Planar-Feature Based 3D SLAM Using Randomized Sigma Point Kalman Filters

Cihan Ulas* and Hakan Temeltas

Istanbul Technical University, Department of Control Engineering, Istanbul, 34469, Turkey
{culas,hakan.temeltas}@itu.edu.tr

Abstract. In this study, a novel filtering method called Randomized Sigma Point Kalman Filter (RSPKF) is introduced for feature based 3D Simultaneous Localization and Mapping (SLAM). Conventional SLAM methods are mostly based on Extended Kalman Filters (EKF) for 'mild' nonlinear processes and Unscented KF (UKF) or Cubature KF (CKF) for 'aggressive' nonlinear processes. A critical problem of the existing filtering methods is that they lead to biased estimates of the state and measurement statistics. The main purpose of this study is to propose a new local filter, RSPKF, based on stochastic integration rules providing an unbiased estimate of an integral for feature based SLAM. The simulation based on point features in 2D and experimental results based on planar features in 3D show that the RSPKF based SLAM method provides more accurate results than the traditional methods.

Keywords: SLAM, feature extraction, randomized sigma point filters.

1 Introduction

Simultaneous Localization and Mapping plays a central role for fully autonomous system when the Global Navigation Satellite System is not available or denied. SLAM is an active research area of the last decade and its solution is considered as the "holy grail" by the robotics researchers [1]. Feature based SLAM (Fb-SLAM) methods requires sophisticated feature extraction methods. These features are principally considered as rotation and translation independent and can be distinguished when they are exists in the two consecutive observations. The aim of the feature based SLAM methods is to estimate the robot pose and landmark locations combined in a state vector.

A traditional representation in SLAM is to use state space model with additive Gaussian noise, which leads to the local filters such as EKF, UKF, and CKF. EKF is the well-known filtering method using the first order approximation of the nonlinear functions [2]. Therefore, it is appropriate for 'mild' nonlinear processes and measurement models. In order to overcome the linearization problem of 'aggressive' nonlinearities, Julier and Uhlman [3] proposed Unscented Kalman Filter (UKF) known as

* Corresponding author.

D. Choukroun et al. (eds.), *Advances in Estimation, Navigation, and Spacecraft Control*,
DOI: 10.1007/978-3-662-44785-7_7

derivative-free approach. The UKF, instead of linearization of the nonlinear functions, estimate the mean values and covariance matrices with sigma points, which are obtained by a deterministic sampling approach. Cubature Kalman Filters (CKF) is proposed as a more accurate filtering method and a more mathematically principled method than UKF for nonlinear state estimation by Arasaratnam and Haykin [4]. CKF is also more stable filter than the UKF and has a square root solution providing numerical advantages and maintains the positive definiteness the of covariance matrix.

The UKF and CKF methods can be jointly considered as sigma point or derivative-free Kalman filters. The difference between these local filter is originated from the approximation used in computation of the integrals. The approximations based on the Taylor expansion, unscented transform, and cubature transform has a significant weakness which is the systematic error emerged by the approximate solution to the integrals [5]. To solve this problem, a randomized unscented Kalman filter (RUKF) has been proposed for solving the integrals without systematic errors very recently [5]. RUKF is based on the stochastic integration rule for infinite regions proposed by the Genz and Monohan [6].

In this paper, Randomized Sigma Point Kalman Filters (RSPKF) are used in simultaneous localization and mapping problem. To test the method in a more challenging SLAM problem, we introduce a novel landmark extraction method based on plane detection. Unlike the conventional methods, the 4D infinite plane parameters are encoded into the state vector and they are estimated with the latest 6D robot pose. The proposed observation model consists of dense trigonometric functions and cannot be considered as a mild nonlinear function; therefore, the RSPKF is obviously suitable for this type of problem. The appropriateness of the proposed SLAM method is validated through both simulations and experimental datasets in 2D and 3D, respectively. In 2D, point features are used as landmarks, and in 3D planes are used as landmarks in the SLAM state vector representation.

In Section 2, the Sigma Point Kalman Filter is introduced. Then the Randomized Sigma Point Kalman Filter (RSPKF) based on stochastic integration rule and RSPKF based SLAM method is presented in Section 3. Finally, the simulation and experimental results are given in Section 4 and a conclusion is drawn in Section 5.

2 Sigma Point Kalman Filter

The Sigma Point Kalman Filters (SPKFs) based on the unscented transform (UT) or cubature transform (CT) is introduced in this section.

2.1 Unscented Transformation

The aim of the unscented transformation is to calculate first two moments of a known nonlinear function $\mathbf{y}=\mathbf{g}(\mathbf{x})$ where $\mathbf{x}$ and $\mathbf{y}$ are the random vector variables. The mean vector $\bar{\mathbf{y}}$, the covariance matrix $\mathbf{P}_y$, and the cross-covariance matrix $\mathbf{P}_{xy}$ are described by

$$\begin{aligned}
&\bar{\mathbf{y}} = E[\mathbf{y}] = E[\mathbf{g}(\mathbf{x})] \\
&\mathbf{P}_y = \mathrm{cov}[y] = E[(\mathbf{y} - \bar{\mathbf{y}})(\mathbf{y} - \bar{\mathbf{y}})^T] \\
&\mathbf{P}_{xy} = E[(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{y} - \bar{\mathbf{y}})^T].
\end{aligned} \tag{1}$$

The solution to the problem with UT is based on the approximation of the random variable $\mathbf{x}$ by using a deterministically chosen set of sigma points, χ_i, and their corresponding weights w_i.

$$\begin{aligned}
&\chi_0 = \bar{\mathbf{x}}, \; w_0 = \frac{\kappa}{n+\kappa}, \\
&\chi_i = \bar{\mathbf{x}} + \left(\sqrt{(n+\kappa)\mathbf{P}}\right)_i, \; w_i = \frac{1}{2(n+\kappa)}, \\
&\chi_{n+i} = \bar{\mathbf{x}} - \left(\sqrt{(n+\kappa)\mathbf{P}}\right)_i, \; w_{i+n} = w_i,
\end{aligned} \tag{2}$$

where i=1,2, … n, and n is the dimension of the state vector. The term $(\bullet)_i$ represents the i^{th} column of the matrix. The covariance matrix satisfy the definition of $\mathbf{P}=\mathbf{S}\mathbf{S}^T$ where $\mathbf{S}$ is the square root of $\mathbf{P}$. Then sigma points are propagated based on the nonlinear function $\mathbf{g}(\mathbf{x})$ as

$$y_i = \mathbf{g}(\chi_i), \; \forall i. \tag{3}$$

Then the mean and covariance values are approximated as follows

$$\begin{aligned}
&\bar{\mathbf{y}}^{ut} = \sum_{i=0}^{2n} w_i y_i \\
&\mathbf{P}_y^{ut} = \sum_{i=0}^{2n} w_i (y_i - \bar{\mathbf{y}}^{ut})(y_i - \bar{\mathbf{y}}^{ut})^T \\
&\mathbf{P}_{xy}^{ut} = \sum_{i=0}^{2n} w_i (\chi_i - \bar{\mathbf{x}})(y_i - \bar{\mathbf{y}}^{ut})^T
\end{aligned} \tag{4}$$

The variable κ is the scaling parameter and suggested setting is the $\kappa = 3 - n$ [3]. However, the positive semi-definiteness is lost for multi-dimensional variable $\mathbf{x}$, which is the indispensable occasion of SLAM methods, because of negative κ ($n > 3$, $\kappa < 0$). For that reason, a possible practical solution is to choose $\kappa = 0$ for the multi-dimensional case although there is no mathematical justification. Moreover, the adaptive setting of the scaling parameters may improve the estimation accuracy of the UT [7].

2.2 Cubature Transformation

Cubature transformation, a more accurate and mathematically principled transformation than the UT, is proposed by Arasaratnam and Haykin [4]. The cubature transformation is based on the cubature theory and it is summarized as follows.

The key point of the Cubature theory [8] is to find multi-dimensional integrals using cubature rules since its integrands are in the form of,

$$non\text{-}linear\ function \times Gaussian.$$

Thus, the Bayesian filter solution is approximated by the help of cubature theory.

Cubature Rules

The cubature rule is used to approximate an *n*-dimensional Gaussian weighted integral as

$$\int_{R^n} f(x)N(\mathbf{x};\bar{\mathbf{x}},\mathbf{P})dx \approx \frac{1}{2n}\sum_{i=1}^{2n} f(\bar{\mathbf{x}}+\sqrt{\mathbf{P}}\zeta_i) \tag{5}$$

where N is the normal distribution of $\mathbf{x}$ with mean $\bar{\mathbf{x}}$ and covariance matrix $\mathbf{P}$. The relation for covariance matrix $\mathbf{P} = \sqrt{\mathbf{P}}\sqrt{\mathbf{P}}^T$ is satisfied. The $2n$ set of cubature array set is defined by ζ, and ζ_i the i^{th} element of the set ζ,

$$\zeta = \sqrt{n}\left\{\begin{pmatrix}1\\.\\.\\.\\0\end{pmatrix},...,\begin{pmatrix}0\\.\\.\\.\\1\end{pmatrix},\begin{pmatrix}-1\\.\\.\\.\\0\end{pmatrix},...,\begin{pmatrix}0\\.\\.\\.\\-1\end{pmatrix}\right\}. \tag{6}$$

2.3 Sigma Point Kalman Filters: UKF and CKF

The UKF and CKF are jointly called as Sigma Point Kalman filters (SPKF). The SPKF methods are based on either the unscented or the cubature transformations. Consider the following discrete time process and observation models.

$$\mathbf{x}_k = f(\mathbf{x}_{k-1},\mathbf{u}_{k-1}) + w_{k-1} \tag{7}$$

where w_{k-1} denotes the zero mean Gaussian distribution noise vector with covariance matrix Q, and $\mathbf{u}_{k-1}$ the control signal or odometry data. The two fundamental steps of the Kalman filters are explained as follows.

Time Update

In the time update step, SPKFs computes the predicted mean $\bar{\mathbf{x}}^-$ and covariance matrix $\mathbf{P}^-$ depending on the transformation.

$$\bar{\mathbf{x}}_k^- = E[f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + w_{k-1} \mid D_{k-1}] \tag{8}$$

where D_{k-1} denotes the history of the input and measurement pairs up to *k-1*. Since w_{k-1} is assumed to be zero mean and independent of the measurement sequence, one can write

$$\begin{aligned} \bar{\mathbf{x}}_k^- &= E[f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \mid D_{k-1}] \\ &= \int_{R^n} f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) p(\mathbf{x}_{k-1} \mid D_{k-1}) d\mathbf{x}_{k-1} \\ &= \int_{R^n} f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) N(\mathbf{x}_{k-1}; \bar{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1}) d\mathbf{x}_{k-1}. \end{aligned} \tag{9}$$

The corresponding error covariance matrix can be written as

$$\begin{aligned} \mathbf{P}_k^- &= E[(\mathbf{x}_k - \bar{\mathbf{x}}_k^-)(\mathbf{x}_k - \bar{\mathbf{x}}_k^-)^T \mid \mathrm{z}_{k-1}] \\ &= \int_{R^n} f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) f^T(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \\ &\quad x\, N(\mathbf{x}_{k-1}; \bar{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1}) d\mathbf{x}_{k-1} - \bar{\mathbf{x}}_{k-1} \bar{\mathbf{x}}_{k-1}^T + Q_{k-1}. \end{aligned} \tag{10}$$

Measurement Update. The predicted measurement vector, the corresponding covariance and cross covariance matrices are given by

$$\begin{aligned} \bar{\mathbf{z}}_k^- &= \int_{R^n} h(\mathbf{x}_k) N(\mathbf{x}_k, \bar{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1})\, d\mathbf{x}_k \\ P_{zz,k} &= \int_{R^n} h(\mathbf{x}_k) h^T(\mathbf{x}_k) N(\mathbf{x}_k; \bar{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1}) d\mathbf{x}_k - \bar{\mathbf{z}}_k \bar{\mathbf{z}}_k^T + R_k \\ P_{xz,k} &= \int_{R^n} \mathbf{x}_k h^T(\mathbf{x}_k) N(\mathbf{x}_k; \bar{\mathbf{x}}_{k-1}, \mathbf{P}_{k-1}) d\mathbf{x}_k - \bar{\mathbf{x}}_{k-1} \bar{\mathbf{z}}_k^T \end{aligned} \tag{11}$$

After new $\mathbf{z}_k$ measurements are obtained, the sigma point Kalman filter updates the state vector and covariance matrix as

$$\begin{aligned} \bar{\mathbf{x}}_k^+ &= \bar{\mathbf{x}}_k^- + K_k(\mathbf{z}_k - \bar{\mathbf{z}}_k) \\ P_k &= P_k^- - K_k P_{zz,k} K_k^T \end{aligned} \tag{12}$$

where the K_k is the Kalman gain given by

$$K_k = P_{xz,k} P_{zz,k}^{-1} \tag{13}$$

The main difference between the UKF and CKF is the approximations used to solve the given integrals. While the UKF filters uses unscented transform, CKF uses the cubature transform for solving the integrals. In the next subsection, we state the problem of the systematic error caused by approximations.

2.4 Problem Statement

The sigma point filters provides an approximate solutions to the nonlinear functions. However, these approximations are biased and generate systematic errors. To keep the equations more certain, we elucidate this situation on the UT. The error ε^{ut} is expressed by means of the Taylor expansion of the actual mean and the approximate mean [3].

$$\begin{aligned}\varepsilon^{ut} &= E\left[\frac{D^4_{\Delta x}\mathbf{g}(\mathbf{x})}{4!}+\frac{D^6_{\Delta x}\mathbf{g}(\mathbf{x})}{6!}+\ldots\right] \\ &-\frac{1}{2(n+\kappa)}\sum_{p=1}^{2n}\left(\frac{D^4_{\varepsilon_p}\mathbf{g}(\chi_p)}{4!}+\frac{D^6_{\varepsilon_p}\mathbf{g}(\chi_p)}{6!}+\ldots\right)\end{aligned} \tag{14}$$

where

$$\frac{D^k_{\varepsilon_p}\mathbf{g}(\mathbf{x})}{k!}=\frac{1}{k!}\left(\sum_{i=1}^{n}(\chi_p(i)-\bar{\mathbf{x}}_i)\frac{\partial}{\partial\chi_p(i)}\right)^k\mathbf{g}(\chi_p)\,|_{\chi_p=\bar{x}} \tag{15}$$

is the k^{th} term of the Taylor series expansion of the p^{th} sigma point $\mathbf{g}(\chi_p)$ and $\chi_p(i)$ is the ith element of χ_p. The error ε^{ut} is different from zero if the function **g** is not a polynomial of degree *2n*. This systematic error is also appears in the computations of the covariance matrices in a similar fashion.

In the next section, the randomized sigma point Kalman filter is presented to eliminate the mentioned systematic error.

3 Randomized SPKF Based Slam

Randomized Sigma Point Kalman Filter (RSPKF) proposed by Dunik et al. [5] uses the stochastic integration rule (SIR) introduced by Genz and Monohan [6]. SIR is explained as follows.

3.1 Stochastic Integration Rule (SIR)

SIR is appropriate for solving the integral of the form

$$\mu=\int_{R^n}\mathbf{g}(\mathbf{x})\left(\frac{1}{2\pi}\right)^{n/2}\mathrm{e}^{-\frac{1}{2}\mathbf{x}^T\mathbf{x}}. \tag{16}$$

This relation can be considered as a computation of the expected value of the function **g** where x is a random variable with $p(\mathbf{x})=N(\mathbf{x};\bar{\mathbf{x}},\mathbf{P})$. The algorithm to solve the integral (16) based on SIR is given by the Algorithm 1 in Table 1.

Table 1. Algorithm 1. Stochastic Integration Rule

Algorithm 1. $\mu = \mathbf{SI}(\bar{\mathbf{x}}, \mathbf{P}, \mathbf{g}(\mathbf{x}))$

1: Define N_{max}
2: Set $\mu = \mathbf{0}$ and compute $\chi_0 = \mathbf{g}(\bar{\mathbf{x}})$
3: **for** i=1 to N_{max} do
4: Generate a uniformly random orthogonal matrix $\mathbf{Q} \in R^{nxn}$ and generate a random number ρ form Chi-distribution with $n+2$ degrees of freedom.
5: Compute a set of points $\boldsymbol{\chi}_i$ and corresponding weights w_i according to

$$\boldsymbol{\chi}_i = -\rho\sqrt{P}\mathbf{Q}\mathbf{e}_i$$

$$\boldsymbol{\chi}_{n+i} = \rho\sqrt{P}\mathbf{Q}\mathbf{e}_i$$

$$w_0 = 1 - \frac{n}{\rho^2},\ w_i = w_{n+i} = \frac{1}{2\rho^2}$$

where i=1,2…n and $\mathbf{e}_i$ is the i^{th} column of the identity matrix.

6: Compute the value **S** of the integral at current iteration

$$\mathbf{S} = -\boldsymbol{\chi}_0\omega_0 + \sum_{i=0}^{n_x}\left(\mathbf{g}(\boldsymbol{\chi}_i) + \mathbf{g}(\boldsymbol{\chi}_{i+n_x})\right)\omega_i$$

and use it to update the approximate mean μ

$$\mu = \mu + (\mathbf{S} - \mu)/i$$

7: **end for**
8: return μ

The matrix **Q** can be generated using a product of appropriately chosen random reflections [6].

3.2 Randomized Sigma Point Kalman Filter

The time update and the measurement update steps of the filter are given as follows.

Time Update
The relations of the time update step is previously given by the equations (8) and (10). Here the integrals are solved by the SIR algorithm.

$$\begin{aligned}\bar{\mathbf{x}}_k^- &= \mathbf{SI}\left(\bar{\mathbf{x}}_k, \mathbf{P}_k, f_k(\mathbf{x}_k)\right)\\ \mathbf{P}_k^- &= \mathbf{SI}\left(\bar{\mathbf{x}}_k, \mathbf{P}_k, (f_k(\mathbf{x}_k) - \bar{\mathbf{x}}_k)(f_k(\mathbf{x}_k) - \bar{\mathbf{x}}_k)^T\right) + Q_k\end{aligned} \tag{17}$$

Measurement Update
The relations of the measurement update step is previously given by the equations (11) and the integral obtained by the SIR as

$$\begin{aligned}
\bar{\mathbf{z}}_k^- &= \mathbf{SI}\left(\bar{\mathbf{x}}_k^-, \mathbf{P}_k^-, h_k(\mathbf{x}_k)\right) \\
\mathbf{P}_{zz,k} &= \mathbf{SI}\left(\bar{\mathbf{x}}_k^-, \mathbf{P}_k^-, (h_k(\mathbf{x}_k) - \bar{\mathbf{z}}_k^-)(h_k(\mathbf{x}_k) - \bar{\mathbf{z}}_k^-)^T\right) + R_k \\
\mathbf{P}_{xz,k} &= \mathbf{SI}\left(\bar{\mathbf{x}}_k^-, \mathbf{P}_k^-, (\mathbf{x}_k - \bar{\mathbf{x}}_k^-)(h_k(\mathbf{x}_k) - \bar{\mathbf{z}}_k^-)^T\right) + R_k
\end{aligned} \tag{18}$$

Finally, the estimated state vector and covariance matrix is computed as in (12).

3.3 SLAM Based on RSPKF

The conventional Fb-SLAM representation consists of three models, which are vehicle model *f*, observation model *h*, and the augmentation model *g*. These representations are expressed as follows.

Vehicle Model
The vehicle model is given by *f*

$$\mathbf{x}_{v_k} = f(\mathbf{x}_{v_{k-1}}, \mathbf{u}_{k-1}) + w_{k-1} \tag{19}$$

where w_{k-1} denotes the zero mean Gaussian distribution noise vector with the covariance matrix Q, and the control signal **u.**

Landmark Model
The landmarks are assumed as stationary $\mathbf{x}_{m_k} = \mathbf{x}_{m_{k+1}}$ and represented in world (W) frame. The SLAM map is augmented with the following state vector representation,

$$\mathbf{x}^a{}_{k+1} = \begin{bmatrix} \mathbf{x}_{v_k} & \mathbf{x}_{m_{k+1}} \end{bmatrix} \in \mathrm{R}^{6+4\mathrm{N}}. \tag{20}$$

Observation Model
Measurement or observation model parameters, z_k are provided by the feature extraction method and stated as

$$z_k = h(\mathbf{x}, \mathbf{u}_{k-1}) + v_{k-1} \tag{21}$$

where *h* is the measurement model and v_{k-1} is the zero mean observation noise with *R* error covariance matrix.

Motion Update

Motion update step is based on the vehicle model (19). The state and covariance matrix is augmented as

$$\begin{aligned} \overline{\mathbf{x}}_k &= \begin{bmatrix} \overline{\mathbf{x}}_{v_k} & \mathbf{u}_k \end{bmatrix}^T \\ \mathbf{P}_k &= \begin{bmatrix} \mathbf{P}_k & 0 \\ 0 & Q_k \end{bmatrix} \end{aligned} \tag{22}$$

where $\overline{\mathbf{x}}_k$ is the state vector in the k^{th} time step and $\mathbf{u}_k$ is the applied control signal at this time. $\mathbf{P}_k$ denotes the state covariance matrix and augmented as in (22). The square root, **S**, of the covariance matrix, $\mathbf{P}_k$, is obtained by the Cholesky decomposition **S**=*chol*($\mathbf{P}_k$). Then the time prediction step of the RSPKF algorithm is applied to the augmented vectors (22). The error covariance matrix of the motion is shown by Q_k .

$$\begin{aligned} \overline{\mathbf{x}}_k^- &= \mathbf{SI}\left(\overline{\mathbf{x}}_k, \mathbf{P}_k, f_k(\mathbf{x}_k)\right) \\ \mathbf{P}_k^- &= \mathbf{SI}\left(\overline{\mathbf{x}}_k, \mathbf{P}_k, (f_k(\mathbf{x}_k) - \overline{\mathbf{x}}_k)(f_k(\mathbf{x}_k) - \overline{\mathbf{x}}_k)^T\right) + Q_k \end{aligned} \tag{23}$$

Measurement Update

The measurement update step is based on the observation model (21) and the state and covariance estimations are obtained using the SIR as in (18)

$$\begin{aligned} \overline{\mathbf{x}}_k^+ &= \overline{\mathbf{x}}_k^- + K_k(\mathbf{z}_k - \overline{\mathbf{z}}_k) \\ P_k &= P_k^- - K_k P_{zz,k} K_k^T \end{aligned} \tag{24}$$

where the K_k is the Kalman gain given by

$$K_k = P_{xz,k} P_{zz,k}^{-1} \tag{25}$$

State Augmentation

The state augmentation is based on the augmentation given by (20) and operated in every new landmark observations. The state augmentation is applied in two steps. Firstly, the state vector and covariance matrix is augmented with the new observations as follows.

$$\begin{aligned} \overline{\mathbf{x}}_k^+ &= \begin{bmatrix} \overline{\mathbf{x}}_{v_k} & z_k \end{bmatrix}^T \\ \mathbf{P}_k &= \begin{bmatrix} \mathbf{P}_k & 0 \\ 0 & R_k \end{bmatrix} \end{aligned} \tag{26}$$

where R_k is the error covariance matrix of the measurement.

The augmented state model $g^a = [\mathbf{x}_v \quad x_m]$ is constructed, and then the augmented state vector and covariance matrix are computed by following the same procedure in motion update step with (22) and (23) which are restated here to save space.

4 Performance Evaluatıons

RSPKF based SLAM performance is compared to the SPKF based SLAM in both simulations and experimental in 2D and 3D respectively.

4.1 Simulation Results in 2D

In this section, an artificial environment containing landmark position in 2D is generated and the robot way points are defined. The aim of the SLAM algorithm is to estimate the landmark positions and last robot pose information using the range and bearing observation model.

Vehicle Model
The explicit vehicle model (19) is given by

$$\begin{aligned} x_{v_k} &= x_{v_{k-1}} + Vdtcos(\gamma_{k-1} + \phi_{v_{k-1}}) + w_{x_{k-1}} \\ y_{v_k} &= y_{v_{k-1}} + Vdtsin(\gamma_{k-1} + \phi_{v_{k-1}}) + w_{y_{k-1}} \\ \phi_{v_k} &= \phi_{v_{k-1}} + Vdtsin(\gamma_{k-1}) / L + w_{\phi_{k-1}} \end{aligned} \tag{27}$$

where V and γ are the control input representing the constant velocity and steering angle with zero mean Gaussian noise w, respectively, and ϕ_{v_k} denotes the vehicle heading angle at time k.

Observation Model
The range and bearing observation model (21) is

$$\begin{aligned} r &= \sqrt{(x_m - x_{v_k})^2 + (y_m - y_{v_k})^2} + v_{r_k} \\ b &= tan^{-1}(\frac{y_m - y_{v_k}}{x_m - x_{v_k}}) - \phi_{v_k} + v_{b_k} \end{aligned} \tag{28}$$

where r is the range and b represents the bearing measurements, and the measurements are with zero mean Gaussian noise v.

State Augmentation Model

The state augmentation model $\mathbf{x}_{m_k}$ is given by

$$\begin{aligned} x_{m_k} &= x_{v_k} + r\cos(\phi_{v_k} + b_k) \\ y_{m_k} &= y_{v_k} + r\sin(\phi_{v_k} + b_k) \end{aligned} \tag{29}$$

The Monte Carlo simulations are carried out and the average position and orientation error norms for UKF and RSPKF SLAM methods are shown in Fig. 1. The control noise is 1 m/s in speed and 1 degree in steering angle. Similarly, the measurement noise in range and bearing is assumed as 1 meter and 1 degree, respectively. Therefore, the process covariance matrix Q=R=*diag*(1, π/180). The vehicle speed is taken as 2 m/s and time interval between two control signals is set by 0.05 seconds. The time interval between the two observations is assumed as 2.5 seconds.

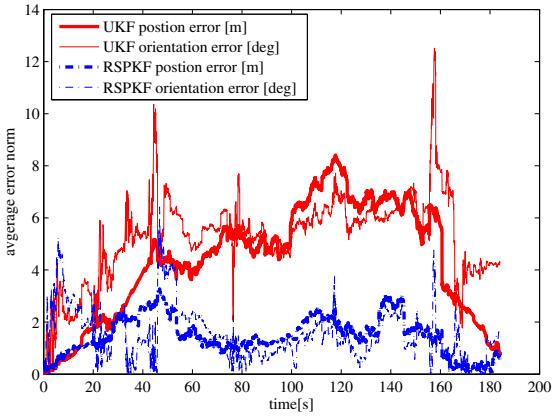

Fig. 1. Average position error norm

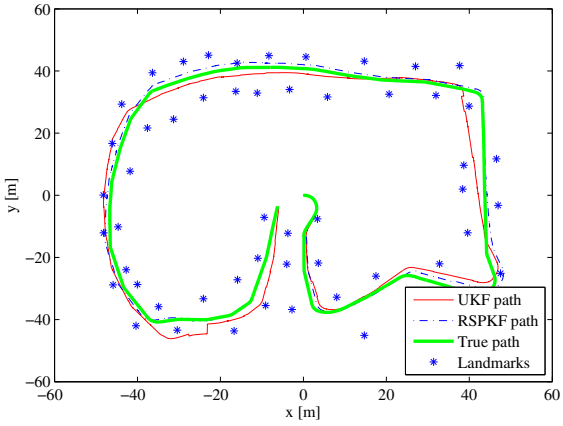

Fig. 2. The feature map and the estimated robot path

The feature map and the estimated paths based on the filtering methods are shown in Fig. 2.

4.2 Experimental Results in 3D

In this section, an experimental data set provided by Oliver Wulf is used [9]. This data set was recorded at the Leibniz University Campus and contains 468 3D scans, each with approximately 20,000 data points. A scan is given by three columns in *x*, *y*, and *z-axes*. The initial pose estimates are given by x_v, y_v as position, and θ_v as orientation in 3D. The ground truth pose data is available in 6D, $x_v = [x_{v_p} \quad x_{v_o}]$, and the size of the map is about 30 meter by 60 meter. The proposed RPSKF-SLAM method requires the Gaussian noise; however, the relative Odometry error variation is neither zero-mean nor Gaussian as shown in Fig. 3. Therefore, the problem becomes more challenging with respect to the Gaussian case. The vehicle model, observation model, and the augmentation models are expressed below.

Vehicle Model

The vehicle model function given by *f,* and it can be disclosed explicitly as in (30) for the odometry data having the relative rigid body transformation parameters. The odometry data is provided by the relations of $\mathbf{u} = [\delta x \ \delta y \ \delta z \ \delta\alpha \ \delta\beta \ \delta\gamma]$. The vehicle state vector is represented by $\mathbf{x}_v = [\mathbf{x}_{v_p} \quad \mathbf{x}_{v_o}]$ where $\mathbf{x}_{v_p} = [x \ y \ z]$ and $\mathbf{x}_{v_o} = [\alpha \ \beta \ \gamma]$ denote the robot position and the orientation, respectively. In the vector representations, the transpose *T* symbol is dropped for convenience.

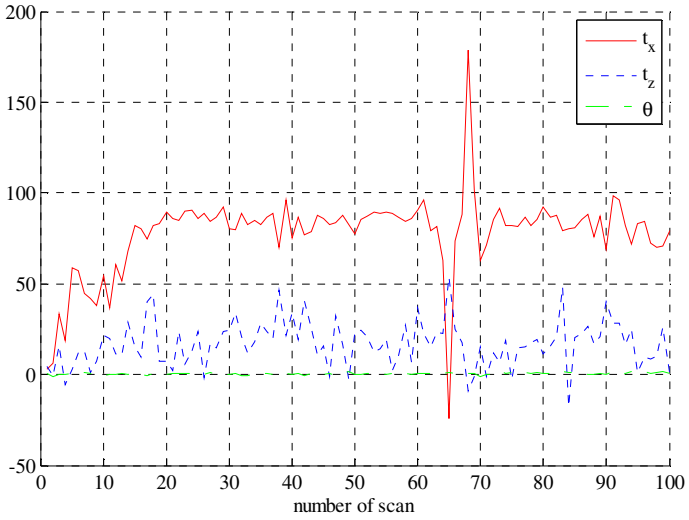

Fig. 3. Odometry error variation for the first 100 scans in Hannover dataset [9]. Translation errors are in cm and rotation error is in radian.

$$\begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ z_{k-1} \end{bmatrix} + Rot(x_{v_{o,k}}) \begin{bmatrix} \delta x_{k-1} \\ \delta y_{k-1} \\ \delta z_{k-1} \end{bmatrix}$$
$$\begin{bmatrix} \alpha_k \\ \beta_k \\ \gamma_k \end{bmatrix} = \begin{bmatrix} \alpha_{k-1} \\ \beta_{k-1} \\ \gamma_{k-1} \end{bmatrix} + \begin{bmatrix} \delta\alpha_{k-1} \\ \delta\beta_{k-1} \\ \delta\gamma_{k-1} \end{bmatrix} \tag{30}$$

where *Rot* matrix represent the three successive rotations defined by the Euler angles in *x*, *y*, and *z-axes*.

Observation Model

The observation model is based on the feature extraction method proposed by Ulas and Temeltas [10]. The plane features are used as landmarks and are encoded in the state vector with their infinite plane representations. The observation model *h* is given by

$$z_k = h(\mathbf{x}, \mathbf{u}_{k-1}) + v_{k-1}$$
$$z_k = \begin{bmatrix} n_{F_k}^L \\ d_{F_k}^L \end{bmatrix} = \begin{bmatrix} Rot^T(x_{v_{o,k}}) n_{F_k}^W \\ d_{F_k}^W + (n_{F_k}^W)^T x_{v_{p,k}} \end{bmatrix} + v_{k-1} \tag{31}$$

where $n_{F_k}^L$ is the plane normal vector represented in the local (*L*) frame, and $d_{F_k}^L$ is the plane minimum distance to the robot location $x_{v_{p,k}}$ provided by the feature extraction method. The reader is referred to [10] for more information about the feature extraction method. Based on the robot orientation $x_{v_{o,k}}$ and location $x_{v_{p,k}}$ the plane patch parameters are transformed to the world (W) frame for state augmentation.

State Augmentation Model

The state augmentation model is given by

$$n_F^W = Rot(x_{v_o}) n_F^L$$
$$d_F^W = d_F^L - (n_F^W)^T x_{v_p} \tag{32}$$

The infinite plane representations in local and world frame are shown in Fig. 4.

For the data association purpose, the other plane properties such as center of gravity of the planes G_F^L, the covariance matrix C_F^L of the plane points and convex hull points $\Delta^L{}_{XYZ,F}$ are also transferred to the world frame by using the estimated robot position.

$$
\begin{aligned}
C_F^W &= Rot(x_{v_o})C_F^L \\
G_F^W &= Rot(x_o)G_F^L + x_{v_p} \\
\Delta^W{}_{XYZ,F} &= Rot(x_{v_o})\Delta^L{}_{XYZ,F} + x_{v_p}
\end{aligned}
\tag{33}
$$

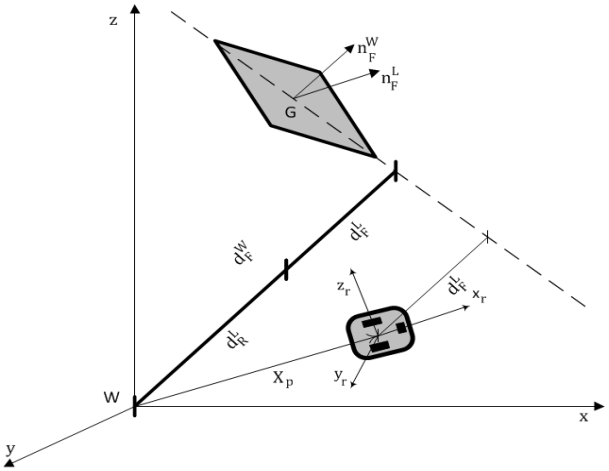

Fig. 4. Infinite Plane representation in local and world frame

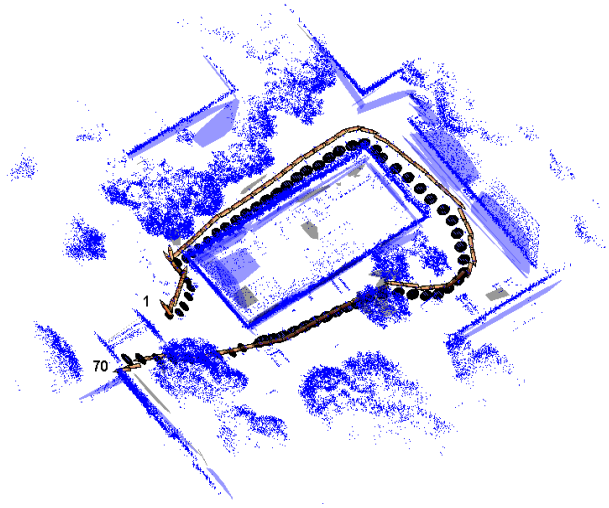

Fig. 5. Estimated planar map of the environment. The estimated robot position with uncertainty ellipsoids and ground truth path (orange) are shown.

In Fig. 5, the planar map constructed form the SLAM and the actual robot path is shown. In addition, the error uncertainty ellipsoids of the robot 3D position with their mean are shown on the map. Here, the point cloud is registered based on the ground truth as the reference.

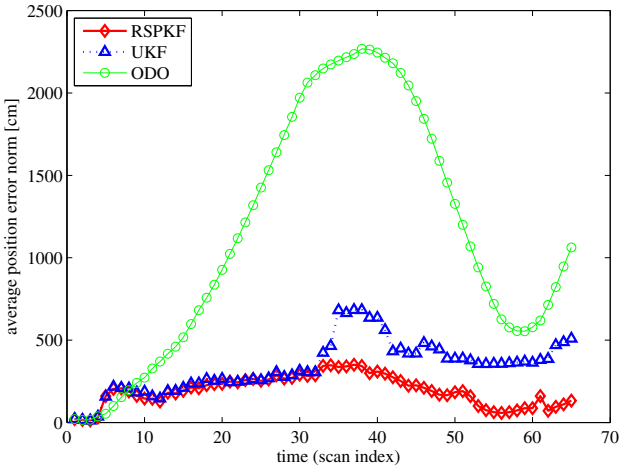

Fig. 6. Average position error norm

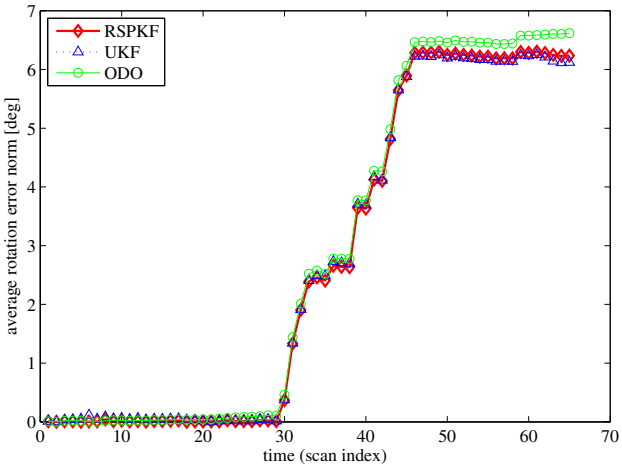

Fig. 7. Average rotation error norm

The results show that UKF and CKF based SLAM satisfy the similar results with a maximum position error around 6.5 meters (in the 35 time index). On the other hand, the RSPKF based SLAM has a maximum of 4 meter position error norm and more accurate than the conventional sigma point approaches. The rotation error norm looks similar for all filter types.

5 Conclusion

In this study, a new filtering method based on randomized sigma point sampling is introduced for localization mapping problem. The advantage of the proposed method is that the estimations are unbiased and does not yield the systematic error which is always the case of the classical filtering approaches. The performance evaluations are given for both simulations and experimental data. The proposed method is more accurate than the traditional sigma point Kalman filters like UKF and CKF which have similar performances. Although this approximation takes more computational time, it can be used accurately in SLAM problems without any systematic error.

Acknowledgments. This work was supported in part by the TUBITAK under Grant 110E194.

References

1. Dissanayake, M.W.M.G., Newman, P., Clark, S., Durrant-Whyte, H.F., Csorba, M.: A solution to the simultaneous localization and map building (SLAM) problem. IEEE Transactions on Robotics and Automation 17, 229–241 (2001)
2. Thrun, W.B.S., Fox, D.: Probabilistic Robotics. MIT Press (2005)
3. Julier, S.J., Uhlmann, J.K.: New extension of the Kalman filter to nonlinear systems. In: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 3068, pp. 182–193 (1997)
4. Arasaratnam, I., Haykin, S.: Cubature Kalman Filters. IEEE Transactions on Automatic Control 54, 1254–1269 (2009)
5. Dunik, J., Straka, O., Simandl, M.: The Development of a Randomised Unscented Kalman Filter. In: Proceedings of the 18th IFAC World Congress, Milano, Italy (2011)
6. Genz, A., Monahan, J.: Stochastic Integration Rules for Infinite Regions. SIAM J. Sci. Comput. 19, 426–439 (1998)
7. Dunik, J., Simandl, M., Straka, O.: Adaptive choice of scaling parameter in derivative-free local filters. In: 13th Conference on Information Fusion (FUSION), pp. 1–8 (2010)
8. Fischler, M.A., Bolles, R.C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM 24, 381–395 (1981)
9. Wulf, O.: Hannover, Leibniz University Campus, `http://kos.informatik.uni-osnabrueck.de/3Dscans/`
10. Ulas, C., Temeltas, H.: A fast and robust scan matching algorithm based on ML-NDT and feature extraction. In: International Conference on Mechatronics and Automation (ICMA), pp. 1751–1756 (2011)