Proceedings of the 3rd
RSI International Conference on Robotics and Mechatronics
October 7-9, 2015, Tehran, Iran

Modified Fast-SLAM For 2D Mapping And 3D Localization

Soheil Gharatappeh, Mohammad Ghorbanian, and Mehdi Keshmiri
Mechanical Engineering Dept., Isfahan University of Technology, Isfahan, Iran
s.gharatape@me.iut.ac.ir, mehdik@cc.iut.ac.ir

Hamid D. Taghirad
Faculty of Electrical Engineering, Department of Systems and Control, K.N.Toosi University of Technology, Tehran, Iran
taghirad@kntu.ac.ir

***Abstract*— Fast Simultaneous Localization and Mapping (SLAM) algorithm is capable of real-time implementation due to logarithmic time complexity which results in decrease of computational cost. In this algorithm state vector of a robot merely includes planar location of the robot and its angle to the horizontal plane. It has fewer components comparing to state vector in extended Kalman filter method which consists of location of all environmental features. In existing methods for implementing this algorithm, robot movement is considered to be totally in planar movement; while if moving on a slope changes the pitch angle of the robot, it causes errors in the algorithm. Correcting these errors will lead to a precise 2D mapping and 3D localization. This paper details the modification added to conventional Fast-Slam algorithm to accommodate this requirement by using an IMU. Simulation and experimental results shows the effectiveness of such modification.**

***Keywords*— Fast-SLAM, 2D localization, 3D Mapping, pitch angle, mobile robot.**

I. Introduction

SLAM problem deals with robot excavation in unknown environment or localization and simultaneous map building. The difficulty of this method is related to the dependency of localization and mapping. To address localization issue, it is necessary to have maps of the environment so that observations could be matched to it and an estimation of robot location could be achieved. On the other hand, to build a map of the robot's environment one must has information about its location at each moment. Putting this information together the robot would be able to complete the map and localize itself.

Various frameworks have been considered to address SLAM problem. As an instance for implementation of SLAM in indoor environments, outdoor environments, submarine environment and air operations might be mentioned. SLAM problem can be solved for two types of environments; 3D and 2D. Algorithms for SLAM solutions are different considering temporal priority and response quality. The main methods may be listed as Extended Kalman filter method, 2. Fast-SLAM and 3. Iterative smoothing and mapping (iSAM). The first efforts regarding SLAM problem were made by Smith, Leonard and Durrant in a probabilistic framework [1], [2], [3]. They considered state vector of the robot and environment in the form of random variables. These studies aimed to estimate Gaussian probability density distribution for each random variable at each time and to describe the environment map using estimated variables. For this purpose previous observations and estimations are exploited.

In 2001 Dissanayake et al proposed a solution for SLAM problem based on Extended Kalman filter [4]. In this article primary estimation was performed based on kinematics of the problem and issues such as map convergence were investigated. Since state space equations for motion and observation models are considered to be nonlinear, Kalman filter cannot be used directly. At each step linear models approximated by first term of Tailor expansion are substituted in equations. This modeling error may result in inconsistency and divergence of the solution after a while.

Another problem associated with EKF-SLAM algorithm is that size of the state vector and number of state parameters increase when the number of robot observations increases. It, in turn, results in larger covariance state matrix with $O(N2)$ complexity degree. In recent decades several efforts have been made to address this problem. One of these solutions was introduced by Thrun et al [5] in 2004. They presented a method for solving SLAM problem using extended information filter called SEIF. Baily et al investigated solution convergence for EKF method in 2006 [6]. At the same year Castellanos et al pointed out some facts regarding convergence improvement and convergence speed in estimation problem using Extended Kalman filter [7].

In Extended Kalman Filter algorithm motion and observation models' equations are considered nonlinear and noises are introduced by Gaussian distributions. In this method, problem would be solved analytically. Fast-SLAM employs particle filter for estimating variables. In contrast to Kalman filters, in particle filters nonlinear models and non-Gaussian noises could be described pretty well and equations are solved numerically. Another issue that EKF encounters is data associations when robot is located in a messy environment. Gaussian distribution is not capable of expressing multi-modal density functions while density distribution function which states the map in symmetric environments is multi-modal.

Employing particle filters and Monte Carlo methods was proposed by Zaritskii, Akashi and Doucet in 1975, 1977 and 2000, respectively [8-10]. In 2001, Montemerlo exploited a particle filter algorithm called Fast-SLAM and achieved acceptable results [11]. Numerous algorithms have been proposed based on Fast-SLAM structure among which [12-14] could be mentioned. An improved version of this method called Fast-SLAM 2 was presented by Montemerlo in 2003 [15].

Another solution called Graph-SLAM algorithm was introduced by Thrun and Montemerlo in 2006 [16]. The basis of this method is drawing a graph where position of environment signs and robot states constitute graph nodes and graph edges include distance between signs which is estimated by robot sensors. In the next step of this algorithm the graph is optimized so that measurement noises could be eliminated. Graph-SLAM is one of the smoothing methods in solving SLAM problems. Another method has been proposed by Kaess in 2008 which is called iSAM. It deals with SLAM problem using smoothing technique [17].

In this study the error resulting from robot movement on slopes and environment obstacles is compensated using IMU sensor and deriving robot pitch angle.

II. FAST-SLAM ALGORITHM

Solving SLAM problem corresponds to obtaining robot state vector (1). In Kalman filter algorithm state vector is composed of three elements representing robot center of gravity's position and robot's yaw angle while 2*n elements are associated with abscissa and ordinate of features extracted from the environment.

$$x = (x_r, y_r, \theta_r, \underbrace{x_1, y_1, x_2, y_2, \ldots, x_n, y_n}_{map}) \tag{1}$$

x_r and y_r are robot's two-dimensional planar coordinates relative to an external coordinate frame θ_r is the yaw angle of robot,x_i and y_i are map's two-dimensional planar coordinates. This filter is categorized as Bayesian filters; so, similar to other filters of this type it includes two steps. The first step is primary prediction based on robot movement model. In the second step the estimation is modified using observation model. The equations of these two model are shown by (2) and (4) for system model and observation model, respectively.

$$\begin{aligned} x_{k+1} &= f\left(x_k, u_k\right) \\ z_{k+1} &= g\left(x_{k+1}\right) \end{aligned} \tag{2}$$

x_{k+1} denotes the robot's position in $(k+1)$ th step and u_k is the robot's control command in (k) th step. Movement model or system model in localization problem is the same as kinetic model of the robot:

$$\begin{aligned} x_{k+1} &= x_k + v*\cos\left(\theta_k\right) \\ y_{k+1} &= y_k + v*\sin\left(\theta_k\right) \\ \theta_{k+1} &= \theta_k + \omega \end{aligned} \tag{3}$$

v is linear velocity and ω is angular velocity.

The system observation model equations are as follows.

$$\begin{aligned} d &= \sqrt{(x_{p_i} - x_r)^2 + (y_{p_i} - y_r)^2} \\ \theta_{p_i} &= a\tan\left(\frac{y_{p_i} - y_r}{x_{p_i} - x_r}\right) \end{aligned} \tag{4}$$

Inclusion of linear velocity and angular velocity in the movement model as inputs provides a prediction of next robot's location. Afterwards, the estimation is modified using observation model. The equations for observation model are extracted based on observation sensor type.

In Fast-SLAM algorithm robot state vector is divided into two sections. The first section corresponds to robot location and second one is related to the environment's map. Since the robot location is known in the Rao-Blackwell method, positions of environmental features are independent of robot location. Hence, the SLAM problem could be divided to two general part: using particle filter to achieve robot localization components and using the Kalman filter to estimate location of each environmental feature (5).

$$p(x_{0:t}, l_{1:M} | z_{1:t}, u_{1:t}) = p(x_{0:t} | z_{1:t}, u_{1:t}) p(l_{1:M} | x_{0:t}, z_{1:t}) \tag{5}$$

$x_{0:t}$ are robot's position, $l_{1:M}$ is the environment's map, $z_{1:t}$: observations and $u_{1:t}$ control command of robot. In this equation, $p(x_{0:t}|z_{1:t}, u_{1:t})$ denotes path posterior and $p(l_{1:M}|x_{0:t}, z_{1:t})$ map posterior.

Kalman filters utilized in this part of algorithm require much less computational power compared to the previous Kalman filter solution for SLAM, which considered the whole state vector (1) as a co-related entity to estimate. Besides, these characteristics facilitate real-time implementation of this algorithm and its implementation on grid-based maps.

III. Modifying Motion Model For 3D Movement

If the system experiences non-planar motion, motion model equations must be modified. For instance, assume that the robot is climbing a slope as shown in Fig. 1. In such circumstances motion model equations of the system might be rewritten as (6):

$$\begin{aligned} x_{k+1} &= x_k + v * \cos(\theta_k) * \cos(\varphi_k) \\ y_{k+1} &= y_k + v * \sin(\theta_k) * \cos(\varphi_k) \\ \theta_{k+1} &= \theta_k + \omega \end{aligned} \tag{6}$$

φ_k is robot's pitch angle in Kth time step. Modified observation model for special robot's movement would be similar to (7).

$$\begin{aligned} d &= \cos(\varphi_k) * \sqrt{(x_{p_i} - x_r)^2 + (y_{p_i} - y_r)^2} \\ \theta_{p_i} &= a\tan\left(\frac{y_{p_i} - y_r}{x_{p_i} - x_r}\right) \end{aligned} \tag{7}$$

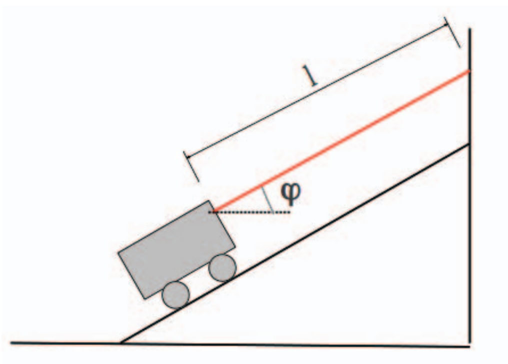


Fig. 1. A robot located in the slope

Fig. 2. SLAM algorithm implementation platform

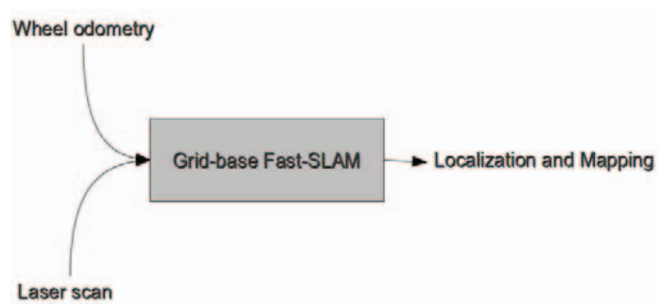
Thus, the localization and mapping could be modified using the angle derived from IMU sensor. As depicted in Fig. 2, the main algorithm uses 2D model for localization and it was not able to identify changes in pitch angle. Including IMU sensor data the mentioned error is corrected.

IV. Algorithm Implementation

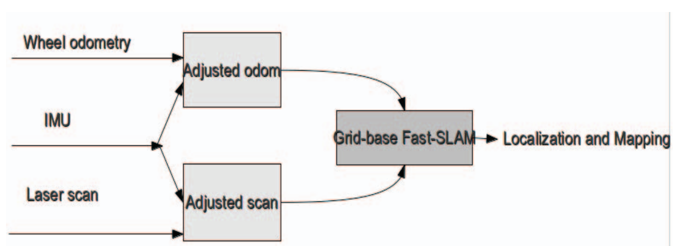
A. Experimental Setup

The modified algorithm was implemented on SLAM-test-bench built in Dynamic and Robotic Centre of mechanical engineering faculty, Isfahan University of Technology. This platform consists of two encoder sensors to measure robot angle in horizontal plane and its forward movement. Moreover, it includes a Laser Range Finder and an IMU sensor.

To implement basic algorithm a prepared package in ROS (Robot Operating System) called G-mapping is utilized. This package is an open-source implementation of Fast-SLAM 2.0 algorithm for robot's planar movement. The ROS distribution used in this study was Fuerte which had been installed on Linux Ubuntu 12.04. The important parameters were set according to table 1. Other values are assumed to be the same as default values in the algorithm. To simulate slope environment, an artificial slope with +30 and -30 slopes was placed in the environment and the algorithm was executed when the robot moves on this slope. The schematic of the slope is illustrated in Fig. 4.



(a)



(b)

Fig. 3. (a) basic diagram of the algorithm and (b) its modification by IMU

TABLE I. VARIABLE DEFINITIONS

Variable	Definition
~srr	error in circular motion data collection resulted from circular motion
~srt	error in circular motion data collection resulted from translation motion
~str	error in translation motion data collection resulted from circular motion
~stt	error in translation motion data collection resulted from translation motion
~particles	number of particles used for introducing robot trajectory

TABLE II. SETTING VARIABLE VALUES

variable	default value	assigned value
~srr	0.1	0.2
~srt	0.1	0.055
~str	0.1	0.055
~stt	0.1	0.05
~particles	30	20

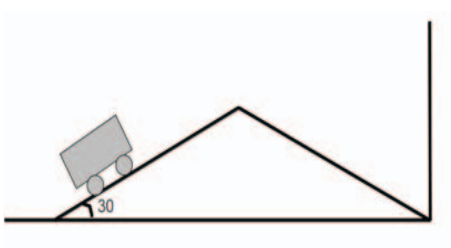


Fig. 4. Side view of the slope

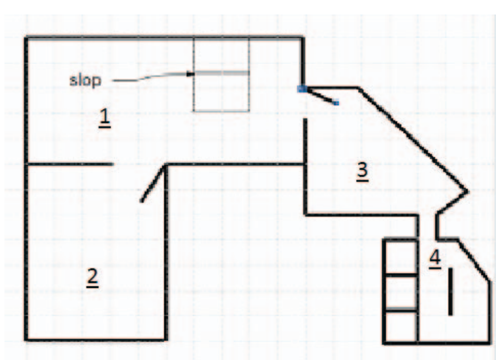


Fig. 5. The actual map of the environment

As there was not a ready-to-use data set, in which contained 3D movement that pitch angle of the robot varies among it's movement, a data set was gathered in Dynamic and Robotic Centre of Mechanical Engineering Faculty, by using an artificial slope inside its environment. To generate ground-truth, modern and expensive tools are needed while validating the algorithm requires a reference. Since such tools was not available *s*parameter was introduced to validate the algorithm as (8)

$$s = \sqrt{(x_{odom} - x_{slam})^2 + (y_{odom} - y_{slam})^2} \qquad (8)$$

At each instant, Fast-SLAM introduces a location as the robot's location to the algorithm based on the odometery (x_{slam}). The algorithm also corrects the position using existing filters. ***s*** parameter demonstrate the distance between location which is obtained by unadjusted algorithm and the location which is obtained by adjusted algorithm. Higher jump in this parameter demonstrates that larger error was made by wheel odometery and algorithm has corrected erroneous location.

B. Estimating altitude component

As mentioned above when the robot is on a slope, kinematic movement equations change and they do not satisfy planar movement equations. New equations might be rewritten as shown in (6). Robot movement equation along with the axis perpendicular to the horizontal plane is:

$$z_{k+1} = z_k + v\sin(\varphi_k) \qquad (8)$$

which provides primary estimation of robot altitude. To obtain more precise estimation of robot location Extended Kalman Filter might be utilized. It needs a correction equation. Correction equations in Kalman filters are naturally more precise than prediction ones. Thus, in most cases more precise and complicated sensors are utilized which, consequently, increases computational load of algorithm and robot cost. To address this problem the constraint generated on robot planar movement is utilized for modifier equation of Kalman filter. It is assumed that in each time span the exact location of robot is known. Using angle reported by IMU sensor and considering constraint for planar movement, its progress and, as a result, its altitude is calculated according to undergoing equation.

$$z_{k+1} = z_k^+ + (\Delta\bar{s})\sin\left(\phi_k\right) \qquad (9)$$

Where

$$\Delta\bar{s} = \Delta s / \cos(\phi)$$
$$\Delta s = \sqrt{\left(x_{k+1}^{+} - x_{k}^{+}\right)^2 + \left(y_{k+1}^{+} - y_{k}^{+}\right)^2} \quad (10)$$

Now the remaining problem is to find exact location of the robot in 2D plane. It is known that Fast-SLAM algorithm returns corrected 2D location of the robot as an output. Hence, the altitude of the robot could be derived by an extended Kalman filter which uses robot 2D location, the constraint generated by robot movement on the slope as corrector equation and IMU sensor data.

C. Results

A. map correction: data collection from the environment shown in Fig. 4 is performed according to table 3. Fig. 5 illustrates the map resulted from modified algorithm. Comparing achieved map with actual one (Fig. 4), reveals high precision of the algorithm.

Fig. 6 depicts variations of s versus time. Step changes in this parameter reveal the points that algorithm corrects the robot's location. According to Fig. 6, when robot enters the slope (390s-450s), the base algorithm constantly corrects robot's location. The jumps inside the diagram related to the basic algorithm might be interpreted as these corrections. The corrected algorithm diagram for this time span is similar to a straight line. Decrease of number and amplitude of jumps means that the algorithm is corrected and the error in slope is reduced.

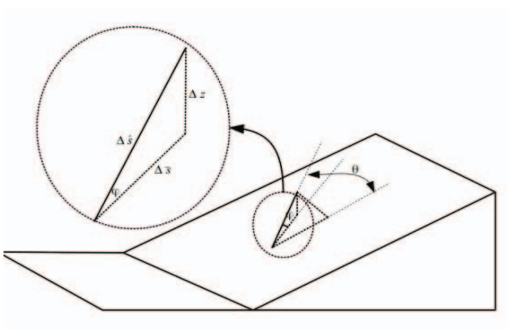


Fig. 6. Schematic of a slop

This correction is prominent in higher slopes; because when the slope of terrains increases, the error of basic algorithm increases as it is dependent on cosine of slope angle (6),(7). Thus, considerable error is expected in large slopes. On the other hand when the robot passes through consecutive positive and negative slopes in a natural environment (such as hilly environment), the base algorithm will lose its precision. The location of robot and map of environment will not be reliable due to accumulated errors.

B. 3D localization: when extended Kalman filter is employed to estimate altitude simultaneous with data collection, Fig. (9) is achieved.

V. Conclusions

When the robot is located in a slope the base algorithm, in which corrections regarding robot's pitch angle in the slope are not considered, is not performing correct solutions. The localization errors might be generated from two aspects. 1. Error in declaring the precise location of the robot. 2. Error in distance from barrier declared by Laser scanner sensor. By correcting these errors, localization will be more precise and the generated map will be more accurate. As can be seen in the figure, although the map is in base location and the algorithm cannot detect lines, the wall in front of the slope is detected as a smooth line in the correct place and with high precision. In Fig. (9) validation is performed using actual altitude of the slope which is 23 cm. As can be seen the altitude of robot which is measured by odometery algorithm reaches more than 25 cm which results in 2 cm of error. The error in extended Kalman filter is significantly reduced and the algorithm shows 23 cm for the altitude. The error of odometery algorithm increases when the robot experiences several slopes as a consequence of error accumulation.

TABLE III. ROBOT SCAN INSIDE THE ENVIRONMENT IN SEPARATED TIME SPANS

Time (S)	Scanned area
0-100	1
100-160	2
160-190	1
190-340	From 3 to 4 and from 4 to 3
340-360	1
360-390	On positive slope
390-415	On negative slope
415-460	1

Fig. 7. The map obtained by executing modified algorithm

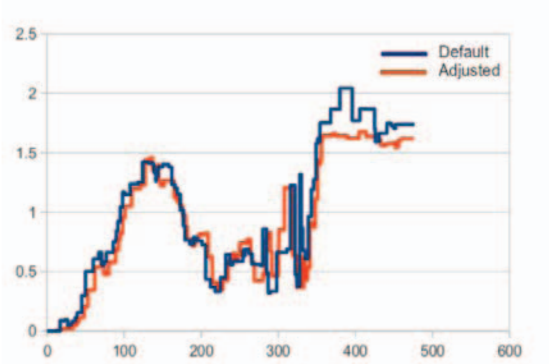


Fig. 8. Variations of s versus time

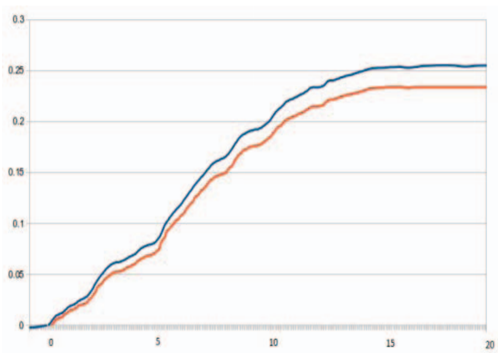


Fig. 9. basic odometery result compared to proposed algorithm result for estimating robot's altitude.

REFERENCES

[1] Smith, R.C., Cheeseman, P., 1986. "On the representation and estimation of spatial uncertainty". The international journal of robotics research, 5(4): p. 56

[2] Smith, R.C., Self, M., Cheeseman, P., 1990. "Estimating uncertain spatial relationships in robotics". Autonomous robot vehicles, 1(3): p.167193.

[3] Leonard, J.J., Durrant-Whyte, H.F., 1991. "Simultaneous map building and localization for an autonomous mobile robot". Intelligence for mechanical systems, proceedings IROS'91. IEEE/RSJ international workshop, 4(2): p.1442–1447.

[4] Dissanayake, G., Newman, P.M, Durrant-Whyte, . H.F., Clark, S., Csorba, M., 2001. " A solution to the simultaneous localization and map building (SLAM) problem". IEEE transaction robotic automation, 17(3): p.229-241.

[5] Thrun, S., Liu, Y., Koller, D., Ghahramani, Z., Durrant-Whyte, H., 2004. "Simultaneous localization and mapping with sparse extended information filters". The international journal of robotics research, 23(7-8): p. 693.

[6] Bailey, T., Nieto, J., Guivant, J., Stevens, M., 2006. "Consistency of the EKF-SLAM algorithm. Intelligent robots and systems". IEEE/RSJ international conference, p.3562-3568.

[7] Castellanos. J.A., Martinez-Cantin, R., Tard´os, J.D., 2006. "Robocentric map joining: Improving the consistency of EKF-SLAM". Robotic and autonomous system journal, 55(4): p. 281-291.

[8] Zaritskii, V., Svetnik, V., Shimelevich, L., 1975. "Monte carlo technique in problems of optimal data processing". Automation and remote control, 12(6): p. 95–103.

[9] Akashi, H., Kumamoto, H., 1977. "Random sampling approach to state estimation in switching environments". Automatica, 13(4): p. 429–434..

[10] Doucet, A., De Freitas, N., Murphy, K., Russell, S., 2000. "Rao-blackwellised particle filtering for dynamic bayesian networks". Proceedings of the sixteenth conference on uncertainty in artificial intelligence, p. 176–183.

[11] Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B., 2002. "FastSLAM: A factored solution to the simultaneous localization and mapping problem". in Proceedings of the national conference on artificial intelligence. Gordon, N.J., Salmond, D.J., Smith, A.F.M., 1993. "Novel approach to nonlinear/non-Gaussian Bayesian state estimation". IEEE proceedings, 140: p. 107–113.

[12] Isard, M. Blake, A., 1998. "A mixed-state condensation tracker with automatic model-switching". Sixth international conference on computer vision, p. 107 - 112

[13] Kanazawa, K., Koller, D., Russell, S., 1995. "Stochastic simulation algorithms for dynamic probabilistic networks". Proceedings of the eleventh annual conference on uncertainty in AI (UAI '95). University of California, Berkeley, CA 94720, USA.

[14] Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B., 2003. "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges". in Proceedings of the sixteenth international joint conference on artificial intelligence (IJCAI). Acapulco, Mexico

[15] Thrun, S., Montemerlo, M., 2006. "The graph SLAM algorithm with applications to large-scale mapping of urban structures". International jornal of robotics, 25(5-6): p. 403.

[16] Kaess, M., Ranganathan, A., Dellaert, F., 2008. "iSAM: Incremental Smoothing and Mapping". Robotics, IEEE transactions, 4(6): p.1365-1378