Article

TRANSACTIONS OF THE INSTITUTE OF MEASUREMENT & CONTROL

Transactions of the Institute of
Measurement and Control
2015, Vol. 37(3) 329–344
© The Author(s) 2014
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0142331214538899
tim.sagepub.com
SAGE

Optimizing the efficiency of the sub-map technique for large-scale simultaneous localization and mapping

Okechukwu Clifford Ihemadu, Wasif Naeem and Stuart Ferguson

Abstract

A technique for optimizing the efficiency of the sub-map method for large-scale simultaneous localization and mapping (SLAM) is proposed. It optimizes the benefits of the sub-map technique to improve the accuracy and consistency of an extended Kalman filter (EKF)-based SLAM. Error models were developed and engaged to investigate some of the outstanding issues in employing the sub-map technique in SLAM. Such issues include the size (distance) of an optimal sub-map, the acceptable error effect caused by the process noise covariance on the predictions and estimations made within a sub-map, when to terminate an existing sub-map and start a new one and the magnitude of the process noise covariance that could produce such an effect. Numerical results obtained from the study and an error-correcting process were engaged to optimize the accuracy and convergence of the Invariant Information Local Sub-map Filter previously proposed. Applying this technique to the EKF-based SLAM algorithm (a) reduces the computational burden of maintaining the global map estimates and (b) simplifies transformation complexities and data association ambiguities usually experienced in fusing sub-maps together. A Monte Carlo analysis of the system is presented as a means of demonstrating the consistency and efficacy of the proposed technique.

Keywords

Extended Kalman filter, simultaneous localization and mapping, sub-map technique, optimization, invariant information, error-correction process, data association

Introduction

The accuracy of localization is a key prerequisite for autonomous mobile robot navigation in an unknown environment. This is important such as for when an autonomous agent is required to enter any partially or completely unknown environment, explore it and produce a metric map of the place, while maintaining an accurate estimation of its own trajectory concurrently (Liu et al., 2007). This problem is generally referred to as simultaneous localization and mapping (SLAM) or Concurrent Localization and Mapping (CLM) (Durrant-Whyte and Bailey, 2006; Pinis and Tards, 2008). If an a priori map were available to the robot, then localization would be a relatively easier task. Likewise, if the robot were to have a precise, externally referenced position estimate, then mapping would be rather straightforward. Put differently, for mapping, the robot requires a good estimate of its location; as for localization, it needs a consistent map of the features.

SLAM has been found helpful in several application domains. For instance, Cocaud and Kubota (2010) proposed a SLAM-based navigation scheme for pinpoint landing on small celestial body. Tomizawa et al. (2012) developed an intelligent robotic cart (senior car) that moves autonomously along a given course to help people transfer luggage and themselves, and an approach for visual SLAM and human tracking for a wheelchair robot was proposed by Wu et al. (2013).

In SLAM, the location of the vehicle is predicted using the odometer encoder data, which may be corrupted with noise, leading to inaccurate performance of the overall system. For instance, if the environment is large, the robot may be unable to recognize already visited sections of the area leading to the problem of *loop-closing*. This is because a robot relies only on its estimated positions, which are inaccurate due to accumulated position estimation errors (Seadan et al., 2007) and (Ihemadu et al., 2012). These errors could result from potential wheel slippage or tilt, undulating surfaces and their associated irregularities, inaccuracies of the actuators and other predominantly unsystematic effects on the vehicle as it moves within an environment. The pose estimation errors emanating from the process noise accumulates as the distance covered by the robot increases, resulting in inaccurate SLAM performance for a long distance. The SLAM problem here is considered to be large scale in terms of the distance/size of an environment, for instance, distance in multiples of 100 m. However, the technique can also be made dependent on the number of landmarks in an environment.

Research on the SLAM problem has attracted keen attention in the field of intelligent robots over the past two

School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast, UK

Corresponding author:
Wasif Naeem, School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast, BT7 INN, UK.
Email: w.naeem@ee.qub.ac.uk

decades. Meanwhile, the extended Kalman filter (EKF)-based method has become the most popular tool for SLAM work, and many researchers consider it to be the key pivot for autonomous navigation. It represents the vehicle pose and the location of a set of environment features in a joint state vector that is estimated under the assumption of white Gaussian noise, thereby providing a way to deal with the uncertainties associated with measurement and motion processes (Brooks and Bailey, 2008; Dudek and Jenkin, 2010; Pinis and Tards, 2008). Nevertheless, in a large environment the EKF continues to suffer a bottleneck problem due to linearization errors and the computational complexity required in updating the covariance matrix. For instance, in a stochastic mapping approach based on an EKF the update requires an $O(n^3)$ matrix inversions (Deans and Hebert, 2000), where n is the number of landmarks or environment states to be estimated.

In general, several techniques proposed to overcome the effect of process noise on SLAM performance can be grouped into three: (a) splitting the area into sub-maps as well as re-observing the sub-map more than once (Blanco et al., 2007; Castellanos et al., 2007; Huang et al., 2006; Paz and Neira, 2006); (b) an appearance-based approach that tries to avoid the use of odometer data in estimating the robot position (Davison and Murray, 2002; Koenig et al., 2008; Paz et al., 2008; Porta and Krose, 2006; Seadan et al., 2007); (c) engaging an adaptive strategy as a means of controlling the motion of the robot (Cho et al., 2002; Harter, 2005; Härter and Campos Velho, 2008; Zhang et al., 2012). It appears that none of these approaches has thoroughly investigated the manner in which the process noise affects predictions and estimations in SLAM with the view of addressing it.

Many state-of-the-art algorithms have been proposed to improve the accuracy of the map by engaging the sub-map techniques. For instance, Paz and Neira (2006) demonstrated that sub-map techniques detect loops correctly through data association and that the maps obtained were more consistent, with 50 times reduction in the computational cost. However, they observed that 'deciding the size of each local map is an important issue which none of the several sub-map based methods have been able to answer' (Paz and Neira, 2006). Furthermore, Huang et al. (2006) raised some other questions, such as when to start a new local map and how to know that the robot has visited an already seen map.

In Paz and Neira (2006), the following questions were raised: (1) Is there an optimal local map size, such that the total computational cost is minimal? (2) Is there a consistent optimal local map size, such that the consistency of the final map is maximal? and (3) Do these optimal points coincide? They derived Equations (1) and (2) for answers to the questions:

$$P = Np - (N - 1)r \tag{1}$$

$$N = \frac{P - r}{p - r} \tag{2}$$

where P is the total number of features and N is the number of sub-maps created in a given environment, while p is defined as the size of the sub-map in terms of the number of features and r is the number of re-observed features. This derivation was based on a number of assumptions, such as known environment size and sensor characteristics; the features to be equally spaced out in all directions; the sensor must observe an equal number m of features that will include r re-observed ones, at each time observation taken. However, these assumptions may be very difficult to attend in real-life environments, thereby invalidating the definition of map size in terms of the number of features only.

Notwithstanding, the sub-map approach still appears to be promising, because it tackles both the linearity and computational complexity drawbacks of EKF-based SLAM simultaneously. However, there are still a number of issues to be addressed, such as the optimal size of a sub-map in terms of distance; the optimal accumulated pose estimation error value that could be accommodated within the sub-map, without infringing on the accuracy and consistency of the map; and the determination of process noise covariance matrix that will generate the acceptable error effect within an optimal sub-map size.

This study investigates the effect of the process noise covariance on both the predicted and the estimated robot positions as the distance increases, with the view to addressing the above issues. The terms predicted and estimated are used in this study to refer to the sensor data and EKF estimation results, respectively. To the best of the authors' knowledge this study is the first of its kind in that it carries out comprehensive analysis of the effect the process noise covariance produces on the SLAM results. The solutions to these questions were used as criteria for ending and initializing sub-maps, thereby demonstrating substantial improvements in SLAM. They can also be engaged for adaptive control of SLAM activities in any environment. The optimal error effect also served as a guide for effective error compensation, leading to an improvement on the accuracy of SLAM results.

The technique begins by proposing an error effect model that will output the error values for both predicted and estimated position of the robot for a given non-linear robot model. Then, Monte Carlo analysis was employed to simulate the error effect resulting from applying a range of randomly generated process noise values to the error model. From the results obtained, one can give a quantitative definition of the relationship between a given process noise value and the error effect it has on the robot pose. The optimal error effect was defined with reference to the distance covered, while the distance where the optimal error was obtained will serve as the optimal distance size for a sub-map. The value of the process noise that yields the optimal error effect could then serve as a reference process noise in designing a process model that will produce a more accurate and efficient SLAM result.

The algorithm was used to optimize the preliminary work on the Invariant Information Local Sub-map Filter (IILSF) (Ihemadu et al., 2013). The IILSF technique, explained in Section 3, relieves the computational burden of updating the covariance matrix and reduces the pose estimation error due to linearization of non-linear models used in EKF SLAM. In addition, it minimizes the storage requirements and the cost of fusing sub-maps to form a single large map. Furthermore, the consistency and the convergence of the map were improved by resolving the usually prevalent data association ambiguities through the use of invariant information as a frame of reference (FoR). However, the size of the sub-maps was heuristically fixed and the issue of pose estimation error caused by the process noise was not addressed (Ihemadu et al., 2013).

The above optimization technique makes use of the numerical results obtained from the investigations to improve the accuracy and efficiency of the IILSF method. An observation routine was developed to monitor the amount of errors that are accumulated within the existing sub-map, while a new sub-map was initialized based on the optimal sub-map size already obtained through Monte Carlo simulations. The accuracy of the SLAM result was also improved by ensuring that none of the errors accumulated in any sub-map was allowed to propagate into the succeeding sub-map. This was achieved through the added advantage of the IILSF approach, which gives the robot the leverage of starting a new origin at any convenient location.

Section 2 describes the models used in this paper, while Section 3 highlights some key features of the IILSF algorithm. Section 4 presents a simulation, where it was demonstrated that the sub-map technique alone does not remove the effect of process noise on both the predictions and estimations of robot pose. The optimal error investigation procedure is explained in Section 5, followed by Monte Carlo simulation in Section 5.1. The error-correcting process is presented in Section 6. The results obtained from the simulations are presented in Section 7 with a brief discussion on them. Finally, the results from simulations using the optimized IILSF are presented as an example of the complete SLAM process in Section 8, while the paper ends with concluding remarks in Section 9.

System models

The various models engaged in this study are presented in the following sections.

Robot model

The robot used in this study was assumed to be a four-wheeled car-like vehicle moving with a constant velocity, and equipped with a laser sensor. Figure 1 shows the kinematic model of the robot, while Equation (3) shows the kinematic equation of the robot:

$$\begin{bmatrix} \dot{x}_v \\ \dot{y}_v \\ \dot{\theta}_v \end{bmatrix} = \begin{bmatrix} V\cos(\theta_v) \\ V\sin(\theta_v) \\ V/L\tan(\phi_v) \end{bmatrix} \tag{3}$$

where V and v denote the velocity and vehicle, respectively, L the distance between the front and back axles, and x_v, y_v and θ_v represent the position and the heading of the vehicle. This kinematic equation could be parametrized in the form shown in Equation (4):

$$\mathbf{X}(k+1) = f(\mathbf{X}(k), \mathbf{U}(k), \mathbf{W}(k)) \tag{4}$$

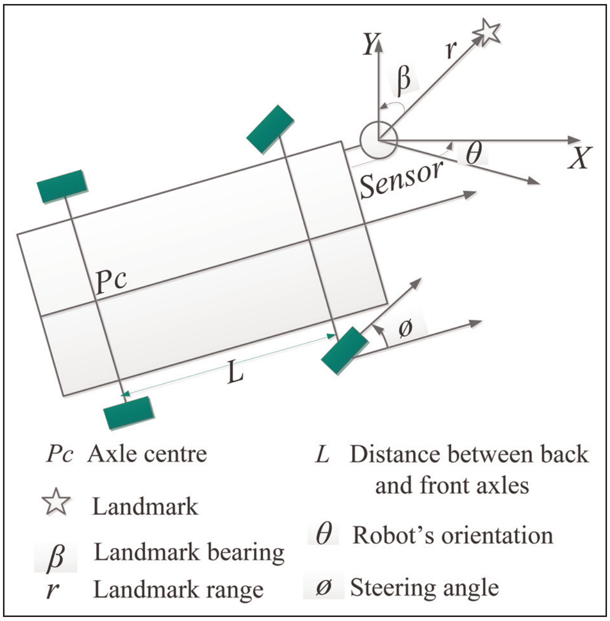

Figure 1. Kinematic of robot model.

Process and observation models

The process model, $f(\mathbf{X}, \mathbf{U}, \mathbf{W})$, used to predict the position of the robot while taking the observation at different locations is given by Equation (5):

$$\begin{bmatrix} x_v \\ y_v \\ \theta_v \end{bmatrix} = \begin{bmatrix} x_v(k) + V\Delta t\cos(\theta_v) \\ y_v(k) + V\Delta t\sin(\theta_v) \\ \theta_v(k) + V\Delta t/L\tan(\phi_v) \end{bmatrix} + \begin{bmatrix} w_x(k) \\ w_y(k) \\ w_\theta(k) \end{bmatrix} \tag{5}$$

where Δt is the time interval and w_x, w_y and w_θ denote the process noise on the state vector x_v, y_v, θ_v. Here the Jacobian of the process noise ($\mathbf{W}(k)$) is computed with Equation (6):

$$\mathbf{W}(k) = \frac{\partial f}{\partial \mathbf{W}} = \begin{bmatrix} \frac{\partial f_1}{\partial w_x} & \frac{\partial f_1}{\partial w_y} & \frac{\partial f_1}{\partial w_\theta} \\ \frac{\partial f_2}{\partial w_x} & \frac{\partial f_2}{\partial w_y} & \frac{\partial f_2}{\partial w_\theta} \\ \frac{\partial f_3}{\partial w_x} & \frac{\partial f_3}{\partial w_y} & \frac{\partial f_3}{\partial w_\theta} \end{bmatrix} \tag{6}$$

It should be noted that uncertainties in velocity, wheel odometer encoder and surface terrain all contribute to the magnitude of the process noise. Here, the Jacobian of the process noise is calculated using the holistic effect of both the inputs and a factor, c (representing the effect of the odometer and surface terrain). This achieves the objective of the research for the purpose of simulation demonstration.

The same process model without the process noise was used to predict the true position of the robot to serve as a reference.

A non-linear observation model, Equation (7), relates the observations with the current robot's pose:

$$\mathbf{Z}(k) = h(\mathbf{X}(k)) + \mathbf{V}(k) \tag{7}$$

which is expressed in discrete form as shown in Equation (8):

$$\mathbf{Z} = \begin{bmatrix} z_r \\ z_\theta \end{bmatrix} = \begin{bmatrix} \sqrt{(x_m - x_v)^2 + (y_m - y_v)^2} + v_r \\ \arctan\left(\frac{y_m - y_v}{x_m - x_v}\right) - \theta_v + v_\theta \end{bmatrix} \tag{8}$$

z_r, z_θ are the range and bearing of observed landmarks, while (x_m, y_m) represent the location of the landmarks and (x_v, y_v) the Cartesian coordinates of the vehicle position. The measurement noise from the range and bearing are represented by $v_r(k)$ and $v_\theta(k)$.

Process noise model

In modelling the wheeled robot, the term 'process noise' is used to account for such noises resulting from potential wheel slippage, surface irregularities, inaccuracies of the actuators and other predominantly unsystematic effects that could not be explicitly modelled. These environmental properties translate into errors affecting the estimation of the robot pose.

The process noise covariance **Q** is modelled as a 3 × 3 matrix representing process noise, which is assumed to be Gaussian. It is considered to be proportional to the effect of the control inputs on $(\Delta x, \Delta y$ and $\Delta\theta)$. So it is usually calculated by multiplying some Gaussian sample **C** with the Jacobian of the process noise, **W**, and $\mathbf{W^T}$, that is:

$$\mathbf{Q} = \mathbf{WCW^T}$$

$$\mathbf{W} = diag([\Delta t \cos\theta, \Delta t \sin\theta, \Delta\theta])$$

$$\mathbf{C} = \begin{bmatrix} c\Delta x_k^2 & 0 & 0 \\ 0 & c\Delta y_k^2 & 0 \\ 0 & 0 & c\Delta\theta_k^2 \end{bmatrix}$$

where $(\Delta x_k, \Delta y_k, \Delta\theta_k)$ are the change in positions due to the applied control inputs on the robot at time step k and **C** represents the effect of odometer noise (process noise). The effect of **C** on each of the coordinates (x_v, y_v, θ_v) is assumed to be equal, and is represented here by c.

Meanwhile, the key features of the preliminary work on the IILSF are presented in the next section to keep the reader abreast of the full technique.

Invariant Information Local Sub-map Filter

Figure 2 illustrates the architecture of the IILSF methodology (Ihemadu et al., 2013). Each block describes the processes involved in the algorithm. It operates a normal SLAM process without any restriction in terms of number of features or the distance; however, if the environment is relatively small and the features very scant, then there may not be any need for creating sub-maps. The following sections detail the functioning of the IILSF according to Figure 2.

Basic SLAM process

The basic SLAM process is performed in the EKF estimation model. Here exploration is achieved through a four-wheeled car-type vehicle equipped with a scanning laser sensor mounted at the front. A process model, given by Equation (5), was used to predict the position of the robot while taking observation at different time steps. In addition, an observation model, shown in Equation (8), extracts the range and bearing of observed landmarks relative to the pose (x_v, y_v, θ_v) of the robot as it moves.

In Figure 2, navigation for each sub-map is initialized with $\mathbf{X_0}$, $\mathbf{P_0}$, $\mathbf{U_0}$ and $\mathbf{Z_0}$, while standard EKF prediction Equations, (9) and (10), were employed for the estimations of the state vector, $\hat{\mathbf{X}}_L$, and the associated covariance matrix, $\mathbf{P}_L$, as the exploration progresses:

$$\hat{\mathbf{X}}^-(k+1) = f(\mathbf{X}(k), \mathbf{U}(k)) \tag{9}$$

$$\mathbf{P}^-{}_{(k+1)} = \nabla f_k \mathbf{P}_k \nabla f_k^T + \mathbf{W}_k \mathbf{Q}_k \mathbf{W}_k^T \tag{10}$$

In addition, a non-linear observation model, Equation (11), was employed for exploration at the sub-map level:

$$\hat{\mathbf{Z}}_L^-(k) = h(\hat{\mathbf{X}}_L + (k), + \mathbf{V}_L(k) \tag{11}$$

where $\hat{\mathbf{Z}}_L^-(k)$ is the observation made at time k with measurement noise $\mathbf{V}_L(k)$. Normal data association techniques such as the *nearest-neighbour* enable reliable validation of observed data.

The estimated state vector, $\hat{\mathbf{X}}_L$, with its associated covariance matrix, $\mathbf{P}_L$, shown as Equations (12) and (13), are obtained through EKF update equations and passed into the local map decoupling model sequentially:

$$\hat{\mathbf{X}}_L = [\hat{x}_v, \hat{y}_v, \hat{\theta}_v, \hat{x}_m(1), \hat{y}_m(1), \cdots, \hat{x}_m(n), \hat{y}_m(n)]T \tag{12}$$

$$\mathbf{P}_L = \begin{bmatrix} \mathbf{P}_{v_L} & P_{vm_L} \\ \mathbf{P}_{vm_L}^T & P_{mm_L} \end{bmatrix} \tag{13}$$

where $\hat{\mathbf{X}}_{L(1:3)} = [\hat{x}_v, \hat{y}_v, \hat{\theta}_v]^T$ are the estimated Cartesian coordinates and orientation angle of the robot, $\hat{\mathbf{X}}_{L(4:n)} = [\hat{x}_m(1), \hat{y}_m(1), \cdots, \hat{x}_m(n), \hat{y}_m(n)]^T$ are the coordinates of n observed landmarks and the subscripts are as follows: L represents local, v robot-to-robot and mm landmark-to-landmark.

Creation of new local sub-map

New local maps are created within the local map decoupling model, where an observer compares the incoming $\hat{\mathbf{X}}_L$ with the preceding ones, to detect when the criteria for creating new local maps are satisfied. A major constraint for creating a new sub-map is to have at least two features in common with the previous sub-area and the proposed new area that will serve as reference base. The pre-set conditions could be based on the number of features already observed or the distance along the x-axis, d, covered by the robot.

Only when these requirements are met does the algorithm initialize a new local map exploration with a new origin that is defined without any uncertainty, while $\hat{\mathbf{X}}_\mathbf{L}$ and $\mathbf{P_L}$ will be decoupled and transferred.

The transfer of information from the local map decoupling model is represented by the switches in Figure 2. By this

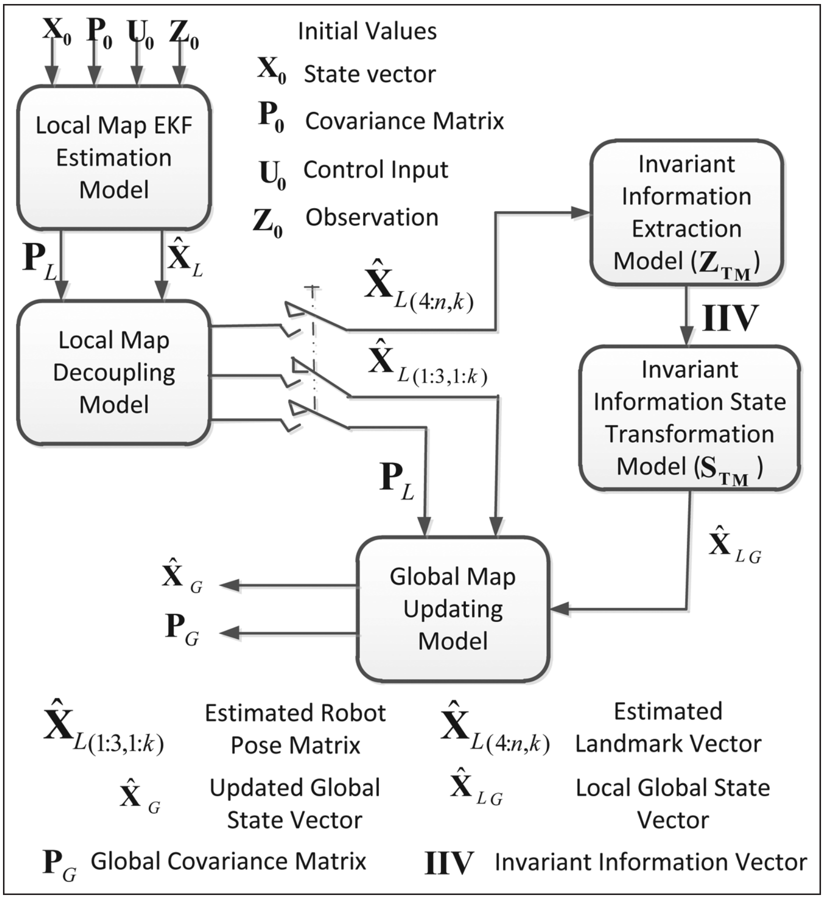
$\mathbf{X}_0$ $\mathbf{P}_0$ $\mathbf{U}_0$ $\mathbf{Z}_0$
Initial Values
$\mathbf{X}_0$ State vector
$\mathbf{P}_0$ Covariance Matrix
$\mathbf{U}_0$ Control Input
$\mathbf{Z}_0$ Observation
Local Map EKF Estimation Model
$\mathbf{P}_L$
$\hat{\mathbf{X}}_L$
Local Map Decoupling Model
$\hat{\mathbf{X}}_{L(4:n,k)}$
$\hat{\mathbf{X}}_{L(1:3,1:k)}$
$\mathbf{P}_L$
Invariant Information Extraction Model ($\mathbf{Z}_{\mathbf{TM}}$)
IIV
Invariant Information State Transformation Model ($\mathbf{S}_{\mathbf{TM}}$)
$\hat{\mathbf{X}}_{LG}$
Global Map Updating Model
$\hat{\mathbf{X}}_G$
$\mathbf{P}_G$
$\hat{\mathbf{X}}_{L(1:3,1:k)}$ Estimated Robot Pose Matrix
$\hat{\mathbf{X}}_{L(4:n,k)}$ Estimated Landmark Vector
$\hat{\mathbf{X}}_G$ Updated Global State Vector
$\hat{\mathbf{X}}_{LG}$ Local Global State Vector
$\mathbf{P}_G$ Global Covariance Matrix
IIV Invariant Information Vector

Figure 2. Invariant information local map filter architecture.

means, $\hat{\mathbf{X}}_{L(4:n,k)}$ will be transferred into the invariant information extraction model, ($\mathbf{Z}_{TM}$), while $\hat{\mathbf{X}}_{L(1:3,1:k)}$, and the associated $\mathbf{P}_L$, will be fed directly into the global map updating model, k and n being the number of steps when observations were taken, and number of landmarks seen, respectively.

The next section explains the technology used in extracting and transforming invariant information from the observed landmarks state vector $\hat{\mathbf{X}}_{4:n,k}$ of a local area

Extraction and transformation of invariant relative information from local maps

When a new local map has been initialized, the $\hat{\mathbf{X}}_{L(4:n,k)}$, which includes the local FoR for the previous map, is received in the invariant information extraction model ($\mathbf{Z}_{TM}$) for extraction of the local invariant information vector relative to that FoR. This is followed by the transformation of the extracted invariant information vector into the global coordinate frame.

Invariant information extraction model ($\mathbf{Z}_{TM}$). Invariant information relative to the FoR will be extracted from $\hat{\mathbf{X}}_{4:n,k}$ in $\mathbf{Z}_{TM}$ to produce the invariant information vector (**IIV**). For instance, Figure 3 illustrates an environment consisting of seven landmarks where two sub-areas (sub-map A and sub-map B) were created. In the environment, the relative distances and bearings between the landmark f_1 and landmarks (f_2,f_3,f_4,f_5,f_6,f_7), depicted by $r_{0,\cdots n}$ and $\beta_{0\cdots n,12}$, constitute the invariant information from the area. The robot started initially from the global origin ($\boldsymbol{O_G}$) in sub-map A, with observed landmarks f_1 and f_2,f_6,f_7. When the horizontal distance, d, reaches a specified value, a new sub-map B is created and a new origin $\boldsymbol{O_L}$ is defined. In B, the robot observed landmarks f_1,f_2,f_3,f_4 and f_5 so that f_1 and f_2 are seen in both sub-maps. In Figure 3, r_{01} and $\beta_{0,1}$ serve as the FoR, while r_{12} forms the reference base for B. The invariant information extraction model, $\mathbf{Z}_{TM}$ in Equation (14), is engaged to extract **IIV** (Equation (15)), where $(\hat{x}_0, \hat{y}_0)$ is the origin of the global map, $\hat{x}_1,\hat{y}_1$ and $\hat{x}_2,\hat{y}_2$ are the coordinates of the base, $\overline{f_1f_2}$, and $\hat{x}_{3\cdots n},\hat{y}_{3\cdots n}$ are the coordinates of observed landmarks:

$$\mathbf{Z}_{TM} = \begin{bmatrix} \sqrt{(\hat{x}_1-\hat{x}_0)^2+(\hat{y}_1-\hat{y}_0)^2} \\ atan2(\hat{y}_1-\hat{y}_0,\hat{x}_1-\hat{x}_0) \\ \sqrt{(\hat{x}_2-\hat{x}_1)^2+(\hat{y}_2-\hat{y}_1)^2} \\ atan2(\hat{y}_2-\hat{y}_1,\hat{x}_2-\hat{x}_1) \\ \sqrt{(\hat{x}_3-\hat{x}_1)^2+(\hat{y}_3-\hat{y}_1)^2} \\ atan2(\hat{y}_3-\hat{y}_1,\hat{x}_3-\hat{x}_1) \\ \cdots \\ \cdots \\ \cdots \\ \sqrt{(\hat{x}_n-\hat{x}_1)^2+(\hat{y}_n-\hat{y}_1)^2} \\ atan2(\hat{y}_n-\hat{y}_1,\hat{x}_n-\hat{x}_1) \end{bmatrix} = \begin{bmatrix} r_{01} \\ \beta_{01} \\ r_{12} \\ \beta_{0,12} \\ r_{13} \\ \beta_{3,12} \\ \cdots \\ \cdots \\ \cdots \\ r_{1n} \\ \beta_{n,12} \end{bmatrix} \tag{14}$$

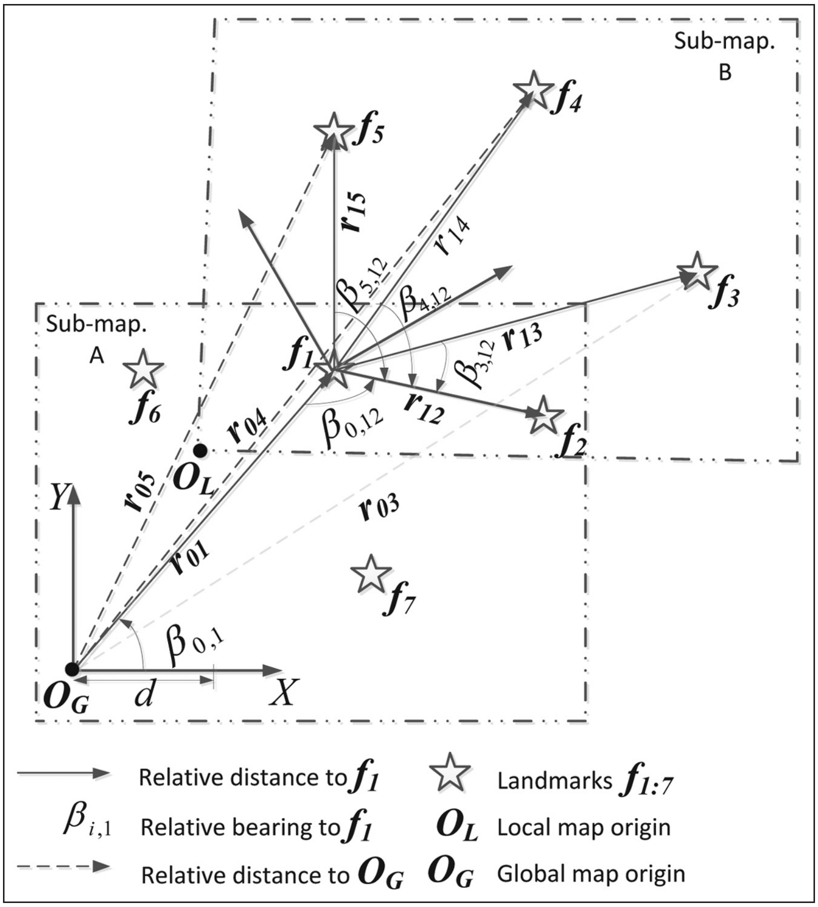

Figure 3. Local to global map transformation.

$$\mathbf{IIV} = [r_{01}, \beta_{01}, r_{12}, \beta_{0,12}, r_{13}, \beta_{3,12}, \cdots r_{1n}, \beta_{n,12}]^T \tag{15}$$

The **IIV** obtained from a local map is in polar form, where the subscript 12 represents the reference base subscripts and $i \cdots n$ depict the index of the landmarks. Thus, r_{1i} is the distance between f_1 and f_i, while $\beta_{i,12}$ is the angle that f_i makes with the base.

Invariant information state transformation model ($\mathbf{S}_{TM}$). Transformation of the **IIV** into a local map state vector in a global frame, $\hat{\mathbf{X}}_{LG}$, is performed in two stages in the $\mathbf{S}_{TM}$ model. The first stage converts the polar **IIV** into Cartesian coordinates using $\hat{x}_i = r_{1,i}\cos(\beta_{i,12})$ and $\hat{y}_i = r_{1,i}\sin(\beta_{i,12})$ to obtain Equation (16):

$$\hat{\mathbf{X}}_{TM} = [\hat{x}_1, \hat{y}_1, \hat{x}_2, \hat{y}_2, \hat{x}_i, \hat{y}_i, \cdots, \hat{x}_n, \hat{y}_n]^T \tag{16}$$

The next stage begins by computing the Jacobian of the transformation model, which grows with the size of $\hat{\mathbf{X}}_{TM}$ using Equation (17), where $\hat{\mathbf{X}}_{L(1:N)}$ are the local map estimations. Then $\hat{\mathbf{X}}_{TM}$ and $\mathbf{J}_{TM}$ are used to evaluate the $\hat{\mathbf{X}}_{LG}(N)$ of each local map with Equation (18):

$$\mathbf{J}_{TM} = \left[\frac{\partial \mathbf{IIV}}{\partial \hat{\mathbf{X}}_{L(1)}}, \cdots, \frac{\partial \mathbf{IIV}}{\partial \hat{\mathbf{X}}_{L(N)}}, 0, \cdots, 0\right] \tag{17}$$

$$\hat{\mathbf{X}}_{LG}(N) = \mathbf{J}_{TM}(N)\hat{\mathbf{X}}_{TM}(N) \tag{18}$$

where N is the index of the local maps. For example, consider that **IIV** obtained from $\hat{\mathbf{X}}_{L(2)}$ for sub-map B in Figure 3 contains $r_{01} = 120\,\text{m}, \beta_{0,1} = 50\,°$, $r_{13} = 70\,\text{m}, \beta_{3,12} = 30\,°$ and $r_{14} = 65\,\text{m}, \beta_{4,12} = 75\,°$, then the $\hat{\mathbf{X}}_{LG}(2)$ (to the nearest whole number) will be

$$\hat{\mathbf{X}}_{LG}(2) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{y}_1 \\ \hat{x}_3 \\ \hat{y}_3 \\ \hat{x}_4 \\ \hat{y}_4 \end{bmatrix} = \begin{bmatrix} 77 \\ 92 \\ 138 \\ 127 \\ 94 \\ 155 \end{bmatrix} \tag{19}$$

yielding the global coordinates of f_1, f_3 and f_4 to be (77,92), (138,127) and (94,155), respectively. Hence, transforming the local map with this model will relate all the invariant information to the global coordinate system through the FoRs.

Updating the global map

$\hat{\mathbf{X}}_{L(1:3,1:k)}$ and $\mathbf{P}_L$ will remain in the global map updating model until $\hat{\mathbf{X}}_{LG}$ from $\mathbf{S}_{TM}$ is received for performing global map update. The global map is updated by cascading $\hat{\mathbf{X}}_{LG}$ obtained from each local map into the corresponding global frame. Equation (20) appends current $\hat{\mathbf{X}}_{LG}$ to the preceding ones, while Equation (21) combines all the local covariance matrices ($\mathbf{P}_L$ s) in block diagonal form:

$$\hat{\mathbf{X}}_G = [\hat{\mathbf{X}}_{LG}(1), \hat{\mathbf{X}}_{LG}(2), \cdots, \hat{\mathbf{X}}_{LG}(N)]^T \qquad (20)$$

where G stands for global, and $1 \cdots N$ are the index of the local maps.

$$\mathbf{P}_G^+ = \begin{bmatrix} \mathbf{P}_{LG}^+(1) & 0 \\ 0 & \mathbf{P}_{LG}^+(N) \end{bmatrix} \qquad (21)$$

The global estimations with the associated covariance matrix are thus updated in the global map updating model. The trajectories ($\hat{\mathbf{X}}_{L(1:3,\,1:k)}$) already stored in the global map updating model can also be retrieved if required. The next section provides the simulation conditions and results when the IILSF technique was engaged as part of a SLAM exploration of a large environment.

Invariant Information Local Sub-map Filter simulations

IILSF simulation conditions

All the simulations in this paper have been carried out in Matlab 2012 on a 64-bit version of the Windows 7 platform running on a 3.00 GHz Intel Dual Core processor. The IILSF technique was employed as part of a basic SLAM using a simple non-linear robot model as described in Section 3.1 to explore a large unstructured environment with a number of landmarks. The robot was assumed to be moving with a constant velocity and small steering angle of 4 m/s and 0.002 radians, respectively. This is to enable it to move more along the x-axis for demonstration purposes, since the environment was rectangular with a length that was longer than the width.

The laser sensor was characterized with a range of 15 m scanning with an 180° span, while the measurement variance, σ_{v_r} and σ_{v_θ}, were estimated to be 0.05 m and 1°, respectively.

In this simulation, new sub-maps were created based on the observation of at least two landmarks in both the existing and the proposed new map, in addition to the distance, d, covered by the robot. The last position of the robot in the preceding local map served as the new origin, while the line joining the last but two and last but one landmarks observed in the previous sub-area, and also seen in the proposed one, formed the reference base. The system then builds sub-maps within the environment whilst simultaneously tracing its own trajectory, in addition to generating a global map of the environment. The IILSF results are compared with those obtained from the standard absolute map filter (AMF) in the next section.

IILSF simulations results

The global map generated by the IILSF is shown in Figure 4(a). It is evident that there are four sub-maps (***A***–***D***) that were created based on the distance d (heuristically chosen to be 100 m in this case). The plots show the predicted robot pose as 'robot pose model', while 'robot pose SLAM' represents the estimated robot's trajectory. Likewise, in the figure the simulated landmark locations are represented by 'landmark true', while the 'landmark-IILSF-SLAM' depict their estimated positions. This result demonstrates that the IILSF algorithm successfully fused all local maps into one large global map without additional constraints, thereby eliminating the cost of enforcing such constraints. For comparison, an AMF-generated global map of the same environment is

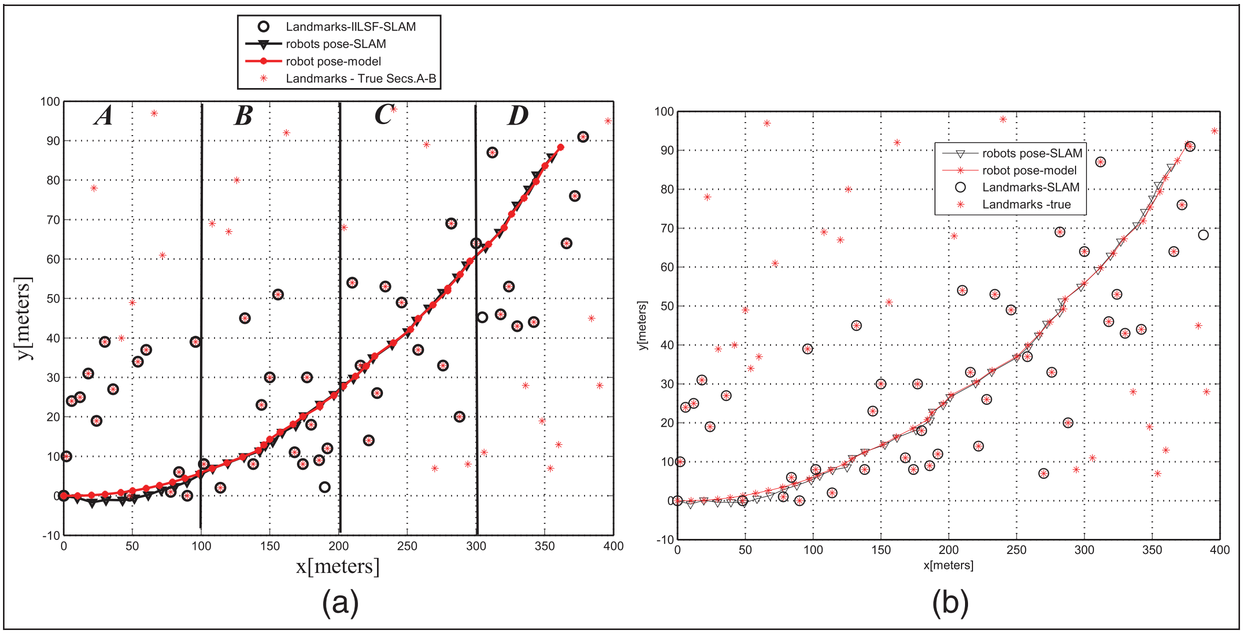

Figure 4. Invariant Information Local Sub-map Filter (a) and absolute map filter (b) constructed global maps.

shown in Figure 4(b), to further demonstrate the advantages of this technique. Figures 4(a) and (b) are identical, highlighting the accuracy in cascading local maps with neither loss of information nor data association ambiguities.

Figure 5 represents separate maps of sections ***A*** and ***B***, plotted together to further explain the approach. Here, the distance (r_{01}) and the angle (β_{01}) serve as the FoR. The reference base is r_{12}, since the new origin at $\boldsymbol{O_L}$ enabled the robot to observe f_1 and f_2 in both sub-maps, while r_{1i}, and $\beta_{i,12}$ are the relative information from sub-map ***B***. Figures 4(a) and 5 confirm the consistency and efficiency of this technique.

The sizes of covariance matrices maintained for both the IILSF and AMF were also observed. For the four sub-maps in the IILSF, the total size was [(49 × 49) + (37 × 37) + (41 × 41) + (49 × 49) = 7852], whilst the size of the single covariance matrix in the AMF was [115 × 115 = 13,225]. This shows a considerable reduction on both the computational burden of updating the covariance matrix and the cost of storage. This saving can be further improved by reducing the number of overlapping landmarks. It will mean restricting the overlapping landmarks to only two. The process of achieving such restrictions and the consequent implications will be considered in future research. These results obtained are eloquent confirmation that the IILSF algorithm is capable of efficiently fusing local maps, thereby generating an accurate and consistent global map of the environment.

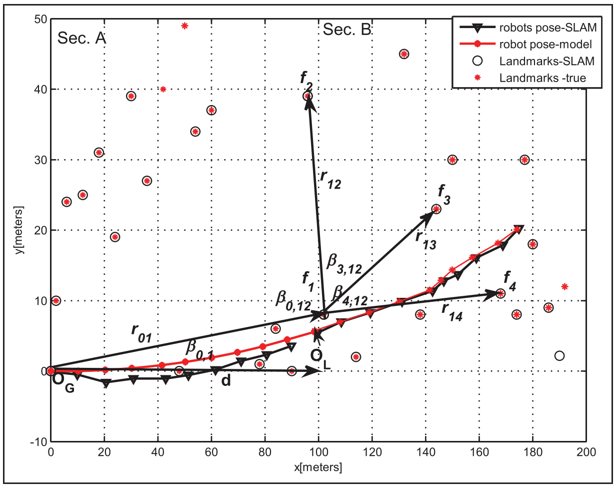

Figure 5. Global map: unstructured environment sections A and B.

In addition, the accumulated pose estimation errors resulting from the AMF and IILSF methods were compared. In Figures 6(a) and 6(b), the average pose error, the estimated pose error and the errors from the axes estimations are displayed.

The average pose errors in the two figures show that errors accumulate in both methods, revealing that the sub-map technique does not reduce the pose estimation error if the robot takes the new origin at its last estimated position in the preceding sub-map. The bulk of the pose estimation error comes from the effect of process noise and it accumulates as distance increases even on a smooth surface. If it is not properly addressed, it could render the SLAM results inaccurate at a global level, even when the sub-map method is engaged. Thus, highlighting the need to search for additional means of reducing the effect of process noise on EKF-SLAM could optimize the efficiency of the sub-map technique. For instance, the IILSF algorithm presented earlier could be optimized through engaging an optimal sub-map size, in addition to being an effective means of preventing the errors accumulated in the previous sub-maps from propagating into the succeeding sub-maps. This discovery motivated the investigation of the effect of process noise in both the predictions and estimations of the robot's position presented in the following section.

Optimal error investigation

This section is aimed at thoroughly investigating the effect of process noise on both the predicted and the estimated

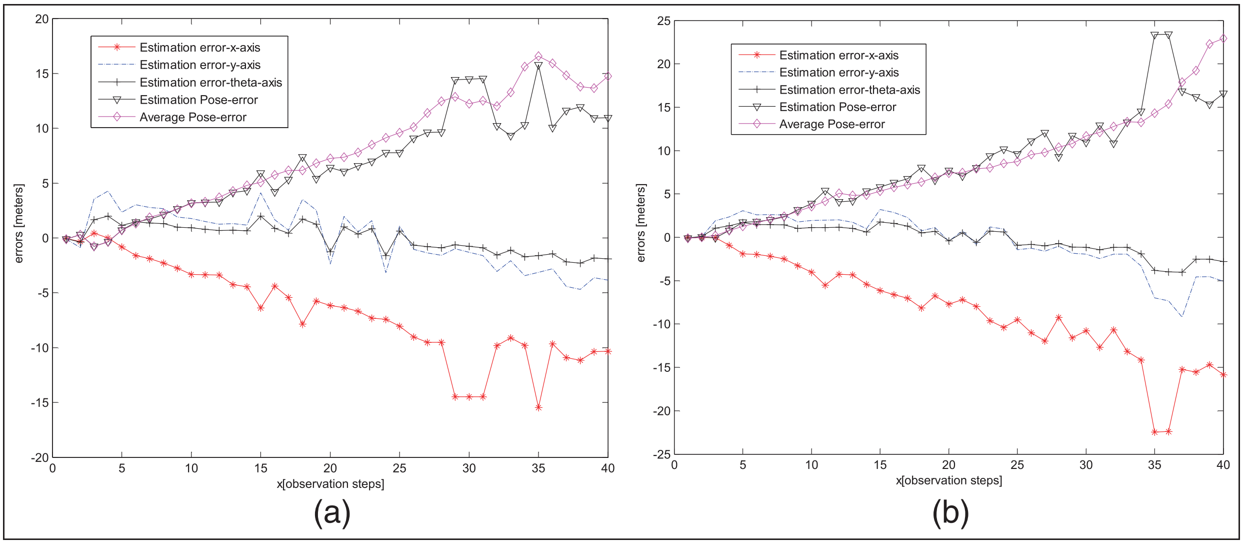

Figure 6. Absolute map filter (a) and Invariant Information Local Sub-map Filter (b) pose error.

positions of the robot. The investigation employs the EKF, which is an extension of the Kalman filter (KF), to handle the non-linear nature of real-life systems. The landmarks are assumed to be identifiable with a laser beam and are also stationary throughout the study. A levelled ground surface was also presumed, while observations were recorded at the instant when the robot has covered a distance equal in length with the range of the laser scanner. The idea here is to provide a yardstick for measuring estimation errors due to increase in distance and possible changes in environmental ground surface. In such a case the optimal error with the optimal distance will serve as a reference.

For this study, a noise-free model is engaged as a reference so that the covariance for both the predictions, $\mathbf{P}_v$, and estimations, $\hat{\mathbf{P}}_v$, of the robot's pose can be computed using Equations (22) and (23):

$$\mathbf{P}_v = E\{(\mathbf{X}_v - \mathbf{X}_{v_{ns}})(\mathbf{X}_v - \mathbf{X}_{v_{ns}}))^T\} \tag{22}$$

$$\hat{\mathbf{P}}_v = E\{(\hat{\mathbf{X}}_v - \hat{\mathbf{X}}_{v_{ns}})(\hat{\mathbf{X}}_v - \hat{\mathbf{X}}_{v_{ns}}))^T\} \tag{23}$$

where the subscript *ns* indicates the output from the noisy model, while $\mathbf{P}_v$ and $\hat{\mathbf{P}}_v$ are based on the sensory data and the results of EKF estimations, respectively.

In order to determine the process noise that will yield the minimum error consistently, over the largest distance, Monte Carlo simulations were carried out using the models described in Section 2. The robot was assumed to be the same as the one described in Section 2.1. The process noise that consistently produced the minimum error for the largest distance covered by the robot is selected as the optimum process noise value, and the error at that point is the acceptable (optimum) error for the sub-map, while the distance at which that occurs is the optimal size of a sub-map. Nearest-neighbour data association was also engaged for reliable validation of observed data.

Monte Carlo simulations

In these simulations, the process noise covariance matrix contains values of normally distributed random noise (**C**). For each value of **C**, the robot covered a distance of about 500 m in a large environment replete with randomly placed landmarks, and the observations were recorded at 25 different positions along its trajectory. A process model with process noise and control without process noise were executed concurrently, thereby defining the effect of the specific process noise on the system by the differences in their results. These effects on both the predictions and estimations are evaluated with Equations (24) and (25), respectively:

$$\tilde{\mathbf{X}}_v = \mathbf{X}_{v_i} - \mathbf{X}_{v_{ns(i)}} \tag{24}$$

$$\tilde{\hat{\mathbf{X}}}_v = \hat{\mathbf{X}}_{v_i} - \hat{\mathbf{X}}_{v_{ns(i)}}, (i = 1 \cdots p) \tag{25}$$

Four sets of simulations at 5, 10, 20 and 50 runs were performed based on the Monte Carlo analysis for each process noise value to validate the consistency of the effect, while Equation (26) was engaged to evaluate its average effect on the predicted positions:

$$\bar{\tilde{\mathbf{X}}}_v = \frac{1}{Ts}\sum_{i}^{Ts} \tilde{\mathbf{X}}_{v_i} \tag{26}$$

Likewise, Equation (27) evaluates the average effect on the estimated trajectories:

$$\bar{\tilde{\hat{\mathbf{X}}}}_v = \frac{1}{Ts}\sum_{i}^{Ts} \tilde{\hat{\mathbf{X}}}_{v_i} \tag{27}$$

where T_s is the total number of runs and $\bar{\tilde{\mathbf{X}}}_v$ and $\bar{\tilde{\hat{\mathbf{X}}}}_v$ are $(1 \times p)$ row vectors containing the average error values at each of the p positions for predicted and estimated trajectories, respectively.

Results. The first three simulations (5, 10 and 20) served as a training to determine the characteristic of the filter, followed by 50 runs, which was used to estimate the optimal values. Figures 7(a) and (b) show that the error caused by the process noise covariance was proportional to the value of the process noise for five runs.

However, as the number of runs increased from 10 and upwards, the filter stabilized and the trend changed, leading to $Pn(6)$ consistently producing the least but one error effect (see Figures 8(a), 8(b), 9(a) and 9(b)). This shows that a filter should be trained and properly tuned for better performance. The rest of the analysis is based on the 50 runs after the filter has stabilized. The predicted and estimated trajectories of the robot, for the 50 simulations showing the discrepancy between the noisy and true measurements, are displayed in Figure 10. Here, 'True-Model Pre-Pose' and 'Noisy-Model Pre-Pose' show the predicted robot positions, while 'True-Model Est-Pose' and 'Noisy-Model Est-Pose', depict the estimations. The markers show the positions where observations were taken, while the accumulated error effect along the *x*-axis and *y*-axis are represented as $\mathbf{e_x}$ and $\mathbf{e_y}$, respectively. The estimations match closely their corresponding predicted positions but differ substantially from the noisy results. Tables 1(a) and (b) contain samples of the average process noise covariance effects (errors) on the pose predictions and estimations, respectively, obtained at five-step intervals. The second columns show the values of the process noise (Pn) used while the average errors are contained under the remaining columns. The subscripts of $\mathbf{X}$ and $\hat{\mathbf{X}}$ represent the predicted and estimated positions of the robot or the distance covered, respectively when the errors were measured, where X_0, X_1, X_2, X_3, X_4 and X_5 are 0, 123, 243, 353, 449 and 528; while $\hat{X}_0$, $\hat{X}_1$, $\hat{X}_2$, $\hat{X}_3$, $\hat{X}_4$ and $\hat{X}_5$ are 0, 100, 219, 331, 431 and 514 meters, for predictions and estimations, respectively. In relation to the tables, the differences between predicted, $\mathbf{X}_v$, and the estimated, $\hat{\mathbf{X}}_v$, positions are the covariance of the state vector with regard to the robot:

$$\mathbf{P}_v(k) = E\{(\mathbf{X}_v(k) - \hat{\mathbf{X}}_v(k))(\mathbf{X}_v(k) - \hat{\mathbf{X}}_v(k))^T\}$$

In addition, Figure 9(a) presents the average effect of each value of the process noise on the predicted robot pose, while Figure 9(b) shows the similar effect on the robot pose estimations.

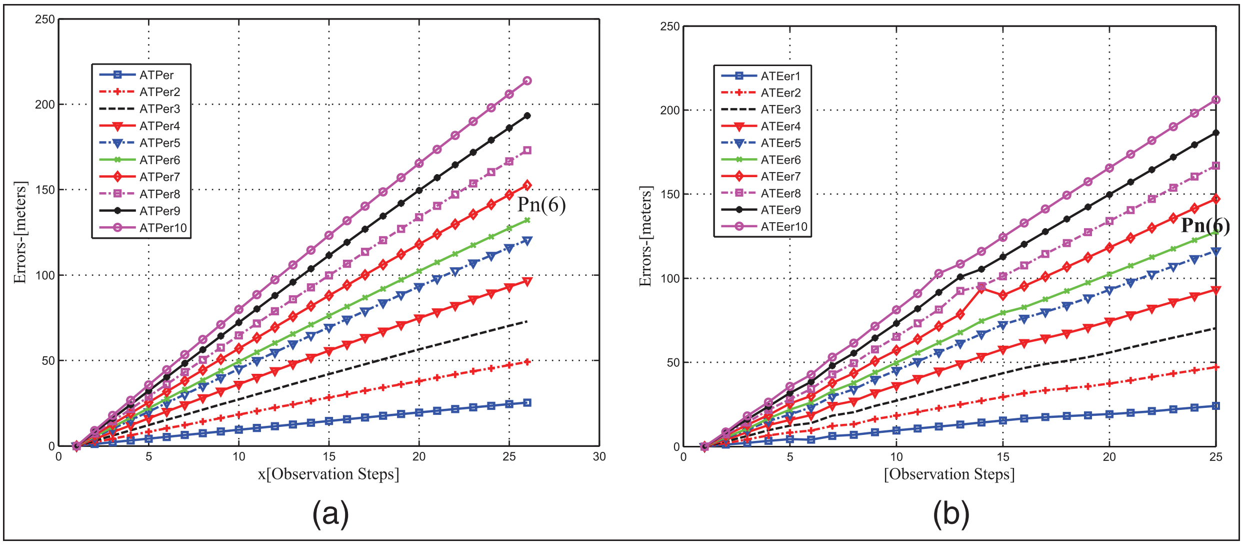

Figure 7. Predicted (a) and estimated (b) pose estimation error for five Monte Carlo runs.

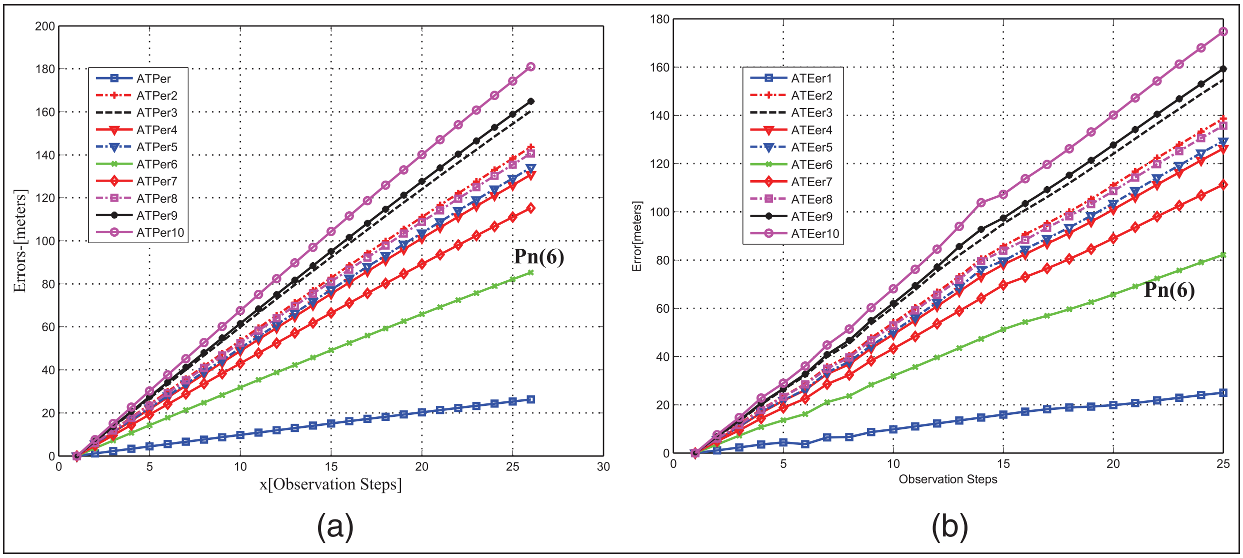

Figure 8. Predicted (a) and estimated (b) pose estimation error for 20 Monte Carlo runs.

Discussion

The accumulated effect of the process noise is shown as $\mathbf{e}_x$ and $\mathbf{e}_y$ along the x and y-axes in Figure 10. It reaffirms that the effect of process noise on the robot position accumulates as the distance increases. It could be observed that at a distance of about 100 m, the error has not accumulated significantly since the trajectories obtained from both the true and noisy models are very similar. If this distance is considered as an optimal size of a sub-map, then the error produced by that process noise could be extracted from the numerical data. For example, Tables 1(a) and (b) show that an average process noise of $Pn = 0.27$ m will effect an error of 14.01 m on the predictions and 10.83 m on the estimations at a distance of 100 m.

However, it was also discovered that the effect of process noise is proportional to the level of noise only during the first few simulation runs. After several simulations, the filter stabilizes at an optimal value occurring at a noise level specific for the filter involved, which in most cases may not be at the least process noise level. For instance, in the 50-run simulation the average error effect caused by $Pn(6)$ is relatively smaller to the others except that of $Pn(1)$ for all the positions, showing that the filter has stabilized, justifying the choice of $Pn(6)$ as the optimal process noise level and 100 m as the optimal size of a sub-map. Hence, a filter should be trained through several runs before determining the optimal error value. In addition, it is evident from Figures 9(a) and (b) that the effects of process noise maintained different values for the predictions and the estimations, emphasizing the need to give careful

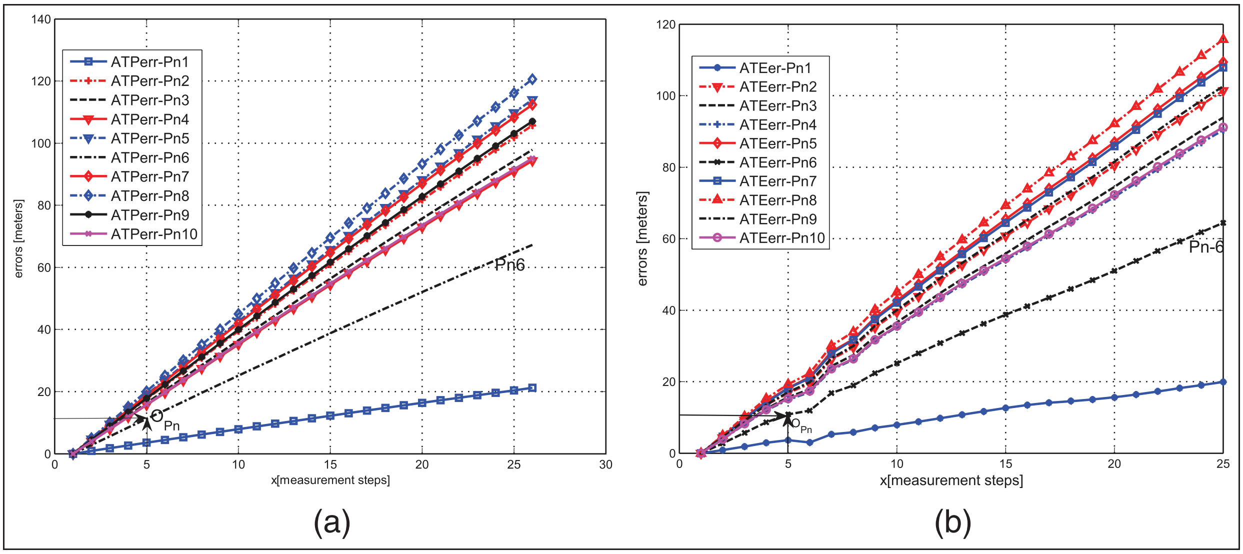

Figure 9. Predicted (a) and estimated (b) pose estimation error for 50 Monte Carlo runs.

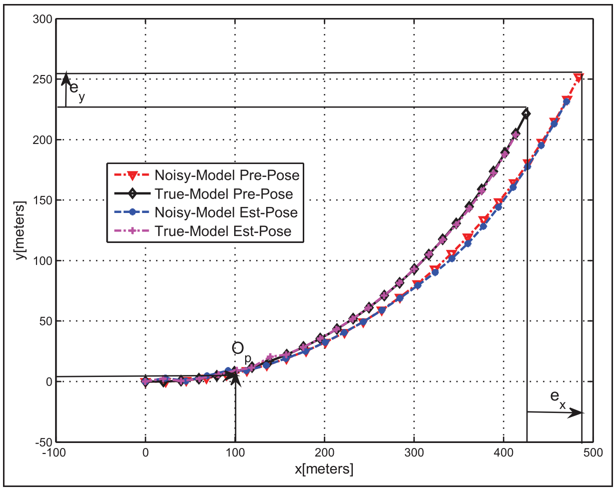

Figure 10. Predicted and estimated robot trajectories showing the discrepancy between the noisy and true measurements.

consideration to both the predicted and estimated error values before deciding the optimal values.

The implication of this result is that a system operating at the optimal noise level under similar scenarios will sustain minimal error if it operates within the optimal sub-map size. Such knowledge serves as a tool for the optimizing sub-map technique, leading to improved accuracy and efficiency in SLAM results. Furthermore, the approximated optimal error effect could serve as a reference to monitor and adjust real-life system estimations, leading to adaptive minimization of the accumulated pose estimation error.

In addition, this process yields a qualitative definition of an optimal sub-map size, based on the distance within the region where the error effect pattern remains similar and minimal. Moreover, the criteria for initializing new sub-maps could include either or both the effect and/or the value of optimal process noise to optimize the sub-map techniques. Such criteria will ensure that the error or process noise level in any sub-map does not exceed the pre-set acceptable value.

The next section contains a proposal for addressing the growth of the process noise and its prevention from being propagated into succeeding sub-maps.

Error-correction process

Although the sub-map technique reduces the accumulated pose estimation errors resulting from non-linear models used in EKF-SLAM, Figures 6(a) and (b) show that it does not address the major error source, which accrues from process noise. In order to improve the accuracy of the localization results, the error-correction process was incorporated into the system. The process model developed in Section 2 was engaged to predict and estimate the robot's pose. In the models, the optimal process noise covariance matrix (already determined) served as the noisy model. Then an observer compares the results obtained from both models and computes the error accumulated in the current sub-map using Equation (28):

$$\tilde{\mathbf{X}}_{v_i} = \mathbf{X}_{v_i ns} - \mathbf{X}_{v_i} \quad (28)$$

where $\mathbf{X}_{v_{ns}}$ denotes the predictions with the optimal noise, while $\mathbf{X}_v$ are the predictions without process noise. The subscripts $i = 1 \cdots p$ indicate the positions where measurements were taken.

When starting a new local map, the robot starting point (new origin) is set to its last estimated position in the previous local map, whereas the computed accumulated position error, $\tilde{\mathbf{X}}_{v_i}$, is set to zero. In this way, both the predictions and estimations in any local area are not affected by the errors accumulated in the previous local sub-maps. The efficacy of the error-correcting process was verified through

Table 1. Validation data for the structured environment.

No	*Pn*	X_0	X_1	X_2	X_3	X_4	X_5
1	0.04	0	4.41	8.78	13.06	17.21	21.19
2	0.09	0	22.01	43.79	65.14	85.84	105.68
3	0.13	0	20.37	40.54	60.30	79.46	97.82
4	0.18	0	19.64	39.08	58.13	76.60	94.30
5	0.22	0	23.74	47.24	70.27	92.60	114.00
6	0.27	0	14.01	27.87	41.46	54.63	67.26
7	0.31	0	23.40	46.57	69.27	91.28	112.37
8	0.35	0	25.11	49.96	74.32	97.93	120.56
9	0.40	0	22.28	44.34	65.96	86.92	107.01
10	0.44	0	19.80	39.39	58.60	77.21	95.06

(a) Effect of noise (*Pn*) on predicted robot position.

Sn	*Pn*	$\hat{X}_0$	$\hat{X}_1$	$\hat{X}_2$	$\hat{X}_3$	$\hat{X}_4$	$\hat{X}_5$
1	0.04	0	3.64	7.89	12.61	15.58	19.91
2	0.09	0	16.78	39.50	60.70	80.49	101.39
3	0.13	0	15.67	36.62	55.86	74.50	93.84
4	0.18	0	14.91	35.28	54.06	71.68	90.38
5	0.22	0	18.24	42.66	65.51	87.00	109.43
6	0.27	0	10.83	25.15	38.74	51.04	64.38
7	0.31	0	18.09	42.07	64.38	85.88	107.88
8	0.35	0	19.26	45.04	69.27	92.11	115.73
9	0.40	0	17.04	40.03	61.20	81.53	102.64
10	0.44	0	15.22	35.58	54.40	72.30	91.14

(a) Effect of noise (*Pn*) on estimated robot position.

two simulations performed in the same environment, which was divided into the same number of local maps. In the first simulation, no correction was applied, while in the second one the error-correcting process was implemented.

Results of the error-correction process

The results obtained from the two simulations are presented in Figures 11(a), and (b). Figure 11(a) represents the predicted robot trajectory with the optimal noise covariance as 'Noisy-Model Pre-Pose', while the 'True-Model Pre-Pose' depicts the predictions without noise. It shows that the error due to process noise continued building up, reaching the value shown as (Total-$\mathbf{e}_\mathbf{x}$) and (Total-$\mathbf{e}_\mathbf{y}$) along the x and y-axes, respectively. Secondly, at the optimal distance (in this case 100 m), which marks the end of the first sub-map, the accumulated pose errors appear to be very small and negligible; however, it could be seen from the tables that their values are still substantial. This underscores the need for additional means of reducing the effect of process noise in EKF-based SLAM. The efficacy of the error-correcting process adopted is shown in Figure 11(b). It could be observed from Figure 11(b) that the error-correcting process was efficient in addressing the error due to process noise, leading to accurate prediction of the robot positions. In the next section the results obtained from optimal error investigation and proposed error-correction process are employed to optimize the performance of the IILSF-based SLAM. These results stimulated an interest to find out how the errors grow within the sub-maps, leading to the results presented in the next section.

Results of investigation of error growths

Error growth in sub-maps was further investigated for both the predicted and estimated positions of the robot. The results obtained further demonstrate the benefits derivable by engaging the error-correction process.

Prediction error growth at the sub-map level

Figures 12(a) and (b) display the results of further investigations on the growth of errors at the sub-map level, in both the normal and optimized sub-map SLAM methods, for the predicted trajectories of the robot. The plots show the pose estimation error accumulated within the sub-maps, while the markers indicate the positions at which measurements were taken. From the *error*-axis of Figures 12(a), it could be seen that errors started at zero for all the optimum sub-maps and accumulated to a maximum of about 1.5 and 9.8 m for sub-maps A and D, respectively, showing the effectiveness of the error-correcting process applied. On the other hand, Figure 12(b) demonstrates that the errors in the normal sub-map techniques kept on accumulating, starting at zero in sub-map A and building up to over 95 m in sub-map D in this case.

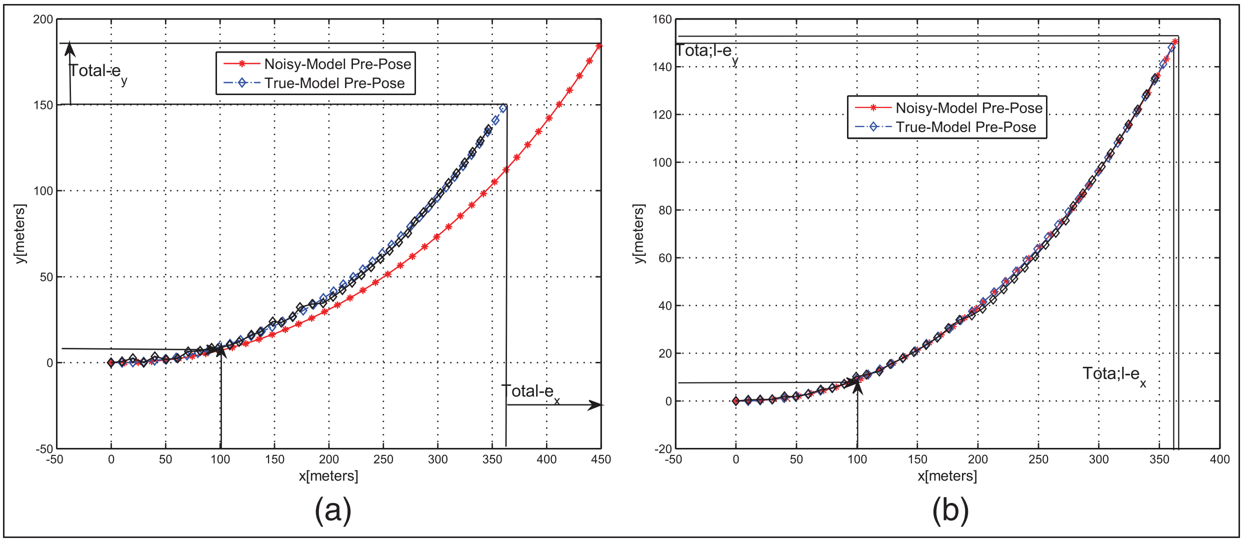

Figure 11. Cumulated versus corrected error effect on predicted trajectories.

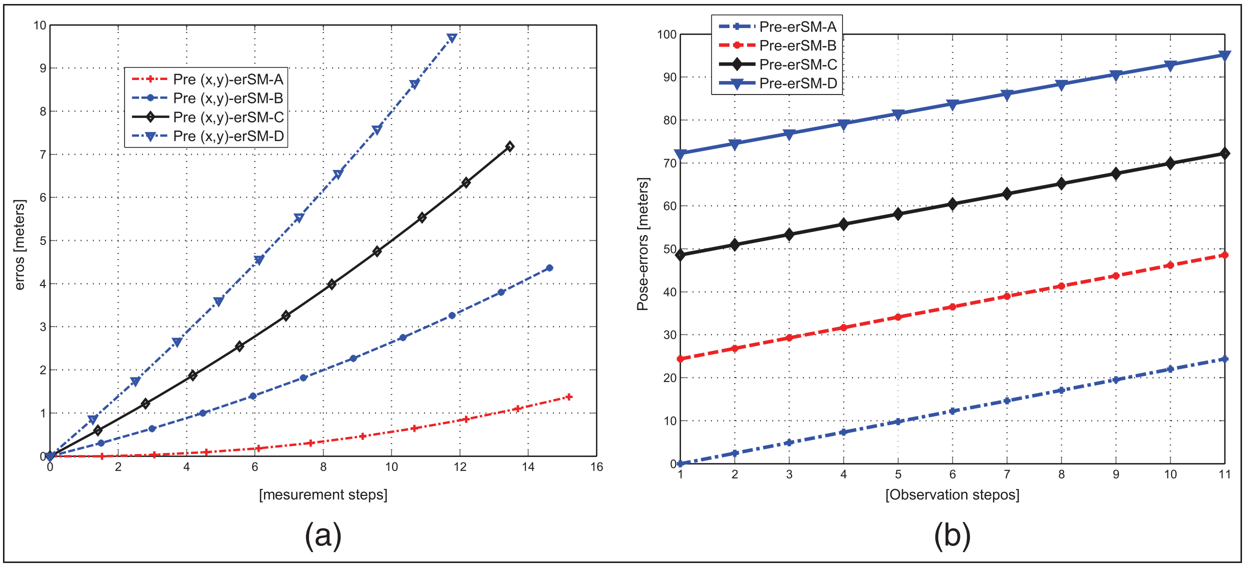

Figure 12. Predicted errors in the sub-map, after applying and without applying the error-correction process.

Estimation error growth at the sub-map level

The results of similar investigations carried out for the estimated trajectories of the robot, for both normal and optimized sub-maps, are presented in Figures 13(a) and (b). These results show that the error corrections performed at the prediction stage effectively worked in the estimation stage. Although the error growth is not uniformly linear in both cases, the benefits of the optimization exercise are apparent from Figure 13(a). For instance, the *y*-axis of Figures 13(a) and (b) show that the errors accumulated in all the sub-maps were limited to about 14 m in sub-maps *A–D* for the optimized sub-map estimations, while the boundaries for the errors in the normal sub-map technique rose from a maximum value of 20 m in sub-map *A* to a maximum level of about 95 m in sub-map *D*. This shows that the optimization process successfully limited the error below 3% of the total distance, leading to over 97% accuracy of the estimations.

The presented optimizing technique and the error-correcting process were combined to optimize the preliminary IILSF algorithm as an example of optimizing the efficiency of the sub-map technique for large-scale SLAM in the following section.

Complete simultaneous localization and mapping simulation

Following the successful optimization of the sub-map technique and the efficacy of the error-correcting process, further simulations were carried out to demonstrate complete SLAM

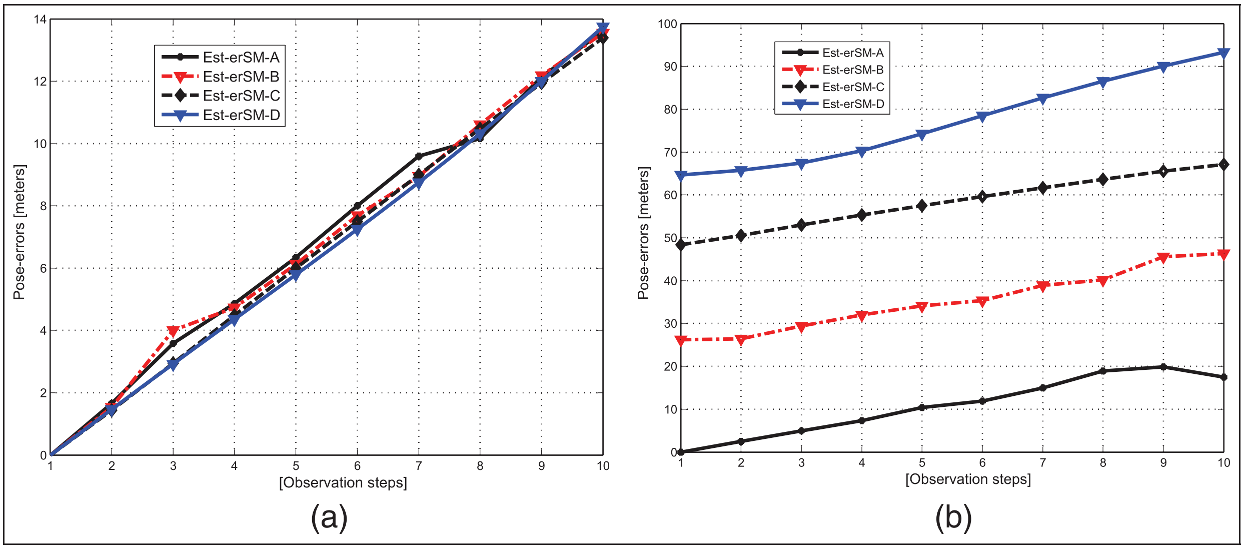

Figure 13. Estimated errors in sub-maps, after applying and without applying the error-correction process.

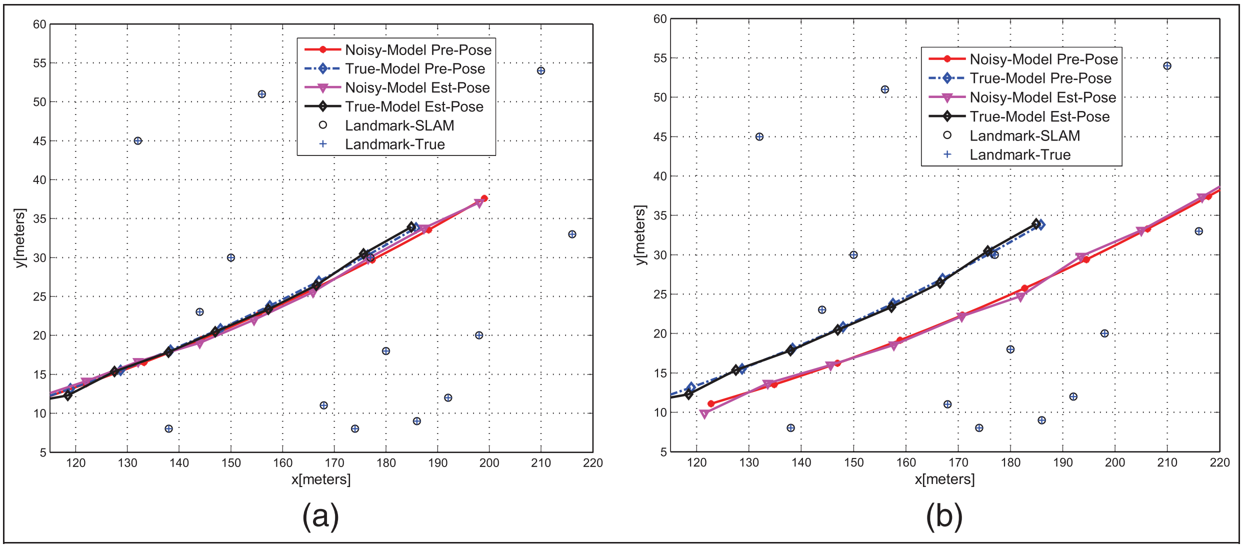

Figure 14. Robot's trajectory in sub-map B: using and without using the proposed optimal algorithm.

performance in a large environment. Both techniques were applied to the earlier preliminary IILSF algorithm as a test case for complete SLAM simulation. The simulations are similar in detail to the ones already described, with the exception that estimations and mapping at the sub-map levels are included. The idea behind further simulations is to investigate how the optimizations performed on the prediction model affect the mapping and localization estimations. In addition, to confirm the efficiency of the technique in optimizing the performance of the IILSF algorithm, the optimum process noise, *Pn*(6) (see Table 1(b)) with a variance of 0.073, computed at the optimum map size was used in this phase of simulations, while every other parameter used in earlier simulations was maintained. In contrast, another simulation was carried out with the same variance without engaging the error-correcting process.

Complete SLAM simulation results

The results of optimized sub-map-based SLAM in a large environment are shown in Figures 14(a) and 15(a). In the figures, 'Noisy-Model Pre-Pose' and 'True-Model Pre-Pose' denote the preliminary and optimized IILSF predicted robot trajectories, respectively. Likewise, 'Noisy-Model Est-Pose' shows the preliminary estimated robot trajectory, while 'True-Model Est-Pose' depicts the optimized estimated robot trajectory. In the maps, 'Landmarks-True' and 'Landmarks-SLAM' represent the actual and SLAM estimated landmarks locations, respectively. These results validate the efficacy of the error-correcting process, performed during the prediction stage, in optimizing the sub-map technique used in the IILSF, for both predicting and estimating the trajectories of the robot. This optimization process resulted into a more

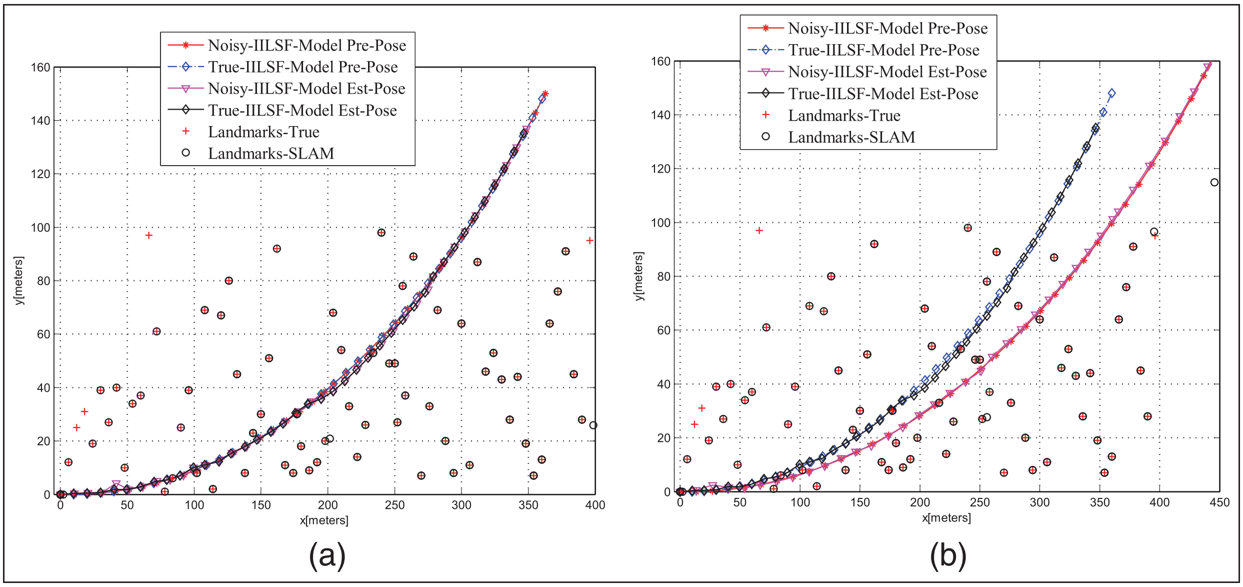

Figure 15. Global map: using and without using the optimized algorithm.

consistent map as could be observed from the figures, demonstrating an improvement on basic SLAM results.

The results of contrast simulations performed employing the preliminary IILSF algorithm within the same environments, but without applying the optimization method, are displayed with the same descriptions in Figures 14(b) and 15(b), reaffirming that the sub-map technique alone does not remove the pose estimation error caused by the process noise. Rather, the accumulated error in the preceding sub-map is propagated into the next sub-map, if no action is taken to counteract its continuous accumulation. Such error accumulations degrade the quality of the map and the SLAM estimated trajectory of the robot, leading to non-convergence of the map. Comparing Figures 14(a) with (b) highlights the benefits derivable by the optimization of the sub-map techniques.

The error-correction process was applied at the prediction stage only: it did not impose additional time requirement. It neither inflates the computational cost nor the storage requirements because it was not repeated during the estimation and updating stages. In addition, observed features were relative to the corrected pose estimations, and the data association decisions were taken based on the optimized pose estimations so remained valid without any additional cost.

Conclusions

This study has successfully presented a method that addresses some outstanding issues with the sub-map-based SLAM technique. It provides a means of determining the optimal size of a sub-map. The magnitude of any error that could be accommodated within a sub-map without infringing on the consistency and accuracy of the SLAM result is evaluated from extensive Monte Carlo simulations. Furthermore, the work successfully engaged the optimal error and the optimal sub-map size to optimize the sub-map method, leading to more efficient large-scale SLAM performance. In addition, the error-correcting process employed resulted in more consistent and accurate simulation results.

Funding

This work was supported by the Education Trust Fund (ETF), Federal Government of Nigeria.

References

Blanco J-L, Fernández-Madrigal J-A and Gonzalez J (2007) A new approach for large-scale localization and mapping: hybrid metric-topological SLAM. In: *IEEE international conference on robotics and automation, IEEE*, pp.2061–2067.

Brooks A and Bailey T (2008) Combining FastSLAM and EKF-SLAM for reliable mapping. In: *workshop on the algorithm fundamentals of robotics (WAFR)*, University of Sydney Australia, pp.1–16.

Castellanos JA, Tardós RM-CJD and Neira J (2007) Robocentric map joining: improving the consistency of EKF-SLAM. *Robotics and Autonomous Systems* 55(1): 21–29.

Cho HC, Fadali MS, Lee KS, et al. (2002) Adaptive position and trajectory theory and control of autonomous mobile robot system with random friction. *IET Control Application* 4(12): 2733–2742.

Cocaud C and Kubota T (2010) SURF-based SLAM scheme using octree occupancy grid for autonomous landing on asteroids. *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, Sapporo, Japan, vol. 30, pp. 275–282.

Davison AJ and Murray DW (2002) Simultaneous localisation and map-building using active vision. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 24(7): 865–880.

Deans MC and Hebert M (2000) invariant filtering for simultaneous localisation and mapping. In: *proceedings of the IEEE international conference on robotics and automation*, San Francisco, CA, 4 October, pp.1042–1047.

Dudek G and Jenkin M (2010) *Computational Principles of Mobile Robotics*. 2nd ed.Vol. 1. New York: Cambridge University Press.

Durrant-Whyte H and Bailey T (2006) Simultaneous localization and mapping tutorial Part I. *IEEE Robotics and Automation Magazine* 13(2): 99–110.

Harter D (2005) Evolving neurodynamic controllers for autonomous robots. In: *proceedings of the 2005 IEEE international joint conference on neural networks (IJCNN'05)*, Vol. 1. IEEE, pp.137–142.

Härter FP and de Campos Velho HF (2008) New approach to applying neural network in nonlinear dynamic model. *Applied Mathematical Modelling* 32(12): 2621–2633.

Huang S, Wang Z and Dissanayake G (2006) Mapping large scale environments using relative position position information among landmarks. In: *proceedings of the IEEE International conference on robots and automation*, Orlando, Florida, 20 June, pp.2297–2302.

Ihemadu OC, Naeem W and Ferguson RS (2012) Statistical approach for place recognition in simultaneous localisation and mapping. In: *proceedings IET conference on Irish signals and systems*, Vol. 1.

Ihemadu OC, Naeem W, Ferguson RS, et al. (2013) Invariant Information Local Sub-map Filter (IILSF) for efficient simultaneous localisation and mapping of large environments. In: *9th international workshop on robot motion and control (RoMoCo'13)*, Wasowo, Poland, accepted 2013.

Koenig A, Kessler J and Gross HM (2008) A graph matching technique for an appearance-based, visual SLAM-approach using Rao-Blackwellized particle filters. In: *international conference on intelligent robots and systems*, Nice, France, pp.1576–1581.

Liu Y, Sun F, Tao T, et al. (2007) A solution to active SLAM problem based on optimal control. In: *international conference on mechatronics and automation (ICMA)*, Harbin, China, 20 August, pp.314–319.

Paz LM and Neira J (2006) Optimal local map size for EKF-based SLAM. In: *2006 IEEE/RSJ international conference on intelligent robots and systems, IEEE*, pp.5019–5025.

Paz LM, Piniés P, Tardós JD and Neira J (2008) Large-scale 6-DOF SLAM with stereo-in-hand. *IEEE Transactions on Robotics* 24(5): 946–957.

Pinies P and Tardos JD (2008) Large-scale SLAM building conditionally independent local maps: Application to monocular vision. *IEEE Transactions on Robotics* 24(5): 1094–1106.

Porta JM and Krose BJ (2006) Appearance-based concurrent map building and localization. *Robotics and Autonomous Systems* 54(2): 159–164.

Seadan M, Chee WL and Marcelo HA (2007) Appearance-based simultaneous localization and mapping with map-loop closing using omni-directional camera. In: *IEEE international conference on advanced intelligent and mechatronics (IEEE/ASME)*, Zurich.

Tomizawa T, Muramatsu S, Sato M, et al. (2012) Development of an Intelligent Senior-Car in a Pedestrian Walkway. *Advanced Robotics* 26(14): 1577–1602.

Wu B-F, Jen C-L, Li W-F, et al. (2013) RGB-D sensor based SLAM and human tracking with Bayesian framework for wheelchair robots. *International Conference on Advanced Robotics and Intelligent Systems (ARIS)*, pp. 110–115. IEEE.

Zhang L, Zapata R and Lépinay P (2012) Self-adaptive Monte Carlo localization for mobile robots using range finders. *Robotica* 30(2): 229–244.