The Eighteenth International Symposium on Artificial Life and Robotics 2013 (AROB 18th '13),
Daejeon Convention Center, Daejeon, Korea, January 30-February 1, 2013

A new solution to the SLAM problem by using an unscented smoother

Keigo Watanabe[1], Satoshi Funamoto[2], and Shoichi Maeyama[1]

[1] Department of Intelligent Mechanical Systems, Division of Industrial Innovation Sciences,
Graduate School of Natural Science and Technology, Okayama University
3-1-1 Tsushima-naka, Kita-ku, Okayama 700-8530, Japan
Tel & Fax: +81-86-251-8064
E-mail: {watanabe, maeyama}@sys.okayama-u.ac.jp
[2] Shin Nippon Koki Co., Ltd., 4-1, Kita-Kyuhoji-Machi 2-chome, Chuo-ku, Osaka, Japan

Abstract: As a solution to the SLAM problem, the extended Kalman filter or unscented Kalman filter (UKF) is often used up to now. In the case of an offline use of estimated results, a fixed-interval smoother is available and it is expected to give much more high accuracy. In this paper, a solution to the SLAM problem is proposed with the unscented Rauch-Tung-Striebel (RTS) smoother (URTSS) and several experimental results are given to show the improvement of the estimation accuracy due to the present method. In particular, the superiority of our method over the conventional UKF based method is demonstrated by evaluating the estimated accuracy of both methods through some simulations using a mobile robot.

Keywords: SLAM, Unscented Kalman filter, Unscented RTS smoother, Mobile robots.

1 INTRODUCTION

For the solution to the Simultaneous Localization and Mapping (SLAM) problem [1], [2], two kinds of information are generally used for the estimation method. One is the self-position obtained from internal sensors such as encoders and inertial sensors. This information is called "control inputs". The other is relative position to the landmarks obtained from external sensors such as camera, laser range finder, etc. This information is called "measurements". However, the both information include noises and the exact position is not acquired. Therefore, probabilistic methods to integrate the both are applied for localization and map building.

Up to now, as a solution to the SLAM problem, extended Kalman filter (EKF) that is one of probabilistic estimation methods, is mainly used. However, the EKF has two problems: one is less-accuracy in the case of the system with any strong nonlinearity because of linearization by Taylor expansion, and the other is that the derivative of model must be calculated for linearization. To overcome these weak points, Julier et al. [3] proposed a novel method so-called Unscented Kalman Filter (UKF). The UKF based on using Unscented Transform (UT) excludes the derivative of model to improve the problems of the EKF and can estimate the state of nonlinear systems accurately.

The SLAM solutions using the EKF and the UKF are available online estimation. That is, it is possible to estimate the latest self-position and the optimal map while capturing the data. On the other hand, it is sometimes considered that there needs a precise trajectory and map information after completing the exploration. For instance, in the case of the exploration of planetary or disaster environments, the offline estimation is applicable.

In the case of offline use of estimation results, a fixed-interval smoother is available and expected to give much higher accuracy. A Rauch–Tung–Striebel (RTS) smoother, which is one of fixed interval smoothers proposed by Rauch et al. [4], was applied to various applications. Recent years, Terejanu et al. [5] and Särkkä [6]proposed an unscented RTS smoother (URTSS) with UT in order to apply it to a model that has strong nonlinearities.

The SLAM problem is one of nonlinear estimation problems so that the URTSS is expected to contribute the improvement of the accuracy on this problem. Therefore, in this paper, the solution to the SLAM problem with the URTSS is proposed and the experimental results are given to demonstrate the effectiveness of the present method by evaluating the estimated accuracy through some simulations.

2 URTSS–SLAM

The objective in the URTSS–SLAM is to simultaneously estimate the self-position and direction of the robot and the landmark position, which are described in a composite state vector $\boldsymbol{x}_t$ such as

$$\begin{aligned}\boldsymbol{x}_t =&(x,\, y,\, \theta,\, \mu_{1,x},\, \mu_{1,y},\, \dots \\ &\mu_{j,x},\, \mu_{j,y},\, \dots,\, \mu_{N,x},\, \mu_{N,y})^T\end{aligned} \tag{1}$$

where x, y and θ are the robot position and the direction (or azimuth), which are called the posture collectively in this paper. $\mu_{j,x}$ and $\mu_{j,y}$ denote the absolute position of the j-th landmark, and the N denotes the number of landmarks.

The action model used here and the measurement model are respectively described by

$$\boldsymbol{x}_t = g(\boldsymbol{u}_{t-1} + \boldsymbol{\rho}_{t-1},\, \boldsymbol{x}_{t-1}) \tag{2}$$
$$\boldsymbol{z}_t = h(\boldsymbol{x}_t) + \boldsymbol{\varepsilon}_t \tag{3}$$

where $\boldsymbol{u}_{t-1}$ is the "control" input. The action model transfers the state $\boldsymbol{x}_{t-1}$ to $\boldsymbol{x}_t$ through the action function g and the control input is affected by the noise $\boldsymbol{\rho}_{t-1}$. The measurement model produces the measurement $\boldsymbol{z}_t$ through the measurement function h and this model is also affected by the noise $\boldsymbol{\varepsilon}_t$.

In this paper, the estimate for $\boldsymbol{x}_t$ is represented by using the mean $\boldsymbol{m}_t$ and the covariance $\boldsymbol{P}_t$. The URTSS–SLAM is manly composed by the following two-steps:

Table 1. Simulation conditions

Meaning	Variable and parameter	Value	Unit
Simulation time	T	76	[s]
Sampling interval	Δt	0.5	[s]
Control noise const.	a_1	0.02	
	a_2	0.02	
	a_3	0.02	
	a_4	0.02	
Measurement noise S.D. for relative distance	σ_r	0.05	[m]
Measurement noise S.D. for relative angle	σ_ϕ	2.0	[deg]

1. Using the UKF-SLAM algorithm for the time interval $[0\ \ T]$, memorize the estimates for the self-position and direction of the robot and for the landmark position.

2. Re-estimate the states using the URTSS–SLAM algorithm backward in time from time T.

The UT technique and both the UKF–SLAM and the URTSS-SLAM algorithms are not described here, but see e.g., Terejanu et al. [5] and Särkkä [6] for the detailed explanations.

3 SIMULATIONS

Simulation experiments are presented to check the estimation accuracy of the proposed method, where the present URTSS–SLAM is compared with the conventional UKF–SLAM.

3.1 Simulation conditions

Assume that the robot navigates with a constant velocity along the path depicted in Fig. 1, where the robot moves from the coordinate $(0,\ 0)$ counterclockwise and the landmarks are assigned every 1.3 [m] in a 4 [m]×6.5 [m] rectangular area. The estimation accuracy is evaluated with 100 trials, where the action and measurement models are given by the previous sections. It is assumed that the measurement rage of the landmarks is restricted such as ± 120 [deg] measured from the centerline of the direction of movement and 4 [m] in maximum detection distance. Furthermore, following the results of Julier et al. [3] and Kim et al. [7], the parameters used in the UT were set to $\alpha = 0.002, \beta = 2$, and $\kappa = 3 - n$, where note that other simulation conditions are shown in Table 1.

3.2 Action model

An odometric model [8] is used for the action model, where the input to the model is the amount of the robot posture change obtained from the odometry. Note that the odometry is a measurement rigorously, so that velocity information should be added to the state, if the odometry is used as a measurement. If it is regarded as an input in general, then it is possible to reduce the number of states. The present paper follows such an approach.

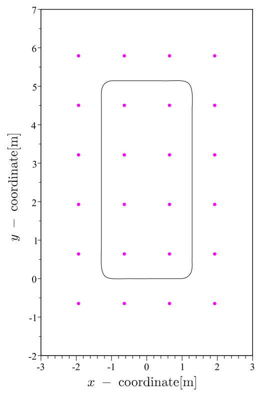

Fig. 1. Simulation environment

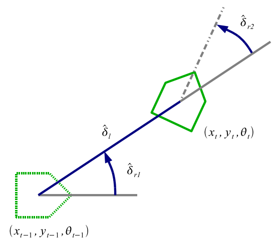

Fig. 2. Motion Model

The robot posture is defined as follows. Let us consider a robot moving on a plane, and its posture is described by

$$\boldsymbol{p} = \begin{pmatrix} x & y & \theta \end{pmatrix}^T \tag{4}$$

where x and y are the robot position, and θ is the azimuth of the robot. Assume that the robot moved from $\boldsymbol{p}_{t-1}$ to $\boldsymbol{p}_t$. Then, the odometry returns noisy $\hat{\boldsymbol{p}}_{t-1} = \begin{pmatrix} \hat{x}_{t-1} & \hat{y}_{t-1} & \hat{\theta}_{t-1} \end{pmatrix}^T$ and $\hat{\boldsymbol{p}}_t = \begin{pmatrix} \hat{x}_t & \hat{y}_t & \hat{\theta}_t \end{pmatrix}^T$. The control input can be obtained as a noisy value that includes noises as shown in Fig. 2, which is given by

$$\hat{\boldsymbol{u}}_{t-1} = \begin{pmatrix} \hat{\delta}_{r1,\,t-1} & \hat{\delta}_{l,\,t-1} & \hat{\delta}_{r2,\,t-1} \end{pmatrix}^T \tag{5}$$

Here, each component of the control input $\hat{\boldsymbol{u}}_{t-1}$ is as follows: $\hat{\delta}_{r1,\,t-1}$ is the amount of rotation before moving, $\hat{\delta}_{l,\,t-1}$ is the amount of movement, and $\hat{\delta}_{r2,\,t-1}$ is the amount of rotation after moving. $\hat{\boldsymbol{u}}_{t-1}$ can be then calculated as below:

$$\hat{\boldsymbol{u}}_{t-1} = \begin{pmatrix} \mathrm{atan2}(\hat{y}_t - \hat{y}_{t-1},\, \hat{x}_t - \hat{x}_{t-1}) - \hat{\theta}_{t-1} \\ \sqrt{(\hat{x}_t - \hat{x}_{t-1})^2 + (\hat{y}_t - \hat{y}_{t-1})^2} \\ \hat{\theta}_t - \mathrm{atan2}(\hat{y}_t - \hat{y}_{t-1},\, \hat{x}_t - \hat{x}_{t-1}) \end{pmatrix} \tag{6}$$

Equation (2) representing the action model is reduced to

$$\begin{aligned} \boldsymbol{x}_t &= g(\boldsymbol{u}_{t-1} + \boldsymbol{\rho}_{t-1},\, \boldsymbol{x}_{t-1}) \\ &= g(\hat{\boldsymbol{u}}_{t-1},\, \boldsymbol{x}_{t-1}) \\ &= \boldsymbol{x}_{t-1} + \begin{pmatrix} \hat{\delta}_{l,\,t-1} \cos(\theta_{t-1}) \\ \hat{\delta}_{l,\,t-1} \sin(\theta_{t-1}) \\ \hat{\delta}_{r1,\,t-1} + \hat{\delta}_{r2,\,t-1} \\ \boldsymbol{0} \end{pmatrix} \end{aligned} \tag{7}$$

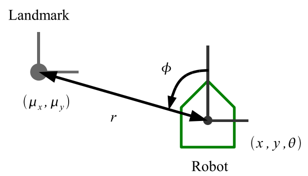

Fig. 3. Measurement model

in which $\boldsymbol{\rho}_{t-1}$ is the action noise defined by

$$\boldsymbol{\rho}_{t-1} = \begin{pmatrix} \mathcal{N}(0,\ a_1\delta^2_{r1,\,t-1} + a_2\delta^2_{l,\,t-1}) \\ \mathcal{N}(0,\ a_3\delta^2_{l,\,t-1} + a_4(\delta^2_{r1,\,t-1} + \delta^2_{r2,\,t-1})) \\ \mathcal{N}(0,\ a_1\delta^2_{r2,\,t-1} + a_2\delta^2_{l,\,t-1}) \end{pmatrix} \tag{8}$$

where a_1 to a_4 are the parameters that determine the amount of noises.

3.3 Measurement model

A rangefinder model is used for the measurement model. This model uses the measurements consisting of the relative distance r between the robot and a landmark, and the relative angle ϕ. When defining the robot posture as $(x,\ y,\ \theta)^T$, the j-th landmark position as $(\mu_{j,x},\ \mu_{j,y})$, and the measurements as $\boldsymbol{z} = (r, \phi)$, the measurement model in Eq. (3) can be rewritten by (see Fig. 3)

$$\begin{aligned} \boldsymbol{z} &= h(\boldsymbol{x}) + \boldsymbol{\varepsilon} \\ h(\boldsymbol{x}) &= \begin{pmatrix} \sqrt{(\mu_{j,x} - x)^2 + (\mu_{j,y} - y)^2} \\ \mathrm{atan2}(\mu_{j,y} - y,\ \mu_{j,x} - x) - \theta \end{pmatrix} \\ \boldsymbol{\varepsilon} &= \begin{pmatrix} \mathcal{N}(0, \sigma_r^2) \\ \mathcal{N}(0, \sigma_\phi^2) \end{pmatrix} \end{aligned} \tag{9}$$

where σ_r denotes the standard deviation for the relative noise and similarly σ_ϕ denotes one for the relative angle.

3.4 Results and considerations

Figures 4 and 5 show the estimation results of one trial for both methods, where in the figures the red dot denotes the landmark position, the black solid line is the movement trajectory of the robot, the light-blue broken line denotes the odometry value, and the blue $\times$ and solid line are the estimation results. For the estimation of self-position, it is found that both the URTSS-SLAM and UKF-SLAM methods have a high accuracy, compared to a method of using only the odometry. In particular, the estimate produced by using only the odometry displays a large deviation around the terminal point, whereas the both estimation methods give good estimates that reach the goal point. This is attributed to the fact that the both methods estimate the self-position by using the information on landmarks. It is seen that they also give all the landmark positions, i.e., the correct map estimate.

Figures 6 to 8 shows the time variation of the rms error in each axis of the robot posture. The broken line denotes the rms error for the UKF–SLAM, whereas the solid line is the rms error for the URTSS–SLAM. The rms error for the URTSS-SLAM is smaller in most of time durations than that for the UKF-SLAM. It is seen from Fig. 6 that the rms error in x-coordinate is gradually improved in its accuracy for the

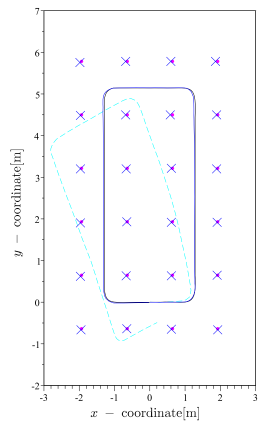

Fig. 4. Estimated robot paths and landmarks (UKF–SLAM)

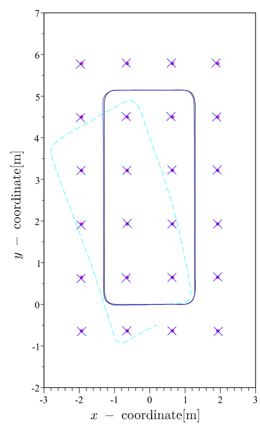

Fig. 5. Estimated robot paths and landmarks (URTSS–SLAM)

latter half of estimation: i.e., its accuracy recovers closely to that of around $t = 10$ [s].

Table 2 shows the time average of the rms error for each axis of the self-position. It is seen that the URTSS-SLAM has about 5.6 [%]smaller rms error in the average of each axis, compared to the UKF-SLAM, so it is claimed that an improved accuracy has been obtained for the self-position estimate in the SLAM problem.

Figures 9 and 10 shows the rms error for the landmark position. It is found from Fig. 9 that the rms error is relatively large because the landmark is estimated with less number of measurements at start, but it becomes gradually small and

Table 2. Simulation results of localization

Algorithm	RMS error		
	x [mm]	y [mm]	θ [rad]
UKF–SLAM	36.233	23.145	0.018659
URTSS–SLAM	35.289	21.159	0.017063

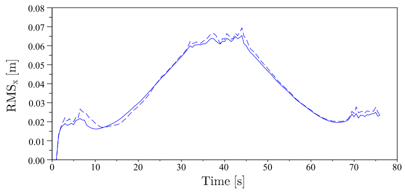

Fig. 6. RMS error for x coordinate

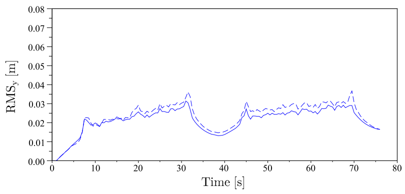

Fig. 7. RMS error for y coordinate

converges as time goes and the number of measurements increases. Two rms errors, related to the landmarks denoted by blue and yellow lines, become large once at about 10 [s] and 12 [s], decrease at about 53 [s], and after that they converge to small values. This is based on the fact that the robot measured those landmarks several times at about 10 [s] and 12 [s], moved out from the measurable range, returned to the original position at about 53 [s], and finally started to re-measure such landmarks. The rms error of the URTSS–SLAM is small and constant approximately, as seen from Fig. 10, because from the property of the URTSS the landmarks are estimated by using all the information for the estimation interval.

Table 3 shows the estimation error for the landmarks, where a landmark error was extracted when the minimum estimation covariance was achieved, and was also averaged by the number of landmarks and trials. The estimation error of the landmarks with the URTSS–SLAM is small about 6.8 [%], compared to that with the UKF-SLAM, so that it is claimed that the map accuracy has been improved, as well as the estimation accuracy of the self-position for the robot.

4 CNCLUSIONS

In this paper, the URTSS has been applied to the SLAM problem and a URTSS–SLAM has been proposed. The URTSS–SLAM is the technique of having aimed at the improvement in accuracy by reprocessing the estimation result

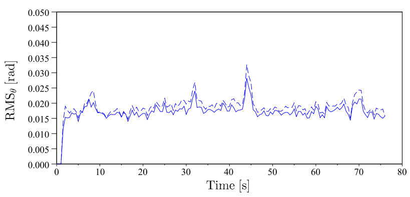

Fig. 8. RMS error for θ coordinate

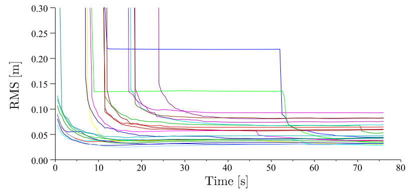

Fig. 9. RMS error for landmark location (UKF–SLAM)

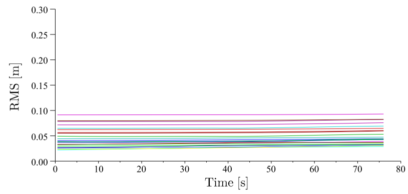

Fig. 10. RMS error for landmark location (URTSS–SLAM)

Table 3. Simulation results of landmark locations

Algorithm	error of landmark locations [mm]
UKF–SLAM	41.049
URTSS–SLAM	38.254

due to the UKF–SLAM by the URTSS. Then, the estimation accuracy of the proposed URTSS–SLAM and the conventional UKF–SLAM was compared through the simulation experiment. As a result, compared to the conventional technique, the estimation accuracy for both the self-position and the map was improved, so it can be said that the URTSS–SLAM is effective in an off-line SLAM problem.

REFERENCES

[1] Durrant-Whyte H, Bailey T (2006), Simultaneous localization and mapping Part I. IEEE Robotics & Automation Magazine 13(2): 99–110

[2] Durrant-Whyte H, Bailey T (2006), Simultaneous localization and mapping Part II. IEEE Robotics & Automation Magazine 13(3): 108–117

[3] Julier SJ, Uhlmann JK, Durrant-Whyte HF (1995), A new approach for filtering nonlinear systems. Proc. of American Control Conference, pp. 1628–1632

[4] Rauch HE, Tung F, Striebel CT (1965), Maximum likelihood estimates of linear dynamic systems. AIAA Journal 3(8): 1445–1450

[5] Terejanu G, Singh T, Scott PD (2007), Unscented Kalman filter/smoother for a CBRN Puff-based dispersion model, Proc. of the 10th Int. Conf. on Information Fusion

[6] Särkkä S (2008), Unscented Rauch–Tung–Striebel smoother. IEEE Trans. on Automatic Control 53(3), 845–849

[7] Kim C, Sakthivel R, Chung W (2008), Unscented fastSLAM: A robust and efficient solution to the SLAM problem. IEEE Trans. on Robotics 24(4), 808–820

[8] Thrun S, Burgard W, Fox D (2005), Probabilistic Robotics. The MIT Press