Lie-Theoretic Multi-Robot Localization

Xiao Li   Gregory S. Chirikjian

Abstract This chapter presents a new distributed cooperative localization technique based on a second order sensor fusion method developed for the special Euclidean group. Uncertainties in the robot pose, sensor measurements and landmark positions (neighboring robots in this case) are modeled as Gaussian distributions in exponential coordinates. This proves to be a better fit for both the prior and posterior distributions resulting from the motion of nonholonomic kinematic systems with stochastic noise (as compared to standard Gaussians in Cartesian coordinates). We provide a recursive closed-form solution to the multi-sensor fusion problem that can be used to incorporate a large number of sensor measurements into the localization routine and can be implemented in real time. The technique can be used for nonlinear sensor models without the need for further simplifications given that the required relative pose and orientation information can be provided, and it is scalable in that the computational complexity does not increase with the size of the robot team and increases linearly with the number of measurements taken from nearby robots. The proposed approach is validated with simulation in Matlab.

1 Introduction

In recent years the field of Robotics and Automation has undergone a dramatic ascendency in terms of its significance in industrial and military applications as well as its growing importance in service applications. Multi-robot systems (also known as multi-agent systems) is a branch of robotics that deals with the collaboration

Xiao Li

Laboratory for Computational Sensing and Robotics (LCSR), Johns Hopkins University e-mail: lixiao89@hotmail.com

Gregory S. Chirikjian

LCSR and Department of Mechanical Engineering, Johns Hopkins University e-mail: gchirik1@jhu.edu

among teams of robots (either homogenous or heterogenous) in accomplishing certain tasks. This section reviews some of the work done in the field of multi-robot localization followed by an overview of the remainder of this chapter.

1.1 Introduction To Multi-Robot Localization

The path to true autonomy starts with robots knowing where they are in a given workspace. Such a problem is known as robot localization. According to [15], the localization problem can be categorized into two subproblems: (1) position tracking (local localization) which aims to compensate for small dead reckoning errors using sensor feedback, this approach is local in that the initial pose is assumed known; and (2) global localization in which the robot "figures out" its position given no knowledge of its initial pose. A tremendous amount of effort has been devoted to effectively and efficiently solving the localization problem and the field has seen major advancements in the establishment of highly practical and easy to implement algorithms with the EKF (Extended Kalman Filter) and PF (Particle Filter) based approaches the most widely accepted solutions to the problem. However, the majority of existing approaches are tailored to localizing a single robot. The field of multi-robot localization remains relatively fresh and to be explored [6].

Performing the localization task with multiple robots possesses the advantage of information sharing. Robots within a team can exchange information with other members so to increase the accuracy and reduce uncertainty in their own estimates. This advantage is shown both in simulation and experimentally in [6] in which two robots explore an indoor environment executing their own single robot localization scheme when they are far away from each other. And the proposed collaborative fusion algorithm is used when the two robots come into each other's detection range. Results show that such an algorithmic reinforcement has the effect of significantly reducing the ambiguities existing in the original estimates. A collaborative architecture of this sort can effectively reduce the hardware cost of the entire team in that if at least one robot has a good knowledge of its location, then other team members can use this information along with relative measurements to infer their own position and reduce estimation errors.

1.2 Comparison of Existing Distributed Localization Methods

The problem of cooperative localization has been tackled with a wide variety of approaches over the years. And similar to single robot localization, many of the existing algorithms can be considered variations of two main categories. The first family of algorithms make use of recursive Gaussian filters. Distributed versions of the Kalman Filter are proposed in [1] [14] to solve the cooperative localization problem. The Extended Kalman Filter (EKF) is utilized in [12] while also provid-

ing analytical expressions for the upper bound of the estimate uncertainty. In [2] the EKF is also used, but the algorithm is reinforced with an entropic criterion to select optimal measurements that reduce global uncertainty. The advantage of using recursive Bayesian filters to fuse information lies in they are incremental in nature, which makes them applicable to real time estimation. Closed-form expressions for state estimation and update also facilitate computational speed. However these types of algorithms deal only with Gaussian noise which may not be the case for some real systems. And EKF linearizes the system dynamics around the state of estimate which is prone to failure when errors grow.

The second family of algorithms are built upon sampling-based nonparametric filters. Monte Carlo Localization methods are used in [9] to estimate the pose of each member robot while using grid cells to describe the entire particle set. A global collaborative localization algorithm is presented in [6] that also builds upon sample-based Markov localization. In addition, [11][8][3] have all approached the problem with different variations of the Particle Filter and have also applied their algorithm in the SLAM (Simultaneous Localization and Mapping) context. Further experimental validation is provided in [11] and [3]. Grid-based and sampling-based Markov localization techniques usually address the problem globally and can be improved via carefully designed resampling processes to counteract localization failures. They can also be used to accommodate non-Gaussian noise models. However like all sampling-based approaches, a large number of grids/samples are usually needed to acquire reasonable outcomes, and the computational cost grows dramatically with the dimension of the problem. A table comparing the two families of methods is provided below.

These two main categories of localization techniques presented in Table 1 currently dominate the field. Both possesses their own pros and cons and the choice of which depend heavily on the type of applications they are desired for. The two approaches can potentially be combined to yield superior outcomes. For more details regarding the Extend Kalman Filter (EKF) applied to multi-robot systems see [14][12]. For details on Collaborative Monte Carlo Localization (MCL) see [6].

The following subsection explains how our approach differs.

1.3 Objectives, Contributions, and Outline

Existing approaches to the multi-robot localization problem usually consider only uncertainties in each robot's pose estimate and sensor measurement. The goal of this chapter is to explore cooperative localization in a more generalized setting where uncertainties in the sources of relative measurements (neighboring robots' pose estimates) are also considered. The distributed localization approach proposed in this chapter makes an effort to providing recursive closed-form expressions for real time cooperative sensor fusion used for pose updates of robots within a team. This work extends the method presented in [10], which considers cooperative localization with only one exact noise-free measurement (relative to a neighboring

Table 1: Comparison Between Distributed EKF And MCL

	Distributed EKF	**Multi-Robot MCL**
Restrictions on Error Distribution	Requires Gaussian process and measurement error	Nonparametric particle representation of posterior belief, no assumptions on noise distribution
Global Localization	No	Yes
State Recovery	No	Possible given a well designed resampling process
Localization Accuarcy	Accurate when error is small	Depends on the number of particles used
Computational Cost	Small due to the closed form propagation and update equations	Depends on the number of particles. Can increase dramatically with the dimension of the state space
Ease of Implementation	Simple	Can be involved
Robustness	Prone to error due to linearization	Quite resistant to errors given multiple beliefs are maintained simultaneously
Process Multiple Detections Simultaneously	No	No
Complexity Relative To Team Size	Fully distributed. Complexity independent of team size	Complexity independent of team size

robot), whereas the technique proposed here can take into account any number of relative measurements while also considering sensor noise. This method is developed under the framework of exponential coordinates for Lie groups which gives this exotic sounding methodology a down-to-earth benefit: Gaussian distribution in Cartesian coordinates, (x, y, θ) - planar coordinates and heading angle, possesses elliptical probability density contours for each fixed θ and for marginal densities in (x, y), whereas the banana-shaped distribution resulting from incremental motions of a stochastic differential-kinematic system (i.e., a probabilistic model of mobile robots with nonholonomic kinematic constraints) is better represented by a Gaussian in exponential coordinates which produce a more conformable density contour (see Figure 2a). This underlying framework allows the proposed algorithm to tolerate higher errors without worrying about collapse of the normality assumption as uncertainty grows. Unlike most existing cooperative localization schemes that consider only uncertainty in the pose of the robot to be estimated and measurement noise, the presented method has also taken into account the uncertainty in the pose of nearby robots from which relative measurements are taken, making it a more realistic and dynamical localization technique. This approach is second order in its expansion of the Gaussians that describes the pose and measurement distributions using the *Baker-Campbell-Hausdorff (BCH) formula* [4], and no simplifications are made regarding the system kinematics, thus preserving the full nonlinear characteristics of the original system. Lastly, the form of sensor measurement in this method

is kept generic without assuming the type of sensor or any underlying characteristics given the Gaussian-in-exponential-coordinate model can be applied.

The remainder of this chapter is outlined as follows. **Section 2** introduces the mathematical foundation on which the proposed approach is based, namely the basics of Matrix Lie Groups and Exponential Mapping. **Section 3** provides a detailed derivation of the proposed technique. **Section 4** describes the experimental setup in simulation and provides a discussion of the results. And **Section 5** concludes this chapter.

2 Mathematical Background for The Group $SE(n)$ and Exponential Mapping

2.1 The Special Euclidean Group and Exponential Coordinates

2.1.1 The Special Euclidean Motion Group

The proposed technique is largely based on the notion of Lie groups and their parameterizations. According to [4], a *group* is defined as a pair $(G, \circ)$ consisting of a set G and a binary operator $\circ$ such that $g_1 \circ g_2 \in G$ whenever $g_1, g_2 \in G$, the operator is associative in the sense that $(g_1 \circ g_2) \circ g_3 = g_1 \circ (g_2 \circ g_3)$ for all elements $g_1, g_2, g_3 \in G$, there exists an identity element $e \in G$ such that for all elements $g \in G$, $g \circ e = e \circ g = g$, and for each $g \in G$ there exists an inverse element $g^{-1} \in G$ such that $g \circ g^{-1} = g^{-1} \circ g = e$. For engineering applications, a group of great interest is the Special Euclidean Group, $SE(n)$, that describes rigid-body motions in n-dimensional Euclidean space. The elements of $SE(n)$ can be represented as $(n+1) \times (n+1)$ homogeneous transformation matrices of the form

$$SE(n) = \left\{ \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \middle| R \in SO(n), \mathbf{t} \in \Re^n \right\} \tag{1}$$

where $SO(n)$ is the Special Orthogonal group consisting of $n \times n$ rotation matrices and $\Re^n$ is the n dimensional vector space representing translations. The binary operation in this context is simply the matrix multiplication. The Special Euclidean Group is also a *Matrix Lie Group* since each element is a real-valued matrix, the whole set is a differentiable manifold, and both the operations of multiplication and inversion of homogeneous transformation matrices are smooth operations. We note that in most practical problems n takes only two values: $n = 2$ for planar motion and $n = 3$ for 3 dimensional space motion.

Now we introduce the concept of *Lie Algebra*. Again following [4], elements of a matrix Lie group can be written as $g = \exp(X)$ for $X \in \mathscr{G}$ where the set $\mathscr{G}$ is the matrix Lie Algebra of G. The Lie Algebra for $SE(2)$ (denoted as $se(2)$) can be represented by the linear combination of a set of basis

$$E_1^{se(2)} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad E_2^{se(2)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \qquad E_3^{se(2)} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

The exponential coordinates for an element of $SE(2)$ can be defined as $\mathbf{x}^{se(2)} = [v_1, v_2, \alpha]^T$ and under this definition an element of the the Lie algebra $se(2)$ can be written as a 3×3 matrix

$$X^{se(2)} = \begin{bmatrix} 0 & -\alpha & v_1 \\ \alpha & 0 & v_2 \\ 0 & 0 & 0 \end{bmatrix} = \sum_{i=1}^{3} E_i^{se(2)} x_i^{se(2)}. \tag{2}$$

The 'hat' and 'vee' notation is convenient to identify an element of $se(2)$ with a vector in $\mathfrak{R}^3$ as follows:

$$\hat{\mathbf{x}}^{se(2)} = X^{se(2)}, \text{ and } X^{se(2)\vee} = \mathbf{x}^{se(2)}.$$

The exponential map $\exp : se(2) \to SE(2)$ is surjective, but is not injective since $\alpha = \pi$ and $-\pi$ correspond to the same group element of $SE(2)$. But by removing from $SE(2)$ the set of measure zero, M, corresponding to $\alpha = \pi$, it is possible to define an inverse map $\log : (SE(2) - M) \to se(2)$. Since the integral of well-behaved functions over $SE(2)$ and $SE(2) - M$ are the same, we do not distinguish between $SE(2)$ and $SE(2) - M$ in the remainder of this chapter.
For $SE(2)$, exponentiation gives

$$R = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \text{ and } \mathbf{t} = \begin{bmatrix} [v_2(\cos(\alpha) - 1) + v_1 \sin(\alpha)]/\alpha \\ [v_1(1 - \cos(\alpha)) + v_2 \sin(\alpha)]/\alpha \end{bmatrix}. \tag{3}$$

2.1.2 Adjoint Matrices

The *adjoint operator* $Ad(g)$ and $ad(X)$ are two important concepts in the derivations that follow, and so their definitions as well as relevant properties are introduced in this section. To define the adjoints, we need to first define the inner product and Lie bracket operations for Lie algebras. According to [4], an *inner product* between arbitrary elements of the Lie algebra $Y = \sum_i y_i E_i$ and $Z = \sum_i z_i E_i$ can be defined such that

$$(Y, Z) = \sum_{i}^{d} y_i z_i \tag{4}$$

where d is the dimension of G. In particular, for $G = SE(n)$ the dimension is $d = n(n+1)/2$. Choosing a basis $\{E_i\}$ and requiring that $(E_i, E_j) = \delta_{ij}$, where δ_{ij} is the Dirac delta function, defines an inner product for $\mathscr{G}$ and a metric for G.

The *Lie Bracket* of $Y, Z \in \mathscr{G}$ is defined as

$$[Y, Z] \doteq YZ - ZY \tag{5}$$

With the above definitions in place, and any $g \in G$, the adjoint operators are

$$
Ad(g)X \doteq \frac{d}{dt}(g \circ \exp(tX) \circ g^{-1})|_{t=0} = \frac{d}{dt}\exp(tgXg^{-1})|_{t=0} = gXg^{-1}
$$
$$
ad(X)Y \doteq \frac{d}{dt}(Ad(e^{tX})Y)\Big|_{t=0}
$$

(6)

Since the adjoint operators are both linear operators, they can both be written as matrices that represent linear mapping. We call these matrices *adjoint matrices* and define them as (in component form) [4]

$$
[Ad(g)]_{ij} = (E_i, Ad(g)E_j) = (E_i, gE_jg^{-1}), \quad [ad(X)]_{ij} = (E_i, ad(X)E_j) = (E_i, [X, E_j])
$$

(7)

And written in terms of columns, the matrices have the form

$$
[Ad(g)] = [(gE_1g^{-1})^{\vee}, ..., (gE_ng^{-1})^{\vee}], \quad [ad(X)] = [[X, E_1]^{\vee}, ..., [X, E_n]^{\vee}]
$$
(8)

Some important properties of the adjoint matrices that are used in the following calculations are listed as follow:

1. $Ad(\exp(X)) = \exp(ad(X))$, $ad(X)X^{\vee} = 0$, $ad(X)Y = XY - YX = [X, Y]$
2. $ad(X)Y^{\vee} = [X, Y]^{\vee}$, $ad([X, Y]) = ad(X)ad(Y) - ad(Y)ad(X)$, $ad(X)Y^{\vee} = -ad(Y)X^{\vee}$
3. $Ad(g_1)Ad(g_2)X = g_1(g_2Xg_2^{-1})g_1^{-1} = (g_1g_2)X(g_1g_2)^{-1} = Ad(g_1g_2)X$
4. $\log^{\vee}(g \circ e^X \circ g^{-1}) = Ad(g)\log^{\vee}(e^X)$

For $SE(2)$, the explicit form of the adjoint matrices are

$$
Ad(g) = \begin{bmatrix} R & M\mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \in \Re^{3 \times 3}, \quad ad(g) = \begin{bmatrix} -\alpha M & M\mathbf{v} \\ \mathbf{0}^T & 0 \end{bmatrix} \in \Re^{3 \times 3}
$$

(9)

where $M = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, R and $\mathbf{t}$ defined by equation 1. $(\mathbf{v}, \alpha) = (v_1, v_2, \alpha)$ are the exponential coordinates of $SE(2)$.

2.1.3 The Baker-Campbell-Hausdorff Formula

The *Baker-Campbell-Hausdorf (BCH) formula* [4] serves as the core of the second order estimation of Gaussian convolutions (described in more detail in the next section). In essence, the BCH formula establishes a relationship between the Lie Bracket (defined in equation 5) and the matrix exponential. Let $X, Y \in \mathscr{G}$ and define $Z(X, Y) = \log(e^X e^Y)$, the BCH formula then takes the form

$$Z(X,Y) = X + Y + \frac{1}{2}[X,Y] + \frac{1}{12}([X,[X,Y]] + [Y,[Y,X]]) + \frac{1}{48}([Y,[X,[Y,X]]] + [X,[Y,[Y,X]]]) + \cdots$$
$$(10)$$

This can be verified by expanding e^X, e^Y using matrix exponential Taylor series $e^X = \sum_{k=0}^{\infty} \frac{X^k}{k!}$ and substitute into the Taylor series for matrix logarithm

$$\log(e^X e^Y) = \log(I + (e^X e^Y - I)) = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{(e^X e^Y - I)^k}{k} \qquad (11)$$

Applying the $\vee$ operator to (10) results in

$$\begin{aligned} \mathbf{z} = &\mathbf{x} + \mathbf{y} + \frac{1}{2} ad(X)\mathbf{y} + \frac{1}{12}(ad(X)ad(X)\mathbf{y} + ad(Y)ad(Y)\mathbf{x}) \\ &+ \frac{1}{48}(ad(Y)ad(X)ad(Y)\mathbf{x} + ad(X)ad(Y)ad(Y)\mathbf{x}) + \cdots \end{aligned} \qquad (12)$$

Equations (11) and (12) will be used extensively.

2.2 Gaussians on $SE(n)$ and Second Order Convolution Theory

A Gaussian on the $SE(n)$ is defined as

$$f(g; \mu, \Sigma) \doteq \frac{1}{C(\Sigma)} \exp \left\{ -\frac{1}{2} [\log^\vee(\mu^{-1}g)]^T \Sigma^{-1} [\log^\vee(\mu^{-1}g)] \right\} \qquad (13)$$

where $\mu, g \in SE(n)$, $C(\Sigma) \approx (2\pi)^{\frac{d}{2}} \|\det(\Sigma)\|^{\frac{1}{2}}$ is the normalizing factor when $\|\Sigma\|$ is small.

For a domain of integration $G = SE(n)$, the mean of the above Gaussian is defined by the value $\mu \in G$ for which

$$\int_G \log^\vee(\mu^{-1}g) f(g) \, \mathrm{d}g = \mathbf{0} \qquad (14)$$

and the covariance is given by

$$\Sigma \doteq \int_G [\log^\vee(\mu^{-1}g)][\log^\vee(\mu^{-1}g)]^T f(g) \, \mathrm{d}g \qquad (15)$$

For details on how to integrate on $SE(n)$, please refer to [15] [16]. Given two Gaussians $f_1(g) = f(g; \mu_1, \Sigma_1)$ and $f_2(g) = f(g; \mu_2, \Sigma_2)$ in the form of (13), their convolution is defined as

$$(f_1 * f_2)(g) = \int_G f_1(h) f_2(h^{-1}g) \, \mathrm{d}h = \int_G \rho_1(\mu_1^{-1}h) \rho_2(\mu_2^{-1}h^{-1}g) \, \mathrm{d}h$$

with $\rho_i(g) = f(g; e, \Sigma_i)$ denoting a Gaussian centered at the identity. It is proven (refer to [16]) that the convolution $(f_1 * f_2)(g)$ results (to the second order) in a Gaussian with mean and covariance

$$\mu_{1*2} = \mu_1 \mu_2, \quad \Sigma_{1*2} = A + B + F(A, B) \tag{16}$$

with the terms A, B and F defined by

$$A = Ad(\mu_2^{-1})\Sigma_1 Ad(\mu_2^{-1})^T \text{ and } B = \Sigma_2 \tag{17}$$

where

$$F(A, B) = \frac{1}{4} \sum_{i,j=1}^{d} ad(E_i) \ B \ ad(E_j)^T A_{ij} + \frac{1}{12} \left\{ [\sum_{i,j=1}^{d} A''_{ij}]B + B^T [\sum_{i,j=1}^{d} A''_{ij}]^T \right\}$$
$$+ \frac{1}{12} \left\{ [\sum_{i,j=1}^{d} B''_{ij}]A + A^T [\sum_{i,j=1}^{d} B''_{ij}]^T \right\} \tag{18}$$

and

$$A''_{ij} = ad(E_i)ad(E_j)A_{ij}, \quad B''_{ij} = ad(E_i)ad(E_j)B_{ij}. \tag{19}$$

The above results will be used for $SE(2)$ in the next section, where the basis elements E_i as well as Ad and ad matrices as defined in the previous section.

3 Derivation of Second Order Bayesian Sensor Fusion on the SE Group

This section presents a detailed derivation of the proposed technique. Again the technique focuses on fusing the relative measurements of neighboring robots and their pose information to reduce the estimation uncertainty of the current robot. A probabilistic approach is adapted where uncertainties in the robot positions and sensor measurements are modeled by Gaussians (refer to [5] for a more generalized formulation of non-linear measurement model approximation on Lie groups). In addition, since the motion of a stochastic system with differential constraints is modeled more accurately with Gaussians in exponential coordinates than that in Cartesian coordinates, the proposed technique is developed under exponential coordinates. The theory will first be developed for a system of two robots (which builds on [10] by taking sensor noise into consideration) and be extended to the multi-robot scenario.

3.1 Localization for A Robot Pair

The problem is given by two mobile robots i and j moving in the plane whose priors in position and orientation are Gaussians $f(a_i^{-1}g_i; \mu_i, \Sigma_i)$ and $f(a_j^{-1}g_j; \mu_j, \Sigma_j)$. Here $a_i, a_j \in SE(2)$ are the known initial positions of the robots relative to the world frame at $t = 0$. At time t, $\mu_i, \mu_j \in SE(2)$ and $\Sigma_i, \Sigma_j \in R^{3 \times 3}$ are the means (defined relative to the initial frames a_i, a_j), and covariances obtained from a previous prediction step which we'll also assume to be known. In addition, a sensor measurement of robot j relative to i is also obtained at time t and is given by the homogeneous matrix $m_{ij} \in SE(2)$. Since we assume the sensor has Gaussian noise, its distribution is then characterized by a Gaussian of the form $M_{ij}(g_i, g_j) = f(g_j; g_i m_{ij}, \Sigma_m)$ which says that according to the sensor, the position of robot j with respective to robot i has a mean of m_{ij} and covariance of Σ_m.

The goal is then to calculate a posterior for the position of robot i using the sensor measurement to update the its prior. Because the sensor provides a relative measurement, we first formulate the joint prior of robot i and j making the assumption that the priors are independent of each other, giving

$$p_{ij}(g_i, g_j) = f(a_i^{-1}g_i; \mu_i; \Sigma_i)f(a_j^{-1}g_j; \mu_j; \Sigma_j). \tag{20}$$

Then according to Bayes' Theorem, the joint posterior is given by

$$\overline{p}_{ij} = \eta_1 p_{ij} M_{ij} \tag{21}$$

where η_1 is a constant normalizing factor. Similar normalizing factors result in all fusion processes that follow and will be denoted by η_i. To save space in the derivations, we will denote $\rho_i(\mu_i^{-1}g_i) = f(g_i; \mu_i; \Sigma_i)$ and the rest follows where $\rho_i(g)$ is a Gaussian with mean at the identity. The marginal distribution of the joint posterior for robot i is then

$$\overline{p}_i(g_i) = f(g_i; \overline{\mu}_i, \overline{\Sigma}_i) = \eta_2 \int_G p_{ij}(g_i, g_j) M_{ij}(g_i, g_j) \, \mathrm{d}g_j$$
$$= \eta_2 \rho_i(\mu_i^{-1} a_i^{-1} g_i) \int_G \rho_j(\mu_j^{-1} a_j^{-1} g_j) \rho_m(m_{ij}^{-1} g_i^{-1} g_j) \, \mathrm{d}g_j \tag{22}$$

The goal is to find closed-form expressions for $\overline{\mu}_i$ and $\overline{\Sigma}_i$. Since ρ_m is symmetric around the mean, we have $\rho_m(m_{ij}^{-1} g_i^{-1} g_j) = \rho_m(g_j^{-1} g_i m_{ij})$. Letting $g' = g_i m_{ij}$, equation 22 becomes

$$\overline{p}_i(g_i) = \eta_2 \rho_i(\mu_i^{-1} a_i^{-1} g_i) \int_G \rho_j(\mu_j^{-1} a_j^{-1} g_j) \rho_m(g_j^{-1} g_i m_{ij}) \, \mathrm{d}g_j$$
$$= \eta_2 \rho_i(\mu_i^{-1} a_i^{-1} g_i) \int_G \rho_j(\mu_j^{-1} a_j^{-1} g_j) \rho_m(e^{-1} g_j^{-1} g') \, \mathrm{d}g_j \tag{23}$$

where $e \in SE(2)$ is the identity element of SE(2). According to the definition of convolution in Section 2, the integral in equation 23 defines a convolution $(f_1 * f_2)(g')$

where $f_1(g') = f(g'; a_j \mu_j, \Sigma_j)$ and $f_2(g') = f(g'; e, \Sigma_m)$. Let $f_{1*2}(g'; \mu_{1*2}, \Sigma_{1*2}) = (f_1 * f_2)(g')$, then (16)-(19) can be used to calculate the closed-form expressions of μ_{1*2} (which equals to $a_j \mu_j$) and Σ_{1*2}. With the integral taken care of, (23) becomes

$$\overline{p_i}(g_i) = f(g_i; \overline{\mu}_i, \overline{\Sigma}_i) = \eta_2 f(\mu_i^{-1} a_i^{-1} g_i; e, \Sigma_i) f(g_i m_{ij}; a_j \mu_j, \Sigma_{1*2}) \qquad (24)$$

For a posterior of robot i formulated in the form of (24), the fusion technique developed in [10] can be used to derive the closed-form expressions for $\overline{\mu}_i$ and $\overline{\Sigma}_i$.

3.2 Localization for Multi-Robot Team

Now we are ready to extend the technique to multi-robot localization. Similar to the previous subsection, the posterior of robot i is what we are trying to estimate, but instead of taking measurement from a single neighboring robot, multiple measurements are taken from however many neighboring robots that are in the sensing range (for derivation purposes we label the neighboring robots as $1, 2, ..., n$). Following a similar approach we have the joint prior

$$\begin{aligned} p_{i,1,...,n} &= f(a_i^{-1} g_i; \mu_i; \Sigma_i) f(a_1^{-1} g_1; \mu_1; \Sigma_1) ... f(a_n^{-1} g_n; \mu_n; \Sigma_n) \\ &= \rho_i(\mu_i^{-1} a_i^{-1} g_i) \rho_1(\mu_1^{-1} a_1^{-1} g_1) ... \rho_n(\mu_n^{-1} a_n^{-1} g_n) \end{aligned} \qquad (25)$$

Let $M_{in} = f(g_n; g_i m_{in}, \Sigma_{in})$ be the distribution of the sensor measurement of robot n relative to robot i and assume independence among all the measurements. Then we have the joint measurement distribution

$$M_{i,1,...,n} = M_{i1} M_{i2} ... M_{in} \qquad (26)$$

To further save space, we will write in short $\rho_i = \rho_i(\mu_i^{-1} a_i^{-1} g_i)$ as the position priors and $\rho_{in} = \rho_{in}(m_{in}^{-1} g_i^{-1} g_n) = M_{in}$ as the measurement distributions. We will further define $g'_{in} = g_i m_{in}$. The posterior for robot i is then

$$\begin{aligned} \overline{p_i}(g_i) = f(g_i; \overline{\mu}_i, \overline{\Sigma}_i) &= \eta_3 \int_G \int_G ... \int_G p_{i,1,...,n} M_{i,1,...,n} \, dg_1 \, dg_2 ... dg_n \\ &= \eta_3 \rho_i \left(\int_G \rho_1 \rho_{i1} \, dg_1 \right) \left(\int_G \rho_2 \rho_{i2} \, dg_2 \right) ... \left(\int_G \rho_n \rho_{in} \, dg_n \right) \end{aligned} \qquad (27)$$

Let $f_n(g'_{in}) = f(g'_{in}; a_n \mu_n, \Sigma_n)$ and $f_{in}(g'_{in}) = f(g'_{in}; e, \Sigma_{in})$, then (27) becomes

$$\overline{p_i}(g_i) = f(g_i; \overline{\mu}_i, \overline{\Sigma}_i) = \eta_3 \rho_i(\mu_i^{-1} a_i^{-1} g_i)(f_1 * f_{i1})(g'_{i1})(f_2 * f_{i2})(g'_{i2}) ... (f_n * f_{in})(g'_{in}) \qquad (28)$$

Calculating the convolutions using (16)-(19), we finally arrive at

$$\boxed{\overline{p_i}(g_i) = f(g_i;\overline{\mu}_i,\overline{\Sigma}_i) = \eta_3 f(\mu_i^{-1}a_i^{-1}g_i;e,\Sigma_i)f(g_i m_{i1};a_1\mu_1,\Sigma_{1*i1}) \times ... \times f(g_i m_{in};a_n\mu_n,\Sigma_{n*in})}$$
$$(29)$$

An extension of the method provided by [10] (which fuses only one measurement) gives the equations to calculate $\overline{\mu}_i$ and $\overline{\Sigma}_i$, and is presented as follows:

For neighboring robots $1,...,k,....,n$

1. Define $q_k = m_{ik}\mu_k^{-1}a_k^{-1}a_i\mu_i$, $\exp(\hat{x}_k) = q_k.$, $\Gamma_k = (I + \frac{1}{2}ad(\hat{x}_k))$, $S_i = \Gamma_i^T\Sigma_i^{-1}\Gamma_i$
2. Define $S_k = \Gamma_m^T Ad^{-T}(m_{ik})\Sigma_{k*ik}^{-1}Ad^{-1}(m_{ik})\Gamma_k$
3. $\overline{S}' = S_i + \sum_{k=1}^{n} S_k$, $\overline{x}' = \overline{S}'^{-1}\sum_{k=1}^{n} S_k x_k$

With the above definitions, the posterior distribution for robot i can be calculated by

$$\boxed{\begin{aligned}\overline{\Sigma}_i &= \quad \overline{\Gamma}'\overline{S}'^{-1}\overline{\Gamma}'^T \\ \overline{\mu}_i &= \quad \mu_i\exp(-\hat{\overline{x}}')\end{aligned}}$$
$$(30)$$

with the operator $\wedge$ and $\vee$ defined in section 2.

3.3 A Complete Distributed Localization Algorithm Using Bayesian Filter In Exponential Coordinates

The fusion technique introduced above defines the state update step for the proposed localization method. However like all Bayesian Filters a complete recursive filter for state estimation consists of a state prediction step as well as a state update step. This section serves to provide the proposed algorithm in such a form.

Similar to the above setting, suppose at time t_k robot i is the robot to be localized, robots $1,...,k,...,n$ are its n neighbors. Their means are $\mu_i(t_k),\mu_k(t_k)$ and covariances $\Sigma_i(t_k),\Sigma_k(t_k)$ respectively. Let the stochastic differential equation (SDE) governing the motion of the robots be of the form

$$(g^{-1}\dot{g})^{\vee}dt = \mathbf{h}dt + Hd\mathbf{w} \qquad (31)$$

where $\mathbf{h}$ is constant. When $g \approx e$ and for a sampling time Δt a constant command u is given to the system resulting in motion of the system from t_k to t_{k+1}, the distributed localization scheme that estimates the location of robot i at time t_{k+1} follows two steps (letting $\Delta t = t_{k+1} - t_k$). These are the prediction and update steps.

Prediction

$$
\begin{aligned}
&\mu_i(\Delta t) = \exp\left(\int_0^{\Delta t} \hat{h}_i \, d\tau\right) \\
&\Sigma_i(\Delta t) = \int_0^{\Delta t} Ad(\mu_i^{-1}(t-\tau)) H_i H_i^T Ad^T(\mu_i^{-1}(t-\tau)) \, d\tau \\
&\mu_i(t_{k+1}^-) = \mu_i(t_k) \circ \mu_i(\Delta t) \\
&\Sigma_i(t_{k+1}^-) = A_i(t_k) + B_i(t_k) + F(A_i(t_k), B_i(t_k))
\end{aligned}
\tag{32}
$$

where

$$
\begin{aligned}
A_i(t_k) &= Ad(\mu_i(\Delta t)^{-1})\Sigma_i(t_k^+)Ad(\mu_i(\Delta t)^{-1})^T, \quad B_i(t_k) = \Sigma_i(\Delta t) \\
A_i(t_k)_{ij}'' &= ad(E_i)ad(E_j)A_i(t_k)_{ij}, \quad B_i(t_k)_{ij}'' = ad(E_i)ad(E_j)B_i(t_k)_{ij}
\end{aligned}
\tag{33}
$$

$$
\begin{aligned}
F_i(A_i(t_k), B_i(t_k)) &= \frac{1}{4}\sum_{i,j=1}^{d} ad(E_i)B_i(t_k)ad(E_j)^T A_i(t_k)_{ij} \\
&+ \frac{1}{12}\left\{[\sum_{i,j=1}^{d} A_i(t_k)_{ij}'']B_i(t) + B_i(t)^T[\sum_{i,j=1}^{d} A_i(t)_{ij}'']^T\right\} \\
&+ \frac{1}{12}\left\{[\sum_{i,j=1}^{d} B_i(t_k)_{ij}'']A_i(t_k) + A_i(t_k)^T[\sum_{i,j=1}^{d} B_i(t_k)_{ij}'']^T\right\}
\end{aligned}
\tag{34}
$$

The second equation in 32 follows from equation (19) in [13] given by

$$
\Sigma_i(\Delta t) = \int_0^{\Delta t} Ad^{-1}[\mu_i^{-1}(\tau)\mu_i(t)]H_i H_i^T Ad^{-T}[\mu_i^{-1}(\tau)\mu_i(t)] \, d\tau
\tag{35}
$$

Since in our context the mean takes form of $\mu = \exp(Xt)$ where $X \in \mathscr{G}$ is a constant, it follows that $\mu_i^{-1}(t)\mu_i(\tau) = \mu_i(\tau)\mu_i^{-1}(t) = \mu_i^{-1}(t-\tau)$. Combined with the property of adjoint $Ad^{-1}(\mu) = Ad(\mu^{-1})$ gives the final expression in 32.

Also in the above equations, $\mu_i(\Delta t)$ and $\Sigma_i(\Delta t)$ define the incremental distribution resulting solely from the input given at the Δt time frame with location given with respective to $\mu_i(t_k)$, not the fixed world frame. In order to take into account the uncertainties already present at time t given by $\Sigma_i(t_k)$, the distribution at time t_k is convolved with the incremental distribution resulting in the predicted distribution given by $\mu_i(t_{k+1}^-), \Sigma_i(t_{k+1}^-)$.

Update

Now to incorporate the relative measurements, for each of the neighboring robots $1, ..., k, ..., n$, obtain the measurement distribution $m_{ik}(t), \Sigma_{ik}(t)$, then

$$
\begin{aligned}
A_{ik}(t_k) &= Ad(m_{ik}(t_k)^{-1})\Sigma_k(t_k)Ad(m_{ik}(t_k)^{-1})^T, \quad B_{ik}(t_k) = \Sigma_{ik}(t_k) \\
A_{ik}(t_k)_{ij}'' &= ad(E_i)ad(E_j)A_{ik}(t_k)_{ij}, \quad B_{ik}(t_k)_{ij}'' = ad(E_i)ad(E_j)B_{ik}(t_k)_{ij}
\end{aligned}
\tag{36}
$$

and using (18),

$$\Sigma_{k*ik}(t(t_k)) = A_{ik}(t_k) + B_{ik}(t_k) + F(A_{ik}(t_k), B_{ik}(t_k)) \tag{37}$$

1. Define $q_k(t_k) = m_{ik}(t_k)\mu_k(t_k)^{-1}a_k^{-1}a_i\mu_i(t_{k+1}^-)$, $\exp(\hat{x}_k(t_k)) = q_k(t_k)$.
2. Define $\Gamma_k(t_k) = (I + \frac{1}{2}ad(\hat{x}_k(t_k)))$, $S_i(t_k) = \Gamma_i(t_k)^T[\Sigma_i(t_{k+1}^-)]^{-1}\Gamma_i(t_k)$
3. Define $S_k(t_k) = \Gamma_m(t_k)^T Ad^{-T}(m_{ik}(t_k))\Sigma_{k*ik}(t_k)^{-1}Ad^{-1}(m_{ik}(t_k))\Gamma_k(t_k)$
4. $\bar{S}'(t_k) = S_i(t_k) + \sum_{k=1}^{n} S_k(t_k)$, $\bar{x}'(t_k) = \bar{S}'(t_k)^{-1} \sum_{k=1}^{n} S_k(t_k)x_k(t_k)$.

Then

$$\boxed{\Sigma_i(t_{k+1}^+) = \bar{\Gamma}'(t_k)\bar{S}'(t_k)^{-1}\bar{\Gamma}'(t)^{t_k}, \quad \mu_i(t_{k+1}^+) = \mu_i(t_{k+1}^-)\exp(-\hat{\bar{x}}'(t_k))} \tag{38}$$

4 Simulation And Discussions

This section provides verification for the proposed technique in a Matlab simulated environment. A team of two-wheeled differential drive robots are moving in the field. The given inputs are such that all robots move along a straight line or a circular arc. However, due to the stochastic nature of the systems, errors accumulate over time such that odometry or dynamics alone is insufficient in estimating the robot poses. The results from the previous sections can therefore be used to update the robots' knowledge of their poses with the help of measuring their positions relative to neighboring robots.

Figure 1 depicts a simple model of the two-wheeled differential drive robot which is very useful in modeling segway-like mobile bases and various multi-robot experimental platforms (E-pucks, iRobot create, Khepera, etc). According to [10], the kinematics of such a mobile robot can be characterized by the stochastic differential equation

$$(g^{-1}\dot{g})^\vee dt = \begin{bmatrix} \frac{r}{2}(\omega_1 + \omega_2) \\ 0 \\ \frac{r}{2}(\omega_1 - \omega_2) \end{bmatrix} dt + \sqrt{D} \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ 0 & 0 \\ \frac{r}{l} & -\frac{r}{l} \end{bmatrix} \begin{bmatrix} dw_1 \\ dw_2 \end{bmatrix} \tag{39}$$

where $g \in SE(2)$ is the homogenous matrix representing the pose of the robot, r is the wheel radius, l is the axle length, ω_1, ω_2 are the wheel angular velocities, dw_i are unit strength Wiener processes and D is a noise coefficient. This stochastic differential system can be simulated using the the Euler-Maruyama Method described in [7]. Equation 39 can be written in short as

$$(g^{-1}\dot{g})^\vee dt = \mathbf{h}dt + \mathbf{H}d\mathbf{w} \tag{40}$$

when g is close to the identity, given an input pair $[\omega_1, \omega_2]^T$, the mean and covariance of system 39 can be estimated by
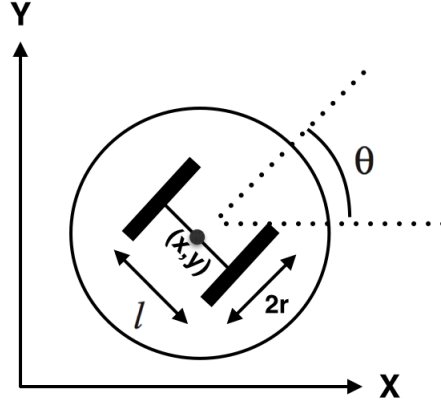
Fig. 1: Simple model for a two-wheeled differential drive mobile system

$$\mu(t) = \exp\left(\int_0^t \hat{\mathbf{h}}\, d\tau\right)$$

$$\Sigma(t) = \int_0^t Ad(\mu^{-1}(t-\tau))\mathbf{H}\mathbf{H}^T Ad^T(\mu^{-1}(t-\tau))\, d\tau \tag{41}$$

For simple motions like straight-line motion when $\omega_1 = \omega_2$, (41) can be evaluated analytically as

$$\mu(t)_{st} = \begin{bmatrix} 1 & 0 & r\omega t \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \Sigma(t)_{st} = \begin{bmatrix} \frac{1}{2}Dr^2 t & 0 & 0 \\ 0 & \frac{2D\omega^2 r^4 t^3}{3l^2} & \frac{D\omega r^3 t^2}{l^2} \\ 0 & \frac{D\omega r^3 t^2}{l^2} & \frac{2Dr^2 t}{l^2} \end{bmatrix} \tag{42}$$

The same can be done with circular motion of constant curvature

$$\mu(t)_{cir} = \begin{bmatrix} \cos(\dot{\alpha}t) & -\sin(\dot{\alpha}t) & a\sin(\dot{\alpha}t) \\ \sin(\dot{\alpha}t) & \cos(\dot{\alpha}t) & a(1-\cos(\dot{\alpha}t)) \\ 0 & 0 & 1 \end{bmatrix}, \quad \Sigma(t)_{cir} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \tag{43}$$

where

$$\sigma_{11} = \frac{c}{8}[(4a^2 + l^2)(2\dot{\alpha}t + \sin(2\dot{\alpha}t)) + 16a^2(\dot{\alpha}t - \sin(2\dot{\alpha}t))]$$

$$\sigma_{12} = \sigma_{21} = \frac{-c}{2}[4a^2(-1 + \cos(2\dot{\alpha}t)) + l^2]\sin(\dot{\alpha}t/2)^2$$

$$\sigma_{13} = \sigma_{31} = 2ca(\dot{\alpha}t - \sin(2\dot{\alpha}t))$$

$$\sigma_{23} = \sigma_{32} = -2ca(-1 + \cos(\dot{\alpha}t)) \tag{44}$$

$$\sigma_{33} = 2c\dot{\alpha}$$

$$c = \frac{Dr^2}{l^2\dot{\alpha}}$$

With the pose priors calculated with (41) - (44), (36) - (38) are then applied with sensor measurements to update the priors. For arbitrary inputs (ω_1, ω_2) an approximation will be applied to (41) which will be discussed in the next section.

This example simulates localization of a robot team in straight and circular motion. In the set-up of this simulation, the model based parameters are set as $r = 0.033, l = 0.2$. The simulation parameters for straight-line motion are $D = 5, v = 0.5, T = 1.3$, $\omega_1 = \omega_2 = \frac{v}{r}$ and $T = 2$, $\omega_1 = 26.166, \omega_2 = 21.408$ (for circular motion). The true robot motions are simulated 500 times using the Euler-Maruyama Method [7] and the end position of each trial is plotted in the following figures. It can be observed that the posterior of such a stochastic differential system (SDE) results in a banana shaped distribution as is also discussed in [15].

In this simulation, all four robots are given the command to travel in a straight line for 1.3 seconds at 0.5 m/s or along an arc of constant curvature of $1m$ at 45 deg/s for 1 second. The blue dashed lines in the figures represent the desired path of travel with the blue points at the two ends representing initial to final position. However due to process noise each robot will eventually end up somewhere near the desired goal and our objective is to estimate their true position along with a quantification of our confidence of this estimate. Specifically for this example, the true pose of the middle robot (cyan colored) is what we're trying to estimate which we'll call robot i, while the neighboring robots (yellow) are members of this team where relative measurements are obtained from. Among all the sampled end positions, one position for each robot is chosen as the true end pose (red point) and this is used to generate the mean of the measurement distribution m_{in}.

As the first step, the prior mean and covariance of robot i is calculated using (42) and (43), and plotted in Figure 2a and 2b, the resultant prediction aligns perfectly with the desire path (blue dash line), and the error "ellipse" marginalized over the heading angle is also plotted from the calculated covariance (magenta loop). Since this error "ellipse" is a contour of the resultant distribution, It can be observed that a Gaussian distribution under exponential coordinates is a much better fit for characterizing the uncertainties in an SDE of this kind than that under Cartesian coordinates. It is obvious that this prediction gives the same resultant distribution regardless of the true position and is only effected by the system dynamics and input commands. Therefore the next step is to update this prediction with measurements relative to neighboring robots.
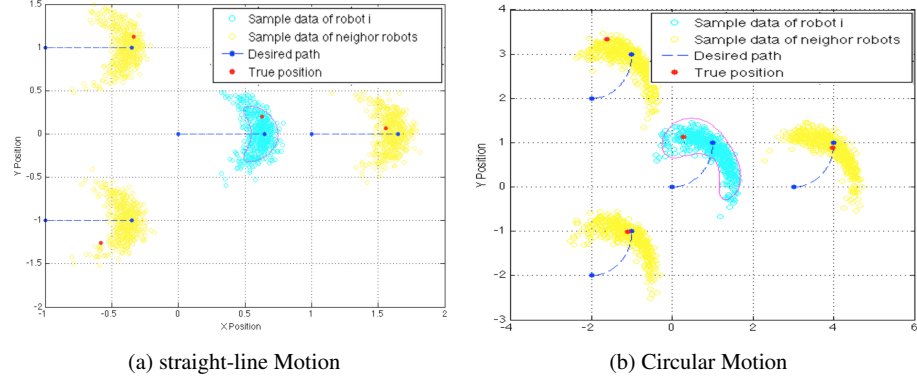
(a) straight-line Motion   (b) Circular Motion

Fig. 2: Localization with only the prediction model

It is assumed that robot i can exchange information with its neighbor when they come into its sensing range, which means when a relative measurement is taken of neighbor j relative to robot i, the belief (mean and covariance in this case) that j holds for its current position can be communicated to i so that i can make use of this information in its localization process (update step). In this example, this belief (μ_j, Σ_j) for each neighboring robot j is taken to be the pose prediction calculated from (42) or (43), but in reality this can very well be the posterior from their own localization results. The covariance of the measurement distribution is chosen to be

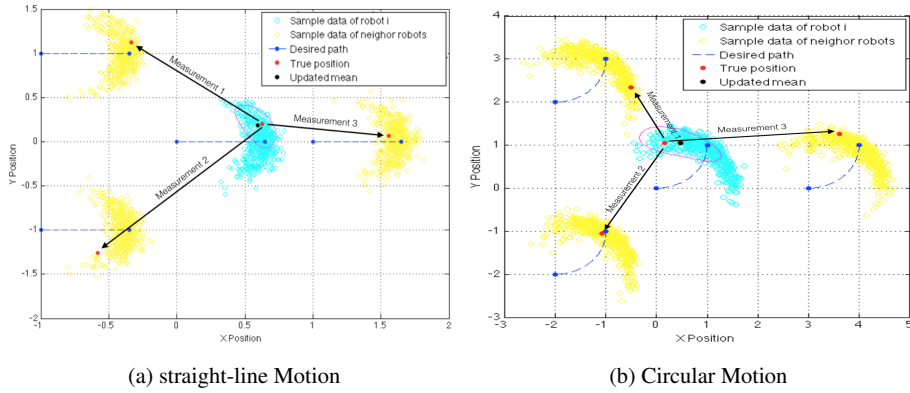$$\Sigma_m = \begin{bmatrix} 0.01 & 0.02 & 0.001 \\ 0.02 & 0.25 & 0.015 \\ 0.002 & 0.0025 & 0.15 \end{bmatrix}$$

(a) straight-line Motion   (b) Circular Motion

Fig. 3: Pose update after sensor measurement and fusion

Figures 3a and 3b show the updated posterior of robot *i* calculated from fusing the relative measurements taken from its three neighbors. The result indicates a more accurate position mean (black dot) and a shrinked error "ellipse" representing higher confidence in the estimate. Since this is a distributed localization technique aimed to be implemented on the embedded processor of each individual robot, the procedure is demonstrated only for one robot and the same goes for all other.

5 Conclusion

This chapter proposed a distributed cooperative localization technique that can incorporate multiple sensor measurements to achieve higher estimation accuracy. Robots in a team can take measurements and exchange information among each other to update their knowledge of the current position. Simulation is used to validate the performance of the approach in Matlab. Results from the Matlab simulation show a good localization accuracy of the presented approach. The proposed technique is distributed in that each robot can perform this localization process without the help of a centralized processor, and is scalable for the computation time does not increase as the robot team enlarges and increases only linearly with the number of measurements taken. The generality of this scheme lies in the fact that uncertainties in the belief of the current robot, neighboring robots and sensor measurements have all been considered which yields a more realistic estimate. Unlike sampling-based approaches, the proposed approach provides closed-form expressions which significantly increases computational efficiency. Most existing cooperative localization schemes possess a subset of the the above attributes but rarely all. Lastly, this technique is of second order in its estimation of an updated posterior which is expected to be more accurate and reliable than first order methods.

The limitation of this method is its dependency on Gaussian noises. Moreover, at present this is a local technique in that it depends on known initial poses and does not recover from localization failures (defined by [15]). In its current state, this approach does not possess the ability to serve as the sole scheme to localize a team of robots in that as errors accumulate in the beliefs of neighboring robots, erroneous information will be given to the current robot that leads to localization failures. However, this technique is local and prone to error accumulation only when none of the member robots have a reasonable estimate of their positions. As long as one robot possesses a good knowledge of its current pose (via more accurate sensors or sophisticated but computationally expensive algorithms) then this information can be used to drastically reduce the uncertainty of the entire team which introduces a level of robustness to the system and can also significantly reduce hardware and computational cost of the team. Table 2 shows a comparison of the proposed method with two of the most representative and accepted approaches.

The accuracy of the exponential localization method is expect to see great increase compared to results shown previously if the algorithm parameters (initial pose covariance, process and measurement noise covariances, etc) can be fine tuned.

Table 2: Final Comparison

	Distibuted EKF	Multi-Robot MCL	Exponential Localization
Restrictions on ErrorDistribution	Requires Gaussian process and measurement error	Nonparametric particle representation of posterior belief, no assumptions on noise distribution	Gaussian
Global Localization	No	Yes	No
State Recovery	No	Possible given a well designed resampling process	No
Localization Accuarcy	Accurate when error is small	Depends on the number of particles used	Accurate within an error range
Computational Cost	Small due to the closed form propagation and update equations	Depends on the number of particles. Can increase dramatically with the dimension of the state space	Small due to closed form equations
Ease of Implementation	Simple	Can be involved	Simple
Robustness	Prone to error due to linearization	Quite resistant to errors given multiple beliefs are maintained simultaneously	Less susceptible to errors given the well conformity to the motion model
Process Multiple Detections	No	No	Yes
Complexity Relative To Team Size	Fully distributed. Complexity independent of team size	Complexity independent of team size	Linear to the Number of measurements processed

Establishing a systematic way of tuning these parameters can be a topic of its own. It is also incredibly beneficial if the proposed method can be combined with sampling based approaches for their global localization and state recovery abilities. Lastly, experiments on hardware are required to fully establish the advantage of the proposed scheme. Overall this chapter has provided an alternative distributed cooperative localization technique in the domain of *Lie Group* and *Exponential Coordinates* and has validated in simulation the potential of this technique as the next state of the art.

Acknowledgment

References

1. P. Barooah, J. Russell, and J. Hespanha. Approximate distributed Kalman filtering for cooperative multi-agent localization. In *Distributed Computing in Sensor Systems*, volume 6131, pages 102–115, 2010.
2. V. Caglioti, A. Citterio, and Fossati A. Cooperative, distributed localization in multi-robot systems: a minimum-entropy approach. In *Distributed Intelligent Systems: Collective Intelligence and Its Applications*, pages 20–30, 2006.
3. L. Carlone, M. Kaouk, J. Du, B. Bona, and M. Indri. Simultaneous localization and mapping using Rao-Blackwellized particle filters in multi robot systems. *J Intell Robot Syst*, 63(283-307), 2011.
4. G. Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 2*. Springer Science+Business Media, 2012.
5. G. Chirikjian and M Kobilarov. Gaussian approximation of non-linear measurement models on Lie groups. *IEEE Conference on Decision and Control*, pages 6401–6406, 2014.
6. D. Fox, W. Burgard, H. Kruppa, and S. Thrun. A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8(3):325–344, 2000.
7. H. Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. In *SIAM review*, pages 525–546, 2001.
8. A. Howard. Multi-robot simultaneous localization and mapping using particle filters. *International Journal of Robotics Research*, 25(12):1243–1256, 2006.
9. J. Liu, W. Yuan, K.and Zou, and Q. Yang. Monte Carlo multi-robot localization based on grid cells and characteristic particles. In *International Conference on Advanced Intelligent Mechatronics*, pages 510–515, 2005.
10. A. Long, K. Wolfe, M. Mashner, and G. Chirikjian. The banana distribution is Gaussian: A localization study with exponential coordinates. In *Robotics: Science and Systems*, 2012.
11. A. Marjovi, L. Marques, and J. Penders. Guardians robot swarm exploration and firefighter assistance. In *IEEE/RSJ International Conference on Intelligent Robots and Systems: Workshop on Network Robot Systems*, 2009.
12. A.I. Mourikis and S.I Roumeliotis. Performance analysis of multirobot cooperative localization. *IEEE Transactions on Robotics*, 22(4):666–681, 2006.
13. W Park, Y LIu, Y Zhou, M Moses, and G. Chirikjian. Kinematic state estimation and motion planning for stochastic nonholonomic systems using the exponential map. *Robotica*, 26:419–434, 2008.
14. S. Roumeliotis and G. Bekey. Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5):781–795, 2002.
15. S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2006.
16. Y. Wang and G. Chirikjian. Nonparametric second-order theory of error propagation on motion groups. *The International Journal of Robotics Research*, 27(11-12):1258–1273, 2008.