement associations in order to create the hypotheses tree. There are two different versions of MHT: measurement-oriented (MO-MHT) [38] and track-oriented (TO-MHT) [22]. While MHT is an optimal algorithm, it is not feasible unless appropriate techniques are used to limit the number of generated hypotheses: pruning, merging, sliding window, etc. When these techniques are used, the algorithm is suboptimal. Lastly, the algorithm is capable of initializing and pruning tracks.

The PHMT is a soft data association algorithm that relaxes the criteria that a track can only produce at most one measurement per time step. It does this by using a combination of the maximum likelihood and expectation maximization algorithm [13] to maximize the posterior probability of the tracks given a batch of measurements or a windowed batch of measurements. In the original form, it didn't have a method of track initialization or pruning, but later variants used the Hugh transform or the MHT as a front end to initialize tracks [10]. The PHMT performance is similar to that of the PDAF which is a single-scan tracker [4]. A variant of the PHMT operates on a windowed batch of data, and can be used as a single-scan tracker.

Recursive random sample consensus (R-RANSAC) is a recently new MTT algorithm with a modular paradigm. Unlike the MTT algorithms previously discussed that rely on additional initialization scheme or a computationally extensive hypotheses tree to initialize tracks, R-RANSAC uses RANSAC over a windowed batch of data to initialize tracks. Once tracks are initialized, it uses any single-scan tracker (GNN, JPDAF, PDAF, PHMT, ect) to update the tracks. In addition, it manages the tracks by pruning and merging them. Another novelty of the algorithm is that the surveillance region doesn't need to be fixed, but can be moved provided that a transformation is available to transform measurements and tracks.

R-RANSAC was first designed to estimate the parameters of multiple static signals [32], but it was quickly extended to track multiple dynamic targets [33]. Since then, R-RANSAC has been improved, modified and changed in various literature [12], [19], [20], [24], [26], [31], [34]–[36], [47]–[49]. All of these improvements seemed to have stemmed from [33], which resulted in many different versions of R-RANSAC. This paper aims at aggregating the best contributions while adding some of our own improvements. Our specific contributions include removing the constraint that measurements come at a fixed time interval, improving the criteria for a good track using a probabilistic frame work, allowing the measurement and process noise covariance to change at every time step, extending the theory of R-RANSAC to any Lie group, modifying the model merging method and give several example implementation.

The rest of the paper is outlined as follows. In section

II we give a brief review of Lie theory to establish important notation, in section III we present a more detailed description of the problem with assumptions, in section IV we present the modular paradigm. Sections V-IX go into more detail on specific parts of R-RANSAC, and in section X we present examples.

II. Lie Theory Review

The objective of this section is to provide a review of the pertinent concepts of Lie theory to establish notation. We assume that the reader is familiar with Lie group and Lie algebra theory. For those unfamiliar with this material, we refer the interested reader primarily to [42]. Solá offers a very gentle introduction to Lie theory that covers the majority of information needed to understand this paper. We also follow much of the notation prescribed by Solá and Hertzberg in [42] and [18]. For those who are interested in a more rigorous treatment of Lie theory, we refer the reader to [1], [5], [8], [17], [23], [43]. In this document we will focus on targets with a discrete system model and constant velocity, this is merely to simplify the presentation of the material. With no loss in generality, everything we present can be extended to targets with a continuous system model and constant acceleration by using a semidirect product group formed from a Lie group and its Lie algebra. A good discussion of the semidirect product group can be found in [15]. If using a continuous system that requires numerical integration on the manifold, we refer the reader to [28]–[30] which describe the Runge-Kutta-Muthe-Kass numerical integration technique.

Let G denote a Lie group and $\mathfrak{g}$ denote it's corresponding Lie algebra. The exponential function is a surjection that maps an element of the Lie algebra to an element of the Lie group, and the logarithmic map is the inverse of the exponential map. We denote these maps as

$$\begin{aligned}\exp &: \mathfrak{g} \to G \\ \log &: G \to \mathfrak{g}.\end{aligned}$$

The definition of these maps is dependent on the Lie group. For matrix Lie groups, the exponential and logarithm maps are defined as the matrix exponential and matrix logarithm.

The Lie algebra is isomorphic to the Euclidean space $\mathbb{R}^n$ where n is the dimension of the Lie algebra. We will denote this Euclidean space as E. The Wedge function maps an element from the Euclidean space to the Lie algebra, and the Vee function is the inverse function which we denote as

$$\begin{aligned}\cdot^{\wedge} &: E \to \mathfrak{g} \quad (v) \mapsto v^{\wedge} \\ \cdot^{\vee} &: \mathfrak{g} \to E \quad \left(v^{\wedge}\right)^{\vee} \mapsto v.\end{aligned}$$

The definitions of these maps are dependent on the Lie algebra. We will rely on context to distinguish between elements of E and $\mathfrak{g}$.

Let $\mathrm{Exp} : E \to G$ and its inverse be defined as the composite function

$$\begin{aligned}\mathrm{Exp}\left(v\right) &= \exp\left(v^{\wedge}\right) \\ \mathrm{Log}\left(g\right) &= \log\left(g\right)^{\vee}.\end{aligned}$$

Other functions of importance are the box-plus/minus and the o-plus/minus defined as

$$\begin{aligned}\boxplus &: G \times \mathfrak{g} \to G \quad (g, u) \mapsto g \bullet \exp\left(u\right) \\ \boxminus &: G \times G \to \mathfrak{g} \quad (g_1, g_2) \mapsto \log\left(g_2^{-1} \bullet g_1\right) \\ \oplus &: G \times E \to G \quad (g, v) \mapsto g \bullet \mathrm{Exp}\left(v\right) \\ \ominus &: G \times G \to E \quad (g_1, g_2) \mapsto \mathrm{Log}\left(g_2^{-1} \bullet g\right).\end{aligned}$$

where $\bullet$ denotes the group operator which we will omit in the future, and g_2^{-1} is the inverse element of g_2. We have based the definition of these function on left trivializations (using vector fields that are left invariant) since we will present the material using the left trivialization. Note that we could easily use the right trivialization as well.

The adjoint of G is a representation of G that acts on $\mathfrak{g}$, and is denoted and generically defined as

$$Ad_g : \mathfrak{g} \to \mathfrak{g}; \quad (u) \mapsto gug^{-1}$$

where $g \in G$. Since the adjoint is a linear function, we can find a matrix version that operates on the associated Euclidean space generically defined as

$$\mathbf{Ad}_g : E \to E; \quad (v) \mapsto \mathbf{Ad}_g v$$

where $\mathbf{Ad}_g$ is the matrix adjoint representation of G.

The adjoint of $\mathfrak{g}$ is a representation of $\mathfrak{g}$ that acts on $\mathfrak{g}$. It is also the Lie bracket $[\cdot, \cdot]$ which we will generically define as

$$ad_u : \mathfrak{g} \to \mathfrak{g}; \quad (v) \mapsto [u, v]$$

where $u \in \mathfrak{g}$. Since the adjoint is a linear function, we can find a matrix version that operates on the associated Euclidean space generically defined as

$$\mathbf{ad}_u : E \to E \quad (v) \mapsto \mathbf{ad}_u v$$

where $u \in \mathfrak{g}$ and $\mathbf{ad}_u$ is the matrix adjoint representation of $\mathfrak{g}$.

As stated in [21], we can define the differential of the exponential mapping as the 'left trivialized' tangent of the exponential map or as the 'right trivialized' tangent of the exponential map. These differentials are also commonly called the right and left Jacobians. The right and left Jacobians $J_r, J_l : E \to GL\left(E\right)$ are defined as

$$J_r\left(v\right) = \frac{{}^{r}\partial \mathrm{Exp}\left(v\right)}{\partial v} \qquad J_l\left(v\right) = \frac{{}^{l}\partial \mathrm{Exp}\left(v\right)}{\partial v}.$$

Their inverses are defined as

$$J_r^{-1}(v) = \frac{{}^r\partial \mathrm{Log}(v)}{\partial v} \qquad J_l^{-1}(v) = \frac{{}^l\partial \mathrm{Log}(v)}{\partial v}.$$

The right Jacobian has the property that for small $\delta v \in E$ and $v \in E$

$$\begin{aligned} \mathrm{Exp}(\delta v + u) &\approx \mathrm{Exp}(v)\,\mathrm{Exp}(J_r(v)\,\delta v) \\ \mathrm{Exp}(v)\,\mathrm{Exp}(\delta v) &\approx \mathrm{Exp}\left(v + J_r^{-1}(v)\,\delta v\right) \\ \mathrm{Log}(\mathrm{Exp}(v)\,\mathrm{Exp}(\delta v)) &\approx v + J_r^{-1}(v)\,\delta v. \end{aligned}$$

The left Jacobian has the similar property that for small $\delta v \in E$ and $v \in E$

$$\begin{aligned} \mathrm{Exp}(v + \delta v) &\approx \mathrm{Exp}(J_l(v)\,\delta v)\,\mathrm{Exp}(v) \\ \mathrm{Exp}(\delta v)\,\mathrm{Exp}(v) &\approx \mathrm{Exp}\left(v + J_l^{-1}(v)\,\delta v\right) \\ \mathrm{Log}(\mathrm{Exp}(\delta v)\,\mathrm{Exp}(v)) &\approx v + J_l^{-1}(v)\,\delta v. \end{aligned}$$

The derivation of the left and right Jacobians stems from the Baker-Campbell-Hausdorff formula and can be studied in [17] and [5].

An infinitesimal generator corresponding to $u \in \mathfrak{g}$ is a smooth vector field on G defined as

$$\xi_u(g) = \left.\frac{d}{dt}\right|_{t=0} g \boxplus (tu)$$

This is a very simplified definition that serves our purpose in deriving the discrete system model. For a more formal and encompassing definition of an infinitesimal generator, we refer the reader to the authors already mentioned.

III. Problem Description

The objective is to track multiple dynamic targets by estimating their states given a set of measurements from multiple sensors without a prior information about the number of targets.

The following notation is used throughout. We denote the current time using a subscript k, the next previous time using a subscript k^-, and an arbitrary previous time using the subscript m such that $m \leq k$. We use δ_k to denote the time elapsed from k^- to k, and $\delta_{m:k}$ to denote the time elapsed from m to k such that $\delta_{m:k}, \delta_{k^-} \geq 0$ and $\delta_{k:m} = -\delta_{m:k}$. Lastly, a subscript of 0 is used to denote the the time at the beginning of the time window, e.g, $\delta_{0:k} = \delta_{k-T_W:k}$.

Let the state of the system at time k be given by $x_k = (g_k, u_k) \in G \times E$ where $g_k \in G$ and $u_k \in E$. We assume the state transition model to be continuous, near constant velocity, and time invariant which we define as

$$\begin{aligned} \dot{g}_k &= \xi_{u_k}(g_k) \\ \dot{u}_k &= w_k^u, \end{aligned}$$

where $\xi_{u_k}(g_k)$ is an infinitesimal generator constructed from u_k, $w_k = \begin{bmatrix} w_k^g \\ w_k^u \end{bmatrix}$ is process noise sampled from a zero-mean, white-noise, Gaussian distribution with covariance Q. Under the assumption that w_k stays constant over a time period δ_k we can discretize the model to get

$$g_k = g_{k^-} \oplus \left(\delta_k u_{k^-} + \delta_k w_k^g + \frac{\delta_k^2}{2} w_k^u\right) \tag{5a}$$

$$u_k = u_{k^-} + \delta_k w_k^u. \tag{5b}$$

This form is similar to the one found in [41] and is an exact solution provided that the Lie group is commutative. If the Lie group is not commutative, the exact numeric solution is cumbersome if not impossible to derive, in which case we use the system model as an approximate numerical solution.

The complete discrete system is defined as

$$x_k = f(x_{k^-}, w_k, \delta_k) \tag{6a}$$

$$y_k = h(x_k, v_k), \tag{6b}$$

where f is the state transition model defined in (5), $y_k \in N$ is a measurement and N is a Lie group, $h : G \to N$ is the observation model, and v_k is measurement noise sampled from a zero-mean, white-noise, Gaussian distribution with covariance R.

The state x is a Lie group generated from the Cartesian product of G and E. The Lie algebra of $G \times E$ is simply $\mathfrak{g} \times E$ which is isomorphic to $\mathbb{R}^{2n}$ with n being the dimension of G. We will denote this space as E^*. All of the operations previously defined are simply inherited. For example, let $x \in G \times \mathfrak{g}$ and $v = (a, b)^\vee \in E^*$, then

$$\begin{aligned} x \oplus v &= (g, u) \oplus v \\ &= (g, u) \boxplus (a, b) \\ &= (g \exp(a), u + b), \end{aligned}$$

where $g \in G$, $a \in \mathfrak{g}$, and $u, b \in E$.

Targets are observed via sensors. Each sensor perceives a subset of the measurement space called a local surveillance region (LSR) which has a local frame that the data is expressed in and a volume denoted S_{vol}. The union of all LSRs is called the global surveillance region (GSR). Measurements are extracted from the data using algorithms. We call the unique sensor and algorithm pair a measurement source or source for short. In the case that multiple sources have the same sensor, it is possible for duplication of information which, when incorporated into a state estimate of a target, would lead to overconfidence of the state estimate. We do not have a way to account for this except to assume it is minimal.

A sensor scan occurs when a source produces new measurements. Measurements from a source can be either false or true measurements. We assume that false measurements are uniformly distributed in the LSR, and that the number of false measurements from a source per sensor scan can be modeled using a Poisson

distribution with parameter Λ being the expected value. The expected number of false measurements per unit volume is called the spacial density and is $\lambda = \frac{\Lambda}{S_{vol}}$. We also assume that a measurement source provides at most one true measurement per target every sensor scan with probability P_D, i.e. the probability of detection.

R-RANSAC works in a single frame requiring all of the measurements to be transformed and expressed in a single global frame before being given to R-RANSAC. In the case that the global frame changes, a transformation T must be provided to R-RANSAC that contains the information necessary to transform all the past measurements and models stored in R-RANSAC into the new global frame.

R-RANSAC uses the new measurements along with all the previous measurements from the time window T_W to track targets. As time progresses, old measurements fall outside the time window and are removed. These measurements are called expired measurements. Below is a summary of our assumptions.

1) We assume that the system is observable.
2) We assume that the process and measurement noises are represented by a white-noise, zero-mean, Gaussian distribution and that their covariances are known.
3) We assume that all measurements are independent.
4) We assume that the expected number of false measurements from each measurement source per sensor scan is modeled using a Poisson distribution with spatial parameter λ, and that the false measurements are uniformly distributed in the LSR.
5) We assume that every measurement given to R-RANSAC is expressed in the current global frame.
6) If the global frame moves, a transformation is provided to R-RANSAC in order to transform the measurements and models to the current global frame.

IV. Modular Paradigm

The paradigm for R-RANSAC is designed to be very modular. In this section we will present the data structures used and modular framework. R-RANSAC can be broken down into three main parts: data management, track initialization, and track management.

A. Data Structures

In R-RANSAC, we use three different data structures: the data tree, clusters, and consensus sets. The structure of the data tree and clusters is a list of R*-Trees [11]. Each R*-Tree contains measurements obtained at a unique time. This allows us to easily remove and add measurements that have the same time stamp while being able to search the data structures quickly. The data tree, $\mathcal{T}$, contains all of the measurements within the time window that are not in a cluster or a consensus set. A cluster, $\mathcal{C}$, contains neighboring measurements as described in [48]. Let $d_C : N \times N \rightarrow \mathbb{R}$ be a metric [27] on the measurement space. A measurement y_j is a neighboring measurement of y_k if $d_C(y_j, y_k) < \tau_{CD}$. A consensus set, $\mathcal{CS}^i$, is a list that contains all of the measurements within the time window that were associated with the i^{th} track. A track is a model of a target consisting of the state estimate $\hat{x}$, error covariance P, consensus set $\mathcal{CS}$, track likelihood, and label. The track likelihood is a measurement of how "good" a track is. Previously, this was based on the inlier ratio and track lifetime, see [36]; however, we will present a more probabilistic approach in section VI.

B. Data Management

Data management occurs after a sensor scan provides new measurements Ψ_k and possibly a new transformation T. First, expired measurements are removed from the data structures and models are propagated forward in time. If a transformation was provided, all of the measurements (not including the new measurements since they should already be expressed in the current global frame) and tracks are transformed to the current global frame, i.e. their state estimate and error covariance are transformed to the current global frame.

The new measurements are checked to see if they should be associated to a track and used to update the track's state estimate and likelihood. The data association and track update could be implemented using any single scan tracker. Our data association method is described in section **??**. Unassociated new measurements are then checked to see if they belong to an existing cluster using the metric d_C. Those that don't belong to a cluster are then used to seed possible new clusters.

New clusters are created by taking an unassociated new measurement and finding neighboring measurements from either the list of unassociated new measurements or the data tree. For every measurement added to the cluster, the process is repeated by finding neighboring measurements to the measurement just added to the cluster.

If the possible cluster has at least τ_{CM} measurements, then the cluster is kept, and the associated measurements are removed from the data tree. The remaining unassociated measurements are placed on the data tree creating a new element in the list.

C. Track Initialization and Track Management

Track initialization and track management occurs at specified times by the user. The track initialization phase performs RANSAC, see section VIII, on each cluster to

try to generate new tracks. If RANSAC generates a new track, the new track's state estimate $\hat{x}$ is filtered and the track's likelihood is initialized using the measurements in the consensus set. The measurements in the consensus set are removed from the respective cluster. Since measurements are taken from a cluster in order to create a new track, we must verify that the remaining measurements in the cluster still form a valid cluster, i.e. the cluster has at least τ_{CM} measurements. If it does, the cluster is kept. If it doesn't, then we check the time stamp of the measurements. If the measurements are old (i.e. near being expired), then we remove them; otherwise, we place them back on the data tree.

The last phase of R-RANSAC is track management which consists of merging and pruning tracks, promoting and demoting tracks, and assigning new good tracks a unique label. Some tracks will begin to coalesce as they are propagated and updated. If these tracks are deemed similar, they are merged together using the track-to-track fusion method discussed in section VII. Since R-RANSAC stores up to M Tracks, if there are more than M tracks we remove the tracks with the lowest likelihood. If a track's likelihood is above the threshold τ_ρ then it is promoted to a good track; otherwise, it is a poor track. Good tracks receive a unique numerical label to identify it , see [19]. At the end of these phases, a list of good tracks is published to the user. For a summary, see algorithm 1.

Algorithm 1 R-RANSAC

Require: Parameters M, T_W, ℓ, τ_E, τ_I, τ_{CD}, τ_{CM}, τ_S, τ_ρ, τ_α, P_D and λ.

```
for each sensor scan do
    Remove expired measurements
    Propagate tracks
    if Transform T provided then
        Transform measurements and tracks
    end if
    Associate new measurements to tracks
    Update track's state estimate and likelihood using associated measurements
    Associate unassociated measurements to clusters using metric d_C
    for y ∈ { unassociated measurements } do
        Try generating possible new clusters C^p from y using unassociated measurements and data tree
        if | C^p | ≥ τ_CM then
            Let C^p be a new cluster
            Remove measurements in C^p from data tree
        end if
    end for
    Add remaining measurements to data tree
    if specified by user then
        Perform RANSAC on each cluster
        Manage clusters
        Merge and prune tracks
        Promote and demote tracks
        Assign new good tracks a unique ID
        Publish good tracks
    end if
end for
```

V. Probabilistic Data Association Filter

As mentioned in the introduction, there are a variety of ways to propagate the models, associate measurements, and update the models with the associated measurements. For our implementation, we have chosen to use the Probabilistic Data Association Filter (PDAF) even though we are tracking multiple targets [2]. Because of the modular design of R-RANSAC we could implement it with a JPDAF, but we decided that the little increase in tracking accuracy is not worth the large increase in computation as discussed in [19]. In this section we present the PDAF in a Lie group setting focusing only on the parts we believe need further discussion. For some additional background on presenting Lie groups in a probabilistic frame work we refer the reader to [**?**], [7]. We also follow some of the derivation, convention, notation and terminology from [39], [40], [44] but adapted for the Lie group setting.

We present the PDAF for one target, but it is trivially extended to the case of multiple targets. Also, we present the PDAF for the case of multiple targets. The following notation is used in this secion. Let $\mathcal{S} = \{1, \dots, N_s\}$ be the indexing set of sources with N_s being the number of sources, and Y_m^s be the set of measurements associated to the model at time m produced by source $s \in \mathcal{S}$. It is possible that $Y_m^s = \emptyset$ if a source didn't produce any measurements. We assume that each source produces at most one measurement originated from a target with probability of detection P_D^s. Also, the source produces clutter (noisy measurements) uniformly in the local surveillance region, where the number of clutter measurements per scan is modeled with a Poisson distribution with spatial density λ^s. The spatial density λ^s is the expected number of clutter measurements per sensor scan divided by the volume of the local surveillance region. Lastly, let $y_m^{s,j} \in Y_m^s$ denote the j^{th} measurement in Y_m^s assuming that $Y_m^s \neq \emptyset$.

A. Propagation

In this section we present the model propagation step used in the PDAF. We denote the state estimate at time m as $\hat{x}_m \in G \times E$ and the true state as $x_m \in G \times E$. Let $\delta x_m \in E^*$ denote a local perturbation around $\hat{x}_m$

that is a zero-mean, Gaussian, random variable with error covariance P_m such that

$$x_m = \hat{x}_m \oplus \delta x_m.$$

x_m and δx_m are both random variables, but $\hat{x}_m$ is not.

We denote the prior distribution, or the belief distribution as $p(x_m)$ and define it as

$$\begin{aligned} p(x_m) &= \eta \exp\left(-\frac{1}{2}\left(x_m \ominus \hat{x}_m\right)^\top P_m^{-1}\left(x_m \ominus \hat{x}_m\right)\right) \\ &= \eta \exp\left(-\frac{1}{2}\delta x_m^\top P_m^{-1} \delta x\right), \end{aligned}$$

where η is a normalizing coefficient corresponding to the Gaussian distribution. We will use η to denote any normalizing coefficient. We denote the state transition probability as $p(x_m \mid x_{m^-})$ which we can calculate by taking the first and second moments of x_m conditioned on x_{m^-}. In other words, calculate the first and second moments of

$$x_m = f(x_{m^-}, w_m, \delta_m).$$

Since $p(x_m \mid x_{m^-})$ is conditioned on x_{m^-}, x_{m^-} is not a random variable; however, the noise term w_m is. In the case that f is non linear in the noise term, we need to linearize f w.r.t. the process noise in order to approximate $p(x_m \mid x_{m^-})$ as Gaussian. Thus

$$x_m \approx f(x_{m^-}, 0, \delta_m) + G_m w_m$$

where

$$G_m = \left.\frac{\partial f(x_{m^-}, w_m, \delta_m)}{\partial w_m}\right|_{w_m = 0}.$$

G_m is evaluated at $w_m = 0$ since 0 is the mean of the process noise. We can now calculate the first and second moments of x_m to get

$$\begin{aligned} \mathrm{E}[x_m] &= f(x_{m^-}, 0, \delta_m) \\ \mathrm{cov}[x_m] &= G_m Q G_m^\top. \end{aligned}$$

Using the first and second moments to approximate $p(x_m \mid x_{m^-})$, we get

$$\begin{aligned} p(x_m \mid x_{m^-}) \approx \eta \exp\Bigg(&-\frac{1}{2}\left(x_m \ominus f(x_{m^-}, 0, \delta_m)\right)^\top \\ &\left(G_m Q G_m^\top\right)^{-1}\left(x_m \ominus f(x_{m^-}, 0, \delta_m)\right)\Bigg). \end{aligned}$$

Let $\bar{p}(x_m)$ denote the probability of x_m after state propagation. According to the theorem of total probability, $\bar{p}(x_m)$ is defined as

$$\begin{aligned} \bar{p}(x_m) &= \int_{x_{m^-}} p(x_m \mid x_{m^-})\, p(x_{m^-})\, dx_{m^-}. \\ &\approx \eta \exp\left(-\frac{1}{2}\left(x_m \ominus \hat{\bar{x}}_m\right)^\top \bar{P}_m^{-1}\left(x_m \ominus \hat{\bar{x}}_m\right)\right) \\ &= \eta \exp\left(-\frac{1}{2}\delta \bar{x}_m^\top \bar{P}_m^{-1} \delta \bar{x}_m\right) \end{aligned}$$

We approximate $\bar{p}(x_m)$ by linearizing $p(x_m \mid x_m)$ w.r.t. the state and process noise so that $\bar{p}(x_m)$ is a Gaussian distribution which is parametrized by the new state estimate $\hat{\bar{x}}_m$ and error covariance $\bar{P}_m$. The derivation of $\bar{p}(x_m)$ for $\mathbb{R}^n$ is shown in [44], and it's extension to the Lie group setting is easy to derived; thus, we don't include it here. The new state estimate and error covariance is calculated as

$$\hat{\bar{x}}_m = f(\hat{x}_{m^-}, \hat{w}_m, \delta_m) \tag{7a}$$

$$\bar{P}_m = F_m P_{m^-} F_m^\top + G_m Q G_m^\top; \tag{7b}$$

with

$$\begin{aligned} F_m &= \left.\frac{\partial x_m \ominus f(x_{m^-}, w_k, \delta_m)}{\partial x_{m^-}}\right|_{\hat{x}_{m^-}, \hat{w}_m, \delta_m} \\ &= \begin{bmatrix} \mathbf{Ad}_{\mathrm{Exp}(\delta_m \hat{u}_{m^-})^{-1}} & J_r(\delta_m \hat{u}_{m^-})\,\delta_m \\ 0 & I \end{bmatrix}, \end{aligned}$$

$$\begin{aligned} G_m &= \left.\frac{\partial x_m \ominus f(x_{m^-}, w_k, \delta_m)}{\partial w_m}\right|_{\hat{x}_{m^-}, \hat{w}_m, \delta_m} \\ &= \begin{bmatrix} J_r(\delta_m \hat{u}_{m^-})\,\delta_m & J_r(\delta_m \hat{u}_{m^-})\,\frac{\delta_m^2}{2} \\ 0 & I\delta_m \end{bmatrix}. \end{aligned}$$

This calculation is called the propagation or the prediction step. Note that this definition of G_m is different than the one above since we linearized about the state and not just the noise.

If a transformation is provided to R-RANSAC, you would transform the model after propagating and before associating new measurements. This is because new measurements are in the current global frame, and so the models must also be in the current global frame for proper data association.

B. Data Association

The PDAF uses a validation region to indicate if a measurement should be associated with a model. The validation region is a volume around a track's current state estimate. Measurements that are within the validation region are associated to the model and are used to update the model. We will use the validation region described in [3], [4].

The volume is determined by the probability of a measurement $y_m^{s,j}$ which is defined as

$$p\left(y_m^{s,j}\right) = \int_{\delta \bar{x}_m} p\left(y_m^{s,j} \mid x_m\right) \bar{p}(x_m)\, d\delta \bar{x}_m.$$

For readability, we will droip the superscripts on $y_m^{s,j}$. In order to solve the integral, we linearize the measurement probability $p(y_m \mid x_m)$ about $\hat{\bar{x}}_m$ and the measurement noise. This serves two purposes: first it

ensures the $p(y_m)$ stays Gaussian and second it exposes δx_m which is the integrating term. Doing so yields

$$p(y_m) \approx \eta \int_{\delta x_m} \exp\left(-\frac{1}{2}\left(y_m \ominus \hat{y}_m + H_m^s \delta \bar{x}_m\right)^\top \left(\bar{R}^s\right)^{-1}\left(y_m \ominus \hat{y}_m + H_m^s \delta \bar{x}_m\right)\right) \exp\left(-\frac{1}{2}\bar{x}_m^\top \bar{P}_m^{-1} \delta \bar{x}_m\right) d\delta \bar{x}_m$$

with

$$\bar{R}^s = \left(V_m R^s V_m^\top\right),$$
$$\hat{y}_m = h\left(\hat{\bar{x}}_m, 0\right)$$

$$H_m^s = \left.\frac{\partial y_m \ominus h(x_m, v_m)}{\partial x_m}\right|_{\substack{x_m = \hat{x}_m, \\ v_m = 0}}$$
$$= \left.\frac{\partial \log(e)}{\partial e}\frac{\partial e}{\partial h^{-1}(x_m, v_m)}\frac{\partial h(x_m, v_m)}{\partial x_m}\right|_{\substack{x_m = \hat{x}_m, \\ v_m = 0}} \quad (8)$$
$$= J_r^{-1}(\hat{e})\, \mathbf{Ad}_{y_m^{-1}} \left.\frac{\partial h(x_m, v_m)}{\partial x_m}\right|_{\substack{x_m = \hat{x}_m, \\ v_m = 0}},$$

$$V_m^s = \left.\frac{\partial y_m \ominus h(x_m, v_m)}{\partial v_m}\right|_{\substack{x_m = \hat{x}_m, \\ v_m = 0}} \quad (9)$$
$$= J_r^{-1}(\hat{e})\, \mathbf{Ad}_{y_m^{-1}} \left.\frac{\partial h(x_m, v_m)}{\partial v_m}\right|_{\substack{x_m = \hat{x}_m, \\ v_m = 0}},$$

and

$$e = h\left(\hat{\bar{x}}_m, 0\right)^{-1} y_m.$$

The superscript on H_m^s, V_m^s and R^s indicate that these terms are for the s^{th} measurement source.

By rearranging the terms in the linearized $p(y_m)$ we get

$$p(y_m) \approx \eta \exp\left(-\frac{1}{2}\left(y_m \ominus \hat{y}_m\right)^\top \left(S^s\right)^{-1}\left(y_m \ominus \hat{y}_m\right)\right) \int_{\delta x_m} \exp\left(-\frac{1}{2} l(\delta \bar{x}_m)^\top \left(S^s\right)^{-1} l(\delta \bar{x}_m)\right) d\delta \bar{x}_m$$

with

$$l(\delta \bar{x}_m) = \left(\delta \bar{x}_m + S^s H_m^s \left(\bar{R}^s\right)^{-1} y_m \ominus \hat{y}_m\right)$$

and the innovation covariance

$$S^s = \left(V_m^s R^s \left(V_m^s\right)^\top + H_m^s P \left(H_m^s\right)^\top\right). \quad (10)$$

Note that we were able to pull a term outside of the exponential, and the term inside of the exponential is a Gaussian distribution which integrates to a constant regardless of $\delta \bar{x}_m$. Therefore, we can absorb the constant integral into η to get

$$p(y_m) \approx \eta \exp\left(-\frac{1}{2}\left(y_m \ominus \hat{y}_m\right)^\top \left(S^s\right)^{-1}\left(y_m \ominus \hat{y}_m\right)\right). \quad (11)$$

We don't want to associate every measurement to the model, only the measurements with a high probability. This is done by using a gating technique as follows. Let $p\left(z_m^{s,j}\right)$ be the standardized Gaussian probability of $p\left(y_m^{s,j}\right)$ where z_m (we drop the superscripts) is defined as

$$z_m = \left(S^s\right)^{-\frac{1}{2}} y_m \ominus \hat{y}_m.$$

Let $d_V : N \times N \to \mathbb{R}$ be a metric defined as

$$d_V(y_m, \hat{y}_m) = \left(y_m \ominus \hat{y}_m\right)^\top \left(S^s\right)^{-1}\left(y_m \ominus \hat{y}_m\right) \quad (12)$$
$$= z_m^\top z_m.$$

Note that the metric d_V is simply the sum of the square of n standard Gaussian distributions. Thus, the values of the metric d_V form a chi-square distribution.

The validation region is the set

$$\nu(\hat{y}_m, \gamma) = \left\{y_m \in Y_m \,:\, d(y_m, \hat{y}_m) \leq \gamma\right\},$$

where Y_m is the set of measurements received at time m and the parameter γ is called the gate threshold. The gate probability P_G is the probability that a measurement produced by a target is within the validation region. This probability is defined as $p\left(d_V(y_m, \hat{y}_m) \leq \gamma\right)$, thus P_G is the value of the chi-square cumulative distribution function (CDF) with parameter γ. We can thus calculate γ for a specific P_G.

The volume of the validation region, V_{vol}^s, is defined as

$$V_{vol}^s = c_n \gamma^{n/2} \left|S^s\right|^{1/2}, \quad (13)$$

where c_n is the volume of the unit hypersphere of dimension n (dimension of the measurement space) calculated as

$$c_n = \frac{\pi^{n/2}}{\Gamma(n/2 + 1)}$$

and Γ is the gamma function. Since the validation region depends on the measurement noise covariance, there will be a different validation region for each measurement source. Also, the number of expected measurements in the validation region is $\lambda^k V_{vol}^k$ for each source.

C. Measurement Weight

The PDAF assumes that a measurement source produces at most one measurement originated from the target and that all other measurements are noise. Thus,

the PDAF assigns a weight to each measurement indicating how likely the measurement originated from the target. These weights are calculated for each measurement source. Let $\beta^{s,j}$ denote the weight assigned to measurement $y_m^{s,j}$. The weight is calculate as

$$\beta^{s,j} = \frac{\mathcal{L}\left(y_m^{s,j}\right)}{1 - P_D P_G + \sum_{y \in Y_m^s} \mathcal{L}\left(y\right)},$$

with

$$\mathcal{L}\left(y_m^{s,j}\right) = \frac{P\left(y_m^{s,j}\right) P_D^s}{\lambda^s}$$

and $P\left(y_m^{s,j}\right)$ being defined in (11). The null weight $\beta^{s,0}$ is the probability that none of the associated measurements from Y_m^s originated from the target and is calculated as

$$\beta^{k,0} = \frac{1 - P_D^s P_G}{1 - P_D^s P_G + \sum_{y \in Y_m^s} \mathcal{L}\left(y\right)}.$$

Note that $\sum_{i=0}^{|Y_m^s|} \beta^{s,i} = 1$.

D. Update

The PDAF originally assumes only one measurement source. In the case of multiple measurement sources, we apply the update step for each source and fuse the results together using the centralized measurement fusion discussed in subsection V-E. For each $Y_m^s \neq \emptyset$ we perform the PDAF update step [4].

The innovation term $\nu_m^{s,j}$ is

$$\nu_m^{s,j} = y_m^{s,j} \ominus h\left(\hat{\bar{x}}_m, 0\right),$$

and the combined innovation term is

$$\nu_m^s = \sum_{j=1}^{|Y_m^s|} \beta^{s,j} \nu_m^{s,j}. \tag{14}$$

Using the combined innovation term, the state update for the s^{th} source is

$$\hat{x}_m^s = \hat{\bar{x}}_m \oplus W_m^s \nu_m^s,$$

where W_m^s is the gain defined as

$$W_m^s = \bar{P}_m \left(H_m^s\right)^\top \left(S_m^s\right)^{-1}.$$

$\bar{P}_m$ is the error covariance before update defined in (7), H_m^s is defined in (8), and S_m^s is the innovation covariance defined in (10).

The covariance update is

$$P_m^s = \beta^{s,0} \bar{P}_m + \left(1 - \beta^{s,0}\right) \breve{P}_m^s + \tilde{P}_m^s, \tag{15}$$

where the covariance of the state updated with the correct measurement is

$$\breve{P}^s = \bar{P}_m - W_m^s S_m^s \left(W_m^s\right)^\top,$$

and the spread of the innovations term

$$\tilde{P}_m^s = W_m^s \left[\sum_{i=1}^{|Y_m^s|} \beta^{s,i} \nu_m^{s,i} \left(\nu_m^{s,i}\right)^\top - \nu_m^s \left(\nu_m^s\right)^\top\right] \left(W_m^s\right)^\top.$$

We now have several updated states and error covariance; one for each source that produced associated measurements. We show how to fuse them together in the next subsection. Also, once the measurements are used to update the model, they are added to the model's consensus set.

E. Centralized Measurement Fusion

We use the centralized measurement fusion algorithm described in [4] and [26] to fuse together the updated states and error covariance. We briefly review this method here only to adapt it to the Lie group setting and for the specific case of the JPDAF. Let $\bar{\mathcal{S}}_m \subseteq \mathcal{S}_m$ be the indexinig set to sourced that produced associated measurements. Then covariance fusion is

$$P_m^{-1} = \bar{P}_m^{-1} + \sum_{s \in \bar{\mathcal{S}}_m} \left(\left(P_m^s\right)^{-1} - \bar{P}_m^{-1}\right),$$

and the state update is

$$\hat{x}_m = \hat{\bar{x}}_m \oplus \left(\bar{P}_m \sum_{s \in \bar{\mathcal{S}}_m} \left(H_m^s\right)^\top \left(S_m^s\right)^{-1} \nu_m^s\right),$$

with P_m^s being defined in (15) and ν_m^s is defined in (14).

VI. Track Likelihood

The track likelihood is a measure of how well a track represents an actual target. In the previous versions of R-RANSAC, the track's inlier ratio and lifetime were used to indicate the track's likelihood which doesn't have much theoretical basis other than intuition. In this section, we present a different approach using a binary Bayes filter with static states. Let θ_m^s denote the number of validated measurements from the s^{th} measurement source at time m, $\theta_{m:k}^s = \cup_{i=m}^k \{\theta_i^s\}$, $\theta_m = \cup_{s \in \mathcal{S}_m} \theta_m^s$ and $\theta_{m:k} = \cup_{j=m}^k \theta_m$ where $m < k$. Let z be a Bernoulli random variable with $z = 1$ meaning that the track represents an actual target and $z = 0$ meaning that the track doesn't represent a target. For notational purposes, we will denote $p\left(z = 1\right)$ as $p\left(z\right)$ and $p\left(z = 0\right)$ as $p\left(\neg z\right)$.

We are interested in calculating the probability $p\left(z \mid \theta_{m:k}\right)$ which is the probability that the track represents an actual target conditioned on the number of validated measurements from each source and at each time step. Using Bayes rule we get

$$p\left(z \mid \theta_{m:k}\right) = \frac{p\left(\theta_k \mid z\right) p\left(z \mid \theta_{m:k^-}\right)}{p\left(\theta_k \mid \theta_{m:k^-}\right)},$$

and

$$p\left(\neg z \mid \theta_{m:k}\right) = \frac{p\left(\theta_k \mid \neg z\right) p\left(\neg z \mid \theta_{m:k^-}\right)}{p\left(\theta_k \mid \theta_{m:k^-}\right)}.$$

Taking the log ratio of the probabilities $p\left(z \mid \theta_{m:k}\right)$ and $p\left(\neg z \mid \theta_{m:k}\right)$, we get

$$\begin{aligned}\log\left(\frac{p\left(z \mid \theta_{m:k}\right)}{p\left(\neg z \mid \theta_{m:k}\right)}\right) &= \log\left(\frac{p\left(\theta_k \mid z\right) p\left(z \mid \theta_{m:k^-}\right)}{p\left(\theta_k \mid \neg z\right) p\left(\neg z \mid \theta_{m:k^-}\right)}\right) \\ &= \underbrace{\log\left(\frac{p\left(\theta_k \mid z\right)}{p\left(\theta_k \mid \neg z\right)}\right)}_{l_k} \\ &\quad + \underbrace{\sum_{n=m}^{k^-} \log\left(\frac{p\left(\theta_n \mid z\right)}{p\left(\theta_n \mid \neg z\right)}\right)}_{l_{k^-}}.\end{aligned}$$

Whenever new validated measurements are received, we calculate l_k and add it to the previous log odds ratio l_{k^-}.

The term l_k can be written as

$$\log\left(\frac{p\left(\theta_k \mid z\right)}{p\left(\theta_k \mid \neg z\right)}\right) = \sum_{s \in \mathcal{S}_k} \log\left(\frac{p\left(\theta_k^s \mid z\right)}{p\left(\theta_k^s \mid \neg z\right)}\right).$$

Thus we are left with calculating $p\left(\theta_k^s \mid z\right)$ and $p\left(\theta_k^s \mid \neg z\right)$ for each measurement source. We assume that a measurement source can only produce at most one true measurement during a sensor scan. Therefore, we have the two possibilities for $p\left(\theta_k^s \mid z\right)$: that all θ_k^s measurements are false or all but one are false. Thus

$$\begin{aligned}p\left(\theta_k^s \mid z\right) &= P_G P_D^s \exp\left(\lambda^s V_{vol}^s\right) \frac{\left(\lambda^s V_{vol}^s\right)^{\left(\theta_k^s - 1\right)}}{\left(\theta_k^s - 1\right)!} \\ &\quad + \left(1 - P_G^s P_D^s\right) \exp\left(\lambda^s V_{vol}^s\right) \frac{\left(\lambda^s V_{vol}^s\right)^{\theta_k^s}}{\left(\theta_k^s\right)!}.\end{aligned}$$

There is only one possibility for $p\left(\theta_k^s \mid \neg z\right)$, that is the probability of all of the measurements being false

$$p\left(\theta_k^s \mid \neg z\right) = \exp\left(\lambda^s V_{vol}^s\right) \frac{\left(\lambda^s V_{vol}^s\right)^{\theta_k^s}}{\left(\theta_k^s\right)!}.$$

Taking the ratio of the two, we get

$$\frac{p\left(\theta_k^s \mid z\right)}{p\left(\theta_k^s \mid \neg z\right)} = 1 + P_G^s P_D^s \left(\frac{\theta_k^s}{\lambda^s V_{vol}^s} - 1\right).$$

The model likelihood, $p\left(z \mid \theta_{m:k}\right)$, can be recovered from the log odds ratio by

$$p\left(z \mid \theta_{m:k}\right) = 1 - \frac{1}{1 + 10^{l_k}}.$$

If a track is outside the LSR of the s^{th} source, then P_D^s should be zero and $p\left(z \mid \theta_{m:k}\right)$ will not be affected. In the case that the track leaves the GSR, we need a way to kill the track in order to make room for other tracks to be initialized. Thus, we assume that a track that has not received a measurement within τ_α seconds is no longer in the GSR and will be pruned. A track is considered a good track if the $p\left(z \mid \theta_{m:k}\right) > \tau_\rho$; otherwise, it is considered a poor track. A good track receives a unique indentifiable label. The parameters λ^s and P_D^s can be easily estimated and P_G^s is defined by the user.

VII. Track to Track Fusion

As tracks are propagated and updated they can begin to coalesce in which case we need to merge/fuse them. In order to optimally fuse them, we would need to calculate the cross covariance between two tracks which is computationally complex and time expensive, or in our case unknown. Fortunately, the covariance intersection (CI) method can be used to fuse two tracks together without calculating the cross covariance [14]. We use the variation of the CI method presented in [45] called the sampled covariance intersection (SCI) method. In order to see if two tracks need to be fused together, we use a track association method prescribed in [4]. In this section we will briefly present the track association and fusion methods adapted to the Lie group setting.

A. Track Association

Let $\hat{x}^i$ denote the state estimate of the i^{th} track and P^i be the corresponding error covariance. The estimation error between two tracks is denoted Δ^{ij} and defined as

$$\Delta^{ij} = \hat{x}^i \ominus \hat{x}^j.$$

Assuming that the two tracks are independent, the covariance of the estimation error is

$$T^{ij} = P^i + P^j.$$

We next define a metric $d_T : (G \times \mathfrak{g}) \times (G \times \mathfrak{g}) \to \mathbb{R}$ to be

$$d_T\left(\hat{x}^i, \hat{x}^j\right) = \left(\Delta^{ij}\right)^\top \left(T^{ij}\right)^{-1} \Delta^{ij}.$$

If $d_T\left(\cdot, \cdot\right) < \tau_S$ where $\tau_S \in \mathbb{R}$, then the two tracks are deemed to be the same and should be merged.

B. Track Fusion

Let $\hat{x}^1, \hat{x}^2, \cdots, \hat{x}^n$ denote the state estimates of the tracks that need to be fused together and P^i be their corresponding error covariance. The fused estimate in SCI is calculated as if the tracks to be fused are independent; therefore,

$$\begin{aligned}P^{-1} &= \sum_{i=1}^{n} \left(P^i\right)^{-1} \\ \hat{x}_{SCI} &= P\left(\sum_{i=1}^{n} \left(P^i\right)^{-1} \hat{x}^i\right).\end{aligned}$$

For Lie group settings, the state update is

$$\hat{x}_{SCI} = \hat{x}^1 \oplus \left(P \sum_{i=2}^{n} \left(\left(P^i \right)^{-1} \left(\hat{x}^i \ominus \hat{x}^1 \right) \right) \right),$$

The error covariance P^{-1} is overly optimistic, so we must adjust the size as follows:

- Generate N (about 100 for a good distribution) random samples x^j, $j = 1, 2, \ldots, N$ from $x \sim \mathcal{N}(0, P)$
- Find

$$r_{\max} = \max_{j=1,\ldots,N} \frac{\left(x^j \right)^\top P^{-1} x^j}{\max\limits_{i=1,\ldots,N} \left\{ \left(x^j \right)^\top \left(P^i \right)^{-1} x^j \right\}}$$

  and

$$r_{\min} = \min_{j=1,\ldots,N} \frac{\left(x^j \right)^\top P^{-1} x^j}{\max\limits_{i=1,\ldots,N} \left\{ \left(x^j \right)^\top \left(P^i \right)^{-1} x^j \right\}}$$

- Then set

$$P_{SCI} = \frac{P}{u r_{\min} + (1-u)\, r_{\max}}$$

  where $u \in [0, 1]$ is used to adjust the performance of the SCI algorithm. When $u = 1$, the fused covariance is conservative and when $u = 0$, the fused covariance is optimistic. The authors of [45] received good results by setting $u = 0.5$.

The fused track will have the state estimate $\hat{x}_{SCI}$ with error covariance P_{SCI}.

In addition, when tracks are fused, we merge their consensus sets together, give the merged track the highest model likelihood out of the tracks merged, and the label pertaining to the track that was first created out of the merged track. We note that it would be more correct to keep a history of the information needed to calculate the model's likelihood so that when tracks are merged we could recompute the model likelihood; however, such a task can require a lot of memory and computation depending on the amount of time merged models have existed.

VIII. RANSAC

RANSAC is a regression algorithm that estimates the parameters of a model while mitigating the effect of gross outliers [16]. There have been many developments and adaptations to the RANSAC algorithm [9]. RANSAC is used to initialize new targets. A target that is not initialized (not being tracked by R-RANSAC) is producing measurements captured by the measurement sources. Provided that the measurements of the un-initialized target are not in the validation region of any initialized target, its measurements are not associated with any existing initialized target; thus, they are given to the data tree. The measurements produced by the un-initialized target are neighboring and will eventually form a cluster [48].

RANSAC uses the measurements in the cluster to try to initialize a new target by first generating many track hypotheses. A track hypothesis is a current hypothetical state x_k^h estimated from a random subset of measurements from the cluster. We discuss the generation of track hypotheses in subsection VIII-A. Each track hypothesis is scored as discussed in subsection VIII-B. The track hypothesis that has the highest score provided that the score meets a minimum threshold is initialized into a new track as we will discuss in subsection VIII-C. In the case that there are multiple clusters, RANSAC is performed on each cluster. If a track is initialized, the measurements used to initialize the track are removed from the cluster.

A. Generating Track Hypotheses

To create a track hypothesis x_k^h, we take a minimum subset of measurements Y^c from the cluster where one measurement is from the current time step, and the other measurements are randomly sampled such that the system is observable. Using Y^c we seek to solve the optimizaiton problem

$$\max_{x_k} p\left(Y^c \mid x_k^h \right),$$

which is a maximum likelihood estimation problem that seeks to maximize the probability of the measurements Y^c conditioned on x_k^h by adjusting x_k^h. This optimization problem is equivalent to minimizing the log maximum likelihood estimation (LMLE) problem

$$\min_{x_k} -\log\left(p\left(Y^c \mid x_k \right) \right), \tag{16}$$

where we dropped the superscript h. Under the assumption that measurements are independent, they are also conditionaly independent which allows us to write the optimization problem (16) as

$$\min_{x_k} -\sum_{m=0}^{k} \sum_{y_m \in Y_m^c} \log\left(p\left(y_m \mid x_k \right) \right),$$

where $Y_m^c \subseteq Y^c$ contians the measurements observed at time m that are being used in the optimization scheme.

In order to solve the optimization problem, we need to invert the system model defined in (5) so that we can estimate a previous state x_m given the current state x_k. Ignoring the noise terms, we can propagate the pose of the system g_m to g_k by

$$g_k = g_m \exp\left(\delta_{m:k} u \right).$$

Solving for g_m yields

$$\begin{aligned} g_m &= \exp^{-1}\left(\delta_{m:k} u\right) g_k \\ &= \exp\left(-\delta_{m:k} u\right) g_k \\ &= \exp\left(\delta_{k:m} u\right) g_k. \end{aligned}$$

Thus, the inverse of the system model $f^{-1}\left(x_k, w_k, \delta_{k:m}\right)$ is defined as

$$g_m = g_k \oplus \left(\delta_{k:m} u_k + \delta_{k:m} w_k^g + \frac{\delta_{k:m}^2}{2} w_k^u\right) \tag{17a}$$

$$u_m = u_k + \delta_{k:m} w_k^u. \tag{17b}$$

and the estimated output at time m is

$$\hat{y}_m = h\left(f^{-1}, v_m\right), \tag{18}$$

where f^{-1} denotes $f^{-1}\left(x_k, w_k, \delta_{k:m}\right)$.

Now that we can invert the model, we can calculate the probability $p\left(y_m \mid x_k\right)$, which is, according to the total probability theorm,

$$p\left(y_m \mid x_k\right) = \int_{\delta x_m} p\left(y_m \mid x_m\right) p\left(x_m \mid x_k\right) d_{\delta x_m}.$$

We approximate this integral by linearizing $p\left(y_m \mid x_m\right)$ and $p\left(x_m \mid x_k\right)$ using a first order Taylor series;

$$\begin{aligned} p\left(x_m \mid x_k\right) \approx \eta \exp\Bigg(-\frac{1}{2}\left(\hat{x}_m \ominus f_0^{-1} + F_{k:m}^{-1} \delta x_m\right)^\top \\ \bar{Q}_{k:m}^{-1}\left(x_m \ominus f^{-1} + F_{k:m}^{-1} \delta x_m\right)\Bigg) \end{aligned}$$

$$\begin{aligned} p\left(y_m^j \mid x_m\right) \approx \eta \exp\Bigg(-\frac{1}{2}\left(y_m \ominus h\left(\hat{x}_m, 0\right) + H_m \delta x_m\right)^\top \\ \bar{R}_m^{-1}\left(y_m \ominus h\left(\hat{x}_m, 0\right) + H_m \delta x_m\right)\Bigg) \end{aligned}$$

where $f_0^{-1} = f^{-1}\left(x_k, 0, \delta_{k:m}\right)$,

$$F_{k:m}^{-1} = \left.\frac{\partial x_m \ominus f^{-1}\left(x_k, w_k, \delta_{k:m}\right)}{\partial x_m}\right|_{\substack{x_m = \hat{x}_m \\ w_k = 0}},$$

$$\begin{aligned} G_{k:m} &= \left.\frac{\partial x_m \ominus f^{-1}\left(x_k, w_m, \delta_{k:m}\right)}{\partial w_m}\right|_{\substack{x_m = \hat{x}_m \\ w_k = 0}} \\ &= \left.\frac{\partial \log(e)}{\partial e} \frac{\partial e}{\partial \left(f^{-1}\right)^{-1}} \frac{\partial \left(f^{-1}\right)^{-1}}{\partial f^{-1}} \frac{\partial f^{-1}}{\partial w_m}\right|_{\substack{x_m = \hat{x}_m \\ w_k = 0}} \\ &= \left.J_r^{-1}(e)\, \mathrm{Ad}\left(x_m\right)\left(-\mathrm{Ad}\left(f^{-1}\right)\right) \frac{\partial f^{-1}}{\partial w_m}\right|_{\substack{x_m = \hat{x}_m \\ w_k = 0}} \\ &= \left.\frac{\partial f^{-1}}{\partial w_m}\right|_{x_m = \hat{x}_m, w_m = 0} \\ &= \begin{bmatrix} J_r\left(\delta_{k:m} u_k\right) \delta_{k:m} & J_r\left(\delta_{k:m} u_k\right) \frac{\delta_{k:m}^2}{2} \\ 0 & I \delta_{k:m} \end{bmatrix}, \end{aligned}$$

$$\begin{aligned} e &= x_m \ominus f^{-1}\left(x_k, w_k, \delta_{k:m}\right), \\ \bar{Q}_{k:m} &= G_{k:m} Q G_{k:m}^\top, \qquad (19a) \\ \bar{R}_m &= V_m R V_m^\top, \qquad (19b) \end{aligned}$$

V_m is defined in (9), H_m is defined in (8) and η is a normalizing coefficient which we overload. Note that we have left off the superscripts on V_m, H_m, and $\bar{R}_m$ indicating which measurement source they correspond to.

The above linearization serves two purposes: first, it keeps the integral a Gaussian distribution and it exposes the random variable δx_m which we wish to integrate over.

We can choose $\hat{x}_m$ to be $f^{-1}\left(x_k, 0, \delta_{k:m}\right)$, since $f^{-1}\left(x_k, 0, \delta_{k:m}\right)$ is the mean of the linearized probability $p\left(x_m \mid x_k\right)$ and thus the most likely value. This allows us to simplify the linearized probability $p\left(x_m \mid x_k\right)$ to

$$p\left(x_m \mid x_k\right) \approx \eta \exp\left(-\frac{1}{2} \delta x_m^\top \left(G_{k:m} Q G_{k:m}^\top\right)^{-1} \delta x_m\right)$$

since $\hat{x}_m \ominus f^{-1}\left(x_k, 0, \delta_{k:m}\right) = 0$ and $F_{k:m}^{-1} = I$.

Multiplying the two linearized probabilities together yields

$$p\left(y_m \mid x_m\right) p\left(x_m \mid x_k\right) \approx \eta \int_{\delta x_m} \exp\left(-L_m\right) d\delta x_m$$

with

$$\begin{aligned} L_m = \frac{1}{2} \delta x_m^\top \bar{Q}_{k:m}^{-1} \delta x_m + \frac{1}{2}\left(y_m \ominus h\left(\hat{x}_m, 0\right) + H_m \delta x_m\right)^\top \\ \bar{R}_m^{-1}\left(y_m \ominus h\left(\hat{x}_m, 0\right) + H_m \delta x_m\right). \end{aligned}$$

In order to solve the integral, we decompose L_m into two functions $L_m\left(x_k, \delta x_m\right) + L_m\left(x_k\right)$ where the first function depends on x_k and δx_m and the latter function only depends on x_k. This will allow us to move the latter function out of the integral so that we have

$$\begin{aligned} \int_{\delta x_m} \exp\left(-L_m\right) d\delta x_m = \eta \exp\left(-L_m\left(x_k\right)\right) \\ \int \exp\left(-L_m\left(x_k, \delta x_m\right)\right) d\delta x_m \end{aligned}$$

By constructing $L_m\left(x_k, \delta x_m\right)$ to be quadratic in δx_m so that $\exp\left(-L_m\left(x_k, \delta x_m\right)\right)$ is a Gaussian distribution, the integral would be a constant, c,

$$\int \exp\left(-L_m\left(x_k, \delta x_m\right)\right) d\delta x_m = c,$$

regardles of δx_m. This allows us to absorb the constant integral into η such that

$$p\left(y_m \mid x_m\right) p\left(x_m \mid x_k\right) \approx \eta \exp\left(-L_m\left(x_k\right)\right).$$

To construct $L_m(x_k, \delta x_m)$, we take the first and second derivative of L_m w.r.t. δx_m to get

$$\begin{aligned}\frac{\partial L_m}{\delta x_m} &= \delta x_m^\top \bar{Q}_{k:m}^{-1} \\ &\quad + y_m \ominus h(\hat{x}_m, 0)^\top \bar{R}_m^{-1} H_m + \delta x_m^\top H_m^\top \bar{R}^{-1} H_m \\ \frac{\partial^2 L_m}{\delta x_m^2} &= H_m^\top \bar{R}_m^{-1} H_m + \bar{Q}_{k:m}^{-1} \\ &= \Psi^{-1}\end{aligned}$$

Setting the first derivative to zero and solving for δx_m gives us

$$\begin{aligned}\delta x_m^\top \Psi^{-1} &= -y_m \ominus h(\hat{x}_m, 0)^\top \bar{R}_m^{-1} H_m \\ \delta x_m &= -\Psi \left(H_m^\top \bar{R}_m^{-1} y_m \ominus h(\hat{x}_m, 0) \right).\end{aligned}$$

We can now define the quadratic function $L_m(x_k, \delta x_m)$ as

$$\begin{aligned}L_m(x_k, \delta x_m) = \frac{1}{2} \left(\delta x_m + \Psi \left(H_m^\top \bar{R}_m^{-1} y_m \ominus h(\hat{x}_m, 0) \right) \right)^\top \\ \Psi^{-1} \left(\delta x_m + \Psi \left(H_m^\top \bar{R}_m^{-1} y_m \ominus h(\hat{x}_m, 0) \right) \right)\end{aligned}$$

and solve for $L_m(x_k) = L_m - L_m(x_k, x_m)$

$$\begin{aligned}L_m(x_k) = &\frac{1}{2} \delta x_m^\top \bar{Q}^{-1} \delta x_m \\ &+ \frac{1}{2} \left(y_m \ominus h(\hat{x}_m, 0) + H_m \delta x_m \right)^\top \\ &\bar{R}_m^{-1} \left(y_m \ominus h(\hat{x}_m, 0) + H_m \delta x_m \right) \\ &- \frac{1}{2} \left(\delta x_m + \Psi \left(H_m^\top \bar{R}_m^{-1} y_m \ominus h(\hat{x}_m, 0) \right) \right)^\top \\ &\Psi^{-1} \left(\delta x_m + \Psi \left(H_m^\top \bar{R}_m^{-1} y_m \ominus h(\hat{x}_m, 0) \right) \right).\end{aligned}$$

With algebraic manipulation, using the Inversion Lemma in [44] and substituting in f_0^{-1} for $\hat{x}_m$ we get

$$\begin{aligned}L_m(x_k) = &\frac{1}{2} y_m \ominus h\left(f_0^{-1}, 0\right)^\top \\ &S_{k:m}^{-1} y_m \ominus h\left(f_0^{-1}, 0\right),\end{aligned}$$

where

$$S_{k:m} = \bar{R}_m + H_m \bar{Q}_{k:m} H_m^\top. \tag{20}$$

Since L_m is independent of δx_m,

$$\begin{aligned}p\left(y_m^j \mid x_k\right) &= \int_{x_m} p(y_m \mid x_m)\, p(x_m \mid x_k)\, d_{x_m} \\ &\approx \eta \exp\left(-L_m(x_k)\right).\end{aligned}$$

and the approximated LMLE is

$$\min_{x_k} \sum_{m=0}^{k} \sum_{j=1}^{N_m} L_m^j(x_k),$$

where N_m is the number of measurements at time step m. To calculate another track hypothesis x_k^h, we would repeat the optimization scheme with a different random subset.

There are many methods that can be used to solve the optimization problem. If the system is linear, the optimization problem reduces to the one described in [**?**]. If the system is non-linear, the optimization problem can be solved using the Gauss-newton method or some other optimization method.

B. Track Hypothesis Scoring

Once a track hypothesis is generated, we score it according to its inliers. Inliers are measurements that support the hypothetical track. To determine if a measurement is an inlier, we use a very similar idea as the validation region discussed in subsection V-B. The only difference is that we create the validation regions not only at the current time step but at previous ones also. We will ignore the derivation, since it is similar to subsection V-B, and present the definition.

Let $S_{k:m}^s$ be the innovation covariance of source s defined in (20). We defined a metric $d_I : N \times N \to \mathbb{R}$ on the measurement space

$$d_I(y_m, \hat{y}_m) = (y_m \ominus \hat{y}_m)^\top \left(S_{k:m}^s\right)^{-1} (y_m \ominus \hat{y}_m),$$

where y_m is a measurement, from time m, in the cluster from which the hypothetical track was generated, and $\hat{y}_m$ is the estimated measurement defined in 18. The measurement y_m is an inlier if

$$d_I\left(y_m^j, \hat{y}_m\right) < \tau_I,$$

where τ_I is an inlier threshold similar in function to the gate threshold discussed in subsection V-B. This similarity allows us to calcuated τ_I based on a probability P_I which we call the probability of being an inlier. The score of a track hypothesis is the number of times a measurement source produced an inlier at a certain time step. This means that if a source produced multiple inliers at time m, the score would only increase by 1 since we assume that at most only one of the measurements originated from the target.

C. Track Initialization

RANSAC generates up to ℓ hypothetical tracks. If the score of a hypothetical track reaches the threshold τ_{RE}, then RANSAC terminates generating hypothetical tracks and initializes a track with that hypothetical track. Othewise, RANSAC uses the best hypotetical track (provided that the hypothetical track has a minimum score of τ_{RM}) to initialize a new track.

To initialize a track, we propagate x_k^h back in time to the oldest time step of an inlier measurement denoted x_n^h, and initialize the error covariance P_n. Using the PDAF, we propgate and update x_n^h using it's inliers to procude a current estimate x_k and error covariance P_k. This process is called filtering. The inliers are also

used to calculate the model likelihood. Once the track is initialized, the inliers are added to the track's consensus set.

IX. LMLE Simplification

The optimization problem defined in (16) can be computationally expensive if the system is nonlinear. The purpose of the optimization problem is to generate a hypothetical track x_k^h which is later refined using its inlieries via filtering. So the hypothetical track produced by the LMLE doesn't have to be super refined; therefore, we can simplify the optimization process in different ways.

Let's assume that the pose of the system is measurable. In other words, your observation model h is defined as

$$h\left(x_k, v_k\right) = g_k \exp\left(v_k\right),$$

thus

$$y_k = g_k \exp\left(v_k\right).$$

When generating a track hypothesis, we always use a measurement from the current time. We can use this measurement as the estimated pose g_k. All that remains is estimating the velocity term u_k using one other measurement y_m where $m < k$. Under the assumption that g_k is the true current state, the measurement y_m is

$$y_m = g_k \exp\left(\delta_{k:m} u_k + \delta_{k:m} w_k^g + \frac{\delta_{k:m}^2}{2} w_k^u\right) \exp\left(v_m\right),$$

which can be approximated as

$$y_m \approx g_k \exp\left(\delta_{k:m} u_k + \delta_{k:m} w_k^g + \frac{\delta_{k:m}^2}{2} w_k^u + v_m\right)$$

By calculating the error between the measurement and the current state we get

$$\begin{aligned} y_m \ominus g_k &= \log\left(g_k^{-1} y_m\right) \\ e_{k:m} &\approx \delta_{k:m} u_k + \delta_{k:m} w_k^g + \frac{\delta_{k:m}^2}{2} w_k^u + v_m. \end{aligned}$$

The expected value of $e_{k:m}$ is

$$\mathrm{E}\left[e_{k:m}\right] = \delta_{k:m} u_k,$$

and the covariance is

$$\begin{aligned} \mathrm{cov}\left[e_{k:m}\right] &\approx \delta_{k:m}^2 Q^g + \frac{\delta_{k:m}^4}{4} Q^u + R \\ &= S_{k:m} \end{aligned}$$

where

$$Q = \begin{bmatrix} Q^g & 0 \\ 0 & Q^u \end{bmatrix}.$$

We can simply estimate u_k by letting $u_k = e_{k:m}/\delta_{k:m}$. We can enhance the estimate using multiple measurements. Let χ be an index set such that $\cup_{m\in\chi}\{y_m\}$ is the set of measurements we will use to estimate u_k, then

$$u_k = \left(\sum_{j\in\Lambda} \delta_{k:j} S_{k:j}^{-1}\right)^{-1} \left(\sum_{j\in\Lambda} S_{k:j}^{-1} e_{k:j}\right).$$

We can simplify the estimation further by setting the covariance terms to identity which yields

$$u_k = \frac{1}{|\chi|} \sum_{j\in\Lambda} e_{k:j}.$$

Another possible simplification for any approach is by setting $S_{k:m}^s$ defined in (20) to identity in the optimization problem (16).

We propose this simplification because computing the covariance $S_{k:m}^s$ can be computationally costly when it has to be done thousands of times whenever RANSAC is performed.

If you want the estimation to be as accurate as possible, we suggest you use the simplified approaches to seed the original optimization problem unless the system is linear since the original optimization problem is already simple.

X. Examples

In this section we provide three examples of increasing complexity. The first example is to show how Lie group framework of R-RANSAC applies to Euclidean spaces of $\mathbb{R}^n$ with $n \in \mathbb{N}$. The second example applies R-RANSAC to tracking targets on an image plane, and the third example applies R-RANSAC to tracking UAVs using radar sensors.

A. Euclidean Space $\mathbb{R}^n$

This is a brief section to show how the Lie group theory applies to Euclidean spaces of $\mathbb{R}^n$, where $n \in \mathbb{N}$, equipped with the standard topology and the group operation of addition. Its corresponding Lie algebra is $\mathbb{R}^n$, and the adjoint, exponential, and left and right Jacobians are the identity map. The box plus/minus and o plus/minus mapping are trivially derived; for example

$$\begin{aligned} g \oplus u &= g + u \\ g_1 \ominus g_2 &= g_1 - g_2 \end{aligned}$$

where $g, g_1, g_2 \in G = \mathbb{R}^2$ and $u \in \mathfrak{g} = \mathbb{R}^2$.

We can write the system model as

$$\begin{aligned} g_k &= g_{k^-} + \delta_k\left(u_{k^-} + w_k^g\right) + \frac{\delta_k^2}{2} w_k^u \\ u_k &= u_{k^-} + \delta_k w_k^u \\ y_k &= g_k + v_k, \end{aligned}$$

$$
\begin{aligned}
G_{k:m} &= \left.\frac{\partial x_m \ominus f^{-1}\left(x_k, w_m, \delta_{k:m}\right)}{\partial w_m}\right|_{x_m=\hat{x}_m, w_m=0} \\
&= \left.\frac{\partial \log\left(e\right)}{\partial e}\frac{\partial e}{\partial\left(f^{-1}\right)^{-1}}\frac{\partial\left(f^{-1}\right)^{-1}}{\partial f^{-1}}\frac{\partial f^{-1}}{\partial w_m}\right|_{x_m=\hat{x}_m, w_m=0} \\
&= J_r^{-1}\left(e\right)\mathrm{Ad}\left(x_m\right)\left(-\mathrm{Ad}\left(f^{-1}\right)\right)\left.\frac{\partial f^{-1}}{\partial w_m}\right|_{x_m=\hat{x}_m, w_m=0} \\
&= I\left.\frac{\partial f^{-1}}{\partial w_m}\right|_{x_m=\hat{x}_m, w_m=0}, \\
\frac{\partial f^{-1}}{\partial w_m} &= \begin{bmatrix} J_r\left(\delta_{k:m}u_k\right)\delta_{k:m} & J_r\left(\delta_{k:m}u_k\right)\frac{\delta_{k:m}^2}{2} \\ 0 & I\delta_{k:m} \end{bmatrix}, \\
e &= x_m \ominus f^{-1}\left(x_k, w_k, \delta_{k:m}\right),
\end{aligned}
$$

We can write this in the familiar matrix notation as

$$
\begin{aligned}
x_k &= A_k x_{k^-} + B_k w_k \\
y_k &= C x_k + v_k
\end{aligned}
$$

where $x_k = \left[g_k^\top, u_k^\top\right]^\top$,

$$
A_k = \begin{bmatrix} I & \delta_k I \\ 0 & I \end{bmatrix}, \qquad B_k = \begin{bmatrix} \delta_k I & \frac{\delta_k^2}{2} I \\ 0 & \delta_k I \end{bmatrix},
$$

and

$$
C = \begin{bmatrix} I_{2\times 2} & 0_{2\times 2} \end{bmatrix}.
$$

In this form, we can easily recognize it as an linear time-invariant system with near constant velocity.

The Jacobians are easily computed and are $F_k = A_k$, $G_k = B_k$, $H_k = C$, $F_{k:m} = CA_{k:m}$, and $G_{k:m} = CB_{k:m}$. For linear systems, the LMLE simplifies to a least squares problem. This type of system used with R-RANSAC is thoroughly discussed in [**?**].

B. Constrained Problem for $SE\left(2\right)$: Tracking in the Image Plane

NOTE: THIS WOULD BE BETTER IF TRACKING WAS DONE IN THE VIRTUAL IMAGE PLANE SINCE THE VIP IS PARALLEL TO THE GROUND, THUS THE MOTION WOULD REALLY BE SE2

Suppose we have a camera mounted on a UAV tracking a moving target that is restricted to a planar surface. The global frame is the current image frame, and the measurements are point measurements, i.e. pixels. In this case, the LSR and the GSR are the same. The volume of the LSR is the total number of pixels. Since the UAV is moving, the global frame is also moving. This means that we must provide a transformation T so that we can transform all measurements, state estimates and error covariances to the current global frame. We do this using the Homography [25] and the method described in [46].

The Lie group configuration of a target moving in the image plane is $SE\left(2\right)$, see appendix A for the system function and important Jacobians. Let $g_k \in SE\left(2\right)$ and $u_k \in \mathfrak{se}\left(2\right)$. They can be written as

$$
\begin{aligned}
g_k &= \begin{bmatrix} R_k & t_k \\ 0_{1\times 2} & 1 \end{bmatrix} \\
u_k &= \begin{bmatrix} \omega_k & \rho_k \\ 0_{1\times 2} & 1 \end{bmatrix},
\end{aligned}
$$

where $R_k \in SO\left(2\right)$, $\omega_k \in \mathfrak{se}\left(2\right)$, and $t_k, \rho_k \in \mathbb{R}^2$. The observation model is

$$
\begin{aligned}
y_k &= h\left(x_k, v_k\right) \\
&= t_k + v_k,
\end{aligned}
$$

where $y_k \in \mathbb{R}^2$. The inverse observation model is

$$
\begin{aligned}
y_m &= h\left(f^{-1}\left(x_k, \delta_{k:m}, w_k\right), v_k\right) \\
&= R_k J_l\left(\delta_{k:m}\left(\omega_k + w_k^R\right) + \frac{\delta_{k:m}^2 w_k^\omega}{2}\right)\left(\delta_{k:m}\left(\rho_k + w_k^\rho\right)\right) \\
&+ t_k + w_k^t + v_m,
\end{aligned}
$$

where $w_k = \left[\left(w_k^t\right)^\top, \left(w_k^R\right)^\top, \left(w_k^\rho\right)^\top, \left(w_k^u\right)^\top\right]^\top$ and J_l is the left Jacobian of $SO\left(2\right)$ defined in equation (26). Based on the observation model, we have three Jacobians to compute: $G_{k:m}$, $H_{k:m}$ and V_m.

$$
H_{k:m} = \frac{\partial h\left(f^{-1}, v_k\right)}{\partial x} = \begin{bmatrix} H_{k:m}^t & H_{k:m}^R & H_{k:m}^\rho & H_{k:m}^\omega \end{bmatrix},
$$

where

$$
\begin{aligned}
H_{k:m}^t &= \frac{\partial h\left(f^{-1}, v_k\right)}{\partial t_k} = R_k \\
H_{k:m}^R &= \frac{\partial h\left(f^{-1}, v_k\right)}{\partial R} = \delta_{k:m} R_k \left[1\right]_\times J_l\left(\delta_{k:m}\omega\right)\rho \\
H_{k:m}^\rho &= \frac{\partial h\left(f^{-1}, v_k\right)}{\partial \rho} = \delta_{k:m}^2 R_k J_l\left(\delta_{k:m}\omega\right) \\
H_{k:m}^\omega &= \frac{\partial h\left(f^{-1}, v_k\right)}{\partial \omega} = \delta_{k:m}^2 R_k J_l^{'}\left(\delta_{k:m}\omega\right)\rho
\end{aligned}
$$

and

$$
J_l^{'}\left(\omega\right) = \frac{\cos\left(\theta\right)\theta - \sin\left(\theta\right)}{\theta^2} I + \frac{\sin\left(\theta\right)\theta - \left(1 - \cos\left(\theta\right)\right)}{\theta^2}\left[1\right]_\times.
$$

We quickly note that

$$
H_k = H_{k:k} = \begin{bmatrix} I & 0_{2\times 4} \end{bmatrix}.
$$

$$
G_{k:m} = \frac{\partial h\left(f^{-1}, v_k\right)}{\partial w_k} = \begin{bmatrix} G_{k:m}^t & G_{k:m}^R & G_{k:m}^\rho & G_{k:m}^\omega \end{bmatrix},
$$

where

$$G^t_{k:m} = \frac{\partial h\left(f^{-1}, v_k\right)}{\partial w^t_k} = I$$
$$G^R_{k:m} = \frac{\partial h\left(f^{-1}, v_k\right)}{\partial w^R_k} = \delta^2_{k:m} R_k V^{'}\left(\delta_{k:m}\omega\right)\rho$$
$$G^\rho_{k:m} = \frac{\partial h\left(f^{-1}, v_k\right)}{\partial w^\rho_k} = \delta^2_{k:m} R_k V\left(\delta_{k:m}\omega\right)$$
$$G^\omega_{k:m} = \frac{\partial h\left(f^{-1}, v_k\right)}{\partial w^\omega_k} = \frac{\delta^3_{k:m}}{2} R_k V^{'}\left(\delta_{k:m}\omega\right)\rho.$$

Lastly $V_m = I$.

Now that we have the Jacobians computed, we need to ensure that the system is observable. Since the system is nonlinear, the best we can do is ensure local observability. Given measurements from three different time steps, the observability matrix is

$$\mathcal{O} = \begin{bmatrix} H_{k:m_1} \\ H_{k:m_2} \\ H_{k:m_3} \end{bmatrix}.$$

The observability matrix is not full rank. However, since we are interested in tracking the target we only need to be able to estimate the position of the target. To do this, we consider two configurations for a constant velocity system. In the first configuration the target has a circular trajectory. In this case we add the constraints that the target's heading is oriented along the body frame velocity, that $\rho = [\rho_x, 0]^\top$, $\rho_x > 0$ and $\omega \neq 0$. Using these constraints, the system is observable. In the second configuration the target is moving in a straight line. In this configuration we impose the same constraints as in the first configuration except that $\omega = 0$. Under these constraints the system is observable. We can identify which configuration the target is in by looking at three different measurements from different times. If the three measurement do not form a line, then the system has first constrains; otherwise, the latter.

To improve the LMLE used to estimate the current state, we can seed it as follows. Let y_k, y_m, and y_n be three measurements taken at different times. Since we observe the position of the target, we can approximate the position of the target at time k, m, and n as $t_k = y_k$, $t_m = y_m$ and $t_n = y_n$. Using these estimated positions, we can numerically approximate the derivative as

$$\dot{t}_k = \frac{t_m - t_k}{\delta_{k:m}},$$
$$\dot{t}_m = \frac{t_n - t_m}{\delta_{n:m}}$$

Using the constraint that the heading of the UAS is aligned with the translational velocity, we get that

$$R = \frac{1}{\left\|\dot{t}_k\right\|}\begin{bmatrix} \dot{t}_{k_x} & -\dot{t}_{k_y} \\ \dot{t}_{k_y} & \dot{t}_{k_x} \end{bmatrix}$$

where

$$\dot{t}_k = \begin{bmatrix} \dot{t}_{k_x} \\ \dot{t}_{k_y} \end{bmatrix},$$

and that

$$\rho_k = \begin{bmatrix} \left\|\dot{t}_k\right\| \\ 0 \end{bmatrix}.$$

Note that

$$\dot{t}_m = R_k \exp\left(\delta_{k:m}\dot{\theta}_k\right)\rho_k,$$

where $\dot{\theta}_k$ is the angular velocity. Let

$$z = R_k^\top \dot{t}_m / \rho_k$$
$$= [z_x, z_y]^\top,$$

then

$$\dot{\theta}_k = \text{atan2}\left(z_y, z_x\right)/\delta_{k:m}$$

These estimates can then be used to seed the LMLE during the RANSAC step of the algorithm.

Hardware Results

C. Constrained Problem for SE(3): Tracking UAVs with Radar

In this example the objective is to track multiple fixed wing aircrafts (FWAs) that are in the GSR and whose configuration manifold is $SE(3)$. The FWA are observed via multiple fixed radar sensors with overlapping LSR which produce point measurements. See appendix B for the FWAs' system function and important Jacobians.

Let $g_k \in SE\left(3\right)$ and $u_k \in \mathfrak{se}\left(3\right)$. These elements can be written as

$$g_k = \begin{bmatrix} R_k & t_k \\ 0_{1\times 2} & 1 \end{bmatrix}$$
$$u_k = \begin{bmatrix} \omega_k & \rho_k \\ 0_{1\times 2} & 1 \end{bmatrix},$$

where $R_k \in SO\left(3\right)$, $\omega_k \in \mathfrak{so}\left(3\right)$, and $t_k, \rho_k \in \mathbb{R}^3$. The observation model is

$$y_k = h\left(x_k, v_k\right)$$
$$= t_k + v_k,$$

where $y_k \in \mathbb{R}^3$. The inverse observation model is

$$y_m = h\left(f^{-1}\left(x_k, \delta_{k:m}, w_k\right), v_k\right)$$
$$= R_k J_l\left(\delta_{k:m}\left(\omega_k + w^R_k\right) + \frac{\delta^2_{k:m} w^\omega_k}{2}\right)\left(\delta_{k:m}\rho_k + \delta_{k:m} w^\rho_k\right)$$
$$+ t_k + w^t_k + v_m,$$

where $w_k = \left[\left(w^t_k\right)^\top, \left(w^R_k\right)^\top, \left(w^\rho_k\right)^\top, \left(w^u_k\right)^\top\right]^\top$ and J_l is defined in equation (32). Based on the observation model, we have three Jacobians to compute: $G_{k:m}$, $H_{k:m}$ and V_m.

$$H_{k:m} = \frac{\partial h\left(f^{-1}, v_k\right)}{\partial x} = \begin{bmatrix} H^t_{k:m} & H^R_{k:m} & H^\rho_{k:m} & H^\omega_{k:m} \end{bmatrix},$$

where

$$H^t_{k:m} = \frac{\partial h\left(f^{-1}, v_k\right)}{\partial t_k} = R_k$$
$$H^R_{k:m} = \frac{\partial h\left(f^{-1}, v_k\right)}{\partial R} = \delta_{k:m} R_k \left[-J_l\left(\delta_{k:m}\omega\right)\rho\right]_\times$$
$$H^\rho_{k:m} = \frac{\partial h\left(f^{-1}, v_k\right)}{\partial \rho} = \delta^2_{k:m} R_k J_l\left(\delta_{k:m}\omega\right)$$
$$H^\omega_{k:m} = \frac{\partial h\left(f^{-1}, v_k\right)}{\partial \omega} = \delta^2_{k:m} R_k \partial J_l\left(\delta_{k:m}\omega, \delta_{k:m}\rho\right),$$

and$\partial J_l\left(\delta_{k:m}\omega, \delta_{k:m}\rho\right)$ is defined in Lemma 1.

$$G_{k:m} = \frac{\partial h\left(f^{-1}, v_k\right)}{\partial w_k} = \begin{bmatrix} G^t_{k:m} & G^R_{k:m} & G^\rho_{k:m} & G^\omega_{k:m} \end{bmatrix},$$

where

$$G^t_{k:m} = \frac{\partial h\left(f^{-1}, v_k\right)}{\partial w^t_k} = I$$
$$G^R_{k:m} = \frac{\partial h\left(f^{-1}, v_k\right)}{\partial w^R_k} = \delta^2_{k:m} R_k \partial J_l\left(\delta_{k:m}\omega, \delta_{k:m}\rho\right)$$
$$G^\rho_{k:m} = \frac{\partial h\left(f^{-1}, v_k\right)}{\partial w^\rho_k} = \delta^2_{k:m} R_k J_l\left(\delta_{k:m}\omega\right)$$
$$G^\omega_{k:m} = \frac{\partial h\left(f^{-1}, v_k\right)}{\partial w^\omega_k} = \frac{\delta^3_{k:m}}{2} R_k \partial J_l\left(\delta_{k:m}\omega, \delta_{k:m}\rho\right).$$

Lastly $V_m = I$.

Now that we have the Jacobians computed, we need to ensure that the system is observable. Since the system is nonlinear, the best we can do is ensure local observability. Given measurements from four different time steps, the observability matrix is

$$\mathcal{O} = \begin{bmatrix} H_{k:m_1} \\ H_{k:m_2} \\ H_{k:m_3} \\ H_{k:m_4} \end{bmatrix}.$$

The observability matrix is not full rank. However, under certain constraints, the system is observable. Since we are interested in tracking the target we only need to be able to estimate the position of the target. To do this, we consider two configurations for a constant velocity system. In the first configuration the target has a circular or spiral trajectory. In this case we add the constraints that the target's heading is oriented along the body frame velocity, that $\rho = [\rho_x, 0]^\top$, $\rho_x > 0$ and $\omega \neq 0$. Using these constraints, the system is observable. In the second configuration the target is moving in a straight line. In this configuration we impose the same constraints as in the first configuration except that $\omega = 0$ and the roll in attitude (using roll, pitch, yaw) is zero. Under these constraints the system is observable. We can identify which configuration the target is in by looking at three different measurements from different times. If the three measurement do not form a line, then the system is in the first configuration; otherwise, the latter.

Since the system is nonlinear, the LMLE can take a while to converge. We can speed up the LMLE by seeding it. To do this, we will use the Euler angles (roll ϕ, pitch θ, yaw ψ) as local coordinates and align the pitch and yaw angles with the translational velocities. The map from Euler angles configured in an NED frame to $SO\left(3\right)$ is defined as

$$\mathcal{R}\left(\phi, \theta, \psi\right) = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix}. \tag{21}$$

Let t_k, t_m, t_n denote measured positions at times n, m, k with k being the current time and $n < m < k$. The derivative of the position vector t is related to ρ through the rotation matrix R. This allows us to write the constraint

$$\begin{aligned} \dot{t} &= R\rho \\ &= \mathcal{R}\left(\phi, \theta, \psi\right)\rho. \end{aligned}$$

Letting $\dot{t}_k = \left[\dot{t}_{k_x}, \dot{t}_{k_y}, \dot{t}_{k_z}\right]^\top$, we get

$$\begin{bmatrix} \dot{t}_x \\ \dot{t}_y \\ \dot{t}_z \end{bmatrix} = \underbrace{\begin{bmatrix} c_\theta c_\psi \\ c_\theta s_\psi \\ -s_\theta \end{bmatrix}}_{R_{k_x}} \rho_x.$$

Since we receive position measurements, we can numerically estimate the positional derivatives $\dot{t}_k = \frac{t_m - t_k}{\delta_{k:m}}$ and $\dot{t}_m = \frac{t_n - t_m}{\delta_{m:n}}$. Using $\dot{t}_k$ we can estimate R_k by solving for ϕ_k,θ_k, ψ_k, and ρ_{k_x} using the equations

$$\begin{aligned} \psi_k &= \text{atan2}\left(\dot{t}_{k_y}, \left|\dot{t}_{k_x}\right|\right) \\ \theta_k &= \arctan\left(\frac{-\cos\left(\psi_k\right)\dot{t}_{k_z}}{\dot{t}_{k_x}}\right) \\ \phi_k &= 0 \\ \rho_{k_x} &= \left\|\dot{t}_k\right\| \end{aligned}$$

We have set $\phi_k = 0$ since we cannot easily estimate it unless we assume the FWA is making a coordinated turn as described in [6]. In the absence of sideslip, a coordinated turn has the constraint

$$\dot{\psi} = \frac{g}{\left\|\dot{t}\right\|}\tan\left(\phi\right),$$

where g is the gravitational constant. Since we can estimate ψ at two different time periods, we can numerically estimate its derivative $\dot{\psi}$. Solving for ϕ we get

$$\phi_k = \arctan\left(\frac{\frac{\psi_k - \psi_m}{\delta_{k:m}}\left\|\dot{t}_k\right\|}{g}\right).$$

From the estimated Euler angles ϕ_k, θ_k and ψ_k we can construct the estimated rotation R_k using equation (21).

Lastly we estimate ω_k using the constraint

$$\dot{t}_m = R_k \exp\left(\delta_{k:m}\omega_k\right)\rho.$$

Using a first order Taylor series approximation for the matrix exponential, we get

$$\frac{R_k^\top \dot{t}_m}{\rho_{k_x}} = \begin{bmatrix}1\\0\\0\end{bmatrix} + \begin{bmatrix}0\\ \omega_{k_z}\\ -\omega_{k_y}\end{bmatrix}\delta_{k:m},$$

From which we can estimate the angular rates ω_{k_z} and ω_{k_y}, and we simply set $\omega_{k_x} = 0$. These approximations are then used to seed the LMLE optimization problem.

Hardware Results

APPENDIX A
SE(2)

The special Euclidean group of two dimensions is a matrix Lie group and is the set

$$SE\left(2\right) = \left\{ \begin{bmatrix} R & t \\ 0_{1\times 2} & 1_{1\times 1}\end{bmatrix} \middle| \; R \in SO\left(2\right) \text{ and } t \in \mathbb{R}^2 \right\}$$

equipped with matrix multiplication. The Lie algebra is

$$\mathfrak{se}\left(2\right) = \left\{ \begin{bmatrix} \omega & \rho \\ 0 & 0\end{bmatrix} \middle| \; \omega \in \mathfrak{so}\left(2\right) \text{ and } \rho \in \mathbb{R}^2 \right\},$$

where $\theta \in \mathbb{R}$,$\omega = \left[\theta\right]_\times$, and $\left[\cdot\right]_\times$ is the skew symmetric operator defined as

$$\left[\theta\right]_\times = \begin{bmatrix}0 & -\theta\\ \theta & 0\end{bmatrix},$$

and

$$\begin{bmatrix}\rho\\ \theta\end{bmatrix} \in \mathbb{R}^3$$

which is the Cartesian vector space.

Let $g \in SE\left(2\right)$ and $u \in \mathfrak{se}\left(2\right)$. Using an element of the Lie algebra, we can define the infinitesimal generator $\xi_u : G \to TG$ as

$$\begin{aligned}\xi_u\left(g\right) &= gu\\ &= \dot{g}.\end{aligned}$$

The discretized state transition function $f : SE\left(2\right) \times \mathbb{R}^2 \to SE\left(2\right) \times \mathbb{R}^2$ is

$$\begin{aligned} g_k &= \text{Exp}\left(\delta_k\left(u_{k^-} + w_k^g\right) + \frac{\delta_k^2}{2} w_k^u\right)\\ u_k &= u_{k^-} + \delta_k w_k^u\end{aligned}$$

A. Group Operations

Let $g_1, g_2 \in SE\left(2\right)$. The inversion and group multiplication is matrix inversion and matrix multiplication as follows

$$\begin{aligned} g_1^{-1} &= \begin{bmatrix} R_1^\top & -R_1^\top t_1\\ 0 & 1\end{bmatrix}\\ g_1 g_2 &= \begin{bmatrix} R_1 R_2 & R_1 t_2 + t_1\\ 0 & 1\end{bmatrix}.\end{aligned}$$

B. Exponential Map

The exponential map is the matrix exponential and it's inverse is the matrix logarithm. The matrix exponential is a surjective function; however, by restricting the domain to

$$U = \left\{u \in \mathfrak{se}\left(2\right) \mid \left\|u\right\| < \pi\right\},$$

it becomes a bijection. In this case, the exponential map and its inverse have a simplified form

$$\begin{aligned}\exp\left(u\right) &= \begin{bmatrix}\exp\left(\omega\right) & J_l\left(\omega\right)\rho\\ 0_{1\times 2} & 1\end{bmatrix}\\ \log\left(g\right) &= \begin{bmatrix}\log\left(R\right) & J_l^{-1}\left(\omega\right)t\\ 0_{1\times 2} & 0\end{bmatrix}\end{aligned}$$

with

$$\theta = \omega^\vee \tag{22}$$

$$\exp\left(\omega\right) = \begin{bmatrix}\cos\left(\theta\right) & -\sin\left(\theta\right)\\ \sin\left(\theta\right) & \cos\left(\theta\right)\end{bmatrix} \tag{23}$$

$$\log\left(R\right) = \arctan\left(R_{21}, R_{11}\right) \tag{24}$$

$$\omega = \log\left(R\right) \tag{25}$$

$$J_l\left(\omega\right) = \begin{cases} I & \text{if } \theta = 0\\ \frac{\sin(\theta)}{\theta} I + \frac{1-\cos(\theta)}{\theta}\left[1\right]_\times & \text{else}\end{cases} \tag{26}$$

$$J_l^{-1}\left(\omega\right) = \begin{cases} I & \text{if } \theta = 0\\ \frac{\theta\sin(\theta)}{2(1-\cos(\theta))} I - \frac{\theta}{2}\left[1\right]_\times & \text{else}\end{cases} \tag{27}$$

and J_l being the left Jacobian of $SO\left(2\right)$

C. Jacobians

1) Adjoint: The group adjoint representation is

$$\mathbf{Ad}_g = \begin{bmatrix} R & -\left[1\right]_\times t\\ 0_{1\times 2} & 1\end{bmatrix}$$

with inverse

$$\mathbf{Ad}_g^{-1} = \begin{bmatrix} R^\top & R^\top\left[1\right]_\times t\\ 0_{1\times 2} & 1\end{bmatrix}.$$

The Lie algebra adjoint representation is

$$\mathbf{ad}_u = \begin{bmatrix}\left[\omega\right]_\times & -\left[1\right]_\times \rho\\ 0_{1\times 2} & 0\end{bmatrix}.$$

2) Left and Right Jacobians: Let

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

then

$$J_r(u) = \begin{bmatrix} W_r(\theta) & D_r(\theta)\rho \\ 0 & 1 \end{bmatrix},$$
$$J_l(u) = \begin{bmatrix} W_l(\theta) & D_l(\theta)\rho \\ 0 & 1 \end{bmatrix}$$
$$J_r^{-1}(u) = \begin{bmatrix} W_r^{-1}(\theta) & -W_r^{-1}(\theta) D_r(\theta)\rho \\ 0 & 1 \end{bmatrix}$$
$$J_l^{-1}(u) = \begin{bmatrix} W_l^{-1}(\theta) & -W_l^{-1}(\theta) D_l(\theta)\rho \\ 0 & 1 \end{bmatrix}.$$

If $\theta = 0$, then the Jacobians and their inverses are the identity function.

APPENDIX B
SE(3)

The special Euclidean group of three dimensions is a matrix Lie group and is the set

$$SE(3) = \left\{ \begin{bmatrix} R & t \\ 0_{1\times 2} & 1_{1\times 1} \end{bmatrix} \middle| \; R \in SO(3) \text{ and } t \in \mathbb{R}^3 \right\}$$

equipped with matrix multiplication. The Lie algebra is

$$\mathfrak{se}(3) = \left\{ \begin{bmatrix} \omega & \rho \\ 0 & 0 \end{bmatrix} \middle| \; \omega \in \mathfrak{so}(3) \text{ and } \rho \in \mathbb{R}^3 \right\},$$

where $\omega = [\theta]_\times$, $\theta = [\theta_1, \theta_2, \theta_3]^\top \in \mathbb{R}^3$ and $[\cdot]_\times$ is the skew symmetric operator defined as

$$[\theta]_\times = \begin{bmatrix} 0 & -\theta_3 & \theta_2 \\ \theta_3 & 0 & -\theta_1 \\ -\theta_2 & \theta_1 & 0 \end{bmatrix},$$

and

$$\begin{bmatrix} \rho \\ \theta \end{bmatrix} \in \mathbb{R}^6$$

which is the Cartesian vector space.

Let $g \in SE(3)$ and $u \in \mathfrak{se}(3)$. Using an element of the Lie algebra, we can define the infinitesimal generator $\xi_u : G \to TG$ as

$$\begin{aligned} \xi_u(g) &= gu \\ &= \dot{g}. \end{aligned}$$

The discretized state transition function $f : SE(3) \times \mathbb{R}^3 \to SE(3) \times \mathbb{R}^3$ is

$$g_k = \exp\left(\delta_k (u_{k^-} + w_k^g) + \frac{\delta_k^2}{2} w_k^u\right)$$
$$u_k = u_{k^-} + \delta_k w_k^u$$

A. Group Operations

Let $g_1, g_2 \in SE(3)$. The inversion and group multiplication is matrix inversion and matrix multiplication as follows

$$g_1^{-1} = \begin{bmatrix} R_1^\top & -R_1^\top t_1 \\ 0 & 1 \end{bmatrix}$$
$$g_1 g_2 = \begin{bmatrix} R_1 R_2 & R_1 t_2 + t_1 \\ 0 & 1 \end{bmatrix}.$$

B. Exponential Map

The exponential map is the matrix exponential and it's inverse is the matrix logarithm. The matrix exponential is a surjective function; however, by restricting the domain to

$$U = \{u \in \mathfrak{se}(3) \mid \|u\| < \pi\},$$

it becomes a bijection. In this case, the exponential map and its inverse have a simplified form

$$\exp(u) = \begin{bmatrix} \exp(\omega) & J_l(\omega)\rho \\ 0 & 1 \end{bmatrix}$$
$$\log(g) = \begin{bmatrix} J_l^{-1}(\omega) t \\ \log(R) \end{bmatrix}$$

with $g = \exp(u)$,

$$\phi = \sqrt{\omega^\top \omega} \tag{28}$$
$$\exp(\omega) = I + \frac{\sin(\phi)}{\phi}\omega + \frac{1-\cos(\phi)}{\phi^2}\omega^2 \tag{29}$$
$$\phi = \arccos\left(\frac{\text{trace}(R) - 1}{2}\right) \tag{30}$$
$$\log(R) = \begin{cases} 0_{2\times 2} & \text{if } \phi = 0 \\ \frac{\phi}{2\sin(\phi)}\left(R - R^\top\right) & \text{else} \end{cases} \tag{31}$$
$$J_l(\omega) = \begin{cases} I & \text{if } \phi = 0 \\ I + \frac{1-\cos(\phi)}{\phi^2}\omega + \frac{\phi - \sin(\phi)}{\phi^3}\omega^2 & \text{else} \end{cases} \tag{32}$$
$$J_l^{-1}(\omega) = \begin{cases} I & \text{if } \theta = 0 \\ I - \frac{\omega}{2} + \left(\frac{1}{\phi^2} - \frac{\phi\sin(\phi)}{2\phi^2(1-\cos(\phi))}\right)\omega^2 & \text{else} \end{cases} \tag{33}$$

where J_l is the left Jacobian of $SO(3)$.

C. Jacobians

1) Adjoint: The group adjoint representation is

$$\mathbf{Ad}_g = \begin{bmatrix} R & [t]_\times R \\ 0_{3\times 3} & R \end{bmatrix}$$

with inverse

$$\mathbf{Ad}_g^{-1} = \begin{bmatrix} R^\top & -R^\top [t]_\times \\ 0_{3\times 3} & R^\top \end{bmatrix}.$$

The Lie algebra representation is

$$\mathbf{ad}_u = \begin{bmatrix} \omega & [\rho]_\times \\ 0_{3\times 3} & \omega \end{bmatrix}.$$

2) Left and Right Jacobians: Let $u = \begin{bmatrix} \omega & \rho \\ 0 & 0 \end{bmatrix} \in \mathfrak{se}(3)$ and

$$\begin{aligned}
\phi &= \sqrt{\omega^\top \omega} \\
a_\phi &= \frac{\cos(\phi) - 1}{\phi^2} \\
b_\phi &= \frac{\phi - \sin(\phi)}{\phi^3} \\
c_\phi &= -\frac{1}{\phi^3}\sin(\phi) + 2\left(\frac{1-\cos(\phi)}{\phi^4}\right) \\
d_\phi &= -\frac{2}{\phi^4} + \frac{3}{\phi^5}\sin(\phi) - \frac{1}{\phi^4}\cos(\phi) \\
q_r(\omega) &= \left(\left(\omega^\vee\right)^\top \rho\right)\left(d_\phi \omega^2 + c_\phi \omega\right) \\
q_l(\omega) &= \left(\left(\omega^\vee\right)^\top \rho\right)\left(d_\phi \omega^2 - c_\phi \omega\right) \\
B_r(u) &= q_r(\omega) + a_\phi [\rho]_\times + b_\phi\left(\omega[\rho]_\times + [\rho]_\times \omega\right) \\
B_l(u) &= q_l(\omega) - a_\phi [\rho]_\times + b_\phi\left(\omega[\rho]_\times + [\rho]_\times \omega\right) \\
J_r(\omega) &= I + \frac{\cos(\phi)-1}{\phi^2}\omega + \frac{\phi - \sin(\phi)}{\phi^3}\omega^2 \\
J_r^{-1}(\omega) &= I + \frac{1}{2}\omega - \frac{\phi\cot\left(\frac{\phi}{2}\right) - 2}{2\phi^2}\omega^2
\end{aligned}$$

then

$$\begin{aligned}
J_r(u) &= \begin{bmatrix} J_r(\omega) & B_r(u) \\ 0 & J_r(\omega) \end{bmatrix} \\
J_l(u) &= \begin{bmatrix} J_l(\omega) & B_l(u) \\ 0 & J_l(\omega) \end{bmatrix} \\
J_r^{-1}(u) &= \begin{bmatrix} J_r^{-1}(\omega) & -J_r^{-1}(\omega) B_r(u) J_r^{-1}(\omega) \\ 0 & J_r^{-1}(\omega) \end{bmatrix} \\
J_l^{-1}(u) &= \begin{bmatrix} J_l^{-1}(\omega) & -J_l^{-1}(\omega) B_l(u) J_l^{-1}(\omega) \\ 0 & J_l^{-1}(\omega) \end{bmatrix}.
\end{aligned}$$

Appendix C
Lemma

Lemma 1. *Let $J_l(\omega)$ denote the left Jacobian of $SO(3)$ and $\rho \in \mathbb{R}^3$. The derivative of $J_l(\omega)\, p$ with respect to ω, denote $\partial J_l(\omega, \rho)$, is*

$$\begin{aligned}
\partial J_l(\omega, \rho) = &\left(\frac{1-\cos(\theta)}{\theta^2}\right)[-\rho]_\times \\
&+ \left(\frac{\theta - \sin(\theta)}{\theta^3}\right)\left(-2\omega[\rho]_\times + [\rho]_\times \omega_\times\right) \\
&+ \left(\frac{\sin(\theta)\,\theta + 2(\cos(\theta) - 1)}{\theta^4}\right)\omega\rho\left(\omega^\vee\right)^\top \\
&+ \left(\frac{3\sin(\theta) - \cos(\theta)\,\theta - 2\theta}{\theta^5}\right)\omega^2\rho\left(\omega^\vee\right)^\top.
\end{aligned}$$

where $\omega \in \mathfrak{so}(3)$, and $\theta = \sqrt{\left(\omega^\vee\right)^\top \omega^\vee}$.

Proof: $J_l(\omega)$ is defined in equation (32). It follows directly that

$$\frac{\partial J_l(\omega)\,\rho}{\partial \omega} = \frac{\partial}{\partial \omega}\left(I + \frac{1-\cos(\theta)}{\theta^2}\omega + \frac{\theta - \sin(\theta)}{\theta^3}\omega^2\right)\rho.$$

Using the product rule, we can break up the derivation into parts:

$$\begin{aligned}
\frac{\partial}{\partial \omega}\frac{1-\cos(\theta)}{\theta^2} &= \frac{\sin(\theta)\,\theta + 2(\cos(\theta)+1)}{\theta^3}\left(\omega^\vee\right)^\top \\
\frac{\partial}{\partial \omega}\frac{\theta - \sin(\theta)}{\theta^3} &= \frac{3\sin(\theta) - \cos(\theta)\,\theta - 2\theta}{\theta^5} \\
\frac{\partial}{\partial \omega}\omega\rho &= [-\rho]_\times
\end{aligned}$$

and

$$\frac{\partial}{\partial \omega}\omega^2\rho = -2\omega[\rho]_\times + [\rho]_\times \omega_\times.$$

Putting the pieces together, we get solution stated in the lemma. ∎

Appendix D
Terminology and Parameters

The terminology, parameters, and notation used is previous version of R-RANSAC have varied which has created confusion and inconsistencies. In this section, we hope to solidify certain terminology, parameters and notation that are suitable for expanding R-RANSAC to work with Lie Groups, and incorporating other improvements we have made to the general algorithm.

A. Terminology

R-RANSAC has very specific terminology that we that will facilitate the discussion.

- **Phenomenon**: Something that produces an observable signal. In the case of target tracking, the phenomenon is referred to as a **target**, which is an object that exists in physical space.
- **Measurement Source**: A sensor equipped with an algorithm that captures information from the environment and produces meaningful measurements used to observe the target.
- **Surveillance Region**: The portion of the environment that is observable by the measurement sources. There is a local surveillance region (LSR) for each measurement source and a global surveillance region (GSR) that is a union of all the local surveillance regions.
- **Frame of reference**: Consists of an abstract coordinate system and the set of physical reference points that uniquely fix (locate and orient) the coordinate system and standardize measurements within that frame. We will often refer to a frame of reference as just **frame**.

- **Local Frame**: The frame that coincides with a local surveillance region.
- **Global Frame**: The frame that coincides with the global surveillance region. It is possible that the global frame is the same as a local frame.
- **Sensor Scan**: When a sensor observes its surveillance region and extracts meaningful data. For example, the sensor scan of a camera occurs when the camera produces a new image of its surveillance region.
- **False Measurement**: A measurement extracted from a sensor scan that does not correspond to a phenomenon of interest. For example, motion in a camera can generate false measurements due to parallax depending on the algorithm. Another example is just noisy sensors.
- **True Measurement**: A measurement extracted from a sensor scan that corresponds to a phenomenon of interest.
- **Model**: This is simply a model of the phenomenon. In regards to target tracking, a model is referred to as a **track**.
- **Model Hypothesis**: This is a hypothetical model of the phenomenon (i.e. a possible model) created by the RANSAC algorithm. A model hypothesis that meets certain criteria becomes a model. In regards to target tracking, a model hypothesis is referred to as a **track hypothesis**. We will often abbreviate the term and mention only **hypothesis**.
- **Model Likelihood**: The probability that a model represents an actual target.
- **Good Model**: A model that is deemed very likely to correctly describe a phenomenon, based on predefined criteria, becomes a good model. In regards to target tracking, a good model is referred to as a **good track**.
- **Poor Model**: A model that is not a good model. In regards to target tracking, a poor model is referred to as a **poor track**.
- **Time Window**: An interval of time extending into the past from the current time.
- **Expired Measurement**: A measurement that was observed in the past outside the time window.
- **Measurement Source**: An algorithm that takes sensor data and produces a measurement. We will often refer to a measurement source as just a source.

B. Parameters

There are various parameters used throughout R-RANSAC. A parameter can either be a scalar or a tuple. In case the context doesn't make it clear, we will specify which one is a scalar and a tuple.

RANSAC parameters

- ℓ - max iterations
- τ_E - If the size of a consensus set is greater than τ_E, than RANSAC will terminate early.
- τ_I - RANSAC inlier threshold
- τ_{RM} - The minimum size of a track hypothesis's consensus set in order for the hypothesis to become a track

Cluster parameters

- τ_{CD} - neighborhood distance threshold
- τ_{CM} - cluster minimum cardinality threshold

Model Management Parameters

- τ_S - Similarity merge threshold
- τ_ρ - Good track threshold
- τ_α- Missed detection threshold

PDAF

- P_D - Probability of detection
- P_G - Probability of being in the validation region
- λ - The spacial density of false measurements in a local surveillance region per sensor scan

Other

- T_W - Time window
- M - The number of models

APPENDIX E
HOMOGRAPHY

Let

$$H = \begin{bmatrix} H_1 & h_2 \\ h_3^\top & h_4 \end{bmatrix},$$

and $\rho^a \in \mathbb{R}^2$ denote a pixel location. Then

$$\rho^b \quad = \quad g\left(H, \rho^a\right)$$

where

$$g\left(H, \rho^a\right) = \frac{H_1 \rho^a + h_2}{h_3^\top \rho^a + h_4}.$$

$$G\left(H, \rho^a\right) = \frac{\left(h_3^\top \rho^a + h_4\right) H_1 - \left(H_1 \rho^a + h_2\right) h_3^\top}{\left(h_3^\top \rho^a + h_4\right)^2}$$

$$\frac{\partial g}{\partial \rho} = G\left(H, \rho\right)$$
$$\frac{\partial g}{\partial \dot{\rho}} = 0$$

$$M = \frac{\partial G\left(H, \rho^a\right) \dot{\rho}^a}{\partial \rho^a} = \frac{-H_1 \dot{\rho}^a h_3^\top}{\left(h_3^\top \rho^a + h_4\right)^2} - \frac{H_1 h_3^\top \dot{\rho}^a}{\left(h_3^\top \rho^a + h_4\right)^2} + \frac{2\left(H_1 \rho^a + h_2\right.}{\left(h_3^\top \rho^a + \right.}$$
$$= \frac{2\left(H_1 \rho^a + h_2\right) h_3^\top \dot{\rho}^a h_3^\top - H_1 \dot{\rho}^a h_3^\top \eta - H_1 h_3^\top \dot{\rho}^a \eta}{\eta^3}$$

which can be simplified to

$$M = -\frac{G\left(\dot{\rho}^a h_3^\top + I h_3^\top \dot{\rho}^a\right)}{\eta}$$

$$f\left(\rho^a, \dot{\rho}^a\right) = \begin{bmatrix} g\left(H, \rho^a\right) \\ G\left(H, \rho^a\right)\dot{\rho}^a \end{bmatrix}$$

$$\eta = \left(h_3^\top \rho^a + h_4\right)$$

Covariance is transformed as

$$T = \begin{bmatrix} G & 0 \\ M & G \end{bmatrix}$$

$$TCT^\top$$

This has been tested

For $SE\left(2\right)$ the rotation matrix can be constructed as

$$R^a = \frac{1}{\|\dot{\rho}^a\|}\begin{bmatrix} \dot{\rho}^a \cdot e_1 & -\dot{\rho}^a \cdot e_2 \\ \dot{\rho}^a \cdot e_2 & \dot{\rho}^a \cdot e_1 \end{bmatrix}.$$

To transform it using the Homography we get

$$R_t = \frac{1}{\|G\dot{\rho}^a\|}\begin{bmatrix} e_1^\top G\dot{\rho}^a & -e_2^\top G\dot{\rho}^a \\ e_2^\top G\dot{\rho}^a & e_1^\top G\dot{\rho}^a \end{bmatrix}$$

Since we assume that the UAV has constant velocity in the body frame, the acceleration in the inertial frame is due to angular rotation, that is,

$$\begin{aligned} \ddot{\rho}^i &= \left[\omega^i\right]_\times \dot{\rho}^i \\ \ddot{\rho}^i &= R_b^i \left[\omega^b\right]_\times \left(R_b^i\right)^\top \dot{\rho}^i \\ &= R_b^i \left[\omega^b\right]_\times \left(R_b^i\right)^\top R_b^i \dot{\rho}^b \\ &= R_b^i \left[\omega^b\right]_\times \dot{\rho}^b \end{aligned}$$

Recall that

$$\dot{\rho}^n = G\left(H, \rho^c\right)\dot{\rho}^c,$$

taking the derivative w.r.t. to time, holding H constant and assuming constant velocity in the body frame we get that

$$\begin{aligned} \ddot{\rho}^n &= \left(\frac{\partial}{\partial \rho^c} G\left(H, \rho^c\right)\dot{\rho}^c\right)\frac{\partial \rho^c}{\partial t} + G\left(H, \rho^c\right)\frac{\partial \dot{\rho}^c}{\partial t} \\ \left[\omega^n\right]_\times \dot{\rho}^n &= M\dot{\rho}^c + G\left(H, \rho^c\right)\left[\omega^c\right]_\times \dot{\rho}^c \\ \left[\omega^n\right]_\times G\left(H, \rho^c\right)\dot{\rho}^c &= \left(M + G\left(H, \rho^c\right)\left[\omega^c\right]_\times\right)\dot{\rho}^c \\ \theta^n \left[1\right]_\times G\left(H, \rho^c\right)\dot{\rho}^c &= \left(M + G\left(H, \rho^c\right)\left[\omega^c\right]_\times\right)\dot{\rho}^c \\ \left(\dot{\rho}^c\right)^\top \left[\omega^n\right]_\times G\left(H, \rho^c\right)\dot{\rho}^c &= \left(\dot{\rho}^c\right)^\top \left(M + G\left(H, \rho^c\right)\left[\omega^c\right]_\times\right)\dot{\rho}^c \\ \left(\dot{\rho}^c\right)^\top \left[\omega^n\right]_\times &= \theta^n \left(\dot{\rho}^c\right)^\top \left[1\right]_\times \\ \theta^n &= \frac{\left(\dot{\rho}^c\right)^\top \left(M + G\left(H, \rho^c\right)\left[\omega^c\right]_\times\right)\dot{\rho}^c}{\left(\dot{\rho}^c\right)^\top \left[1\right]_\times G\left(H, \rho^c\right)\dot{\rho}^c} \end{aligned}$$

$$\left(\left[\omega^n\right]_\times G\left(H, \rho^c\right) - \left(M + G\left(H, \rho^c\right)\left[\omega^c\right]_\times\right)\right)\dot{\rho}^c = 0_{2\times 2}$$

implies that

$$\left[\omega^n\right]_\times = \left(M + G\left(H, \rho^c\right)\left[\omega^c\right]_\times\right) G\left(H, \rho^c\right)^{-1}$$

Note that this is only valid in the virtual image frame. In other words

$$H = R - \frac{tn^\top}{d}$$

where $R = R\left(\psi\right)$ and $n^\top = [0, 0, 1]$. Note that this transformation can be simplified to

$$\left[\omega^n\right]_\times = \left[\omega^c\right]_\times$$

A. Homography NVIP SE2

When working with a camera and tracking a target whose manifold configuration is $SE\left(2\right)$, tracking is done in the normalized virtual image frame (NVIF). In other words, the normalized virtual image plane is parallel to the ground and the virtual optical axis is pointing down to the Earth's core. Tracking has to be done in the NVIF, becuase when the UAS pitches or rolls, the circular trajectory of the target becomes an ellipse which doesn't follow the constraint that the target has constant velocity.

Tracking targets in the NVIF simplifies the homography to

$$H = \begin{bmatrix} R\left(\psi\right) & -\pi_{e_3} P/d \\ 0 & 1 - e_3^\top P/d \end{bmatrix}$$

where $R\left(\psi\right)$ is the rotation of the UAS due to yaw, P is the translation of the UAS, d is the distance the UAS is from the tracking plane and π_{e_3} is the projection onto the plane orthogonal to e_3. Using the decomposition of the homography

$$H = \begin{bmatrix} H_1 & h_2 \\ h_3^\top & h_4 \end{bmatrix},$$

the position, rotation and velocity transformation is

$$\begin{aligned} \dot{\rho}^n &= \frac{H_1 \dot{\rho}^c + h_2 \dot{\rho}^c}{h_4} \\ R^n &= H_1 R^c \\ \dot{\rho}^n &= \frac{\rho^c}{h_4} \\ \omega^n &= \omega^c. \end{aligned}$$

The covariance transformation is

$$C^n = TC^cT^\top$$

where C is the covariance and

$$T = \text{diag}\left(\frac{1}{h_4}, \frac{1}{h_4}, 1, \frac{1}{h_4}, \frac{1}{h_4}, 1\right).$$

The 5th entry is really zero since we assume that the body velocity in the left and right direction is zero.

APPENDIX F
PDAF WITH VARYING NOISE COVARIANCES

State update

$$x_k = x_{k^-} + \sum_i K_{k,i}\beta_i\nu_i$$

where ν_i is the innovation term associated with the i^{th} measurement and

$$K_{k,i} = \bar{P}_k H_k^\top S_{k,i}^{-1}$$

where

$$S_{k,i} = \left(H_k \bar{P}_k H_k^\top + V_k R_{k,i} V_k^\top\right)$$

and $R_{k,i}$ is the measurement noise associated with the i^{th} measurement.

The covariance update is

$$P_k = \bar{P}_k - \sum_i \beta_i K_{k,i} S_{k,i} K_{k,i}^\top + \sum_i \left(\beta_i K_{k,i}\nu_i\nu_i^\top - \nu\nu^\top K_{k,i}^\top\right)$$

where

$$\nu = \sum_i \beta_i \nu_i$$

APPENDIX G
LMLE REVISED

Depending on the method used, you may need compute the derivative of $y_m^j \ominus h\left(f^{-1}\left(x_k, w_k, \delta_{k:m}\right), v_m\right)$ with respect to the state. We denote this derivative as $H_{k:m}$ which is defined as

$$\begin{aligned}
H_{k:m} &= \left.\frac{\partial y_m \ominus h\left(f^{-1}, v_m\right)}{\partial x_k}\right|_{\hat{x}_k, \hat{w}_k, \hat{v}_m} \\
&= J_r^{-1}\left(\hat{e}_{k:m}\right) \mathbf{Ad}_{y_m^{-1}} \left.\frac{\partial h\left(f^{-1}, v_m\right)}{\partial x_k}\right|_{\hat{x}_k, \hat{w}_k, \hat{v}_m}.
\end{aligned}$$

$$H_m = J_r^{-1}\left(\hat{e}\right) \mathbf{Ad}_{y_m^{-1}} \left.\frac{\partial h\left(x_m, v_m\right)}{\partial x_m}\right|_{x_m=\hat{x}_m, V_m=0}$$

$$G_{k:m}^o = J_r^{-1}\left(e\right) Ad_y \frac{\partial h\left(f^{-1}, v_m\right)}{\partial f^{-1}} \frac{\partial f^{-1}}{\partial w}$$

$$\begin{aligned}
G_{k:m} &= \left.\frac{\partial x_m \ominus f^{-1}\left(x_k, w_m, \delta_{k:m}\right)}{\partial w_m}\right|_{x_m=\hat{x}_m, w_m=0} \\
&= \frac{\partial \log\left(e\right)}{\partial e} \frac{\partial e}{\partial\left(f^{-1}\right)^{-1}} \frac{\partial\left(f^{-1}\right)^{-1}}{\partial f^{-1}} \left.\frac{\partial f^{-1}}{\partial w_m}\right|_{x_m=\hat{x}_m, w_m=0} \\
&= J_r^{-1}\left(e\right) \mathrm{Ad}\left(x_m\right)\left(-\mathrm{Ad}\left(f^{-1}\right)\right) \left.\frac{\partial f^{-1}}{\partial w_m}\right|_{x_m=\hat{x}_m, w_m=0} \\
&= I \left.\frac{\partial f^{-1}}{\partial w_m}\right|_{x_m=\hat{x}_m, w_m=0},
\end{aligned}$$

$$\frac{\partial f^{-1}}{\partial w_m} = \begin{bmatrix} J_r\left(\delta_{k:m} u_k\right)\delta_{k:m} & J_r\left(\delta_{k:m} u_k\right)\frac{\delta_{k:m}^2}{2} \\ 0 & I\delta_{k:m} \end{bmatrix},$$

$$e = x_m \ominus f^{-1}\left(x_k, w_k, \delta_{k:m}\right),$$

$$\bar{R} + H_m^\top \bar{Q} H_m = \left(V_m R V_m^\top\right) + H_m G_{k:m} Q G_{k:m}^\top H_m^\top$$

REFERENCES

[1] Ralph Abraham, Jerrold Mardsen, and Tudor Ratiu. *Manifolds, Tensor Analysis, and Applications*. Springer-Verlag, New York, first edition, 1998.

[2] Yaakov Bar-Shalom, Fred Daum, and Jim Huang. The Probabilistic Data Association Filter: Estimation in the presence of measurement origin uncertainty. *IEEE Control Systems*, 29(6):82–100, 2009.

[3] Yaakov Bar-Shalom and Fortmaan Thomas E. *Tracking and Data Association*. Academin Press, Inc., 1988.

[4] Yaakov Bar-Shalom, Peter Willett, and Xin Tian. *Tracking And Data Fusion: A Handbook of Algorithms*. YBS Publishing, 2011.

[5] Timothy Barfoot. *State Estimation For Robotics*. Cambridge University Press, 2019.

[6] Randal W. Beard. Quadrotor Dynamics and Control. *Brigham Young University*, pages 1–47, 2008.

[7] Guillaume Bourmaud, Rémi Mégret, Marc Arnaudon, and Audrey Giremus. Continuous-Discrete Extended Kalman Filter on Matrix Lie Groups Using Concentrated Gaussian Distributions. *Journal of Mathematical Imaging and Vision*, 51(1):209–228, 2014.

[8] Francesco Bullo, Andrew D. Lewis, and Bill Goodwine. *Geometric Control of Mechanical Systems*, volume 50. Springer Science+Business Media, LCC, New York, 2005.

[9] Sunglok Choi, Taemin Kim, and Wonpil Yu. Performance evaluation of RANSAC family. *British Machine Vision Conference, BMVC 2009 - Proceedings*, (January), 2009.

[10] David F. Crousey, Marco Guerrieroy, Peter Willetty, Roy Streitz, and Darin Dunham. A look at the PMHT. *2009 12th International Conference on Information Fusion, FUSION 2009*, 4(2):332–339, 2009.

[11] M. Daszykowski and B. Walczak. Density-Based Clustering Methods. *Comprehensive Chemometrics*, 2:635–654, 2009.

[12] Patrick C Defranco, Randal W Beard, Karl F Warnick, and Timothy W Mclain. Detecting and Tracking Moving Objects from a Small Unmanned Air Vehicle. *All Theses and Dissertations*, (March), 2015.

[13] A. P. Dempster, N. M. Laird, and D. B. Rubin. *Maximum Likelihood from Incomplete Data Via the EM Algorithm* , volume 39. 1977.

[14] Zili Deng, Peng Zhang, Wenjuan Qi, Jinfang Liu, and Yuan Gao. Sequential covariance intersection fusion Kalman filter. *Information Sciences*, 189:293–309, 2012.

[15] Kenth Engø. Partitioned Runge-Kutta methods in Lie-group setting. *Reports in Informatics*, 43(202):21–39, 2000.

[16] Martin A Fischler and Robert C Bolles. RANSAC1981.pdf. *Graphics and Image Processing*, 24(6):381–395, 1981.

[17] Brian C. Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*. Springer-Verlag New York, Inc, 2003.

[18] Christoph Hertzberg, René Wagner, Udo Frese, and Lutz Schröder. Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds. *Information Fusion*, 2013.

[19] Kyle Ingersoll. *Vision Based Multiple Target Tracking Using Recursive RANSAC*. PhD thesis, Brigham Young University, 2015.

[20] Kyle Ingersoll, Peter C. Niedfeldt, and Randal W. Beard. Multiple target tracking and stationary object detection in video with Recursive-RANSAC and tracker-sensor feedback. *2015 International Conference on Unmanned Aircraft Systems, ICUAS 2015*, pages 1320–1329, 2015.

[21] Arieh Iserles, Hans Z. Munthe-kaas, Syvert P. Nørsett, and Antonella Zanna. Lie-group methods. *Acta Numerica*, 9(March 2015):215–365, 2000.

[22] T. Kurien. Issues in the Design of Practical Multitarget Tracking Algorithms. *Multitarget-multisensor track-ing: advanced applications*, pages 43–83, 1990.

[23] John M. Lee. *An Introduction to Smooth Manifolds*. Springer Science+Business Media, LCC, 2013.

[24] Parker C. Lusk and Randal W. Beard. Visual Multiple Target Tracking from a Descending Aerial Platform. *Proceedings of the American Control Conference*, 2018-June:5088–5093, 2018.

[25] Yi Ma, Stefano Soatto, Jana Košecká, and Sastru S. Shankar. *An Invitation to 3-D Vision From Images to Geometric Models*. Springer, 2010.

[26] Jeffrey Millard. *Multiple Target Tracking in Realistic Environments Using Recursive-RANSAC in a Data Fusion Framework*. PhD thesis, Brigham Young University, 2017.

[27] Todd K Moon and Wynn C. Stirling. *Mathematical Methods and Algorithms for Signal Processing*. Prentice-Hall, Uppder Saddle River, New Jersey, 2000.

[28] Hans Munthe-Kaas. Lie-Butcher theory for Runge-Kutta methods. *BIT Numerical Mathematics*, 35(4):572–587, 1995.

[29] Hans Munthe-Kaas. Runge-Kutta methods on Lie groups. *BIT Numerical Mathematics*, 38(1):92–111, 1998.

[30] Hans Munthe-Kaas. High order Runge-Kutta methods on manifolds. *Applied Numerical Mathematics*, 29(1):115–127, 1999.

[31] Peter C. Niedfeldt. Recursive-RANSAC: A Novel Algorithm for Tracking Multiple Targets in Clutter. *All Theses and Dissertations*, (July):Paper 4195, 2014.

[32] Peter C. Niedfeldt and Randal W. Beard. *Recursive RANSAC: Multiple signal estimation with outliers*, volume 9. IFAC, 2013.

[33] Peter C. Niedfeldt and Randal W. Beard. Multiple Target Tracking using Recursive RANSAC. *Proceedings of the American Control Conference*, pages 3393–3398, 2014.

[34] Peter C. Niedfeldt and Randal W. Beard. Robust estimation with faulty measurements using recursive-RANSAC. *Proceedings of the IEEE Conference on Decision and Control*, 2015-Febru(February):4160–4165, 2014.

[35] Peter C. Niedfeldt and Randal W. Beard. Convergence and Complexity Analysis of Recursive-RANSAC: A New Multiple Target Tracking Algorithm. *IEEE Transactions on Automatic Control*, 61(2):456–461, 2016.

[36] Peter C. Niedfeldt, Kyle Ingersoll, and Randal W. Beard. Comparison and Analysis of Recursive-RANSAC for Multiple Target Tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 53(1):461–476, 2017.

[37] G.W. Pulford. Taxonomy of Multiple Target Tracking Methods. *IEE Proceedings-Radar, Sonar and Navigation*, 152(4):291–304, 2005.

[38] Donald B. Reid. An Algorithm for Tracking Multiple Targets. *IEEE Transactions on Automatic Control*, 24(6), 1979.

[39] Ian Reid. Estimation I. Technical report, 2001.

[40] Ian Reid. Estimation II 1 Discrete-time Kalman filter. Technical report, 2001.

[41] A. M. Sjøberg and O. Egeland. An EKF for Lie groups with application to crane load dynamics. *Modeling, Identification and Control*, 40(2):109–124, 2019.

[42] Joan Solà, Jeremie Deray, and Dinesh Atchuthan. A micro Lie theory for state estimation in robotics. pages 1–17, 2018.

[43] John Stillwell. *Naive Lie Theory*. Springer Science+Business Media, LCC, 2008.

[44] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. Massachusetts Institue of Technology, Cambridge, 2006.

[45] Xin Tian, Yaakov Bar-Shalom, and Genshe Chen. A no-loss covariance intersection algorithm for track-to-track fusion. *Signal and Data Processing of Small Targets 2010*, 7698(April 2010):76980S, 2010.

[46] Jacob White. *Real-Time Visual Multi-Target Tracking*. PhD thesis, Brigham Young University, 2019.

[47] Jared Kevin Wikle. Integration of a Complete Detect and Avoid System for Small Unmanned Aircraft Systems. *All Theses and Dissertations*, 2017.

[48] Feng Yang, Weikang Tang, and Hua Lan. A density-based recursive RANSAC algorithm for unmanned aerial vehicle multi-target tracking in dense clutter. *IEEE International Conference on Control and Automation, ICCA*, (k 1):23–27, 2017.

[49] Feng Yang, Weikang Tang, and Yan Liang. A novel track initialization algorithm based on random sample consensus in dense clutter. *International Journal of Advanced Robotic Systems*, 15(6):1–11, 2018.