The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Augmented EKF based SLAM method for Improving the Accuracy of the Feature Map

Jeong-Gwan Kang, Won-Seok Choi, Su-Yong An, and Se-Young Oh, *Senior Member, IEEE*

Abstract—**In this paper, we address a method for improving the accuracy of the feature map from the extended Kalman filter based SLAM (EKF SLAM) by estimating the systematic parameters of the robot. Most error of the robot while traveling is divided into two categories: systematic and non systematic error. The systematic error contributes much more to odometry errors than non systematic one on most smooth indoor surfaces. So, we appended the systematic parameters of the robot to the state vector of EKF SLAM as its elements and estimated the systematic parameters while performing the prediction and update state of EKF SLAM. Because the additional elements to be estimated are appended to the state vector of the EKF SLAM, this is called an augmented EKF SLAM (AEKF SLAM). Experimental result is presented to validate that our AEKF SLAM is able to generate a more accurate feature map than conventional EKF SLAM by decreasing odometric error of the robot.**

I. Introduction

TO navigate in an unknown environment, an autonomous mobile robot needs to build a map by the relative observations of environment and simultaneously to compute the pose of the robot by using this map. This is called Simultaneous Localization And Mapping (SLAM) problem in mobile robotics.

Two analytic approaches have been widely used to solve SLAM. The first uses an Extended Kalman Filter (EKF) to generate the feature map and to manage the associated uncertainty by generating a stochastic feature map with a single state vector and a covariance matrix [1]. Successful SLAM algorithms that use EKF have been developed for various applications [2]-[9]. The second analytic approach to SLAM uses a Rao-Blackwellized Particle Filter (RBPF) [10],[11]. One version, FastSLAM [12],[13] uses a particle filter to represent potential trajectories of the robot; the probability that a particle survives is proportional to the likelihood of observation given that particle and its map. FastSLAM can therefore represent an arbitrary distribution over the robot pose and thus operate when noise is non-Gaussian. FastSLAM is better than EKF SLAM at solving the data association problem for detecting loop closures.

Most errors in SLAM are caused by biased noises arising from unequal wheel diameters, wheel diameters that differ from their nominal values, wheel misalignment, and uncertain effective wheel base [14]. These biased noises cause a velocity difference between the left and right wheels, which skews the robot's odometric trajectory to one side and conventional SLAM methods including EKF/Fast SLAM cannot guarantee good performance in this situation because they assume zero mean noise when correcting odometric error.

Borenstein and Feng [14] divided possible error sources in the robot into two categories including systematic and non systematic errors and developed a calibration technique called UMB method which calibrated for systematic errors of the differential wheeled mobile robot. Doh et al. [15] proposed a calibration method called PC-method for compensating the systematic error. Antonelli [16] identified 3-parameters odometric model and presented a calibration method based on the lest-squares technique. These odometry calibration methods are off-line method, so the odometry calibration is carried out after traveling a specific path.

Larsen et al. [17] proposed an on-line calibration method for the systematic error using an Augmented Extended Kalman Filter (AEKF) which simultaneously estimates the robot pose and the parameters characterizing the systematic error. Martinelli [18] extended the AEKF based approach which estimated not only systematic errors but also non systematic errors. Yun [19] suggested an odometric error calibration method using AEKF and inherent home positioning for differential wheel typed home service robot. However, an accurate feature map is needed to implement the AEKF based approach, because it is modified from the EKF localization method.

In this paper, we apply the AEKF method to the SLAM problem. The AEKF computes the parameters characterizing the systematic error using a feature map from EKF SLAM and the EKF SLAM adopts the systematic parameters of the robot for compensating an odometry error in its prediction step. Because the state of EKF SLAM and systematic error parameters are dependent and complimentary to each other, AEKF SLAM is able to generate a more accurate feature map by decreasing odometry error of the robot.

Manuscript received March 10, 2010. This work was supported by National Strategic R&D Program for Industrial Technology of the Korean Ministry of Knowledge Economy.

J. G. Kang is with the Electrical Engineering Department, Pohang University of Science and Technology, San 31, Hyojadong, Namgu, Pohang, Republic of Korea. (phone: 82-54-279-2904, fax: 82-54-279-5594, e-mail: naroo1@postech.ac.kr).

W. S. Choi is with the Electrical Engineering Department, Pohang University of Science and Technology, Pohang, Republic of Korea (e-mail:onlydol@postech.ac.kr).

S. Y. An is with the Electrical Engineering Department, Pohang University of Science and Technology, Pohang, Republic of Korea (e-mail:grasshop@postech.ac.kr).

S.Y. Oh is with the Electrical Engineering Department, Pohang University o f Science and Technology, Pohang, Republic of Korea (e-mail: syoh@postech.ac.kr).

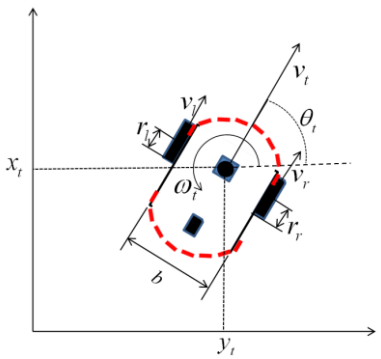


Fig. 1. Model of differential wheeled mobile robot with relevant variables.

This paper is organized as follows. Section II presents a short primer on the odometry error model of differential wheeled robot. Section III presents the architecture of our AEKF SLAM framework for generating an accurate feature map. In Section IV, the performance of the proposed scheme is evaluated. Conclusions are presented in Section V.

II. Odometric Error Model of Differential Wheeled Robot

Most error of the robot while traveling is divided into two categories: systematic and non systematic error. Non systematic errors are those errors that are caused by interaction of the robot with unpredictable features of the environment. These non systematic errors are generated from uneven floor of wheel-slippage due to the slippery floors, over-acceleration, fast turning, external/internal forces and nonpoint wheel contact with the floor [14]. It is almost impossible to predict these non systematic errors.

Systematic errors are vehicle specific and usually caused by imperfections in the design and mechanical implementation of a mobile robot. In case of differential wheeled robot, several sources which generate systematic errors exist including unequal diameters between left and right wheel, difference between actual average of wheel diameters and nominal diameter, misalignment of wheels and uncertainty about the effective wheelbase. Especially, *unequal wheel diameters* and *the uncertainty about the effective wheel base* are two most notorious systematic error sources [14].

The odometry error of the robot is more affected by the systematic error in most of the indoor environment. Unequal wheel diameters noise produce a velocity difference between the left and right wheels of the robot, making its odometry trajectory lean toward left or right, even when the robot actually moves straight and the uncertainty about the effective wheel base has an effect only when turning.

A simple way to characterize the odometry error for a mobile robot with a differential drive system is obtained by modeling separately the error in the translation of each wheel [20]). Let r_l, r_r and b be the nominal value of left/right wheel diameter and the robot wheel base (Fig. 1), respectively and N be the number of wheel rotation. The linear and angular velocity of the robot can be computed as follows:

$$D_{l/r} = 2\pi r_{l/r} \cdot N \tag{1}$$

$$v_{l/r} = \frac{D_{l/r}}{dt} = \frac{2\pi r_{l/r} \cdot N}{dt} \tag{2}$$

$$v = \frac{v_l + v_r}{2} = \frac{\pi \cdot N\left(r_l + r_r\right)}{dt} \tag{3}$$

$$\omega = \frac{v_r - v_l}{b} = \frac{1}{b} \cdot \frac{2\pi \cdot \left(r_r - r_l\right) \cdot N}{dt} \tag{4}$$

Let the systematic parameters of the robot for left/right wheel diameter and robot wheel base be δ_l, δ_r and δ_b, respectively. The actual diameter of left/right wheel, $r^*_{l/r}$, and actual wheel base of the robot, b^*, can be defined as follows:

$$r^*_l = \delta_l r_l \tag{5}$$

$$r^*_r = \delta_r r_r \tag{6}$$

$$b^* = \delta_b b \tag{7}$$

Then actual translation velocity of the left/right wheel is calculated as:

$$v^*_{l/r} = \frac{2\pi\delta_{l/r} r_{l/r} \cdot N}{dt} = \delta_{l/r} v_{l/r} \tag{8}$$

So, the actual linear and rotate velocities of the robot considering the systematic error are represented as:

$$v^* = \frac{v^*_l + v^*_r}{2} = \frac{\delta_l v_l + \delta_r v_r}{2} \tag{9}$$

$$\omega^* = \frac{v^*_r - v^*_l}{b^*} = \frac{\delta_r v_r - \delta_l v_l}{\delta_b b} \tag{10}$$

III. Improve the Accuracy of the Feature Map using Systematic Parameter Estimation

The EKF based SLAM is very sensitive to the system and sensor noise model. Therefore, an accurate noise modeling is required. However it is difficult to find accurate motion/sensor models and noise variances in the real robot system. In addition to this, since EKF has a usual assumption of white Gaussian noise, if there are colored noise or systematic bias error, EKF SLAM cannot guarantee an accurate feature map. For example, if the distance between the left and right wheels or the wheel radius is not precisely known, the time update of EKF SLAM will lead the estimates to drift. In the absence of sensor measurements, this could cause the estimation errors to grow unbound.

These odometry errors of the robot can be reduced by estimating the systematic parameters δ_l, δ_r and δ_b based on the AEKF method. In AEKF SLAM, the systematic parameters of the robot are augmented to the state vector as its elements. Thus, the systematic parameters, the robot pose and the feature positions are updated using the Kalman gain and the innovation in the measurement update step. The updated

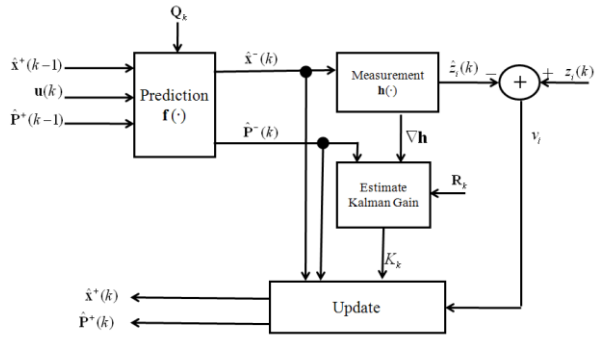


Fig. 2. Overall architecture of AEKF-SLAM. It has a same structure as standard EKF SLAM and the motion model of the robot in the prediction step is modified for considering the systematic parameters of the robot.

systematic parameters reduce the odometric error of the robot in the prediction step. Then the reduced odometric error in the prediction step makes the feature map of EKF SLAM more accurate. Finally, the proposed AEKF SLAM can provide an accurate feature map and well estimated systematic parameters of the robot simultaneously after traveling in an unknown environment.

A. Overall Architecture of AEKF SLAM

Let $\mathbf{x}_w(k)=[\delta_l,\ \delta_l,\ \delta_b]^T$ be a state vector which has the systematic error parameters as its elements, $\mathbf{x}_v(k)=[x_v(k), y_v(k), \theta_v(k)]^T$ be the estimated robot pose at time k, $\mathbf{x}_m(k)=[x_1, y_1, \ldots, x_n, y_n]^T$ be the feature position vector which has the positions of the observed features. Then, the state vector for the EKF SLAM with systematic error parameter is defined as follows:

$$\begin{aligned}\mathbf{x}(k) &= [\mathbf{x}_v(k), \mathbf{x}_w(k), \mathbf{x}_m(k)]^T \\ &= \left[x_v(k), y_v(k), \theta_v(k), \delta_l, \delta_r, \delta_b, x_1, y_1, \ldots, x_n, y_n\right]^T\end{aligned} \tag{11}$$

The covariance matrix $\mathbf{P}(k)$ which represent the uncertainty of the robot state vector is defined as follows:

$$\mathbf{P}(k) = \begin{bmatrix} \mathbf{P}_{vv} & \mathbf{P}_{vw} & \mathbf{P}_{vm} \\ \mathbf{P}_{vw}^T & \mathbf{P}_{ww} & \mathbf{P}_{wm} \\ \mathbf{P}_{vm}^T & \mathbf{P}_{wm}^T & \mathbf{P}_{mm} \end{bmatrix} \tag{12}$$

Let $\hat{\mathbf{x}}(k)$ and $\hat{\mathbf{P}}(k)$ be the estimated state vector and covariance. Then the filter recursively updates the mean $\hat{\mathbf{x}}^+(k)$ and covariance $\hat{\mathbf{P}}^+(k)$ of the state by combining the predicted mean $\hat{\mathbf{x}}^-(k)$ and covariance $\hat{\mathbf{P}}^-(k)$ with current noisy measurement $\mathbf{z}(k)$.

The AEKF SLAM consists of two steps; prediction and update step (Fig. 2) and the motion model of the robot in the prediction step is modified for considering the systematic parameters of the robot.

B. AEKF SLAM for Improving the Accuracy of the Feature Map

As we mentioned before, the AEKF SLAM is divided into two steps: prediction and update step.

1) Prediction Step: In the prediction step, when the robot pose at time k-1 is $\hat{\mathbf{x}}_v^+(k-1)=[\hat{x}_v^+(k-1), \hat{y}_v^+(k-1), \hat{\theta}_v^+(k-1)]^T$ and the control command of the robot between time k-1 and k is $\mathbf{u}(k)=[v_l, v_r]^T$ which has the translation velocities of left and right wheel. Then the state vector and covariance of EKF SLAM at time k are predicted as follows:

$$\hat{\mathbf{x}}^-(k) = \begin{bmatrix} \hat{\mathbf{x}}_v^-(k) \\ \hat{\mathbf{x}}_w^-(k) \\ \hat{\mathbf{x}}_m^-(k) \end{bmatrix} = \begin{bmatrix} \mathbf{f}\left(\hat{\mathbf{x}}_v^+(k-1),\ \hat{\mathbf{x}}_w^+(k-1), \mathbf{u}(k)\right) \\ \hat{\mathbf{x}}_w^+(k-1) \\ \hat{\mathbf{x}}_m^+(k-1) \end{bmatrix} \tag{13}$$

where, $\mathbf{f}(\cdot)$ is a motion model of the robot considering the systematic parameters.

$$\begin{aligned}\hat{\mathbf{x}}_v^-(k) &= \mathbf{f}\left(\hat{\mathbf{x}}_v^+(k-1),\ \hat{\mathbf{x}}_w^+(k-1), \mathbf{u}(k)\right) \\ &= \begin{bmatrix} \hat{x}_v^+(k-1) \\ \hat{y}_v^+(k-1) \\ \hat{\theta}_v^+(k-1) \end{bmatrix} + \begin{bmatrix} \frac{\left(\hat{\delta}_l v_l + \hat{\delta}_r v_r\right)}{2} \cdot dt \cdot \cos\left(\hat{\theta}_v^+(k-1)\right) \\ \frac{\left(\hat{\delta}_l v_l + \hat{\delta}_r v_r\right)}{2} \cdot dt \cdot \sin\left(\hat{\theta}_v^+(k-1)\right) \\ \frac{\left(\hat{\delta}_r v_r - \hat{\delta}_l v_l\right)}{\hat{\delta}_b b} \cdot dt \end{bmatrix}\end{aligned} \tag{14}$$

where b is the nominal value of left/right wheel diameter and the robot wheel base. The covariance of the state vector is also predicted as follows:

$$\begin{aligned}\hat{\mathbf{P}}^-(k) &= \nabla\mathbf{f}_{\mathbf{x}}\hat{\mathbf{P}}^+(k-1)\nabla\mathbf{f}_{\mathbf{x}}^T + \nabla\mathbf{f}_{\mathbf{u}}\mathbf{Q}_k\nabla\mathbf{f}_{\mathbf{u}}^T \\ &= \begin{bmatrix} \nabla_{\mathbf{x}_v}\mathbf{f} & \nabla_{\mathbf{x}_w}\mathbf{f} & 0 \\ 0 & \mathbf{I}_{\mathbf{x}_w} & 0 \\ 0 & 0 & \mathbf{I}_{\mathbf{x}_m} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{P}_{vv} & \mathbf{P}_{vw} & \mathbf{P}_{vm} \\ \mathbf{P}_{vw}^T & \mathbf{P}_{ww} & \mathbf{P}_{wm} \\ \mathbf{P}_{vm}^T & \mathbf{P}_{wm}^T & \mathbf{P}_{mm} \end{bmatrix} \cdot \begin{bmatrix} \nabla_{\mathbf{x}_v}\mathbf{f} & \nabla_{\mathbf{x}_w}\mathbf{f} & 0 \\ 0 & \mathbf{I}_{\mathbf{x}_w} & 0 \\ 0 & 0 & \mathbf{I}_{\mathbf{x}_m} \end{bmatrix}^T \\ &\quad + \begin{bmatrix} \nabla_{\mathbf{u}}\mathbf{f} \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} \sigma_{v_{left}}^2 & 0 \\ 0 & \sigma_{v_{right}}^2 \end{bmatrix} \begin{bmatrix} \nabla_{\mathbf{u}}\mathbf{f} \\ 0 \\ 0 \end{bmatrix}^T\end{aligned} \tag{15}$$

$$\begin{aligned}\nabla\mathbf{f}_{\mathbf{x}_v} &= \frac{\partial\mathbf{f}}{\partial\mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}_v^+(k-1)} \\ &= \begin{bmatrix} 1 & 0 & -\left(\hat{\delta}_l v_l + \hat{\delta}_r v_r\right)/2 \cdot dt \cdot \sin\left(\hat{\theta}_v^+(k-1)\right) \\ 0 & 1 & \left(\hat{\delta}_l v_l + \hat{\delta}_r v_r\right)/2 \cdot dt \cdot \cos\left(\hat{\theta}_v^+(k-1)\right) \\ 0 & 0 & 1 \end{bmatrix}\end{aligned} \tag{16}$$

$$\begin{aligned}\nabla\mathbf{f}_{\mathbf{u}} &= \frac{\partial\mathbf{f}}{\partial\mathbf{u}}\Big|_{\mathbf{u}=\mathbf{u}(k)} \\ &= \begin{bmatrix} \frac{\hat{\delta}_l}{2} \cdot dt \cdot \cos\left(\hat{\theta}_v^+(k-1)\right) & \frac{\hat{\delta}_r}{2} \cdot dt \cdot \cos\left(\hat{\theta}_v^+(k-1)\right) \\ \frac{\hat{\delta}_l}{2} \cdot dt \cdot \sin\left(\hat{\theta}_v^+(k-1)\right) & \frac{\hat{\delta}_r}{2} \cdot dt \cdot \sin\left(\hat{\theta}_v^+(k-1)\right) \\ \frac{-\hat{\delta}_l}{\hat{\delta}_b b} \cdot dt & \frac{\hat{\delta}_r}{\hat{\delta}_b b} \cdot dt \end{bmatrix}\end{aligned} \tag{17}$$

$$\nabla \mathbf{f}_{\mathbf{x}_w} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_w^+(k-1)} = \begin{bmatrix} \frac{v_l}{2}\cdot dt\cdot\cos\left(\hat{\theta}_v^+(k-1)\right) & \frac{v_r}{2}\cdot dt\cdot\cos\left(\hat{\theta}_v^+(k-1)\right) & 0 \\ \frac{v_l}{2}\cdot dt\cdot\sin\left(\hat{\theta}_v^+(k-1)\right) & \frac{v_r}{2}\cdot dt\cdot\sin\left(\hat{\theta}_v^+(k-1)\right) & 0 \\ \frac{-v_l}{\hat{\delta}_b b}\cdot dt & \frac{v_r}{\hat{\delta}_b b}\cdot dt & -\frac{\left(\hat{\delta}_r v_r - \hat{\delta}_l v_l\right)}{\hat{\delta}_b^2 b}\cdot dt \end{bmatrix} \quad (18)$$

where, $\nabla \mathbf{f}_{\mathbf{x}}$ and $\nabla \mathbf{f}_{\mathbf{u}}$ are jacobians of the robot motion model w.r.t the robot state vector and the control command, respectively and $\mathbf{Q}_k$ represents the error covariance matrix of the motion model. In this paper, the value of $[\sigma_{v_{left}}, \sigma_{v_{right}}]$ is set to $[0.3m/\text{sec},\ 0.3m/\text{sec}]$ and these are three times larger than actual standard deviation for generating a zero mean Gaussian noise.

2) Update Step: In update step, let $m_i = [\hat{x}_i, \hat{y}_i]^T$ be the position of an *i* th measured feature which already exists in the state vector $\hat{\mathbf{x}}(k)$ and this feature position can be represented as its range (ρ), and bearing (ϕ), based on the predicted robot pose, $\hat{\mathbf{x}}_v^-(k) = [\hat{x}_v^-(k), \hat{y}_v^-(k), \hat{\theta}_v^-(k)]^T$.

$$\hat{\mathbf{z}}_i = \mathbf{h}_i\left(\hat{\mathbf{x}}_v^-(k)\right) = \begin{bmatrix} \hat{\rho}_i \\ \hat{\phi}_i \end{bmatrix} = \begin{bmatrix} \sqrt{(\hat{x}_i - \hat{x}_v^-(k))^2 + (\hat{y}_i - \hat{y}_v^-(k))^2} \\ \tan^{-1}\left(\frac{\hat{y}_i - \hat{y}_v^-(k)}{\hat{x}_i - \hat{x}_v^-(k)}\right) - \hat{\theta}_v^-(k) \end{bmatrix} \quad (19)$$

This feature is measured from the sensor in terms of its range and bearing which are given as $\mathbf{z}_i = [\rho_i, \phi_i]^T$, then we can update mean and covariance of the robot state by following process:

$$\mathbf{S}_k = \nabla \mathbf{h} \hat{\mathbf{P}}^-(k) (\nabla \mathbf{h})^T + \mathbf{R}_k \quad (20)$$

$$\mathbf{K}_k = \hat{\mathbf{P}}^-(k) (\nabla \mathbf{h})^T (\mathbf{S}_k)^{-1} \quad (21)$$

$$\hat{\mathbf{x}}^+(k) = \hat{\mathbf{x}}^-(k) + \mathbf{K}_k (z_n - \hat{z}_n) \quad (22)$$

$$\hat{\mathbf{P}}^+(k) = (\mathbf{I} - \mathbf{K}_k \nabla \mathbf{h}) \hat{\mathbf{P}}^-(k) \quad (23)$$

$$\nabla \mathbf{h}_i = \left.\frac{\partial \mathbf{h}_i}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}^-(k)} = \begin{bmatrix} \frac{\partial \hat{\rho}_i}{\partial \hat{x}_v} & \frac{\partial \hat{\rho}_i}{\partial \hat{y}_v} & \frac{\partial \hat{\rho}_i}{\partial \hat{\theta}_v} & \frac{\partial \hat{\rho}_i}{\partial \hat{\delta}_l} & \frac{\partial \hat{\rho}_i}{\partial \hat{\delta}_r} & \frac{\partial \hat{\rho}_i}{\partial \hat{\delta}_b} & \frac{\partial \hat{\rho}_i}{\partial \hat{x}_1} & \frac{\partial \hat{\rho}_i}{\partial \hat{y}_1} & \frac{\partial \hat{\rho}_i}{\partial \hat{x}_2} & \cdots & \frac{\partial \hat{\rho}_i}{\partial \hat{x}_i} & \frac{\partial \hat{\rho}_i}{\partial \hat{y}_i} & \cdots & \frac{\partial \hat{\rho}_i}{\partial \hat{y}_n} \\ \frac{\partial \hat{\phi}_i}{\partial \hat{x}_v} & \frac{\partial \hat{\phi}_i}{\partial \hat{y}_v} & \frac{\partial \hat{\phi}_i}{\partial \hat{\theta}_v} & \frac{\partial \hat{\phi}_i}{\partial \hat{\delta}_l} & \frac{\partial \hat{\phi}_i}{\partial \hat{\delta}_r} & \frac{\partial \hat{\phi}_i}{\partial \hat{\delta}_b} & \frac{\partial \hat{\phi}_i}{\partial \hat{x}_1} & \frac{\partial \hat{\phi}_i}{\partial \hat{y}_1} & \frac{\partial \hat{\phi}_i}{\partial \hat{x}_2} & \cdots & \frac{\partial \hat{\phi}_i}{\partial \hat{x}_i} & \frac{\partial \hat{\phi}_i}{\partial \hat{y}_i} & \cdots & \frac{\partial \hat{\phi}_i}{\partial \hat{y}_n} \end{bmatrix} \quad (24)$$

$$= \frac{1}{\hat{\rho}_i} \begin{bmatrix} -D_x & -D_y & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & D_x & D_y & \cdots & 0 \\ \frac{D_y}{\hat{\rho}_i} & \frac{-D_x}{\hat{\rho}_i} & -\hat{\rho}_i & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \frac{-D_y}{\hat{\rho}_i} & \frac{D_x}{\hat{\rho}_i} & \cdots & 0 \end{bmatrix}$$

$$\mathbf{R}_k = \begin{bmatrix} \sigma_\rho^2 & 0 \\ 0 & \sigma_\phi^2 \end{bmatrix} \quad (25)$$

$$D_x = \hat{x}_i - \hat{x}_v^-(k),\ D_y = \hat{y}_i - \hat{y}_v^-(k) \quad (26)$$

where, $\nabla \mathbf{h}_i$ is the jacobian matrix of the observation model w.r.t the $\hat{\mathbf{x}}^-(k)$, $\mathbf{R}_k$ is measurement noise covariance matrix whose elements σ_ρ and σ_ϕ are set to $[0.1m,\ 1^\circ \cdot \pi/180 rad]$, respectively, and $\mathbf{K}_k$ is a Kalman gain. Then the mean and covariance of the robot state is updated by (22) and (23), respectively.

3) State Augmentation: Let *n* be the number of features in the state vector $\hat{\mathbf{x}}^+(k)$. If the *n*+1 th feature, $\mathbf{z}_{n+1}$, which does not exist in the state vector, is measured in terms of its range (ρ) and bearing (ϕ) in the robot centered frame, this new feature is added to the state vector through following process. Let $\mathbf{z}_{n+1}=[\rho_{n+1},\ \phi_{n+1}]^T$ be the position of new feature in the robot centered coordinate, then $\mathbf{z}_{n+1}$ is transformed to the world coordinate frame as follows:

$$\mathbf{x}_{m_{n+1}} = g(\hat{\mathbf{x}}_v^+(k), \mathbf{z}_{n+1}) = \begin{bmatrix} \hat{x}_v^+(k) + \rho_{n+1}\cos\left(\hat{\theta}_v^+ + \phi_{n+1}\right) \\ \hat{y}_v^+(k) + \rho_{n+1}\sin\left(\hat{\theta}_v^+ + \phi_{n+1}\right) \end{bmatrix} \quad (27)$$

Suppose that $\mathbf{x}^*$ is the state vector of the robot which includes a position of newly measured feature,

$$\hat{\mathbf{x}}^*(k) = \begin{bmatrix} \hat{\mathbf{x}}_v^+(k) \\ \hat{\mathbf{x}}_{\mathbf{w}}^+ \\ \hat{\mathbf{x}}_{\mathbf{m}}^+ \\ g(\hat{\mathbf{x}}_v^+(k), z_{n+1}) \end{bmatrix} \quad (28)$$

and the covariance of the state vector $\hat{\mathbf{x}}^*(k)$ is computed as follows:

$$\hat{\mathbf{P}}^*(k) = \nabla \mathbf{Y}_{\mathbf{x},\mathbf{z}} \begin{bmatrix} \hat{\mathbf{P}}^+(k) & 0 \\ 0 & \mathbf{R}_k \end{bmatrix} \nabla \mathbf{Y}_{\mathbf{x},\mathbf{z}}^T \quad (29)$$

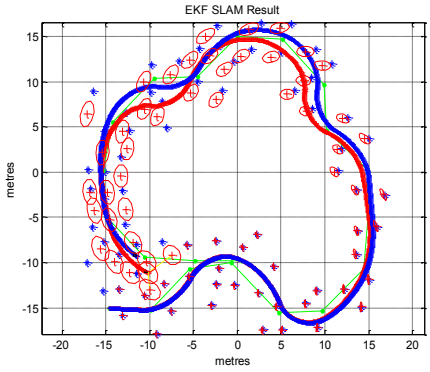
$$= \begin{bmatrix} \mathbf{P}_{vv} & \mathbf{P}_{vw} & \mathbf{P}_{vm} & \mathbf{P}_{vv}\mathbf{G}_v^T \\ \mathbf{P}_{vw}^T & \mathbf{P}_{ww} & \mathbf{P}_{wm} & \mathbf{P}_{vw}^T\mathbf{G}_v^T \\ \mathbf{P}_{vm}^T & \mathbf{P}_{wm}^T & \mathbf{P}_{mm} & \mathbf{P}_{vm}^T\mathbf{G}_v^T \\ \mathbf{G}_v\mathbf{P}_{vv} & \mathbf{G}_v\mathbf{P}_{vw} & \mathbf{G}_v\mathbf{P}_{vm} & \mathbf{G}_v\mathbf{P}_{vv}\mathbf{G}_v^T + \mathbf{G}_{\mathbf{z}}\mathbf{R}_k\mathbf{G}_{\mathbf{z}} \end{bmatrix}$$

$$\nabla \mathbf{Y}_{\mathbf{x},\mathbf{z}} = \begin{bmatrix} \mathbf{I}_v & 0 & 0 & 0 \\ 0 & \mathbf{I}_w & 0 & 0 \\ 0 & 0 & \mathbf{I}_m & 0 \\ \mathbf{G}_v & 0 & 0 & \mathbf{G}_{\mathbf{z}} \end{bmatrix} \quad (30)$$

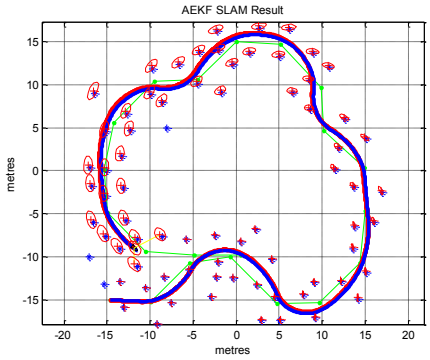
$$\nabla \mathbf{G}_v = \left.\frac{\partial g}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_v^+(k)},\ \nabla \mathbf{G}_{\mathbf{z}} = \left.\frac{\partial g}{\partial \mathbf{z}}\right|_{\mathbf{z}=\mathbf{z}_{n+1}} \quad (31)$$

IV. Experimental Results

The EKF-SLAM with systematic parameter estimation has been evaluated both simulated and real indoor environment. For the simulation environment, we used the Matlab simulation developed by Bailey et al [21]. In fact, the packages served as an excellent platform for learning and analyzing existing Kalman filter to compare the performance of the proposed SLAM method against. With the vehicle



(a) A feature map and trajectory from the conventional EKF SLAM. (Blue solid line: ground truth trajectory, Red solid line: EKF SLAM trajectory, Green thin line: connects the way points which are used for control the robot in the simulator. Blue star: ground truth feature position, Red cross and ellipse: feature positions and their uncertainty ellipse from EKF SLAM



(b) A feature map and trajectory from the AEKF SLAM.

Fig. 3. The feature maps and trajectories from conventional EKF and AEKF SLAM in the simulation environment with biased noise.

model used in Bailey's simulation program being the Ackerman model [22], we had to modify it to a differential wheel model [23] and also set the maximum detection range of the sensor to 4m. An update step is carried out after eight consecutive prediction steps and this helps in reducing the computational burden of the SLAM algorithm.

The real experiment has been carried out on well-known data sets, named as Freiburg building [24]. In the Freiburg building data set, the size of map is 40×18m and odometry and range measurements were also recorded by a Pioneer and a SICK laser scanner. In the middle of the map, a long corridor exists and this is a good for testing the odometric error compensating performance of the EKF SLAM.

A. Simulation Result

The simulation experiment was carried out under biased noise which represented the systematic error of the robot. The simulation environment was a cyclic path spanning a space of approximately 15x15m. The robot starts from the lower left section of Fig. 3 and eventually returns to the start position. To simulate the performance of EKF SLAM working under the biased noise, we deliberately added two types of error of the robot's left and right wheel velocities. The first one was a zero mean Gaussian noise with σ_l = 0.1m/s and σ_r = 0.1m/s

TABLE I
RMS ERROR OF THE EKF/AEKF SLAM JUST BEFORE LOOP CLOSING IS DETECTED

	EKF SLAM	AEKF-SLAM
RMS error (m)	2.223	0.4

and the second was additional 1% and 2% of the left and right wheel velocities, respectively. Thus, let v_l^* and v_r^* be the actual translation velocity of left and right wheel, respectively, the nominal velocities v_l and v_l which were used to compute the odometry pose of the robot was calculated as follows:

$$v_l = v_l^* + \Delta v_{l,bias} + \Delta v_{l,random}, \quad v_r = v_r^* + \Delta v_{r,bias} + \Delta v_{r,random} \tag{32}$$

$$\begin{bmatrix} \Delta v_{l,random} \\ \Delta v_{r,random} \end{bmatrix} = \begin{bmatrix} v_l \cdot N(0,\sigma_l^2) \\ v_r \cdot N(0,\sigma_r^2) \end{bmatrix} \tag{33}$$

$$\begin{bmatrix} \Delta v_{l,bias} \\ \Delta v_{r,bias} \end{bmatrix} = \begin{bmatrix} 0.01 \cdot v_l \\ 0.02 \cdot v_r \end{bmatrix} \tag{34}$$

where, $N(0,\sigma^2)$ means a Gaussian noise with zero means and σ^2 variance. In addition to this, we subtracted 1% noise to the effective wheel base of the robot, so nominal effective wheel base of the robot is

$$b = b^* - 0.01 \cdot b^* \tag{35}$$

where b^* is actual effective wheel base of the robot and it is set to 0.4m in this paper.

The feature maps were generated using the standard EKF SLAM and the proposed AEKF SLAM methods (Fig. 3). The RMS error in AEKF SLAM just before loop closing was much smaller than the one in the EKF SLAM (Table I). Because the ground-truth pose of the robot was available in the simulator, we could test the filter consistency by using the Normalized Estimation Error Squared(*NEES*):

$$\varepsilon_k = (\mathbf{x}_k - \hat{\mathbf{x}}_k)^T \mathbf{P}_k^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_k) \tag{36}$$

Multiple Monte Carlo runs and computing the average *NEES* was carried out for evaluating the consistency of EKF/AEKF. Given *N* runs, the average *NEES* computed as follow:

$$\hat{\varepsilon}_k = \frac{1}{N} \sum_{i=1}^{N} \varepsilon_{i_k} \tag{37}$$

In this research, 30 Monte Carlo simulations were performed with the two sided 95% *probability concentration region* and it is bounded by the interval [2.19, 3.93]. If $\hat{\varepsilon}_k$ rises significantly higher than the upper bound, the filter is optimistic. If it tends below the lower bound, the filter is conservative. Optimistic means that estimated covariance of the robot is smaller than the actual one [25].

The EKF became optimistic after 500 steps (Fig. 4). This

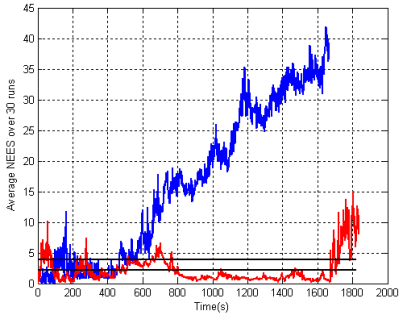


Fig. 4. Average NEES of the robot pose state over 30 Monte Carlo runs. The horizontal dashed line mark the 95% probability concentration region for a 3-dimensional state vector. Blue and red solid lines indicate the average NEES of EKF and AEKF SLAM, respectively.

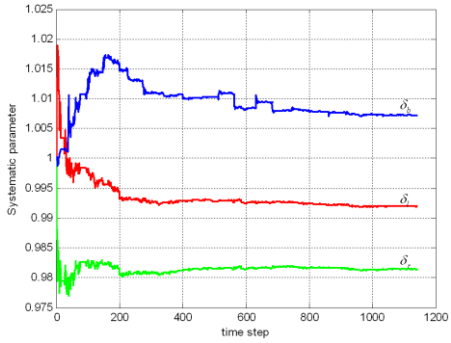


Fig. 5. Convergence of the systematic error parameters according to the time step Red, green and blue solid lines represent the systematic error parameters δ_l, δ_r and δ_b, respectively.

means that the estimated covariance of EKF SLAM does not contains the ground truth pose of the robot and EKF fails to compensate the odometric error accurately. However AEKF did not become optimistic and remained consistent.

In AEKF SLAM, the systematic parameter of the robot, $\mathbf{x}_w(k)=[\delta_l, \delta_r, \delta_b]^T$, should be $[0.99, 0.98, 1.01]^T$, because we appended 1% and 2% noise to the translation velocity of the left and right wheel in (34) and 1% noise to effective wheel base in (). Finally estimated systematic parameters via systematic parameter estimation in AEKF frame and estimated value is $[\delta_l, \delta_r, \delta_b]^T = [0.9920, 0.9814, 1.0072]^T$ (Fig. 5).

B. Real Experiment in the Corridor Environment

Because a long corridor exists in the Freiburg building and the corridor a rich environment to extract the line segment (Fig. 6), we extracted the line feature from the laser scanner measurements data. The systematic error of the robot skews the robot' odometric trajectory to right, even when the robot actually moves straight (Fig. 7). Then, we generated the line based feature map using EKF and AEKF SLAM (Fig. 8). The line feature from the range measurements data, $l_i = [\rho_i, \phi_i]^T$, is parameterized by its distance $\rho_i \geq 0$ from the origin and the direction $\phi_i \in (-\pi, \pi]$ of the normal passing through the origin and we adopt the measurement equation of the line feature for applying EKF SLAM from Garulli's method [26].

To check the performance of the odometry error compensation in the real environment, we selected two walls

Fig. 6. University of Freiburg, building 079 [23].

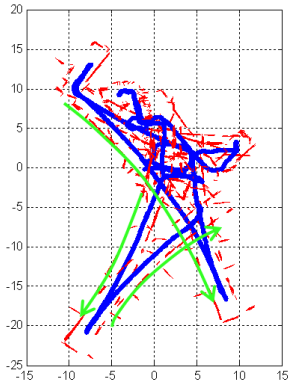


Fig. 7. Robot trajectory and feature map of Freiburg data set based on the odometry. Green solid arrows indicate that the systematic error makes the robot' odometric trajectory lean toward right, even when the robot actually moves straight.

at both ends of the corridor, l_{start} and l_{end} (Fig. 8a). If the SLAM algorithm compensates for the odometric error perfectly, the parameters of these two lines should be exactly same. Let the D_l be the Euclidean distance between two lines l_{start} and l_{end} .

$$D_l = \sqrt{(\rho_{start} - \rho_{end})^2 + (\phi_{start} - \phi_{end})^2} \tag{38}$$

D_l from AEKF-SLAM was much smaller than the one from the EKF SLAM (Table II). This result also shows that the AEKF SLAM can compensate the odometric error more exactly and generate more accurate feature map than the EKF SLAM in the real environment.

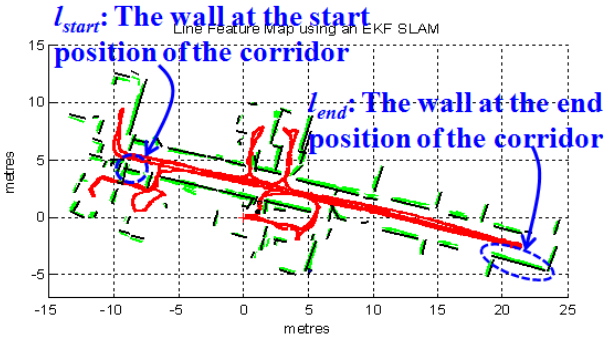
TABLE II
THE LINE PARAMETERS AT BOTH ENDS OF THE CORRIDOR IN FREIBURG DATA SET

	$l_{start} = [\rho, \phi]$	$l_{end} = [\rho, \phi]$	D_l
EKF SLAM	[1.7359, 1.3063]	[2.3590, 1.2768]	0.6058
AEKF-SLAM	[1,7407, 1.3740]	[1.6519, 1.3888]	0.0900

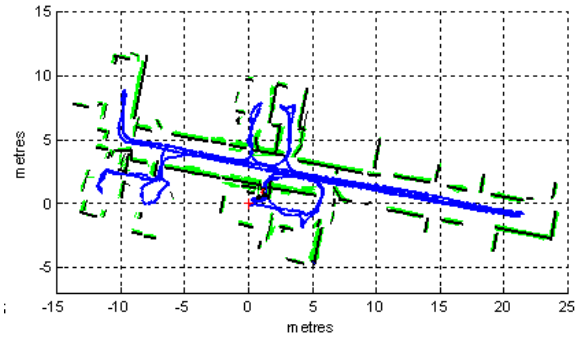
V. CONCLUSION

In this paper, we have proposed the AEKF SLAM for increasing the accuracy of EKF SLAM. In AEKF SLAM, the systematic parameters of the robot are estimated and these systematic parameters of the robot are used for compensating the odometric error of the robot in the prediction step. The online estimation of systematic parameters was implemented in the EKF SLAM process by augmenting the systematic parameters as the elements of the EKF SLAM.

So the proposed AEKF SLAM was able to generate a



(a) The line based feature map from EKF SLAM



(b) The line based feature map from AEKF SLAM

Fig. 8. Robot trajectory and feature map of Freiburg data set based on the trajectory from the EKF and AEKF SLAM. Red/Blue solid line is the robot trajectory from the SLAM process, green solid lines represent the line features which are generated from the Freiburg data set and black solid lines mean that the position of walls computed from the AEKF SLAM

feature map while estimating a proper set of systematic parameters. For this reason, we could model the systematic error parameters of the robot and reduce the biased motion noise. Comparing the experimental results in simulation/real scenarios show that compensating an odometry error of the robot by using the systematic parameters of the robot. Estimating systematic parameters enhances the accuracy of the EKF SLAM, thus the proposed AEKF SLAM shows a superior accuracy in the feature map than EKF SLAM.

REFERENCES

[1] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte and M. Csorba, "A Solution to the Simultaneous Localization and Map Building (SLAM) Problem," *IEEE Trans. Robot. Autom*, vol. 17, no. 3, pp. 229-241, 2001.

[2] P. Newman, J. Leonard, J. D. Tardòs, J. Neira, "Explore and Return: Experimental Validation of Real-Time Concurrent Mapping and Localization," *IEEE. Int. conf. Robotics and Automation*, pp. 1802-1809, 2002.

[3] Y. H. Choi, T. K. Lee and S. Y Oh, "A line feature based SLAM with low grade range sensors using grade sensors using geometric constraints and active exploration for mobile robot," *Autonomous Robots*, vol. 24, no. 1, pp 13-27, 2008.

[4] S. H. Kim and S. Y. Oh, "SLAM in Indoor Environments using Omni-directional Vertical and Horizontal Line Features," *Journal of Intelligent Robotic Systems*, vol. 51, pp. 31-43, 2008.

[5] J. Guivant and E. Nebot, "Optimization of the simultaneous localization and map-building algorithm for real-time implementation," *IEEE Trans. Robot. Autom*., vol. 17, no. 3, pp. 242–257, 2001.

[6] P. Newman and J. Leonard, "Pure range-only subsea SLAM," *in Proc. IEEE Int. Conf. Robotics and Automation*, pp. 1921–1926, 2003.

[7] A. J. Davison, "Real-time simultaneous localization and mapping with a single camera," *in Proc. the 9th IEEE Int. Conf. on Computer Vision*, pp. 1403-1410, 2003.

[8] S. Se, D. Lowe and J. Little, "Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks," *Int. J. Robot. Res.*, vol. 21, no. 8, pp. 735-758, 2002.

[9] W. Y. Jeong and K. M. Lee, "CV-SLAM: a New Ceiling Vision-based SLAM Technique," *Int. Conf. Intelligent Robots and Systems*, pp. 3195-3200, 2005.

[10] K. Murphy, "Bayesian map learning in dynamic environments," *in Proc. Conf. Neural Inf. Process. Syst.*, pp. 1015–1021, 1999.

[11] A. Doucet, J. Freitas, K. Murphy and S. Russel, "Rao–Blackwellized particle filtering for dynamic Bayesian networks," *in Proc. Conf. Uncertain. Artif. Intell.*, pp. 176–183, 2000.

[12] M. Montemerlo, S. Thrun, D. Koller and B. Wegbreit, "Fast-SLAM: A factored solution to the simultaneous localization and mapping problem," *in Proc. AAAI Nat. Conf. Artif. Intell.*, pp. 593–598, 2002.

[13] M. Montemerlo, S. Thrun, D. Koller and B. Wegbreit, "Fast-SLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," *in Proc. Int. Joint Conf. Artif. Intell.*, pp. 1151–1156, 2003.

[14] J. Borenstein and L. Feng, "Measurement and Correction of Systematic Odometry Errors in Mobile Robots," *IEEE. Trans. Robotics and Automation*, vol. 12, no. 6, pp. 869-880, 1996.

[15] N. Doh, H. Choset and W. K. Chung, "Accurate relative localization using odometry," in *Proc. IEEE. Int. Conf. Robotics and Automation,* pp. 1606-1612, 2003.

[16] G. Antonelli, S. Chiaverini and G. Fusco, "A systematic calibration method for odometry of mobile robots based on the least-squares technique: Theory and experimental validation," *IEEE. Trans. Robotics and Automation*, vol. 21, no. 5, pp. 994-1004, 2005.

[17] T. D. Larsen, M. Bak, N. A. Andersen and O. Ravn, "Location estimation for autonomously guided vehicle using an augmented Kalman filter to autocalibrate the odometry," *in Proc. FUSION98 Spie Conference*, 1998.

[18] A. Martinelli, N. Tomatis and R. Siegwart, "Simultaneous localization and odometry self calibration for mobile robot," *Autonomous Robots*, vol. 22, pp. 75-85, 2007.

[19] Y. M. Yun, B. J. Park and W. K. Chung, "Odometry Calibration using Home Positioning Function for Mobile Robot," in *Proc. IEEE. Int. Conf. Robotics and Automation,* pp. 2116-2121, 2008.

[20] K. S. Chong and L. Kleeman, "Accurate odometry and error modeling for a mobile robot," in *Proc. IEEE. Int. Conf. Robotics and Automation,* pp. 2783-2788, 1997.

[21] T. Bailey, J. Neito, J. Guivant, M. Stevens and E. Nebot, "Consistency of the EKF-SLAM Algorithm," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 3562-3568, 2006.

[22] M. A. Sotelo, "Lateral control strategy for autonomous steering of Ackerman-like vehicles," *Robotics and Autonomous Systems*, vol. 24, pp. 223-233, 2003.

[23] G. W. Lucas, "A Tutorial and Elementary Trajectory Model for the Differential Steering System of Robot Wheel Actuators," Available: http://rossum.sourceforge.net/papers/DiffSteer/DiffSteer.html.

[24] M. Batalin, "The Robotics Data Set Repository (Radish)," http://radish.sourceforge.net.

[25] T. Bailey, J. Neito, J. Guivant, M. Stevens and E. Nebot, "Consistency of the EKF-SLAM Algorithm," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 3562-3568, 2006.

[26] A. Garulli, et al. "Mobile robot SLAM for line-based environment representation," *IEEE. Conf. Decision and Control and European Control Conference (ICDC)*, pp. 2041-2046, 2005.