Addressing Data Association in Maximum Likelihood SLAM with Random Finite Sets

Felipe Inostroza[1], *Student Member IEEE* and Martin Adams[2], *Senior Member IEEE*

Abstract—Recently, various algorithms which adopt Random Finite Sets (RFS) for the solution of the fundamental, autonomous robotic, feature based, Simultaneous Localization and Mapping (SLAM) problem, have been proposed. In contrast to their vector based counterparts, these techniques jointly estimate the vehicle and map state and map cardinality. Most of the proposed RFS solutions are based on a Rao-Blackwellized particle filter representing the vehicle state, accompanied by an RFS filter to represent the map. This article shows that an RFS maximum likelihood approach to SLAM is also possible.

By maximizing the RFS based measurement likelihood this article demonstrates that Maximum Likelihood (ML) SLAM is possible without the need for external data association algorithms. It will be demonstrated that RFS based ML-SLAM converges to the same solution as its traditional vector-based counterpart. However, fundamentally RFS-ML-SLAM does not require the correct data association decisions necessary for the correct convergence of traditional random vector based approaches.

I. INTRODUCTION

Random finite set (RFS)-based filters have been shown to outperform traditional vector-based filters when exposed to significant clutter (i.e., false alarms), both in target tracking and simultaneous localization and mapping (SLAM) applications [1]. Probability hypothesis density (PHD) filters have been used in solutions to SLAM in dynamic environments [2, 3]. However, the PHD filter, which is the main RFS based method that has been applied to SLAM using experimental data to date, has been described as having "poor memory"; i.e., it tends to discard old information in favour of new measurements [4]. It has also been found to have a lower performance than vector based filters under low amounts of clutter in some SLAM applications [1]. Some of the methods that attempt to solve the problems of the PHD filter are the cardinalized PHD (CPHD) and cardinality balanced multi-target multi-Bernoulli (CB-MemBer) filters, and more recently the labelled multi Bernoulli (LMB) filter [4, 5, 6]. The LMB filter has been adapted to SLAM and has been shown to outperform PHD-SLAM in simulation [7].

Concurrent to the development of RFS based filters in the tracking community, the robotics SLAM community has moved away from filtering based solutions to batch estimation approaches, which use non-linear optimization methods to obtain a Maximum Likelihood (ML) solution [8, 9]. Such methods include graphSLAM [10], iSAM [11] and iSAM2 [8]. These algorithms, which are usually based on non-linear least squares optimization, provide more accurate solutions over larger datasets than their filtering counterparts. However, these methods still rely on external routines to perform data association and map management, usually based on either maximum likelihood or place recognition algorithms. The RFS formulation of the problem in this batch form complicates the optimization process considerably, even when assuming a known number of landmarks.

The facts presented above provide a compelling reason to derive a maximum likelihood RFS based method, which should result in a superior performance both for large datasets and under high clutter and detection uncertainty. In this paper an initial version of such a method will be presented. By maximizing the RFS based measurement likelihood, this article demonstrates that Maximum Likelihood (ML) SLAM is possible without the need for external data association algorithms. Fundamentally, it will be shown that RFS based ML-SLAM converges to an equivalent solution as its traditional vector-based counterpart, but without the need for correct data association information. The validity of this approach will be shown through the solution of 1D and 2D simulated datasets.

Section II shows the traditionally used random vector least squares approach. In Section III the formulation of the maximum likelihood SLAM problem will be presented within the RFS framework. In Section IV the particle swarm optimization algorithm is presented, which is then adapted in Section V to the RFS-SLAM problem. Results of 1D and 2D simulations are shown in Section VI, and Section VII summarizes the main contributions of the article.

II. RANDOM VECTOR BASED SLAM BATCH ESTIMATION

SLAM solutions using the batch estimation approach are usually maximum likelihood strategies in which the measurement likelihood function, $p\left(\mathcal{Z}_{0:k} \middle| \boldsymbol{x}_{0:k}, \boldsymbol{m}, \boldsymbol{u}_{0:k-1}\right)$, is maximized over all possible trajectories, $\boldsymbol{x}_{0:k}$, and feature maps, $\boldsymbol{m} = [\boldsymbol{m}^1, \dots, \boldsymbol{m}^{|\boldsymbol{m}|}]$. $\mathcal{Z}_{0:k}$ corresponds to all measurement detections $\mathcal{Z}_i = \{\boldsymbol{z}_i^1, \dots, \boldsymbol{z}_i^{|\mathcal{Z}_i|}\}$ from time 0 to k inclusive, and $\boldsymbol{u}_{0:k-1}$ corresponds to odometry measurements from time 0 to $k-1$ inclusive. Assuming that the number of landmarks is known, and using a known data association hypothesis

[1]F. Inostroza is a Ph.D. research student in the Dept. of Electrical Engineering, Universidad de Chile, Av. Tupper 2007, 837-0451, Santiago, Chile. His Ph.D. is being financed by CONICYT-PCHA/Doctorado Nacional/2014 finostro@ug.uchile.cl

[2]M. Adams is Professor of Electrical Engineering, Dept. of Electrical Engineering, and Principle Investigator (PI) and in the Advanced Mining Technology Center (AMTC), Universidad de Chile, Av. Tupper 2007, 837-0451, Santiago, Chile. martin@ing.uchile.cl

function $\theta(\cdot)$, this likelihood can be expressed as:

$$p\left(\mathcal{Z}_{0:k}\middle|\boldsymbol{x}_{0:k},\boldsymbol{m},\boldsymbol{u}_{0:k-1},\theta\right)$$
$$=\prod_{i=1}^{k}\boldsymbol{g}(\boldsymbol{x}_i|\boldsymbol{x}_{i-1},\boldsymbol{u}_{i-1})\prod_{i=0}^{k}\prod_{j=1}^{|\mathcal{Z}_i|}\boldsymbol{h}(\boldsymbol{z}_i^j|\boldsymbol{x}_i,\boldsymbol{m}^{\theta(j)})\ ,\quad(1)$$

where the function $\theta(j)$ represents the data association of detection $\boldsymbol{z}_i^j$ with map element $\boldsymbol{m}^{\theta(j)}$. Assuming that the motion and measurements are modelled with additive Gaussian noise,

$$\boldsymbol{g}(\boldsymbol{x}_i|\boldsymbol{x}_{i-1},\boldsymbol{u}_{i-1})=N(\boldsymbol{x}_i;\hat{\boldsymbol{x}}_i(\boldsymbol{x}_{i-1},\boldsymbol{u}_{i-1}),\Sigma_g),\quad(2)$$
$$\boldsymbol{h}(\boldsymbol{z}_i^j|\boldsymbol{x}_i,\boldsymbol{m}^{\theta(j)})=N(\boldsymbol{z}_i^j;\hat{\boldsymbol{z}}_i^j(\boldsymbol{x}_{i-1},\boldsymbol{m}^{\theta(j)}),\Sigma_h)\ ,\quad(3)$$

where $\hat{\boldsymbol{z}}_i^j(\boldsymbol{x}_{i-1},\boldsymbol{m}^{\theta(j)})$ and Σ_h are the mean and covariance of the measurement model, and $\hat{\boldsymbol{x}}_i(\boldsymbol{x}_{i-1},\boldsymbol{u}_{i-1})$ and Σ_g are the mean and covariance of the motion model. This allows the maximization likelihood problem to be expressed in a non-linear least squares form, by applying the $\log()$ function to (1):

$$\arg\max_{\boldsymbol{x}_{1:k},\boldsymbol{m}}\prod_{i=1}^{k}\boldsymbol{g}(\boldsymbol{x}_i|\boldsymbol{x}_{i-1},\boldsymbol{u}_{i-1})\prod_{i=0}^{k}\prod_{j=1}^{|\mathcal{Z}_i|}\boldsymbol{h}(\boldsymbol{z}_i^j|\boldsymbol{x}_i,\boldsymbol{m}^{\theta(j)})$$
$$=\arg\min_{\boldsymbol{x}_{1:k},\boldsymbol{m}}\sum_{i=1}^{k}(\boldsymbol{x}_i-\hat{\boldsymbol{x}}_i)^T\Sigma_g^{-1}(\boldsymbol{x}_i-\hat{\boldsymbol{x}}_i)$$
$$+\sum_{i=0}^{k}\sum_{j}^{|\mathcal{Z}_i|}(\boldsymbol{z}_i^j-\hat{\boldsymbol{z}}_i^j)^T\Sigma_h^{-1}(\boldsymbol{z}_i^j-\hat{\boldsymbol{z}}_i^j)\ ,\quad(4)$$

where, for briefness, the arguments of $\hat{\boldsymbol{z}}_i^j$ and $\hat{\boldsymbol{x}}_i$ have been omitted, i.e. $\hat{\boldsymbol{z}}_i^j=\hat{\boldsymbol{z}}_i^j(\boldsymbol{x}_{i-1},\boldsymbol{m}^{\theta(j)})$ and $\hat{\boldsymbol{x}}_i=\hat{\boldsymbol{x}}_i(\boldsymbol{x}_{i-1},\boldsymbol{u}_{i-1})$. This can be solved using any of the several dedicated algorithms to solve non-linear least squares problems, such as the Levenberg-Marquardt (LM) [12] or Broyden-Fletcher-Goldfarb-Shanno (BFGS) [13] methods.

A. A note on assuming data association

A problem with the formulation stated previously was shown in [4] (page 340). This problem is that by conditioning the measurement likelihood in (1) to the data association function $\theta(\cdot)$ a particular *order* or *numbering* of the measurements is necessarily assumed. If a different order of the measurements $\mathcal{Z}_i$ were to be measured, then the association function $\theta(\cdot)$ would have to change also. Given this, it may be that by including this extraneous information, on the order of measurements, a statistical bias could be introduced.

To address this problem, RFS methods replace the measurement likelihood with the average over all possible data associations:

$$p\left(\mathcal{Z}_i\middle|\boldsymbol{x}_k,\mathcal{M}\right)=\sum_{\theta}p\left(\mathcal{Z}_i\middle|\boldsymbol{x}_k,\mathcal{M},\theta\right)\quad(5)$$

where θ represents a possible data association between map element $\boldsymbol{m}^{\theta(j)}$ and measurement $\boldsymbol{z}_i^j$. In this likelihood, if the order of measurements is changed, then the correct data association is still part of the total sum, and therefore the likelihood does not change.

III. RFS-BASED SLAM BATCH ESTIMATION

Similarly to the formulation in the previous section, the objective function $p\left(\mathcal{Z}_{0:k}\middle|\boldsymbol{x}_{0:k},\boldsymbol{m},\boldsymbol{u}_{0:k-1}\right)$ can be stated using the RFS framework, by using the set-based measurement and motion models:

$$p\left(\mathcal{Z}_{0:k}\middle|\boldsymbol{x}_{0:k},\boldsymbol{m},\boldsymbol{u}_{0:k-1}\right)$$
$$=\prod_{i=1}^{k}\boldsymbol{g}(\boldsymbol{x}_i|\boldsymbol{x}_{i-1},\boldsymbol{u}_{i-1})\prod_{i=0}^{k}p\left(\mathcal{Z}_i\middle|\boldsymbol{x}_i,\mathcal{M}\right)\ .\quad(6)$$

As can be seen from (6), the likelihood is very similar to (1) but uses the RFS based measurement model $p\left(\mathcal{Z}_i\middle|\boldsymbol{x}_i,\mathcal{M}\right)$:

$$p\left(\mathcal{Z}_i\middle|\mathbf{x}_i,\mathcal{M}\right)=p\left(\mathcal{Z}_i\middle|\mathbf{x}_i,\left\{\boldsymbol{m}^1,\boldsymbol{m}^2,\dots,\boldsymbol{m}^{|\mathcal{M}|}\right\}\right)$$
$$=\sum_{\theta}p_\kappa\left(\mathcal{Z}_i^{\bar{A}_\theta}\right)\prod_{\boldsymbol{m}^j\in\mathcal{M}^{\bar{A}_\theta}}\left(1-P_{\mathrm{D}}\left(\boldsymbol{m}^j|\boldsymbol{x}_k\right)\right)$$
$$\cdot\prod_{\boldsymbol{z}_i^j\in\mathcal{Z}_i^{A_\theta}}P_{\mathrm{D}}\left(\boldsymbol{m}^{\theta(j)}|\boldsymbol{x}_i\right)p\left(\boldsymbol{z}_i^j\middle|\boldsymbol{m}^{\theta(j)},\mathbf{x}_i\right)\quad(7)$$

where θ is a possible association between the elements of $\mathcal{M}$ and $\mathcal{Z}_i$, and $\mathcal{Z}_i^{\bar{A}_\theta}$ is the set of measurements in $\mathcal{Z}_i$ that are not associated with a landmark in the map according to θ. $p_\kappa\left(\mathcal{Z}_k\right)$ is the probability of all measurements in $\mathcal{Z}_k$ being clutter[1]. Similarly, $\mathcal{Z}_i^{A_\theta}$ is the set of measurements in $\mathcal{Z}_i$ that are associated with a landmark in $\mathcal{M}$, again according to θ. Applying the $\log()$ function to (6) and using the relation from (7) gives:

$$l\left(\mathcal{Z}_{0:k}\middle|\boldsymbol{x}_{0:k},\boldsymbol{m},\boldsymbol{u}_{0:k-1}\right)$$
$$=\sum_{i=1}^{k}\log\left(\boldsymbol{g}(\boldsymbol{x}_i|\boldsymbol{x}_{i-1},\boldsymbol{u}_{i-1})\right)+\sum_{i=0}^{k}\log\left(p\left(\mathcal{Z}_i\middle|\boldsymbol{x}_i,\mathcal{M}\right)\right)$$
$$=\sum_{i=1}^{k}\log\left(\boldsymbol{g}(\boldsymbol{x}_i|\boldsymbol{x}_{i-1},\boldsymbol{u}_{i-1})\right)+\sum_{i=0}^{k}\log\left(\sum_{\theta}p\left(\mathcal{Z}_i\middle|\boldsymbol{x}_i,\mathcal{M},\theta\right)\right)\ ,\quad(8)$$

Where $l(\cdot)=\log(p(\cdot))$ represents the log-likelihood. In [1, 14], (7) is derived and shown to generalize its random vector based counterpart.

Using the RFS-based objective function from (7), non-linear optimization can be performed over all possible map sizes $|\mathcal{M}|$. This function would not require a data association hypothesis to perform the optimization. However, the partly integer nature of the optimization (i.e. the number of features in the map) is expected to cause significant difficulties for the optimization method.

IV. PARTICLE SWARM OPTIMIZATION

One method to maximize a non-linear function, as needed by any Maximum Likelihood solution to SLAM, is Particle Swarm Optimization (PSO) [15]. Due to the heuristic nature of

---

[1]Measurements/detections resulting from "objects on non-interest" - i.e. not part of the target state.

PSO, many variations exist. An explanation of one of the most common variations, known as Standard PSO 2007 (SPSO-2007), and used in this article, follows.

Let the optimization problem to solve be

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \ , \tag{9}$$

where $\boldsymbol{x}$ is the variable to optimize with respect to function $f(\cdot)$. In PSO a set of particles with positions $\boldsymbol{x}$ and velocities $\boldsymbol{v}$ is defined with the expectation that, as the particles move according to their velocities

$$\boldsymbol{x}_{j+1}^i = \boldsymbol{x}_j^i + \boldsymbol{v}_j^i \ , \tag{10}$$

the positions of the particles will converge to the optimal solution $\boldsymbol{x}^*$. SPSO-2007 uses a fixed number of particles n_p according to

$$n_p = \lfloor 10 + 2\sqrt{D} \rfloor \tag{11}$$

where D is the number of dimensions of $\boldsymbol{x}$. This formula is considered a suggested particle swarm size, but is not required for SPSO-2007 compatibility. Particle positions and velocities are initialized randomly following the uniform distribution within the search space $[\boldsymbol{x}_{min}, \boldsymbol{x}_{max}]$

$$\boldsymbol{x}_0^i = U(\boldsymbol{x}_{min}, \boldsymbol{x}_{max}) \tag{12}$$
$$\boldsymbol{v}_0^i = 0.5(U(\boldsymbol{x}_{min}, \boldsymbol{x}_{max}) - \boldsymbol{x}_0^i) \ . \tag{13}$$

Where $U(a, b)$ is the uniform distribution in the interval $[a, b]$. Then each particle will calculate its new velocity as a function of its own position and current velocity and the positions of the particles within a neighbourhood - i.e. for each component of particle i at iteration j, the velocity will be

$$\boldsymbol{v}_j^i = w\boldsymbol{v}_j^i + r_1\phi_p(\boldsymbol{l}_j^i - \boldsymbol{x}_j^i) + r_2\phi_g(\boldsymbol{g}_j^i - \boldsymbol{x}_j^i) \ , \tag{14}$$

where r_1 and r_2 are random numbers uniformly distributed in the interval $[0, 1]$, $\boldsymbol{l}_j^i$ is the best (lowest $f(\cdot)$) position visited by particle i and $\boldsymbol{g}_j^i$ is the best position visited by any of the particles in particle i's neighbourhood. w, ϕ_p and ϕ_g are parameters typically set to the values[2]:

$$w = \frac{1}{2\ln(2)} \simeq 0.721 \tag{15}$$
$$\phi_p = \phi_g = 0.5 + \ln(2) \simeq 1.193 \ .$$

This neighbourhood is defined randomly by having each particle inform K other particles at random (i.e. adding themselves to the other particles' neighbourhoods). Typically $K = 3$. If at any iteration the best solution found does not improve, then all neighbourhoods are randomly redrawn using the same process [15].

[2]As stated in [15], these parameters have been shown empirically to produce good results.

V. PSO-RFS-SLAM

To maximize the likelihood from RFS based equation (6) using a PSO approach, a modified SPSO-2007 is used. First, for numerical stability, the log-likelihood (8) is used. A set of particles is created, each with state consisting of a trajectory and a map:

$$\boldsymbol{x}_j^i = (\mathbf{x}_{0:k}^i, \mathcal{M}^i). \tag{16}$$

A. Particle initialization

Given that the search space is not constrained as in the SPSO-2007 definition, to initialize the particle set, the robot motion model is used to generate a random dead reckoning trajectory. Once a particle has a trajectory for each of the measurements obtained, a new landmark is created with probability P_{init}. Landmarks are created with the inverse measurement model and the already initialized robot pose, and sampling noise from the landmark uncertainty generated from the inverse measurement model.

B. Particle motion

In most PSO definitions, the state is a vector. This means that equation (14) can be directly applied to the robot trajectory $\mathbf{x}_{0:k}$, however, the map $\mathcal{M}$ is a set, thus (14) needs to be adapted.

We therefore define the velocity of a set by attaching a velocity component to each vector element of the set. Then, to modify the velocity of a set we use a method inspired by the optimal sub-pattern assignment (OSPA) metric [16]. Map elements are associated by linear assignment as if calculating the OSPA metric, and then the velocity of the associated elements can be calculated directly, using equation (14). For the unassociated measurements a new parameter ϕ_{card} is added. Then, if a map element in $\boldsymbol{x}_j^i$ is not associated it will be eliminated with probability equal to ϕ_{card}, and unassociated map elements in $\boldsymbol{l}_j^i$ and $\boldsymbol{g}_j^i$ will be added to $\boldsymbol{x}_j^i$ with probability ϕ_{card}. This parameter is set to a small value (e.g. $\phi_{card} \simeq 0.15$). As an added modification, even associated features can be removed with a smaller probability ($\simeq 0.01$).

C. Gradient based optimization step

Given the high correlation between states, which characterizes the SLAM problem, when the particle motion is applied to the trajectory, the measurement likelihood usually decreases. To compensate for this and to improve the convergence properties of PSO, a gradient based optimization step is performed for every particle every K_{opt} steps ($K_{\mathrm{opt}} \in [10, 100]$).

To maximize the likelihood from equation (6) the log-likelihood (8) is used. The summation on the left of (8) is just a traditional non-linear least squares optimization problem (assuming Gaussian noise in $\boldsymbol{g}(\cdot)$), however, in the summation on the right, the $\log(\cdot)$ does not cancel with the exponentials. For the left summation the gradient can be calculated in a

straight forward manner, but for the summation on the right we obtain

$$\frac{\partial}{\partial \boldsymbol{x}_k} \log \left(\sum_{\theta} p \left(\mathcal{Z}_i \middle| \boldsymbol{x}_k, \mathcal{M}, \theta \right) \right) = \frac{\frac{\partial}{\partial \boldsymbol{x}_k} \sum_{\theta} p \left(\mathcal{Z}_i \middle| \boldsymbol{x}_k, \mathcal{M}, \theta \right)}{\sum_{\theta} p \left(\mathcal{Z}_i \middle| \boldsymbol{x}_k, \mathcal{M}, \theta \right)}. \tag{17}$$

If the single feature measurement model is Gaussian and the clutter rate and probability of detection are assumed constant then the gradient would be

$$\frac{1}{\sum_{\theta} p \left(\mathcal{Z}_i \middle| \boldsymbol{x}_k, \mathcal{M}, \theta \right)} \sum_{\theta} \left[p \left(\mathcal{Z}_i \middle| \boldsymbol{x}_k, \mathcal{M}, \theta \right) \right.$$

$$\left. \cdot \sum_{\boldsymbol{z}_i^j \in \mathcal{Z}_i^{A_\theta}} \Omega_{\boldsymbol{z}^j} (\boldsymbol{z}_i^j - \hat{\boldsymbol{z}}_i^j(\boldsymbol{m}^{\theta(j)}, \boldsymbol{x}_k)) \frac{\partial}{\partial \boldsymbol{x}_k} \hat{\boldsymbol{z}}_i^j(\boldsymbol{m}^{\theta(j)}, \boldsymbol{x}_k) \right] \tag{18}$$

where $\Omega_{\boldsymbol{z}^j}$ is the square root of the information matrix of measurement $\boldsymbol{z}^j$. Therefore, we obtain a weighted average of the traditional non-linear least squares, weighted by the RFS measurement likelihoods. However, the non-log likelihoods still need to be evaluated, which can cause numerical instabilities, especially if the initialization point is too far away from the real values for the map and trajectory (as occurs most of the time with a random initialization). Therefore to increase the numerical stability, the log-sum-exp trick is used

1) The log-sum-exp trick: Adding a log-exp pair to the right hand side term in equation (8) gives

$$l \left(\mathcal{Z}_{0:k} \middle| \boldsymbol{x}_{0:k}, \boldsymbol{m}, \boldsymbol{u}_{0:k-1} \right) = \sum_{i=1}^{k} \log \left(\boldsymbol{g}(\boldsymbol{x}_i | \boldsymbol{x}_{i-1}, \boldsymbol{u}_{i-1}) \right)$$

$$+ \sum_{i=0}^{k} \log \left(\sum_{\theta} \exp(\log(p \left(\mathcal{Z}_i \middle| \boldsymbol{x}_i, \mathcal{M}, \theta \right))) \right). \tag{19}$$

Expanding $p \left(\mathcal{Z}_i \middle| \boldsymbol{x}_i, \mathcal{M}, \theta \right)$ in the right hand side of (19):

$$\text{rhs(19)} = \sum_{i=1}^{k} \log \left(\boldsymbol{g}(\boldsymbol{x}_i | \boldsymbol{x}_{i-1}, \boldsymbol{u}_{i-1}) \right) + \sum_{i=0}^{k} \log \left(\sum_{\theta} \exp \left[\right. \right.$$

$$\log \left(\frac{\prod_{\boldsymbol{z}_i^j \in \mathcal{Z}_i} \kappa(\boldsymbol{z}_i^j)}{\exp(\int \kappa(\boldsymbol{z}) d\boldsymbol{z})} \prod_{\boldsymbol{m}^j \in \mathcal{M}} (1 - P_{\mathrm{D}}(\boldsymbol{m}^j | \boldsymbol{x}_i)) \right.$$

$$\left. \left. \left. \prod_{\boldsymbol{z}_i^j \in \mathcal{Z}^{A_\theta}} \frac{P_{\mathrm{D}}(\boldsymbol{m}^{\theta(j)} | \boldsymbol{x}_i)}{(1 - P_{\mathrm{D}}(\boldsymbol{m}^{\theta(j)} | \boldsymbol{x}_i)) \kappa(\boldsymbol{z}_i^j)} p \left(\boldsymbol{z}_i^j \middle| \boldsymbol{x}_i, \boldsymbol{m}^{\theta(j)} \right) \right) \right] \right). \tag{20}$$

Turning the log of the products into sums gives:

$$l \left(\mathcal{Z}_{0:k} \middle| \boldsymbol{x}_{0:k}, \boldsymbol{m}, \boldsymbol{u}_{0:k-1} \right) = \sum_{i=1}^{k} \log \left(\boldsymbol{g}(\boldsymbol{x}_i | \boldsymbol{x}_{i-1}, \boldsymbol{u}_{i-1}) \right)$$

$$+ \sum_{i=0}^{k} \log \left(\sum_{\theta} \exp \left[\sum_{\boldsymbol{z}_i^j \in \mathcal{Z}_i} \log(\kappa(\boldsymbol{z}_i^j)) + \int \kappa(\boldsymbol{z}) d\boldsymbol{z} \right. \right.$$

$$+ \sum_{\boldsymbol{m}^j \in \mathcal{M}^{A_\theta}} \log(1 - P_{\mathrm{D}}(\boldsymbol{m}^j | \boldsymbol{x}_i)) + \sum_{\boldsymbol{z}_i^j \in \mathcal{Z}^{A_\theta}} \mathrm{logit}(P_{\mathrm{D}}(\boldsymbol{m}^{\theta(j)} | \boldsymbol{x}_i))$$

$$\left. \left. - \log(\kappa(\boldsymbol{z}_i^j)) + l \left(\boldsymbol{z}_i^j \middle| \boldsymbol{x}_i, \boldsymbol{m}^{\theta(j)} \right) \right] \right) \tag{21}$$

where $l \left(\cdot | \cdot \right)$ is the single-feature log-likelihood, and the logit function is defined as $\mathrm{logit}(x) = \frac{\log(x)}{1 - \log(x)}$. Then the most likely data association can be taken out of the log function, by subtracting its value in every exponential giving:

$$l \left(\mathcal{Z}_{0:k} \middle| \boldsymbol{x}_{0:k}, \boldsymbol{m}, \boldsymbol{u}_{0:k-1} \right) = \sum_{i=1}^{k} \log \left(\boldsymbol{g}(\boldsymbol{x}_i | \boldsymbol{x}_{i-1}, \boldsymbol{u}_{i-1}) \right)$$

$$+ \sum_{i=0}^{k} \sum_{\boldsymbol{z}_i^j \in \mathcal{Z}_i} \left[\log(\kappa(\boldsymbol{z}_i^j)) + \int \kappa(\boldsymbol{z}) d\boldsymbol{z} \right.$$

$$+ \sum_{\boldsymbol{m}^j \in \mathcal{M}} \log(1 - P_{\mathrm{D}}(\boldsymbol{m}^j | \boldsymbol{x}_i)) + \sum_{\boldsymbol{z}_i^j \in \mathcal{Z}^{A_{\max}^\theta}} \mathrm{logit}(P_{\mathrm{D}}(\boldsymbol{m}^{\theta_{\max}(j)} | \boldsymbol{x}_i))$$

$$\left. - \log(\kappa(\boldsymbol{z}_i^j)) + l \left(\boldsymbol{z}_i^j \middle| \boldsymbol{x}_i, \boldsymbol{m}^{\theta_{\max}(j)} \right) \right]$$

$$+ \log \left(\sum_{\theta} \exp \left[\sum_{\boldsymbol{z}_i^j \in \mathcal{Z}^{A_\theta}} \mathrm{logit}(P_{\mathrm{D}}(\boldsymbol{m}^{\theta(j)} | \boldsymbol{x}_i)) - \log(\kappa(\boldsymbol{z}_i^j)) \right. \right.$$

$$+ l \left(\boldsymbol{z}_i^j \middle| \boldsymbol{x}_i, \boldsymbol{m}^{\theta(j)} \right) - \sum_{\boldsymbol{z}_i^j \in \mathcal{Z}^{A_{\max}^\theta}} \mathrm{logit}(P_{\mathrm{D}}(\boldsymbol{m}^{\theta_{\max}(j)} | \boldsymbol{x}_i))$$

$$\left. \left. - \log(\kappa(\boldsymbol{z}_i^j)) + l \left(\boldsymbol{z}_i^j \middle| \boldsymbol{x}_i, \boldsymbol{m}^{\theta_{\max}(j)} \right) \right] \right). \tag{22}$$

This objective function can now be interpreted as the log-likelihood using the most likely data association $\theta_{\max}$, which is calculated using the Hungarian method, plus a correction factor, which accounts for all other association possibilities. This correction factor approaches zero whenever only one data association is likely. Conversely, the correction factor will be nonzero whenever multiple data associations are likely. Note therefore, that the Hungarian method is only necessary in order to rewrite equation (21) in terms of differences between the log-likelihoods of the associations and the most likely association, resulting in a more numerically stable form. This avoids the calculation of exponentials of large negative values. This function is then optimized using the BFGS algorithm. Note that only the spatial position of the landmarks and poses of the robot are optimized, meaning that during this step the map size does not change. However, the estimate of the map size is optimized in the PSO particle motion step.

VI. Simulated SLAM Results

A. Proof of concept for RFS maximum likelihood SLAM in 1D

To show that using an RFS approach to maximum likelihood SLAM is feasible, a 1D simulation is carried out and compared with a traditional method using the correct data association. In this simulation, a 1D robot moves through a 1D map, observing landmarks and its own odometry. Both of these measurements are corrupted by zero mean Gaussian noise. To be able to use a traditional least squares random vector method, measurements are simulated with a label representing the landmark that generated it. These labels are then used by the traditional method for measurement detection to landmark association. On the contrary, the RFS formulation ignores these labels. A constant probability of detection of 0.9 and a clutter intensity of 0.01 were used to generate missed detections and Poisson distributed false alarms in the robot's field of view. Measurements were generated with a spatial variance of $10^{-4}[m^2]$, while odometry readings used a variance of $10^{-2}[m^2]$.

Figures 1 and 2 show an example solution to this problem, solved by maximizing the traditional measurement likelihood and the RFS based measurement likelihood from (6), respectively. As can be seen from the figures, both solutions converge to the ground truth trajectory and map. An important note is that, in addition to not having the data association, the RFS-based solution is also solving for the map size, making the optimization problem partly integer, which is significantly more difficult than solving only for the positions of the map elements. Figure 3 shows the robot position error of both algorithms, averaged over 5 Monte Carlo runs. Both errors are of the same order of magnitude.

B. RFS maximum likelihood in 2D

To demonstrate the validity of RFS ML SLAM further, a 2D simulation was carried out and compared with the traditional vector based approach. Measurements were simulated with a 0.9 probability of detection and a clutter intensity of 0.001 in the sensor's field of view. Measurements were generated with a range variance of $10^{-4}[m^2]$ and a bearing variance of $10^{-4}[rad^2]$, while odometry readings used a variance of $2 \times 10^{-4}[m^2]$, both in the x and y directions, and an angular variance of $2 \times 10^{-4}[rad^2]$. As can be seen from figures 4 and 5, both trajectories and maps converge to a solution very close to ground truth. Further, both solutions are similar to each other, presenting a small translational error in the same direction. This error may be because of the random nature of the measurements, meaning that the most likely trajectory is not necessarily the correct one. Figure 6 shows the x and y positional errors (averaged over 5 Monte Carlo runs) between the estimates and the simulated ground truth. As can be seen from the figure, again both algorithms yield very similar error curves, to the point that they are closer to each other than to the ground truth.
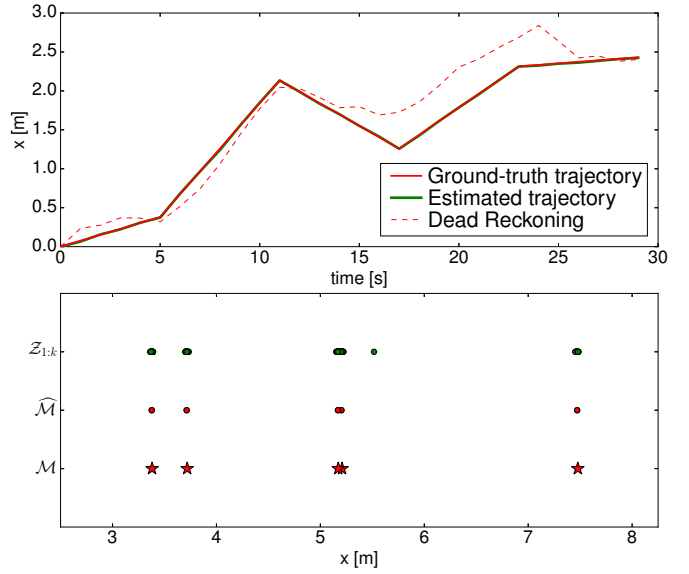

Fig. 1. *The solution to a 1D SLAM problem using the traditional least squares approach. The ground truth, estimated and dead reckoning trajectories are plotted as red, green, and dashed red lines, respectively (top). In the lower graph, the ground truth map is shown as red stars , the estimated map as red points and all the measurements are shown as green points.*
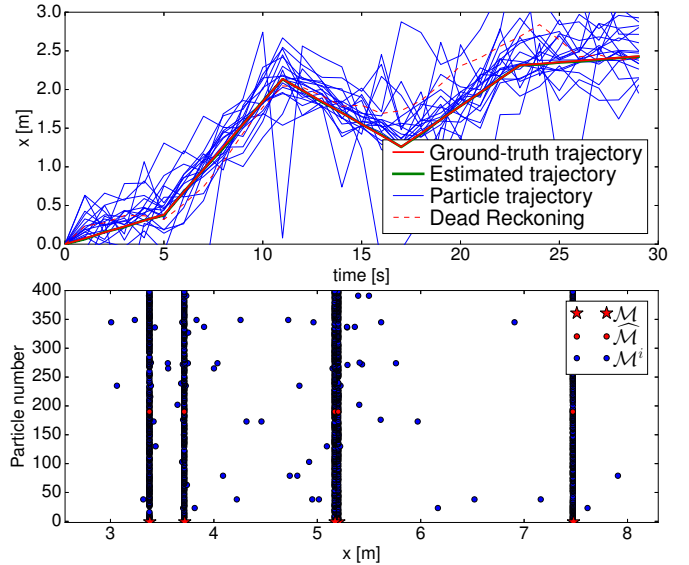

Fig. 2. *The solution to a 1D SLAM problem by maximizing the RFS measurement likelihood using an adapted PSO method. The particle, ground truth, estimated and dead reckoning trajectories are plotted as blue, red, green, and dashed red lines, respectively (top). Note that most particles have converged to the ground truth so they appear almost superimposed. The noisy trajectories that can be seen are a small fraction of the particles. In the lower graph, the ground truth map is shown as red stars (at $y = -1$), the estimated map is shown as red points (at $y =$ index of the most likely particle) and the estimated map of each particle is shown as blue points plotted at $y =$ index of each particle.*
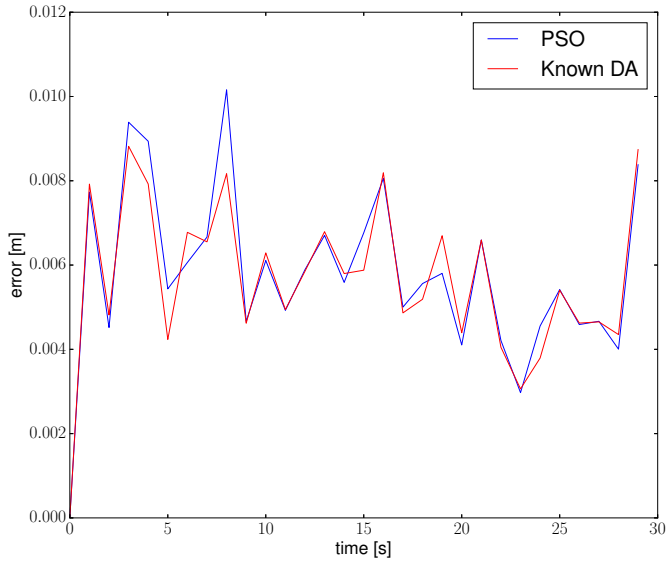
Fig. 3. *The absolute robot position error of the maximum likelihood solution obtained through the traditional vector based, least squares, approach plotted in red, and the RFS based maximum likelihood solution plotted in blue. Errors were averaged over 5 Monte Carlo runs. As can be seen both errors are of the same order of magnitude.*
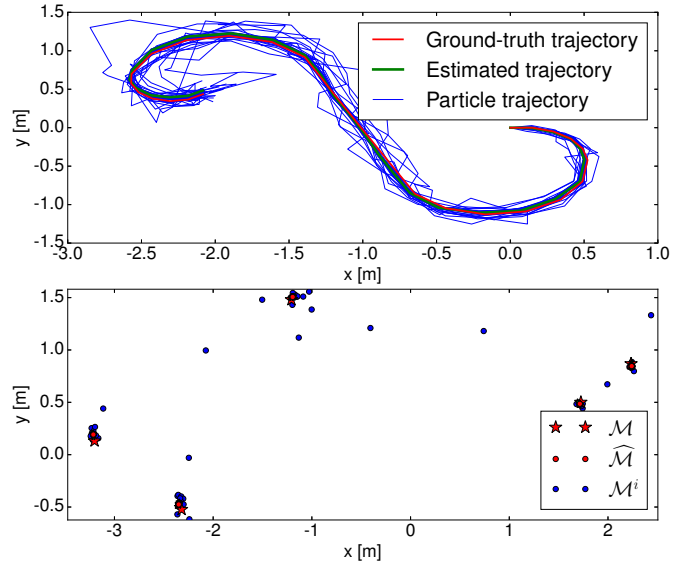

Fig. 5. *The solution to a 2D SLAM problem by maximizing the RFS measurement likelihood using an adapted PSO method. The particle, ground truth and estimated trajectories are plotted as blue, red, and green lines, respectively (top). Note that most of the particles have converged to the ground truth so they are shown superimposed. In the lower graph, the ground truth map is shown as red stars, the estimated map is shown as red points and the estimated map of each particle is shown as blue points.*
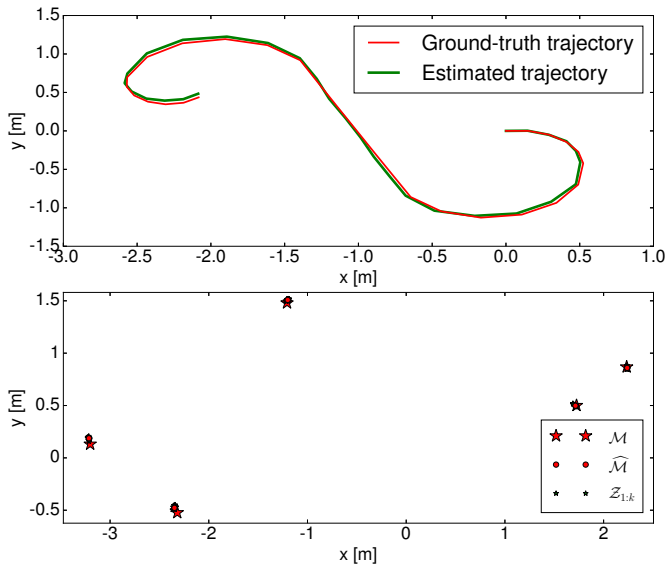

Fig. 4. *The solution to a 2D SLAM problem using the traditional least squares method. The ground truth and estimated trajectories are plotted as red and green lines, respectively (top). In the lower graph, the ground truth map is shown as red stars, the estimated map is shown as red points and all the measurements are shown as small green stars (mostly shown under the marker for the estimated landmarks).*
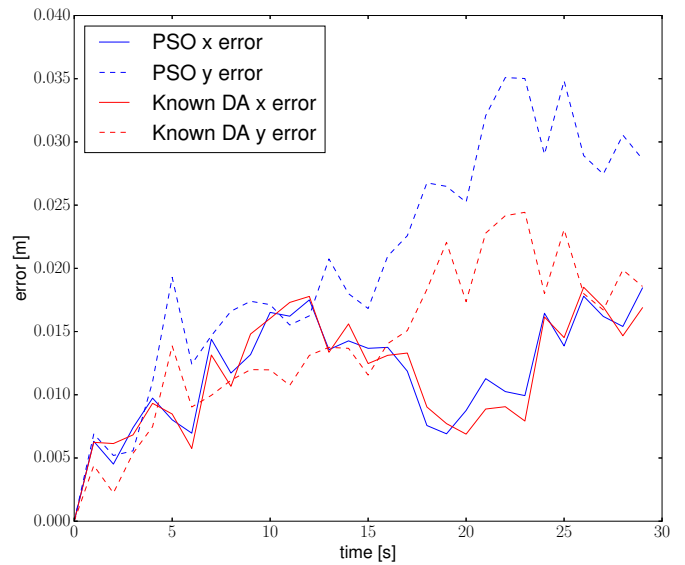

Fig. 6. *The absolute robot position error of the maximum likelihood solution obtained through the traditional vector based, least squares, approach plotted in red, and the RFS based maximum likelihood solution plotted in blue. Errors were averaged over 5 Monte Carlo runs. The errors in the x variable are plotted as continuous lines and the errors in the y variable are plotted as dashed lines. Both error curves are similar.*

VII. Summary

In this article, an RFS based maximum likelihood solution to SLAM has been introduced. The likelihood function to maximize was presented using RFS theory and the PSO algorithm was adapted to this problem and used together with gradient based methods. The optimization of this function was compared to the use of non-linear least squares solvers on the traditional likelihood function which assumes known data association and no clutter, using 1D and 2D simulated SLAM datasets. It was shown that the optimization of this new function can converge to the solution obtained with known data association, even though no such information was provided to the ML-RFS-SLAM algorithm. The ability to perform robust SLAM, without the necessity of fragile data association decisions, opens avenues for significant future research. Algorithms which maximize the RFS measurement likelihood and simultaneously offer computationally tractable solutions could vastly increase the robustness of SLAM.

Future work will address the computational complexity of the optimization method, so that the RFS maximum likelihood SLAM concept can be successfully applied to larger real world datasets.

Acknowledgements

References

[1] K. Y. K. Leung, F. Inostroza, and M. Adams, "Relating random vector and random finite set estimation in navigation, mapping, and tracking," *IEEE Transactions on Signal Processing*, vol. 65, no. 17, pp. 4609–4623, Sept 2017.

[2] C. S. Lee, D. Clark, and J. Salvi, "SLAM with dynamic targets via single-cluster PHD filtering," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 7, no. 3, pp. 543–552, 2013.

[3] C. Evers and P. A. Naylor, "Optimized self-localization for slam in dynamic scenes using probability hypothesis density filters," *IEEE Transactions on Signal Processing*, vol. 66, no. 4, pp. 863–878, Feb 2018.

[4] R. P. S. Mahler, *Statistical multisource-multitarget information fusion*. Artech House Boston, 2007.

[5] B.-T. Vo, B.-N. Vo, and A. Cantoni, "The cardinality balanced multi-target multi-Bernoulli filter and its implementations," *Signal Processing, IEEE Transactions on*, vol. 57, no. 2, pp. 409–423, 2009.

[6] S. Reuter, B.-T. Vo, B.-N. Vo, and K. Dietmayer, "The labeled multi-Bernoulli filter," *Signal Processing, IEEE Transactions on*, vol. 62, no. 12, pp. 3246–3260, June 2014.

[7] H. Deusch, S. Reuter, and K. Dietmayer, "The labeled multi-Bernoulli SLAM filter," *Signal Processing Letters, IEEE*, vol. 22, no. 10, pp. 1561–1565, Oct 2015.

[8] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping using the bayes tree," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[9] H. Strasdat, J. Montiel, and A. Davison, "Visual SLAM: Why filter?" *Image and Vision Computing*, vol. 30, no. 2, pp. 65–77, 2012.

[10] S. Thrun and M. Montemerlo, "The graphSLAM algorithm with applications to large-scale mapping of urban structures," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 403–429, 2006.

[11] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.

[12] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[13] M. Avriel, *Nonlinear programming: analysis and methods*. Courier Corporation, 2003.

[14] K. Leung, F. Inostroza, and M. Adams, "Generalizing random-vector SLAM with random finite sets," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2015, pp. 4583–4588.

[15] B. K. Panigrahi, Y. Shi, and M.-H. Lim, *Handbook of swarm intelligence: concepts, principles and applications*. Springer Science & Business Media, 2011, vol. 8.

[16] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *Signal Processing, IEEE Transactions on*, vol. 56, no. 8, pp. 3447–3457, 2008.