2010 3rd International Conference on Computer and Electrical Engineering (ICCEE 2010)
IPCSIT vol. 53 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V53.No.1.65

A New SLAM Algorithm Particle Filter-based for Autonomous Underwater Vehicle

Bo He[1+], Tao Zhang[1] and Guijie Liu[2]

[1]School of Information Science and Engineering, Ocean University of China, Qingdao, P. R. China
[2]School of Engineering, Ocean University of China, Qingdao, P. R. China

Abstract. SLAM (Simultaneous Localization and Mapping) has attracted more and more attention in AUV(Autonomous Underwater Vehicle), and some approaches to SLAM has also got rapid development. The dominant approach to the SLAM is the algorithm based on Kalman filter or Particle Filter. In order to adapt to different motion environment, FASTSLAM based on Rao-Blackweellized filter is generated which is implemented faster and more effective. Because the environment detector is sonar with lower output frequency, this paper uses inertial device whose sampling frequency is higher than velocity sensor to detect the point feature more accurately. This paper presents a new algorithm named auxiliary FASTSLAM, which is changed from FASTSLAM through adding some auxiliary variable, to realize more accurate navigation and localization without GPS. The simulation results show that the algorithm is more stable and reliable than pure EKFSLAM.

Keywords: SLAM; KalmanFilter; ParticleFilter; Auxiliary FASTSLAM

1. Introduction

All SLAM (Simultaneous Localization and Mapping) has got more and more and more attention in the robotics literature. Since [1], through the 20 years' development, many excellent works come out. The dominant approach is based on Kalman filter or particle filter. Kalman filter is optimal when handling linear-Gaussian problem, but particle filter is better when the problem is nonlinear because it needn't linearize [2, 3]. To deal with the drawback that the computational complexity is increased quadratically with the state vector's growth, Rao-Blackweellized particle filter was raised in [4,5]. Reference [6, 7] present the FASTSLAM algorithm to improve the efficiency of map processing and the accuracy of the data association. In this paper, a modified FASTSLAM is presented, named auxiliary FASTSLAM, which is adapting to our sensors (inertial measurement units (IMU)) in the robot, to realize the SLAM in an unknown environment. Section Ⅱ of this paper gives a brief overview of FASTSLAM. In the following section the auxiliary FASTSLAM will be described. Section Ⅳ presents and discusses the simulation results obtained and Section Ⅴ provides a conclusion and indication of the future work.

2. Overview of Fastslam

The popular online-SLAM is to estimate the posterior $p(s_t,M|Z^t,C^t,D^t)$ over robot pose and map information where s_t is the current robot pose and M is the map. In the planar SLAM, the robot pose includes the coordinate x and y with its heading. M presents the landmarks' position in the environment. The posterior at time t can be computed recursively by that at time $t-1$ by the definition of the motion model and the measurement model because the SLAM problem can be treated as a probabilistic Markov chain. The motion model $p(s_t|s_{t-1},C^t)$ is a function about controls, and the measurement model $p(Z^t|s^t,M,D^t)$ conditioned on the

+ Corresponding author.
E-mail address: bhe@ouc.edu.cn.

robot pose and the map. To implement the FASTSLAM effectively, the SLAM posterior will be described as $p(s^t,M|Z^t,C^t,D^t)$ which can be factored into a product of simpler terms. Observations in the environment are independent each other given knowledge of the robot's path, so the SLAM posterior can be factored in detail [6]. Equation (1) can be derived directly:

$$\begin{aligned} & p(s^t,MM|^t,C^t,D^t) \\ & = p(s^t|Z^t,C^t,D^t)p(M|s^t,Z^t,C^t,D^t) \\ & = \underbrace{p(s^t|Z^t,C^t,D^t)}_{pathestima\ tor}\underbrace{\prod_{i=1}^{m} p(m_i|s^t,Z^t,C^t,D^t)}_{landmarkes\ timators} \end{aligned} \tag{1}$$

The leftmost term in (1) can be realized by particle filter and the last term by Kalman filter. Because all landmarks' estimate is nonlinear and conditioned on the robot's path, each particle contains many EKF processing. If there are n particles, then $n \bullet m$ EKFs will be implemented. Each particle is of the form: $s_t^{[j]}=\left\{s^{t,[j]},\mu_{1,t}^{[j]},\sum_{1,t}^{[j]},\ldots,\mu_{m,t}^{[j]},\sum_{m,t}^{[j]}\right\}$, where the bracket notation $[j]$ indicates the index of the particle; $s^{t,[j]}$ is the j-th particle's pose estimate, and $\mu_{m,t}^{[j]}$, $\sum_{m,t}^{[j]}$ are the mean and covariance of the m-th feature in the j-th particle. The whole algorithm of FASTSLAM is just one more step than the pure particle filter [9], and is performed in four steps: first, a new robot pose will be received by sampling the old particles, then the landmark EKFs corresponding to the map features are updated or augmented with the new observations. Next, the importance weight will be computed and the importance resampling will be done for the true posterior at last.

3. Auxiliary Fastslam

In the real underwater situation, velocity sensor has low output frequency, and the environment detector is sonar with lower frequency. For making the feature detected from environment exactly, this paper uses inertial device with higher output frequency. But the inertial error accumulates with the time going on, so some corrections must be done for better results. All the reasons cause the new algorithm named auxiliary FASTSLAM produced. Velocity in the planar will be treated as both state variable and observation variable. In the system, state vector is $X=[s,M,V_x,V_y]$, where $s=[x,y,\theta]$ is the robot pose. Meanwhile the other sensors will result in the observation vector $Z^l=[V_x,V_y]$ and $Z^n=[r,\psi]$, where superscript l stand s for linear and n nonlinear. As mentioned above, the whole posterior can be described as $p(s^t,v_x,v_y,M|Z^{l,t},Z^{n,t},C^t,D^t)$. Equation (2) is the motion model, and the processing noise is Gaussian and notation is $W^n=[N_x,N_y,N_\theta]$ whose mean is zero with covariance Q^n. The measure model is (3),where s in Z^n stands for robot pose and m is feature's position, and N^n has a similar property like W^n 。

$$X^n \Rightarrow \begin{cases} x_t = x_{t-1} + (v_{x,t-1}\times\Delta t + 0.5\times a_{x,t-1}\times\Delta t\times\Delta t)\times\cos(\theta_{t-1}) \\ \qquad -(v_{y,t-1}\times\Delta t + 0.5\times a_{y,t-1}\times\Delta t\times\Delta t)\times\sin(\theta_{t-1}) + N_{x,t-1} \\ y_t = y_{t-1} + (v_{x,t-1}\times\Delta t + 0.5\times a_{x,t-1}\times\Delta t\times\Delta t)\times\sin(\theta_{t-1}) \\ \qquad +(v_{y,t-1}\times\Delta t + 0.5\times a_{y,t-1}\times\Delta t\times\Delta t)\times\cos(\theta_{t-1}) + N_{y,t-1} \\ \theta_t = \theta_{t-1} + w_{t-1}\times\Delta t + N_{\theta,t-1} \end{cases} \tag{2}$$

$$X^l \Rightarrow \begin{cases} v_{x\,t} = v_{x,t-1} + a_{x,t-1}\times\Delta t + N_{vx,t-1} \\ v_{y\,t} = v_{y,t-1} + a_{y,t-1}\times\Delta t + N_{Vy,t-1} \end{cases}$$

$$\begin{aligned} & Z^{l,t} = \begin{bmatrix} V_{x,t} \\ V_{y,t} \end{bmatrix} = \begin{bmatrix} V_{x,t} \\ V_{y,t} \end{bmatrix} + N_t^l \\ & Z^{n,t} = \begin{bmatrix} r(s,m) \\ \psi(s,m) \end{bmatrix} = \begin{bmatrix} \sqrt{(m_x-x)^2+(m_y-y)^2} \\ \tan^{-1}(\dfrac{m_y-y}{m_x-x})-\theta \end{bmatrix} + N_t^n \end{aligned} \tag{3}$$

$$\begin{aligned} & p(X^{l,t}|X^{n,t},C,D)p(X^{l,t}|Z^{l,t},C,D)p(X^{n,t}|Z^{n,t},C,D) \\ & \prod_{i=1}^{m} p(m_i|X^{n,t},Z^{n,t},C,D) \end{aligned} \tag{4}$$

$$Gz = \begin{bmatrix} \frac{m_x - x}{\sqrt{q}} & \frac{m_y - y}{\sqrt{q}} \\ \frac{m_y - y}{q} & \frac{m_x - x}{q} \end{bmatrix}$$

$$Gs = \begin{bmatrix} \frac{x - m_x}{\sqrt{q}} & \frac{y - m_y}{\sqrt{q}} & 0 \\ \frac{y - m_y}{q} & \frac{x - m_x}{q} & 1 \end{bmatrix} \tag{5}$$

$$\begin{aligned} X_t^1 &= X_{t-1}^1 + \Psi_2 \times C_{t-1} + W_{t-1} \\ z &= C_b^n \times (X_t^n - X_{t-1}^n) - \Psi_1 \times C_{t-1} - \Phi \times X_{t-1}^l + C_n^b \times W_{t-1}^n \end{aligned} \tag{6}$$

$$\begin{aligned} & T_{t-1} = \Phi \times P_{t-1|t-1} \times \Phi^T + Q_{t-1}^n \\ & L_{t-1} = P_{t-1|t-1} \times \Phi^T \times T_{t-1}^l \\ & P_{t-1|t-1}^* = P_{t-1|t-1} - L_{t-1} \times T_{t-1} \times L_{t-1}^T \\ & X_{t-1|t-1}^{l\ *} = X_{t-1|t-1}^l + L_{t-1} \times (z - \Phi \times X_{t-1}^l) \end{aligned} \tag{7}$$

$$\begin{aligned} & \hat{X}_{t|t-1}^l = X_{t-1|t-1}^{l\ *} + L_{t-1} \times (z - \Phi \times X_{t-1}^l) \\ & P_{t|t-1}^* = P_{t-1|t-1} - L_{t-1} \times T_{t-1} \times L_{t-1}^T + Q^l \end{aligned} \tag{8}$$

According to the thought in [10], the posterior can be factored into (4) by some conditions because Z^n and X^l are independent, meanwhile Z^l and X^n are independent. The second term in (4) is linear-Gaussian, so optimal results will be got by Kalman filter. And the last two terms will be handled by FASTSLAM. To the first term, an algorithm similar to Kalman filter is used to update and predict the last linear state. This makes the linear state more accurate. And that will cause the nonlinear state more accurate.

The whole algorithm can be summarized as following:

a) Initialize the state vector that will be estimated.

b) Given a new acceleration and angular speed, nonlinear state X^n that is the robot pose will be predicte.

c) If some landmark is observed, then the augment and update about the landmark will be implemented by EKFs in particles and the Jacobian matrix is (5), where $q = (m_x - x)^2 + (m_y - y)^2$. After this, robot pose will be updated and importance weight will be computed. If there is no landmark, this step will be skipped and step four will be implemented directly.

d) If data come from sensor about linear state X^l, the Kalman filter will be implemented. If there is no such data, step five will be directly realized.

*e) An optimal pose will be received by dealing with the particles. And then X^n will be treated as an observation to update X^l by a function that is similar to Kalman filter. The details are demonstrated in (6) and (7), where C_n^b is the transformation matrix from navigation coordinate frame to robot body frame, Ψ_2 and Ψ_1 are transformation matrix about controls, Φ is about the linear model, $P_{t-1|t-1}^*$ which is notated by a star * to interpret that this updating result is the covariance before a new control comes. Equation (7) is the second measurement update process.*

f) Equation (8) will help to realize the prediction of X^l. And then step two recurs.

4. Simulation Of Auxiliary Fastslam

According to real situation, this paper adopts east-north coordinate frame. Simulated data is a sequence of acceleration and angular speed. In the simulation, the output frequency of the inertial devices is 20HZ, speed measurement device is 5Hz, and feature detecting frequency is 1.25Hz. To make the verification of the algorithm easier, control is only done on east and this will cause error in the north bigger. Both a ground truth of the robot trajectory and a trajectory with noise are derived by simple dead-reckon. In the Fig.1, the blue(solid) line stands for the true trajectory ,and the red(dash) is the trajectory added noise. The green dots which are up to 80 are environment features. Fig.2 is error of path without using SLAM. In this paper, 100 particles are used.

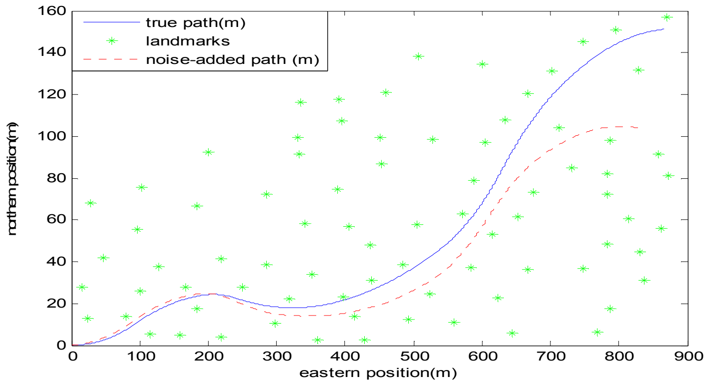

Figure 1. True robot trajectory, true position of landmarks, and the trajectory added noise

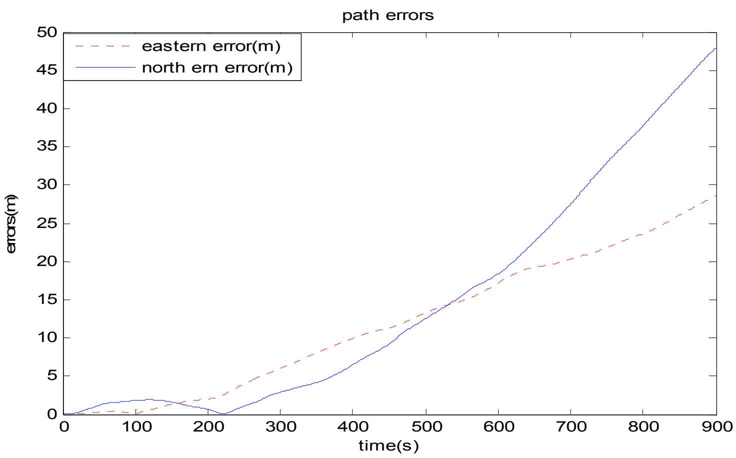

Figure 2. Error of path without using SLAM

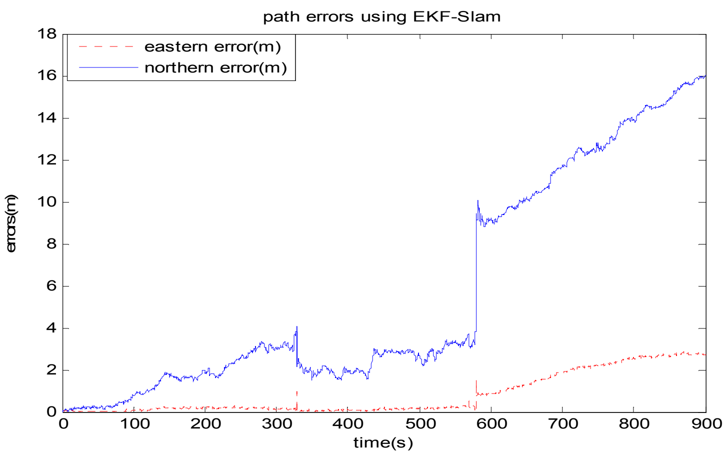

Figure 3. Error of path by using EKFSLAM

This paper shows the results of auxiliary FASTSLAM by comparing pure EKFSLAM. After 900 seconds simulation, it is clear that after using SLAM, the path of robot is corrected well.

Fig.3 is error by implementing the EKFSLAM, where the biggest eastern error is 2.1 meters and the northern 16.2 meters. Fig.5 shows the results after using auxiliary

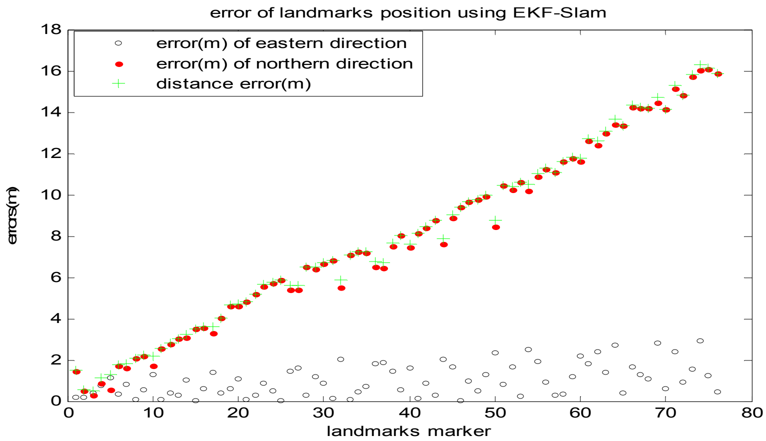

Figure 4. Position error of the landmarks by using EKFSLAM

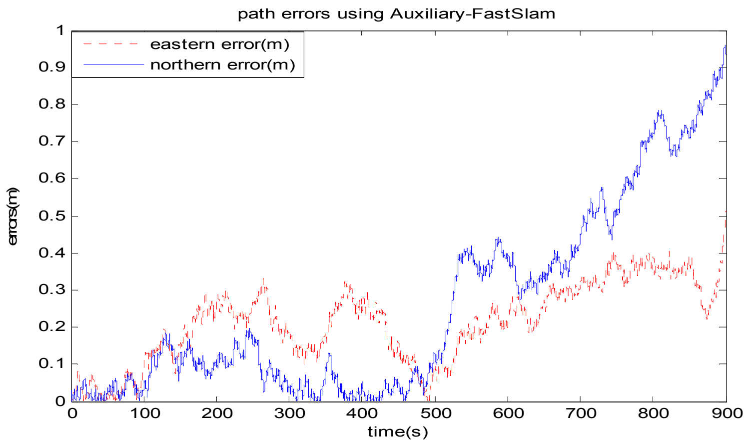

Figure 5. Error of path by using auxiliary FASTSLAM

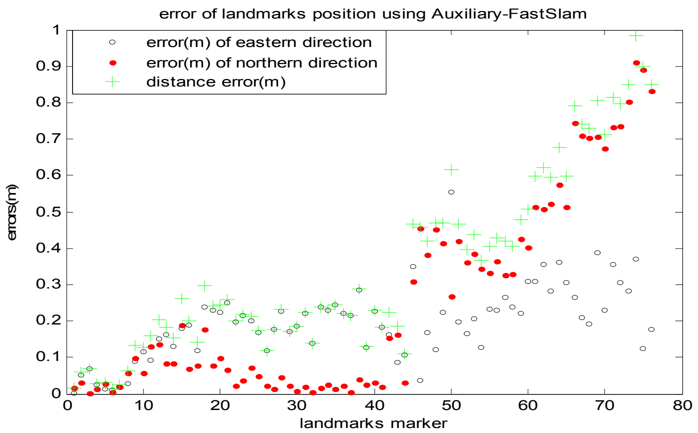

Figure 6. Position error of the landmarks by using auxiliary FASTSLAM

FASTSLAM, where the biggest eastern error is 0.51 meters and the northern 0.96 meters. Fig.4 shows the different position error about landmarks, and the biggest Eustachian distance error is 16.3 meters.

Fig.6 has a smaller distance error that is 1.00 meters. Through table 1, EKFSLAM can diverge easily and auxiliary FASTSLAM keep the estimate more stable and more accurate.

Table 1. Stability of The Two Algorithms

error / R	EKFSLAM			Auxiliary FASTSLAM		
	Eastern path(m)	***Northern path(m)***	***Distance of landmarks(m)***	***Eastern path(m)***	***Northern path（m）***	***Distance of landmarks(m)***
$\begin{matrix} 0.04^2 & 0 \\ 0 & 0.04^2 \end{matrix}$	3.5	17.8	19.7	0.51	0.97	1.00

$\begin{matrix} 0.06^2 & 0 \\ 0 & 0.06^2 \end{matrix}$	13.8	6.0	69.6	0.49	0.95	0.98
$\begin{matrix} 0.08^2 & 0 \\ 0 & 0.08^2 \end{matrix}$	41.4	10.9	42.4	0.51	0.92	0.94
$\begin{matrix} 0.1^2 & 0 \\ 0 & 0.1^2 \end{matrix}$	Diverge	Diverge	Diverge	0.51	0.94	0.97

5. Conclusion And Future Work

Although the dominate EKFSLM can deal with the problem we have confronted, it has low accuracy and stability. This paper presents a new algorithm named auxiliary FASTSLAM to decrease the computational complex and implement easily and make the robot path and map more accurate. And the simulated results also show clearly that auxiliary FASTSLAM is more stable than pure EKFSLAM.

Even though we obtained promising results with our technique, there are still tasks for future work. One potential problem with the specific algorithm is to combine strapdown inertial navigation (SIN) technology to realize the 3-dimention environment which has a higher nonlinear degree. And that will make us get a more flexible algorithm to adapt to more environments.

6. Acknowledgment

This research has been funded in part by the High Technology Research and Development Program of China (No.2006AA09Z231) and Science and Technology Development Program of Shandong Province (No.2008GG1055011).

7. References

[1] R. C. Smith and P. Cheeseman, “On the representation and estimation of spatial uncertainty.” International Journal of Robotics Research, 5(4):56–68, 1986.

[2] N.J. Gordon, D.J. Salmond, and A.F.M. Smith, “A novel approach to nonlinear/non-Gaussian Bayesian state estimation.” Proc IEEE Radar and Signal Processing, Apr 1993, vol 140, pp. 107-113.

[3] A. Doucet, N. de Freitas, and N. Gordon, editors, “Sequential Monte Carlo Methods in Practice.” Springer Verlag, 2001.

[4] A Doucet, N. de Freitas, K. Murphy and S. Russell, “Rao-Blackwellised particle filtering for dynamic Bayesian networks.” Proc UAI Conference on Uncertainty in Artificial Intelligence (UAI 2000), pp. 176-183

[5] Schon T, Gustafsson F, and Nordlund P J, “Marginalized particle filters for mixed linear/nonlinear state-space models”. IEEE Transaction on Signal Processing, , vol 53(7), pp. 2279-2289, 2005

[6] M.Montemerlo and S.Thrun, “Simultaneous localization and mapping with unknown data association using FastSLAM.” Proc IEEE International Conference on Robotics and Automation (ICRA 2003),. Sept. 2003, vol 2, pp. 1985 - 1991, doi:10.1109/ROBOT.2003.1241885

[7] 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.” Proc IEEE International Joint Conference on Artificial Intelligence (IJCAI 2003), Citeseer press ,vol 18, pp: 1151-1156 doi: 10.1109/IROS.2006.282528

[8] T. Sch¨on, F. Gustafsson, and P.-J. Nordlund, “Marginalized particle filters for mixed linear/nonlinear state-space models.” IEEE Transactions on Signal Processing, 2004.

[9] S. Arulampalam and B. Ristic, “Comparison of the particle filter with range parameterized and modified polar EKF’s for angle-only tracking,” Proc. SPIE, April 2000, vol. 4048, pp. 288–299, 2000, doi:10.1117/12.391985

[10] N. Bergman, “Recursive Bayesian estimation: Navigation and tracking applications,” Ph.D. dissertation, Linköping Univ., Linköping, Sweden,1999.

[11] Bo He,Ke Yang,Shuai Zhao and Yitong Wang, ”Underwater Simultaneous localization and mapping based on EKF and point-features,” Proc. IEEE International Conference on Mechatronics and Automation(ICMA 2009). IEEE Press, Aug. 2009, pp. 4845-4850, doi:10.1109/ICMA.2009.5246398