Consistency of the FastSLAM algorithm

Conference Paper *in* Proceedings - IEEE International Conference on Robotics and Automation · January 2006

DOI: 10.1109/ROBOT.2006.1641748 · Source: DBLP

3 authors, including:

Juan I. Nieto
ETH Zurich
118 PUBLICATIONS **1,610** CITATIONS

SEE PROFILE

Eduardo Mario Nebot
University of Sydney
160 PUBLICATIONS **5,086** CITATIONS

SEE PROFILE

Consistency of the FastSLAM Algorithm

Tim Bailey, Juan Nieto and Eduardo Nebot
Australian Centre for Field Robotics
University of Sydney, NSW, Australia
Email: tbailey@acfr.usyd.edu.au

***Abstract*— This paper presents an analysis of FastSLAM—a Rao-Blackwellised particle filter formulation of *simultaneous localisation and mapping*. It shows that the algorithm degenerates with time, regardless of the number of particles used or the density of landmarks within the environment, and will always produce optimistic estimates of uncertainty in the long-term. In essence, FastSLAM behaves like a non-optimal local search algorithm; in the short-term it may produce consistent uncertainty estimates but, in the long-term, it is unable to adequately explore the state-space to be a reasonable Bayesian estimator. However, the number of particles and landmarks *does* affect the accuracy of the *estimated mean* and, given sufficient particles, FastSLAM can produce good non-stochastic estimates in practice. FastSLAM also has several practical advantages, particularly with regard to data association, and will probably work well in combination with other versions of stochastic SLAM, such as EKF-based SLAM.**

I. Introduction

The problem of simultaneous localisation and mapping (SLAM) is a fundamental capability for autonomous navigation in unknown environments. The stochastic solution to SLAM by Smith *et al.* [18] was the first to formally address measurement error correlations that arise during the map building process. As a robot builds a map, the landmark location errors are dependent on the robot pose error and, as the robot localises from this map, its pose estimate is dependent on the landmark errors. Stochastic SLAM addresses these dependencies explicitly by maintaining a joint estimate of the vehicle and map states using a recursive Bayesian filter. Throughout the 1990s, the dominant realisation of stochastic SLAM was built on the extended Kalman filter (EKF). This approach is subject to a variety of problems, with regard to computational complexity, non-linearity and data association, many of which have been addressed in the recent literature (e.g., [1], [3], [14]).

The *FastSLAM* algorithm [12] is a new solution to stochastic SLAM that is not based on the Kalman filter but, instead, uses a *particle filter* [8] to approximate the ideal recursive Bayesian filter. More precisely, it involves a partitioned state-space whereby the robot pose states are represented by particles and the landmark states are estimated analytically by Kalman filters. This factoring of the state into a sampled part and an analytical part is termed *Rao-Blackwellisation* [4], and the FastSLAM algorithm is a Rao-Blackwellised particle filter.

By representing the robot pose with samples, FastSLAM addresses the worst of SLAM's non-linearity issues—the accumulated non-Gaussianess of the pose uncertainty distribution. It requires $O(NM)$ computation and storage, where N is the number of particles and M is the number of landmarks in the map. Thus, for fixed N, it has linear time-complexity in M; hence the appellation "*Fast*". (In fact, Montemerlo presents an implementation that has $O(N \log_2 M)$ time-complexity [12].) FastSLAM has the added attraction that it permits each particle to perform its own data association decisions, independent of other particles, and so facilitates a simple form of multi-hypothesis data association [13], [15].

An important property of the FastSLAM algorithm is that each particle does not represent a single momentary robot pose. Rather, it represents an entire robot path history and associated map of landmarks. This has important implications in terms of consistency—the ability of the filter to accurately estimate uncertainty. That FastSLAM degenerates over time has been noted in the literature (e.g., [12, Section 4.1],[17]). In this paper, we examine this degeneracy quantitatively; we examine how quickly particle diversity is lost, what the effect of diversity loss has on the filter's uncertainty estimate, and how factors like number of particles and landmark density affect estimation errors.

This paper foregoes full derivation of FastSLAM 1.0 and 2.0, as they are well explained in [12]. Rather, we focus on the definition of the FastSLAM state, the effects of resampling, and the ability of the algorithm to approximate the "true" state uncertainty. We found FastSLAM 2.0 to be superior to FastSLAM 1.0 in all respects due to its application of the *optimal importance function* [7]. Therefore, all results in this paper concern simulations of FastSLAM 2.0.

The next section presents the process and observation models used in our simulation experiments, which sets the context for these results in terms of a 2-D vehicle with a range-bearing sensor. Section III describes the Rao-Blackwellised SLAM state and its properties. Section IV discusses the need for resampling and its effects. Section V introduces the NEES measure as a gauge of filter consistency. Section VI presents the results of the simulation experiments in terms of estimate consistency and rate of degeneration. The final two sections provide discussion and conclusions.

II. Models for 2-D Range-Bearing SLAM

In this paper, we specify the SLAM state as the vehicle pose (position and heading) and the locations of stationary landmarks observed in the environment. The state at time k is represented by a joint state-vector $\mathbf{x}_k$.

$$\mathbf{x}_k = [x_{v_k}, y_{v_k}, \phi_{v_k}, x_1, y_1, \ldots, x_M, y_M]^T = \begin{bmatrix} \mathbf{x}_{v_k} \\ \mathbf{m} \end{bmatrix} \quad (1)$$

Notice that the map parameters $\mathbf{m} = [x_1, y_1, \ldots, x_M, y_M]^T$ do not have a time subscript as they are modelled as stationary.

To describe the vehicle motion, we use the kinematic model for the trajectory of the front wheel of a bicycle subject to rolling motion constraints (i.e., assuming zero wheel slip).

$$\mathbf{x}_{v_k} = \mathbf{f}_v\left(\mathbf{x}_{v_{k-1}}, \mathbf{u}_k\right) = \begin{bmatrix} x_{v_{k-1}} + V_k \Delta T \cos(\phi_{v_{k-1}} + \gamma_k) \\ y_{v_{k-1}} + V_k \Delta T \sin(\phi_{v_{k-1}} + \gamma_k) \\ \phi_{v_{k-1}} + \frac{V_k \Delta T}{B} \sin(\gamma_k) \end{bmatrix} \tag{2}$$

Here the time from $k-1$ to k is denoted ΔT, and during this period the velocity V_k and steering angle γ_k of the front wheel are assumed constant. Collectively, the velocity and steering values $\mathbf{u}_k = [V_k, \gamma_k]^T$ are termed the "controls". The wheelbase between the front and rear axles is B. The process model for the joint SLAM state is simply a concatenation of the vehicle motion model and the stationary landmark model.

$$\mathbf{x}_k = \mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{u}_k\right) = \begin{bmatrix} \mathbf{f}_v\left(\mathbf{x}_{v_{k-1}}, \mathbf{u}_k\right) \\ \mathbf{m} \end{bmatrix} \tag{3}$$

For a range-bearing measurement from the vehicle to landmark $\mathbf{m}_i = [x_i, y_i]^T$, the observation model is given by

$$\mathbf{z}_{i_k} = \mathbf{h}_i\left(\mathbf{x}_k\right) = \begin{bmatrix} \sqrt{(x_i - x_{v_k})^2 + (y_i - y_{v_k})^2} \\ \arctan \frac{y_i - y_{v_k}}{x_i - x_{v_k}} - \phi_{v_k} \end{bmatrix} \tag{4}$$

Adding new landmarks to the map uses an inverse form of the observation model as described in [1, Section 2.2.4] and [10, Section 2].

The vehicle motion model, the observation model, and the measured values of the control parameters $\mathbf{u}_k$, are not exact, but are subject to noise, which lead to uncertainty in the state estimate. For this reason, we require a probabilistic filter to recursively estimate a distribution over the state given noisy information.

We make the usual basic assumptions regarding the dependencies of measurement errors. That is, the process model is first-order Markov and independent of the map states,[1]

$$p\left(\mathbf{x}_{v_k} | \mathbf{X}_{v_{0:k-1}}, \mathbf{m}, \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k}\right) = p\left(\mathbf{x}_{v_k} | \mathbf{x}_{v_{k-1}}, \mathbf{u}_k\right) \tag{5}$$

and the observations are independent conditioned on the joint SLAM state.

$$p\left(\mathbf{Z}_{0:k} | \mathbf{X}_{v_{0:k}}, \mathbf{m}, \mathbf{U}_{0:k}\right) = \prod_{i=0}^{k} p\left(\mathbf{z}_i | \mathbf{x}_{v_i}, \mathbf{m}\right) \tag{6}$$

Multiple observations made at time k are also assumed conditionally independent. We further assume that the distributions in Eqns. 5 and 6 are Gaussian as this allows certain parts of the FastSLAM algorithm to be reasonably approximated by EKF equations.

[1] A brief note on notation. The *probability density function* (PDF) of a random variable $\mathbf{x}$ is denoted $p\left(\mathbf{x}\right)$ and a sample drawn from this distribution is $\mathbf{x}^{(i)}$. A history of values $\{\mathbf{x}_0, \ldots, \mathbf{x}_k\}$ from time 0 to time t is denoted $\mathbf{X}_{0:k}$.

III. Rao-Blackwellisated SLAM

Rao-Blackwellisation is a *variance reduction* technique for Monte Carlo integration, whereby a joint *probability density function* (PDF) is factored according to the product rule so as to reduce the dimensionality of the simulation-space.

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} \tag{7}$$

$$p\left(\mathbf{x}\right) = p\left(\mathbf{x}_2 | \mathbf{x}_1\right) p\left(\mathbf{x}_1\right) \tag{8}$$

It is applicable when $p\left(\mathbf{x}_2 | \mathbf{x}_1\right)$ can be evaluated analytically so that simulation is restricted to the space of $p\left(\mathbf{x}_1\right)$,

$$\mathbf{x}_1^{(i)} \sim p\left(\mathbf{x}_1\right) \tag{9}$$

and exact inference can be used to compute $p\left(\mathbf{x}_2 | \mathbf{x}_1^{(i)}\right)$. Marginal estimates may be found as sums.

$$p\left(\mathbf{x}_2\right) \approx \frac{1}{N} \sum_{i=1}^{N} p\left(\mathbf{x}_2 | \mathbf{x}_1^{(i)}\right) \tag{10}$$

For a fixed number of samples, the random variation in estimates based on Rao-Blackwellised sampling is always less than (or equal to) that obtained by sampling from the entire space of $p\left(\mathbf{x}\right)$.

A discussion of Rao-Blackwellisation applied to various forms of Monte Carlo sampling is provided in [4], and is presented as a way to reduce weight variance for particle filters in [7]. A good discussion in the context of particle filters is given in [16, Section 3.3], which shows the derivation of a general state-space problem partitioned into particle filter and Kalman filter components.

FastSLAM employs a particle filter primarily to address the problem of representing a system that is non-linear and non-Gaussian. It applies Rao-Blackwellisation to reduce the filter's sample-space from the joint state $[\mathbf{x}_{v_k}, \mathbf{m}]^T$ to just the vehicle pose states $\mathbf{x}_{v_k}$. The state partitioning in FastSLAM is defined as follows.

$$\begin{aligned} &p\left(\mathbf{X}_{v_{0:k}}, \mathbf{m} | \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}\right) \\ &\quad = p\left(\mathbf{X}_{v_{0:k}} | \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}\right) p\left(\mathbf{m} | \mathbf{X}_{v_{0:k}}, \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}\right) \\ &\quad = p\left(\mathbf{X}_{v_{0:k}} | \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}\right) p\left(\mathbf{m} | \mathbf{X}_{v_{0:k}}, \mathbf{Z}_{0:k}\right) \end{aligned} \tag{11}$$

Here the joint posterior is factored into a vehicle pose part and a map part conditioned on the pose. Notice that the joint PDF is defined in terms of the vehicle pose *history* $\mathbf{X}_{v_{0:k}}$. This is a critical aspect of the FastSLAM formulation as it permits efficient estimation of the map states. That is, given $\mathbf{X}_{v_{0:k}}$, and as a consequence of Eqn. 6, the individual landmark PDFs are independent.

$$p\left(\mathbf{m} | \mathbf{X}_{v_{0:k}}, \mathbf{Z}_{0:k}\right) = \prod_{i=1}^{M} p\left(\mathbf{m}_i | \mathbf{X}_{v_{0:k}}, \mathbf{Z}_{i_{0:k}}\right) \tag{12}$$

The FastSLAM posterior at time k is represented as a set of N weighted particles $\{w_0, \mathbf{X}_{v_{0:k}}^{(0)}, \ldots, w_N, \mathbf{X}_{v_{0:k}}^{(N)}\}$ over the vehicle pose states. Each sample $\mathbf{X}_{v_{0:k}}^{(i)}$ has an associated map PDF $p\left(\mathbf{m} | \mathbf{X}_{v_{0:k}}^{(i)}, \mathbf{Z}_{0:k}\right)$, which is assumed approximately

Gaussian and is manipulated by an EKF. From Eqn. 12, the map PDF is a set of M independent 2-D Gaussians rather than a single joint $2M$-D Gaussian. Thus, each particle actually represents: a pose history $\mathbf{X}_{v_{0:k}}^{(i)}$, its weight w_i, and its map $\{\hat{\mathbf{m}}_1, \mathbf{P}_1, \ldots, \hat{\mathbf{m}}_M, \mathbf{P}_M\}$.

Conditioning the map by the vehicle pose *history* is essential for the efficiency of the algorithm; keeping the landmark estimates independent avoids the quadratic cost of computing a joint map covariance matrix. Dependence on pose history is also the key weakness of the FastSLAM algorithm as it means the implicit dimension of the state-space increases with time.

IV. Effects of Resampling

When implementing the FastSLAM algorithm, it is easy to forget that each particle represents a history $\mathbf{X}_{v_{0:k}}^{(i)}$ since the recursive equations at each time-step only require the momentary pose estimate $\mathbf{x}_{v_k}^{(i)}$. However, the dependence on pose history is recorded in the sample weight and, most significantly, in its map estimate. It is important to recognise that a particle's estimate of the map at time k is based on the assumption that the vehicle pose was *exactly known for all preceeding time-steps*. As time progresses, the likelihood of adequately exploring the sample-space becomes vanishingly small. This is seen most clearly in the particle *resampling* step.

A property of particle filters is that the variance of sample weights increases with time [7].The filter degenerates until eventually all samples but one possess negligible weight. To offset this problem, resampling is performed, whereby samples are chosen with replacement from the original sample set, to generate an equal-weight sample set. This has been shown to permit consistent recursive estimation with a fixed number of particles provided a system exhibits "exponential forgetting" of its past estimate errors [6].

The problem with FastSLAM is that past pose estimate errors are not forgotten; they are recorded in the map estimates. Whenever resampling is performed, for each particle *not* selected, an entire pose history and map hypothesis is lost forever. This depletes the number of samples representing past poses and consequently erodes the statistics of the landmark estimates conditioned on these past poses. After resampling, some particles share a common ancestry, and loss of track independence and loss of landmark estimate diversity increases monotonically.

V. Measuring Consistency

Ideally, to measure if a filter is consistent, one would compare its estimate with the probability density function obtained from an ideal Bayesian filter. This is not practical for the FastSLAM algorithm. When the "true" PDF is not available, but the true state $\mathbf{x}_k$ is known, we can use the *normalised estimation error squared* (NEES) [2, page 234] to characterise the filter performance,

$$\epsilon_k = (\mathbf{x}_k - \hat{\mathbf{x}}_k)^T \mathbf{P}_k^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_k) \tag{13}$$

where $\{\hat{\mathbf{x}}_k, \mathbf{P}_k\}$ are the estimated mean and covariance.

A measure of filter consistency is found by examination of the *average* NEES over N Monte Carlo runs of the filter.[2] Under the hypothesis that the filter is consistent and approximately linear-Gaussian, ϵ_k is χ^2 (chi-square) distributed with $\dim(\mathbf{x}_k)$ degrees of freedom. Then the average value of ϵ_k tends towards the dimension of the state as N approaches infinity.

$$E[\epsilon_k] = \dim(\mathbf{x}_k) \tag{14}$$

The validity of this hypothesis can be subjected to a χ^2 acceptance test.

Consistency of FastSLAM is evaluated by performing multiple Monte Carlo runs and computing the average NEES. Given N runs, the average NEES is computed as

$$\bar{\epsilon}_k = \frac{1}{N} \sum_{i=1}^{N} \epsilon_{i_k} \tag{15}$$

Given the hypothesis of a consistent linear-Gaussian filter, $N\bar{\epsilon}_k$ has a χ^2 density with $N \dim(\mathbf{x}_k)$ degrees of freedom. Thus, for the 3-dimensional vehicle pose, with $N = 50$, the 95% *probability concentration region* for $\bar{\epsilon}_k$ is bounded by the interval $[2.36, 3.72]$. If $\bar{\epsilon}_k$ rises significantly higher than the upper bound, the filter is optimistic, if it tends below the lower bound, the filter is conservative.

VI. Experimental Results

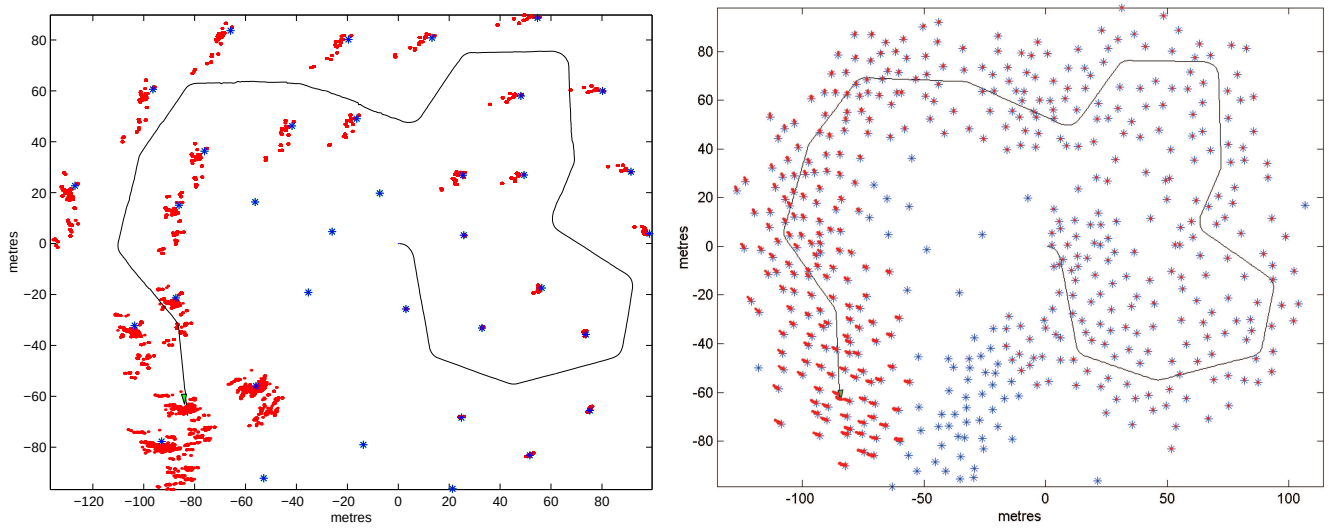
In the following experiments, the vehicle model wheelbase is 4 metres, the control noise is $(\sigma_V = 0.3m/s, \sigma_\gamma = 3^\circ)$, and the observation noise is $(\sigma_r = 0.1m, \sigma_\theta = 1^\circ)$. Controls are updated at 40 Hz and observation scans are obtained every 5 Hz. Each scan consists of range-bearing measurements to all landmarks within a 30 metre radius in front of the vehicle. Data association is assumed known throughout.

Experiments were performed in the two simulated environments in Fig. 1, one sparsely populated and one dense. In each environment, simulations were run with 100 particles and 1000 particles. Resampling was performed, not after each observation, but once the "effective sample size" [7] falls below 75% of the total number of particles. Each run involved two loops of the trajectory partially shown in Fig. 1 and they were each repeated for 50 Monte Carlo trials.

It is immediately apparent from Fig. 1 that the dense map permits more accurate results than the sparse map in terms of real errors; the particles have smaller spread and are nearer the true state. However, FastSLAM's ability to estimate these errors is less intuitive.

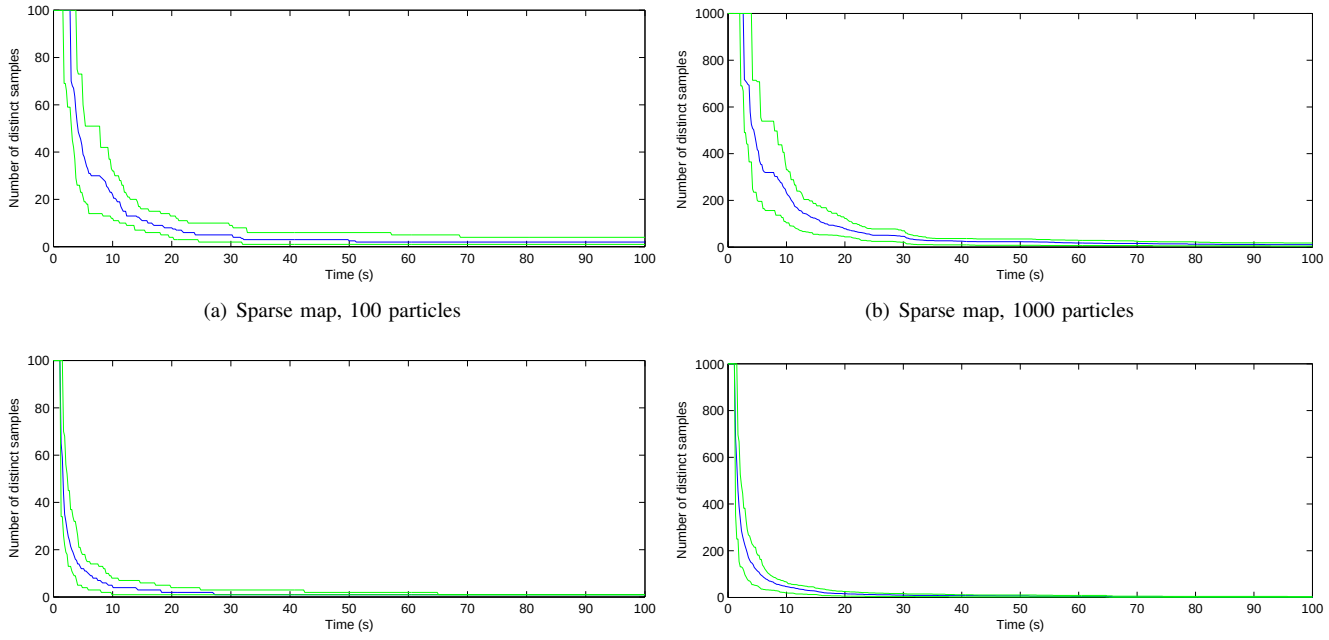
An estimate of the rate of loss of particle diversity is obtained by recording the number of distinct particles representing a chosen landmark. Once the landmark goes out of sight, resampling causes some estimates to be lost and others to be multiplied, and diversity is depleted. Fig. 2 shows that

[2] A commonly used test of filter consistency is to examine the *sequence* of normalised errors $\{\epsilon_0, \ldots, \epsilon_k\}$ over a *single* run. This test is not adequate as the error sequence is correlated and does not follow a χ^2 distribution. Thus, even for a linear system, a single run of a consistent filter may appear inconsistent and a single run of an inconsistent filter may appear consistent.



(a) Sparse map (b) Dense map

Fig. 1. Feature maps used in simulation experiments. The stars denote the true landmark locations. The dots are the particle estimates for vehicle and landmark locations during a typical run with 1000 particles, and the line is the mean estimate of the vehicle trajectory.



(a) Sparse map, 100 particles (b) Sparse map, 1000 particles

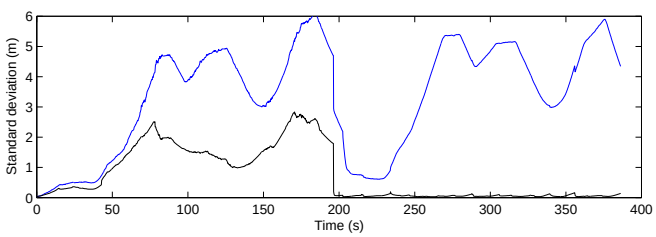
(c) Dense map, 100 particles (d) Dense map, 1000 particles

Fig. 2. Particle diversity for a landmark that is no longer visible. Each figure shows the number of distinct samples representing a single landmark from the moment it disappears from the vehicle's field-of-view. The three lines are the minimum, maximum and median diversity obtained over 50 Monte Carlo runs.

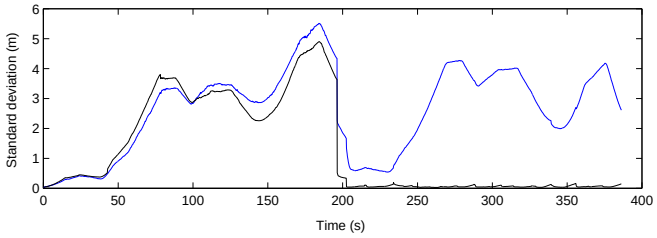
diversity is lost exponentially. That Figs. 2(a) and 2(b) have the same shape indicates that the ratio of distinct samples to total number of samples remains approximately constant for a fixed landmark density; sample size does not affect depletion rate. However, Figs. 2(c) and 2(d) show that the rate of diversity loss increases with landmark density. Thus, counter-intuitively, more observation information implies faster depletion.

A comparison of the real errors and FastSLAM's error estimates is shown in Fig. 3. These figures plot the standard deviations of the x-axis component of the vehicle pose.[3] An empirical estimate is calculated as the standard deviation of $x_{v_k} - \hat{x}_{v_k}$ over 50 Monte Carlo runs, where x_{v_k} is the true state and $\hat{x}_{v_k}$ is the sample mean of the particle filter. The

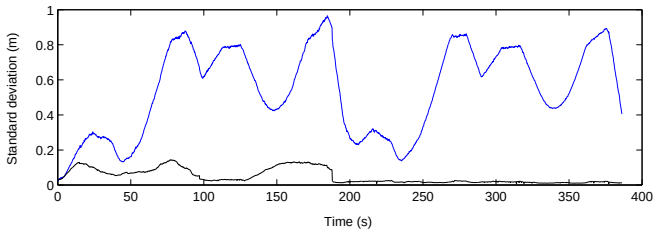
[3]These results tend to make FastSLAM look better than it really is as process noise injects diversity into the pose estimate and maintains its variance. If a landmark state had been chosen instead, the difference between empirical and estimated variance would occur sooner and be more pronounced.



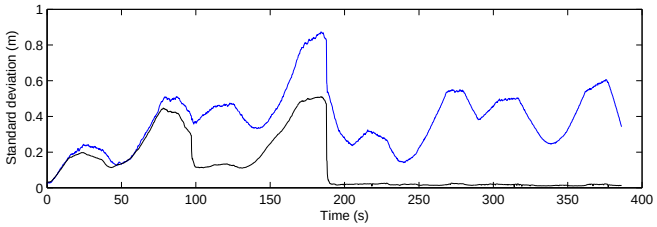
(a) Sparse map, 100 particles



(b) Sparse map, 1000 particles

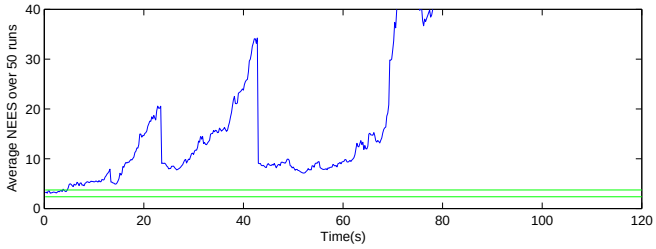


(c) Dense map, 100 particles

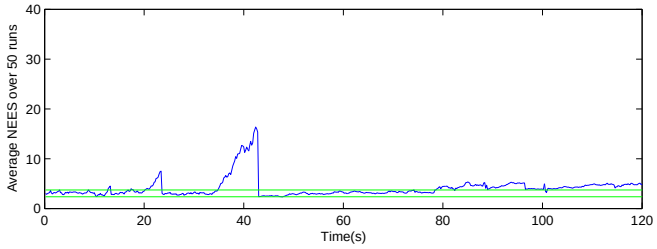


(d) Dense map, 1000 particles

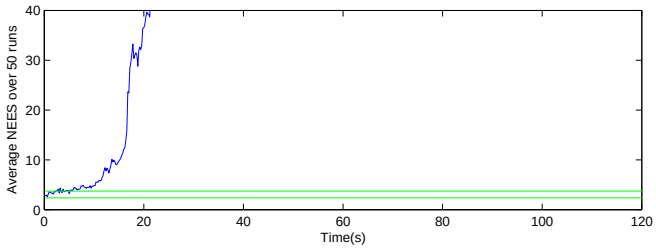
Fig. 3. Estimated variance versus true variance. These figures show the standard deviation in the x-axis component of the vehicle pose estimate. The top line is an empirical calculation obtained from $x_{v_k} - \hat{x}_{v_k}$ over 50 Monte Carlo runs. The lower line is the average particle filter estimate; that is, the square-root of the sample variance averaged over 50 runs.



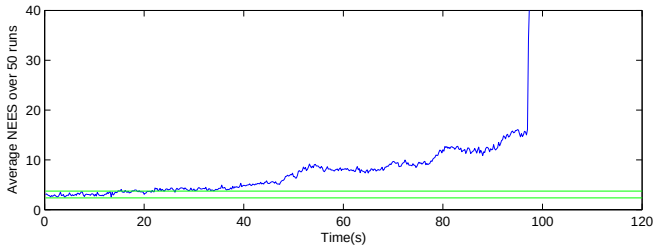
(a) Sparse map, 100 particles



(b) Sparse map, 1000 particles



(c) Dense map, 100 particles



(d) Dense map, 1000 particles

Fig. 4. Average NEES of the vehicle pose states $\mathbf{x}_{v_k}$ over 50 Monte Carlo runs. The two horizontal lines indicate the 95% probability concentration region for a consistent filter.

FastSLAM estimate of uncertainty is the sample variance of the particle filter, $\sigma^2_{x_k}$, and the average of σ_{x_k} over 50 runs indicates the typical result. It is worth noting that the variation in σ_{x_k} from run to run is very large, sometimes slightly greater than the empirical value, sometimes much smaller. This variation is due to the sparseness with which the samples cover the state-space, as is apparent from Fig. 1(a). (For the sake of clarity, the individual run estimates are not shown in Fig. 3.) However, after closing the loop, all runs collapsed to a single particle and produced basically the same variance estimate.

Figs. 3(a) and 3(b) indicate that the number of particles does not greatly affect the real accuracy of the filter, although there is a slight improvement with more samples. However, the increased number of particles does improve the quality of the estimated uncertainty. Comparing these results with Figs. 3(c) and 3(d), we find that a more dense environment produces smaller real errors, but the variance estimate tends to be worse. Thus, FastSLAM can produce quite accurate results but will always underestimate its uncertainty.

The NEES results in Fig. 4 provide a more formal measure of consistency. These results are computed over the vehicle pose states $\mathbf{x}_{v_k}$ and, in each case, show that the filter becomes

rapidly optimistic. General trends are that consistency is prolonged by using more particles but degrades more quickly with higher landmark density. The result in Fig. 4(b) stays reasonably consistent for the longest period but becomes grossly optimistic shortly after the 120 seconds shown.

VII. Discussion

Our results show that the rapid loss of particle diversity prevents a consistent long-term estimate of the joint state PDF. And yet, the quality of the FastSLAM results in the literature (e.g., [12], [9]) indicate that it is quite effective in practice. We would suggest that the accuracy of these results is testament to quality of the sensors used (typically a scanning range laser) rather than to the ability of the FastSLAM algorithm. In essence, FastSLAM provides a non-optimal search, over a finite time-horizon, for the most likely trajectory.

To prolong the time-period over which FastSLAM is reasonably consistent, it is necessary to reduce the impact of resampling. We found that tactics like oversampling and resampling before computing the proposal distribution did not provide any real advantage. One possible improvement might be to replace resampling with *partial rejection control* [11], which draws samples from several steps in the past rather than the preceding time-step. Drawing from a past time-horizon might produce more uniform weighting and slower depletion.

Montemerlo provides a proof showing that FastSLAM 2.0 converges asymptotically in the linear case with only one particle [12]. This is very interesting as one-particle FastSLAM is virtually identical to performing EKF-SLAM while ignoring cross-correlations.[4] Thus, no-correlation EKF-SLAM will converge in the linear case. It is worth considering the pioneering work by Castellanos *et al.* [5], where EKF-SLAM with and without correlations is examined. There is no great difference in real errors between the two forms, but full-correlation EKF produces a consistent estimate of uncertainty and the other does not.

VIII. Conclusion

FastSLAM in its current form cannot produce consistent estimates in the long-term. Each particle implicitly records a pose history in the statistics of its associated map. Every time a particle is lost due to resampling, an entire map hypothesis is lost and there is a depletion in historical information. As a consequence the overall map statistics degrade.

In practice FastSLAM may produce quite accurate results in terms of deviation from the true state. The final quality of this result is dependent on sensor precision. However, FastSLAM's estimate of its accuracy soon becomes optimistic; it tends to underestimate its own uncertainty. In other words, a higher density of landmarks, or equivalently a more precise sensor or more frequent observations, will improve accuracy in terms of *real* errors, but it will also speed up particle depletion. Therefore, in the long-term, FastSLAM is an inconsistent stochastic filter but, as a heuristic (non-stochastic) estimator, where only the mean or mode is valued, it can be both tractable and highly accurate.

In the short-term, FastSLAM might produce consistent results given a sufficient number of particles. It also has practical properties that make it an attractive short-term estimator, particularly the ability to perform an intuitive type of multi-hypothesis data association. One possible use for FastSLAM is as a front-end SLAM component that processes current data and forms a short-term local map, and later converts this map to a joint Gaussian PDF and merges it with a global EKF-SLAM map.

[4]The only difference is that FastSLAM 2.0 introduces a small random jitter into the pose estimate at each step, and so is slightly *less* accurate than its "no-correlation" EKF counterpart.

References

[1] T. Bailey. *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments*. PhD thesis, University of Sydney, Australian Centre for Field Robotics, 2002.

[2] Y. Bar-Shalom, X.R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley and Sons, 2001.

[3] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller. An Atlas framework for scalable mapping. In *IEEE International Conference on Robotics and Automation*, pages 1899–1906, 2003.

[4] G. Casella and C.P. Robert. Rao-blackellisation of sampling schemes. *Biometrika*, 83(1):81–94, 1996.

[5] J.A. Castellanos, J.D. Tardós, and G. Schmidt. Building a global map of the environment of a mobile robot: The importance of correlations. In *IEEE International Conference on Robotics and Automation*, pages 1053–1059, 1997.

[6] D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on Signal Processing*, 50(3):736–746, 2002.

[7] A. Doucet. On sequential simulation-based methods for Bayesian filtering. Technical report, Cambridge University, Department of Engineering, 1998.

[8] A. Doucet, N. de Freitas, and N. Gordon. An introduction to sequential Monte Carlo methods. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*, pages 3–14. Springer-Verlag, 2001.

[9] D. Hähnel, W. Burgard, D. Fox, and S. Thrun. An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 206–211, 2003.

[10] S.J. Julier and J.K. Uhlmann. A counter example to the theory of simultaneous localization and map building. In *IEEE International Conference on Robotics and Automation*, pages 4238–4243, 2001.

[11] J.S. Liu, R. Chen, and T. Logvinenko. A theoretical framework for sequential importance sampling with resampling. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*, pages 225–246. Springer-Verlag, 2001.

[12] M. Montemerlo. *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem With Unknown Data Association*. PhD thesis, Carnegie Mellon University, 2003.

[13] M. Montemerlo and S. Thrun. Simultaneous localization and mapping with unknown data association using FastSLAM. In *IEEE International Conference on Robotics and Automation*, pages 1985–1991, 2003.

[14] J. Neira, J.D. Tardós, and J.A. Castellanos. Linear time vehicle relocation in SLAM. In *IEEE International Conference on Robotics and Automation*, 2003.

[15] J. Nieto, J. Guivant, E. Nebot, and S. Thrun. Real time data association for FastSLAM. In *IEEE International Conference on Robotics and Automation*, pages 412–418, 2003.

[16] P.-J. Nordlund. *Sequential Monte Carlo Filters and Integrated Navigation*. PhD thesis, Linköpings University, Department of Electrical Engineering, 2001.

[17] M.A. Paskin. Thin junction tree filters for simultaneous localization and mapping. Technical report, University of California, Computer Science Division, 2002.

[18] R. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. In *International Symposium of Robotics Research*, pages 467–474, 1987.