Proc. 2002 Australasian Conference on Robotics and Automation
Auckland, 27-29 November 2002

A Comparison between Extended Kalman Filtering and Sequential Monte Carlo Techniques for Simultaneous Localisation and Map-building

David C.K. Yuen and Bruce A. MacDonald
Department of Electrical and Electronic Engineering,
University of Auckland, Private Bag 92019, Auckland, New Zealand.
d.yuen@auckland.ac.nz, b.macdonald@auckland.ac.nz

Abstract

Monte Carlo Localisation has been applied to solve many different classes of localisation problems. In this paper, we present a possible Simultaneous Localisation and Map-building implementation using the Sequential Monte Carlo technique. Multiple particle filters are created to estimate both the robot and landmark positions simultaneously. The proposed technique shows promising results when compared with those obtained with the Extended Kalman filter.

1 Introduction

Simultaneous Localisation And Map-building (SLAM), also known as Concurrent Mapping and Localisation (CML), assumes both the robot and landmark positions are unknown or uncertain. It was introduced originally by Motarlier and Chatila [Motarlier and Chatila, 1989] and Smith *et.al* [R. Smith and Cheeseman, 1988] and has become an active research topic in robotics recently. Both localisation and map building processes can be formulated as estimation problems. One of the main difficulties involved is to overcome the strong coupling between these two estimation tasks: without a reliable map, the robot cannot determine its position precisely; without a good robot position estimate, it is difficult to place the detected landmark onto a map.

Extended Kalman filtering (EKF) is the prevalent SLAM approach and usually offers near real time performance. Dissanayake *et.al* [Dissanayake *et al.*, 2001] proved the convergence properties and steady state behaviour of a Kalman filter based SLAM solution. Since the complexity of the SLAM solution is proportional to $O(N^2)$, with N being the number of landmarks in the map, available computer resources may be exhausted in prolonged operation ("state explosion"). Guivant and Nebot [Guivant and Nebot, 2001] showed an optimal solution ($O(N_a^2)$) can still be obtained by considering only the landmarks on the sub-map (N_a). For an approximated solution, the complexity can further be reduced to $O(N_a)$. In addition to range finders, EKF based methods have also been demonstrated to work with active stereo vision [Davison and Murray, 2002; Davison and Kita, 2001] or with multiple sensor types [Castellanos *et al.*, 2001]. The original Kalman filtering algorithm expects a linear model with Gaussian error sources. While EKF extended the application of Kalman filter to nonlinear systems that modelled well with first order Taylor approximation, it is not applicable for more complicated systems.

Other types of nonlinear stochastic estimation filters have been considered. Thrun *et.al.* [Thrun *et al.*, 1998] addressed the SLAM problem with an expectation-maximisation approach. The Sequential Monte Carlo (SMC) method, also known as particle filtering, provides an attractive simulation-based approach to compute the posterior distribution. It is easy to implement and is not restricted by the linear Gaussian state-space assumption. Monte Carlo Localisation (MCL) refers to the implementation of SMC methods for robot localisation. Many different forms of localisation problems, including position tracking, global localisation and even the difficult kidnapped robot problem[Doucet *et al.*, 2000] have been demonstrated to work under the MCL architecture. Murphy [Murphy, 2000] applied the Rao-Blackwellised particle filter to solve the SLAM problem in a 10×10 grid world.

By initialising multiple particle filters, we extend the MCL SLAM architecture to continuous state-space. The localisation performance of this proposed SMC SLAM algorithm is compared with that of an EKF implementation. It is our belief that, similar to other MCL implementations, the proposed change will improve the robustness of SLAM.

2 Sequential Monte Carlo SLAM

2.1 Monte Carlo Localisation

The goal of SLAM is to estimate both the robot position and landmark locations all at the same time. A review for the more general estimation problem is provided first. For more detailed analysis, please refer to [Doucet *et al.*, 2000, 3–14] and [Arulampalam *et al.*, 2002]. For a system with hidden states $\{x_t; t \in \mathbb{N}\}$, an estimation process can be expressed as the search for the posterior distribution $p(x_{0:t}|y_{1:t})$ and its associated features using the available observations $\{y_t; t \in \mathbb{N}\}$. At any given time, the posterior distribution can be evaluated with the Bayes' Theorem:

$$p(x_{0:t+1}|y_{1:t-1}) = p(x_{0:t}|y_{1:t}) \frac{p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t)}{p(y_{t+1}|y_{1:t})} \quad (1)$$

For mobile robot localisation, the hidden states usually include the 2-D coordinates and the heading direction of the robot. The posterior x_t given the previous observations and user control inputs can be defined as the *belief*,

$$Bel(x_t) = p(x_t|y_t, u_{t-1}, y_{t-1}, \ldots, u_0, y_0). \quad (2)$$

Bayes' Theorem can be re-expressed in terms of $Bel(x_t)$. Applying the Markov assumptions that the observations y_t are conditionally independent with the past, a recursive estimator can be formed, where η is a proportional constant.

$$Bel(x_t) = \eta\, p(y_t|x_t) \int p(x_t|x_{t-1}, u_{t-1}) Bel(x_{t-1})\, dx_{t-1} \quad (3)$$

The MCL algorithm is a realisation of the Bayesian filter. The set of m particles (weighted samples) are defined as:

$$\Psi = \{x^i, f^i\}_{i=1,\ldots,m}. \quad (4)$$

where f is the importance factor associated with each particle. During initialisation, the particles are assigned to random positions in the work space. The importance factors are set to $\frac{1}{m}$.

Algorithm 1 Bayesian filter realisation

for each particle in Ψ **do**
  1. Importance sampling
  2. State update
  3. Measurement update
end for
Importance factor normalisation

As shown in Algorithm 1, the recursive updating filter is implemented in three steps, which effectively evaluates the expression in Equation 3 from right to left. First, a particle i is sampled from Ψ in accordance to the importance factor f_{t-1}. Then, the next state estimate x_t^i and its conditional probability $p(x_t|x_{t-1}, u_{t-1})$ is calculated from the latest control input u_{t-1} and the state variables x_{t-1}^i of the previous particle. Finally, the likelihood of measurement $p(y_t|x_t^i)$ is evaluated. The product of $p(y_t|x_t^i)$, $p(x_t|x_{t-1}, u_{t-1})$ and f_{t-1} yields the importance factor for the next particle. After the updating of all the particles in Ψ, the importance factors are normalised such that $\sum_{i=1}^{n} f_i = 1$.

2.2 An Extension to SLAM

SLAM algorithms are supposed to update the robot and landmark positions concurrently. A possible strategy is to consider the unknown landmark positions as extra hidden states and augment them to the existing particles. Unfortunately, a lot more particles will be required because of the increase in the dimension of the state-spaces. Besides, there is no objective measure to determine the important factors after the addition of the extra states.

Our SLAM implementation is similar to the standard MCL method introduced by Fox *et.al* [Fox *et al.*, 2000]. The individual particle filters are the same as that mentioned in Algorithm 1. The localisation and landmark position estimation problems are slightly different in nature. Localisation can be considered as a task similar to tracking while finding the location of the landmarks is more similar to the parameter estimation problem. Separated sets of particle filters are therefore introduced.

Algorithm 2 outlines our SMC SLAM algorithm. Extra landmark handling stages are added. Similar to MCL, a nonlinear Bayesian particle filter Ψ_0 with m particles is initialised to track the robot position. Since SLAM makes no assumption to any external reference frame, rather than spreading uniformly across the whole work space, it is reasonable to assign the particles randomly around an arbitrary starting point.

The measurement data is examined at the start of each time step. A new n-particle filter Ψ_L, where L is the total number of new landmarks under consideration, is initialised whenever a new landmark is detected. These particle filters $\Psi_i, i = 0, \ldots, L$ are then updated one after another with Algorithm 1. The order of updating is unimportant, which makes parallel implementation in the future possible. The presence of dynamic objects has not been explicitly considered in this study. Since the landmarks are assumed to be static, the associated particles may be discarded once after a reasonable position estimate has been obtained. The distribution within the landmark position estimation filters are examined after each complete round of filter update. If the distribution is sufficiently consistent, the expected state of the particles is likely to be the correct landmark position. The expected state of the filter will be taken as the landmark location and stored permanently before removing the filter. The particle removal procedure not only reduces computa-

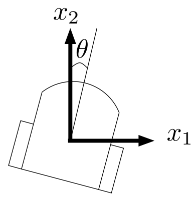

Figure 1: The convention for the coordinate system

tion, but also stabilises the estimation. The sensors equipped with the robot usually have limited range. If we do not store the learned landmarks, the estimation may drift away from the correct value due to the lack of further observations when the landmarks are outside the detection zone of the robot.

Algorithm 2 SMC SLAM

Initialise filter Ψ_0 for robot position
$L = 0$ {L=total number of new landmarks}
repeat
  if new landmark is identified **then**
    $L = L + 1$
    Initialise new filter Ψ_L
  end if
  Update all the filters $(\Psi_{i,i=0,\ldots,L})$
  for all $\Psi_{i,i=1,\ldots,L}$ **do**
    Check for possible removal of Ψ_i
  end for
until the last time step

To compute the likelihood of the measurements $p(y_t|x_t)$, the current robot and landmark positions are required. It implies the close relationship between the particle filters. In SLAM, neither the robot nor the landmark positions can be assumed as completely reliable at any time. A probabilistic sampling approach is thus adopted. When updating the particle filter Ψ_j for the j-th new landmark (or the robot position if $j = 0$), a random particle is picked from each $\Psi_{i,i=0,\ldots,L;i\neq j}$. The state variables of these particles are taken as the respective robot or landmark positions. It provides the necessary coupling between these dependent variables and is an important part for the algorithm.

2.3 Implementation details

The system dependent implementation details are covered in this section. To evaluate the SMC SLAM approach, a simple simulated environment is adopted. The obstacle boundaries are approximated by point objects.

The simulated robot has 2 degrees-of-freedom, (move forward/backward and "turn-on-the-spot"). The 2-D coordinates x_1, x_2 and the heading direction θ represent the unobserved states of the system (Figure 1). The kinematic equations for the system can be expressed as follow:

$$\phi_{t-1} = \frac{\pi}{2} - \theta_{t-1} \tag{5}$$

$$x_{1t} = x_{1(t-1)} + u_{1(t-1)} \cos\phi_{t-1} \tag{6}$$

$$x_{2t} = x_{2(t-1)} + u_{1(t-1)} \sin\phi_{t-1} \tag{7}$$

$$\theta_t = \theta_{t-1} + u_{2(t-1)} \tag{8}$$

where u_1 and u_2 are the control inputs for the linear and angular motions respectively. The state variables are corrupted by independent zero mean Gaussian noise streams.

The robot is equipped with a ring of 24 ultrasonic-like range finders. The work space is expected to be larger than the maximum range of the sensors. As illustrated in the last section, a particle is sampled from each filter so that current robot and landmark position estimates can be extracted to build the perceptual model. The measured obstacle distance is affected by two independent error sources: a Gaussian measurement noise component and an exponential component that represents more catastrophic events such as multiple reflection or cross talk between sensors. The joint probability density function between two independent sources is the convolution of the original sources. For a mixed noise source Z with Gaussian G and exponential E noise components, $Z = G + E_1 - E_2$, the probability density function $f_z(z)$ is

$$\Phi(v) = \int_{-\infty}^{v} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}u^2\right) du$$

$$f_z(z) = \frac{\lambda}{2} \exp\left(\frac{\sigma^2\lambda^2}{2}\right)[\exp(-\lambda z)(1 - \Phi(-\frac{z - \sigma^2\lambda}{\sigma})) + \exp(\lambda z)\Phi(-\frac{z + \sigma^2\lambda}{\sigma})] \tag{9}$$

The expected state $\bar{x_t}$ can be computed from the kinematic equations (Equation 5–8). A Gaussian distributed random number is added to each state variable. The probabilities of these independent events are multiplied to give the transition probability $p(x_t|x_{t-1}, u_{t-1})$. The expected measurement $\bar{y_t}$ is then calculated using the perceptual model. The difference between actual and expected measurements, y_t and $\bar{y_t}$, is substituted into Equation 9 to give the measurement likelihood $p(y_t|x_t)$.

A few simple landmark management rules are established to determine the number of particle filters that needs to be maintained by the SMC SLAM algorithm. First, we consider a filter addition rule. A new landmark particle filter is initialised if the discrepancy between the actual and expected measurements is greater than a certain threshold. In this study, the expected robot position is calculated from Ψ_0 at the start of each time step; the expected measurements are given by the perceptual model. Two filter removal rules are then introduced. If a genuine landmark is detected, the particle distribution of the filter will become more compact after a few recursive updates. The landmark position can be confirmed when the standard deviation of the particles reduces to below a certain threshold. The expected landmark position will be recorded before the removal of the filter. Random error sometimes triggers the estimation for new landmark. The distribution of these particles usually does not converge. The associated particle filter will be removed after monitored for a few more updates.

3 EKF SLAM

EKF is the most common SLAM approach. It extends the application of Kalman filtering to nonlinear systems by applying linear Taylor approximation to the state-space equations. It should be noted that, unlike the original Kalman filter, EKF is not an optimal estimator. The recursive updating procedure of EKF is shown as follow:

- state update

$$x_{k|k-1} = f_k(x_{k-1}) \tag{10}$$

$$P_{k|k-1} = F_k P_{k-1} F_k^T + Q_{i-1} \tag{11}$$

- measurement update

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} + R_{k-1})^{-1} \tag{12}$$

$$x_k = x_{k|k-1} + K_i(y_k - H_k x_{k|k-1}) \tag{13}$$

$$P_k = (I - K_k H_k) P_{k|k-1} \tag{14}$$

where P is the covariance matrix of the states. K is the Kalman filter gain. $f_k()$ and $h_k()$ are the state and measurement equations; F_k and H_k are the local linearisation of these functions.

In most estimation problems, the number of parameters to be evaluated is constant. Due to obstacle occlusion and the limited range of the sensors, extra landmarks may appear when the robot navigates to different parts of the environment. A landmark addition rule is established for this EKF SLAM implementation.

For each state variable vector, the first 3 elements represent the current robot pose x_1, x_2, θ. The landmark position estimates fill up the rest of the state vector x. After the state updating stage, the expected measurement $y_{k|k-1} = h_k(x_{k|k-1})$ is calculated.

A new landmark is assumed to be found if the difference between the expected $y_{k|k-1}$ and actual measurement y_k is larger than a certain threshold. The current robot position estimates and sensor readings give the initial estimate $x_{1(LM)}, x_{2(LM)}$ for the new landmark. The state variable and covariance matrix now become $[x_{k|k-1}, x_{1(LM)}, x_{2(LM)}]$ and $[P_{k|k-1}, 0; 0, P_{LM0}]$, where P_{LM0} is a diagonal matrix with arbitrary values that initialises the submatrix corresponding to the covariance of the new states.

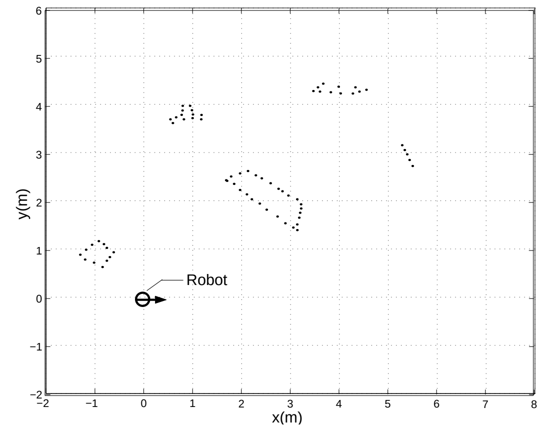

Figure 2: The landmark distribution in the simulation environment.

4 Results and Discussions

The simulation environment is shown in Figure 2. The obstacles are outlined by point series. Mixed Gaussian and exponential measurement noise sources were applied to the system initially. Since this violated the Gaussian noise Kalman filtering assumption, the EKF SLAM failed as expected. To provide a fairer comparison between EKF and SMC SLAM, only independent zero-mean Gaussian ($\sigma = 0.1$) process and measurement noise sources were added instead.

The robot is placed onto the map without knowing any landmark features in each simulation. For the SMC method, 300 particles are allocated to estimate each of the new landmark position. Another 500 particles are created for robot position tracking. The actual and estimated robot trajectories for EKF and SMC SLAM are shown in Figure 3a and b. The corresponding position and orientation errors are compared in Figure 4. The SMC method performs better throughout the test. EKF and SMC SLAM are approximated Bayesian filtering methods. Depending on the system complexity, the first order Taylor approximation of the EKF method may no longer hold. The nonlinearity introduced by the landmark management rules is also a concern. While the simulation results are by no means conclusive, it indicates the potential of SMC for SLAM applications.

The map-building process is not as successful when compared with the localisation module. As shown in Figure 5, many landmarks are placed at incorrect positions when using SMC SLAM. The robot sometimes fails to recognise some of the already detected landmarks after moves to a new position. Redundant particle filters are created to estimate these landmarks again. Due to localisation error, the duplicated landmarks will appear with an offset from the originals. Clearly, it is one of the aspects that needs to be improved. In addition, the existing implementation should also be extended to handle the more realistic polygonal obstacle model.

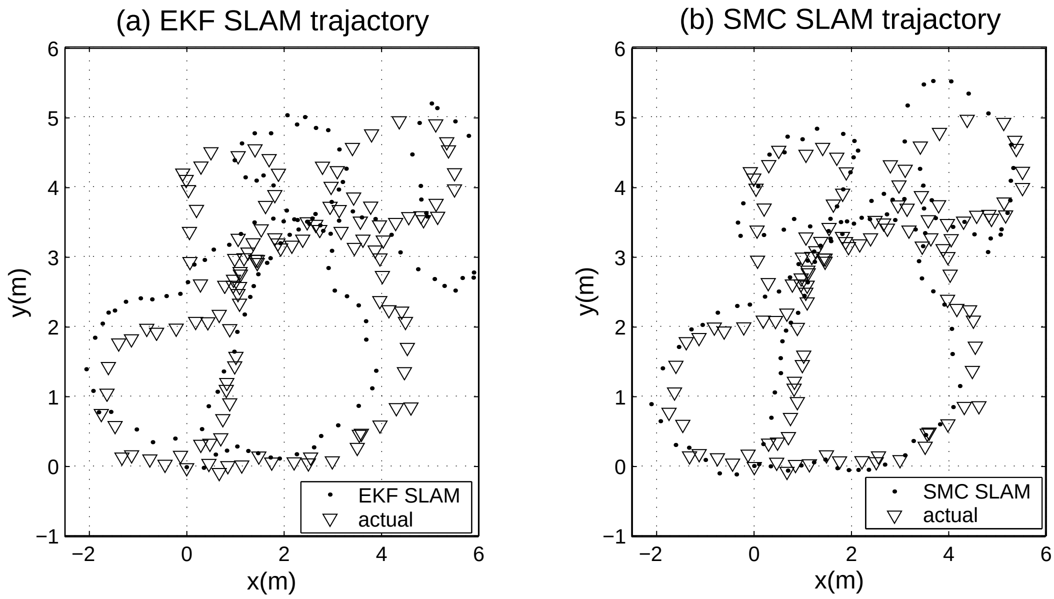

Figure 3: The actual and estimated robot trajectories for EKF and SMC SLAM.

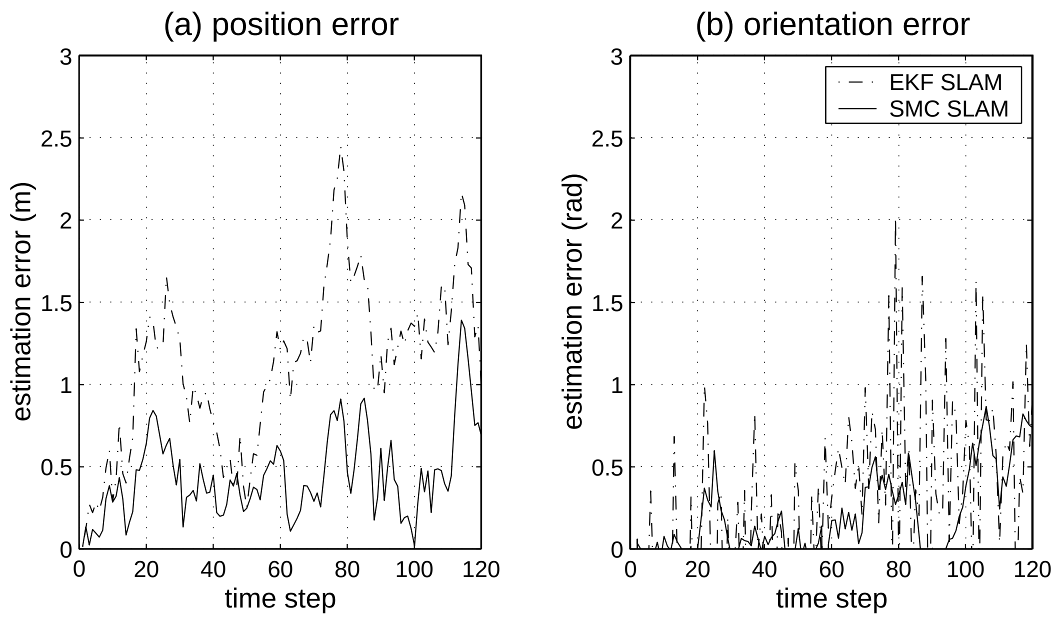

Figure 4: A comparison between the positioning and orientation error of EKF and SMC SLAM.

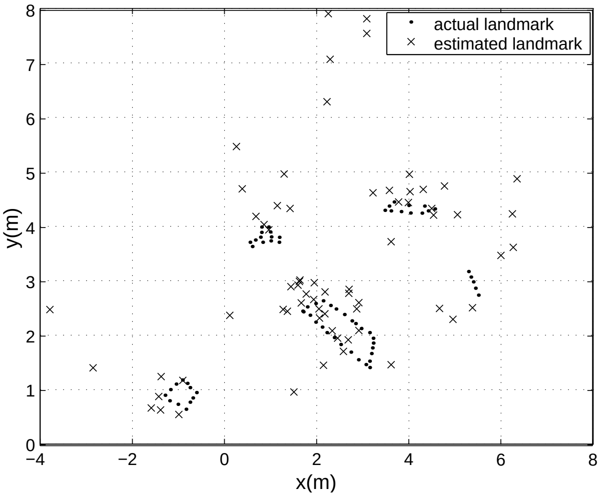

Figure 5: Actual and estimated landmark (with SMC SLAM) positions.

5 Conclusion

A possible SMC SLAM algorithm has been outlined in this paper. The simulated results from EKF and SMC SLAM algorithms are compared. While system complexity and implementation details affect the performance of these algorithms, the preliminary results suggest the feasibility of the SMC SLAM idea. The algorithm will be tested more thoroughly on a real robot in the near future.

Acknowledgements

This work was supported in part by the Foundation for Research, Science and Technology, New Zealand, with a Top Achiever Doctoral Scholarship.

References

[Arulampalam *et al.*, 2002] M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, 50(2), 2002.

[Castellanos *et al.*, 2001] J.A. Castellanos, J. Neira, and J.D. Tardos. Multisensor fusion for simultaneous localization and map building. *IEEE Transactions on Robotics and Automation*, 17(6):908–914, 2001.

[Davison and Kita, 2001] Andrew Davison and Nobuyuki Kita. 3d simultaneous localisation and map building using active vision for a robot moving on undulating terrain. In *CVPR*, 2001.

[Davison and Murray, 2002] A.J. Davison and D.W. Murray. Simultaneous localization and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligenc*, 24(7):865–880, 2002.

[Dissanayake *et al.*, 2001] M.W.M.G. Dissanayake, P. Newman, S. Clark, and M. Csorba H.F. Durrant-Whyte. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001.

[Doucet *et al.*, 2000] Arnaud Doucet, Nando de Freitas, and Neil Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, 2000.

[Fox *et al.*, 2000] Dieter Fox, Sebastian Thrun, Wolfram Burgard, and Frank Dellaert. *Sequential Monte Carlo Methods in Practice*, chapter Particle filters for mobile robot localization. Springer, 2000.

[Guivant and Nebot, 2001] J.E. Guivant and E.M. Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, 2001.

[Motarlier and Chatila, 1989] P. Motarlier and R. Chatila. Stochastic multi-sensory data fusion for mobile robot location and environmental modeling. In *Fifth Symp. Robot. Res.*, pages 85–94, Tokyo, 1989.

[Murphy, 2000] K.P. Murphy. Bayesian map learning in dynamic environments. In *Advances in Neural Information Processing System*, volume 12, pages 1015–1021. MIT Press, 2000.

[R. Smith and Cheeseman, 1988] M. Self R. Smith and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Uncertainty in Artificial Intelligence*, volume 2, pages 435–461, New York, 1988. Elsevier Science.

[Thrun *et al.*, 1998] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31(1-3), 1998.