Optimal Proposal Distribution FastSLAM with Suitable Gaussian Weighted Integral Solutions

Qingling Li[1,*], Yu Song[2], ZengGuang Hou[3], and Bin Zhu[1]

[1] Department of Mechanical Engineering,
China University of Mining & Technology, Beijing, China
doudouhit@163.com, xcumtzhubin@126.com
[2] School of Electronic & Information Engineering, Beijing Jiaotong University, Beijing, China
songyu@bjtu.edu.cn
[3] Institute of Automation, Chinese Academy of Science, Beijing, China
hou@compsys.ia.ac.cn

Abstract. One of the key issues in Gaussian SLAM is to calculate nonlinear transition density of Gaussian prior, i.e. to calculate Gaussian Weight Integral (GWI) whose integrand is with the form *nonlinear function*×*Gaussian prior density*. Up to now, some GWI solutions have been applied in SLAM (e.g. linearization, unscented transform and cubature rule), and different SLAM algorithms were derived based on theirs GWI solutions. While, how to select suitable GWI solution for SLAM is still lack of theoretical analysis. In this paper, we proposed an optimal proposal FastSLAM algorithm with suitable GWI solutions. The main contributions of this work lies that: (1) an unified FastSLAM framework with optimal proposal distribution is summarized; (2) a SLAM dimensionality based GWI solution selection criterion is designed; (3) we propose a new SLAM algorithm. The performance of the proposed SLAM is investigated and compared with the FastSLAM2.0 and UFastSLAM using simulations and our opinion is confirmed by the results.

Keywords: Mobile Robot, SLAM, Unscented Transform, Cubature Rule.

1 Introduction

The Rao-Blackwellized particle filter (RBPF) [1] based SLAM algorithm was introduced by Montemerlo in 2003. Now it has became popular in SLAM due to its low computational cost and robustness for incorrect data association. The algorithm has two editions: FastSLAM1.0 [2] and FastSLAM2.0 [3]. The former utilizes common particle filter to track robot path, and each particle independently keeps a feature landmark map based on a set of EKFs. To overcome particle set degeneracy in FastSLAM1.0, the FastSLAM2.0 utilizes EKF to design better proposal distribution. Based on the similar idea, Gristti proposed an adaptive grid map FastSLAM [4]. The technique combines laser scan-matching with mobile robot odometry to optimize the proposal distribution

* Corresponding author.

M. Lee et al. (Eds.): ICONIP 2013, Part I, LNCS 8226, pp. 301–309, 2013.
© Springer-Verlag Berlin Heidelberg 2013

of particle filter. Sim proposed a stereo vision FastSLAM, and the particle proposal distribution was designed by using visual odometery prior [5]. Kim proposed Unscented FastSLAM [6], which utilizes scaled unscented transformation [7][8] to estimate the nonlinear transition density of Gaussian prior. In [9], a non-static environment FastSLAM was proposed by sampling new generation particles from multiple ancestor priors. Moreno designed a new approach to fuse the grid-map and feature-map for FastSLAM [10]. In our previous work, a Cubature FastSLAM [11] was derived by utilizing cubature rule [12] as the Gaussian weighted integral solutions.

One of the key issues in Gaussian SLAM is to calculate nonlinear transition density of Gaussian prior, i.e., to calculate Gaussian Weight Integral (GWI). Up to now, some GWI solutions have been applied in SLAM filter. While, how to choose a suitable GWI solution for SLAM is still lack of theoretical analysis. In this paper, the GWI solution selection problem is discussed and a new SLAM with better GWI solution is derived. The rest paper is organized as follows: Section 2 gives a brief review of FastSLAM and a unified optimal proposal FastSLAM is summarized. A GWI solution selection criterion is discussed in Section 3. In Section 4 we proposed a SLAM algorithm with suitable GWI solutions. Section 5 presents simulations, followed by the conclusions.

2 Unified FastSLAM with Optimal Proposal

2.1 Brief Review of FastSLAM

The key idea of FastSLAM is to estimate the joint posterior $p(\boldsymbol{s}^k, \boldsymbol{\Theta} | \boldsymbol{z}^k, \boldsymbol{u}^{k-1})$ about the map $\boldsymbol{\Theta} = \boldsymbol{\theta}_1 \cdots \boldsymbol{\theta}_M$ and the mobile robot path $\boldsymbol{s}^k = \boldsymbol{s}_1 \cdots \boldsymbol{s}_k$ based on a set of observations $\boldsymbol{z}^k = \boldsymbol{z}_1 \cdots \boldsymbol{z}_k$ and control inputs $\boldsymbol{u}^{k-1} = \boldsymbol{u}_0 \cdots \boldsymbol{u}_{k-1}$. Based on conditional independence property of the full SLAM problem, the joint posterior $p(\boldsymbol{s}^k, \boldsymbol{\Theta} | \boldsymbol{z}^k, \boldsymbol{u}^{k-1})$ is factored as

$$p(\boldsymbol{s}^k, \boldsymbol{\Theta} | \boldsymbol{z}^k, \boldsymbol{u}^{k-1}) = p(\boldsymbol{s}^k | \boldsymbol{z}^k, \boldsymbol{u}^{k-1}) \cdot \prod_{m=1}^{M} p(\boldsymbol{\theta}_m | \boldsymbol{s}^k, \boldsymbol{z}^k, \boldsymbol{u}^{k-1}) \tag{1}$$

In FastSLAM, the path posterior $p(\boldsymbol{s}^k | \boldsymbol{z}^k, \boldsymbol{u}^{k-1})$ is estimated with a particle filter and the map $p(\boldsymbol{\Theta} | \boldsymbol{s}^k, \boldsymbol{z}^k, \boldsymbol{u}^{k-1})$ is analytically estimated utilizing M separate Kalman filters. Each particle for the FastSLAM is assembled by the robot path and the landmarks:

$$\boldsymbol{S}_k^{[i]} = < \boldsymbol{s}^{k,[i]}, \left(\boldsymbol{\mu}_k^{[i][1]}, \boldsymbol{\Sigma}_k^{[i][1]}\right) \cdots \left(\boldsymbol{\mu}_k^{[i][M]}, \boldsymbol{\Sigma}_k^{[i][M]}\right) > \tag{2}$$

where, $[i]$ and $[m]$ are indicate index of the particle and the landmark; $\boldsymbol{s}^{k[i]}$ is robot path hypothesis; $(\boldsymbol{\mu}_k^{[i][m]}, \boldsymbol{\Sigma}_k^{[i][m]})$ is the Gaussian representation of the mth feature landmark.

2.2 Unified FastSLAM Framework with Optimal Proposal

The noisy robot motion model and environment observation motion model are

$$\begin{cases} \boldsymbol{s}_k = f\left(\boldsymbol{s}_{k-1}, \boldsymbol{u}_{k-1} + \boldsymbol{\delta u}_{k-1}\right) \\ \boldsymbol{z}_k = h\left(\boldsymbol{s}_k, \boldsymbol{\mu}_{k-1}^{[m]}\right) + \boldsymbol{\delta z}_k \end{cases} \tag{3}$$

where, $\boldsymbol{s}_k \in \mathbb{R}^{n_s}$ is the robot pose; $z_k \in \mathbb{R}^{n_z}$ is the observation; f and h are the nonlinear robot motion and observation model, respectively; $\boldsymbol{u}_{k-1} \in \mathbb{R}^{n_u}$ is the control input in the

time interval $[k-1,k)$, $\boldsymbol{\delta u}_{k-1} \in \mathbb{R}^{n_u}$ is control noise with covariance $\boldsymbol{Q}$; $\boldsymbol{\delta z}_{k-1} \in \mathbb{R}^{n_z}$ is observation noise with covariance $\boldsymbol{R}$; $\boldsymbol{\mu}_{k-1}^{[m]} \in \mathbb{R}^{n_\mu}$ is the $[m]$-th landmark state.

To predict new robot state $\boldsymbol{s}_k^{[i]}$, the previous robot state $\boldsymbol{s}_{k-1}^{[i]}$ requires to be augmented with the control input, given by

$$\bar{\boldsymbol{a}} = \begin{pmatrix} \boldsymbol{s}_{k-1}^{[i]} \\ \boldsymbol{u}_{k-1} \end{pmatrix}, \boldsymbol{P}_a = \begin{pmatrix} \boldsymbol{P}_{k-1}^{[i]} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q} \end{pmatrix} \tag{4}$$

Where, $\bar{\boldsymbol{a}} \in \mathbb{R}^{n_s+n_u}$ and $\boldsymbol{P}_a \in \mathbb{R}^{(n_s+n_u)\times(n_s+n_u)}$ are the augmented robot state and its covariance, respectively. $\boldsymbol{s}_{k-1}^{[i]} \in \mathbb{R}^{n_s}$ and $\boldsymbol{P}_{k-1}^{[i]} \in \mathbb{R}^{n_s \times n_s}$ are the previous mean and covariance of the robot state. The augmented state $\boldsymbol{a}$ satisfies Gaussian $\boldsymbol{a} \sim \mathcal{N}(\bar{\boldsymbol{a}}, \boldsymbol{P}_a)$.

Consequently, the predicted robot state and its covariance are calculated by

$$\begin{aligned} \boldsymbol{s}_{k,k-1}^{[i]} &= \int f^*(\boldsymbol{a})\, \mathcal{N}(\bar{\boldsymbol{a}}, \boldsymbol{P}_a)\, d\boldsymbol{a} \\ \boldsymbol{P}_{k,k-1}^{[i]} &= \int \left[f^*(\boldsymbol{a}) - \boldsymbol{s}_{k,k-1}^{[i]} \right] \left[f^*(\boldsymbol{a}) - \boldsymbol{s}_{k,k-1}^{[i]} \right]^{\mathrm{T}} \mathcal{N}(\bar{\boldsymbol{a}}, \boldsymbol{P}_a)\, d\boldsymbol{a} \end{aligned} \tag{5}$$

When a landmark with index $[m]$ is revisited by the robot, the robot state and its covariance can be updated. To do this, $\boldsymbol{s}_{k|k-1}^{[i]}$ and $\boldsymbol{P}_{k|k-1}^{[i]}$ are augmented by integrating the robot state and the revisited landmark state into one Gaussian, that is

$$\bar{\boldsymbol{b}} = \begin{pmatrix} \boldsymbol{s}_{k,k-1}^{[i]} \\ \boldsymbol{\mu}_{k-1}^{[i][m]} \end{pmatrix}, \boldsymbol{P}_b = \begin{pmatrix} \boldsymbol{P}_{k,k-1}^{[i]} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma}_{k-1}^{[i][m]} \end{pmatrix} \tag{6}$$

where, $\bar{\boldsymbol{b}} \in \mathbb{R}^{n_s+n_\mu}$ and $\boldsymbol{P}_b \in \mathbb{R}^{(n_s+n_\mu)\times(n_s+n_\mu)}$ are the augmented prediction state and its covariance, respectively. The density of $\boldsymbol{b}$ is Gaussian distribution $\mathcal{N}(\bar{\boldsymbol{b}}, \boldsymbol{P}_b)$.

Based on $\mathcal{N}(\bar{\boldsymbol{b}}, \boldsymbol{P}_b)$, the predicted measurement $\boldsymbol{z}_{k,k-1}^{[i][m]}$, the measurement innovation covariance $\boldsymbol{P}_{zz}^{[i][m]}$ and the cross covariance $\boldsymbol{P}_{sz}^{[i][m]}$ are calculated as

$$\begin{aligned} \boldsymbol{z}_{k,k-1}^{[i][m]} &= \int h^*(\boldsymbol{b})\, \mathcal{N}(\bar{\boldsymbol{b}}, \boldsymbol{P}_b)\, d\boldsymbol{b} \\ \boldsymbol{P}_{zz}^{[i][m]} &= \int \left[h^*(\boldsymbol{b}) - \boldsymbol{z}_{k,k-1}^{[i][m]} \right] \left[h^*(\boldsymbol{b}) - \boldsymbol{z}_{k,k-1}^{[i][m]} \right]^{\mathrm{T}} \mathcal{N}(\bar{\boldsymbol{b}}, \boldsymbol{P}_b)\, d\boldsymbol{b} + \boldsymbol{R} \\ \boldsymbol{P}_{sz}^{[i][m]} &= \int \left[[\boldsymbol{I}_{n_s \times n_s} \quad \boldsymbol{0}_{n_s \times n_\mu}] \boldsymbol{b} - \boldsymbol{s}_{k,k-1}^{[i]} \right] \left[h^*(\boldsymbol{b}) - \boldsymbol{z}_{k,k-1}^{[i][m]} \right]^{\mathrm{T}} \mathcal{N}(\bar{\boldsymbol{b}}, \boldsymbol{P}_b)\, d\boldsymbol{b} \end{aligned} \tag{7}$$

The state update is performed based on standard Kalman filtering algorithm

$$\boldsymbol{K}_s = \boldsymbol{P}_{sz}^{[i][m]} \left(\boldsymbol{P}_{zz}^{[i][m]} \right)^{-1} \qquad \begin{cases} \bar{\boldsymbol{s}}_k^{[i]} = \boldsymbol{s}_{k,k-1}^{[i]} + \boldsymbol{K}_s \left(\boldsymbol{z}_k^{[m]} - \boldsymbol{z}_{k,k-1}^{[i][m]} \right) \\ \boldsymbol{P}_k^{[i]} = \boldsymbol{P}_{k,k-1}^{[i]} - \boldsymbol{K}_s \boldsymbol{P}_{zz}^{[i][m]} \left(\boldsymbol{K}_s \right)^{\mathrm{T}} \end{cases} \tag{8}$$

where, $\boldsymbol{z}_k^{[m]}$ is the true sensor measurement for $[m]$-th revisited landmark.

For the multiple observations case, Eq.6- Eq.8 are repeated for each landmark, and $\bar{\boldsymbol{s}}_k^{[i]}$ and $\boldsymbol{P}_k^{[i]}$ are updated based on their previously updated one. When the update

routine is accomplished, the new robot state $\boldsymbol{s}_k^{[i]}$ is sampled from the Gaussian $\mathcal{N}\left(\overline{\boldsymbol{s}}_k^{[i]}, \boldsymbol{P}_k^{[i]}\right)$, and its importance factor $\omega_k^{[i]}$ is calculated as

$$\omega_k^{[i]} = \prod_m \left\{ \frac{1}{\sqrt{\left|2\pi \boldsymbol{P}_{zz}^{[i][m]}\right|}} \exp\left[-\frac{\left(\boldsymbol{z}_{k,k\text{-}1}^{[i][m]} - \boldsymbol{z}_k^{[m]}\right)^{\mathrm{T}} \left(\boldsymbol{z}_{k,k\text{-}1}^{[i][m]} - \boldsymbol{z}_k^{[m]}\right)}{2\boldsymbol{P}_{zz}^{[i][m]}} \right] \right\} \tag{9}$$

where, $\boldsymbol{z}_{k|k\text{-}1}^{[i][m]} - \boldsymbol{z}_k^{[m]}$ and $\boldsymbol{P}_{zz}^{[i][m]}$ are the measurement innovation and the innovation covariance matrix for the [m]-th landmark.

For each revisited landmark with index [m], the landmark state equation and observation equation are modeled as

$$\begin{cases} \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}\left(\boldsymbol{\mu}_{k-1}^{[i][m]}, \boldsymbol{\Sigma}_{k-1}^{[i][m]}\right) \\ \boldsymbol{z}_k^{[i][m]} = h(\boldsymbol{s}_k^{[i]}, \boldsymbol{\mu}_{k\text{-}1}^{[i][m]}) + \delta z_k \end{cases} \tag{10}$$

where, $\boldsymbol{s}_k^{[i]}$ is the "known" robot state; $\boldsymbol{\mu}_{k-1}^{[i][m]}$ and $\boldsymbol{\Sigma}_{k-1}^{[i][m]}$ are the mean and the covariance of the [m]-th revisited landmark, respectively.

The predicted measurement and its covariance with the known robot state are

$$\begin{aligned} \boldsymbol{z}_{k,k\text{-}1}^{*[i][m]} &= \int h\left(\boldsymbol{\mu}, \boldsymbol{s}_k^{[i]}\right) \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) d\boldsymbol{\mu} \\ \boldsymbol{P}_{zz}^{[m]} &= \int \left[h\left(\boldsymbol{\mu}, \boldsymbol{s}_k^{[i]}\right) - \boldsymbol{z}_{k,k\text{-}1}^{*[i][m]} \right] \left[h\left(\boldsymbol{\mu}, \boldsymbol{s}_k^{[i]}\right) - \boldsymbol{z}_{k,k\text{-}1}^{*[i][m]} \right]^{\mathrm{T}} \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) d\boldsymbol{\mu} + \boldsymbol{R} \\ \boldsymbol{P}_{\boldsymbol{\mu}\boldsymbol{z}}^{[m]} &= \int \left[\boldsymbol{\mu} - \boldsymbol{\mu}_{k\text{-}1}^{[i][m]} \right] \left[h\left(\boldsymbol{\mu}, \boldsymbol{s}_k^{[i]}\right) - \boldsymbol{z}_{k,k\text{-}1}^{*[i][m]} \right]^{\mathrm{T}} \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) d\boldsymbol{\mu} \end{aligned} \tag{11}$$

The state update of the [m]-th revisited landmark is accomplished by a Kalman filter

$$\boldsymbol{K}_{\mu} = \boldsymbol{P}_{\mu z}^{[m]} \left(\boldsymbol{P}_{zz}^{[m]} \right)^{-1} \qquad \begin{cases} \boldsymbol{\mu}_k^{[i][m]} = \boldsymbol{\mu}_{k\text{-}1}^{[i][m]} + \boldsymbol{K}_{\mu} \left(\boldsymbol{z}_k^{[m]} - \boldsymbol{z}_{k,k\text{-}1}^{*[i][m]} \right) \\ \boldsymbol{\Sigma}_k^{[i][m]} = \boldsymbol{\Sigma}_{k\text{-}1}^{[i][m]} - \boldsymbol{K}_{\mu} \boldsymbol{P}_{zz}^{[m]} (\boldsymbol{K}_{\mu})^{\mathrm{T}} \end{cases} \tag{12}$$

For the [n]-th new visited landmark, the measurement $\boldsymbol{z}_k^{[n]}$ and its covariance $\boldsymbol{R}$ are used to initialize the landmark's Gaussian representation, given by

$$\begin{aligned} \boldsymbol{\mu}_k^{[i][n]} &= \int h^{-1}\left(z, \boldsymbol{s}_k^{[i]}\right) \mathcal{N}\left(\boldsymbol{z}_k^{[n]}, \boldsymbol{R}\right) dz \\ \boldsymbol{\Sigma}_k^{[i][n]} &= \int \left[h^{-1}\left(z, \boldsymbol{s}_k^{[i]}\right) - \boldsymbol{\mu}_k^{[i][n]} \right] \left[h^{-1}\left(z, \boldsymbol{s}_k^{[i]}\right) - \boldsymbol{\mu}_k^{[i][n]} \right]^{\mathrm{T}} \mathcal{N}\left(\boldsymbol{z}_k^{[n]}, \boldsymbol{R}\right) dz \end{aligned} \tag{13}$$

3 Solutions for Gaussian Weighted Integral

Without loss generality, an unify form of the nonlinear functions in SLAM can be defined as $\boldsymbol{y} = g(\boldsymbol{x})$ with $\boldsymbol{x} \in \mathbb{R}^{n_x}$ satisfies Gaussian $\mathcal{N}(\overline{\boldsymbol{x}}, \boldsymbol{P}_x)$, then the following two GWIs are used to calculate the nonlinear transition density $\mathcal{N}\left(\overline{\boldsymbol{y}}, \boldsymbol{P}_y\right)$

$$\begin{aligned} \overline{\boldsymbol{y}} &= \int g(\boldsymbol{x}) \mathcal{N}(\overline{\boldsymbol{x}}, \boldsymbol{P}_x) d\boldsymbol{x} = \int g\left(\sqrt{\boldsymbol{P}_x}\boldsymbol{t} + \overline{\boldsymbol{x}}\right) \mathcal{N}(\boldsymbol{0}, \mathbf{I}) d\boldsymbol{t} \\ \boldsymbol{P}_y &= \int g\left(\sqrt{\boldsymbol{P}_x}\boldsymbol{t} + \overline{\boldsymbol{x}}\right) g^{\mathrm{T}}\left(\sqrt{\boldsymbol{P}_x}\boldsymbol{t} + \overline{\boldsymbol{x}}\right) \mathcal{N}(\boldsymbol{0}, \mathbf{I}) d\boldsymbol{t} - \overline{\boldsymbol{y}}(\overline{\boldsymbol{y}})^{\mathrm{T}} \end{aligned} \tag{14}$$

3.1 GWI Solutions Utilized in SLAM

1) **Linearization.** The linearization solution are utilized in FastSLAM1.0 and FastSLAM2.0, in which the GWI is calculated based on the first-order linearization of the nonlinear function g, denote as $g(\boldsymbol{x}) \approx g(\bar{\boldsymbol{x}}) + g'(\bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}})$. And the transition density $\mathcal{N}(\bar{\boldsymbol{y}}, \boldsymbol{P}_y)$ is easily obtained by $\bar{\boldsymbol{y}} \approx g(\bar{\boldsymbol{x}})$ and $\boldsymbol{P}_y \approx g'(\boldsymbol{x})\boldsymbol{P}_x g'(\boldsymbol{x})^{\mathrm{T}}$.
2) **Unscented Transform.** In UT, $2n_x + 1$ sigma-points $\{\mathcal{X}_i\} = \{\bar{\boldsymbol{x}} \pm (\sqrt{(n_x + \kappa)\boldsymbol{P}_x})_i\}$ are utilized to capture some low-order moments of the prior Gaussian $\mathcal{N}(\bar{\boldsymbol{x}}, \boldsymbol{P}_x)$. And the transformed sigma-point set $\{\mathcal{Y}_i\}$ is computed by passing the $\{\mathcal{X}_i\}$ through the nonlinear function g, as $\{\mathcal{Y}_i = g(\mathcal{X}_i)\}$. So, $\mathcal{N}(\bar{\boldsymbol{y}}, \boldsymbol{P}_y)$ are obtained by weighted linear regression of $\{\mathcal{Y}_i\}$, given by $\bar{\boldsymbol{y}} \approx \sum \omega_c \mathcal{Y}_i$ and $\boldsymbol{P}_y \approx \sum \omega_g [\mathcal{Y}_i - \bar{\boldsymbol{y}}][\mathcal{Y}_i - \bar{\boldsymbol{y}}]^{\mathrm{T}}$.
3) **Cubature Rule.** The GWI are calculated by using $2n_x$ cubature-points, given by $\bar{\boldsymbol{y}} \approx \sum_{j=1}^{2n_x} g(\sqrt{\boldsymbol{P}_x}\boldsymbol{\xi}_j + \bar{\boldsymbol{x}}) / 2n_x$ and $\boldsymbol{P}_y \approx \sum_{j=1}^{2n_x} g(\sqrt{\boldsymbol{P}_x}\boldsymbol{\xi}_j + \bar{\boldsymbol{x}}) g^{\mathrm{T}}(\sqrt{\boldsymbol{P}_x}\boldsymbol{\xi}_j + \bar{\boldsymbol{x}}) / 2n_x - \bar{\boldsymbol{y}}\bar{\boldsymbol{y}}^{\mathrm{T}}$ .where, the cubature-point set is calculated as $\{\boldsymbol{\xi}_j\} = \sqrt{n_x}\{[1]_j\}$, and $\{[1]_j\} \in \mathbb{R}^{n_x}$ are the $2n_x$ intersection points of the n_x dimension coordinate axes with the unit hypersphere.

Table 1 and Table 2 show the moment characteristics and numerical characteristics for the three GWI solutions, respectively.

Table 1. Moment Characteristics for Different GWI Solutions

GWI Solutions	One-dimensional Gaussian case (n=1)	Multiple-dimensional Gaussian case (n>1)
Linearization	up to the 1-th order	up to the 1-th order
Unscented Transform	up to the 5-th order	up to the 3-th order
Cubature Rule	up to the 3-th order	up to the 3-th order

Table 2. Numerical Calculation Characteristics

	Numerical stbility factor		Is the $\boldsymbol{P}_y$ non-negative definite?	
	$n \le 3$	$n > 3$	$n \le 3$	$n > 3$
Unscented Transform	1	$2n/3 - 1$	yes	unsure
Cubature Rule	1	1	yes	yes

3.2 Criterion for Select Suitable GWI Solution

The UT is a dimensionality sensitive GWI solution. For the low dimensional cases ($n \le 3$), the moments of sigma-point set can hit some the 4-th order moments, and the transformed covariance is nonnegative definiteness, also the stability factor is as good as cubature rule. While, for high dimensional Gaussian cases (n>3), the transformed covariance may be negative definiteness, and the stability factor $\mathrm{S}_{\mathrm{UT}} = (2n/3) - 1$ is increased with n. Compared with UT, the cubature rule is a dimensionality insensitive

GWI solution both from moment characteristic and numerical characteristic. For any Gaussian dimensionality n, the cubature rule based GWI solution is correctly calculated up to the 3-order nonlinearity, and the non-negative definiteness of the transformed covariance can be guaranteed, also the stability factor always equal to 1. Based on above analysis, we design a dimensionality based GWI solution selection criterion: (i) for the case of dimensionality $n<3$, the UT is better than cubature rule because of partial of the fourth order moments of piror Gaussian are preserved by sigma-point set; (ii) for $n=3$, the cubature rule is equivalent to the UT; (iii) for $n>3$, cubature rule is a good choice because the transformed covariance by UT may be negative definiteness and as well the numerical stability factor is increased linearly with n for UT.

3.3 Proposed FastSLAM with Suitable GWI Solutions

From Section 2, four main parts are included in SLAM: (1) particle state prediction, (2) particle state updating, (3) revisited landmarks updating and (4) new landmarks initialization. Because each part of the algorithm has its own Gaussian dimensionality n, hence the suitable GWI solution for each part can be selected for SLAM implementation.

1) For the particle state prediction, the dimensionality of augmented robot state $\boldsymbol{a}$ is beyond three (i.e. $n_s + n_u > 3$, see Eq.4). Therefore, the cubature rule is a better GWI solution. For example, in 3D.O.F SLAM, the vehicle state is $\boldsymbol{s} = (x, y, \theta)^{\mathrm{T}}$ and the control input is comprised by velocity and steering angle. So, the dimensionality of the augmented state $\boldsymbol{a}$ is 5. For the case, the predict covariance $\boldsymbol{P}_{k,k\text{-}1}^{[i]}$ (in Eq.5) may be negative definite with UT because the weight of center sigma-point is negative $\omega_0 = -2/3 < 0$. Also, the numerical stability factor of the UT is bad due to $\mathrm{S_{UT}} = 2.33 > 1$;

2) For the particle state update part, the dimensionality of augmented robot state $\boldsymbol{b}$ is also larger than 3 (i.e. $n_s + n_\mu > 3$, see Eq.6). So, the cubature rule is selected as GWI solution with the same reasons for the first part.

3) For the revisited landmarks updating step and the new landmarks initialization step, the dimensionalities of Gaussians are determined by the landmark state $\boldsymbol{\mu}$ and the measurement $\boldsymbol{z}$, respectively. For the two steps, the UT is selected as GWI solution. There are two reasons for the utilization of UT: on one hand, the SLAM covariance matrixes are all nonnegative definiteness with UT, and the numerical stability factor is as good as cubature rule. On the other hand, the errors for the fourth order moments of the sigma-point set are smaller than that of cubature point set.

4 Simulation Results

The performance of the proposed SLAM is compared with FastSLAM2.0 and UFastSLAM. The simulation scenario is a $100\mathrm{m} \times 100\mathrm{m}$ size rectangular shaped map with 41 landmarks. For each simulation run, the robot starts with its initial state $s_0 = [0,0,0]^{\mathrm{T}}$, and then travels along six global planning points until closes the loop. Fig.1 is the result of the proposed SLAM (The measurement noise is set to be 0.1 m in range, 6° in bearing, and the control noise is set to be 0.5 m/s in velocity, 2^{o} in steering

angle). As can be seen, the robot path tracking is converged and landmarks are accurately positioned.

Two noise levels are used in simulations. With each noise level, ten independence SLAM simulation runs were carried out for each SLAM. For the first noise level, the measurement noise is set to be 0.1 m in range, 1° in bearing, and the robot control noise is set to be 0.3 m/s in velocity, 1^{o} in steering angle. As it can be seen from Fig.2, the vehicle path error of the proposed SLAM is lower than that of other two algorithms, and the landmarks are more accurately mapped by proposed SLAM. For the second noise level, the noise parameters are increased: the measurement noise is set to be 0.1 m in range, 6° in bearing, and the control noise is set to be 0.5 m/s in velocity, 2^{o} in steering angle. We can see from Fig.3, the SLAM errors for the three algorithms are increased with the noise level. And among the three algorithms, the proposed SLAM has the lowest SLAM errors both for vehicle localization and for environmental mapping.

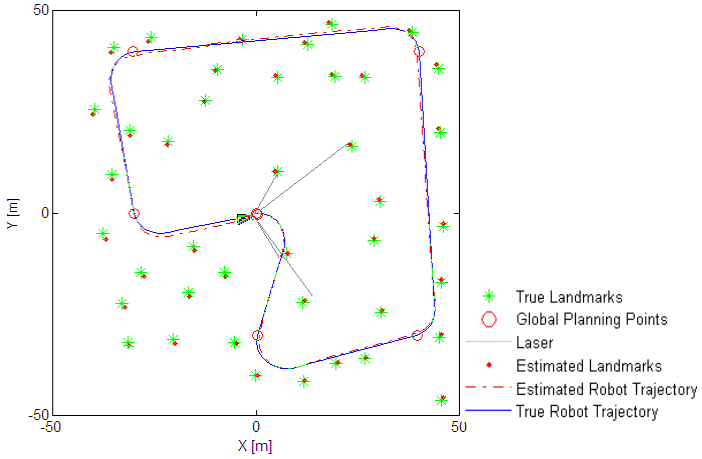
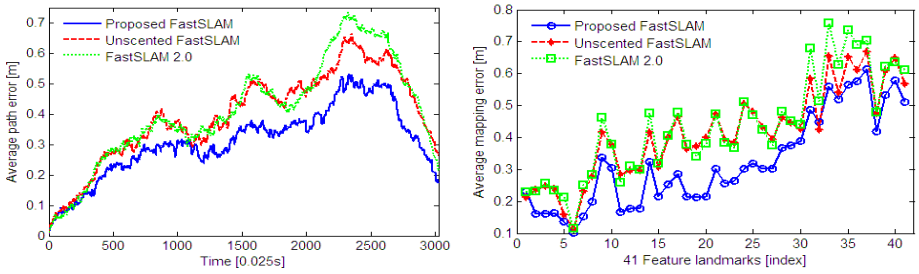
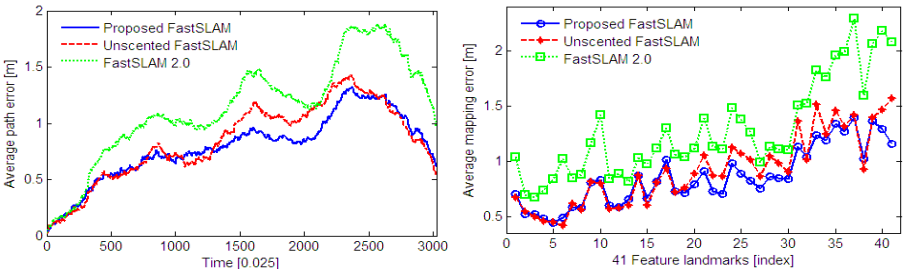


Fig. 1. Performance of the proposed SLAM



(a) Average error norm for vehicle path (b) Average error norm for 41 feature landmarks

Fig. 2. Performance evaluation for SLAM algorithms with the measurement noise is 0.1 m in range, 1° in bearing, and the control noise is 0.3 m/s in velocity, 1° in steering angle



(a) Average error norm for vehicle path (b) Average error norm for 41 feature landmarks

Fig. 3. Performance evaluation for SLAM algorithms with the measurement noise is 0.1 m in range, 6° in bearing, and the control noise is 0.5 m/s in velocity, 2° in steering angle

5 Conclusion

In this paper, we proposed a new FastSLAM by choosing the suitable GWI solutions for different parts of the SLAM implementations. In proposed SLAM, the cubature rule is utilized to construct the particle filter proposal distribution and the unscented transform is used to initialize the Gaussian representations of the environment feature landmarks and to estimate the posteriors of revisited feature landmarks. The effectiveness of the proposed SLAM is verified by SLAM simulations. Results show that the proposed SLAM outperforms FastSLAM2.0 and UFastSLAM.

Acknowledgments. This research is supported by the National Nature Science Foundation of China (Grants #61005070 and #61175076).

References

1. Doucet, A., Freitas, N.D.: Rao-Blackwellised particle filtering for dynamic Bayesian networks. In: Sixteenth Conference on Uncertainty in Artificial Intelligence, pp. 176–183 (2000)
2. Montemerlo, M., Thrun, S.: Simultaneous localization and mapping with unknown data association using FastSLAM. In: IEEE International Conference on Robotics and Automation, ICRA, pp. 1985–1991 (2003)
3. Montemerlo, M., Thrun, S., Koller, D., et al.: FastSLAM2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In: Eighteenth International Joint Conference on Artificial Intelligence (IJCAI), pp. 1151–1156 (2003)
4. Grisetti, G., Stachniss, C., Burgard, W.: Improved techniques for grid mapping with Rao-Blackwellized particle filters. IEEE Transactions on Robotics (TRO) 23(1), 34–46 (2007)
5. Sim, R., Elinas, P., Little, J.: A study of the Rao–Blackwellised particle filter for efficient and accurate vision-based SLAM. International Journal on Computer Vision (IJCV) 74(3), 303–318 (2007)

6. Kim, C., Sakthivel, R., Chung, W.K.: Unscented FastSLAM: a robust and efficient solution to SLAM problem. IEEE Transactions on Robotics (TRO) 24(4), 808–820 (2008)
7. Julier, S.J., Uhlmann, J.K.: Unscented filtering and nonlinear estimation. Proceedings of the IEEE 92(3), 401–422 (2004)
8. Julier, S.J.: A new method for the nonlinear transformation of means and covariances in filters and estimators. IEEE Transactions on Automatic Control 45(3), 477–482 (2000)
9. Lee, J.S., Kim, C.: Robust RBPF-SLAM using sonar sensors in non-static environments. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 250–256 (2010)
10. Moreno, L., Garrido, S., Blanco, D.: Bridging the gap between feature-and grid-based SLAM. Robotics and Autonomous Systems (RAS) 58, 140–148 (2010)
11. Song, Y., Li, Q.L., Kang, Y.F., Song, Y.D.: CFastSLAM: a new Jacobian free solution to SLAM problem. In: IEEE International Conference on Robotics and Automation (2012)
12. Arasaratnam, I., Haykin, S.: Cubature kalman filters. IEEE Transaction on Automatic Control (TAC) 54(6), 1254–1269 (2009)