Proceedings of the 2011 IEEE
International Conference on Mechatronics
April 13-15, 2011, Istanbul, Turkey

An Adaptive Neuro-Fuzzy Rao-Blackwellized Particle Filter for SLAM

Ramazan Havangi, Mohammad Teshnehlab, Mohammad Ali Nekoui, Hamid Taghirad

Control Department K. N. Toosi University of Technology
Seyed khandan bridge, Tehran, Iran
Havangi@ee.kntu.ac.ir

Abstract—***The Rao-Blackwellized particle filter SLAM (RBPF-SLAM) that is also known as FastSLAM is a framework for simultaneous localization using a Rao-Blackwellized particle filter. The performance and the quality of the estimation of the Rao-Blackwellized particle filter depends heavily on the correct a priori knowledge of the process and measurement noise covariance matrices (Q_t and R_t) that are in most applications unknown. On the other hand, an incorrect a priori knowledge of Q_t and R_t may seriously degrade their performance. To solve these problems, this paper presents an adaptive Neuro-Fuzzy Rao-Blackwellized particle filter. The free parameters of adaptive Neuro-Fuzzy inference systems are trained using the steepest gradient descent (GD) to minimize the differences of the actual value of the covariance of the residual with its theoretical value as much as possible.***

Keywords—**SLAM, Rao-Blackwellized particle filter, Neuro-Fuzzy.**

I. INTRODUCTION

The two key computational solutions to SLAM are the extended kalman filter (EKF-SLAM) and Rao-Blackwellized particle filter (RBPF-SLAM or FastSLAM). However, EKF-SLAM suffers from two major problems: the computational complexity and data association [1]. FastSLAM algorithm approach has been proposed as an alternative approach to solve the SLAM problem. FastSLAM is an instance of Rao-Blackwellized particle filter, which partitions the SLAM posterior into a localization problem and an independent landmark position estimation problem. There are two versions of FastSLAM in the literature, FastSLAM1.0 [2,4-5] and FastSLAM2.0 [3,4-5]. As FasSLAM2.0 is superior to FastSLAM1.0, we focus on the second version. In FastSLAM, extended kalman particle filter (EKPF) is used for the mobile robot position estimation and EKF is used for the feature location's estimation. The key feature of FastSLAM, unlike EKF-SLAM, is the fact that the data association decisions can be determined on a per-particle basis, and hence different particles can be associated with different landmarks. Each particle in FastSLAM may even have a different number of landmarks in its respective map. This characteristic gives the FastSLAM the possibility of dealing with multi-hypothesis association problem. The ability to simultaneously pursue multiple data associations makes FastSLAM significantly more robust to the data association problems than other algorithms based on incremental maximum likelihood data association such as EKF-SLAM. The other advantage of FastSLAM over EKF-SLAM arises from the fact that particle filters can cope with nonlinear and non-Gaussian robot motion models, whereas EKF approaches approximate such models via linear functions. There have been many investigations on FastSLAM [6-17]. In references [6-8] it has been noted that FastSLAM degenerates over time. This degeneracy is due to the fact that a particle set estimating the pose of the robot loses its diversity. One of the main reasons for loosing particle diversity in FastSLAM is sample impoverishment. It occurs when likelihood lies in the tail of the proposal distribution [3-5]. Researchers have been trying to solve those problems in [7-17]. In all previous research on FastSLAM, it is assumed that a priori knowledge of the process and measurement noise statistics is completely known. However, in most application these matrixes are unknown. On the other hand, an incorrect a prior knowledge of Q_t and R_t may seriously degrade the Rao-Blackwellized particle filter performance. In this paper to solve these problems, an adaptive Neuro-Fuzzy Rao-Blackwellized particle filter is proposed for SLAM.

II. THE SLAM PROBLEM

To describe SLAM, let us denote the map by Θ and the pose of the robot at time t by s_t. The map consists of a collection of features, each of which will be denoted by θ_n and the total number of stationary features will be denoted by N. In this situation, the SLAM problem can be formulized in a Bayesian probabilistic framework by representing each of the robot's position and map location as a probabilistic density function as:

$$p(s_t,\Theta \mid z^t,u^t,n^t) \tag{1}$$

In essence, it is necessary to estimate the posterior density of maps Θ and poses s_t given that we know the observation $z^t=\{z_1,...,z_t\}$, the control input $u^t=\{u_1,...,u_t\}$ and the data association n^t. Here, data association represents the mapping between map points in Θ and observation in z^t. The SLAM problem is then achieved by applying Bayes filtering as follows:

$$p(s_t,\Theta \mid z^t,u^t,n^t) \propto p(z_t \mid s_t,\Theta,n_t)\, p(s_t,\Theta \mid z^{t-1},u^t,n^t) \tag{2}$$

With

$$p(s_t,\Theta \mid z^{t-1},u^t,n^t)=\int p(s_t \mid s_{t-1},u_t)p(s_{t-1},\Theta \mid z^{t-1},u^t,n^t)ds_{t-1} \tag{3}$$

Where $p(s_t \mid s_{t-1},u_t)$ is the dynamics motion model and $p(z_t \mid s_t,\Theta,n^t)$ is the measurement model. The structure of SLAM enables particle filters to be applicable. This special particle filter is known as the Rao-Blackwellized particle filter (RBPF). The RBPF was introduced as an effective means to solve the SLAM problem by Murphy [20]. The term 'Rao-Blackwellized' means factoring of a state into a sampled part and an analytical part. FastSLAM computes the posterior over maps and a robot path. The key mathematical insight of FastSLAM pertains to the fact that the full SLAM posterior can be factorized as follows when data association n^t is known:

$$p(s^t,\Theta \mid z^t,u^t,n^t)=p(s^t \mid z^t,u^t,n^t)\prod_{n=1}^{N} p(\theta_n \mid s^t,z^t,u^t,n^t) \tag{4}$$

Where $s^t=\{s_1,...,s_t\}$ is a robot path. This factorization states that the SLAM problem can be decomposed into estimating the product of a posterior over robot path and N landmark posteriors given the knowledge of the robot's path. The FastSLAM algorithm implements the path estimator $p(s^t \mid z^t,u^t,n^t)$ using a particle filter, and the landmarks pose $p(\theta_n \mid s^t,z^t,u^t,n^t)$ are realized by EKF, using separate filters for different landmarks. Structure of the M particles is as follow:

$$s_t^{[m]}=<s^{t,[m]},\mu_{1,t}^{[m]},\Sigma_{1,t}^{[m]},...,\mu_{N,t}^{[m]},\Sigma_{N,t}^{[m]}> \tag{5}$$

Where $[m]$ indicates the index of the particle, $s^{t,[m]}$ is the m th particle's path estimate, and $\mu_{N,t}^{[m]},\Sigma_{N,t}^{[m]}$ are the mean and the covariance of the Gaussian distribution representing the n th feature location conditioned on the path $s^{t,[m]}$. In FastSLAM2.0, vehicle poses are sampled with respect to both the control u_t and measurement z_t, which is denoted as follows [3-5]:

$$q(s^{t,[m]} \mid z^{t-1},u^t,n^{t-1})=p(s^{t,[m]} \mid z^{t-1},u^t,n^{t-1}) \tag{6}$$

As a result, the FastSLAM2.0 is superior to FastSLAM1.0 in all aspects.

A. *Sampling Strategy in FastSLAM2.0*

In FastSLAM2.0, poses are sampled with respect to both the motion u_t and the measurement z_t. This is formally denoted by the following sampling distribution, which now takes the measurement z_t into consideration:

$$q(s_t^{[m]} \mid s^{t-1,[m]},z^t,u^t,n^t) \tag{7}$$

An effective approach to accomplish this is to use EKF generated Gaussian approximation.

$$q(s_t^{[m]} \mid s^{t-1,[m]},z^t,u^t,n^t)\sim N(s_t;s_t^{[m]},P_t^{[m]}) \tag{8}$$

In this approach, each particle is updated at the measurement time using the EKF according to the following equations:

$$\hat{s}_{t+1}^{[m]}=f(s_t^{[m]},u_t) \tag{9}$$

$$P_{t+1}^{[m]-}=\nabla f_t P_t^{[m]}\nabla f_t^T+\nabla G_u Q_t \nabla G_u^T \tag{10}$$

where

$$\nabla f_t=\left.\frac{\partial f}{\partial s_t}\right|_{s_t=s_t^{[m]}} \qquad \nabla G_u=\frac{\partial f}{\partial u} \tag{11}$$

where

$$Q_t=\begin{bmatrix}\sigma_r^2 & 0\\ 0 & \sigma_\theta^2\end{bmatrix} \tag{12}$$

and

$$K_t^{[m]}=P_t^{[m]-}H_t^T(H_t P_t^{[m]-}H_t^T+R_t)^{-1} \tag{13}$$

$$s_t^{[m]}=\hat{s}_t^{[m]}+K_t^{[m]}(z_k-h(\hat{s}_t^{[m]})) \tag{14}$$

From the Gaussian distribution generated by the estimated mean and covariance of the vehicle, the state of each particle is sampled:

$$s_t^{[m]}\sim N(s_t;s_t^{[m]},P_t^{[m]}) \tag{15}$$

When there is no observation, the vehicle state is predicted without the measurement update using (9) and (10). If many landmarks are observed at the same time, (13) and (14) are repeated for each observed landmark, and the mean and the covariance of the vehicle are updated based on the previously updated one.

B. *Landmark Estimate based EKF*

The FastSLAM represents the posterior landmark estimates $p(\theta_n \mid s^t,z^t,u^t,n^t)$ using low dimensional EKF. In fact FastSLAM updates the posterior over the landmark estimates, respected by the mean $\mu_{n,t-1}^{[m]}$ and the covariance $\Sigma_{n,t-1}^{[m]}$. The updated values $\mu_{n,t}^{[m]}$ and $\Sigma_{n,t}^{[m]}$ are then added to the temporary particle set S_t, along with the new pose. The update depends on whether or not a landmark n was observed at time t. For $n\neq n_t$, the posterior over the landmark remains unchanged as follows [4]:

$$\mu_{n,t}^{[m]}=\mu_{n,t-1}^{[m]} \tag{16}$$

$$\Sigma_{n,t}^{[m]}=\Sigma_{n,t-1}^{[m]} \tag{17}$$

For the observed feature $n=n_t$, the update is specified through the following equation [3, 4]:

$$p(\theta_{n_t} \mid s^{t,[m]},n^t,z^t)= \\ =\eta\,\underbrace{p(z_t \mid \theta_{n_t},s^{t,[m]},n^t,z^{t-1})}_{\sim N(z_t,g(\theta_{n_t},s_t^{[m]},R_t)}\underbrace{p(\theta_{n_t} \mid s^{t,[m]},n^t,z^{t-1})}_{\sim N(\theta_{n_t},\mu_{n,t-1}^{[m]},\Sigma_{n,t-1}^{[m]})} \tag{18}$$

The probability $p(\theta_{n_t} \mid s^{t,[m]},n^t,z^{t-1})$ at time $t-1$ is represented by a Gaussian distribution with mean $\mu_{n,t-1}^{[m]}$ and covariance $\Sigma_{n,t-1}^{[m]}$. For the new estimate at time t to also be Gaussian, FastSLAM linearizes the perceptual model $p(z_t \mid \theta_{n_t},s^{t,[m]},n^t,z^{t-1})$ by EKF. Especially, FastSLAM approximates the measurement function g by the following first degree Taylor expansion [4]:

$$g(\theta_{n_t}, s_t^{[m]}) = \underbrace{g(\mu_{n,t-1}^{[m]}, s_t^{[m]})}_{\hat{z}_t^{[m]}} + \underbrace{g'(s_t^{[m]}, \mu_{n,t-1}^{[m]})}_{G_t^{[m]}}(\theta_{n_t} - \mu_{n,t-1}^{[m]}) \tag{19}$$

$$= \hat{z}_t^{[m]} + G_t^{[m]}(\theta_{n_t} - \mu_{n,t-1}^{[m]})$$

Under this approximation, the posterior of landmark n_t is indeed Gaussian. The mean and covariance are obtained using the following measurement update:

$$\hat{z}_t = g(s_t^{[m]}, \mu_{n_t,t-1}) \tag{20}$$

$$G_{\theta_{n_t}} = \nabla_{\theta_{n_t}} g(s_t, \theta_{n_t})|_{s_t = s_t^{[m]}; \theta_{n_t} = \mu_{n_t,t-1}^{[m]}} \tag{21}$$

$$Z_{n,t} = G_{\theta_{n_t}} \Sigma_{n_t,t-1}^{[m]} G_{\theta_{n_t}}^T + R_t \tag{22}$$

$$K_t = \Sigma_{n_t,t-1}^{[m]} G_{\theta_{n_t}}^T Z_{n,t}^{-1} \tag{23}$$

$$\mu_{n_t,t}^{[m]} = \mu_{n_t,t-1}^{[m]} + K_t(z_t - \hat{z}_t) \tag{24}$$

III. The Adaptive Neuro-Fuzzy Rao-Blackwellized Particle filter for FastSLAM

As already motioned, in the conventional Rao-Blackwellized particle filter, complete a priori knowledge of the process and measurement noise statistics is assumed (matrices Q_t and R_t). However, in most applications these matrixes are unknown. An incorrect a priori knowledge of Q_t and R_t may lead to performance degradation. This is because EKF is used in the design of the proposal distribution and landmark position estimation. On the other hand, the performance of EKF depends largely on the accuracy of the knowledge of process covariance matrix Q_t and measurement noise covariance R_t. An incorrect a priori knowledge of Q_t and R_t may lead to performance degradation [18-19] and it can even lead to practical divergence [18-19]. One of the efficient ways to overcome the above weakness is to use an adaptive algorithm. Two major approaches that have been proposed for adaptive EKF are multiple model adaptive estimation (MMAE) and innovation adaptive estimation (IAE) [18-19]. Although the implementation of these approaches is different, they both share the same concept of utilizing new statistical information obtained from the residual (innovation) sequence. The covariance matrix R_t represents the accuracy of the measurement instrument. The enlargement of the covariance matrix R_t for measured data means that we trust this measured data less and more on the prediction. As landmarks are static, we adapt the covariance matrix R_t when updating the landmark position. Hence, the algorithm to tuning the measurement noise covariance R_t can be derived. The adaptation adaptively adjusts the measurement noise covariance matrix R_t by using an adaptive Neuro-fuzzy System. In this case, an innovation based adaptive estimation (IAE) algorithm to adapt the measurement noise covariance matrix R_t is derived. The technique known as covariance-matching is used. The basic idea behind this technique is to make the actual value of the covariance of the residual to be consistent with its theoretical value. The innovation sequence $r_t = (z_t - \hat{z}_{n_t,t})$ has a theoretical covariance that is obtained from the EKF algorithm

$$S_{t,EKF} = G_{\theta_{n_t}} \Sigma_{n_t,t-1}^{[m]} G_{\theta_{n_t}}^T + R_t \tag{25}$$

The actual residual covariance $\hat{C}_t$ can be approximated by its sample covariance, through averaging inside a moving window of size N as follows:

$$\hat{C}_t = \frac{1}{N} \sum_{i=k-N+1}^{t} (r_i^T r_i) \tag{26}$$

If the actual value of covariance $\hat{C}_t$ has discrepancies with its theoretical value, then the diagonal elements of R_t based on the size of this discrepancy can be adjusted. The objective of these adjustments is to correct this mismatch as far as possible. The size of the mentioned discrepancy is given by a variable called the degree of mismatch (DOM_t), defined as

$$DOM_{t,EKF} = S_{t,EKF} - \hat{C}_t \tag{27}$$

The basic idea used to adapt the matrix R_t is as follows: from equation (25) an increment in R_t will increase S_t and vice versa. Thus, R_t can be used to vary $S_{t,EKF}$ in accordance with the value of $DOM_{t,EKF}$ in order to reduce the discrepancies between $S_{t,EKF}$ and $\hat{C}_t$. The adaptation of the (i,i) th element of R_t is made in accordance with the (i,i) th element of $DOM_{k,EKF}$. The general rules of adaptation are as follows:

If $DOM_{t,EKF}(i,i) \cong 0$ then maintain R_t unchanged.

If $DOM_{t,EKF}(i,i) > 0$ then decrease R_t.

If $DOM_{t,EKF}(i,i) \cong 0$ then increase R_t.

In this paper the IAE adaptive scheme of the EKF, coupled with adaptive Neuro-fuzzy inference system (ANFIS) is presented to adjust R_t. The overall adaptive Neuro-fuzzy inference system employs a bank of subsystems where each subsystem employs a two-input-single-output ANFIS. This is because the dimensions of DOM_k and R_k are both 2×2. These two-input-single-output Neuro-fuzzy inference systems are employed to tune each diagonal of element of R_k

A. *The ANFIS Architecture*

The ANFIS model has been considered as a two-input-single-output system. This ANFIS is a five layers network as shown in Fig.1. The inputs of ANFIS are $DOM_{t,EKF}$ and $DeltaDOM_{t,EKF}$. Here, $DeltaDOM_{t,EKF}$ is defined as:

$$DeltaDOM_{t,EKF} = DOM_{t,EKF} - DOM_{t-1,EKF} \tag{28}$$

Finally, adjustment of R_t is performed using the following relation:

$$R_t = R_t + \Delta R_t \tag{29}$$

where ΔR_t is the ANFIS output. The fuzzy rules which complete the ANFIS rule base are as in Table 1.

B. Learning Algorithm

The aim of the training algorithm is to adjust the network weights through the minimization of the following cast function:

$$E = \frac{1}{2} tr(e_t^2) \tag{30}$$

Where

$$e_t = S_{t,EKF} - \hat{C}_t \tag{31}$$

By using the Bake Propagation (BP) learning algorithm, the weighting vector of ANFIS is adjusted such that the error defined in (30) is less than a desired threshold value after a given number of training cycles. The well-known BP algorithm may be written as:

$$W(t+1) = W(t) - \eta \frac{\partial E(t)}{\partial W(t)} \tag{32}$$

Here η and W represent the learning rate and tuning parameter of ANFIS, respectively.

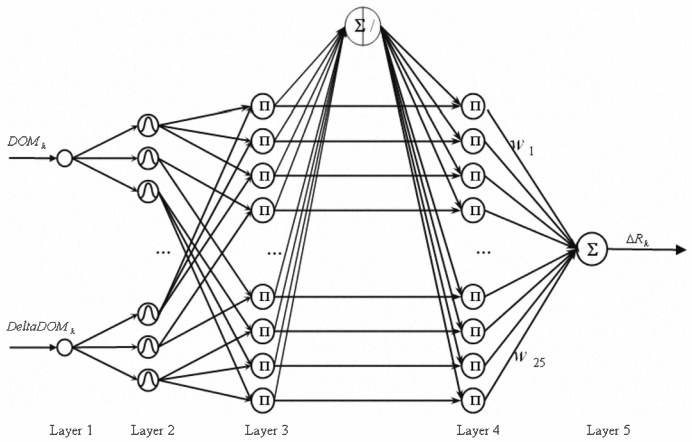


Fig .1 The Neuro-fuzzy system for feature update

TABLE .1 Rule Table

DeltaDOM$_k$ \ DOM$_k$	L	LM	Z	HM	H
L	S7	S7	S6	S5	S4
LM	S7	S6	S5	S4	S3
Z	S6	S5	S4	S3	S2
HM	S5	S4	S3	S2	S1
H	S4	S3	S2	S1	S1

IV. IMPLEMENTATION AND RESULTS

Simulation experiments have been carried out to evaluate the performance of the proposed approach in comparison with the classical method. The proposed solution for the SLAM problem has been tested for the benchmark environment, with varied number and position of the landmarks, available in [21].

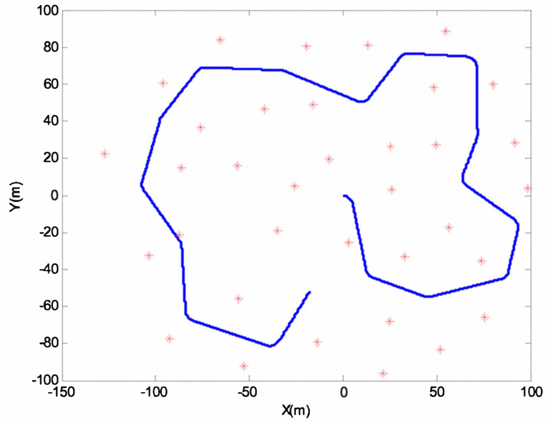


Fig.2 The experiment environment: The star point "*" denote the landmark positions and blue line is the path of robot

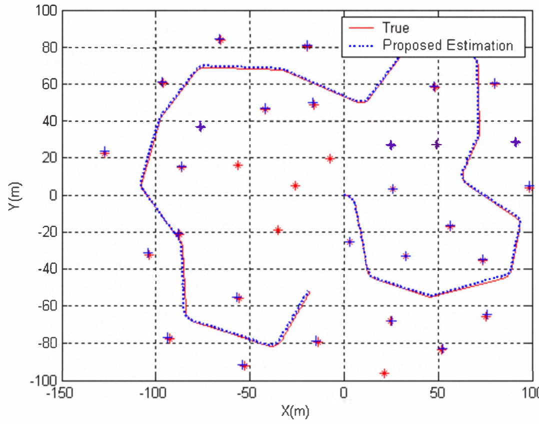
Fig.2 shows the robot trajectory and landmark location. The star points (*) depict the location of the landmarks that are known and stationary in the environment. The state of the robot can be modeled as (x, y, ϕ) where (x, y) are the Cartesian coordinates and ϕ is the orientation respective to the global environment. The kinematics equations for the mobile robot are as the following form:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} (V + v_v)\cos(\phi + [\gamma + v_\gamma]) \\ (V + v_v)\sin(\phi + [\gamma + v_\gamma]) \\ \frac{(V + v_v)}{B}\sin(\gamma + v_\gamma) \end{bmatrix} \tag{33}$$

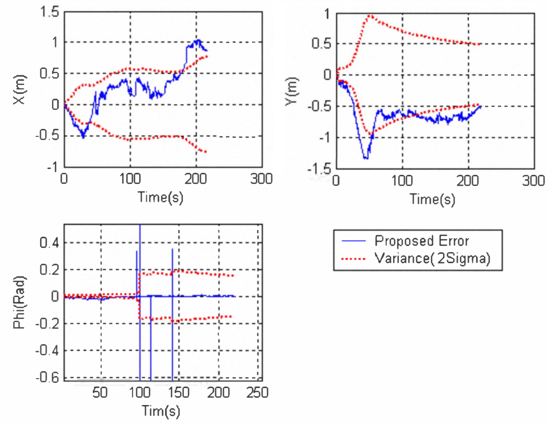
Where B is the base line of the vehicle and $u = [V \quad \gamma]^T$ is the control input at time t consisting of a velocity input V and a steer input γ. The process noise $v = [v_v \quad v_\gamma]^T$ is assumed to be Gaussian. The vehicle is assumed to be equipped with a range-bearing sensor that provides a measurement of range r_i and bearing θ_i to an observed feature ρ_i relative to the vehicle. The observation z of feature ρ_i in the map can be expressed as:

$$\begin{bmatrix} r_i \\ \theta_i \end{bmatrix} = \begin{bmatrix} \sqrt{(x - x_i)^2 + (y - y_i)^2} + \omega_r \\ \tan^{-1}\frac{y - y_i}{x - x_i} - \phi + \omega_\theta \end{bmatrix} \tag{34}$$

Where (x_i, y_i) is the landmark position in map and $W = [\omega_r \quad \omega_\theta]^T$ relates to observation noise. The initial position of the robot is assumed to be $x_0 = 0$. The robot moves at a speed 3m/s and with a maximum steering angle of 30 degrees. Also, the robot has 4 meters wheel base and is equipped with a range bearing sensor with a maximum range of 20 meters and a 180 degrees frontal field-of-view. The control noise is $\sigma_v = 0.3$ m/s and $\sigma_\gamma = 3^o$. A control frequency is 40 HZ and observation scans are obtained at 5 HZ . The measurement noise is 0.1 m in range and 0.1^o in bearing. Data association is assumed uknown. To evaluate the proposed method the performance of it is compared with FastSLAM2.0 for the benchmark environment. We consider the situation where measurement noise is wrongly considered as $\sigma_r = 0.7$, $\sigma_\theta = 1.0$ .The performance of the proposed method is compared with FastSLAM2.0 where its measurement covariance matrix R_t is kept static throughout the experiment with 35 landmarks. The proposed algorithm starts with a wrongly known statistics and then adapts the R_t in EKPF and EKF through ANFIS and attempts to minimize the mismatch between the theoretical and actual values of the innovation sequence in EKF and EKPF. The free parameters of ANFIS are automatically learned by GD during training. Fig.3 and Fig.5 show the comparison between the proposed algorithm and the FastSLAM2.0. It can be clearly seen that the results of the proposed algorithm are better than that of FastSLAM2.0. In other words, in the proposed algorithm, the estimated vehicle path and estimated landmark coincide as closely as possible with the actual path and the actual positions landmarks. This is because the proposed method adaptively tuned the measurement covariance matrix R_t . In fact, matrix R_t converges to the actual covariance matrix R_t while matrices R_t in FastSLAM2.0 is kept fixed over time. Also Fig.4 and Fig.6 show that measurement covariance matrix measurement R_t converges to the actual covariance matrix R_t in our proposed method.



(a)



(b)

Fig.3 Proposed method: In this experiment measurement noise is wrongly considered as $\sigma_r = 0.7$, $\sigma_\theta = 1.0$ and control noise is truly considered as $\sigma_v = 0.3$ m/s , $\sigma_\gamma = 3^o$. Also, number of particles is 20 (n=20).Also, the results obtain over 50 Monte Carlo runs. a) Estimated robot path and estimated landmark with true robot path and true landmark. The "…" is the estimated path, the "+" are the estimated landmark positions. b) Estimated pose error with $2-\sigma$ bound.

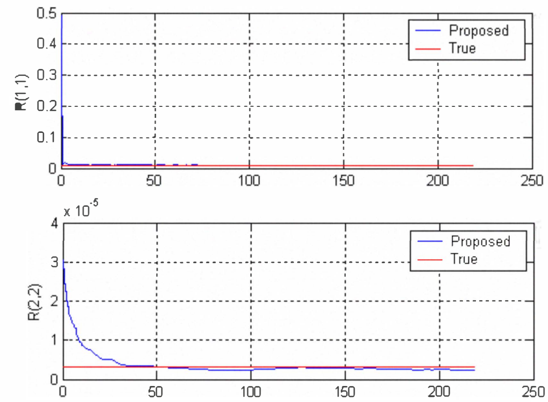
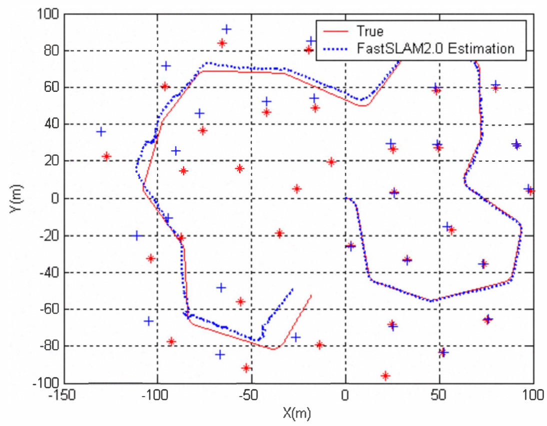
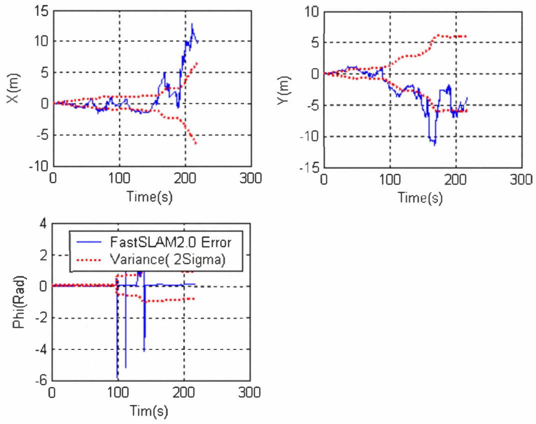


Fig.4 Proposed Method (n=20)



(a)



(b)

Fig.5 FastSLAM2.0: In this experiment measurement noise is wrongly considered as $\sigma_r = 0.7$, $\sigma_\theta = 1.0$ and control noise is truly considered as $\sigma_v = 0.3$ m/s , $\sigma_\gamma = 3^o$. Also, number of particles is 20 (n=20). Also, the results obtain over 50 Monte Carlo runs. a) Estimated robot path and estimated landmark with true robot path and true landmark. The "…" is the estimated path, the "+" are the estimated landmark positions. b) Estimated pose error with $2-\sigma$ bound.

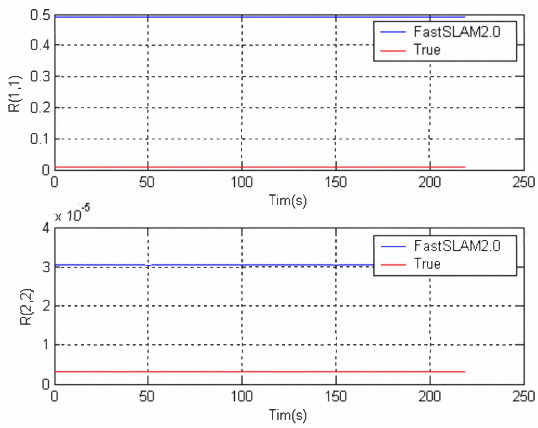


Fig.6 FastSLAM2.0 (n=20)

V. CONCLUSION

This paper presents an adaptive Neuro-Fuzzy Rao-Blackwellized particle filter. In the proposed method, a Neuro-Fuzzy EKPF for robot pose estimation, and a Neuro-Fuzzy EKF for landmark feature estimation is developed. The main advantage of our proposed method is its more consistency than classical FastSLAM2.0. This is because in our proposed method, the theoretical value of the innovation sequence matches with its real value.

REFERENCES

[1] S.Thrun, W. Burgard, D. Fox, "Probabilistic robotics", MIT Press, Cambridge, 2005.

[2] M. Montemerlo, S. Thrun, D. Koller, B.Wegbreit, "FastSLAM: a factored solution to the simultaneous localization and mapping problem", Proc. of AAAI national conf. Artif. Intell., Edmonton, AB, Canada, 2002.

[3] M.Montemerlo, S.Thrun, D.Koller, B.Wegbreit, "Fastslam 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges", in the Int. Joint Conf. on Artificial Intelligence, 2003.

[4] S. Thrun, M. Montemerlo, D. Koller, B.Wegbreit, J.Nieto, E. Nebot, "FastSLAM: An efficient solution to the simultaneous Localization and mapping problem with unknown data association", Journal of Machine Learning Research, 2004.

[5] M.Montemerlo, "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association", PhD thesis, Carnegie Mellon University, 2003.

[6] T.Bailey, J. Nieto and E.Nebot, "Consistency of the FastSLAM algorithm". IEEE Int. Conf, on Robotics and Automation.pp.424-429, 2006.

[7] L. Zhang, X. Meng, Y. Chen, "Convergence and Consistency analysis for FastSLAM", Intelligent Vehicles Symposium, 2009.

[8] N. Kwak, I. K. Kim, H. C. Lee, B. H. Lee, "Adaptive Prior Boosting Technique for the Efficient Sample Size in FastSLAM", in the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (San Diego, CA) pp. 630–635, 2007.

[9] N.Kwak, G.Kim,B.Lee, "A new compensation technique based on analysis of resampling process in FastSLAM", Journal of Robotica, Vol.26, pp. 205–217, 2007.

[10] I.Kim, N.Kwak, H.Lee, B.Lee, "Improved particle fusing geometric relation between particles in FastSLAM", Journal of Robotica, Vol. 27, pp. 853–859, 2009.

[11] C. Kim, R. Sakthivel, W. K. Chung, "Unscented FastSLAM: A Robust algorithm for the simultaneous localization and mapping problem", in IEEE Int. Conf. Robot. Autom., pp.2439–2445, 2007.

[12] C. Kim, R.Sakthivel, W. K. Chung, "Unscented FastSLAM: A Robust and Efficient Solution to the SLAM Problem", IEEE Trans. Robot., Vol. 24, No. 4, 2008.

[13] M. Cugliari , F. Martinelli, "A FastSLAM algorithm based on the unscented filtering with adaptive selective resampling", Field and Service Robotics, vol. 42, pp. 359–368, Jun, 2008.

[14] Yi-min Xia, Yi-min Yang, "A Simultaneous Localization and Mapping Algorithm of Mobile Robot Based on Improved FastSLAM", Global Congress on Intelligent Systems, IEEE Computer society, 2009.

[15] X.Chen, "An Adaptive UKF-Based Particle Filter for Mobil Robot SLAM", International Joint Conference on Artificial Inteligence, 2009.

[16] X. Wang, H. Zhang, "A UPF-UKF framework for SLAM", IEEE Intl. Conf. on Robotics and Automation, pp 1664-1669, 2007.

[17] M.Li, H.Bing-rong, L.Rong-hua, W.Zhen-Hua, "A novel method for mobile robot simultaneous localization and mapping", Journal of Zhejiang university Scince A, P937-944, 2006.

[18] R. J. Fitzgerald, "Divergence of the Kalman filter", IEEE Trans. Autom. Control, Vol. AC-16, No. 6, pp. 736–747, Dec. 1971.

[19] R. K. Mehra, "On the identification of variances and adaptive Kalman filtering", IEEE Trans Autom Control, Vol. AC-15, No.2, pp. 175–184, Apr, 1970.

[20] K.Murphy, "Bayesian map learning in dynamic environments", Proc. of the Conf. on Neural Information Processing Systems (NIPS), Denver, CO, USA, pp.1015–1021, 1999.

[21] Online Avaiable:http://www.personal.acfr.usyd.edu.au/tbailey/software / index .html, 2008.