J Intell Robot Syst (2011) 63:575–594
DOI 10.1007/s10846-010-9495-7

Experimental Study of Iterated Kalman Filters for Simultaneous Localization and Mapping of Autonomous Mobile Robots

Khoshnam Shojaei · Alireza Mohammad Shahri

Received: 29 October 2009 / Accepted: 13 October 2010 / Published online: 12 November 2010

Abstract In this paper, we investigate the role of iteration in Kalman filters family for improvement of the estimation accuracy of states in simultaneous localization and mapping (SLAM). The linearized error propagation existing in Kalman filters family can result in large errors and inconsistency in the SLAM problem. One approach to alleviate this situation is the use of iteration in extended Kalman filter (EKF) and sigma point Kalman filter (SPKF) based SLAM. The main contribution is to present that the iterated versions of Kalman filters can increase consistency and robustness of these filters against linear error propagation. Experimental results are presented to validate this improvement of state estimate convergence through repetitive linearization of the nonlinear observation model in EKF-SLAM and SPKF-SLAM algorithms.

Keywords Consistency · Extended Kalman filter · Iterated Kalman filters · Mobile robot · Nonlinear estimation · Sigma point Kalman filter · SLAM

1 Introduction

The most popular approach for navigation of an autonomous mobile agent is simultaneous localization and mapping (SLAM). This fundamental problem has been investigated by many researchers in the mobile robotic community in recent

K. Shojaei · A. Mohammad Shahri (✉)
Mechatronics and Robotics Research Laboratory, Electronic Research Center,
Electrical Engineering Department, Iran University of Science and Technology,
Tehran, Iran
e-mail: shahri@iust.ac.ir

K. Shojaei
e-mail: shojaei@ee.iust.ac.ir

years. A variety of solutions to the SLAM problem have been surveyed in [13]. This problem can be explained as follows: the autonomous mobile robot starts from an initial pose without a prior knowledge of the environment and trys to get perceptions from its surroundings by its onboard sensors measurements. The robot fuses these measurements at the heart of a nonlinear Bayesian estimator to estimate the location of salient features of the environment (mapping process) and simultaneously estimates its own pose in this incrementally built map (localization process) [1, 12].

Kalman filters family has shown its competency to solve this problem as proposed in the literature [1, 12, 17, 19, 21]. The most dominant approach as a solution to the SLAM problem is extended Kalman filter (EKF) for nonlinear filtering which was proposed in a seminal work by Smith et al. [11]. Recently, a number of derivative free alternatives to the EKF including *unscented Kalman filter* (UKF), *central difference Kalman filter* (CDKF) and their square root versions [5, 6] have been proposed as useful tools for integrated navigation [8] after their first introduction by Julier and Uhlmann [4]. Some researchers have applied these new extensions of the Kalman filters family called *Sigma Point Kalman Filters* (SPKF) to solve the SLAM problem [3, 7]. In spite of superiority of the SPKF over the EKF, both of these filters suffer from analytical and statistical linear error propagation, respectively. The linearization method in these filters may cause inconsistency in SLAM problem [9, 14].

Because of strong correlation between robot pose and map estimate errors in SLAM algorithm structure, every error caused by linearization of the process and measurement models will be propagated through the whole of the map and robot pose. This correlation of the states is essential to maintain consistency of the algorithm [15]. However, linear error propagation due to this correlation in EKF-SLAM and SPKF-SLAM algorithms will jeopardize this consistency. Therefore, consistency evaluation of such nonlinear estimators is necessary toward a robust solution to the SLAM problem.

To the best of our knowledge, there are little works to experimentally study the improvement of Kalman filters based SLAM algorithms through the modification of filter structure. Guivant [22] proposed compressed EKF to improve the computational efficiency of EKF-SLAM in large environments. Zhou et al. [21] proposed mean extended Kalman filter (MEKF) to improve estimation accuracy and computational complexity in EKF based SLAM. In the direction of aforementioned valuable works, the main contribution of this paper is to investigate experimentally whether the iteration of linearization in EKF and SPKF can lead to a more robust and consistent solution to the SLAM problem. In this regard, some computational analyses (such as Monte Carlo simulation) are performed in this paper. The mathematical interpretation and experimental results of the SLAM algorithms based on iterated extended Kalman filter and iterated sigma point Kalman filter are presented as strong points of the current work. The rest of the paper is structured as follows: Section 2 summarizes the EKF based SLAM algorithm in a probabilistic form. Then, we will briefly describe the SPKF based SLAM in Section 3. In Section 4, the mathematical derivation of the iterated Kalman filters is presented by Gauss-Newton method. Section 5 reviews the consistency checking approach to compare filters performance. Section 6 compares estimation accuracy, computational complexity and consistency of the EKF-SLAM and SPKF-SLAM solutions with their iterated versions by simulation results. All mentioned algorithms have been successfully implemented on NAJI mobile robot in our laboratory and experimental results are presented to validate simulation results in this section. Finally, Section 7 concludes the paper.

2 SLAM Algorithm

In this section, the probabilistic form of the SLAM algorithm is reviewed. An autonomous mobile robot starts from an arbitrary initial pose in an unknown environment and gets measurements from its extroceptive sensors such as sonar and laser range finders. The robot normally fuses these measurements with the proprioceptive sensors readings (i.e., dead reckoning sensors data) at the heart of a Bayesian nonlinear filter to incrementally estimate the location of geometric features in the environment (mapping). The robot determines its own pose relative to this estimated map simultaneously (localization) [1, 12, 15]. The recursive Bayesian formulation of the SLAM algorithm can be written as follows [19]. The state vector of the robot at time step k is denoted by $X_{r,k}$. Assuming N stationary features in the environment, the state vector of ith feature is $X_{i,k}$. We are searching for the posterior Probability Distribution Function (PDF):

$$P\big(X_k|Z^k, U^k\big) \sim N\big(X_k;\ \hat{X}_k^+, P_k^+\big), \tag{1}$$

where Z^k and U^k are measurements and control inputs up to and including time step k, respectively. $\hat{X}_k^+$ and P_k^+ are the mean and covariance of the probability distribution. $X_k = [X_{r,k}^T,\ X_{m,k}^T]^T$ is the augmented state vector such that $m = 1, \ldots, N$. Therefore, this probability distribution can be explained as the conditional joint PDF of the robot pose and features position. The prior PDF is also defined by

$$P\big(X_k|Z^{k-1}, U^k\big) \sim N\big(X_k;\ \hat{X}_k^-, P_k^-\big), \tag{2}$$

which is calculated based on proprioceptive sensors data within time-update step:

$$P\big(X_k|Z^{k-1}, U^k\big) = \int P\big(X_{r,k}|X_{r,k-1},\ U^k\big)\,.\,P\big(X_{k-1}|Z^{k-1}, U^{k-1}\big)\,dx_{k-1}. \tag{3}$$

The posterior PDF will be computed by correction of prior PDF after receiving data from extroceptive sensors by applying the Bayes' Formula within measurement-update step:

$$P\big(X_k|Z^k, U^k\big) = \eta\ P(z_k|X_{r,k},\ X_{m,k}).P\big(X_k|Z^{k-1}, U^k\big), \tag{4}$$

where $P(X_{r,k}|X_{r,k-1},\ U^k)$ is the action or process model in Eq. 3 that can be derived from the mobile robot kinematics, $P(z_k|X_{r,k},\ X_{m,k})$ is the measurement model and η is a normalizing constant [19].

3 SPKF Based SLAM

As frequently reported in literature [5, 10, 14], the analytical linearization of the nonlinear model in EKF can result in considerable estimation errors in robot pose and feature position in the SLAM algorithm. Here, we propose sigma point Kalman filter to avoid this type of linearization and improve state estimates. The heart of sigma point Kalman filter is a minimal set of deterministically chosen weighted sample points called *sigma-points*. These points completely capture true mean and covariance of the prior probability distribution of random variable [4]. Suppose that N-dimensional random state variable (RSV) X relates to variable y through the nonlinear function f:

$$y = f(X). \tag{5}$$

Probability distribution of X has mean $\hat{X}$ and its corresponding covariance matrix is P. First, we choose sigma points set with the following scheme. The number of sigma points is $2N + 1$ that first point is the mean of the probability distribution. The other points are created with displacement of the mean by adding and subtracting of the scaled covariance matrix square root. Therefore, these sigma points are symmetric around the mean as can be seen in Fig. 1.

$$\chi_0 = \hat{X} \tag{6}$$

$$\chi_i = \hat{X} + \left(\gamma\sqrt{P}\right)_i \ , \quad i = 1\,, \ldots, \ N \tag{7}$$

$$\chi_i = \hat{X} - \left(\gamma\sqrt{P}\right)_i \ , \ \ i = \ N + 1\,, \ldots, \ 2N, \tag{8}$$

where γ is a scaling factor that controls the spread of sigma points around the mean. For efficient calculation of covariance matrix square root, if the covariance matrix is positive semi-definite, the Cholesky factorization may be used [4, 5]. Then, sigma points matrix can be created as follows:

$$\chi\,(k) = [\,\chi_0(k)\,,\ \chi_i(k)\,,\ \chi_{i+N}(k)\,]\,,\ \ i = \ 1\,, \ldots, \ N. \tag{9}$$

Transformed sigma points are calculated by propagating of them through the nonlinear function f:

$$\mu_i = f(\chi_i) \quad i = 0, \ \ldots, \ 2N. \tag{10}$$

Finally, mean and covariance matrices of posterior distribution given by:

$$\hat{y} \approx \sum_{i=0}^{2N} w_i^m\, \mu_i \tag{11}$$

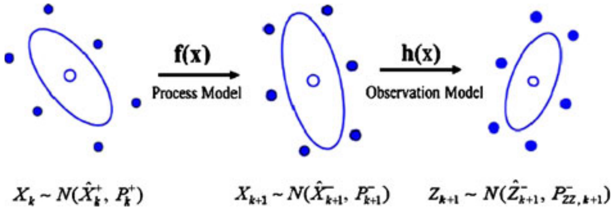

f(x)
Process Model
h(x)
Observation Model
$X_k \sim N(\hat{X}_k^+, P_k^+)$
$X_{k+1} \sim N(\hat{X}_{k+1}^-, P_{k+1}^-)$
$Z_{k+1} \sim N(\hat{Z}_{k+1}^-, P_{ZZ,k+1}^-)$

Fig. 1 Demonstration of the sigma points and probability distributions before and after passing through nonlinear models

$$P_y \approx \sum_{i=0}^{2N} w_i^C (\mu_i - \hat{y})(\mu_i - \hat{y})^T \tag{12}$$

$$P_{xy} \approx \sum_{i=0}^{2N} w_i^C (\chi_i - \hat{x})\,(\mu_i - \hat{y})^T. \tag{13}$$

These equations are accurate up to second order in Taylor series expansion. The parameters w_i^m and w_i^C are scalar positive valued weights. Superscripts (m) and (C) in these parameters indicate the weights related to mean and covariance of the PDF. The interested reader is referred to the references [2] and [5] for the details of choosing weights w_i^m and w_i^C. As one can see, there is no analytical derivative or Jacobian in above equations. This approach can be interpreted as statistical linearization of the nonlinear function by a technique called *weighted statistical linear regression* (WSLR) which is completely discussed in [5].

This type of linearization is more accurate than linearization via first order Taylor series expansion. The sigma point Kalman filters behave exactly like the second order Gauss filter, but there is no need to calculate Jacobians or Hessians [4]. We may use the square root versions of this family that perform numerically stable and efficient. Detailed formulation of SPKF based SLAM may be found in [20].

4 Mathematical Derivation of Iterated Kalman Filters

In this section, the measurement update equations for iterated Kalman filters are derived by Gauss-Newton method which is presented in [2] for the first time. The EKF linearizes process and measurement models around the latest state estimates. When models are highly nonlinear, the analytical linearization of models may result in large errors which cause divergence of the SLAM algorithm. A reasonable solution to overcome this difficulty is the iteration of the linearization. In order to derive update equations of iterated filters, one may consider the prior PDF and the measurement probability as follows:

$$P\big(X_k|Z^{k-1}, U^k\big) = \frac{1}{\sqrt{2\pi|P_k^-|}} \exp\big(-0.5\big(\hat{x}_k^- - x_k\big)(P_k^-)^{-1}\big(\hat{x}_k^- - x_k\big)^T\big) \tag{14}$$

$$P(z_k|X_{r,k}\,,\;\; X_{m,k}) = \frac{1}{\sqrt{2\pi|R|}} \exp\big(-0.5\big(z_k - h\big(\hat{x}_k^-\big)\big)R^{-1}\big(z_k - h\big(\hat{x}_k^-\big)\big)^T\big). \tag{15}$$

Form the viewpoint of Bayesian framework, we seek the x_k that maximizes $P(z_k|X_{r,k}\,,\;\; X_{m,k}).P(X_k|Z^{k-1}, U^k)$ in Eq. 4, or minimizes its logarithm equivalently:

$$L(x) = 0.5\big(z_k - h\big(\hat{x}_k^-\big)\big)R^{-1}\big(z_k - h\big(\hat{x}_k^-\big)\big)^T + 0.5\big(\hat{x}_k^- - x_k\big)(P_k^-)^{-1}\big(\hat{x}_k^- - x_k\big)^T + c, \tag{16}$$

where c is a constant and will be ignored in the sequel. By definition of the following functions and matrices:

$$Z = [z_k,\, \hat{x}_k^-]^T\;,\quad G(x) = [h(\hat{x}_k^-),\, x_k]^T\,,\;\; \Sigma = \begin{bmatrix} R & 0 \\ 0 & P_k^- \end{bmatrix}\quad . \tag{17}$$

The function $L(x)$ is approximated as follows:

$$L(x) \approx 0.5[(Z - G(x))\Sigma^{-1}(Z - G(x))] := \frac{1}{2} \parallel f(x) \parallel^2 , \tag{18}$$

where $f(x) = C(Z - G(x))$ and $\Sigma^{-1} = C^T C$. Now, one may use Newton solution to achieve the iterative sequence

$$x_{i+1} = x_i - (\nabla^2 L(x_i))^{-1} \nabla L(x_i), \tag{19}$$

We may remember that Gauss-Newton method approximates $\nabla^2 L(x_i) = F(x_i)^T F(x_i)$, where $F(x_i)$ is Jacobian of $f(x)$ and $F(x_i) = -CG_i$. The matrix G_i is also Jacobian of $G(x)$. Finally, noting that the gradient of $L(x)$ is $\nabla L(x_i) = F(x_i)^T f(x_i)$, we can achieve

$$x_{i+1} = \left(G_i^T \Sigma^{-1} G_i\right)^{-1} G_i^T \Sigma^{-1}(Z - G(x_i) + G_i x_i), \tag{20}$$

One may consider the following matrix inversion lemmas and expand the above equation.

$$(H^T R^{-1} H + P^{-1})^{-1} H^T R^{-1} = PH^T (HPH^T + R)^{-1} \tag{21}$$

$$(H^T R^{-1} H + P^{-1})^{-1} = P - PH^T (HPH^T + R)^{-1} HP \tag{22}$$

After some manipulation, the update equation of Iterated Extended Kalman Filter (IEKF) may be achieved as follows:

$$\hat{x}_k^{i+1} = \hat{x}_k^- + K_k\big(z_k - h\big(\hat{x}_k^i\big) - H_i\big(\hat{x}_k^- - \hat{x}_k^i\big)\big) \tag{23}$$

$$K_k = P_k^- H_i^T \left(H_i P_k^- H_i^T + R\right)^{-1}, \tag{24}$$

where K_k is Kalman gain. The covariance matrix is also approximated by mentioned lemmas as:

$$\begin{aligned} P_k^+ = \left(G_i^T \Sigma^{-1} G_i\right)^{-1} &= \left(H_i^T R^{-1} H_i + (P_k^-)^{-1}\right)^{-1} \\ &= P_k^- - P_k^- H_i^T \left(H_i P_k^- H_i^T + R\right)^{-1} H_i P_k^- . \end{aligned} \tag{25}$$

As a result, the IEKF repeatedly calculates the Kalman gain and an intermediate posterior state estimate $\hat{x}_k^i$, where i is the iteration number. The intermediate state estimate starts from the mean of prior PDF (i.e., $\hat{x}_k^0 = \hat{x}_k^-$) and the Jacobian matrix is calculated at the recent value of $\hat{x}_k^i$. After a certain number of iterations, IEKF computes the mean and covariance matrix of the posterior PDF based on the latest value of $\hat{x}_k^i$, K_k and H_k. In fact, the IEKF decreases the linearization error by re-linearizing the measurement model and tries to find the best estimate of the state [10]. Note that the IEKF only re-linearizes the observation model.

Figure 2 graphically shows re-linearization of the measurement function for two iterations. As shown by the figure, each intermediate state estimate $\hat{x}_k^i$ can be achieved by the true measurement z_k and the calculated line from re-linearization of the function around the previous intermediate state estimate $\hat{x}_k^{i-1}$. Whenever

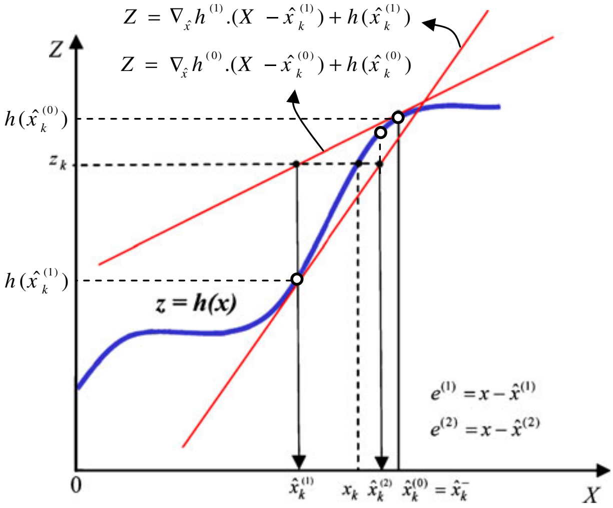

$Z = \nabla_{\hat{x}} h^{(1)}.(X - \hat{x}_k^{(1)}) + h(\hat{x}_k^{(1)})$
$Z = \nabla_{\hat{x}} h^{(0)}.(X - \hat{x}_k^{(0)}) + h(\hat{x}_k^{(0)})$
Z
$h(\hat{x}_k^{(0)})$
z_k
$h(\hat{x}_k^{(1)})$
z = h(x)
$e^{(1)} = x - \hat{x}^{(1)}$
$e^{(2)} = x - \hat{x}^{(2)}$
0
$\hat{x}_k^{(1)}$
x_k
$\hat{x}_k^{(2)}$
$\hat{x}_k^{(0)} = \hat{x}_k^-$
X

Fig. 2 Demonstration of the re-linearization process of the measurement model in IEKF for two iterations. In the ith step, a tangent line is plotted at the point $\hat{x}_k^{(i)}$ on the curve $h(x)$. Then, the next intermediate state $\hat{x}_k^{(i+1)}$ can be achieved through the intersection of the tangent line and the true measurement $Z = z_k$

the difference between two successive intermediate state estimates, i.e., $\Delta\hat{x}_k^i := \hat{x}_k^i - \hat{x}_k^{i-1}$, is less than a certain threshold, the re-linearization may be stopped. Thus, the linearization error at ith iteration, i.e., $e_k^i = x_k - x_k^i$, may be effectively reduced by re-linearization of the measurement model. Note that the IEKF achieves good results if the measurement model is relatively linear between the true state x_k and the calculated posterior intermediate state estimate after one linearization step. Otherwise, the linearization error increases due to re-linearization and the IEKF fails to improve the state estimate.

As mentioned before, the sigma point approach in SPKF can be interpreted as statistical linearization of the nonlinear function by WSLR. The iterated version of the SPKF approach can also improve the state estimate accuracy and filter consistency in SLAM algorithm [2]. The iterated sigma point Kalman filter update equations can be derived by considering linearized error propagation terms in Kalman filter:

$$cov\,(z) = HPH^T\,,\ \ cov\,(x, z) = PH^T. \tag{26}$$

These terms are defined for SPKF as:

$$cov(\mathrm{Z}) = \sum_{i=0}^{2N} w_i^C \left(\mu_{i,k+1}^-\right) . \left(\mu_{i,k+1}^-\right)^T,\ \ cov(\chi, \mathrm{Z}) = \sum_{i=0}^{2N} w_i^C \left(\xi_{i,k+1}^-\right) . \left(\mu_{i,k+1}^-\right)^T. \tag{27}$$

where $\mu_{i,k+1}^-$ is the difference between samples of predicted observation and the mean, $\xi_{i,k+1}^-$ denotes the difference between sigma points set and the mean of the prior PDF in the SPKF-based SLAM formulation which is well defined in reference [20]. By considering Eqs. 26 and 27 and IEKF update equations, we may compute the Kalman gain as follows:

$$K_k = cov(\chi, \mathrm{Z})\,(cov(\mathrm{Z}) + R)^{-1}. \tag{28}$$

The measurement update equations of ISPKF may also be written as:

$$\hat{x}_k^{i+1} = \hat{x}_k^- + K_k\big(z_k - h\big(\hat{x}_k^i\big) \ - cov(\chi, \mathrm{Z})^T P^{-1}\big(\hat{x}_k^- - \hat{x}_k^i\big)\big), \tag{29}$$

$$P_k^+ = P_k^- - cov(\chi, \mathrm{Z})\,(cov(\mathrm{Z}) + R)^{-1} cov(\chi, \mathrm{Z})^{\,T}. \tag{30}$$

In fact, the ISPKF approach tries to improve the state estimates by statistical relinearization of the measurement model. In the next section, we will compare the performance of EKF and SPKF approaches with their iterated versions for the sake of estimation accuracy and filter consistency.

5 Monte Carlo Simulations for Evaluation of SLAM Filters Efficiency and Consistency

Since the SLAM algorithm is a nonlinear problem, the consistency is necessary for filter optimality. According to definition, a consistent filter must satisfy two conditions: (1) unbiasedness, (2) the actual mean square error (MSE) matches the filter calculated covariances.

The practical procedure to assess the filter efficiency and consistency is using the *Monte Carlo simulation* that provides N independent samples ε_k^i , $i = 1, \ldots, N$ of the random variable ε_k. When a ground truth is available, the *normalized estimation error squared* (NEES) can be calculated over N Monte Carlo runs of the filters to characterize the performance of each filter. The NEES is defined as

$$\varepsilon(k) = e(k|k)^T.\, P(k|k)^{-1}.\, e(k|k), \tag{31}$$

where $\varepsilon(k)$ is chi-square distributed with n_x degrees of freedom where n_x is the dimension of x, $e_{k|k} = x_k - \hat{x}_{k|k}$ is the state estimation error and $P(k|k)$ is the covariance matrix of RSV. The average NEES can evaluate the filter consistency:

$$\bar{\varepsilon}(k) = \frac{1}{N}\sum_{i=1}^{N} \varepsilon_k^i. \tag{32}$$

Then, $\bar{\varepsilon}(k)$ will have a chi-squared density with Nn_x degrees of freedom. The state estimation errors are consistent with the calculated covariance matrix by the filter if

$$\bar{\varepsilon}(k) \in [r1, r2], \tag{33}$$

where $r1$ and $r2$ are the lower and upper limits of this acceptance interval [14]. If $\bar{\varepsilon}(k)$ exceeds the upper bound, the filter is optimistic. This means that the calculated covariance of the filter is too small compared to the actual MSE and it may lead to filter divergence. If $\bar{\varepsilon}(k)$ is below the lower bound, the filter is pessimistic which means that the covariance is too large. In the next section, the results of Monte Carlo simulations are presented.

6 Experimental Studies

6.1 Simulation Results

In this section, some computer simulation results are presented to confirm improving effects of the IEKF and ISPKF relative to non-iterated methods. A differential drive mobile robot is considered for simulation with the following process model:

$$\begin{bmatrix} x_{R,k+1} \\ y_{R,k+1} \\ \varphi_{R,k+1} \end{bmatrix} = \begin{bmatrix} x_{R,k} + \delta S .\ \cos\left(\varphi_{R,k} + \frac{\delta\varphi_R}{2}\right) \\ y_{R,k} + \delta S .\ \sin\left(\varphi_{R,k} + \frac{\delta\varphi_R}{2}\right) \\ \varphi_{R,k} + \delta\varphi_R \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \\ v_\varphi \end{bmatrix}, \tag{34}$$

where $\delta\varphi_R$ and δS are defined as:

$$\delta\varphi_R = \frac{\Delta s_r - \Delta s_l}{b}, \quad \delta S = \frac{\Delta s_r + \Delta s_l}{2}. \tag{35}$$

The parameter b denotes the distance between two wheels of differential drive robot. Δs_r and Δs_l are traveled distances for the right and left wheels, respectively. The nonsystematic odometry errors are considered in both Cartesian space and joint space which their covariance matrices are denoted by $Q_{3\times 3}$ and $U_{2\times 2}$, respectively, such that

$$U = cov(\Delta s_l, \Delta s_r) = \begin{bmatrix} k_r\,|\Delta s_r| & 0 \\ 0 & k_l\,|\Delta s_l| \end{bmatrix}, \tag{36}$$

where k_r and k_l are odometry error constants that are assumed 5×10^{-5} m in the simulation process [23]. The robot equipped with a laser range finder (LRF) with a maximum range of 10 meters to observe simple geometric features. These features or landmarks may be trunks of trees in an outdoor environment. The *inscribed angle variance* (IAV) algorithm [18] is used to cluster arc segments from laser range data. After that, a simple geometric method is applied to extract range and bearing of center of landmarks and calculate measurement covariance matrix that are described in the Appendix section. The robot starts from initial pose [0 m, –2 m, 0 rad.] and travels around a 34 × 29 meters simulated outdoor environment at a constant speed along a circular path whose radius is 10 meters. In order to simplify the simulations, an open-loop control strategy is considered to simulate the circular path. It is also assumed that the real locations of the robot and landmarks are measured by a global positioning system (GPS). A nearest neighbor standard filter (NNSF) is used for data association in SLAM algorithms. The observation model also can be written as follows:

$$\begin{bmatrix} Z_{r,k} \\ Z_{\theta,k} \end{bmatrix} = \begin{bmatrix} \sqrt{(x_{r,k} - x_{i,k})^2 + (y_{r,k} - y_{i,k})^2} \\ arctg\left(\frac{y_{r,k} - y_{i,k}}{x_{r,k} - x_{i,k}}\right) - \varphi_{r,k} \end{bmatrix} + \begin{bmatrix} w_r \\ w_\theta \end{bmatrix}, \tag{37}$$

The first simulation is carried out to compare the EKF and IEKF approaches. The second simulation is also shown to verify the superiority of the ISPKF over the SPKF in solving SLAM problem.

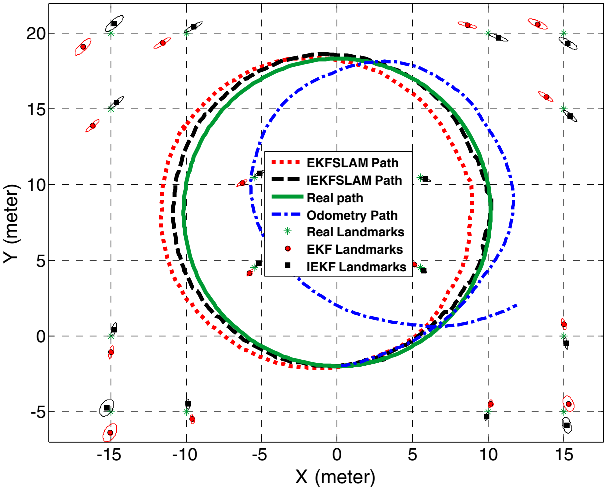


Fig. 3 The path and 2D map of the environment estimated by both EKF (*dotted line*) and IEKF (*dashed line*) based SLAM

In the first simulation, both EKF-SLAM and IEKF-SLAM algorithms are run simultaneously by a single program code in MATLAB software and the data are recorded for about 180 seconds for both algorithms. A large amount of odometry errors are considered for the robot to magnify the effects of iteration in EKF-SALM. The estimation error and uncertainty will rise in prior PDF by increasing the odometry errors. Therefore, the Kalman gain will increase and consequently re-linearization of the measurement model will effectively reduce the errors and uncertainty. Figure 3 illustrates the estimated path and features location by both extended Kalman filter and its iterated version in a 2D map of the environment.

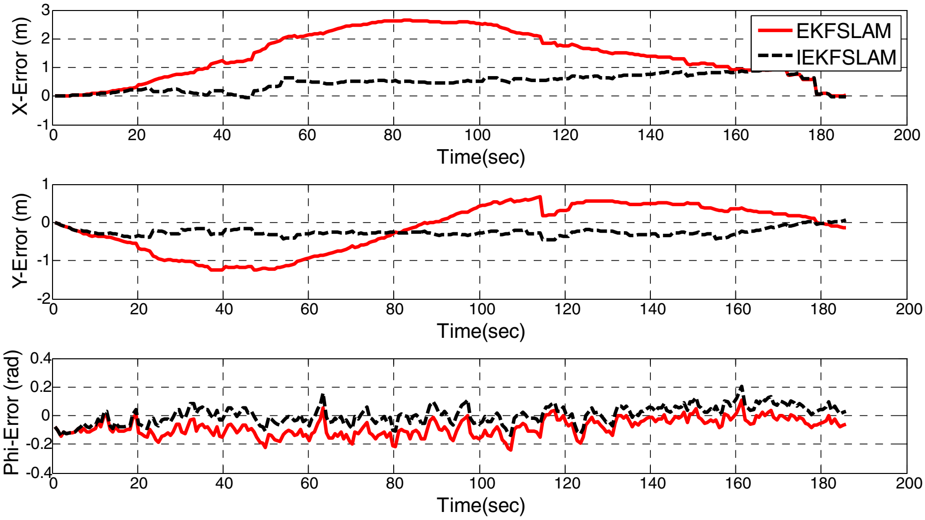


Fig. 4 Comparison of robot pose estimation errors for EKF (*solid line*) vs. IEKF (*dashed line*)

Table 1 Comparison of performance (estimation accuracy, consistency and computational complexity) of robot position and orientation for EKF vs. IEKF and SPKF vs. ISPKF in SLAM algorithm

Algorithm	Mean square error			Average	CPU elapsed time for
	x_R (m)	y_R (m)	φ_R (rad)	NEES (%)	update subroutine (for 50 detected landmarks) (sec.)
EKF-SLAM	2.7468	0.4121	0.0103	83.82	0.031
IEKF-SLAM	0.3012	0.0740	0.0039	91.6	0.187
SPKF-SLAM	0.39096	0.13471	0.00313	71.84	0.46
ISPKF-SLAM	0.03680	0.02821	0.00116	82	2.53

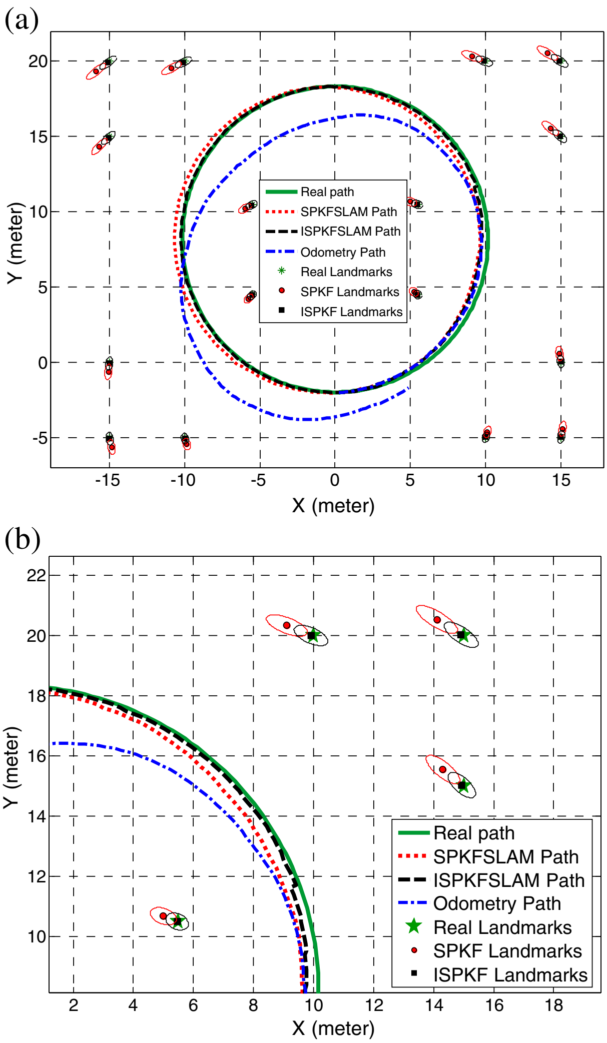
The estimated path by the IEKF approach is plotted with black dashed line. The estimated point features by EKF approach are shown by circles and the estimated features by IEKF approach are shown by squares. It is clear that the estimated map by the IEKF is more accurate than EKF. The robot pose errors for both filters are also plotted in Fig. 4 to show differences. The illustrated results are achieved for five iterations in IEKF-SLAM in our simulation. Table 1 also displays the estimation accuracy in terms of *mean squared error* (MSE) for both filters for the sake of quantitative comparison. It can be seen that the MSE of robot pose for IEKF is less than EKF that justifies better accuracy of state estimates by IEKF approach. This simulation is repeated several times with different initial conditions and all results consistently verified better performance of IEKF algorithm over the traditional EKF.

In the second simulation, the data are recorded for 250 seconds for both SPKF-SLAM and ISPKF-SLAM algorithms. Figure 5a illustrates the path and 2D map of the environment estimated by both sigma point Kalman filter and its iterated version. The estimated positions of features along with covariance ellipses are plotted by black squares for ISPKF approach. An enlarged portion of the map is also plotted in Fig. 5b to show the superior estimation accuracy of the ISPKF over the SPKF so clearly. The illustrated results are achieved for five iterations in our simulation.

Figure 6 shows the estimated robot pose errors. The estimation accuracy in ISPKF approach has been increased significantly because of statistical re-linearization of the measurement model. The performance of both filters is compared in terms of MSE in Table 1. Therefore, the superiority of the iterated versions of EKF and SPKF approaches are confirmed by single-run simulation tests. However, it should be noted that a single program run does not necessarily indicate that the filter is convergent or consistent [14]. Therefore, we compare the average NEES of each filter with its iterated version over 100 Monte Carlo runs to check the effect of iteration in consistency of SLAM algorithm. For the 3-dimensional robot pose, with $N = 100$, the *95% probability region* for $\bar{\varepsilon}$ is bounded by the acceptance interval [2.6, 3.4]. Each algorithm was tested for two loops of robot path that their results are shown in Fig. 7. Each filter tends to behave optimistic due to analytically and statistically linearized error propagation in extended and Sigma Point Kalman filters, respectively. When the initial robot uncertainty is set to zero, the average NEES of robot states lies below the upper bound of the acceptance interval most of the time for short-term experiments.

The results show that there is a little improvement in consistency of iterated filters due to re-linearization of measurement model. According to results shown in Fig. 7, each filter performs similarly to its iterated version and minor reduction in $\bar{\varepsilon}(k)$ for

Fig. 5 **a** The path and 2D map of the environment estimated by odometry, SPKF and ISPKF based SLAM. **b** An enlarged portion of the map to show differences clearly



IEKF and ISPKF are observed. However, as reported in the literature [16, 17], a level of initial uncertainty in the robot location can lead to an optimistic behavior of filters due to linearization errors and this situation may be worse for iterated filters and re-linearization enhances their divergence. For example, Fig. 8 shows the average NEES for both SPKF and ISPKF algorithms with non-zero initial uncertainty. The SPKF and ISPKF fail to maintain their consistency and are overoptimistic (i.e., filters covariances are too small) and this behavior leads to divergence of the filters.

The effect of iteration on computational cost is clearly negative. For the case of ISPKF-SLAM, when a new landmark is augmented to the rest of the map, 4 sigma points will be augmented to the sigma points matrix. The iteration mechanism in ISPKF will also intensify this complexity. For example, Fig. 9 shows elapsed CPU time for the execution of the estimation update subroutine in SPKF and ISPKF based

SLAM algorithms on a 2.8 GHz Pentium 4 for a 65 × 50 meters simulated outdoor environment with a feature density of 0.2 feature/m.

6.2 Experimental Results

The presented iterated Kalman filters based SLAM algorithms have been successfully implemented and evaluated using a differential drive mobile robot which is called NAJI in our laboratory. The implementation scenario is described in the sequel.

The NAJI robot is equipped with sonar and laser rangefinders for perception of its surroundings, wheel encoders and inertial measurement unit for prediction of its pose and an embedded IPC to store captured data through a WLAN in a log file. As is shown in Fig. 10, there are eight cylindrical landmarks around the environment. The robot travels along a rectangular-shaped path in an environment of 1.96 × 2.68 meters with a constant velocity of 10 cm/s. In order to simplify the experiments, an open-loop control strategy is considered to generate the path of the robot. The sampling frequency of odometry data from wheel encoders is about 8.34 Hz and a laser scanner sweeps about 240 degrees field of view every 30 odometry samples.

The cylindrical landmarks are selected as point-wise features to simplify feature extraction process. The IAV algorithm and geometric method which is used in simulation is also applied to extract range and bearing of landmarks and calculate measurement covariance matrix in this experiment. Similar to simulation results, NNSF is also used for data association in the SLAM algorithm. The ground truth is measured manually by placing a marker in the middle of two wheels of the robot so that it can plot the robot path during its motion. Then, some points are selected carefully for plotting the ground truth as a reference for filters performance assessment.

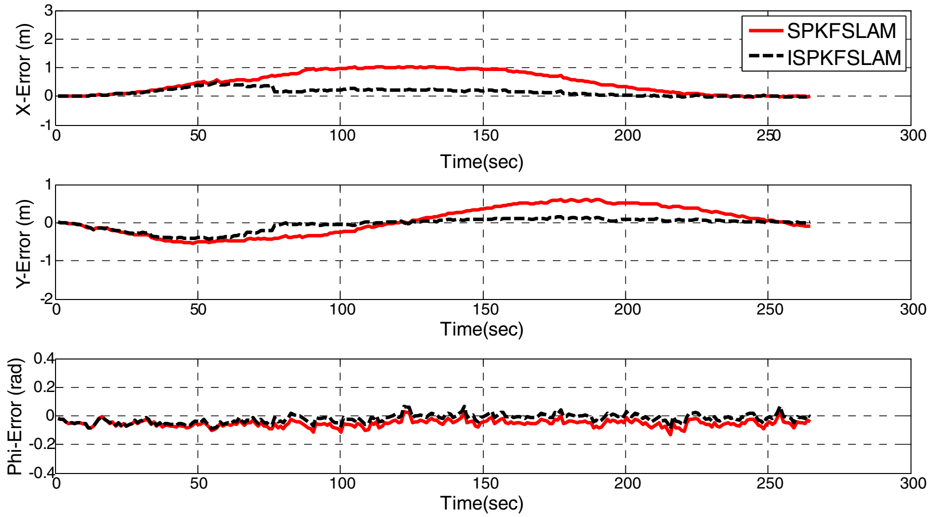


Fig. 6 Comparison of robot pose estimation errors for SPKF (*solid line*) vs. ISPKF (*dashed line*)

The robot starts from initial pose [27 cm, 50 cm, 0 rad.] and travels about 5 meters in a loop to return to its initial position. The Kalman Filter based SLAM algorithms and their modified iterated versions read stored log file and simultaneously estimate the robot pose and landmarks positions. The estimated robot pose errors are plotted in all directions for all algorithms in Figs. 11 and 12. Figures 11b and 12b illustrate the estimated robot pose and 2D map of the environment by EKF-SLAM, IEKF-SLAM and SPKF-SLAM, ISPKF-SLAM algorithms, respectively. The odometry

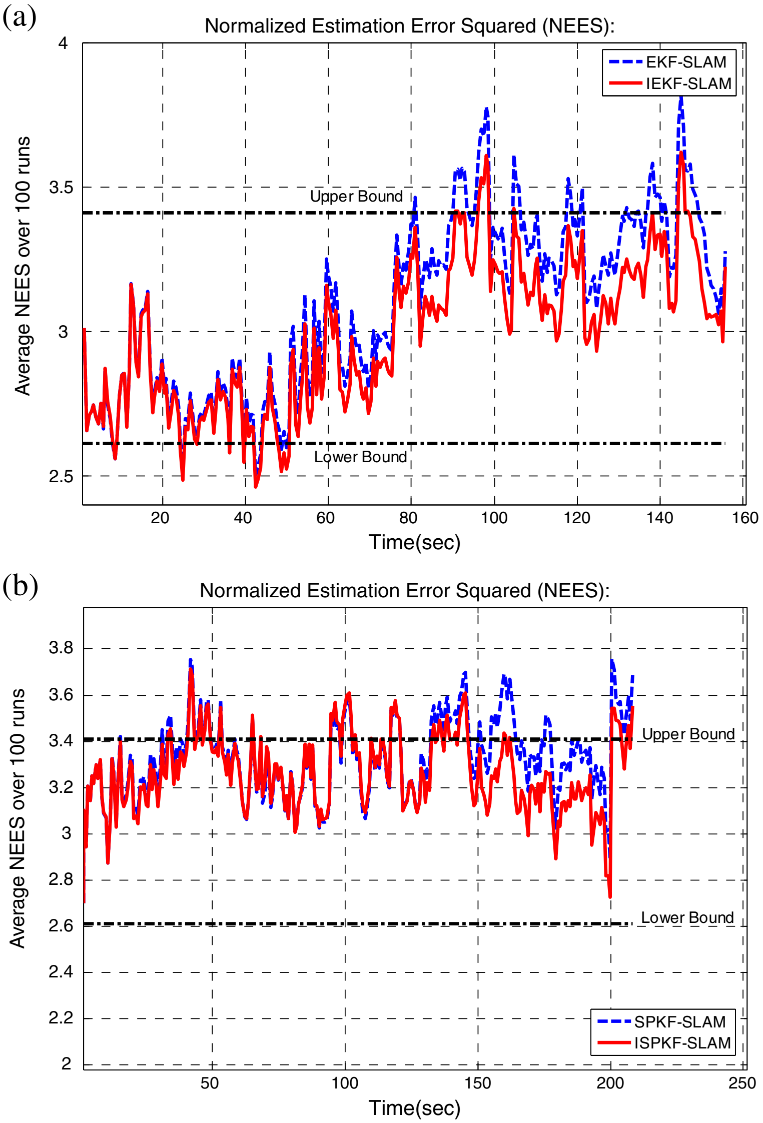


Fig. 7 Average NEES of robot pose states over 100 Monte Carlo runs with 95% probability region for **a** EKF-IEKF, **b** SPKF-ISPKF. The initial uncertainty is set to zero

Fig. 8 Average NEES of vehicle pose states over 100 Monte Carlo runs. The iteration speeds up divergence of the filter when initial uncertainty is set to non-zero

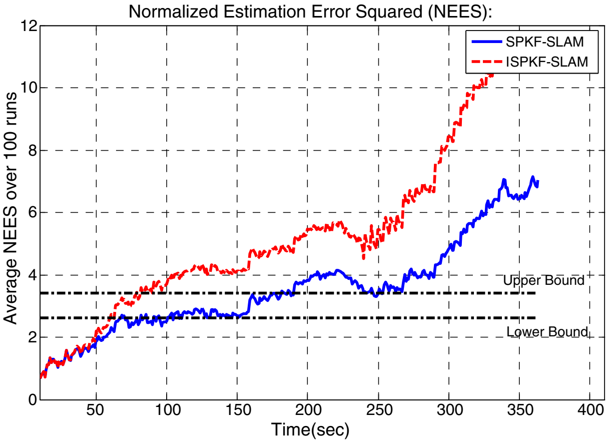


Fig. 9 Elapsed CPU time for estimation update in SPKF and ISPKF algorithms

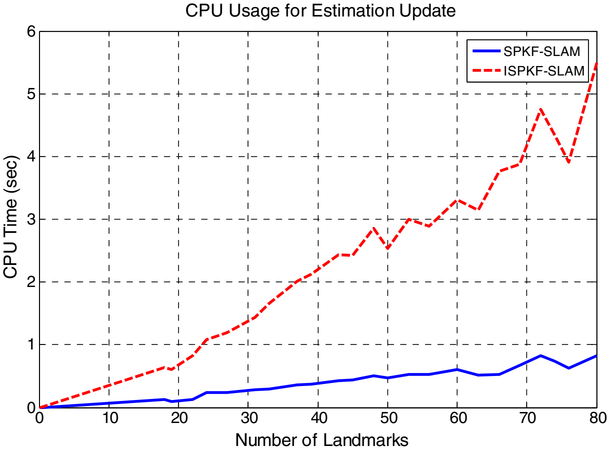
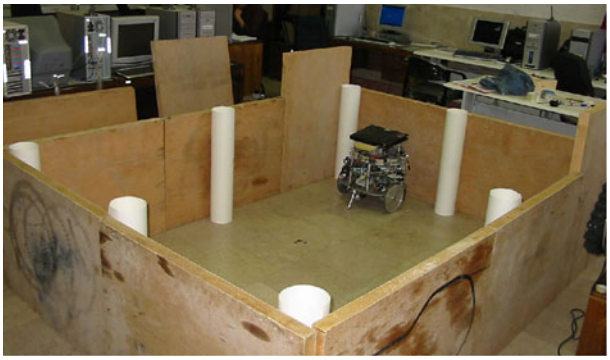


Fig. 10 Experimental setup with NAJI mobile robot: eight cylindrical landmarks are placed around wooden walls as point features



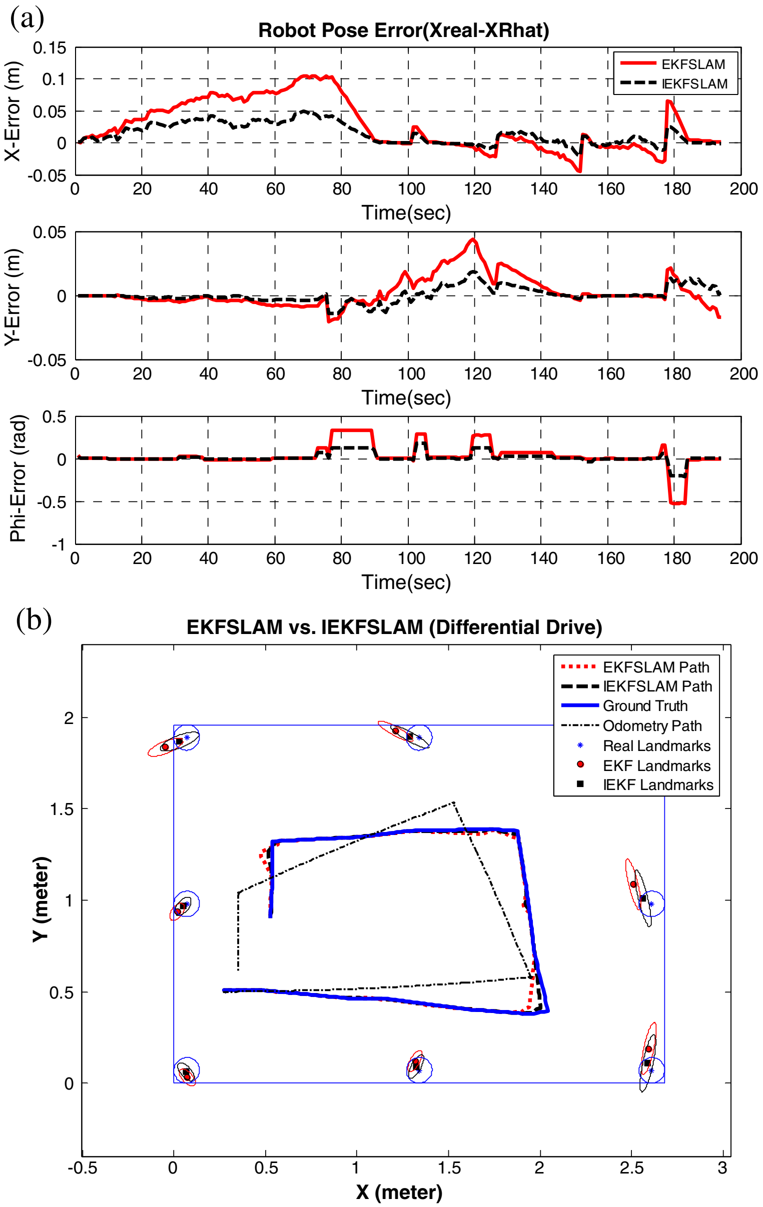


Fig. 11 **a** Robot pose estimation errors for comparison of EKF and IEKF based SLAM. **b** The path and 2D map of the environment estimated by odometry, EKF and IEKF based SLAM

path is shown by dashed line that has a great remarkable difference with ground truth and estimated trajectories due to accumulative errors. The real locations of centers of cylinders are shown by stars and the estimated landmarks positions are characterized by filled circles (for EKF-SLAM, SPKF-SLAM) and squares (for IEKF-SLAM,

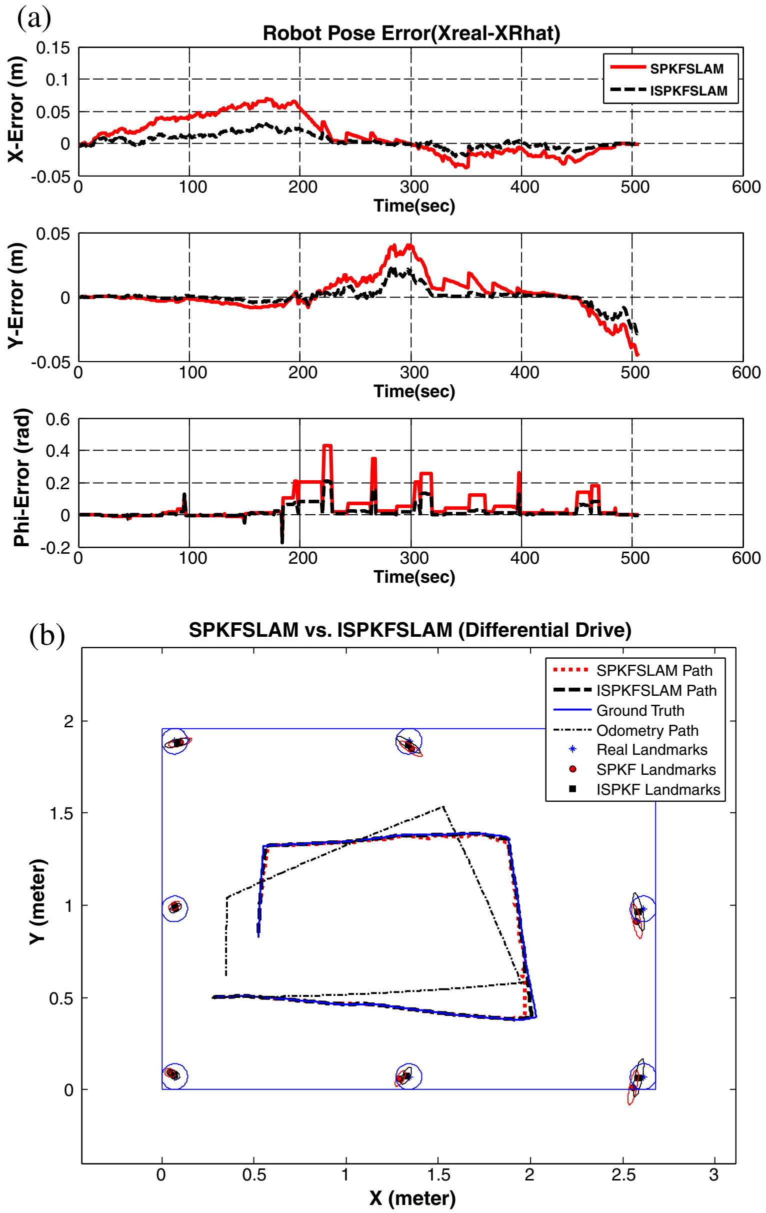


Fig. 12 **a** Robot pose estimation errors for comparison of SPKF and ISPKF based SLAM. **b** The path and 2D map of the environment estimated by odometry, SPKF and ISPKF based SLAM

ISPKF-SLAM). The covariance ellipses are plotted around estimated position (the PDF mean) in order to indicate estimated uncertainty. It is clear that re-linearization of the measurement model in the update step has reduced the estimated error dramatically and accuracy of estimation has greatly improved as shown in Figs. 11 and 12.

Roughly speaking, improvement of about 50% estimation accuracy is achieved in our single-run experiment in expense of increasing the computational complexity. It should be noted that intrinsic complexity of Kalman filter based SLAM is $O(n^2)$, where n is the number of landmarks in the environment, while the iteration also increases this complexity about 5.5 times. This value is achieved by the measurement of the CPU elapsed time for the update subroutine for five iterations in this experiment. By regarding rapid development in high speed hardware design, iterated versions of Kalman filters based SLAM may result in better solution to SLAM problem.

7 Conclusion and Discussion

In this paper, the effects of iteration in EKF and SPKF frameworks are experimentally investigated to improve the estimation accuracy in SLAM algorithm. All single-run simulation results consistently verified better performance and more accurate states estimation of the ISPKF and IEKF approaches over the non-iterated filters. However, the re-linearization of the measurement model in non-linear filters shows minor improvement on consistency and convergence for short-term experiments over 100 Monte Carlo runs. We saw that in the presence of large initial uncertainty in robot location, the iteration may lead to large estimation errors and consequently divergence of the IEKF-SLAM and ISPKF-SLAM algorithms. The computational complexity of these algorithms is clearly increased due to re-linearization. However, a tradeoff between estimation accuracy and computational cost should be made in the iteration mechanism.

Finally, single-run experiments were carried out for IEKF-SLAM and ISPKF-SLAM by NAJI mobile platform in a structured environment. Experimental results showed that iteration can improve the estimation accuracy in expense of high computational cost. Since the iterated Kalman filters only re-linearize the observation model, results of this work can similarly be achieved by executing the presented algorithms on other types of mobile robots other than NAJI. In addition, arbitrary trajectories other than circles and straight lines can easily be applied for such experiments by including a closed-loop trajectory tracking controller to the navigation algorithm. As mentioned in the work of Zhou et al. [21], one drawback of the iterated Kalman filters is that they will not make sense about the nonlinearity of the observation function in the re-linearization region. Thus, future works will be focused on further analysis of iterated Kalman filters in order to improve re-linearization mechanism.

Acknowledgements The authors would like to thank Rudolph van der Merwe from the OGI School of Science & Engineering at OHSU for the ReBeL toolkit, at http://choosh.cse.ogi.edu/rebel/. The financial support of Iran Telecommunication Research Center (ITRC) for this work is gratefully acknowledged. Islamic Azad University of Qazvin is also acknowledged for technical support.

Appendix: Feature Extraction Algorithm

In this section, a simple geometric method for feature extraction which is applied to our simulation is explained. After separating arc segments from LRF data by IAV method, the range and bearing (ρ, α) of cylindrical-shaped landmarks (such as the trunk of trees) in robot coordinate frame can be extracted from each arc cluster by the following equations. By considering a landmark in the field of view of the LRF (see Fig. 13), one may write

$$Sin\,\beta = \frac{r_C}{r_C + L} \quad , \quad L = \rho - r_C. \tag{38}$$

Thus, we have

$$r_C = \frac{L\,Sin\,\beta}{1 - Sin\,\beta} \quad , \quad \beta = \frac{\theta_2 - \theta_1}{2}, \tag{39}$$

$$\rho = L + r_C = \frac{L}{1 - Sin\,\beta} \quad , \quad \alpha = \frac{\theta_1 + \theta_2}{2}, \tag{40}$$

where r_C is approximated radius of the landmark. Angles θ_1 and θ_2 are the bearing measurement of the first and last point of each arc cluster. L is the distance of the nearest point to the robot. The covariance matrix may also be computed as:

$$\begin{bmatrix} \rho \\ \alpha \end{bmatrix} = Obs\left(\begin{bmatrix} L \\ \theta_1 \\ \theta_2 \end{bmatrix}\right), \tag{41}$$

$$R = \nabla Obs.P_i.\nabla Obs^T, \tag{42}$$

where $P_i = diag(\sigma_r^2, \sigma_\theta^2)$ is the covariance matrix of each measurement point in LRF data and ∇Obs is Jacobian matrix of the function Obs relative to parameters (L, θ_1, θ_2).

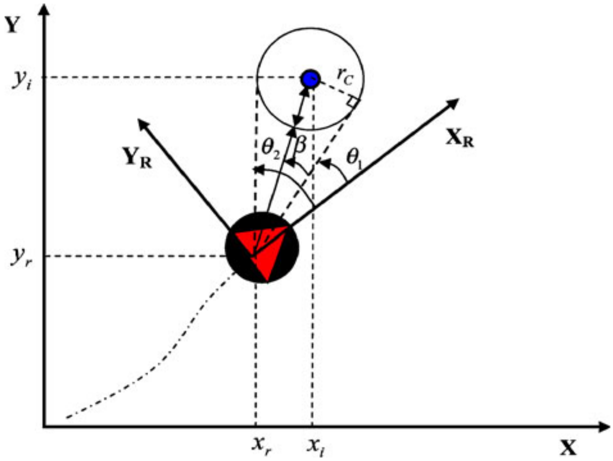


Fig. 13 Extraction of range and bearing of cylindrical landmarks from LRF data in robot coordinate frame by using three points in each cluster

References

1. Dissanayake, M.W.M.G., Newman, P., Clark, S., Durrant-Whyte, H.F., Csorba, M.: Multaneous localization and map building (SLAM) problem. IEEE Trans. Robot. Autom. **17**(3), 229–241 (2001)
2. Sibley, G., Sukhatme, G., Matthies, L.: The iterated sigma point Kalman filter with applications to long range stereo. In: Proceeding of Robotics: Science and Systems. Philadelphia, USA (2006)
3. Andrade-Cetto, J., Vidal-Calleja, T., Sanfeliu, A.: Unscented transformation of vehicle states in SLAM. In: Proc. of the 2005 IEEE Int. Conf. on Robotics and Automation, pp. 324–329. Barcelona, Spain (2005)
4. Julier, S.J., Uhlmann, J.K.: A new extension of the Kalman filter to nonlinear systems. In: The Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management II. SPIE (1997)
5. van der Merwe, R.: Sigma-point Kalman filters for probabilistic inference in dynamic state-space models. PhD thesis, OGI School of Science & Engineering, Oregon Health & Science University (2004)
6. Wan, E.A., van der Merwe, R.: The unscented Kalman filter for nonlinear estimation. In: Symposium 2000 on Adaptive Systems for Signal Processing (2000)
7. Martinez-Cantin, R., Castellanos, J.A.: Unscented SLAM for large-scale outdoor environments. In: Proc. of the 2005 IEEE Int. Conf. on Intelligent Robots and Systems, pp. 3427–3432 (2005)
8. van der Merwe, R., Wan, E., Julier, S.J.: Sigma-point Kalman filters for nonlinear estimation and sensor-fusion: applications to integrated navigation. In: Proceedings of the AIAA Guidance, Navigation & Control Conference. Providence, RI (2004)
9. Julier, S., Uhlmann, J.K.: A counter example to the theory of simultaneous localization and map building. In: Proc. 2000 IEEE Int. Conf. on Robotics and Automation, pp. 4238–4243. Seoul, Korea (2001)
10. Negenborn, R.: Robot localization and Kalman filters on finding your position in a noisy world. M.S thesis, Utrecht University (2003)
11. Smith, R., Self, M., Cheeseman, P.: Estimating uncertain spatial relationships in robotics. In: Autonomous Robot Vehicles. Springer (1990)
12. Bailey, T.: Mobile robot localization and mapping in extensive outdoor environments. PhD thesis, University of Sydney, Australian Centre for Field Robotics (2002)
13. Frese, U.: An O(log n) algorithm for simultaneous localization and mapping of mobile robots in indoor environments. Ph.D. thesis, University of Erlangen-Nurnberg (2004)
14. Bar-Shalom, Y., Li, X.-R., Kirubarajan, T.: Estimation with Applications to Tracking and Navigation. John Wiley and Sons Inc. (2001)
15. Williams, S.B.: Efficient solution to autonomous mapping and navigation problems. PhD thesis, University of Sydney, Australian Centre for Field Robotics (2001)
16. Bailey, T., Nieto, J., Guivant, J., Stevens, M., Nebot, E.: Consistency of the EKF-SLAM algorithm. In: IEEE/RSJ International Conference on Robotics and Automation (2006)
17. Castellanos, J.A., Neira, J., Tardos, J.D.: Limits to the consistency of EKF-based SLAM. In: IFAC Symposium on Intelligent Autonomous Veshicles (2004)
18. Xavier, J., Pacheco, M., Castro, D., Ruano, A., Nunes, U.: Fast line arc/circle and leg detection from laser scan data in a player driver. In: IEEE International Conference on Robotics and Automation. Barcelona (2005)
19. Durrant-Whyte, H., Bailey, T.: Simultaneous localization and mapping: part I. IEEE Robot. Autom. Mag. **13**(2), 99–110 (2006)
20. Shojaei, Kh., Shahri, A.M.: Iterated unscented SLAM algorithm for navigation of an autonomous mobile robot. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, 22–26 September, Nice, France (2008)
21. Zhou, W., Zhao, Ch., Guo, J.: The study of improving Kalman filters family for nonlinear SLAM. J. Intell. Robot. Syst. **56**(5), 543–564 (2009)
22. Guivant, J.: Efficient simultaneous localization and mapping in large environments. PhD thesis, University of Sydney, Australian Centre for Field Robotics (2002)
23. Arras, K.O.: Feature-based robot navigation in known and unknown environments. PhD thesis, EPFL, Lausanne (2003)