ORIGINAL ARTICLE

Implementation of Odometry with EKF in Hector SLAM Methods

Ming-Yi Ju[1], Yu-Jen Chen[2], and Wei-Cheng Jiang[3, *]
[1]*National University of Tainan*
[2]*National Chung Cheng University*
[3]*National Sun Yat-sen University*
(**Received** 9 August 2017; **Accepted** 3 October 2017; **Published online** 1 March 2018)
*Corresponding author: enjoysea0605@gmail.com
DOI: 10.5875/ausmt.v8i1.1558

Abstract: Map building for plain spatial soundings, such as a long and straight corridor in simultaneous localization and mapping (SLAM) is a challenging problem because of lacks of distinguishable landmarks. Such an environment is highly possible to induce erroneous mapping results, such as alias problems. This paper presents a scan matching algorithm with odometer prediction using Extended Kalman Filter (EKF) and an optimal path planning based on regression subgoals. The scan matching process can relax the problems of local minima by means of an effective correction in the odometrical information. By iterating odometrical corrections in each step of running motion model, the matching result can be better than one only believes in individual information from scanning or odometry. Meanwhile, an optimal path planning utilizing an A^* algorithm with a regression method is introduced to ensure a mobile robot be able to move elaborately around the corner and speed up along a straight line. Experiments in an indoor environment have been conducted to verify the effectiveness and validation of the proposed techniques.

Keywords: SLAM; Autonomous robot; Extended Kalman Filter; Path planning; Regression

Introduction

In today's world robots are getting more important in human life, not just robot arms for the manufacturing propose. Autonomous robots have been applied to industry, academic applications, and so on. The costs and the sizes of sensors and microprocessor reduced tremendously recently make the autonomous robots affordable for office and home. Indoor navigation techniques have experienced an amount of interest for research proposes over the last three decades. Recent research is working on utilizing the existing algorithm to find an effective way for autonomous robots that will make robots into our society without harm and people can work with robots, live with robots. The majority of recent study implementation and conceptual theory of autonomous robot navigation are in the field of tight-budget projects, such as small-scale mapping with the high-end device or large-scale SLAM with many closed loops. In the recent years, low-cost laser rangefinders have been used commonly among autonomous robot applications due to its price. The limitations of high-end laser rangefinders are energy consumption and heat dissipation of the device. The low-cost laser rangefinders have the ability to trace terrains in the contiguous area and consume much lower power than the high-end device, makes it really suited for small area inspection, mapping proposes.

Robot navigation in a dynamical environment can be classified into two main categories; one is how to create an accurate map of the environment's characteristics; the other is how navigation generates a safe path with the area-correct map and dynamic obstacles. Many types of research on SLAM and navigation algorithms have been developed to solve common scenarios in the environment [1-3]. These algorithms should guarantee their performance with cheap sensors and can be easily obtained from the navigation program [4]. Studies on the SLAM problem have been done in recent years, typical indoor environment using Rao-Blackwellized particle filters (RBPFs) will be a high-performance solution, and has been written into open source software such as Gmapping [5]. The other renowned technique is the Bayes filter method, which iteratively calculates posterior distributions of robots' poses by identification of landmarks, and a variation of this method come as Kalman Filters (KFs) [6]. In recent decades, there is a rich number of studies on KF-based algorithms being implemented in the literature, such as the Extended Kalman Filters (EKFs) and RBPFs. Lately, some popular methods make use of high scanning rates of rangefinders, which heavily rely on consecutive scan matching of sensor data and combine with multiresolution occupancy grid maps (Hector SLAM) [7] or dynamic and approximate likelihood fields for measurement [8]. Hector SLAM is adopted in this paper for the final result of position prediction. An advanced method likes sensor fusion in complementary characteristics of Laser Range Finder (LRF) that can map under reduced visibility conditions, e.g. particles of smoke [9]. In addition, classifying graph-based algorithms [10][11] which use a robust function that generalizes classification and discard irrelevant measurements also is an efficiency solution to maintain large-scale maps. On the other hand, calculating shortest paths with the existed maps is the core partition in the navigation process. Numerous classical graph search algorithms have been developed and implement in a real-life application, such as Dijkstra's algorithm [12], and A* search algorithm [13]. A* algorithm searches for the minimization of a cost function to generate an adequate path, which ensures the optimality of the produced path, but path evaluate from lowest cost will probably not be the path human want. Dijkstra is a special case for A* algorithm with the heuristics set to zero. Both algorithms return an optimal path, but the redundant path pose (node) in the map will be more as the resolution of the map getting higher. Regression reduction method proposed by this paper could be the solution to this problem. Both for SLAM and navigation, the most important problem, localization, is the process of estimating the next pose of the robot with regard to a given map of the environment. Compare with outdoor environments, localization in the indoor environment is far more complicated. Roughly speaking, the main task of a robot in an indoor navigation or mapping is able to sense how it moves when it receives a motion command (odometer) and the information of environment around it (vision or rangefinder). Bring together sensory observations as the base information, the robot can find out exactly current position by mean of maximum likelihood of observations given already exist map [14][15]. Present work done by this paper is on the navigation of differential wheeled odometer robot in an indoor environment without the use of high-end rangefinders or other positioning sensors. Optimal path planning method, which utilizes multi-regression line as the alignment of subgoal pose, will be used to reduce the complexity of all navigation poses in the route. Full navigation and mapping system is implemented and verified in experiments on a differential wheeled robot. This paper is divided into five sections as follows. In Section II, the background knowledge including extended Kalman filter and scan matching method is briefly introduced; In Section III, detailed methods of proposed navigation design for the differential wheeled robot are explained. In Section IV, the implementation of the proposed navigation system and experimental results of drawing a corridor environment are reported to verify the proposed method. Finally, Section V concludes the paper.

Ming-Yi Ju (M'10) received the B.S. degree in Electrical Engineering from Tatung University, Taipei, Taiwan, in 1993, and the Ph.D. degree from the Electrical Engineering Department, National Chung Cheng University, Chia-Yi, Taiwan, in 2001. He became an assistant professor of the Department of Information Engineering, I-Shou University, Kaohsiung, Taiwan, from August 2003 to July 2006. Since August 2006, he has been with the Department of Computer Science and Information Engineering, National University of Tainan, Tainan, Taiwan, where he is currently an Associate Professor. His research interests include intelligent robotic systems, robot vision, soft computing, and optimization.

Yu-Jen Chen received the B.S. degree in electrical engineering from the Tatung Institute of Technology, Taipei, Taiwan, in 1994, and the M.S. and Ph.D. degrees in electrical engineering from National Chung Cheng University, Chia-Yi, Taiwan, in 1997 and 2009, respectively. From 2004 to 2009, he was an Adjunct Lecturer with the Center for General Education, National Chung Cheng University. Since 2010, he has been an Assistant Professor with the Electrical Engineering Department, National Chung Cheng University. His current research interests include machine learning, robotics, neural networks, and embedded systems.

Wei-Cheng Jiang received the B.S. degree in Computer Science and information Engineering and M.S. degree in Electro-Optical and Materials Science from National Formosa University, Yunlin, Taiwan, in 2007 and 2009, respectively. He received the Ph.D. degree in Electric Engineering from National Chung Cheng University, Chiayi, Taiwan, in 2013. He was a posdoc with the Electrical Engineering Department at National Sun Yat-sen University, Kaohsiung, Taiwan. He is currently a visiting scholar in the Department of Electrical and Systems Engineering, Washington University in St. Louis, Saint Louis, MO, USA. His research interests include machine learning, neural networks, and intelligent control.

Background

Hector SLAM

Hector SLAM algorithm is selected as the framework for this work. This is primarily because this SLAM algorithm is suited to the condition that odometer information cannot be acquired, or error of the odometer is over the tolerance. A long-range rangefinder with high scanning rates is required in this method and the world coordinate setting with the z-axis pointing upwards and the x-axis pointing into the forward direction of the differential wheeled robot at startup. The map available is the occupancy grid maps, which has discrete nature limitation of the precision that is not directly accessible and will be achieved by computation of interpolated values or derivatives. Bilinear filtering is employed for interpolating sub-grid cell value to estimating occupancy probabilities and derivatives. In this way, the grid map discrete value surface is continuous in any single point of the map. Given a point in a continuous map, P_m , the occupancy value, $M(P_m)$, the gradient will be in the form:

$$\nabla M(P_m) = \left(\frac{\partial M(P_m)}{\partial x}, \frac{\partial M(P_m)}{\partial y} \right). \tag{1}$$

If the occupancy value is approximated by using the bilinear method, linear interpolation with closest integer coordinates $P_{00}, P_{10}, P_{10}, P_{11}$ can be represented as:

$$\begin{aligned} M(P_m) \approx & \frac{y-y_0}{y_1-y_0}\left(\frac{x-x_0}{x_1-x_0} M(P_{11}) + \frac{x_1-x}{x_1-x_0} M(P_{01}) \right) + \\ & \frac{y_1-y}{y_1-y_0}\left(\frac{x-x_0}{x_1-x_0} M(P_{10}) + \frac{x_1-x}{x_1-x_0} M(P_{00}) \right) \end{aligned} \tag{2}$$

And derivatives of the map in a specific point can be shown as:

$$\begin{aligned} \frac{\partial M(P_m)}{\partial x} \approx & \frac{y-y_0}{y_1-y_0}\left(M(P_{11})+M(P_{01})\right)+ \\ & \frac{y_1-y}{y_1-y_0}\left(M(P_{11})+M(P_{01})\right) \\ \frac{\partial M(P_m)}{\partial y} \approx & \frac{x-x_0}{x_1-x_0}\left(M(P_{11})+M(P_{01})\right)+ \\ & \frac{x_1-x}{x_1-x_0}\left(M(P_{11})+M(P_{01})\right) \end{aligned} \tag{3}$$

Scan matching is based on optimization of the alignment of beam endpoints with the map obtained so far. Gauss-Newton approach is utilized to predict the next pose without search data association between the end-point. Begin with a start estimate pose, $\xi = (p_x, p_y, \psi)^T$, scan matching aims to minimize the error of the occupancy of end-point $M(S_i(\xi))$ and map (value 1, means the obstacle exist in the map), and can be written as follows,

$$\xi^* = \arg\min_{\xi} \sum_{i=1}^{n} [1 - M(S_i(\xi))]^2 \tag{4}$$

where $S_i(\xi)$ is the transform of the end-point scan received in robot frame to the world frame,

$$S_i(\xi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} s_{i,x} \\ s_{i,y} \end{bmatrix} + \begin{bmatrix} p_x \\ p_y \end{bmatrix} \tag{5}$$

Gauss–Newton algorithm is used to solve non-linear least squares problems, target function r_i can be defined as:

$$r_i = 1 - M(S_i(\xi)) \tag{6}$$

The recurrence relation for Newton's method for minimizing a target function can be represented as follows:

$$\xi_t = \xi_{t-1} - H^{-1} G \tag{7}$$

Assume the robot position has a very little movement $\Delta\xi$ which is small enough to be ignored, the gradient vector G of target function can be written as:

$$\begin{aligned} G &= \sum_{i=1}^{n} r_i \frac{\partial r_i}{\partial \xi} \\ &= \sum_{i=1}^{n} \left[\nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi} \right]^T [1 - M(S_i(\xi))] \end{aligned} \tag{8}$$

And H denotes the Hessian matrix, obtained by ignoring the second-order derivative terms.

$$\begin{aligned} H &= \sum_{i=1}^{n} \frac{\partial r_i}{\partial \xi} \frac{\partial r_i}{\partial \xi} \\ &= \sum_{i=1}^{n} \left[\nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi} \right]^T \left[\nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi} \right] \end{aligned} \tag{9}$$

where the derivative of $S_i(\xi)$ can be shown by matrix:

$$\frac{\partial S_i(\xi)}{\partial \xi} = \begin{bmatrix} 1 & 0 & -\sin(\psi)s_{i,x} - \cos(\psi)s_{i,y} \\ 0 & 1 & \cos(\psi)s_{i,x} - \sin(\psi)s_{i,y} \end{bmatrix} \tag{10}$$

And now, calculate a step $\Delta\xi$ towards the minimum,

$$\Delta\xi = H^{-1} \sum_{i=1}^{n} \left[\nabla M(S_i(\xi)) \frac{\partial S_i(\xi)}{\partial \xi} \right]^T [1 - M(S_i(\xi))] \tag{11}$$

The non-smooth linear approximation method in Hector SLAM with a point coordinate in a map relies on the scan matching information at each end-point for proper convergence and suffers from strong local minima in the long corridor environment. The pseudocode of scan matching can be summarized as follows:

1. **Hector_Scan_Matching** $(\xi_{t-1}, s_{i,x}, s_{i,y}, M)$
2. $S_i(\xi_{t-1}) = \begin{bmatrix} \cos(\psi_{t-1}) & -\sin(\psi_{t-1}) \\ \sin(\psi_{t-1}) & \cos(\psi_{t-1}) \end{bmatrix} \begin{bmatrix} s_{i,x} \\ s_{i,y} \end{bmatrix} + \xi_{t-1}$
3. $G = \sum_{i=1}^{n} \left[\nabla M(S_i(\xi_{t-1})) \frac{\partial S_i(\xi_{t-1})}{\partial \xi} \right]^T [1 - M(S_i(\xi_{t-1}))]$

4.	$H=\sum_{i=1}^{n}\left[\nabla M(S_i(\xi_{t-1}))\frac{\partial S_i(\xi_{t-1})}{\partial \xi}\right]^T\left[\nabla M(S_i(\xi_{t-1}))\frac{\partial S_i(\xi_{t-1})}{\partial \xi}\right]$
5.	$\Delta\xi=H^{-1}G$
6.	$\xi_t=\xi_{t-1}+\Delta\xi$
7.	**return** ξ_t

Extended Kalman Filter

The Markov localization model represents robot position using a probability density function, which is very general but lacks efficiency. Sensor fusion problem is the solution to robust localization, not just relies on the probability density curve or motion model. Nowadays robots usually come with a lot of sensors on it, each sensor reading provides a portion to minimize the probability in the current robot position; however, each sensor is suffering from noise under certain condition. Optimal localization should take as much sensor readings into account as possible, but also carefully handle the information provided by all of these sensors. In Kalman filter, the next state probability (Motion Model) $p(x_t|u_t, x_{t-1})$ must be a linear function with Gaussian noise, and can be expressed by following equation:

$$x_t=A_t x_{t-1}+B_t u_t+\delta_t \tag{12}$$

where x_t and x_{t-1} are state vectors, and u_t is control command at time t. A_t is a matrix implied that how the state evaluate from previous the state without controls or noise. B_t is a matrix of corresponding control changes map to the next state. δ_t is the random variable represents prediction noise, zero mean Gaussian noise.

The measurement probability (Observation Model) $p(x_t|u_t, x_{t-1})$ is given as following:

$$z_t=C_t x_t+\sigma_t \tag{13}$$

where C_t is a matrix of relation of observation and state. When robot is in the state x_t, the observation will be received ideally.

Extended Kalman Filter (EKF) can overcome the linearity assumption of Kalman Filter (KF) that both the motion model and sensor model are linear Gaussian. The key idea of EKF is called linearization. The probability of next state and measurement probability are governed by nonlinear functions g and h,

$$x_t=g(u_t,x_{t-1},)+\delta_t \tag{14}$$

$$z_t=h(x_t)+\sigma_t \tag{15}$$

EKF calculates an approximation to the true belief, which is represented by a Gaussian. The belief state $bel(x_t)$ at time t is represented by a mean μ_t and a covariance Σ_t. EKF algorithm is similar to the Kalman filter algorithm and is stated in the following table:

1.	**Extende_Kalman_Filter** $(u_{t-1},\Sigma_{t-1},u_t,z_t)$:
2.	$\bar{u}_t=g(u_{t-1},u_t)$
3.	$\bar{\Sigma}_t=A_t\Sigma_{t-1}A_t^T+R_t$
4.	$K_t=\bar{\Sigma}_t C_t^T\left(C_t\Sigma_t C_t^T+Q_t\right)^{-1}$
5.	$\mu_t=\bar{\mu}_t+K_t(z_t-h(\bar{\mu}_t))$
6.	$\Sigma_t=(I-K_tC_t)\bar{\Sigma}_t$
7.	**return** μ_t,Σ_t

A Search Algorithm*

A* is a graph search algorithm that finds the minimum cost path. Start from a given initial node, A* search the shortest (low-cost) path to the goal node. This algorithm can be primarily real-time applied in computer games or robot navigation task to find the shortest path. It potentially needs to search through a huge amount of the data node. A* search algorithm uses an efficient heuristic function to reduce search space. If heuristic function equals to zero, the algorithm becomes Dijkstra's pathfinding algorithm; if the heuristic function has a high value, the algorithm becomes Breadth-First-Search (BFS). Thus heuristic value is the key idea of the behavior of A*. In the condition of low heuristic value, the algorithm will slow down, and try to find the shortest path, but may waste a lot of time. If it is a very high value, then the process becomes very fast but the shortest path will not be ideal. The tradeoff between efficiency and accuracy of the algorithm depends upon chosen the value and it should be very carefully selected in a different scenario. The classic expression of the A* algorithm is as follow:

$$f(x)=g(x)+h(x) \tag{16}$$

where $f(x)$ is the sum of path-cost function $g(x)$ and heuristic function $h(x)$. $g(x)$ is the actual total cost of the current node x from the start node. $h(x)$ is the estimation of cost of the current node to the target node estimates, which tells how far the distance to the goal node from the current node x. Heuristic estimate of $h(x)$ determines the way an agent searches the path. In the way, an agent keeps finding the lowest-cost node in the neighbor nodes, A* algorithm is guaranteed to give the shortest path if possible. Detailed steps of A* search algorithm are shown below:

1.	**A_Star_Search**(*start*, *goal*):
2.	**Initialized** *open*, *close* = {}
3.	**Initialized** *current*, *neighbor* = empty
4.	open add start
5.	**while** *current* is not *goal*

```
    current = lowest f cost node in open
     open remove current
     close add current
     foreach neighbor of current
        if neighbor is not traversable then
           skip to next neighbor
        end if
        g(neighbor)=g(current)+cost(current, neighbor)
        h(neighbor) = distance to the goal
        f(neighbor) = g(neighbor) + h(neighbor)
        open add neighbor
     end Foreach
end while
return close
```

Proposed Method

System overview of the proposed scheme is illustrated in Fig. 1. The proposed scheme can be divided into two parts, one is the map construction in SLAM, and the other is path poses reduction in the navigation process. The main contribution of this paper in SLAM is the fusion of the odometer readings and control command to calculate the initial estimate pose of a robot. It is helpful to perform SLAM in a long corridor as only a short-range rangefinder is available. The desired input, input command and odometer reading information received by a robot are fused in the EKF in SLAM process. In the navigation part, the A* path planner generates raw path poses and then the subgoal filter is applied to remove the redundant poses from raw path poses. Finally the remained key poses in the path are sent to the navigation controller of the mobile robot.

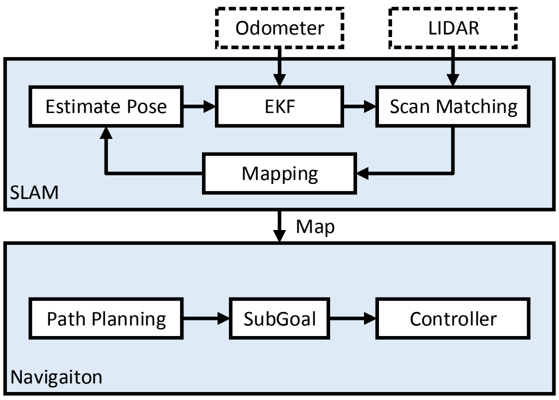

Figure 1. System overview of the proposed scheme.

Odometer EKF with Scan Matching

Original scan matching method proposed in Hector SLAM is not able to jump out of local minima in a long distance corridor environment, it only performs a good result with the high-end laser rangefinder device. After several experiments, the reason why the iteration result is not satisfied the real environment is that Gauss-Newton equation works well in acquiring local minima position, but the position of the robot is not always in the local minima when it is still moving. This situation casues the scan matching process to derive wrong positions in acquiring the map so far. If the workspace is lack of distinguishable landmarks, the scan-based matching algorithm will fail.

The architecture of the combined algorithm is shown in Fig. 2. Method proposed is a modification in the odometer sensor reading, by fusing raw sensor readings with control command $u = (v, \omega)^T$. Based on the EKF motion model as equation 14, a nonlinear motion equation is given as follows:

$$g(\xi_t, \mu_t) = \begin{bmatrix} \cos(\psi_{t-1}+\omega) & 0 \\ \sin(\psi_{t-1}+\omega) & 0 \\ 0 & 1 \end{bmatrix} \mu_t + \xi_{t-1} \tag{17}$$

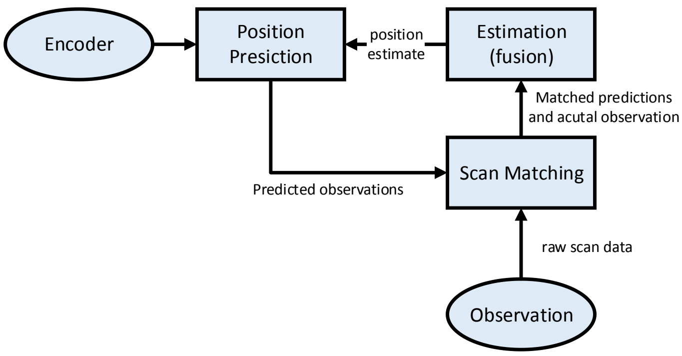

Figure 2. Architecture of combined EKF and scan matching.

Given the readings of odometer sensors, $z_t = (v_l, v_r)^{\mathrm{T}}$, K_t in the EKF can be calculated as follows:

$$K_t = \bar{\Sigma}_t C_t^T \left(C_t \Sigma_t C_t^T + Q_t \right)^{-1}, \tag{18}$$

and the next state of the odometer can be computed by the equation

$$\xi_t = \bar{\xi}_t + K_t \left(z_t - C_t \bar{\xi}_t \right). \tag{19}$$

This new variance matrix is updated by:

$$\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t \tag{20}$$

The end-point can be transformed into the world coordinate by submitting ξ_t into scan matching process:

$$S_i(\xi_t) = \begin{bmatrix} \cos(\psi_t) & -\sin(\psi_t) \\ \sin(\psi_t) & \cos(\psi_t) \end{bmatrix} \begin{bmatrix} s_{i,x} \\ s_{i,y} \end{bmatrix} + \xi_t \tag{21}$$

The gradient vector of the target function will be

$$G=\sum_{i=1}^{n}\left[\nabla M(S_i(\xi_{t-1}))\frac{\partial S_i(\xi_{t-1})}{\partial \xi}\right]^T\left[1-M(S_i(\xi_{t-1}))\right] \quad (22)$$

And Hessian matrix is given as follows:

$$H=\sum_{i=1}^{n}\left[\nabla M(S_i(\xi_{t-1}))\frac{\partial S_i(\xi_{t-1})}{\partial \xi}\right]^T\left[\nabla M(S_i(\xi_{t-1}))\frac{\partial S_i(\xi_{t-1})}{\partial \xi}\right] \quad (23)$$

Finally, add the step pose $\Delta\xi_t$ to ξ_t:

$$\xi_t=\xi_{t-1}+H^{-1}G \quad (24)$$

Algorithm will draw the scan lines with regard to pose ξ_t into the map at the end of the scan matching process. The first part is the use of the odometer readings and control command in the EKF. Second part is the scan matching method which remains the same as Hector SLAM. In this scheme, before the mapping action, robot will go forward or backward if command u is none zero, and then the scan matching will calculate gauss-newton approximation according to the new position. The odometry EKF with scan matching is shown below, and flowchart of the proposed process is shown in Fig. 3.

Odometry_EKF_ScanMatching $(\xi_{t-1},\Sigma_{t-1},\mu_t,z_t,s_{i,x},s_{i,y},M)$

$\bar{\xi}_t=g(\xi_{t-1},u_t)$

$\bar{\Sigma}_t=\Sigma_{t-1}+R_t$

$K_t=\bar{\Sigma}_t C_t^T\left(C_t\bar{\Sigma}_t C_t^T+Q_t\right)^{-1}$

$\xi_t=\bar{\xi}_t+K_t\left(z_t-C_t\bar{\xi}_t\right)$

$\Sigma_t=\left(I-K_tC_t\right)\bar{\Sigma}_t$

$S_i(\xi_t)=\begin{bmatrix}\cos(\psi_t) & -\sin(\psi_t)\\ \sin(\psi_t) & \cos(\psi_t)\end{bmatrix}\begin{bmatrix}s_{i,x}\\ s_{i,y}\end{bmatrix}+\xi_t$

$G=\sum_{i=1}^{n}\left[\nabla M(S_i(\xi_{t-1}))\frac{\partial S_i(\xi_{t-1})}{\partial \xi}\right]^T\left[1-M(S_i(\xi_{t-1}))\right]$

$H=\sum_{i=1}^{n}\left[\nabla M(S_i(\xi_{t-1}))\frac{\partial S_i(\xi_{t-1})}{\partial \xi}\right]^T\left[\nabla M(S_i(\xi_{t-1}))\frac{\partial S_i(\xi_{t-1})}{\partial \xi}\right]$

$\xi_t=\xi_{t-1}+H^{-1}G$

return ξ_t

Path Planning with Subgoal

A* path search algorithm generates an optimal path in the environment with obstacle cost. Each node has a state value of the sum of total cost g and heuristic function value h, and next neighbor state value will accumulate the current state value into the total cost value. Equation is shown as follows:

g(*neighbor*) = g(*current*) + cost(*current*, *neighbor*) (25)

The cost function is the distance between the current state and the neighbor state.

h(*neighbor*) = d(g, *current*) (26)

The heuristic function value h is the distance between current $d(goal, current)$. In the whole process, make sure pick the lowest state value (or lower heuristic value when the same), the shortest path will be found if it exists.

The subgoal filter takes the path $P=\{P_1,P_2\dots P_n\}$ pass from the A* algorithm as the input, and uses the first and second poses in the path as the standard regression line, definition is shown below:

$$L=(P_{2x}-P_{1x},P_{2y}-P_{1y})^T \quad (27)$$

And measure whether the line L', which is between the next two points, is in the tolerance angle range.

$$L\leftarrow\begin{cases}L', & \text{if angle } L' \text{ is large than } L\\ L, & \text{if angle } L' \text{ is small than } L\end{cases} \quad (28)$$

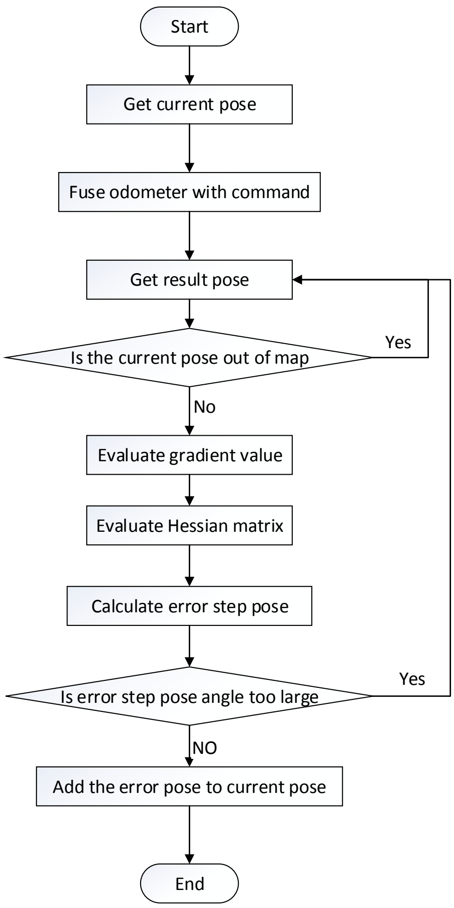

Figure 3. Flowchart of scan matching with odometer EKF.

Assume the empty output set Q = {}, if angle is larger than the previous regression line, add the current point P_i into the output Q set,

$$Q\leftarrow P_i \quad (29)$$

After examining all node in the path, the Q set will remain the refined poses. The subgoal filter algorithm is shown below.

```
Subgoal(P):
    Initialized Q= {}
    L ←(P2x - P1x, P2y - P1y)^T
    Pi ←Pa\
    Q←P1, P2
    while Pi is not Pn
        L' ←(Pix – P(i-1)x, Piy - P(i-1)y)^T
        if angle(L') larger than angle(L) than
            L ← L
            Q ← Pi
        end if
        Pi ← next
    end while
return Q
```

Experiment and Discussion

In this section, the implementation of the proposed method on a U-bot robot platform is described. The experimental setup and experimental results are conducted to compare the performance of localization and mapping.

Robot specification and Environment Setup

The mobile robot used in our experiments is the differential wheeled robot made by Industrial Technology Research Institute of Taiwan. The robot appearance is shown in Fig. 4. A short-range laser rangefinder (URG-04LX) mounted on the U-bot platform is used to acquire the ground truth of the environment. The computing platform (Laptop) has an Intel Core i5-5200U, 4GB memory, GT 930M graphics card. Robot operating system is used in our experiments as messaging platform for the subsystems to exchange various messages.

Figure 4. The U-bot robot platform developed by ITRI.

The workspace is a $60 \times 22.2m^2$ corridor field with a $6m$ foyer in the middle of the map. Fig. 5 shows the map built from sensor data of a long-range laser rangefinder (UTM-30LX). The map is perfect in distance translation and angular transformation.

Fig. 6 depicts the feature points in the workspace consisting of the short corridor, long corridor, and a foyer. The long corridor has almost the same scan information along the way and has a long distance path which can determine the straight line characteristic of the SLAM. This scenario can test the performance of the SLAM with the odometer noise and scan noises exist at every step. The short corridor has the identical scan information in each scan, but less noises or obstacles in the corridor, which implies the stability of the SLAM with a pure surrounding circumstance. The foyer is a special case in the SLAM problems. When robot passes the foyer, if scan distance is less than the width of the foyer, the acquired data from laser rangefinder is not enough for mapping correctly. Therefore SLAM localization relies on scan matching turn out to be failed.

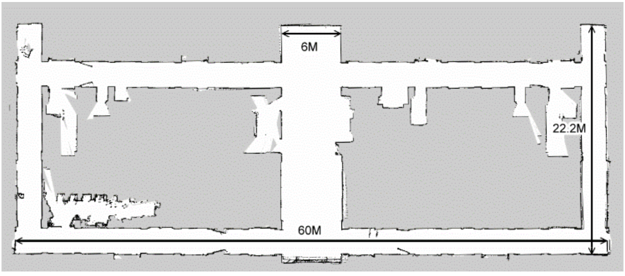

Figure 5. Map generate with 30m Laser rangefinder.

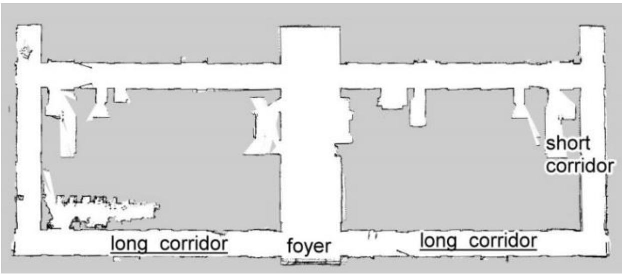

Figure 6. Feature point in the map.

Experiment Results

To verify the effectiveness of the proposed approach, experiments are performed in the same environment described in the previous section. Hector SLAM, OdometryUpdate SLAM, and OdometryEKF SLAM are performed for comparison by means of retrieving laser scanning data from a record file. A record file, which is called a "bag" file in the ROS platform, contains all of the odometry measurement, command of linear velocity and angular velocity, laser scan sensor readings and all the other topic's messages. Through the replay of the bag file, all experiments can be assumed conducted in the same condition, same sensor readings. All the algorithms are processed on the ROS platform which involved many useful libraries and tools. However, there is a frequency issue as programs running on the ROS platform, thus the results do not converge to the same map contour.

As shown in Fig. 7, a mobile robot using the Hector SLAM method creates a map with incorrect length in long corridor environment due to the laser scanning cannot identify the difference of walls along the route. Besides, lack of distinguishable landmarks in the foyer area results in wrong mapping both in the length and angular distance. Fig. 8 shows the result of the improved scan matching method, called OdometryUpdate SLAM, which believes in odometer in short distance movement and applies it to the current position. The map created by OdometryUpdate SLAM satisfies the length of route in long corridor environments. It is worth noting that overshot in the position estimate occurs when scan matching calculation falls in the same direction of the odometer updating. Compared to the original Hector SLAM, maps created by the OdometryUpdate SLAM are much better. Therefore the fusion of the information from odometer and laser rangefinder might be a good solution to this phenomenon.

Fig. 9 illustrates the map built by scan matching with the odometer in extended Kalman filter, called OdometerEKF SLAM. The consequence of OdometerEKF SLAM is not too much optimistic and not too much pessimistic with regard to the current position, and the measure of corridor length is between the Hector SLAM and OdometryUpdate SLAM.

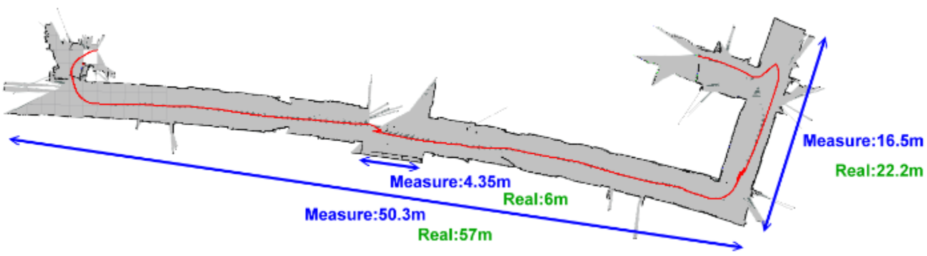

Figure 7. Hector SLAM mapping result.

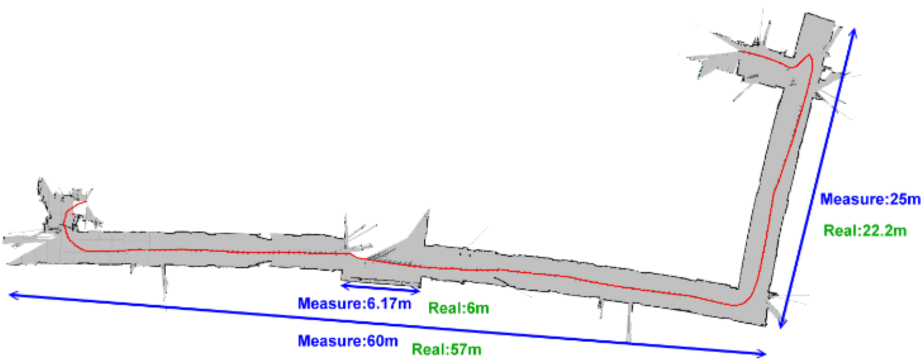

Figure 8. Odometer update and scan matching.

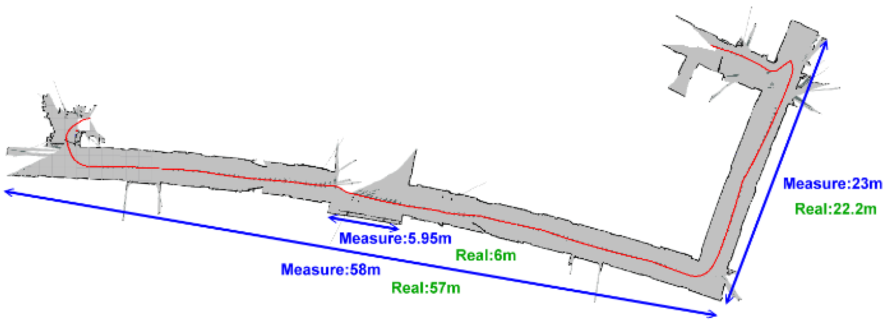

Figure 9. OdometerEKF and scan matching.

Three trajectories generated by Hector SLAM, OdometryUpdate SLAM, and OdometryEKF SLAM are compared in Fig. 10. Detailed analysis of the three featured areas are illustrated in Fig. 11-13. Each figure has a real data line which represents the real-life world length.

In Fig. 11, Hector mapping length is always not able to map the real value, in contrast, OdometryEKF SLAM and OdometryUpdate SLAM are close to the real value. In Fig. 12, a foyer is the area that robot forward scan readings are vain due to a short-range laser rangefinder, thus the

Hector SLAM will fail in this area. In Fig. 13, a short corridor is the area that shows the effect of OdometryEKF SLAM keeping the position close to real value while OdometryUpdate SLAM has an obvious overshoot in the length measurement of the short corridor.

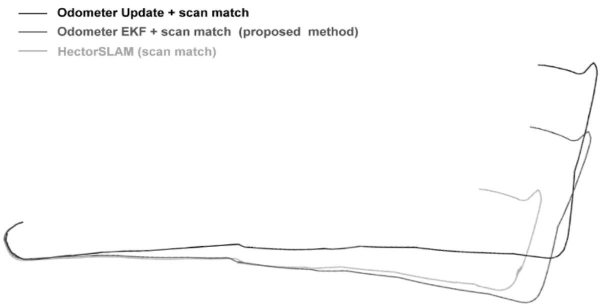

Figure 10. Comparison of trajectories generated by the three methods.

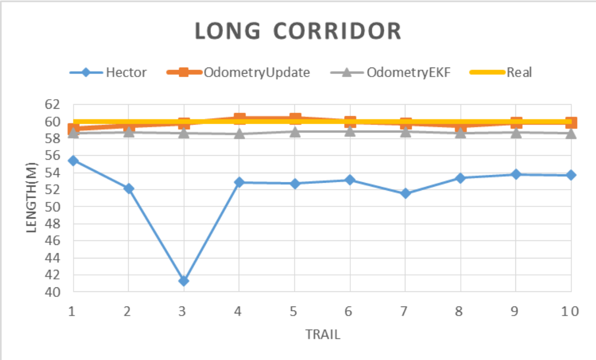

Figure 11. Length graph in long corridor area.

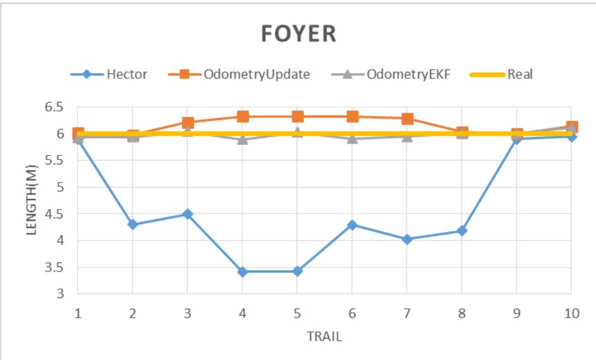

Figure 12. Length graph in foyer area.

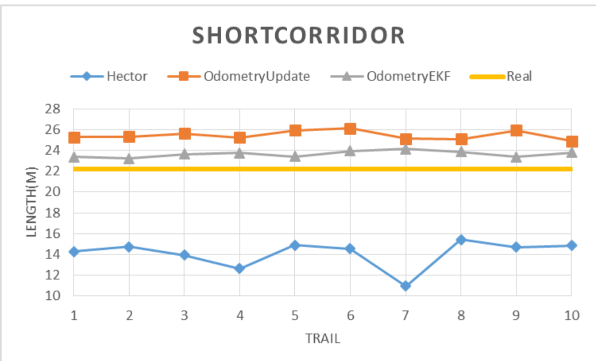

Figure 13. Length graph in short corridor area.

Conclusions

This paper provides a method for solving SLAM localization problem in long corridor environments with a low-cost short-range laser rangefinder. In path planning, a redundant pose reduction method called subgoal filter is applied to the path generated by A* algorithm for increasing the efficiency of navigation when a robot executes path following behavior. The map created by the proposed OdometryEKF SLAM can be guaranteed in distance accuracy, but accuracy of angular distance in a long corridor with changeless scenery is not good enough. It will induce just a little impact on navigation if robot localization is correct. Correction of the angular distance in long distance may need a recalculation in the closed loop. Lack of further scan information, the robot assumes the odometry and local scan information can be fused in short distance. It is helpful for map construction if localization depends on not only scan features but also odometer readings and makes the proposed method more flexible and effective than the original Hector scheme. Path planning provides users the full detail of the path on the map. But only corners and the points close to obstacles are essential and should be taken into consideration for path following. It can speed up the robot in the simple straight line navigation and corner turn, but the orientation will be the extended problem if the robot has to ensure the heading direction. From what has been mentioned above, the proposed SLAM localization using EKF is effective in any place including corridor scenario.

References

[1] J. E. Guivant and E. M. Nebot, "Optimization of the simultaneous localization and map-building algorithm for real-time implementation," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 3, pp. 242-257, 2001.
doi: 10.1109/70.938382

[2] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in proceeding of *the 16th Int. Joint Conf. on Artificial Intelligence*, pp. 1151-1156, 2003.

[3] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*, MIT Press, 2005.

[4] I. Ashokaraj, P. Silson, and A. Tsourdos, "Application of an extended Kalman filter to multiple low cost navigation sensors in wheeled mobile robots," in proceeding of *IEEE Sensors*, Orlando, FL, USA, June

12-14, 2002, vol. 2, pp. 1660-1664.
doi: 10.1109/ICSENS.2002.1037373

[5] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.

[6] J. Jung and H. Myung, "Indoor localization using particle filter and map-based NLOS ranging model," in proceeding of *IEEE Int. Conf. Robot. Autom*. (*ICRA*), Shanghai, China, May 9-13, 2011, pp. 5185–5190,.
doi: 10.1109/ICRA.2011.5980383

[7] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Trans. on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.
doi: 10.1109/TRO.2006.889486

[8] S. Kohlbrecher, J. Meyer, O. Von Stryk, and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation," in proceeding of *Int. Symp. Safety, Security and Rescue Robot*, Kyoto, Japan, Nov. 1-5, 2011, pp. 155-160.
doi: 10.1109/SSRR.2011.6106777

[9] E. Pedrosa, N. Lau, and A. Pereira, "Online SLAM based on a fast scanmatching algorithm," *Artificial Intelligence*, ser. Lecture Notes in Computer Sci., L. Correia, L. P. Reis, and J. Cascalho, Eds. Berlin, Germany: Springer-Verlag, vol. 8154, pp. 295-306, 2013.
doi: 10.1007/978-3-642-40669-0_26

[10] J. M. Santos, M. Couceiro, D. Portugal, and R. P. Rocha, "Fusing sonars and LRF data to perform SLAM in reduced visibility scenarios," in proceeding of *the 14th Int. Conf. Auton. Robot Syst. Competitions*, Espinho, Portugal, May 14-15, 2014, pp. 116-121.
doi: 10.1109/ICARSC.2014.6849772

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

[12] P. Agarwal, G. D. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard, "Robust map optimization using dynamic covariance scaling," in proceeding of *Int. Conf. Robot. Autom.*, Karlsruhe, Germany, pp. 62-69, 2013.

[13] E. W. Dijkstra, "A note on two pmblem in comexion with gaphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269-271, 1959.
doi: 10.1007/BF01386390

[14] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. on Systems Science and Cybernetics*, vol. SSC-4, no. 2, pp. 100-107, 1968.
doi: 10.1109/TSSC.1968.300136

[15] H. Johannsson, M. Kaess, M. Fallon, and J. J. Leonard, "Temporally scalable visual SLAM using a reduced pose graph," in *RSS Workshop on Long-term Operation of Autonomous Robotic Systems in Changing Environments*, 2012.
doi: 10.1109/ICRA.2013.6630556