

# Unscented FastSLAM: A Robust Algorithm for the Simultaneous Localization and Mapping Problem

Chanki Kim, R. Sakthivel and Wan Kyun Chung

**Abstract**—Rao-Blackwellized Particle Filter and FastSLAM have become popular tools to solve the Simultaneous Localization and Mapping (SLAM) problem. However, the above techniques have two important potential limitations, which are the derivation of the Jacobian matrices and the linear approximations to the nonlinear functions. Also, one of the major challenges of both Rao-Blackwellized particle filter and FastSLAM is to reduce the number of particles while maintaining the estimation accuracy. This paper proposes a new algorithm based on unscented transformation called Unscented FastSLAM (UFastSLAM) that overcomes important drawbacks of the Rao-Blackwellized particle filter frameworks by directly using nonlinear relations. Experimental results in large scale environments are presented that demonstrate the effectiveness of the UFastSLAM algorithm over the previous approaches.

## I. INTRODUCTION

The simultaneous localization and mapping is the problem to determine the position and the heading of an autonomous vehicle moving through an unknown environment and, simultaneously, to estimate features of the environment. In the past decade, a lot of researchers focused on this problem.

Recently, the estimation algorithms can be roughly classified according to their underlying basic principle. The most popular approaches to SLAM problem are the Extended Kalman Filter (EKF-SLAM) and the Rao Blackwellized Particle Filter (RBPF-SLAM) [1]. The effectiveness of the EKF approach comes from the fact that they estimate a fully correlated posterior over feature maps and vehicle poses. However, the EKF-SLAM assumes linearization of the motion and measurement models and Gaussianity for the probability density functions, so a consistency analysis and an improvement of the classical algorithm have been reported [2]–[4]. In [5], Murphy introduced the RBPF as an effective approach to solve the SLAM problem by decoupling the state of the vehicle and the map. Montemerlo *et al.* [6], [7] developed a framework called the FastSLAM for landmark based maps.

One of the major challenges of both the RBPF and the FastSLAM is to reduce the number of particles while maintaining the estimation accuracy. Additionally, the resampling step of the particle filter can eliminate the correct particles [8]. The noticeable one is that both the RBPF and the FastSLAM linearize the motion and the measurement model in the same manner as the EKF-SLAM. As a result, unfortunately, the EKF based RBPF frameworks have two important potential drawbacks which are the derivation of

the Jacobian matrices and the linear approximations to the nonlinear functions. Inherently, inaccurate approximations of the nonlinear function should be solved since it can lead to filter divergence [2], [4], [9].

Martinez-Cantin and Castellanos [2] introduced the Unscented Kalman Filter (UKF) [10]–[12] in SLAM problem for large scale outdoor environments. This approach is used to avoid the analytical linearization based on Taylor series expansion of both the motion and the measurement model. Merwe *et al.* [8] introduced the Unscented Particle Filter (UPF) algorithm for a novel proposal distribution in the sampling step. The unscented filter estimates the mean and the covariance by a linear weighted regression of evaluation of the true nonlinear models at the *sigma points*. As proved in the previous researches, the number of *sigma points* precisely and accurately approximates the posterior covariance up to the third order, whereas the EKF relies on a first order approximation. More recently, Li *et al.* [13] introduced the UKF based RBPF for SLAM problem using a monocular vision but they did not apply the UKF based RBPF in a proper way<sup>1</sup>.

In this paper, a new framework called the Unscented FastSLAM (UFastSLAM) is developed and demonstrated to overcome the important limitations of the RBPF frameworks. The UFastSLAM consists of the UKF for feature update and the Unscented Transformation (UT) for feature initialization and the modified UPF for improved sampling of the vehicle state.

We observe experimentally that the proposed UFastSLAM algorithm, even with 3 particles, yield significantly more accurate results compared with the FastSLAM2.0. And this improvement is salient with large uncertainty.

We begin Section II by describing the UFastSLAM algorithm. In this section, we present the measurement update, initialization of the feature, then how the proposal distribution is calculated using the unscented filtering. At the end of this section, we introduce the resampling technique. In Section III, the results are presented to demonstrate the effectiveness of the proposed algorithm. The results validate that the UFastSLAM shows the better performance. Section IV contains the conclusion and the future works.

<sup>1</sup>Some invalid parameter values are used to deterministically generate *sigma points*. In particular, the selected value for  $\alpha = 0$  explicitly reveals that the value of parameter  $\lambda = -n$ . This leads (2) is invalid and consequently the further steps are in question. Moreover, the authors claimed that UKF is used for feature update. The way in which they applied UKF for feature update is obviously wrong. Just UT is used for approximating the nonlinear measurement function and subsequently the new mean and covariance are obtained using the classical EKF technique.

Chanki Kim, R. Sakthivel and Wan Kyun Chung are with the Department of Mechanical Engineering, Pohang University of Science and Technology, Korea {minekiki, Sakthi, wkchung}@postech.ac.kr

## II. BACKGROUND: FASTSLAM FRAMEWORK

The FastSLAM algorithm, introduced by Montemerlo *et al.* [7] is an efficient algorithm for the SLAM problem that is based on a straightforward factorization. This algorithm partitions SLAM posterior into a localization problem and independent landmark positions estimation problem conditioned on the vehicle pose estimate. The conditional independence property of the SLAM problem enables us to estimate the posterior in the following factored form:

$$p(x^t, \theta | z^t, u^t, n^t) = p(x^t | z^t, u^t, n^t) \prod_{k=1}^N p(\theta_k | x^t, z^t, u^t, n^t).$$

Here the path of the vehicle is denoted by  $x^t = x_1, \dots, x_t$ ,  $z^t = z_1, \dots, z_t$  is a sequence of measurements and  $u^t = u_1, \dots, u_t$  is a sequence of control inputs.  $n^t = n_1, \dots, n_t$  are data association variables, in which each  $n_t$  specifies the identity of the landmark observed at time  $t$ . Each landmark is denoted by  $\theta_k$  for  $k = 1, \dots, N$  and  $\theta$  means the set of all landmarks.

FastSLAM uses a particle filter to approximate the ideal recursive Bayesian filter for estimating the vehicle pose. The remaining posterior of feature locations are analytically calculated by using the EKF filters. So the FastSLAM algorithm is a Rao-Blackwellized particle filter. Each particle in the FastSLAM is of the form

$$X_t^{[m]} = \langle x_t^{[m]}, \mu_{1,t}^{[m]}, \Sigma_{1,t}^{[m]}, \dots, \mu_{N,t}^{[m]}, \Sigma_{N,t}^{[m]} \rangle$$

where  $[m]$  indicates the index of the particle,  $x_t^{[m]}$  is its path estimate, and  $\mu_{k,t}^{[m]}$  and  $\Sigma_{k,t}^{[m]}$  are the mean and covariance of the Gaussian, representing the  $k$ -th landmark location that is attached to the  $m$ -th particle. In FastSLAM1.0, new poses are sampled using the most recent motion command  $u_t$ :

$$x_t^{[m]} \sim p(x_t | x_{t-1}^{[m]}, u_t).$$

It is important to note that this proposal distribution uses only the motion control  $u_t$ , but ignores the current measurement  $z_t$ . So the FastSLAM1.0 approach is particularly troublesome if the observation is too accurate relative to the vehicle's motion noise. To overcome this problem, an improved version called FastSLAM2.0 is proposed by Montemerlo *et al.* In FastSLAM2.0, the vehicle poses are sampled under consideration of both the control  $u_t$  and the measurement  $z_t$  which is denoted by the following sampling distribution

$$x_t^{[m]} \sim p(x_t | x_{t-1}^{[m]}, u_t, z^t, n^t)$$

and, as a result, the FastSLAM2.0 is superior to the FastSLAM1.0 in all aspects [17].

In the FastSLAM algorithm, each particle does not represent a single momentary vehicle pose. Rather, it represents an entire vehicle path history and the path history is recorded in the map estimates. The map is represented as a set of independent Gaussians, with linear time-complexity, rather than a joint map covariance with quadratic time-complexity. These are key properties of the FastSLAM and are the reason for its computational speed. FastSLAM offers

some improvements over the traditional EKF-based SLAM framework. In addition to the obvious reduction in computational complexity, the technique intrinsically provides a way of maintaining multi-hypothesis data association. This is an important advantage, given that a single-hypothesis data association error can cause the EKF-based strategy to fatally diverge. In FastSLAM, particles with incorrect data association eventually “die” because of their low importance weights, thereby reducing the impact of such errors.

## III. UNSCENTED FASTSLAM

### A. Feature Update and Initialization using the UKF in the FastSLAM framework

The UKF is more accurate than the EKF in the aspect of nonlinear approximation accuracy and does not employ Jacobian matrices for calculating feature covariance. Instead of approximating the measurement nonlinear function by Taylor series expansion, the UKF deterministically extracts so-called *sigma points* from the Gaussian and passes these through the nonlinear function. These *sigma points* are defined using previous mean and covariance of features.

$$\begin{aligned} \chi^{[0][m]} &= \mu_{n_t, t-1}^{[m]} \\ \chi^{[i][m]} &= \mu_{n_t, t-1}^{[m]} + \left( \sqrt{(n + \lambda) \Sigma_{n_t, t-1}^{[m]}} \right)_i, (i = 1, \dots, n) \\ \chi^{[i][m]} &= \mu_{n_t, t-1}^{[m]} - \left( \sqrt{(n + \lambda) \Sigma_{n_t, t-1}^{[m]}} \right)_i, (i = n+1, \dots, 2n) \end{aligned} \quad (1)$$

Here  $n$  is the dimension of feature state ( $n = 2$ ).  $\lambda = \alpha^2(n + \kappa) - n$ , with  $\alpha = 0.01$  and  $\kappa = 0$  being scaling parameters that determine how far the sigma points are separated from the mean. Because landmarks on a planar environment have only 2 degrees of freedom, so in the feature part, it is enough to use 2 dimensional Gaussian.  $\mu_{n_t, t-1}^{[m]}$  is the mean of the  $n$ -th feature which is registered in previous step.  $\chi^{[i][m]}$  is the *sigma points* on Cartesian coordinate.  $[m]$  means an index of particles.  $\Sigma_{n_t, t-1}^{[m]}$  is the covariance matrix of the  $n$ -th feature, and it has 2 by 2 dimension in the FastSLAM framework.

The UKF requires computation of a matrix square root which can be implemented directly using the Cholesky factorization. However, the covariance matrices have low dimensions and can be expressed recursively. So, not only does the UKF outperform the EKF in accuracy and robustness, it does so at no extra computational cost [8].

A weight  $w_g^{[i]}$  is used when computing the mean and the weight  $w_c^{[i]}$  is used when recovering the covariance of the Gaussian. These weights are calculated by

$$\begin{aligned} w_g^{[0]} &= \frac{\lambda}{(n + \lambda)}, w_c^{[0]} = \frac{\lambda}{(n + \lambda)} + (1 - \alpha^2 + \beta) \\ w_g^{[i]} &= w_c^{[i]} = \frac{1}{2(n + \lambda)} \quad (i = 1, \dots, 2n) \end{aligned} \quad (2)$$

The parameter  $\beta = 2$  can be chosen to encode additional knowledge about the distribution underlying the Gaussian representation. The predicted measurement  $\hat{z}_t^{[m]}$  and the

Kalman gain  $K_t^{[m]}$  is calculated as follows

$$\begin{aligned}\bar{Z}_t^{[i][m]} &= \mathbf{h}(\chi^{[i][m]}, x_t^{[m]}) \quad (i = 0, \dots, 2n) \\ \hat{z}_t^{[m]} &= \sum_{i=0}^{2n} w_g^{[i]} \bar{Z}_t^{[i][m]} \\ S_t^{[m]} &= \sum_{i=0}^{2n} w_c^{[i]} \left( \bar{Z}_t^{[i][m]} - \hat{z}_t^{[m]} \right) \left( \bar{Z}_t^{[i][m]} - \hat{z}_t^{[m]} \right)^T + R_t \\ \bar{\Sigma}_t^{[m]} &= \sum_{i=0}^{2n} w_c^{[i]} \left( \chi^{[i][m]} - \mu_{n_t, t-1}^{[m]} \right) \left( \bar{Z}_t^{[i][m]} - \hat{z}_t^{[m]} \right)^T \\ K_t^{[m]} &= \bar{\Sigma}_t^{[m]} \left( S_t^{[m]} \right)^{-1}\end{aligned}\quad (3)$$

Here,  $\mathbf{h}(\cdot)$  is the observation model.  $\bar{Z}_t^{[i][m]}$  is the *sigma points* on the polar coordinate made by nonlinear transformation. In this observation model, the current vehicle state of  $m$ -th particle  $x_t^{[m]}$  is used. We assume that the measurement noise covariance  $R_t$  is additive.  $K_t^{[m]}$  is the Kalman gain calculated from pure nonlinear transformation. This is more accurate than the EKF. Finally the mean  $\mu_{n_t, t}^{[m]}$  and the covariance  $\Sigma_{n_t, t}^{[m]}$  of the  $n$ -th feature is updated by

$$\mu_{n_t, t}^{[m]} = \mu_{n_t, t-1}^{[m]} + K_t^{[m]} \left( z_t - \hat{z}_t^{[m]} \right) \quad (4)$$

$$\Sigma_{n_t, t}^{[m]} = \Sigma_{n_t, t-1}^{[m]} - K_t^{[m]} S_t^{[m]} \left( K_t^{[m]} \right)^T \quad (5)$$

$z_t$  is the true measurement. We can use the Cholesky factorization in this feature update to make the algorithm more stable numerically.

In the feature initialization, the current measurement  $z_t$  and measurement noise covariance  $R_t$  compose the *sigma points*  $\psi^{[i][m]}$ . In the FastSLAM and the RBPF framework, the feature initialization is separated from the vehicle state estimator. This means that it is possible to use the measurement noise covariance as an initial feature covariance [6], [7].

Finally, the feature mean  $\mu_{n_t, t}^{[m]}$  and the feature covariance  $\Sigma_{n_t, t}^{[m]}$  in the feature initialization are calculated by

$$\begin{aligned}\psi^{[0][m]} &= z_t \\ \psi^{[i][m]} &= z_t + \left( \sqrt{(n+\lambda)R_t} \right)_i, \quad (i = 1, \dots, n) \\ \psi^{[i][m]} &= z_t - \left( \sqrt{(n+\lambda)R_t} \right)_i, \quad (i = n+1, \dots, 2n) \\ \bar{M}_t^{[i][m]} &= \mathbf{h}^{-1}(\psi^{[i][m]}, x_t^{[m]}), \quad (i = 0, \dots, 2n) \\ \mu_{n_t, t}^{[m]} &= \sum_{i=0}^{2n} w_g^{[i]} \bar{M}_t^{[i][m]} \\ \Sigma_{n_t, t}^{[m]} &= \sum_{i=0}^{2n} w_c^{[i]} \left( \bar{M}_t^{[i][m]} - \mu_{n_t, t}^{[m]} \right) \left( \bar{M}_t^{[i][m]} - \mu_{n_t, t}^{[m]} \right)^T\end{aligned}$$

### B. Sampling Particles using the modified UPF in the FastSLAM framework

The particle filter in the FastSLAM and the RBPF framework rely on the importance sampling and, as a result,

requires the design of proposal distributions that can approximate the true posterior reasonably well. The most common strategy is to sample from the motion model. This strategy can, however, fail if the new measurements appear in the tail of the proposal distribution or if the likelihood is too sharp in comparison to the proposal distribution.

Several researchers introduced the current observation into the proposal distribution and used their own techniques, as a result, could make the upgraded proposal distribution which is almost same with the true posterior [7], [15], [16]. However, the Jacobian and inaccurate linear approximation still exist on the covariance related part. The proposed algorithm reducing linearization error in the proposal distribution shows more accurate result than the FastSLAM2.0 in both the vehicle and the feature estimation. The resulting filter performs better accuracy and robustness, but it does not use the Jacobian and linear approximation.

In this section, modified UPF is used to sample the particles from the improved proposal distribution [8].

Since the observation is not always detected, constructing the proposal distribution and sampling from this prior have two phases. One is a prediction step, another is a measurement update step. At first, the state vector is augmented with a control input and the observation

$$\bar{x}_{t-1}^{a[m]} = \begin{bmatrix} \bar{x}_{x, t-1}^{[m]} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \bar{x}_{x, t-1}^{[m]} \\ \bar{x}_{y, t-1}^{[m]} \\ \bar{x}_{\theta, t-1}^{[m]} \\ \mathbf{0} \end{bmatrix}, P_{t-1}^{a[m]} = \begin{bmatrix} P_{t-1}^{[m]} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & Q_t & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & R_t \end{bmatrix} \quad (6)$$

$\bar{x}_{t-1}^{a[m]}$  is the augmented vector for the state and  $\bar{x}_{t-1}^{[m]}$  is the previous mean of the vehicle.  $Q_t$  and  $R_t$  are the control noise covariance and the measurement noise covariance respectively. The augmented covariance matrix  $P_{t-1}^{a[m]}$  has 7 by 7 dimension. A symmetric set of  $2L + 1$  *sigma points*  $\chi_{t-1}^{a[i][m]}$  for the augmented state vector with  $L = 7$  is given by

$$\begin{aligned}\chi_{t-1}^{a[0][m]} &= \bar{x}_{t-1}^{a[m]} \\ \chi_{t-1}^{a[i][m]} &= \bar{x}_{t-1}^{a[m]} + \left( \sqrt{(L+\lambda)P_{t-1}^{a[m]}} \right)_i, \quad (i = 1, \dots, L) \\ \chi_{t-1}^{a[i][m]} &= \bar{x}_{t-1}^{a[m]} - \left( \sqrt{(L+\lambda)P_{t-1}^{a[m]}} \right)_i, \quad (i = L+1, \dots, 2L)\end{aligned}$$

Here  $\alpha = 0.0001$ ,  $\kappa = 0$  and  $\beta = 2$  are enough for estimating the vehicle pose. Each *sigma point*  $\chi_{t-1}^{a[i][m]}$  contains the state, control, and measurement components given by

$$\chi_{t-1}^{a[i][m]} = \begin{bmatrix} \chi_{t-1}^{[i][m]} \\ u^{[i][m]} \\ z^{[i][m]} \end{bmatrix} \quad (7)$$

The motion model is characterized by a nonlinear function and the set of *sigma points*  $\chi_{t-1}^{a[i][m]}$  are transformed by the

motion model  $\mathbf{f}$ , using the control  $u_t^{[m]}$  with the added control noise component  $\chi_t^{u[i][m]}$  of each *sigma point*.

$$\bar{\chi}_t^{[i][m]} = \mathbf{f}(u_t^{[m]} + \chi_t^{u[i][m]}, \chi_{t-1}^{[i][m]}) = \begin{bmatrix} \bar{\chi}_{x,t}^{[i][m]} \\ \bar{\chi}_{y,t}^{[i][m]} \\ \bar{\chi}_{\theta,t}^{[i][m]} \end{bmatrix} \quad (8)$$

Here,  $\bar{\chi}_t^{[i][m]}$  is the transformed vehicle pose and on the sigma domain. The first two moments of the predicted vehicle state are computed by a weighted summation of the transformed *sigma points*  $\bar{\chi}_t^{[i][m]}$ , and the weights are calculated as in the previous step using (2).

$$\bar{x}_{t|t-1}^{[m]} = \sum_{i=0}^{2L} w_g^{[i]} \bar{\chi}_t^{[i][m]}$$

$$P_{t|t-1}^{[m]} = \sum_{i=0}^{2L} w_c^{[i]} \left( \bar{\chi}_t^{[i][m]} - \bar{x}_{t|t-1}^{[m]} \right) \left( \bar{\chi}_t^{[i][m]} - \bar{x}_{t|t-1}^{[m]} \right)^T$$

As some features are observed, data association provides its identity and (9) ~ (15) are employed to update the estimated mean  $\bar{x}_t^{[m]}$  and the covariance  $P_t^{[m]}$  of the vehicle. The measurement *sigma points*  $\bar{N}_t^{[i][m]}$  are calculated using the observation model  $\mathbf{h}$ , characterized by a nonlinear function, with the added measurement noise component  $\chi_t^{z[i][m]}$ .

$$\bar{N}_t^{[i][m]} = \mathbf{h}(\bar{\chi}_t^{[i][m]}) + \chi_t^{z[i][m]} \quad (9)$$

$$\hat{n}_t^{[m]} = \sum_{i=0}^{2L} w_g^{[i]} \bar{N}_t^{[i][m]} \quad (10)$$

$$S_t^{[m]} = \sum_{i=0}^{2L} w_c^{[i]} \left( \bar{N}_t^{[i][m]} - \hat{n}_t^{[m]} \right) \left( \bar{N}_t^{[i][m]} - \hat{n}_t^{[m]} \right)^T \quad (11)$$

$$\Sigma_t^{x,n[m]} = \sum_{i=0}^{2L} w_c^{[i]} \left( \bar{\chi}_t^{[i][m]} - \bar{x}_{t|t-1}^{[m]} \right) \left( \bar{N}_t^{[i][m]} - \hat{n}_t^{[m]} \right)^T \quad (12)$$

$$K_t^{[m]} = \Sigma_t^{x,n[m]} \left( S_t^{[m]} \right)^{-1} \quad (13)$$

Here,  $\hat{n}_t^{[m]}$  is the predicted measurement and  $S_t^{[m]}$  is the innovation covariance. The cross-covariance  $\Sigma_t^{x,n[m]}$  corresponds to the Jacobian term of the EKF algorithm.  $K_t^{[m]}$  is the Kalman gain in the measurement update.

Notice that we do not use the measurement noise covariance  $R_t$  as an additive term for calculating the innovation covariance  $S_t^{[m]}$ . This is because the measurement noise is already included in the augmented covariance (6) and considered to calculate the predicted measurement  $\hat{n}_t^{[m]}$ . In a different way, one can define the augmented state vector and the augmented covariance (6) using the state and control inputs and, rather than containing the measurements in the augmentation, can handle the measurement noise covariance as an additive term in (11). However, it is important to note that this can lead to a negative definite of the covariance of the vehicle  $P_t^{[m]}$  in a very small measurement noise with multiple measurements.

The estimated mean and its covariance at time  $t$  are calculated by:

$$\bar{x}_t^{[m]} = \bar{x}_{t|t-1}^{[m]} + K_t^{[m]} \left( z_t - \hat{n}_t^{[m]} \right) \quad (14)$$

$$P_t^{[m]} = P_{t|t-1}^{[m]} - K_t^{[m]} S_t^{[m]} \left( K_t^{[m]} \right)^T \quad (15)$$

From the Gaussian distribution generated by the estimated mean and covariance of the vehicle, the state of each particle is sampled.

$$x_t^{[m]} \sim \mathcal{N}(\bar{x}_t^{[m]}, P_t^{[m]}) \quad (16)$$

Because the next augmented state  $\bar{x}_t^{a[m]}$  employs the previous mean  $\bar{x}_{t-1}^{[m]}$ , so the updated mean should be memorized during the iterative process. Multiple measurements are obtained frequently. In this case, (9) ~ (15) are repeated according to each feature, and the mean and the covariance of the vehicle are updated based on the previously updated one.

### C. Selective Resampling

One of the major influences on the performance of the particle filter is the resampling step. The particles with a low importance weight are replaced by samples with a high weight during the resampling. For the resampling technique we apply the selective resampling strategy based on the number of effective particles [15].

## IV. RESULTS

Bailey *et al.* developed the SLAM simulator and opened to the public on the web-site [18]. By using this simulator, the comparison of the different SLAM algorithms is permitted and Bailey, Nieto and Nebot [17] discussed an accuracy of the FastSLAM2.0 using this simulator. We also used the same simulator and the FastSLAM2.0 algorithm which are used in [17].

Before comparing the UFastSLAM with the FastSLAM2.0, we introduce two important versions of the modified FastSLAM2.0 which are the ‘UKF aided FastSLAM2.0’ and the ‘modified UPF aided FastSLAM’.

### A. Preliminary Experiments

The preliminary experiment is executed on a rectangular-shaped trajectories of 100 meters x 20 meters as shown in Fig. 1. The vehicle has 0.26 meters wheel base and is equipped with a range-bearing sensor with a maximum range of 20 meters and a 180 degrees frontal field-of-view. Gaussian noise covariances are generated for both the measurement and the motion. The control noise is  $\sigma_V = 0.1m/s$ ,  $\sigma_\gamma = 3^\circ$ . A control frequency is 40 Hz and observation scans are obtained every 5Hz. Data association is assumed known. The measurement noise is 0.2m in range and  $8^\circ$  in bearing. We used only 3 particles. The feature and the vehicle states are predicted and updated by the pure nonlinear transformation in the UFastSLAM which outperforms the FastSLAM2.0 as shown in Fig. 1.

Each bar in Fig. 2 represents the mean of the estimated error of the vehicle pose. The mean and variance of the MSE

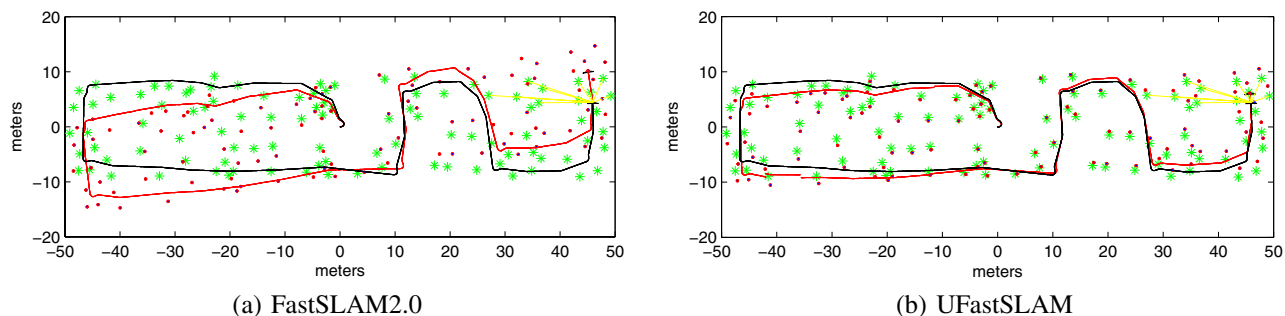


Fig. 1. Estimated and true vehicle path with estimated and true landmarks. The black line and green stars denote the true path and landmark positions, respectively. The red line is the mean estimate of the vehicle and the red dots are estimated landmarks. The measurement noises are relatively large ( $\sigma_r = 0.2\text{m}$ ,  $\sigma_\phi = 8^\circ$ ) and only 3 particles are used.

are calculated over 10 independent runs for each algorithm. Additionally, only 3 particles are used. The ‘UKF aided FastSLAM2.0’ employs the sampling technique same as the FastSLAM2.0 but feature states are updated using the UKF and initialized by the UT. In Fig. 2 and Tab. I, the estimation accuracy of the ‘UKF aided FastSLAM2.0’ (B in Fig. 2) is increased comparing with the FastSLAM2.0 (A in Fig. 2). This is because more accurate transformation operates upon the estimator rather than linear approximation of the nonlinear function. However, one can find that the reduction of the MSE is only 1 meter. This means that the effect of the unscented filter in the ‘UKF aided FastSLAM2.0’ (B in Fig. 2) is not so large. This is because the feature covariance has low dimension.

The ‘modified UPF aided FastSLAM’ (C in Fig. 2) consists of the unscented filter based sampling technique and the EKF feature update. The most important factor of the particle filter approach is the sampling process. The FastSLAM framework is also particle filter based estimator, and as a result, the novel sampling technique affects the performance of the estimator, directly. In this case, we can observe the greatly increased performance compared with the ‘UKF aided FastSLAM2.0’ (B in Fig. 2).

Finally, the proposed algorithm UFastSLAM (D in Fig. 2) shows the most accurate result comparing with the other 3 algorithms (FastSLAM2.0, UKF aided FastSLAM2.0 and

modified UPF aided FastSLAM) as shown in Fig. 2.

### B. Performance Comparison Between UFastSLAM and FastSLAM2.0

In this section, we consider three types of simulation experiments to demonstrate the effectiveness of the proposed algorithm, UFastSLAM.

The first set of experiments has been designed to validate the effect of the measurement uncertainty. In this experiment, 10 particles were used with various measurement noise levels in the relatively small environment (20 meters by 25 meters). The mean and variance of the MSE are calculated over 5 independent runs for each algorithm. As the measurement noise is increased, the estimation errors of the FastSLAM2.0 and the UFastSLAM algorithms are increased as shown in Fig. 3. However, the increasing rate of the estimation error of the FastSLAM2.0 is almost twice than the UFastSLAM and the variance of the MSE of the FastSLAM2.0 is larger than the UFastSLAM. So we can conclude that the UFastSLAM is more robust to the sensor noise than the FastSLAM2.0.

In the second experiment, the number of particles of the UFastSLAM is considered to obtain the same estimation accuracy with the FastSLAM2.0. A size of the environment is about 100 meters by 100 meters and the vehicle traveled about 500 meters. Tab. II presents the mean squared errors and the number of particles. Although the UFastSLAM uses one-fifth of the particles of the FastSLAM2.0, the estimated error is almost the same as the FastSLAM2.0.

Finally, we compared the performance of both UFastSLAM and FastSLAM2.0 algorithms with an unknown data association and with the same number of particles. The individual compatibility nearest neighbor test with the  $2\sigma$  acceptance region is used for the data association. At a position ‘G’ of Fig. 4(a), we notice that the vehicle again observed the landmark A, B and C but the data association is failed because of the large heading errors, and at a position ‘H’ of Fig. 4(a), the landmark B and C are mismatched again. Consequently, the filter is diverged and is failed to close a first loop. On the other hand, the UFastSLAM, in Fig. 4(b), estimates both the vehicle and the feature states accurately by avoiding the analytical linearization of the nonlinear equations. The accurate estimation induced a

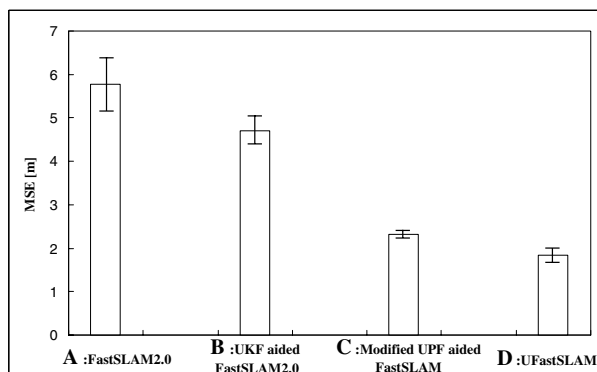


Fig. 2. Comparisons of the developed algorithms with FastSLAM2.0.

TABLE I  
PRELIMINARY EXPERIMENTS

SLAM algorithm (with 3 particles)	MSE [m]		heading errors [deg]
	mean	var	
FastSLAM2.0	5.783	0.500	-6.6927
UKF aided FastSLAM2.0	4.705	0.352	-5.4442
Modified UPF aided FastSLAM	2.322	0.001	-2.9198
Unscented FastSLAM	1.831	0.076	-2.4861

sequence of successful data associations of the observed landmarks A, B and C. Then further, consider landmarks D, E and F in Fig. 4(b), which contribute to the first loop closing event and we observe that the first loop is closed successfully. Since the Kalman gain of the feature update (3) and the Kalman gain in the proposal distribution (13) are calculated from the pure nonlinear transformation, the UFastSLAM compensates the angular uncertainty robustly as shown in Fig. 4(b) comparing with Fig. 4(a).

## V. CONCLUSIONS

In this paper, we proposed the UFastSLAM algorithm which is an improved framework to solve the SLAM problem efficiently. The main advantage of this algorithm is that it does not use the derivation of the Jacobian matrices and the linear approximations to the nonlinear functions in the RBFP framework.

In the proposed algorithm, the feature state and the covariance are updated by the UKF to avoid the linearization errors and the Jacobian calculations. For sampling the particles, the UT is implemented in the prediction step of the vehicle state, and the unscented filter is employed to obtain the better proposal distribution to overcome linearization errors and the Jacobian calculations in the measurement update step of the vehicle state. Due to this, the accuracy of the state

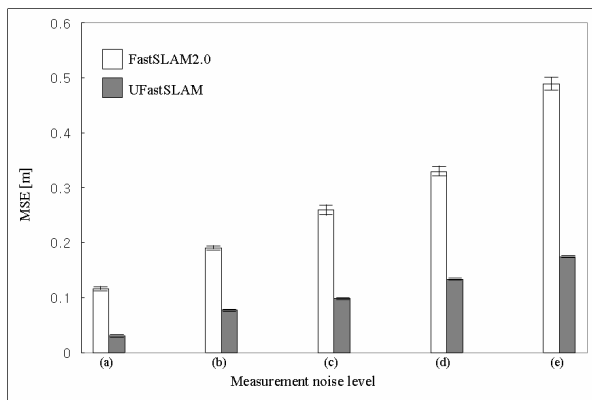


Fig. 3. The effectiveness of UFastSLAM according to large measurement noises. Each bar means its mean of the estimated vehicle pose. The mean and variance of the MSE are calculated over 5 independent runs for each SLAM algorithm. 10 particles are used in small environment (20m x 25m). Measurement noises of each experiment are (a)  $\sigma_r = 0.1, \sigma_\phi = 3^\circ$  (b)  $\sigma_r = 0.2, \sigma_\phi = 6^\circ$  (c)  $\sigma_r = 0.2, \sigma_\phi = 8^\circ$  (d)  $\sigma_r = 0.3, \sigma_\phi = 10^\circ$  (e)  $\sigma_r = 0.3, \sigma_\phi = 14^\circ$ . The loop closure with known data association and a long range of an observation (20 meters) induced an absolutely small MSEs in entire experiments.

TABLE II  
SECOND EXPERIMENT

SLAM algorithm	Number of particles	MSE[m]
		(size of environment, traveling)
FastSLAM2.0	50	1.6815 (100m x 100m, $\approx$ 500m)
UFastSLAM	10	1.6963 (100m x 100m, $\approx$ 500m)

estimation has been improved over the previous approaches. Also this technique has the additional advantage of reducing the number of particles while maintaining the estimation accuracy.

We conclude from the results in large scale environments with various sensor noise uncertainties that the proposed UFastSLAM algorithm, even with less particles, yields significantly more accurate and robust results compared with the FastSLAM2.0.

Future work considers on resampling technique for improving the consistency of the Rao-Blackwellized particle filters.

## REFERENCES

- [1] H. Durrant-Whyte and T. Bailey, "Simultaneous Localization and Mapping: Part I", *IEEE Robotics and Automation Magazine*, vol. 13 (2), 2006, pp 99-108.
- [2] R. Martinez-Cantin and J. A. Castellanos, "Unscented SLAM for Large-Scale Outdoor Environments", *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 2005, pp 328-333.
- [3] S. Huang and G. Dissanayake, "Convergence Analysis for Extended Kalman Filter Based SLAM", *IEEE Intl. Conf. on Robotics and Automation*, 2006, pp 412-417.
- [4] R. Martinez-Cantin and J. A. Castellanos, "Bounding Uncertainty in EKF-SLAM: The Robocentric Local Approach", *IEEE Intl. Conf. on Robotics and Automation*, 2006, pp 430-435.
- [5] K. Murphy, "Bayesian Map Learning in Dynamic Environments", *Neural Info. Proc. Systems (NIPS)*, MIT Press, 1999.
- [6] M. Montemerlo and S. Thrun, "Simultaneous Localization and Mapping with Unknown Data Association using FastSLAM", *IEEE Intl. Conf. on Robotics and Automation*, 2003, pp 1985-1991.
- [7] M. Montemerlo, "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association", *PhD thesis*, Carnegie Mellon University, 2003.
- [8] R. Merwe, A. Doucet, N. Freitas and E. Wan, "The Unscented Particle Filter", *Technical Report CUED/F-INFENG/TR 380*, Cambridge University Engineering Department, 2000.
- [9] Y. Bar-Sahlom, X. R. Li and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, John Wiley and Sons, 2001.
- [10] S. Julier, J. K. Uhlmann and Durrant-Whyte, "A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators", *IEEE Trans. on Automatic Control*, vol. 45 (3), 2000, pp 477-482.
- [11] S. Julier and J. K. Uhlmann, "A Counter Example to the Theory of Simultaneous Localization and Map Building", *IEEE Intl. Conf. on Robotics and Automation*, 2001, pp 4238-4243.
- [12] S. Julier and J. K. Uhlmann, "Unscented Filtering and Nonlinear Estimation", *Proceeding of the IEEE*, vol. 92 (3), 2004, pp 401-422.
- [13] M. Li, B. Hong, Z. Cai and R. Luo, "Novel Rao-Blackwellized Particle Filter for Mobile Robot SLAM using Monocular Vision", *Intl. Journal of Intelligent Technology*, vol. 1 (1), 2006, pp 63-69.
- [14] S. Thrun, W. Burgard and D. Fox, *Probabilistic Robotics*, The MIT Press, 2005.
- [15] G. Grisetti, C. Stachniss and W. Burgard, "Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposal and Selective Resampling", *IEEE Intl. Conf. on Robotics and Automation*, 2005, pp 2443-2448.
- [16] P. Beeson, A. Murarka and B. Kuipers "Adapting Proposal Distributions for Accurate, Efficient Mobile Robot Localization", *IEEE Intl. Conf. on Robotics and Automation*, 2006, pp 49-55.

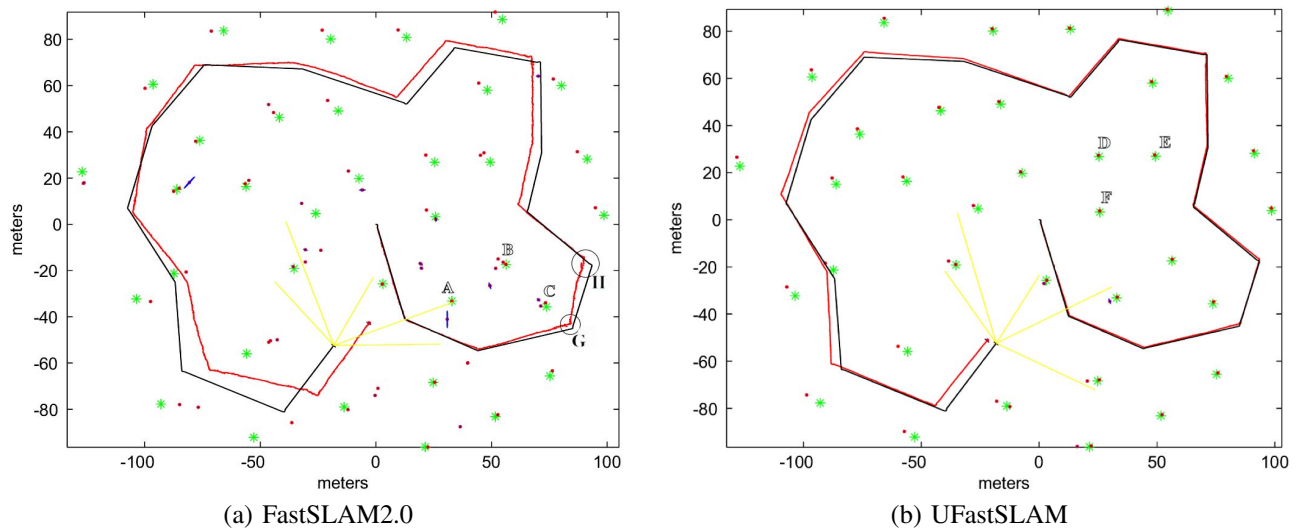


Fig. 4. Unknown data association. Estimated and true vehicle path with estimated and true landmarks. The black line and green stars denote the true path and landmark positions, respectively. The red line is the mean estimate of the vehicle and the red dots are estimated landmarks. In this experiment, the control noise is  $\sigma_V = 0.3m/s$ ,  $\sigma_\gamma = 3^\circ$ . The measurement noises are relatively large ( $\sigma_r = 0.2m$ ,  $\sigma_\phi = 8^\circ$ ) and only 3 particles are used for both FastSLAM2.0 and UFastSLAM algorithms.

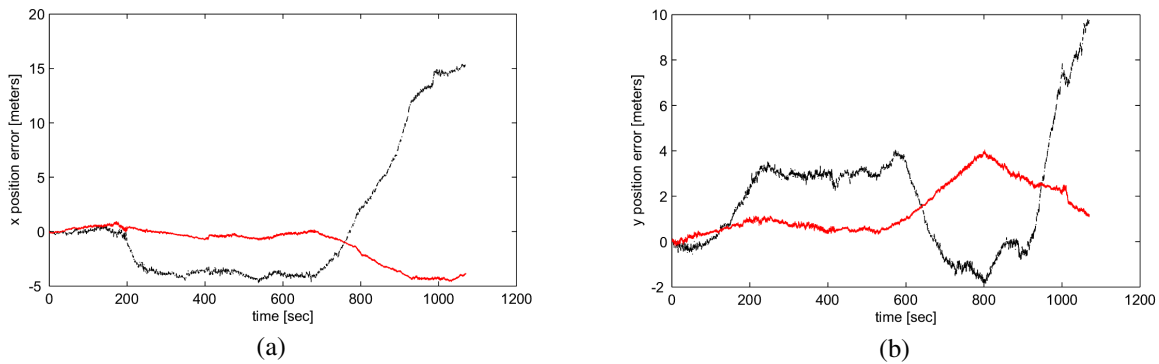


Fig. 5. Estimated errors of the vehicle. The dashed line represent FastSLAM2.0 and the solid (red) line represent UFastSLAM: (a) x position errors and (b) y position errors

[17] T. Bailey, J. Nieto and E. Nebot, "Consistency of the FastSLAM Algorithm", *IEEE Intl. Conf. on Robotics and Automation*, 2006, pp 424-429.

[18] <http://www.acfr.usyd.edu.au/homepages/academic/tbailey/software/index.html>