

# Analysis of Particle Methods for Simultaneous Robot Localization and Mapping and a New Algorithm: Marginal-SLAM

Ruben Martinez-Cantin, Nando de Freitas and José A. Castellanos

**Abstract**—This paper presents a new particle method, with stochastic parameter estimation, to solve the SLAM problem. The underlying algorithm is rooted on a solid probabilistic foundation and is guaranteed to converge asymptotically, unlike many existing popular approaches. Moreover, it is efficient in storage and computation. The new algorithm carries out filtering only in the marginal filtering space, thereby allowing for the recursive computation of low variance estimates of the map. The paper provides mathematical arguments and empirical evidence to substantiate the fact that the new method represents an improvement over the existing particle filtering approaches for SLAM, which work on the joint path state space.

## I. INTRODUCTION

For many years, Simultaneous Localization and Mapping (SLAM) has occupied the center-stage in robotics research [1], [2]. Probabilistic methods, such as Bayes filters, have often been adopted to manage the uncertainty in the sensors and actuators. This sound approach in conjunction with considerable heuristic engineering has produced reasonable solutions; to the point that some researchers have begun voicing the opinion that “*the SLAM problem is solved*”.

While not endorsing the view that SLAM should occupy the center-stage, it is the thesis of this paper that the existing SLAM solutions are built upon questionable assumptions and procedures, including linearity, Gaussian distributions, treating static maps with dynamic models and neglecting the variance increase due to sampling in spaces of increasing (potentially unbounded) dimension. The paper provides some arguments as well as empirical evidence to substantiate this statement. The bottom line is that there is an urgent need for designing a principled SLAM framework so as to relax these assumptions and eliminate the need for brittle heuristics. It is the intent of this paper to take a step in this direction and to note that the problem is still open.

The first probabilistic treatment of SLAM dates back to the seminal work of Smith et al. [3]. Having cast the SLAM problem as one of optimal filtering, a solution was obtained with the extended Kalman filter (EKF-SLAM). This solution, still popular, relies on linearization of the process and measurement functions as well as the assumption that all the involved distributions are Gaussian.

Since then, we have witnessed a large proliferation of EKF-SLAM implementations in indoors, outdoors, underwater and airborne domains. Sophisticated techniques that

exploit sparsity in the information matrix, such as thin junction trees, have allowed for significant increases in the computational efficiency of the method [4]. However, the consistency issues of EKF-SLAM have started attracting the attention of the research community recently, taking priority over faster computation.

It is well known that the EKF linearization can lead to filter divergence [5], [6], [7]. In the last few years, several works have proposed techniques to reduce the linearization effect. Among these techniques are higher order approximations with unscented transformations and changes of coordinates [8], [9], [10]. Though these techniques result in some improvement, the inherent approximations continue to cause accumulation of errors and irreversible filter divergence.

The introduction of particle filters (PFs) gave researchers the power and flexibility to handle nonlinearity and non-Gaussian distributions routinely [11]. Moreover, it enabled researchers to exploit conditional independence, using the Rao-Blackwellized particle filtering (RBPF) variance reduction technique, to obtain more efficient Monte Carlo schemes. RBPFs were applied to dynamic maps [12] and subsequently to static maps with the celebrated FastSLAM algorithm [13]. The application of RBPF to dynamic maps is only sensible inasmuch as one has a good model to describe the evolution of the *dynamic* map. On the other hand, the application of RBPF to static maps has come into question. It has become popular knowledge that the approach can diverge. The inconsistency of FastSLAM was indeed the topic of a recent robotics paper [14]. In loose terms, learning static variables (the map) by conditioning on increasing histories of the state variables results in an accumulation of Monte Carlo errors and explosion of variance. Heuristic approaches to ameliorate the situation [15], [16] have been proposed, but these do not solve the fundamental problem at hand. We note that the problem of PF divergence resulting from learning fixed variables by conditioning on increasing paths was already described in as early as 1999 [17].

This lack of statistical consistency of the most popular SLAM methods has naturally created some justified concern. In this paper, we present Marginal-SLAM, a novel approach with two key ingredients to ensure *consistency*. The first ingredient is to consider SLAM as a robot localization problem with unknown observation model parameters. Thus, the core is to *treat static maps as parameters*, which by necessity are learned using maximum likelihood (ML) or maximum a posteriori inference. The idea of treating maps as parameters is not new. It has been central to the incremental ML method [18]. However, this method resorts to an ML estimate of the state and hence fails to manage the uncertainty in the

R. Martinez-Cantin and J.A. Castellanos are with Department of Computer Science and System Engineering, University of Zaragoza, 50018 Zaragoza, Spain {rmcantin, jacaste}@unizar.es

N. de Freitas is with the Department of Computer Science, University of British Columbia, Vancouver, BC V6T 1Z4, Canada nando@cs.ubc.ca

robot state properly, as elaborated in [2]. A variation based on learning the distribution of the robot states and using an ML estimator of the map as a function of the existing and *growing state trajectories* was proposed in [19]. This approach unfortunately suffers from the same consistency problems as FastSLAM. EM approaches have been proposed [2], but they apply only when learning the map off-line.

The second key ingredient in Marginal-SLAM is being able to compute the filter derivative for the recursive ML. This has only become possible very recently following new advances in particle simulation [20], [21], which will be presented in Section III.B. Static parameter estimation in nonlinear, non-Gaussian dynamic models is a formidable challenge, so it comes as no surprise that probabilistic SLAM has proved to be so demanding. We note that we adopt recursive ML estimates of the map because the problem of doing efficient full-Bayesian recursive parameter estimation has not been solved.

## II. PROBLEM FORMULATION

We present a general formulation of the problem that is applicable to both feature-based maps and grid-based maps. A more specific and detailed model appears in the experimental section. The unknown robot pose (location and heading),  $\mathbf{x}_t \in \mathcal{X}$ , is modelled as a Markov process of initial distribution  $p(\mathbf{x}_1)$  and transition prior  $p(\mathbf{x}_t | \mathbf{x}_{t-1})$ , which represents the motion model<sup>1</sup>. The observations,  $\mathbf{y}_t \in \mathcal{Y}$ , are assumed to be conditionally independent given the process  $\{\mathbf{x}_t\}$  and of marginal distribution  $p_\theta(\mathbf{y}_t | \mathbf{x}_t)$ , where  $\theta$  is a vector describing the elements of the map. Hence, the model consists of the following two distributions:

$$\begin{aligned} p(\mathbf{x}_t | \mathbf{x}_{t-1}) \\ p_\theta(\mathbf{y}_t | \mathbf{x}_t) \quad t \geq 1. \end{aligned}$$

We denote by  $\mathbf{x}_{1:t} \triangleq \{\mathbf{x}_1, \dots, \mathbf{x}_t\}$  and  $\mathbf{y}_{1:t} \triangleq \{\mathbf{y}_1, \dots, \mathbf{y}_t\}$ , respectively, the robot state and the observations up to time  $t$ , and define  $p(\mathbf{x}_1 | \mathbf{x}_0) \triangleq p(\mathbf{x}_1)$  for notational convenience.

In this formulation, we have used  $\theta$  to describe the map only. However,  $\theta$  could be used to include other parameters in the transition and measurement models, as well as, to include data association variables [2]. For simplicity of presentation, we will assume that the associations are given.

Our aim is to compute sequentially in time the *filtering distribution*  $p(\mathbf{x}_t | \mathbf{y}_{1:t})$  and point estimates of the map  $\theta$ .

## III. PARTICLE METHODS

### A. The Joint Path Space Approach

In classical Rao-Blackwellized particle filtering, one first notes the following decomposition of the joint posterior:

$$p(\theta, \mathbf{x}_{1:t} | \mathbf{y}_{1:t}) = p(\theta | \mathbf{x}_{1:t}, \mathbf{y}_{1:t}) p(\mathbf{x}_{1:t} | \mathbf{y}_{1:t}).$$

Consequently, given the state path  $\mathbf{x}_{1:t}$ , we can solve for the map  $\theta$  analytically. This leaves us with only having to carry out particle filtering to compute the posterior distribution of the robot state  $p(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})$ .

<sup>1</sup>For clarification, we drop the action value or motion command from the transition prior  $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t)$

If we had a set of samples (or *particles*)  $\{\mathbf{x}_t^{(i)}\}_{i=1}^N$  from  $p(\mathbf{x}_t | \mathbf{y}_{1:t})$ , we could approximate the distribution with the Monte Carlo estimate

$$\widehat{p}(d\mathbf{x}_t | \mathbf{y}_{1:t}) = \frac{1}{N} \sum_{i=1}^N \delta_{\mathbf{x}_t^{(i)}}(d\mathbf{x}_t),$$

where  $\delta_{\mathbf{x}_t^{(i)}}(d\mathbf{x}_t)$  denotes the delta Dirac function. This estimate converges almost surely to the true expectation as  $N$  goes to infinity. Unfortunately, one cannot easily sample from the marginal distribution  $p(\mathbf{x}_t | \mathbf{y}_{1:t})$  directly. Instead, we draw particles from  $p(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})$  and samples  $\mathbf{x}_{1:t-1}$  are ignored. This is a valid way to draw samples from a marginal distribution and is at the core of most Monte Carlo statistical methods. The unknown normalizing constant precludes us from sampling directly from the posterior. Instead, we draw samples from a proposal distribution  $q$  and weight the particles according to the following importance ratio:

$$w_t(\mathbf{x}_{1:t}) = \frac{p(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})}{q(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})}$$

The proposal distribution is constructed sequentially

$$q(\mathbf{x}_{1:t} | \mathbf{y}_{1:t}) = q(\mathbf{x}_{1:t-1} | \mathbf{y}_{1:t-1}) q(\mathbf{x}_t | \mathbf{y}_t, \mathbf{x}_{t-1})$$

and, hence, the importance weights can be updated recursively in time

$$w_t(\mathbf{x}_{1:t}) = \frac{p(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})}{p(\mathbf{x}_{1:t-1} | \mathbf{y}_{1:t-1}) q(\mathbf{x}_t | \mathbf{y}_t, \mathbf{x}_{t-1})} w_{t-1}(\mathbf{x}_{1:t-1}). \quad (1)$$

Given a set of  $N$  particles  $\{\mathbf{x}_{1:t-1}^{(i)}\}_{i=1}^N$ , we obtain a set of particles  $\{\mathbf{x}_{1:t}^{(i)}\}_{i=1}^N$  by sampling from  $q(\mathbf{x}_t | \mathbf{y}_t, \mathbf{x}_{t-1}^{(i)})$  and applying the weights of equation (1).

The familiar particle filtering equations for this model are obtained by remarking that

$$p(\mathbf{x}_{1:t} | \mathbf{y}_{1:t}) \propto p(\mathbf{x}_{1:t}, \mathbf{y}_{1:t}) = \prod_{k=1}^t p(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{x}_{k-1}),$$

given which, equation (1) becomes

$$\widetilde{w}_t^{(i)} \propto \frac{p(\mathbf{y}_t | \mathbf{x}_t^{(i)}) p(\mathbf{x}_t^{(i)} | \mathbf{x}_{t-1}^{(i)})}{q(\mathbf{x}_t^{(i)} | \mathbf{y}_t, \mathbf{x}_{t-1}^{(i)})} \widetilde{w}_{t-1}^{(i)}.$$

This iterative scheme produces a weighted measure  $\{\mathbf{x}_{1:t}^{(i)}, w_t^{(i)}\}_{i=1}^N$ , where  $w_t^{(i)} = \widetilde{w}_t^{(i)} / \sum_j \widetilde{w}_t^{(j)}$ , and is known as Sequential Importance Sampling (SIS).

It has been proved [22] that *the variance of the importance weights in SIS increases over time*. This causes most particles to have very small probability. A common strategy to solve this *degeneracy*, consist on using a resampling step (SIR) after updating the weights to replicate samples with high probability and prune those with negligible weight [23].

This is the procedure in common use by practitioners. It can be deceptive: although only the state  $\mathbf{x}_t$  is being updated every round, the algorithm is nonetheless importance sampling in the *growing joint path space*  $\mathcal{X}^t$ .

Formally, the resampling step should be done along the full path  $\{\mathbf{x}_{1:t}^{(i)}, w_{1:t}^{(i)}\}_{i=1}^N$ . Since dynamic systems forget the past exponentially fast, several authors carry out resampling over

the marginal space  $\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^N$ . This would be fine if, for example, we were interested in tracking dynamic maps.

Static parameter estimation and model selection problems do not necessarily exhibit an exponential forgetting behavior. For example, static maps depend on the whole state trajectory. Resampling these trajectories in the joint path space is guaranteed to deplete the past in finite time; since there is only a bounded number of trajectories. Alternatively, resampling from the marginal space still leaves us with an accumulation of Monte Carlo errors over time. The estimate of the map will depend on this increasing sequence of errors. Some implementations have introduced artificial dynamics or Markov chain Monte Carlo (MCMC) rejuvenation steps to reduce the severity of the problem, but these approaches do not overcome the problem [24]. *In conclusion, whether we resample or not, learning the static map as a function of a growing path in Monte Carlo simulation is a bad idea.*

The same degeneracy problem arises if we try to obtain estimates of the filter derivative  $\nabla_{\theta} p_{\theta}(\mathbf{x}_t | \mathbf{y}_{1:t})$  for recursive (online) map estimation. To see this, let  $\nabla_{\theta} p_{\theta}(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})$  denote the gradient vector of the path posterior with respect to the map. Then, we have

$$\nabla_{\theta} p_{\theta}(\mathbf{x}_{1:t} | \mathbf{y}_{1:t}) = \frac{\nabla_{\theta} p_{\theta}(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})}{p_{\theta}(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})} p_{\theta}(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})$$

and, consequently the filter derivative, necessary for online map learning, is given by:

$$\nabla_{\theta} p_{\theta}(\mathbf{x}_t | \mathbf{y}_{1:t}) = \int_{\mathcal{X}^{t-1}} \frac{\nabla_{\theta} p_{\theta}(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})}{p_{\theta}(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})} p_{\theta}(\mathbf{x}_{1:t} | \mathbf{y}_{1:t}) d\mathbf{x}_{1:t-1} \quad (2)$$

Using standard particle filters to approximate the filter derivative we are implicitly carrying out importance sampling on a vast growing space with proposal  $p_{\theta}(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})$  and weight  $\frac{\nabla_{\theta} p_{\theta}(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})}{p_{\theta}(\mathbf{x}_{1:t} | \mathbf{y}_{1:t})}$ . This should be enough reason to call for a new approach. Yet, the problem is even worse.

The filter derivative is a signed-measure, and not a standard probability measure. It consists of positive and negative functions over disjoint parts of the state space and it sums to zero over the entire state space. A serious problem, when carrying out classical particle filtering to estimate this signed-measure, is that particles with positive and negative weights will cancel each other, say, in parts of the space where the derivative is close to zero. This is wasteful and statistically harmful. See Figure 1 of [21] for a beautiful depiction of this problem.

The technique presented in the following section overcomes these deficiencies.

## B. The Marginal Space Approach

1) *Marginal Filtering and Filter Derivative:* To eliminate the problems discussed in the previous section, we will perform particle filtering *directly* on the marginal distribution  $p(\mathbf{x}_t | \mathbf{y}_{1:t})$  instead of on the joint space [20], [25], [21]. To do so, we begin by noting that the predictive density can be obtained by marginalization:

$$p_{\theta}(\mathbf{x}_t | \mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t | \mathbf{x}_{t-1}) p_{\theta}(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1} \quad (3)$$

To simplify the exposition later on, we introduce the following notation [21]:

$$p_{\theta}(\mathbf{x}_t | \mathbf{y}_{1:t}) \triangleq \frac{\xi_{\theta}(\mathbf{x}_t, \mathbf{y}_{1:t})}{\int \xi_{\theta}(\mathbf{x}_t, \mathbf{y}_{1:t}) d\mathbf{x}_t} \quad (4)$$

where  $\xi$  represents the unnormalized distribution.

Using equation (3) and Bayes rule, the unnormalized filtering distribution can be expanded as follows:

$$\xi_{\theta}(\mathbf{x}_t, \mathbf{y}_{1:t}) = p_{\theta}(\mathbf{y}_t | \mathbf{x}_t) \int p(\mathbf{x}_t | \mathbf{x}_{t-1}) p_{\theta}(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1}.$$

In order to obtain the gradient vector with respect to the map variables, we apply standard differentiation rules to equation (4), yielding:

$$\begin{aligned} \nabla_{\theta} p_{\theta}(\mathbf{x}_t | \mathbf{y}_{1:t}) &= \frac{\nabla_{\theta} \xi_{\theta}(\mathbf{x}_t, \mathbf{y}_{1:t})}{\int \xi_{\theta}(\mathbf{x}_t, \mathbf{y}_{1:t}) d\mathbf{x}_t} \\ &- p_{\theta}(\mathbf{x}_t | \mathbf{y}_{1:t}) \frac{\int \nabla_{\theta} \xi_{\theta}(\mathbf{x}_t, \mathbf{y}_{1:t}) d\mathbf{x}_t}{\int \xi_{\theta}(\mathbf{x}_t, \mathbf{y}_{1:t}) d\mathbf{x}_t}. \end{aligned} \quad (5)$$

Similarly, using the expansions for the derivatives of logs, the gradient of  $\xi(\cdot)$  can be written as follows:

$$\begin{aligned} \nabla_{\theta} \xi_{\theta}(\mathbf{x}_t | \mathbf{y}_{1:t}) &= \xi_{\theta}(\mathbf{y}_t | \mathbf{x}_t) \nabla_{\theta} \log p_{\theta}(\mathbf{y}_t | \mathbf{x}_t) \\ &+ p_{\theta}(\mathbf{y}_t | \mathbf{x}_t) \int p(\mathbf{x}_t | \mathbf{x}_{t-1}) \nabla_{\theta} p_{\theta}(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1} \end{aligned} \quad (6)$$

2) *Monte Carlo Implementation:* We are now ready to present the particle algorithm of [21] for approximating the filter derivative efficiently. Assume that at time  $t-1$ , we have a Monte Carlo approximations of the filter and its gradient. We denote the normalized and unnormalized filter and gradient approximations by:  $\widehat{p}_{\theta}(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1}) = \sum w_{t-1}^{(i)} \delta_{\mathbf{x}_{t-1}^{(i)}}$ ,  $\widehat{\xi}_{\theta}(\mathbf{x}_t, \mathbf{y}_{1:t}) = \frac{1}{N} \sum \widetilde{w}_t^{(i)} \delta_{\mathbf{x}_t^{(i)}}$ ,  $\widehat{\nabla_{\theta} p_{\theta}}(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1}) = \sum_{i=1}^N w_{t-1}^{(i)} \beta_{t-1}^{(i)} \delta_{\mathbf{x}_{t-1}^{(i)}}(\mathbf{x}_{t-1})$ ,  $\widehat{\nabla_{\theta} \xi_{\theta}}(\mathbf{x}_t, \mathbf{y}_{1:t}) = \frac{1}{N} \sum_{i=1}^N \widetilde{\rho}_t^{(i)} \delta_{\mathbf{x}_t^{(i)}}(\mathbf{x}_t)$ . The integral in equation (3) is generally not solveable analytically, but since we have the particle approximation, we can approximate it as the weighted kernel density estimate

$$\widehat{p}_{\theta}(\mathbf{x}_t | \mathbf{y}_{1:t-1}) = \sum_{j=1}^N w_{t-1}^{(j)} p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(j)}).$$

While we are free to choose any proposal distribution that has appropriate support, it is convenient to assume that the marginal proposal takes a similar form, namely

$$q_{\theta}(\mathbf{x}_t | \mathbf{y}_{1:t}) = \sum_{j=1}^N w_{t-1}^{(j)} q_{\theta}(\mathbf{x}_t | \mathbf{y}_t, \mathbf{x}_{t-1}^{(j)}).$$

We can easily draw particles from this proposal using multinomial or stratified sampling and compute the new unnormalized importance weights:

$$\widetilde{w}_t^{(i)} = \frac{p_{\theta}(\mathbf{y}_t | \mathbf{x}_t^{(i)}) \sum_{j=1}^N w_{t-1}^{(j)} p(\mathbf{x}_t^{(i)} | \mathbf{x}_{t-1}^{(j)})}{q_{\theta}(\mathbf{x}_t^{(i)} | \mathbf{y}_{1:t})}$$

$$\widetilde{\rho}_t^{(i)} = \widetilde{w}_t^{(i)} \nabla_{\theta} \log p_{\theta}(\mathbf{y}_t | \mathbf{x}_t^{(i)}) + \frac{\sum_j w_{t-1}^{(j)} p(\mathbf{x}_t^{(i)} | \mathbf{x}_{t-1}^{(j)}) \beta_{t-1}^{(j)}}{q_{\theta}(\mathbf{x}_t^{(i)} | \mathbf{y}_{1:t})}$$

Finally, substituting the above Monte Carlo estimates into the expression for the derivative of  $p_\theta$  in terms of  $\xi_\theta$ , we obtain the normalized weights at time  $t$ .

$$w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_j \tilde{w}_t^{(j)}}; w_t^{(i)} \beta_t^{(i)} = \frac{\tilde{\rho}_t^{(i)}}{\sum_j \tilde{w}_t^{(j)}} - w_t^{(i)} \frac{\sum_j \tilde{\rho}_t^{(j)}}{\sum_j \tilde{w}_t^{(j)}}$$

Note the advantages of marginal filtering. First, *the importance sampling process now happens in the marginal space*. In addition, the last integral in equation (6) can be expanded using the score identity:

$$\int p(\mathbf{x}_t | \mathbf{x}_{t-1}) \nabla_\theta \log [p_\theta(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1})] p_\theta(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1}$$

That is we sample from the marginal filtering distribution and weight with  $\beta \triangleq \nabla_\theta \log [p_\theta(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1})]$ . Contrast this with equation (2). The other thing to note, as pointed out in [21] is that the marginal filter derivative allows us to obtain a particle approximation of the Hahn-Jordan decomposition. This implies that we can surmount the problem of particles of opposite signs cancelling each other out in infinitesimal neighborhoods of the state space (see Figure 1 of [21]).

3) *On-Line Map Learning*: Armed with Monte Carlo estimates of the filter derivative, we can now attack the problem of developing recursive map estimators. Here, we choose to maximize the predictive likelihood (also known as the innovations or evidence):

$$p_\theta(\mathbf{y}_t | \mathbf{y}_{1:t-1}) = \int p_\theta(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{x}_{t-1}) p_\theta(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1:t}$$

To accomplish this, we adopt the following stochastic approximation algorithm:

$$\theta_t = \theta_{t-1} + \gamma_t \nabla_\theta \log \widehat{p}_\theta(\mathbf{y}_t | \mathbf{y}_{1:t-1})$$

Provided that the step size  $\gamma_t$  satisfies standard stochastic approximation criteria, see for example [26], it can be shown that  $\theta_t$  converges to true parameters which are the global maxima of  $l(\theta)$ , where [21]

$$l(\theta) = \lim_{k \rightarrow \infty} \frac{1}{k+1} \sum_{t=1}^k \log p_\theta(\mathbf{y}_t | \mathbf{y}_{1:t-1})$$

A detailed analysis is presented in [27]. The only remaining detail is to derive the Monte Carlo approximation of the gradient of the predictive distribution

$$\begin{aligned} \nabla_\theta \log \widehat{p}_\theta(\mathbf{y}_t | \mathbf{y}_{1:t-1}) &= \frac{\widehat{\nabla_\theta p_\theta(\mathbf{y}_t | \mathbf{y}_{1:t-1})}}{\widehat{p}_\theta(\mathbf{y}_t | \mathbf{y}_{1:t-1})} = \\ &= \frac{\int \widehat{\nabla_\theta \xi_\theta(\mathbf{x}_t, \mathbf{y}_{1:t})} d\mathbf{x}_t}{\int \widehat{\xi_\theta(\mathbf{x}_t, \mathbf{y}_{1:t})} d\mathbf{x}_t} = \frac{\sum_{j=1}^N \tilde{\rho}_t^{(j)}}{\sum_{j=1}^N \tilde{w}_t^{(j)}} \end{aligned}$$

4) *Pseudo-Code for Marginal-SLAM*: The Marginal-SLAM algorithm is depicted in Figure 1. Note that it is linear in the number of features. It has an  $O(N^2)$  complexity in terms of the number of samples, but this can be reduced to  $O(N \log N)$  using the fast multipole expansions and metric tree recursions proposed in [20].

The marginal particle filter is an old idea [28], [29]. Yet, because of its large computational cost, it was not fully

### Marginal-SLAM

- For  $i = 1, \dots, N$ , sample the robot state from the proposal

$$\mathbf{x}_t^{(i)} \sim \sum_{j=1}^N w_{t-1}^{(j)} q(\mathbf{x}_t | \mathbf{y}_t, \mathbf{x}_{t-1}^{(j)})$$

- For  $i = 1, \dots, N$ , evaluate the importance weights

$$\tilde{w}_t^{(i)} = \frac{p_\theta(\mathbf{y}_t | \mathbf{x}_t^{(i)}) \sum_{j=1}^N w_{t-1}^{(j)} p(\mathbf{x}_t^{(i)} | \mathbf{x}_{t-1}^{(j)})}{q_\theta(\mathbf{x}_t^{(i)} | \mathbf{y}_{1:t})}$$

$$\tilde{\rho}_t^{(i)} = \tilde{w}_t^{(i)} \nabla_\theta \log p_\theta(\mathbf{y}_t | \mathbf{x}_t^{(i)}) + \frac{\sum_j w_{t-1}^{(j)} p(\mathbf{x}_t^{(i)} | \mathbf{x}_{t-1}^{(j)}) \beta_{t-1}^{(j)}}{q_\theta(\mathbf{x}_t^{(i)} | \mathbf{y}_{1:t})}$$

- Normalise the importance weights

$$w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_j \tilde{w}_t^{(j)}}; w_t^{(i)} \beta_t^{(i)} = \frac{\tilde{\rho}_t^{(i)}}{\sum_j \tilde{w}_t^{(j)}} - w_t^{(i)} \frac{\sum_j \tilde{\rho}_t^{(j)}}{\sum_j \tilde{w}_t^{(j)}}$$

- Update the map vector

$$\theta_t = \theta_{t-1} + \gamma_t \frac{\sum_j \tilde{\rho}_t^{(j)}}{\sum_j \tilde{w}_t^{(j)}}$$

- Update the learning rate  $\gamma_t$ .

Fig. 1. The Marginal-SLAM algorithm at time  $t$ .

explored until the introduction of fast methods [20]. When using the transition prior as proposal, the marginal filter and classical particle filter are equivalent, but this is no longer true when computing the derivative of the filter as outlined in [25] and this paper.

## IV. EXPERIMENTS

We compare Marginal-SLAM and FastSLAM in a large scale, highly noisy simulated environment with known data association. The environment is a square-like corridor with point landmarks in the walls. For simplicity, we use the transition prior as the proposal distribution for both techniques. In future work, optimal proposals could be considered like in FastSLAM 2.0.

The robot motion, that is, the transition model, is based on a simple differential drive vehicle

$$\begin{bmatrix} X_t \\ Y_t \\ \psi_t \end{bmatrix} = \begin{bmatrix} X_{t-1} \\ Y_{t-1} \\ \psi_{t-1} \end{bmatrix} + \begin{bmatrix} d_t \cos(\psi_{t-1}) \\ d_t \sin(\psi_{t-1}) \\ \alpha_t \end{bmatrix}$$

where  $\mathbf{x}_t = [X_t, Y_t, \psi_t]$  denotes the robot position and orientation and  $\mathbf{u}_t = [d, \alpha]$  is the motion command (displacement and heading) at time  $t$  with corresponding Gaussian noise  $v_t \sim \mathcal{N}(0, \text{diag}(\sigma_d, \sigma_\alpha))$ . The observations are gathered using range and bearing sensors  $\mathbf{y}_t = [\rho, \phi]$  with a point feature detector

$$\begin{bmatrix} \hat{\rho} \\ \hat{\phi} \end{bmatrix} = \begin{bmatrix} \sqrt{\Delta_x^2 + \Delta_y^2} \\ \arctan\left(\frac{-\Delta_x \sin(\theta_t^R) + \Delta_y \cos(\theta_t^R)}{\Delta_x \cos(\theta_t^R) + \Delta_y \sin(\theta_t^R)}\right) \end{bmatrix}$$

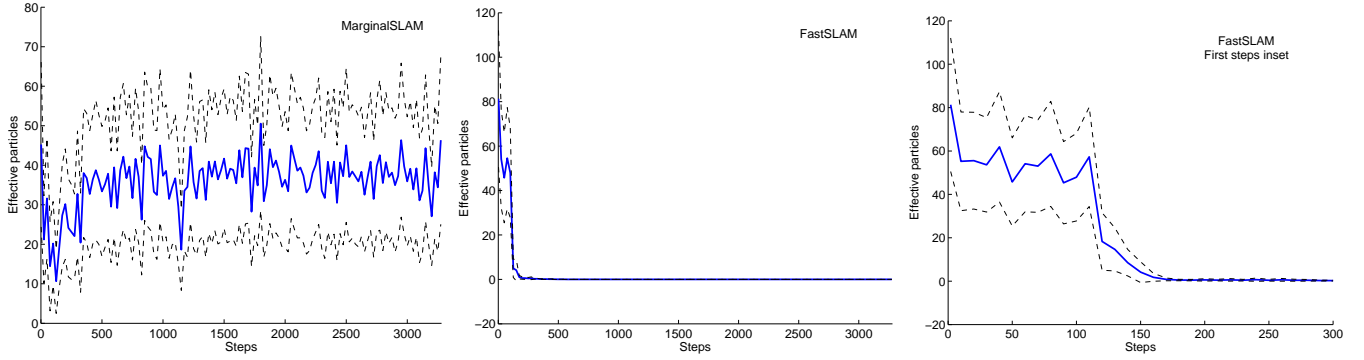


Fig. 2. The plots represent the average number of effective particles  $N_{eff}$ , with corresponding confidence intervals, in 25 long term simulations. The total number of particles is 200 in both cases. The left plot corresponds to Marginal-SLAM, where the filter achieves a steady state with a 25% effective number of particles, allowing to update the same are for indefinite time. The middle plot shows the same experiment for FastSLAM. It quickly drops in every simulation and fails after 200 steps. Just few realizations survive more steps. The right plot is a zoomed view of the middle plot.

where  $\Delta_x = \theta_x - X_t$  and  $\Delta_y = \theta_y - Y_t$ ; with  $[\theta_x, \theta_y]$  denoting the feature location. The sensor has white Gaussian noise  $v_t \sim \mathcal{N}(0, \text{diag}(\sigma_\rho, \sigma_\phi))$ .

Hence, the likelihood function for a single feature is

$$p(\rho, \phi | \mathbf{x}) = \frac{1}{\sqrt{2\pi}\sigma_\rho\sigma_\phi} \exp \left[ -\frac{1}{2} \left( \frac{(\rho - \hat{\rho})^2}{\sigma_\rho^2} + \frac{(\phi - \hat{\phi})^2}{\sigma_\phi^2} \right) \right]$$

Thus, the gradient of the log-likelihood corresponds to

$$\nabla \log(p(\rho, \phi)) = \frac{1}{\rho} \begin{bmatrix} \frac{\Delta_x(\rho - \hat{\rho})}{\sigma_\rho^2} - \frac{\Delta_y(\phi - \hat{\phi})}{\rho\sigma_\phi^2} \\ \frac{\Delta_y(\rho - \hat{\rho})}{\sigma_\rho^2} + \frac{\Delta_x(\phi - \hat{\phi})}{\rho\sigma_\phi^2} \end{bmatrix}$$

We carried out several simulations by varying the amount of sensor and motion noise, landmark density and loop size. The system is able to close large loops with large range and bearing noise. However, very large sensor noise (e.g. sonar) in large loops is still a difficult task. Data driven proposals could be adopted in the future to eliminate this problem.

Figure 2 shows a comparison between the number of effective particles  $N_{eff} = 1 / \sum_{i=1}^N w_i^2$  in Marginal-SLAM and FastSLAM<sup>2</sup>. Clearly, the marginal particle filter reaches a steady state, but FastSLAM quickly loses particle diversity.

The final map using both approaches is shown in Figure 3. In the Marginal-SLAM plot, the relative locations of the landmarks and the robot converge to the true solution. However, the global location is biased with respect to the ground truth. This is due to the observability assumptions in our SLAM model. The robot location can only be measured through the landmark location. The final map is valid up to an isometric transformation (translation plus rotation). This effect can be reduced if we fix a landmark location.

Although our goal has been to develop a method for static maps using decreasing learning rates, it is possible to adopt small constant learning rates to track slowly changing map features. Mapping in real scenarios requires the ability to deal with pseudo-dynamic objects, like chairs and doors. Those elements are difficult to identify as dynamic, but their

movements can lead to inconsistent maps. Figure 4 shows the evolution of a parameter estimate when the true landmark location changes. This ability to track moving features also implies that it would enable Marginal-SLAM to recover from wrong data association, biased map merging and loop closing, provided that the correct correspondences are obtained in subsequent steps. The selection of the learning ratio depends on the required accuracy, the convergency speed and the ability to track objects.

## V. CONCLUSIONS

The experiments and arguments indicate that Marginal-SLAM is an important new direction in the design of particle methods for SLAM. Algorithms designed to work on the marginal space appear to behave better than the ones designed to work on the path space.

In this preliminary work, Marginal-SLAM exhibits nice properties, such as being able to track slowly moving objects and potentially being able to recover from erroneous data-association. Marginal-SLAM does not suffer from some shortcomings of existing particle methods for SLAM. When mapping known areas, the algorithm reaches an accurate steady state without diverging. However, efficient convergence in large-scale SLAM domains is still an open question. The presented method requires a considerable amount of information to converge to the solution, which is a strong assumption in SLAM. *What is missing is a fully Bayesian way of estimating the static map parameters, while integrating over the states recursively in time.*

In future work, we plan to test the algorithm more thoroughly in real domains and introduce known improvements like the N-body methods or the more efficient Hessian based maximum likelihood. We also plan to focus on solving the problem of designing efficient full-Bayesian recursive parameter estimators for nonlinear state spaces.

## VI. ACKNOWLEDGMENTS

We are very thankful to Arnaud Doucet for many thoughtful discussions. This work was supported in part by the Direcció General de Investigaci6n of Spain under project DPI2003-07986 and by NSERC and MITACS.

<sup>2</sup>Results are based on the following parameter settings:  $N = 200$ ,  $\sigma_d = 0.1m$ ,  $\sigma_\alpha = 0.5deg$ ,  $\sigma_\rho = 0.025m$ ,  $\sigma_\phi = 3deg$ .

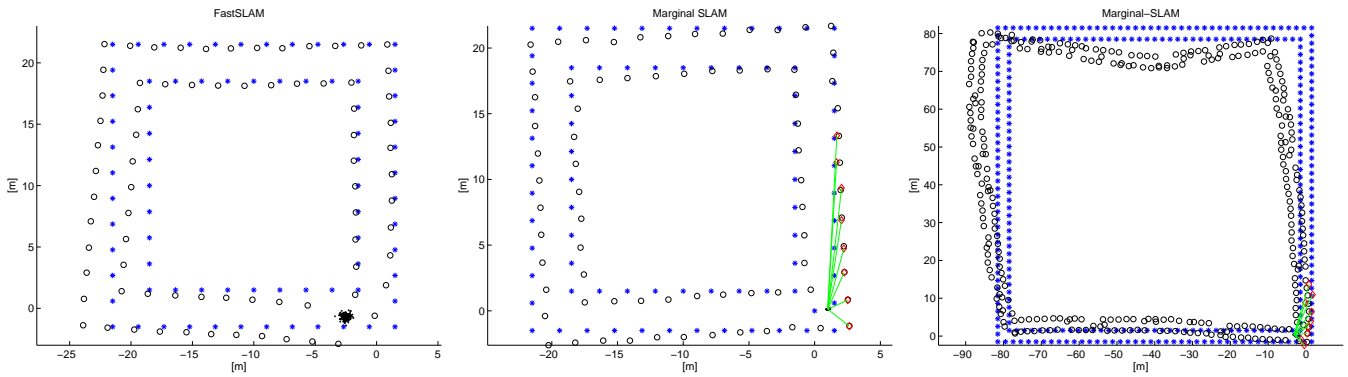


Fig. 3. a) Particle degeneracy in FastSLAM prevents it from closing the loop. b) Marginal-SLAM is able to close the loop and converges to the true map after 10 laps. Although the map seems rotated, the relative location of the features and the robot location is almost perfect. c) Marginal-SLAM after 10 laps in a bigger loop with high motion and observation noise  $\sigma_\alpha = 10deg, \sigma_\phi = 10deg$ . The loop is closed but the convergence is slow.

### REFERENCES

- [1] H. Durrant-Whyte and T. Bailey, "Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms," *Robotics and Automation Magazine*, vol. 13, pp. 99–110, 2006.
- [2] S. Thrun, "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millenium*, G. Lakemeyer and B. Nebel, Eds. Morgan Kaufmann, 2002.
- [3] R. Smith, M. Self, and P. Cheeseman, "A stochastic map for uncertain spatial relationships," in *Robotics Research, The Fourth Int. Symposium*, O. Faugeras and G. Giralt, Eds. The MIT Press, 1988, pp. 467–474.
- [4] M. Paskin, "Thin junction tree filters for simultaneous localization and mapping," in *Int. Joint Conf. on Artificial Intelligence*, 2003, pp. 1157–1164.
- [5] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, "Consistency of the EKF-SLAM algorithm," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2006.
- [6] S. Julier and J. K. Uhlmann, "A counter example to the theory of simultaneous localization and map building," in *IEEE Int. Conf. on Robotics and Automation*, 2001, pp. 4238–4243.
- [7] J. Castellanos, J. Neira, and J. Tardós, "Limits to the consistency of EKF-based SLAM," in *5th IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon, July 2004.
- [8] U. Frese, "Treemap: An  $O(\log n)$  Algorithm for Simultaneous Localization and Mapping," in *Spatial Cognition IV*, C. Freksa, Ed. Springer Verlag, 2005, pp. 455–476.
- [9] J. Castellanos, R. Martinez-Cantin, J. Tardós, and J. Neira, "Robocentric map joining: Improving the consistency of EKF-SLAM," *Robotics and Autonomous Systems*, vol. 55, pp. 21–29, 2007.
- [10] R. Martinez-Cantin and J. Castellanos, "Unscented SLAM for Large-Scale Outdoor Environments," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2005, pp. 328–333.
- [11] D. Fox, S. Thrun, W. Burgard, and F. Dellaert, "Particle filters for mobile robot localization," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. J. Gordon, Eds. Springer-Verlag, 2001.
- [12] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-Blackwellised particle filtering for dynamic Bayesian networks," in *Uncertainty in Artificial Intelligence*, C. Boutilier and M. Godsztmidt, Eds. Morgan Kaufmann Publishers, 2000, pp. 176–183.
- [13] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Int. Joint Conf. on Artificial Intelligence*, Acapulco, Mexico, 2003.
- [14] T. Bailey, J. Nieto, and E. Nebot, "Consistency of the FastSLAM algorithm," in *IEEE Int. Conf. on Robotics and Automation*, 2006.
- [15] C. Stachniss, G. Grisetti, and W. Burgard, "Recovering particle diversity in a Rao-Blackwellized particle filter for SLAM after actively closing loops," in *IEEE Int. Conf. on Robotics and Automation*, 2005, pp. 667–672.
- [16] P. Elinas, R. Sim, and J. Little, " $\sigma$ SLAM: Stereo vision SLAM using the Rao-Blackwellised particle filter and a novel mixture proposal distribution," in *IEEE Int. Conf. on Robotics and Automation*, 2006.
- [17] C. Andrieu, N. de Freitas, and A. Doucet, "Sequential MCMC for Bayesian model selection," in *IEEE Higher Order Statistics Workshop*, Caesarea, Israel, 1999, pp. 130–134.
- [18] S. Thrun, "Exploration and model building in mobile robot domains," in *Proceedings of the ICNN*, E. Ruspini, Ed., 1993, pp. 175–180.
- [19] —, "A probabilistic online mapping algorithm for teams of mobile robots," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 335–363, 2001.
- [20] M. Klaas, N. de Freitas, and A. Doucet, "Toward practical  $n^2$  Monte Carlo: The marginal particle filter," in *Uncertainty in Artificial Intelligence*, 2005.
- [21] G. Poyadjis, A. Doucet, and S. Singh, "Maximum likelihood parameter estimation using particle methods," in *Joint Statistical Meeting*, 2005.
- [22] A. Doucet, "On sequential simulation-based methods for Bayesian filtering," Dept. of Engineering, Cambridge University, Tech. Rep. CUED/F-INFENG/TR 310, 1998.
- [23] A. Doucet, N. de Freitas, and N. J. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.
- [24] J. Liu and M. West, "Combined parameter and state estimation in simulation-based filtering," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. J. Gordon, Eds. Springer-Verlag, 2001.
- [25] G. Poyadjis, A. Doucet, and S. Singh, "Particle methods for optimal filter derivative: Application to parameter estimation," in *IEEE ICASSP*, 2005.
- [26] J. C. Spall, *Introduction to Stochastic Search and Optimization*. Wiley, 2005.
- [27] V. B. Tadic and A. Doucet, "Exponential forgetting and geometric ergodicity for optimal filtering in general state-space models," *Stochastic Processes and Their Applications*, vol. 115, pp. 1408–1436, 2005.
- [28] P. Fearnhead, "Sequential Monte Carlo methods in filter theory," Ph.D. dissertation, Dept. of Statistics, Oxford University, England, 1998.
- [29] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *JASA*, vol. 94, no. 446, pp. 590–599, 1999.

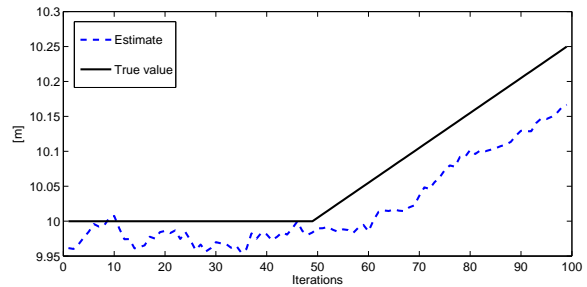


Fig. 4. Tracking a moving map feature with Marginal-SLAM with  $\sigma_d = 0.01m, \sigma_\alpha = 0.05deg, \sigma_\rho = 0.1m, \sigma_\phi = 0.5deg$ . Black thick line: true location. Blue thin line: estimated location.