ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

COMPUTATIONAL COST ANALYSIS OF EXTENDED KALMAN FILTER IN SIMULTANEOUS LOCALIZATION & MAPPING (EKF-SLAM) PROBLEM FOR AUTONOMOUS VEHICLE

Saiful Bahri Samsuri, Hairi Zamzuri, Mohd Azizi Abdul Rahman, Saiful Amri Mazlan and Abdul Hadi Abd Rahman
Malaysia-Japan International Institute of Technology (MJIIT), Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia
E-Mail: saiful_bahri_samsuri@yahoo.com

ABSTRACT

Extended Kalman filter (EKF) based solution is one of the most popular techniques for solving simultaneous localization and mapping (SLAM) problem. However, previous research showed the implementation of EKF for SLAM suffered with high computational costs, which affect the performance in real time application. This paper investigates the computational cost performance of an EKF-SLAM algorithm. The analysis was done by time measurement on sub-step motion update and measurement update on EKF by considering the total numbers of landmarks and numerous setting on range observation distance. The analytical results show that as the number of landmarks or range observation distances increased, the computational cost in measurement update step required more computation time compare to motion update step. Furthermore, improvements are needed to optimize the computational cost for the update step.

Keywords: extended kalman filter, slam, computational cost, autonomous vehicle.

INTRODUCTION

Simultaneous localization and mapping (SLAM) is the operation of the robot building a map of an environment, while simultaneously localizing itself in it. SLAM was introduced by Smith [1], which was the first to focus on measurement error correlations during the map building process. Various different solutions [2–6] for SLAM have been proposed during the past 20 years. Extended Kalman Filter, (EKF) algorithm have been widely used as a solution to SLAM problem, [7, 8] has been formulated and solved as a hypothetical problem in a number of different structures. It's had been widely implemented in SLAM for over a decade. However, its suffers several problems such as data association, convergence, linearity and computational complexity [9, 10]. Several work on improvement of EKF SLAM had been done such as Unscented Kalman Filter [2], Sparse Extended Information Filter [11], Compressed Extended Kalman Filter [12] and FastSlam [13, 14]. However, there are still some limitations, especially on computational cost during real time, and intern of kalman filter update step [10]. In order to improve these problems, specific analysis is required. Several analysis on EKF had been done such as consistency and position error analysis [15,16] the author in [17, 18] worked on computational complexity analysis by derivation. However, there is no previous work on CPU runtime analysis.

In this paper, computational analysis on EKF algorithm by CPU runtime measurement will be covered. Through explicit analysis, our results showed that the number of landmarks and laser range observation distances are the factors to cause computational cost in measurement update step and required more computation time compared to motion update step which is as the observed number of features or landmark which when increased, the computational required also increased. This paper is organized as follows. The next section is general introduction of SLAM. Then, the third section briefly describes the EKF algorithm and a complexity analysis were derived. Finally, the simulation and analysis results are presented in the fourth section.

SIMULTANEOUS LOCALIZATION AND MAPPING (SLAM)

SLAM describes the process of building a map of an unknown environment and simultaneously computing robot position with the constructed map. The autonomous vehicle is equipped with a set of proprioceptive, odometry to compute vehicle's pose that includes its orientation (yaw) and x, y – coordinates of its position (2D case) and exteroceptive sensors, a range and bearing sensors that measure the distance to the landmark together with its bearing with respect to the autonomous vehicle's current frame. Originate the most common model used to achieve the second task of the SLAM mapping problem. Noise introduced by the imperfection of sensors makes the already difficult task of SLAM more complicated.

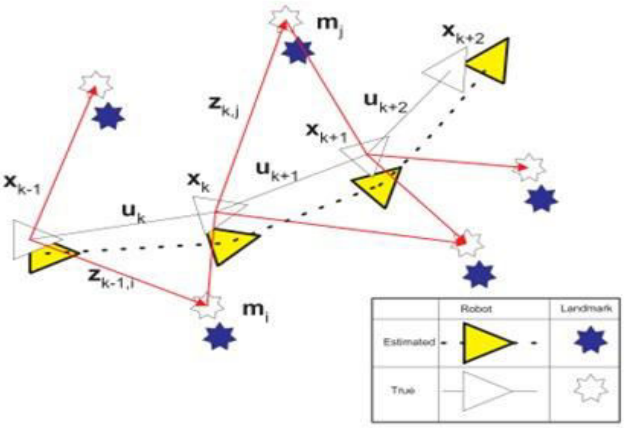

Figure-1. SLAM [8, 19].

VOL. 10, NO. 17, SEPTEMBER 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

An autonomous vehicle navigates in the environment while taking measurements of landmarks within the sensing range. Figure-1 shows a simultaneous estimate of both robot and landmark locations where the true locations are never known or measured directly. Observations are made between the true robot and landmark locations. At each step-k an autonomous vehicle possesses (1) the mean vector that contains an autonomous vehicle's estimated pose (x_k) and estimated positions of detecting landmarks, (m_k), (2) control vector applied in a previous step (u_k) and (3) a vector of observations taken at k^{th} step (z_k). In probabilistic form, the solution of the SLAM problem is represented a probability distribution, $P(x_k,m|Z_{0:k},x_0)$ where (a.) $Z_{0:k}=\{z_1,z_2,z_3 \ldots z_k\}$;a sequence of all observations,(b.) $U_{0:k}=\{u_1,u_2,u_3\ldots u_k\}$;a sequence of applied control inputs and x_0 ; an autonomous vehicle's initial pose. A probability distribution describes a joint posterior density of an autonomous vehicle's pose and landmarks' positions given the initial state of an Autonomous vehicle, a sequence of applied control inputs, and a sequence of taking observations. Computation of a joint posterior is based on Bayes rule. At any step k, a previous joint posterior (x_{k-1}, $m|Z_{0:k-1}$, $u_{0:k-1}$, x_0), a vector of control inputs, and a vector of taking measurements are known. The current autonomous vehicle's pose is assumed to depend only on the previous pose and the current control input. The motion model $P(x_k|x_{k-1},u_k)$ is based on vehicle model. The measurement model $P(z_k|x_k,m)$ is defined by the sensor model. Kalman filter (KF) was the first technique to implement the bayes rule in SLAM, but it suffers from linearization thus EKF was introduced to overcome the problem.

COMPUTATIONAL COMPLEXITY EKF-SLAM ALGORITHM

The Extended Kalman Filter (EKF) is an alternative of a Bayesian filter for SLAM [10]. In EKF SLAM, a map (x,Σ) includes the state distribution x to be estimated, which consists of the current vehicle location and the landmarks position in environment features. The covariance of the distribution, represented by Σ, gives an idea of the precision in the estimation. EKF SLAM is repetitive process from motion update to measurement update, the full process of EKF SLAM summarized in Table-1. The detail has been derived by [17,18].

Table-1. Ziegler-Nichols formula for oscillatory response method.

Motion Update	Prediction	$x_{t\|t-1} = g(x_t, x_{t-1})$	$O(1)$
		$\Sigma_{t\|t-1} = F_t \Sigma_{t-1} F_t^T + G_t R_{t-1} G_t^T$	$O(n)$
Measurement Update	Innovation	$v_t = z_t - h(x_{t\|t-1})$	$O(r)$
		$S_t = H_t \Sigma_{t\|t-1} H_t^T + Q_t$	$O(r^3)$
	χ^2	$D^2 = v_t^T S_t^{-1} v_t$	$O(r^3)$
	Update	$K_t = \Sigma_{t\|t-1} H_t^T / S_t$	$O(nr^2)$
		$\Sigma_t = (I - K_t H_t)\Sigma_{t\|t-1}$	$O(n^2 r)$
		$x_t = x_{t\|t-1} + K_t v_t$	$O(n)$

Variable n is the current state of x_t which contains the current pose and total landmark that have been observed, and r is the size of the measurement vector z_t or the current number of landmark observed at the current vehicle pose. The test x_2 is only required for data association [20]. During exploratory trajectories, the autonomous vehicle observes r features that have been observed from the previous vehicle position. As the new features detected it will keep it to the state vector, x_t thus the size of the map n grows linearly. Computational complexity of each EKF update step is O (n^2) and the total cost known to be O (n^3). Given the limited speed, angle, range and bearing, the Jacobian matrix as shown in Table-2:

Table-2. Jacobians required for the EKF.

Jacobians	
$F_t = \dfrac{\partial g(x_t, x_{t-1})}{\partial x_{t-1}} \|\hat{x}_{t-1}$	$O(1)$
$G_t = \dfrac{\partial g(x_t, x_{t-1})}{\partial x_t} \|\hat{x}_t$	$O(1)$
$H_t = \dfrac{\partial g(u_t, u_{t-1})}{\partial u_{t\|t-1}} \|\hat{u}_{t\|t-1}$	$O(1)$

Variable r is the size of the measurement vector z_t [20]. The fact regarding computational complexity in EKF SLAM is that, Jacobians matrices F_t,G_t and H_t are sparse [17]. One of the most time consuming part is the matrix

 ISSN 1819-6608

ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

inversion in S_t^{-1} with a complexity of approximately $O(r^{2.4})$ [10,21].

RESULTS AND DISCUSSIONS

Based on the theorems above, we identify the effect on computational time with a variation of the number of features or landmark and laser range. The parameter that was considered is number of landmarks and the range of laser observation where this parameter is able to extend the size of each matrix in EKF. In the following experiments, the vehicle model wheelbase and vehicle speed is 4 meters and 17 m/s, the control noise is ($\sigma V = 0.2$m/s, $\sigma\gamma = 1°$), and the observation noise is ($\sigma r = 0.1$m, $\sigma\theta = 1°$). The controls are updated at 100 Hz and observation scans are obtained every 50Hz. Data association is assumed unknown throughout. All the experiments are performed on a computer with two Intel Core i5 CPUs at 3.3 GHz with 4GB RAM. Our implementation is coded in C and the algorithm was run 5 times. The system stopwatch was used to measure the computational time on every each step in EKF. In this experiment only the matrix that required the higher computational complexity from Table 1 is measured and the 2 types of the scenario were considered which are:

Scenario 1: CPU average runtime by various setting of laser observations range with 50 landmarks

Scenario 2: CPU average runtime by various number of landmarks with fixed 30m laser observation range.

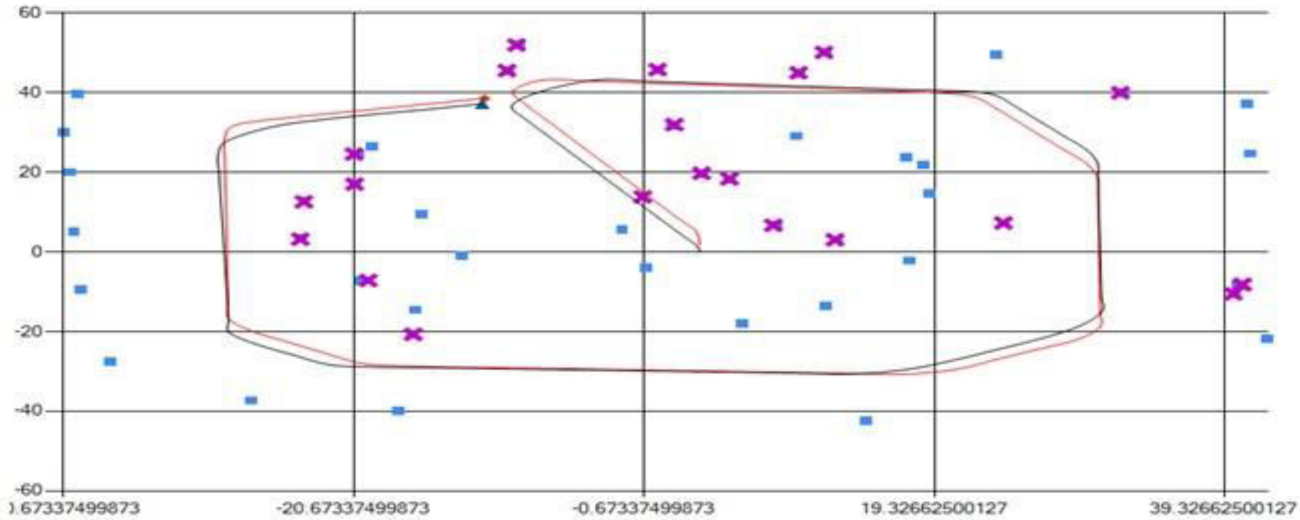

Figure-2. Example of the final map builds by EKF SLAM with 50 Landmarks and 10m laser range.

The example of final map created by EKF SLAM simulation is shown in Figure-2. The '■' represents the natural features or landmark of the environment while the '**X**' represents the landmark estimated by EKF from the current vehicle pose. As the vehicle move, any natural feature such as trees and lamp posts which are located within observation range from the current vehicle pose will be used to compute the location of landmarks (features) and the current position of the vehicle.

The results show that as the number of landmarks or range observation distances increased, the computational cost in measurement update step required more computation time compared to motion update step. That caused by a number of steps required to compute in measurement update is higher than motion update. The theoretical output derived from Table-1 shows that the variable n and r were able to modify the size of the matrix, thus affecting the speed of computation time during real time exploration. This verified the result show in Figure-3 which, as the number of landmarks observed is more than 3, the computations required for Σ, K and S starts to increase. This is caused by quadratic component such as Σ required $O(n^2r)$, K required $O(nr^2)$ and S is $O(r^3)$. If there is no feature detected or is out of the observation range the variable r will reset to empty or (r=0). This is opposite for variable n, where it will remain from the previous iteration.

CONCLUSIONS

In this paper, the effect on computational time with variation of the number of features or landmark and laser range was investigated. The computational cost on each step in EKF was measured by using system stopwatch. The two types of scenario such as variation on the number of landmark and variation on laser observation range have been considered which effect the computational time. Through analysis, we got the results of effects of variation of the number of features or landmark and laser range on computational time. We showed that the number of landmarks and range observation distances are the factors to cause computational cost in measurement update step and required more computation time compared to the motion update step. Moreover, the information covariance matrix,

ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

Σ and Kalman gain, K required higher time computational compared with other step in EKF. Furthermore, improvements are needed to optimize the computational cost for the update step especially on Σ covariance matrix and Kalman gain, K.

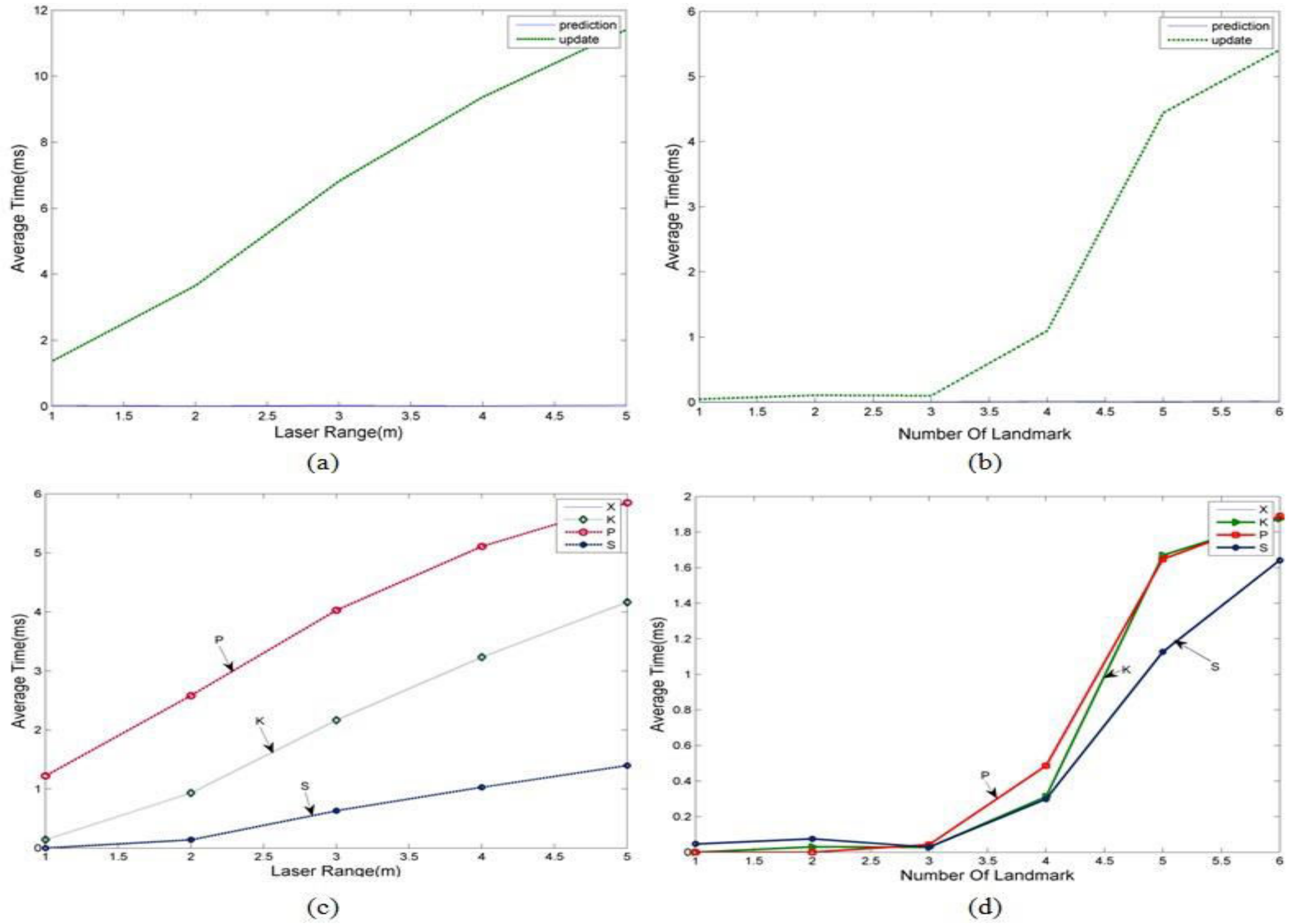

Figure-3. Computational cost for scenario 1(left) and Computational cost for scenario 2 (right).

ACKNOWLEDGEMENTS

The authors would like to thank Malaysia-Japan International Institute of Technology (MJIIT) in UTM Kuala Lumpur, Ministry of Education Malaysia (MOE) and Universiti Teknologi Malaysia under University Research Grant (vote 05H70) for funding and supporting this research.

REFERENCES

[1] R. Smith, M. Self, P. Cheeseman and M. Park. 1988. A stochastic map for uncertain spatial relationships the stochastic map. In: Proceedings of International Symposium of Robotics Research. Santa Clara. pp. 467−474.

[2] E. A. Wan, R. Van Der Merwe and N. W. W. Rd. 2000. The unscented Kalman filter for nonlinear estimation. In: Adaptive Systems for Signal Processing Communications, and Control Symposium.

[3] M. Montemerlo, S. Thrun, D. Koller and B. Wegbreit. 2002. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In: Proceeding of 8th National Conference on Artificial Intelligence/14th Conference on Innovative Applications of Artificial Intelligence. pp. 593–598.

[4] E. M. Nebot and S. Member. 2001. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. Vol. 17, No. 3, pp. 242–257.

[5] S. Thrun. 2003. Results for outdoor-SLAM using sparse extended information filters. In: 2003 IEEE Int. Conf. Robot. Autom. (Cat. No.03CH37422). pp. 1227–1233.

[6] M. Montemerlo and S. Thrun. 2006. Large-scale robotic 3-D mapping of urban structures. Springer Tracts Adv. Robot. Vol. 21, pp. 141–150.

[7] H. Durrant-whyte and T. Bailey. 2006. Simultaneous localisation and mapping (SLAM): Part I The Essential Algorithms. IEEE Robotics and Automation Magazine. Vol. 2, pp. 1–9.

[8] T. Bailey and H. Durrant-whyte. 2006. Simultaneous localisation and mapping (SLAM): Part II State of the Art, Robotics and Automation Magazine, pp. 1–10.

[9] L. M. Paz and J. D. Tard. 2008. Divide and conquer : EKF SLAM in O(n). Robot. IEEE Trans. Vol. 24, No. 5, pp. 1107–1120.

[10] S. Thrun. 2002. Probabilistic robotics. Communications of the ACM. Vol. 45.

[11] S. Thrun and Y. Liu. 2005. Multi-robot SLAM with Sparse Extended Information Filers, in Robotics Research. Vol. 15, pp. 254–266.

[12] E. Guivant. 2002. Efficient simultaneous localization and mapping in large environments. CEKF PhD Thesis.

[13] M. Montemerlo, S. Thrun, D. Koller and B. Wegbreit. 2003. FastSLAM 2.0 : An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In: Proceedings of IJCAI.

[14] M. Montemerlo and S. Thrun. Simultaneous localization and mapping with unknown data association using FastSLAM. IEEE Int. Conf. Robot. Autom. (Cat. No.03CH37422). Vol. 2.

[15] T. Bailey, J. Nieto, J. Guivant, M. Stevens and E. Nebot. 2006. Consistency of the EKF-SLAM algorithm. In: IEEE International Conference on Intelligent Robots and Systems. pp. 3562–3568.

[16] G. P. Huang, A. I. Mourikis and S.I. Roumeliotis. 2008. Analysis and improvement of the consistency of extended Kalman filter based SLAM. IEEE International Conference on Robotics and Automation. pp. 473–479.

[17] L. M. Paz, J. D. Tardós and J. Neira. 2008. Divide and conquer: EKF SLAM in O(n). IEEE Trans. Robot. Vol. 24, pp. 1107–1120.

[18] V. Bonato, V. Bonato, E. Marques, E. Marques and G. A. Constantinides. A floating-point extended Kalman filter implementation for Autonomous Mobile Robots. J. Signal Process. Syst. Vol. 56, pp. 41–50.

[19] B. Y. H. Durrant-whyte, T. I. M. Bailey, P. Cheeseman, J. Crowley and H. Durrant. 2006. Simultaneous localization and mapping: Part I, Robotics & Automation Magazine. IEEE. Vol. 13, No. 2, pp. 99–110.

[20] C. Dario and C. Lerma. 2011. Efficient simultaneous localisation and mapping in large and complex environments. University of Zaragoza. PhD Thesis.

[21] C. Graphics and H. Bischof. 2009. Simultaneous localisation and mapping for mobile robots with recent sensor technologies. Graz University of Technology. Master Thesis.