A Reciprocal Sampling Algorithm for Lightweight Distributed Multi-Robot Localization

Amanda Prorok and Alcherio Martinoli

***Abstract*— This work is situated in the context of collaboratively solving the localization problem for unknown initial conditions. We address this problem with a novel, fully decentralized, real-time particle filter algorithm, designed to accommodate realistic robotic assumptions including noisy sensors, and asynchronous and lossy communication. In particular, we introduce a collaborative reciprocal sampling algorithm which allows a drastic reduction in the number of particles needed to achieve localization. We elaborate an analysis of our reciprocal sampling method and support our conclusions with simulation results. Finally, we validate our approach on a team of four real robots within a controlled experimental setup.**

I. Introduction

Accurate position localization is an enabling technology, with a large body of publications manifesting its significance for the mobile robotics domain. In this paper, we consider the problem of absolute localization of a team of mobile robots for unknown initial pose estimates (i.e., *global* localization). We design an algorithm targeting miniaturized, computationally limited platforms equipped with noisy, low-power sensing modalities. Given its efficiency in solving localization problems for unknown initial conditions, and for accommodating arbitrary probability density functions, our method of choice is the particle filter, building on the probabilistic framework of Monte-Carlo Localization (MCL) presented in [2]. Our collaboration strategy uses associated range and bearing observations and inter-robot communication. We develop a range and bearing robot detection model which we introduce into our localization formalism. In particular, we implement an efficient sampling method based on reciprocal robot observations. Jointly with our detection model, this allows us to drastically reduce the number of particles needed to localize, without compromising the performance of the algorithm. In order to assess the impact of our collaborative localization strategy independently from any additional feature-based localization information, we run the algorithm (in real-time) on a team of robots by using only odometry measurements and by sharing only their relative range and bearing observations. Finally, we ensure the scalability of our approach by presenting a localization strategy that in practice has a computational cost which is constant with respect to the number of robots N.

Amanda Prorok and Alcherio Martinoli are with the Distributed Intelligent Systems and Algorithms Laboratory, School of Architecture, Civil and Environmental Engineering at the Ecole Polytechnique Fédérale de Lausanne. The work presented in this paper was supported by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 51NF40-111400. firstname.lastname@epfl.ch

A. Related Work

The problem of collaborative multi-robot localization was first addressed by Kurazume et al. [3], and was followed by a number of subsequent studies. In an early work, Roumeliotis et al. [10] enable the distribution of a Kalman estimation scheme. Yet, as covariance matrix updates occur during each update step and require information exchange between all robots and a centralized processor, the method is particularly vulnerable to single-point failures, and assumes a communication infrastructure without any packet loss. The method scales in $O(N^3)$ with respect to the number of robots, and thus limits its scalability due the high computational cost. Martinelli et al. [4] propose an extension to [10], by generalizing the formalism, but without further improving the algorithm's scalability and cost. Nerurkar et al. [7] address the reduction of computational complexity and single-point failures by implementing a maximum a posteriori estimation method. Nevertheless, the $O(N^2)$ computational cost is significant. Also, the proposed method requires synchronous communication among the robots, and its feasibility still remains to be validated on real robots. Mourikis et al. [5] consider the problem of resource-constrained collaborative localization by limiting the number of measurements processed at each time step, with the goal of deriving optimal sensing frequencies. Yet, as exteroceptive data is dealt with in a centralized way, the sensing frequencies inevitably decrease with an increasing number of robots, thus limiting the scalability of the approach. Finally, a decentralized localization algorithm based on an extended information filter presented in [1] tries to alleviate the problems described above. However, its computational cost increases for every new observation made, and it assumes bidirectional synchronous communication, the feasibility of which remains to be evaluated on real robots.

We note that all above mentioned approaches assume Gaussian noise models and known initial positions for all robots in the team (i.e., *local* localization). Fox et al. [2] first introduce a multi-robot Monte-Carlo localization algorithm for *global* localization, which relaxes noise assumptions as well as inter-robot dependencies. They propose a method with which robots mutually synchronize their position beliefs upon detection, and show successful localization. However, the method proposed has limited scalability due to over-confidence (particle collapse) occurring upon multiple robot detections. Simultaneously, large particle sets are required to avoid particle depletion, ultimately driving up the computational requirements.

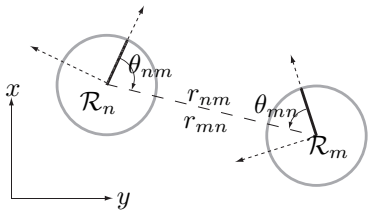


Fig. 1. System of two robots ($\mathcal{R}_n$ and $\mathcal{R}_m$) sharing a common localization frame. The figure illustrates the robots' relative range (r_{nm} and r_{mn}) and bearing (θ_{nm} and θ_{mn}) values.

B. Problem Formulation

Our problem is described as follows. We have a multi-robot system of N robots $\mathcal{R}_1, \mathcal{R}_2, ...\mathcal{R}_N$, where the number N does not need to be known by the robots. The robots navigate in a bounded space. For a robot $\mathcal{R}_n$, at time t, the pose $\mathbf{x}_{n,t}$ is given by the Cartesian coordinates $x_{n,t}, y_{n,t}$ and orientation $\phi_{n,t}$. Also, at time t, a robot $\mathcal{R}_m$ is in the set of neighbors $\mathcal{N}_{n,t}$ of robot $\mathcal{R}_n$ if robot $\mathcal{R}_m$ can determine a range $r_{mn,t}$ and bearing $\theta_{mn,t}$ to robot $\mathcal{R}_n$. Thus, at every moment in time, the neighborhood topology is defined by the physical characteristics of the relative observation sensors deployed on the robots. We make the assumption that a robot $\mathcal{R}_m$ can communicate with a robot $\mathcal{R}_n$, if $\mathcal{R}_m \in \mathcal{N}_{n,t}$. Apart from a sensing modality which enables the robots to determine relative range and bearing, they are also equipped with a dead-reckoning self-localization module (e.g., odometry), but do not possess any exteroceptive sensors capable of accurate feature recognition, such as laser-range finders or cameras. Given these specificities, the goal is to localize all robots, without any prior knowledge of the initial state or previous measurements.

II. Collaborative Monte-Carlo Localization

In this section, we briefly review Monte-Carlo Localization [12] (MCL) as it forms the baseline for our work. We then extend the standard MCL formalism to a fully decentralized, collaborative adaptation to match our problem formulation (see Section I-B).

A. Preliminaries

Let us from hereon consider a robot $\mathcal{R}_n$. At time t, after a sequence of motion control actions $u_{n,t}$ and a sequence of observations $z_{n,t}$ the recursive update equation of the Bayes filter is denoted

$$\mathbf{Bel}(\mathbf{x}_{n,t}) = \eta\, p(z_{n,t}|\mathbf{x}_{n,t}) \int p(\mathbf{x}_{n,t}|\mathbf{x}_{n,t-1}, u_{n,t-1}) \mathbf{Bel}(\mathbf{x}_{n,t-1})\, \mathrm{d}\mathbf{x}_{n,t-1} \quad (1)$$

where $\mathbf{Bel}(\mathbf{x}_{n,t})$ estimates of the posterior state $\mathbf{x}_{n,t}$ and is called a *belief*. The value η is a normalization constant, $p(z_{n,t}|\mathbf{x}_{n,t})$ is the measurement model, and $p(\mathbf{x}_{n,t}|\mathbf{x}_{n,t-1}, u_{n,t-1})$ the motion model.

The main idea of MCL lies in the way the belief is represented—samples, or particles, are drawn from the posterior probability distribution of the robot pose to form a set of particles. By weighting these particles one obtains a discrete probability function that approximates the continuous belief $\mathbf{Bel}(\mathbf{x}_{n,t})$, and hence we have

$$\mathbf{Bel}(\mathbf{x}_{n,t}) \sim \{\langle \mathbf{x}_{n,t}^{[i]}, w_{n,t}^{[i]} \rangle | i = 1, ..., M\} = X_{n,t} \quad (2)$$

where M is the number of particles, $\mathbf{x}_{n,t}^{[i]}$ is a sample of the random variable $\mathbf{x}_{n,t}$ (the pose), and $w_{n,t}^{[i]}$ is its weight. The symbol $X_{n,t}$ refers to the set of particles $\langle \mathbf{x}_{n,t}^{[i]}, w_{n,t}^{[i]} \rangle$ at time t belonging to robot $\mathcal{R}_n$. In contrast to other methods (for example Kalman filtering), the advantage of this form of representation is that it can approximate probability densities of any shape. Given this flexibility, MCL is also able to accommodate arbitrary sensor characteristics and noise distributions.

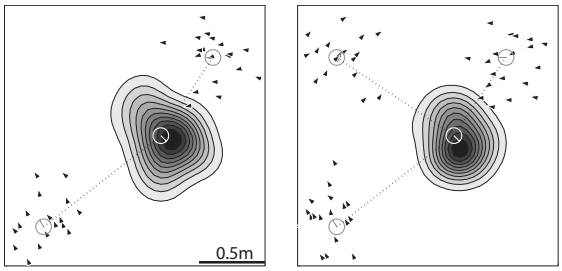
B. Multi-Robot MCL

The framework presented in Section II-A takes into account a single robot. However, when operating a collaborative multi-robot system, the baseline formalism must be adapted to integrate measurements taken on different platforms [2]. If we make the assumption that individual robot poses are independent, we can formulate the event that robot $\mathcal{R}_n$ is detected by robot $\mathcal{R}_m$ as

$$\mathbf{Bel}(\mathbf{x}_{n,t}) = p(\mathbf{x}_{n,t}|z_{n,0..t}, u_{n,0..t}) \int p(\mathbf{x}_{n,t}|\mathbf{x}_{m,t}, r_{mn,t}, \theta_{mn,t}) \mathbf{Bel}(\mathbf{x}_{m,t})\, \mathrm{d}\mathbf{x}_{m,t} \quad (3)$$

where $p(\mathbf{x}_{n,t}|z_{n,0..t}, u_{n,0..t})$ describes the n^{th} robot's current belief, and $\int p(\mathbf{x}_{n,t}|\mathbf{x}_{m,t}, r_{mn,t}, \theta_{mn,t})\ \mathbf{Bel}(\mathbf{x}_{m,t})\, \mathrm{d}\mathbf{x}_{m,t}$ describes the m^{th} robot's belief about the position of robot $\mathcal{R}_n$. For such a collaboration to take place, robot $\mathcal{R}_m$ needs to communicate $r_{mn,t}, \theta_{mn,t}$ and $\mathbf{Bel}(\mathbf{x}_{m,t})$ to robot $\mathcal{R}_n$. Thus a communication message is composed as $d_{mn,t} = \langle r_{mn,t}, \theta_{mn,t}, X_{m,t} \rangle$. If several robots in a neighborhood $\mathcal{N}_{n,t}$ communicate with robot $\mathcal{R}_n$, the received information is the set of all relative observations of robot $\mathcal{R}_n$ at time t, as well as the belief representations $X_{m,t}$ of all detecting robots $\mathcal{R}_m \in \mathcal{N}_{n,t}$. We denote this data set as $D_{n,t} = \{d_{mn,t} | \mathcal{R}_m \in \mathcal{N}_{n,t}\}$. We note that the collaborative aspect of this formalism lies in the integration of robot $\mathcal{R}_m$'s belief into that of robot $\mathcal{R}_n$. This update step is shown in Algorithm 3 (line 5).

As previously discussed in [2], there are certain limitations to this approach. Due to the fact that robot $\mathcal{R}_m$ integrates its position belief into that of robot $\mathcal{R}_n$ upon detection, subsequent detections would induce multiple integrations of this belief, ultimately leading to an overconfident (and possibly erroneous) belief of the actual pose. Fox et al. remedy this shortcoming by considering two rules: *(i)* their approach does not consider negative sights (no detection) of other robots, and *(ii)* they define a minimum travel distance which a robot has to complete before detecting a same robot again. Although rule *(i)* is a practical consideration, rule *(ii)* strongly limits the scalability and robustness of the approach. The frequency of potentially useful information needs to be artificially bounded. Also, it depends on the mobility of a given team member, and does not hold, for instance, for a hybrid network of partially static and partially mobile robotic nodes.



(a) (b)

Fig. 2. Detection model for multiple detecting robots, (a) for two robots and (b) for three robots. Here, a set of 20 particles is shown, represented by oriented triangles. The detected robot is shown in white. The model's probability density is superimposed on the detected robot. The dotted line and the orientation of the robots show the actual relative range and bearing.

III. Range & Bearing Detection Model

The idea of the detection model is to propose a probability density function which is based on the relative observations made by the detection sensors, and which is also based on the belief of the detecting robot. Here, for the purpose of our case-study, we use a simple Gaussian distribution in polar coordinates, but all reasonings are valid for completely arbitrary distributions. Indeed, as we use a particle filter, we can keep the same framework for any possible sensor model and possible underlying range and bearing hardware not fulfilling the Gaussian assumption. For brevity, we omit the subscript t in the following derivations.

As pointed out in Section II-B, when a robot $\mathcal{R}_m$ detects a robot $\mathcal{R}_n$ it sends its detection data d_{mn}. We will now formulate the detection model as $\mathrm{P}_{mn}(\mathbf{x}_n|d_{mn})$ which describes the probability that robot $\mathcal{R}_m$ detects robot $\mathcal{R}_n$ at pose $\mathbf{x}_n = [x_n\ y_n\ \phi_n]$, given the detection data d_{mn}. For a given particle i in robot $\mathcal{R}_m$'s belief, we define the range difference Δr_{mn}, and the bearing difference $\Delta\theta_{mn}$. The range and bearing differences are given by the geometric relations

$$\Delta r_{mn} = \sqrt{\Delta x_{mn}^2 + \Delta y_{mn}^2} - r_{mn} \quad (4)$$

$$\Delta\theta_{mn} = \mathrm{atan2}(\Delta y_{mn}, \Delta x_{mn}) - (\phi_m^{[i]} + \theta_{mn}) \quad (5)$$

where we denote $\Delta x_{mn} = (x_m^{[i]} - x_n)$ and $\Delta y_{mn} = (y_m^{[i]} - y_n)$. Assuming Gaussian noise and knowledge of the range and bearing standard deviation (σ_r and σ_θ, respectively), and the independence of range and bearing measurements, the detection probability is

$$\mathrm{P}_{mn}(\mathbf{x}_n|d_{mn}) = \eta \cdot \sum_{\left\langle \substack{\mathbf{x}_m^{[i]} \\ w_m^{[i]} \end{smallmatrix} \right\rangle \in X_m} \Phi\left(\begin{bmatrix} \Delta r_{mn} \\ \Delta\theta_{mn} \end{bmatrix}, \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} \right) \cdot w_m^{[i]} \quad (6)$$

where $\Phi(\cdot, S)$ is the zero-mean, multivariate normal probability density function with the covariance matrix S and where η is a normalization constant. Also, in the case where robot $\mathcal{R}_n$ reciprocally detects robot $\mathcal{R}_m$, it can use the additional information of its own relative observations to determine the orientation difference $\Delta\phi_{mn}$, which is defined by the following geometric relation

$$\Delta\phi_{mn} = \pi - \phi_m^{[i]} - \phi_n + \theta_{mn} - \theta_{nm}. \quad (7)$$

The detection probability is then augmented by an additional component, resulting in

$$\mathrm{P}_{mn}(\mathbf{x}_n|d_{mn}) = \eta \cdot \sum_{\left\langle \substack{\mathbf{x}_m^{[i]} \\ w_m^{[i]} \end{smallmatrix} \right\rangle \in X_m} \Phi\left(\begin{bmatrix} \Delta r_{mn} \\ \Delta\theta_{mn} \\ \Delta\phi_{mn} \end{bmatrix}, \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_\theta^2 & 0 \\ 0 & 0 & 4\sigma_\theta^2 \end{bmatrix} \right) \cdot w_m^{[i]} \quad (8)$$

Finally, the detection model incorporating the detection data from multiple detecting robots can be formulated as an update equation as shown in Algorithm 1.

Algorithm 1 `Detection_Model`$(D_{n,t}, \mathbf{x}_t^{[i]}, w_t^{[i]})$

1: $w \leftarrow w_t^{[i]} \cdot \prod_{d_{mn} \in D_{n,t}} \mathrm{P}_{mn}(\mathbf{x}_t^{[i]}|d_{mn})$
2: return w

Figure 2 shows an illustration of the probability density function resulting from the detection model, (a) for two detecting robots, and (b) for three detecting robots. We notice that when detection data from multiple robots is integrated into the range and bearing model, the detection precision increases.

IV. Multi-Robot Reciprocal MCL

In this section, we present a novel approach to multi-robot MCL. Motivated by the goal of overcoming the limitations of current multi-robot localization algorithms, which to-date are hard to employ on large-scale, distributed systems for unknown initial conditions, we develop an any-time, fully scalable localization algorithm which takes advantage of reciprocal robot observations to reduce the number of particles needed while maintaining good performance.

A. Concept of Reciprocal Sampling

In addition to using a robot detection model for updating the belief representation $\mathbf{Bel}(\mathbf{x}_{n,t})$, our approach relies on a *reciprocal* sampling method. As for a standard MCL algorithm, the posterior estimate of reciprocal MCL is represented by $\mathbf{Bel}(\mathbf{x}_{n,t})$—the difference between the two methods lies in the proposal distribution. Let us refer to the iterative process described in Algorithm 3: instead of sampling from $\mathbf{Bel}(\mathbf{x}_{n,t-1})$ in line 11, the reciprocal MCL algorithm samples from the distribution $\mathbf{x}_{n,t}^{[i]} \sim p(D_{n,t}|\mathbf{x}_{n,t}^{[i]})$, according to a robot detection model. Thus, samples are drawn at poses which are probable given the reciprocal robot observations, and which are independent of the previous belief $\mathbf{Bel}(\mathbf{x}_{n,t-1})$. Then, by employing the reciprocal sampling algorithm within the collaborative paradigm of our general framework, a detected robot augments its own belief with new pose estimates deduced from reciprocal robot observations with a fixed proportion of α. In particular, as this method exploits the information available in a whole robot team, it continuously creates particles in areas of the pose space which are likely to be significant, and allows for very small particle set sizes.

The idea of extending standard MCL with additional sampling methods was first shown in [11]. In this previous work, the resulting algorithm named *Mixture* MCL was shown to increase the robustness of single-robot global localization. Our method differs from the one presented in [11] in that it extends to collaborative multi-robot localization algorithms by sampling from the detection model of one or several mobile robots (whose positions are unknown) as opposed to sampling from the detection model of a potentially large set of static environmental features (whose positions are known). Indeed, for complex environments, the method in [11] must be preceded by a cumbersome fingerprinting process.

B. Reciprocal Sampling Algorithm

The reciprocal sampling routine is shown in Algorithm 2, where line 4 represents the reciprocal sampling step. Algorithm 3 shows the complete routine of multi-robot reciprocal MCL. The second part (lines 8–16) resamples particles from the weighted proposal distribution to create a new, updated pose belief. Particles are sampled from the robot's own belief with a probability $1-\alpha$, and with a probability of α, particles are sampled from the probability density function proposed by the detection model (line 13). There are a multitude of methods which can be applied to sample from a given distribution. Here, we employ the slice sampling method [6], which is a low-cost method based on Markov chains, and particularly useful since it can sample from arbitrary shaped distributions. The symbol $\mathcal{U}$ on line 9 of Algorithm 3 refers to the uniform distribution.

Algorithm 2 `Reciprocal_Sampling`$(D_{n,t}, \bar{X}_{n,t})$

1: **if** $D_{n,t} = \emptyset$ **then**
2: $\quad \mathbf{x} \leftarrow$ `Sampling`$(\bar{X}_{n,t})$
3: **else**
4: $\quad \mathbf{x} \sim \prod_{d_{mn} \in D_{n,t}} \mathrm{P}_{mn}(\mathbf{x}|d_{mn})$
5: **end if**
6: return $\mathbf{x}$

Algorithm 3 `MultiRob_Recip_MCL`$(X_{n,t-1}, u_{n,t}, z_{n,t}, D_{n,t})$

1: $\bar{X}_{n,t} = X_{n,t} = \emptyset$
2: **for** $i = 1$ to M **do**
3: $\quad \mathbf{x}_{n,t}^{[i]} \leftarrow$ `Motion_Model`$(u_{n,t}, \mathbf{x}_{n,t-1}^{[i]})$
4: $\quad w_{n,t}^{[i]} \leftarrow$ `Measurement_Model`$(\mathbf{x}_{n,t}^{[i]})$
5: $\quad w_{n,t}^{[i]} \leftarrow$ `Detection_Model`$(D_{n,t}, \mathbf{x}_{n,t}^{[i]}, w_{n,t}^{[i]})$
6: $\quad \bar{X}_{n,t} \leftarrow \bar{X}_{n,t} + \left\langle \mathbf{x}_{n,t}^{[i]}, w_{n,t}^{[i]} \right\rangle$
7: **end for**
8: **for** $i = 1$ to M **do**
9: $\quad r \sim \mathcal{U}(0,1)$
10: $\quad$ **if** $r \leq (1-\alpha)$ **then**
11: $\quad\quad \mathbf{x}_{n,t}^{[i]} \leftarrow$ `Sampling`$(\bar{X}_{n,t})$
12: $\quad$ **else**
13: $\quad\quad \mathbf{x}_{n,t}^{[i]} \leftarrow$ `Reciprocal_Sampling`$(D_{n,t}, \bar{X}_{n,t})$
14: $\quad$ **end if**
15: $\quad X_{n,t} \leftarrow X_{n,t} + \left\langle \mathbf{x}_{n,t}^{[i]}, w_{n,t}^{[i]} \right\rangle$
16: **end for**
17: return $X_{n,t}$

C. Analysis

It is clear from the previous sections that performance of the reciprocal sampling algorithm depends on the accuracy of the detection sensors. Simultaneously, we need to find an appropriate reciprocal sampling proportion α which enables fast convergence to a low localization error. Thus, for the purpose of analysis, we resort to a minimal scenario consisting of two collaborative robots moving randomly in bounded space. The goal of the exercise is to discuss the localization performance of the first robot $\mathcal{R}_n$, which is initially unlocalized, given two alternative conditions for the second robot $\mathcal{R}_m$: *(1)* $\mathcal{R}_m$ is continuously localized (i.e., receives absolute position fixes at a frequency that is as least as high as the robot detection frequency) and *(2)* $\mathcal{R}_m$ is only initially localized (and then receives no more absolute position fixes). We note that our evaluations throughout this paper do not consider the time it takes for the first robot to localize, as this is part of a separate process and independent from our algorithm.

In this analysis, we will show that the reciprocal sampling proportion α can be tuned to affect the steady-state performance, and how the convergence speed is affected when employing a finite number of particles. We derive the formulations for both the standard sampling algorithm (Algorithm 3 with $\alpha = 0$), as well as our reciprocal sampling algorithm (Algorithm 3 with $\alpha > 0$). To simplify the following formalisms, without loss of generality, we consider that the origin of the coordinate system coincides with the true position of robot $\mathcal{R}_n$. Thus, we assume that the state of any robot can be expressed in a 1-dimensional space as a position ρ, bounded by $-\rho_{\max} \leq \rho \leq \rho_{\max}$. The belief of robot $\mathcal{R}_n$ at time t is thus simply given by $\mathrm{Bel}(\rho_{n,t})$. In consequence, we formulate our error metric as the expectancy over all possible estimated distances $\rho_{n,t}$ of robot $\mathcal{R}_n$ to the origin

$$\mathbb{E}(\rho_{n,t}^2) = \int \mathbf{Bel}(\rho_{n,t}) \cdot \rho_{n,t}^2 \mathrm{d}\rho_n. \tag{9}$$

The following two paragraphs detail $\mathbf{Bel}(\rho_{n,t})$ as well as $\mathbb{E}(\rho_{n,t}^2)$ for $t \to \infty$, for both the standard sampling algorithm (SS) and the reciprocal sampling algorithm (RS), in cases *(1)* and *(2)*.

1) $\mathcal{R}_m$ **continuously localized:** We initialize $\mathbf{Bel}_{n,t=0} = 1/(2\rho_{max})$ as a uniform distribution over ρ, and $\mathbf{Bel}_{m,t} = \delta(\rho_m)$, for all t, where ρ_m is the true position of robot $\mathcal{R}_m$ and $\delta(\cdot)$ the Dirac function.

Let us assume that the observation and motion models can be modeled with $\Phi(\cdot, \sigma)$ a the zero-mean, normal probability density function with the standard deviation σ_{mn} and σ_n, respectively. The belief $\mathbf{Bel}(\rho_{n,t})$ of robot $\mathcal{R}_n$ at time t for the standard sampling algorithm then reads

$$\begin{aligned} \mathbf{Bel}_{\mathrm{SS}}(\rho_{n,t}) &= \Phi(\rho_{n,t}, \sigma_{mn}) \\ &\int \Phi(\rho_{n,t} - \rho_{n,t-1}, \sigma_n) \cdot \mathbf{Bel}_{\mathrm{SS}}(\rho_{n,t-1}) \mathrm{d}\rho_{t-1}. \end{aligned} \tag{10}$$

Given that the probability density of Equation 10 is normal, we can easily calculate the steady-state error. For $t \to \infty$, Equation 9 is

$$\mathbb{E}_{SS}(\rho_{n,t}^2) \underset{t\to\infty}{=} \frac{\sigma_n}{2}\left(\sqrt{\sigma_n^2 + 4\sigma_{mn}^2} - \sigma_n^2\right). \tag{11}$$

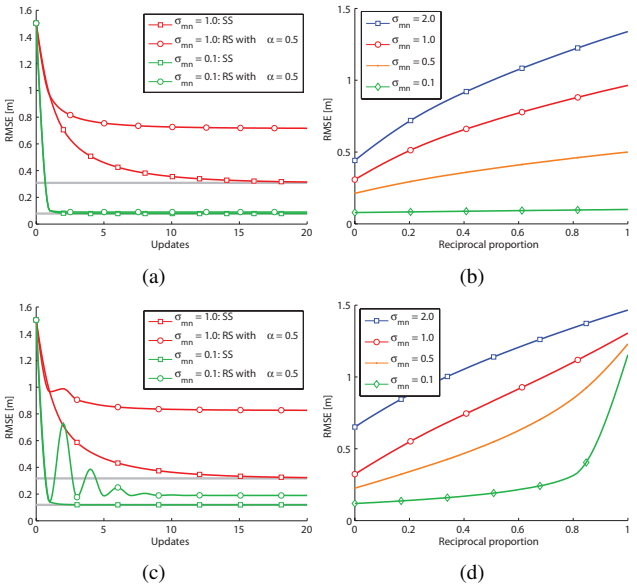


Fig. 3. The plots show in (a) and (c) the development of the the root-mean-square-error for the standard sampling (SS) and reciprocal sampling (RS) algorithms, and in (b) and (d) the steady-state error in function of the reciprocal sampling proportion. The first row shows results for robot $\mathcal{R}_m$ continuously localized, and the second row for robot $\mathcal{R}_m$ only initially localized. The space was defined by $\rho_{max} = 2.6$m and a constant motion noise of $\sigma_n = 0.1$ was employed.

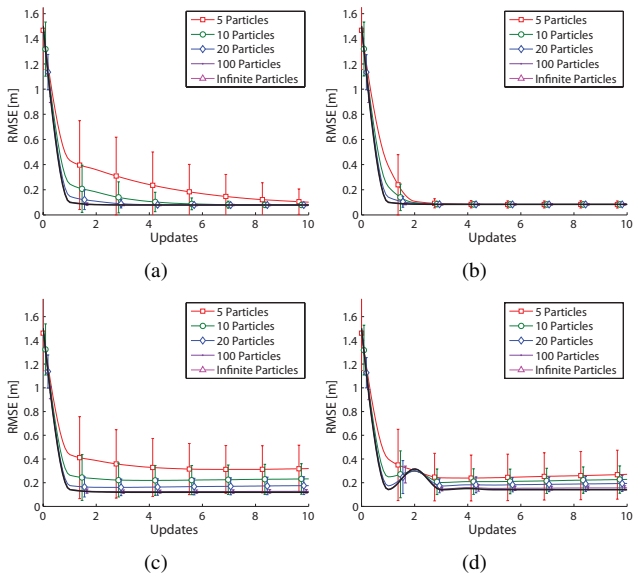


Fig. 4. The figures show the root-mean-square-error for (a), (c) standard sampling and (b), (d) reciprocal sampling, in a space defined by $\rho_{max} = 2.6$m. A variable number of particles was employed, ranging from a total of 5 to 100 particles. Additionally, the solid black line shows the results for an infinity of particles. We employed a motion noise $\sigma_n = 0.1$ and detection noise $\sigma_{mn} = 0.1$ for all plots, and a reciprocal sampling proportion of $\alpha = 0.2$ for plots (b) and (d). The first row shows results for robot $\mathcal{R}_m$ continuously localized, the second row for robot $\mathcal{R}_m$ only initially localized. For each experiment employing a finite set of particles, 5000 runs were performed. The errorbars show the standard deviation.

We now extend this formalism to the case of reciprocal sampling with a reciprocal sampling proportion α. We have

$$\mathbf{Bel}_{\mathrm{RS}}(\rho_{n,t}) = \alpha \cdot \Phi(\rho_{n,t}, \sigma_{mn}) + (1-\alpha) \cdot \Phi(\rho_{n,t}, \sigma_{mn}) \int \Phi(\rho_{n,t} - \rho_{n,t-1}, \sigma_n) \cdot \mathbf{Bel}_{\mathrm{RS}}(\rho_{n,t-1}) \mathrm{d}\rho_{n,t-1}. \quad (12)$$

The steady-state error of Equation 12 will vary, depending on the reciprocal proportion α. For this work, it suffices to consider the maximum possible steady state error, which is simply given by σ_{mn}^2 (for $\alpha = 1$).

2) $\mathcal{R}_m$ **only initially localized:** As before, we initialize $\mathbf{Bel}_{n,t=0} = 1/(2\rho_{max})$, and $\mathbf{Bel}_{m,t=0} = \delta(\rho_m)$. As robot $\mathcal{R}_m$ does not receive regular position fixes anymore, we must now reformulate the equations above to include the reciprocal sampling mechanism for both robots simultaneously. Without loss of generality, we assume that the moment of information exchange takes place at each update step after application of the motion model when the prior $\overline{\mathbf{Bel}}(\rho_{\cdot,t})$ has been calculated. Thus, when a robot $\mathcal{R}_m$ detects a robot $\mathcal{R}_n$, it transmits its prior $\overline{\mathbf{Bel}}(\rho_{m,t})$, defined as

$$\overline{\mathbf{Bel}}(\rho_{m,t}) = \int \Phi(\rho_{m,t} - \rho_{m,t-1}, \sigma_m) \cdot \mathbf{Bel}(\rho_{m,t-1}) \mathrm{d}\rho_{m,t-1}. \quad (13)$$

The belief of robot $\mathcal{R}_n$ for the standard sampling algorithm is then

$$\mathbf{Bel}_{\mathrm{SS}}(\rho_{n,t}) = \int \Phi(\rho_{n,t}, \sigma_{mn}) \cdot \overline{\mathbf{Bel}}_{SS}(\rho_{m,t}) \mathrm{d}\rho_{m,t} \cdot \overline{\mathbf{Bel}}_{SS}(\rho_{n,t}). \quad (14)$$

As in Equation 11, we formulate the steady state error of Equation 14 as

$$\mathbb{E}_{SS}(\rho_{n,t}^2) \underset{t\to\infty}{=} \sigma_n \sqrt{\sigma_{mn}^2 + \sigma_n^2}. \quad (15)$$

Analogously, we extend the formalism for the reciprocal sampling algorithm

$$\mathbf{Bel}_{\mathrm{RS}}(\rho_{n,t}) = \alpha \cdot \int \Phi(\rho_{n,t}, \sigma_{mn}) \cdot \overline{\mathbf{Bel}}_{RS}(\rho_{m,t}) \mathrm{d}\rho_{m,t} + (1-\alpha) \cdot \int \Phi(\rho_{n,t}, \sigma_{mn}) \cdot \overline{\mathbf{Bel}}_{SS}(\rho_{m,t}) \mathrm{d}\rho_{m,t} \cdot \overline{\mathbf{Bel}}_{RS}(\rho_{n,t}). \quad (16)$$

In this particular case, the steady-state cannot be found analytically. Yet, we note that the the maximal steady-state error of Equation 16 for $\alpha = 1$ is unbounded for an unbounded space. Equations 13–16 are equally formulated for robot $\mathcal{R}_m$.

D. Discussion

In order to analyze the performance of the filters in Equations 10, 12, 14 and 16, we resort to a numerical solution for each update t, and discuss the localization performance of robot $\mathcal{R}_n$.

Figure 3 shows the performance for standard and reciprocal sampling algorithms in an ideal filter (with an infinity of particles). A gray line marks the lower-bound error derived from Equations 11 and 15. The steady-state bounds in Fig. 3(b) vary between the steady-state errors defined by Equation 11 and σ_{mn}, and provide a lower bound on the steady-state performance in Fig. 3(d). With the current analysis discussing an ideal filter with an infinity of particles, we cannot expect any benefit from the reciprocal sampling algorithm.

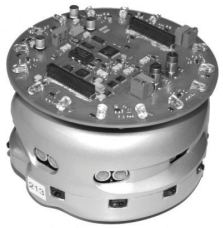
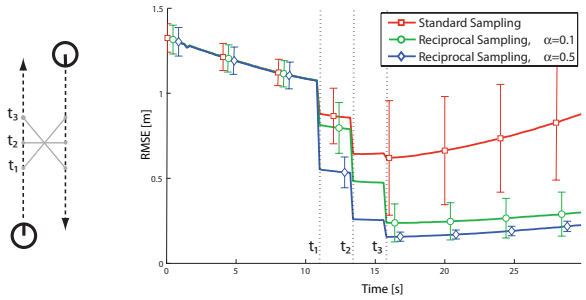


Fig. 5. The Khepera III robot with a relative range and bearing module. The board is composed of a ring of 16 infrared LEDs.



(a) (b)

Fig. 6. (a) Schematic illustration of two robots driving past each other. Three detections are made. (b) Localization error for an initially unlocalized robot. It detects a localized robot three times along its path. The standard and reciprocal sampling algorithms (employing 50 particles) are tested 1000 times on the data set. The times at which the observations are made are marked by dotted lines (11.2s, 13.6s, 16s).

However, the results show that in particular for moderate observation noise values σ_{mn}, the loss of accuracy is very small, regardless of the reciprocal sampling proportion α.

Figure 4 shows the localization error for both standard (first column) and reciprocal sampling (second column) for a variable set of finite particle numbers. We see that for an increasing number of particles, the localization error converges to that of the ideal filter (with an infinity of particles). In contrast to Figure 3, we observe how, in the case of a finite number of particles, the reciprocal sampling algorithm converges faster to the steady-state error. Even when employing as little as 5 particles in total, 1 reciprocal particle is enough to accelerate convergence nearly 10-fold.

Indeed, for an infinity of particles, the standard sampling algorithm will always outperform reciprocal sampling. Yet, for any finite number of particles, the reciprocal sampling algorithm is highly likely to accelerate convergence. Moreover, we note that for modest noise values σ_{mn}, we are able to benefit from the reciprocal sampling algorithm, regardless of the sampling proportion α, without significantly affecting the steady-state performance of the system. In conclusion, we note that while σ_{mn} and σ_n are often given by the real system at hand, parameters such as α and the number of particles can be tuned and thus adapted to available computational and communication resources.

V. Experiments

We validate our proposed approach by performing experiments on a team of Khepera III robots[1] [8]. Our real experimental setup consists of a 3m large empty square arena. In order to measure the ground truth, we installed an overhead camera system as detailed in our previous work [8]. For all experiments, the robots move straight at a speed of one robot-size per second (12cm/s), and perform standard Braitenberg obstacle avoidance. The robots are equipped with wheel encoders and use odometry for self-localization. We note that our measurement model routine (line 4 in Algorithm 3) simply reduces the particles' weights as they leave the bounded space, and does not take into account any exteroceptive sensor readings. The robots use a relative range and bearing module [9], which provides the measures used by the detection model. Figure 5 shows the sixteen evenly-spaced infrared Light Emitting Diodes (LEDs) that this platform uses. In our experimental arena, the boards have a proportional range noise of $\sigma_r = 0.15 \cdot r_{mn}$, a bearing noise of $\sigma_\theta = 0.15$rad, and the detection range is bounded at 2.5m. We discuss the localization performance in terms of the mean positioning error of all particles in the robots' beliefs.

[1]http://www.k-team.com/

A. Two Robots Meet

We illustrate the effect of reciprocal robot detections by performing a short experiment involving two Khepera III robots, one of which is initially localized. The sensor data is gathered and then tested offline on the standard sampling algorithm as well as on the reciprocal sampling algorithm with 50 particles. Figure 6 shows the localization error for the second, initially unlocalized, robot. In comparison to the standard sampling algorithm, we see that the reciprocal sampling algorithm reduces the localization error by taking better advantage of information available on the localized team-member. Additionally, in this case where the first robot is well localized during this short time span, an increased reciprocal sampling proportion α is more efficient due to the higher probability of drawing accurate reciprocal samples.

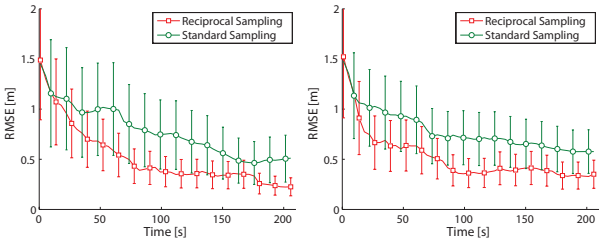
B. Four Robots Meet

To give the readers a feel for the behavior of the reciprocal sampling algorithm, we performed an experiment employing a team of four Khepera III robots. The robots are placed randomly inside the arena at the beginning of the experiment, and sensor data is subsequently gathered throughout the run. Figure 7 shows eight snapshots of the localization process for one robot initially localized. In the 35s time-span, an average of 10 reciprocal robot observations were made per robot. The snapshots show that, even though the robots employ a very sparse particle set (20 particles per robot over a 9m^2 space), the reciprocal sampling algorithm is able to produce particles which coincide with the approximate true locations of the robots. An additional experiment taking place in a real office space can be viewed at `http://disalw3.epfl.ch/publications/IROS11_demo.pdf`.

C. Systematic Evaluation

We perform a final experiment to confirm the reproducibility of our results. We run our algorithms on-board in real-time on a team of four Khepera III robots. We consider two scenarios, analogous to the two cases presented in IV-C. In a first scenario, one robot continuously receives position fixes at a frequency of 1Hz from the overhead camera system, while the three other robots are unlocalized. In a second scenario, one robot receives a one-time position fix at startup from the overhead camera system, while again, the three other robots are unlocalized. To begin, the robots are placed randomly in the arena. Each robot has a total of 50 particles, and the reciprocal sampling algorithm has is implemented with $\alpha = 0.06$. We repeat the experiment 10 times, where each run lasts 3.5 minutes.

Figure 8 shows the positioning error averaged over the number of runs for the three initially unlocalized robots. An average of 14 reciprocal robot observations were made per robot per run. Throughout the 3.5min time-span, the reciprocal sampling algorithm outperforms the standard sampling algorithm. A larger performance gap between the first (Figure 8(a)) and second (Figure 8(b)) scenario is to be expected for larger time-spans as the robot which is only initially localized loses its position accuracy. This experiment shows a successful application of our approach, also confirming the feasibility of our algorithm with respect to real-time and resource constraints.



(a) (b)

Fig. 8. Average localization estimation error obtained over 10 runs of 3.5min duration, performed in real-time on 4 Khepera III robots with 50 particles each, and a reciprocal sampling proportion of $\alpha = 0.06$. Two scenarios are considered, (a) one continuously localized robot, three initially unlocalized and (b) one initially localized robot, three initially unlocalized. The errorbars show the standard deviations.

VI. Conclusion

In this work, we presented a fully scalable, probabilistic multi-robot localization algorithm. By relaxing communication constraints (any-time, asynchronous), we provided an efficient framework for collaborative localization. In particular, we introduced an intrinsically distributed reciprocal sampling algorithm, designed to enable good localization performance in face of rigid system constraints. Our approach was successfully experimentally validated on four resource-bounded robots, confirming superior performance of our reciprocal sampling algorithm over a standard sampling algorithm. More work needs to be done in order to explore arbitrarily distributed detection models as an extension to our generalizable framework.

References

[1] A. Cristofaro, A. Renzaglia, and A. Martinelli. Distributed information filters for MAV cooperative localization. In *International Symposium on Distributed Autonomous Robotic Systems (DARS)*, 2010, to appear.

[2] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8:325–344, 2000.

[3] R. Kurazume, S. Nagata, and S. Hirose. Cooperative positioning with multiple robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1250–1257, 1994.

[4] A. Martinelli, F. Pont, and R. Siegwart. Multi-robot localization using relative observations. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2797–2802, 2005.

[5] A.I. Mourikis and S.I. Roumeliotis. Optimal sensor scheduling for resource-constrained localization of mobile robot formations. *IEEE Transactions on Robotics*, 22(5):917–931, 2006.

[6] B. Neal. Slice sampling. *The Annals of Statistics*, 31:705–757, 2003.

[7] E.D. Nerurkar, S.I. Roumeliotis, and A. Martinelli. Distributed maximum a posteriori estimation for multi-robot cooperative localization. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1402–1409, 2009.

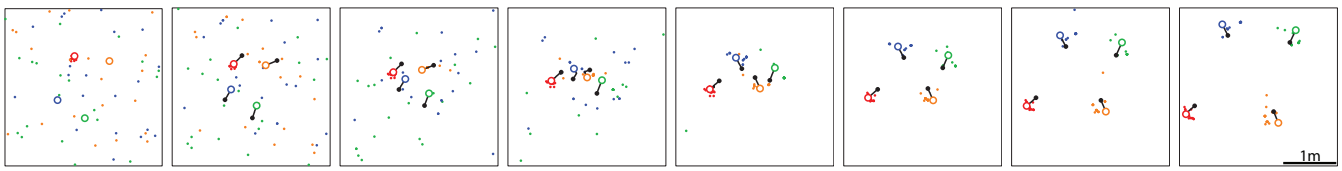
[8] A. Prorok, A. Arfire, A. Bahr, J.R. Farserotu, and A. Martinoli. Indoor navigation research with the Khepera III mobile robot: An experimental baseline with a case-study on ultra-wideband positioning. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2010. doi: 10.1109/IPIN.2010.5647880.

[9] J. Pugh, X. Raemy, C. Favre, R. Falconi, and A. Martinoli. A fast on-board relative positioning module for multi-robot systems. *IEEE Transactions on Mechatronics*, 14(2):151–162, 2009.

[10] Stergios I. Roumeliotis and George A. Bekey. Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 14(5): 781–795, 2002.

[11] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2000.

[12] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005.



(a) t = 0 (b) t = 5s (c) t = 10s (d) t = 15s (e) t = 20s (f) t = 25s (g) t = 30s (h) t = 35s

Fig. 7. The figure shows eight snapshots in 5s intervals of the reciprocal sampling algorithm run on data from an experiment with four real robots. Each robot employed 20 particles, with a reciprocal proportion $\alpha = 0.1$. The black lines show the trajectories completed in the time intervals between snapshots. The red robot was initially localized (and then received no further position fixes).