

SIMULATION AND EXPERIMENTAL EVALUATION OF THE EKF SIMULTANEOUS LOCALIZATION AND MAPPING ALGORITHM ON THE WIFIBOT MOBILE ROBOT

Noura Ayadi¹ and Nabil Derbel¹, Nicolas Morette², Cyril Novales² and Gérard Poisson²

¹*Department of Electrical Engineering, National School of Engineering, B.W 3038, Sfax, Tunisia*

²*PRISME Institute, 63, Lattre de Tassigny, Bourges, France*

Submitted: 24th January 2017; Accepted: 24th March 2017

Abstract

In recent years, autonomous navigation for mobile robots has been considered a highly active research field. Within this context, we are interested to apply the Simultaneous Localization And Mapping (SLAM) approach for a wheeled mobile robot. The Extended Kalman Filter has been chosen to perform the SLAM algorithm. In this work, we explicit all steps of the approach. Performances of the developed algorithm have been assessed through simulation in the case of a small scale map. Then, we present several experiments on a real robot that are proceeded in order to exploit a programmed SLAM unit and to generate the navigation map. Based on experimental results, simulation of the SLAM method in the case of a large scale map is then realized. Obtained results are exploited in order to evaluate and compare the algorithm's consistency and robustness for both cases.

Keywords: mobile robot, localisation, EKF, SLAM, consistency

1 Introduction

Relatively to the artificial intelligence, cartography is a highly crucial step for localization and autonomous navigation of a mobile robot. Starting from an initial position, a robot relies on its exteroceptive sensors to build a map for the navigation environnement. The robot localizes itself and learns to react in function of existing constraints. But obtaining the most suitable map is not always obvious for several reasons. On one hand, programming the map manually is a difficult and time requiring operation. Besides, every observer has its own perspective. On the other hand, the navigation environnement is usually dynamic and affected by

frequent changements. In this context, Simultaneous Localization And Mapping (SLAM) is considered a solution for described restrictions. SLAM first appeared with Smith, Self and Cheeseman [17] and developed by Leonard and Durrant-Whyte [6]. The general idea of SLAM is an observation of the whole environnement in a closed loop. The robot must assess when to observe every existing object from a known and a precise position. So, the algorithm's task is to update the robot's position every sample time depending on its current position. This way, we guarantee less uncertainty on obtaining information by estimation. Yet, another source of uncertainty comes from odometry. Odometric errors are very common in mobile robots navigation.

In order to avoid them, we rely on different exteroceptive sensors. In our case, the laser telemeter is the chosen sensor providing the robot's position. In the literature, the SLAM approach has proved its efficiency and robustness especially when it applies the Extended Kalman Filter (EKF). The EKF SLAM is the first technique to be implemented and every later solution has been drifted from it. It has proved its efficiency when applied for several types of robots [19]. The main characteristic of the conventional EKF is the fact of its simple implementation and operating with white gaussian noise. Several improved versions of the EKF based SLAM appear in recent works. We can indeed mention the Compressed Extended Kalman Filter (CEKF) which divides the state vector into an active and passive parts. When the robot is in a local area, the algorithm updates only the active part. Once the robot moves, the whole update is performed. The Unscented Kalman filter (UKF) is another modified version of the classic EKF. The UKF avoids deriving the Jacobian matrices and seeks to reduce estimation errors [2, 10]. The Augmented Kalman Filter (AKF) tends to improve the algorithm accuracy and compensate systematic errors essentially odometric ones. It creates and estimates a new system of odometric parameters [5, 22]. Limitations of the EKF are noted especially in the case of large scale navigation environments. Thus, several approaches appeared for SLAM such as the Fast SLAM, tiny SLAM and Linear SLAM. The Fast SLAM is considered the most successful algorithm, it uses a Rao-Blackwellized particle filter and operates with non-gaussian noises. Unlike the EKF SLAM, fast SLAM relies on posterior estimation over a robot's poses [13, 14, 18]. As for the tiny SLAM, it uses the Incremental Maximum Likelihood (IML) and based on two effective optimization methods: genetic and Monte Carlo algorithms in order to improve the approach's accuracy and speed. However, for large scale and complicated environments, the algorithm becomes slower and gives questionable results [16]. For large scale maps, the linear SLAM appears as a significant solution. This method is based on joining submaps already obtained by conventional SLAM techniques. Significant results are shown by solving a sequence of linear least squares problems [4, 8, 21].

In spite of the fact that the EKF is the most used technique, we can not deny the efficiency of rela-

tively newer approaches as the use of genetic algorithms. Several approaches appear with multiple techniques as the Island model Genetic Algorithm [11] or the use particle filter and steady-state genetic algorithm [7, 15].

The rest of this work is organized as follows: Section 2 is deals with a general presentation of the SLAM approach. We focus on the difference between localisation and cartography leading to the appearance of SLAM. In Section 3, we deal with simulation of the EKF SLAM algorithm behaviour. The robot is located in the map in presence of all existing features and tracking a defined trajectory. Section 4 describes real tests and experiments realized on a mobile robot to exploit Core SLAM, an integrated unit on the robot. The purpose is generating the navigation map as it is seen by the robot during its movement. Simulation of the SLAM method basing on the obtained map is realized. Results are then analysed to give an idea about the consistency of the EKF SLAM. Finally, conclusions are presented in Section 5.

2 Presentation of SLAM Basis

On one hand, the localisation process relies essentially on dead reckoning, sensors feedback and control results. The robot position is then estimated combining all results within a probabilistic distribution. In spite of the highly precise position, computation is very complex. Besides, we discard in the process all characteristics of the navigation environment. In the other hand, the cartography is the step of building a map of the navigation environment by exploiting exteroceptive sensors data. However, carrying out manually the cartography is considered as a hard operation. In this case, the robot position depends on avoidable changes and the dynamic aspect of navigation areas. Both localisation and cartography limitations and inaccuracy are overcome with SLAM. The robot gains the ability to locate itself and distinguish the most suitable instant in order to observe existing landmarks. In this way, localisation and cartography are operating simultaneously which represent the basis of SLAM. The probabilistic distribution to perform SLAM is presented by the following expression

$$P(x_k, M | Z_{0:k}, U_{0:k}, x_0), \quad (1)$$

with:

- $X_{0:k} = \{x_0 \dots x_k\}$: state vector,
- $Z_{0:k} = \{z_0 \dots z_k\}$: observation vector,
- $U_{0:k} = \{u_0 \dots u_k\}$: Control law vector,
- M : vector of landmarks positions.

A simplified SLAM performance can be translated into a scheme described by the Figure 1.

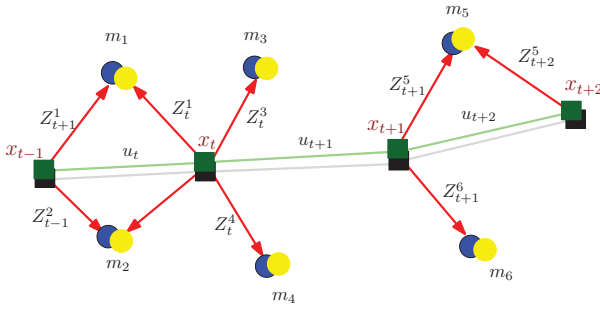


Figure 1. SLAM performance

3 EKF SLAM Simulation

In this part, an EKF SLAM based simulation is realized for a mobile robot. The robot follows a specific trajectory.

3.1 Extended Kalman Filter Algorithm Formulation

EKF SLAM depends on existing elements on the navigation environment called landmarks so it can update the robot's position [12]. Both robot and landmarks positions are kept in guard by the algorithm while processing.

We explicit the implementation of the EKF SLAM for a robot navigation in a defined environment with stationary features. The process model depends essentially on both robot and map landmarks coordinates presented respectively by X_t and

M_t . It is defined by the state vector which dimension is equal to $(2n + 3)$.

$$X = f(X_t, M_t) = \begin{bmatrix} x \\ y \\ \theta \\ x_{l1} \\ y_{l1} \\ \vdots \\ x_{ln} \\ y_{ln} \end{bmatrix} \quad (2)$$

(x, y, θ) is the robot position while $(x_{a1}, y_{a1}, \dots, x_{an}, y_{an})$ define n landmarks positions.

During the robot movement, we control both of its speeds: linear and angular ones presented by $u = (v, \omega)$.

The navigation equation is then defined by:

$$f(X_t, u) = \begin{bmatrix} x + vdt \cos \theta \\ y + vdt \sin \theta \\ \theta + \omega dt \\ x_{l1} \\ y_{l1} \\ \vdots \\ x_{ln} \\ y_{ln} \end{bmatrix}$$

As a result, we obtain the corresponding Jacobian matrix:

$$A = \begin{pmatrix} 1 & 0 & -\sin \theta & 0 & \dots & 0 \\ 0 & 1 & \cos \theta & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & \dots & \dots & 0 \end{pmatrix}$$

For every observed landmark " l_i ", we calculate its relative position using the observation model:

$$h_i(X_t) = \begin{bmatrix} \sqrt{(x - x_{li})^2 + (y - y_{li})^2} \\ \arctan\left(\frac{y_{li} - y}{x_{li} - x}\right) - \theta \end{bmatrix}$$

The resulted jacobian matrix is then presented by:

$$H_i = \begin{pmatrix} \frac{x - x_{ai}}{\sqrt{(x - x_{ai})^2 + (y - y_{ai})^2}} & \frac{y - y_{ai}}{\sqrt{(x - x_{ai})^2 + (y - y_{ai})^2}} & 0 \\ -\frac{y - y_{ai}}{(x_{ai} - x)^2 + (y_{ai} - y)^2} & \frac{x - x_{ai}}{(x_{ai} - x)^2 + (y_{ai} - y)^2} & -1 \end{pmatrix}$$

To estimate both robot and landmark positions, we follow Extended Kalman Filter steps.

- **State prediction:** The predicted state estimate is expressed as:

$$X_t^* = f(\hat{X}_{t-1}, u_t)$$

- **Covariance prediction:** The predicted covariance estimate is given by:

$$P_t^* = A \cdot \hat{P}_{t-1} \cdot A^T + Q$$

- **Observation prediction:** This step handles prediction's measurement or innovation.

$$Y_t^* = h(X_t^*)$$

- **Observation:** Measurements and data association are realized. Thus, estimated positions and noise are obtained.

- **Correction of the predicted state:** In this step, we obtain both updated state and covariance estimates.

$$\begin{aligned} \hat{X}_t &= X_t^* + K(Y - Y_t^*) \\ \hat{P}_t &= P_t^* - KHP_t^* \end{aligned}$$

where K is the Kalman gain:

$$K = P_t^* H^T (H \cdot P_t^* \cdot H^T + P_Y)^{-1}$$

3.2 Simulation Results: Case of a Small Scale Map

A simulation-based analysis of the EKF algorithm for the SLAM method is realised in this Section. We aim to show the robot's behaviour while tracking a defined way through the EKF SLAM algorithm. We define in this simulation a set of waypoints for the robot to reach. Stationary landmarks are presented in dots in presence of a squared obstacle. The robot starts to move with a speed of 2 m/s from its initial position defined by (30, 25) and a initial orientation equal to 0° .

The trajectory is presented in Figure 2.

All along the trajectory, both real and estimated robots positions are drawn with triangles and following each other. Every time a landmark is seen, uncertainty gaussian ellipses are applied around it.

Described features of the map are presented by Figure 3.

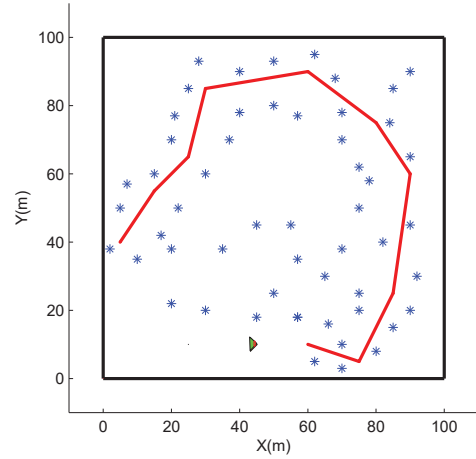


Figure 2. Defined trajectory and landmarks

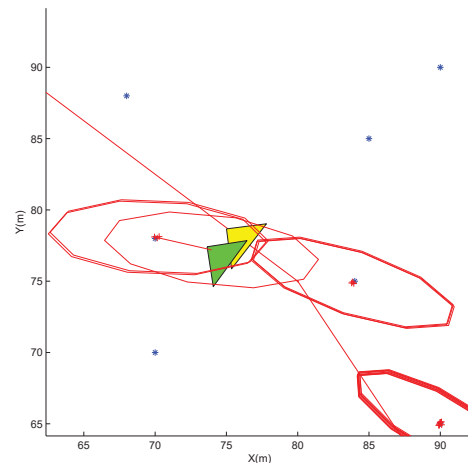


Figure 3. Real and estimated robots positions are presented with triangles. All seen landmarks are surrounded by gaussian ellipses.

Performances of EKF SLAM is evaluated for two different cases. First, the robot's perception of landmarks is limited. Subsequently, only 16 of 54 defined landmarks are seen. The resulted loop is executed in 27.3 seconds. Figure 4 presents simulation results.

We increase in a second experience the perception range of the robot sensors. The program becomes slower which is justified by taking into account 36 landmarks. Results of this second case are shown in Figure 5.

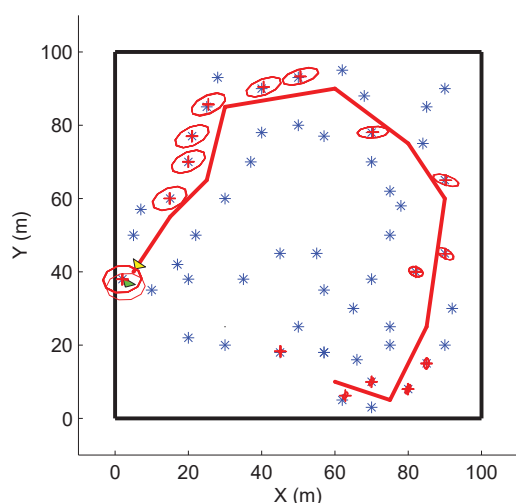


Figure 4. Navigation path resulted from SLAM algorithm for a restricted range of perception. The defined trajectory is presented by continuous line.

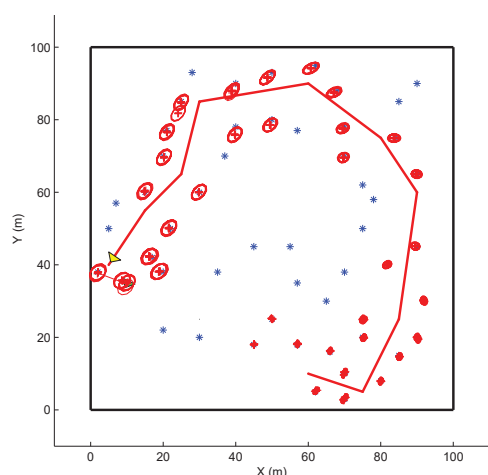


Figure 5. Navigation path resulted from SLAM algorithm. The defined trajectory is presented by continuous line.

4 SLAM Application on a Mobile Robot

4.1 Presentation of the Robotic System

Experimental tests are realized on Wifibot, a well designed robot offering to the user a platform to develop and manipulate different applications in an easy way. The Wifi board ensures a wireless connection of the system with the configured access point which is provided freely. It uses the Hokuyo *UTM-30LX* laser sensor of the robot which covers 270° and allows to detect all objects from a highly long distance. Control and measure acquisitions are

easier and more reliable using the laser telemeter sensor. By doing 40 scans/second, it can achieve an accuracy of 0.25° . SLAM algorithm is applied on Wifibot through the Core SLAM unit. This unit has been developed by pioneered researchers in robotic applications. It uses the laser telemeter's measures and odometric data in order to built the navigation map.

Several units are available and compatible to be connected to Core SLAM unit. They allow the user to follow and localize the Wifibot during its movement.

Once units are connected and the robot starts to move, we obtain a display of the navigation map with the robot. Figure 6 describes the resulted map with white and grey colours and where we can find the robot whose orientation is indicated by an arrow.



Figure 6. The observed environnement through Core SLAM units

4.2 Experimental Tests

In order to read the obtained map, we needed to exploit all inputs and outputs and proceed with several tests on the robot.

The Core SLAM unit's output named *MAPS_characteristics* (size, res, orig) is an array of 4 components describing the map. We have:

- map_size = 2048,
- res_xy = 20,
- map_origin_x = 20480,
- map_origin_y = 20480.

Knowing that we have a map of 2048 pixel with a resolution of $20\text{mm}/\text{pixel}$, we obtain a square with

a length equal to 40960 mm. The map's origin is situated in the coordinate center (20480, 20480). Realized tests on Wifibot lead a great amount of data which makes very difficult to understand the way of Core SLAM map's allocation in the memory. So, we have limited our experience to test the robot by proposing 3 different positions around only 100 pixels of the map. These three positions are defined as follows:

- $1023 \times 2048 + 1023$: central position,
- $1000 \times 2048 + 1023$: left position,
- $1046 \times 2048 + 1023$: right position.

An obstacle is indeed put for the realized test and situated in the level of the central and left positions. Figure 7 explains these 3 robot's positions in presence of the obstacle.

Simulation display of Core SLAM map is presented in Figure 8.

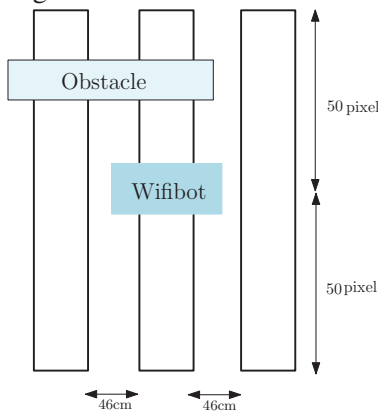


Figure 7. Proposed test to obtain a sample of Core SLAM data



Figure 8. Observed environment by the robot though Core SLAM units

We can notice that the robot is already in the center of the map. White colour is obviously the

space that the robot can see while the grey is an undefined area in the environment. The obstacle is clearly situated in front and on the left of the robot. For previous 3 defined positions, we obtain 3 sets of values where $\{0, 128, 255\}$ are repeated several times. In figures 9, 10 and 11, we present corresponding values for each position.

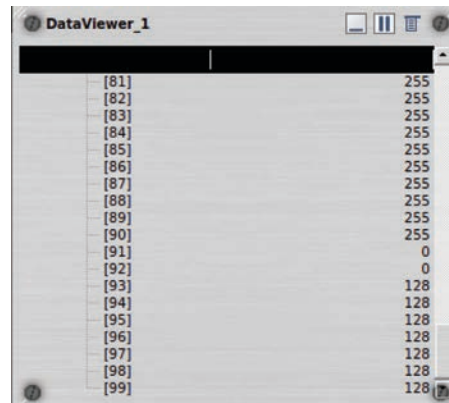


Figure 9. Results of Core SLAM simulation for the robot's central position

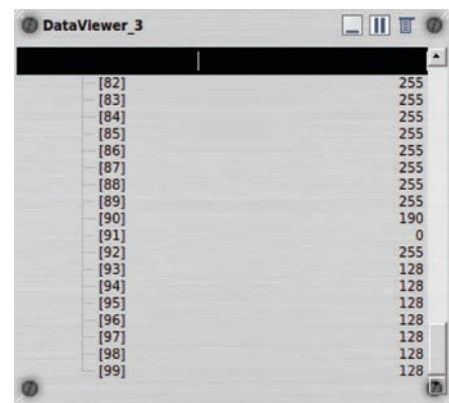


Figure 10. Results of Core SLAM simulation for the robot's left position

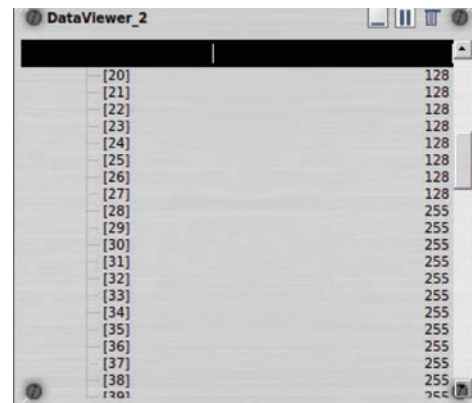


Figure 11. Results of Core SLAM simulation for the robot's right position

Realized experiences make it easier to read the map and understand the way how Core SLAM unit works. As we already know both robot and obstacle's positions, understanding those values is not a difficult task and Core SLAM unit is no longer considered as a black box. We conclude that for values:

- 0 \rightarrow 128: presence of an obstacle,
- 128: undefined state for the robot,
- 128 \rightarrow 255: free space for navigation.

The map is being read from right side to the left side and from the bottom to the top. The achieved experience to exploit Core SLAM unit demonstrates that it is a very useful in robotic mapping and localization especially in uncertain environment.

Resulted map based from unit experiments is considered a large scale map. Simulation of EKF SLAM performances have been realized on these new dimensions. As the purpose of the EKF is to predict the robot position, figures 12 and 13 show real and estimated trajectories respectively comparing to the one resulted from defined waypoints.

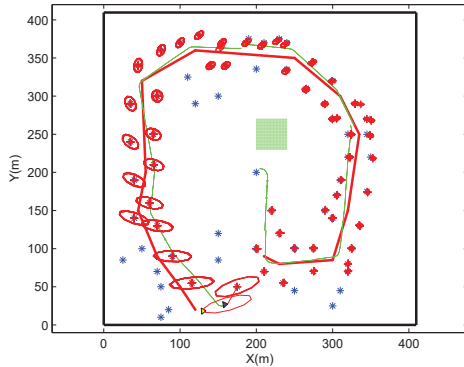


Figure 12. Real trajectory

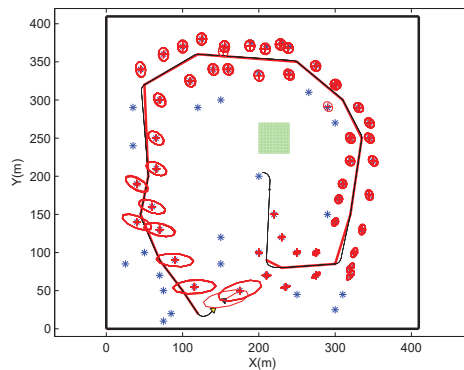


Figure 13. Estimated trajectory

4.3 Evaluation on SLAM Behaviour

As the robot moves, uncertainty increases. Besides, the environment is considered a large scale map. Consequently, we experience failure and inevitable inconsistency in the EKF SLAM algorithm. In several works, consistency of the algorithm is evaluated by checking the difference between the true state covariance P_k and the estimated state covariance $P_{k|k}$. If:

$$P_{k|k} - P_k \geq 0$$

then the filter is consistent. Otherwise, it is not consistent. This method is used when $P_{k|k}$ and P_k are available [3]. In our case, it is difficult to know the covariance state unlike the process model which we are able to fix both of its real and estimated states. For that, we consider the Normalized Estimation Error Squared (NEES) to evaluate the consistency of SLAM algorithm [20]. with:

$$NEES = (\hat{X}_k - X_k)^T P_k^{-1} (\hat{X}_k - X_k)$$

In order to evaluate the performance of EKF SLAM, we rely on Monte Carlo simulation technique. This method is very useful to test estimators consistency or compare estimators on the basis of bias and precision.

Thus, we calculate the average NEES for N runs. The NEES is noted ϵ_k and given by equation 3. The average NEES is then presented as follows:

$$\bar{\epsilon}_k = \frac{1}{N} \sum_{i=1}^N \epsilon_{ik} \quad (3)$$

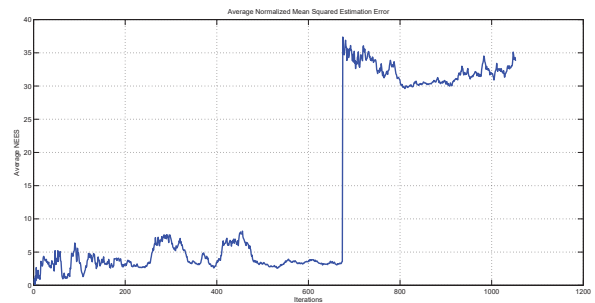


Figure 14. Average Normalized Mean Squared Estimation Error

In Figure 15, we present the heading performance during the robot's navigation. Comparison between real and estimated headings is considered a tool for detecting inconsistencies in the behavior of the EKF SLAM.

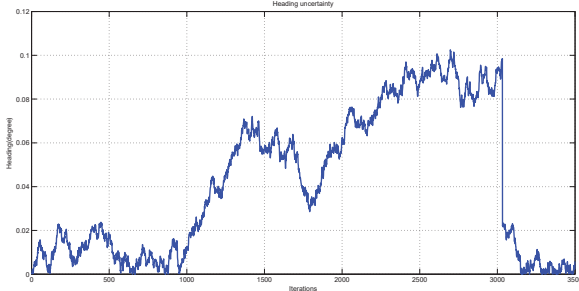


Figure 15. Headings uncertainty

The headings' curve is rising with several spikes which indicate failure of the EKF SLAM. Consequently, we can conclude that at every time, the heading difference is smaller, the EKF SLAM is more consistent. In the case of large scale maps, the algorithm inconsistency is frequent and due to several elements. It can be caused by observation time or measurements and process noises as proved by many studies [1, 3, 9]. To prevent the inflation of the headings difference and improve the filter performance, optimization techniques are required to be integrated in the algorithm.

4.4 Consistency Test for EKF SLAM

We present a necessary condition for consistency based on the chi-square distribution test of a linear gaussian filter. $N\bar{\epsilon}_k$ has a χ^2 density with $Ndim(X_k)$ degrees of freedom. It is primordial to satisfy for N runs the following condition

$$\bar{\epsilon}_k \in [r_1, r_2]$$

where r_1 and r_2 represent limits of the tolerated probability region. For 100 runs, the 95% probability concentration region for $\bar{\epsilon}_k$ has to be bounded by the interval cutting 5% from the upper limit. Relying on the gaussian table of distribution, the region can be defined by $[0, \chi^2(0.095)] = [0, 5.99]$.

The obtained result is presented in Figure 16.

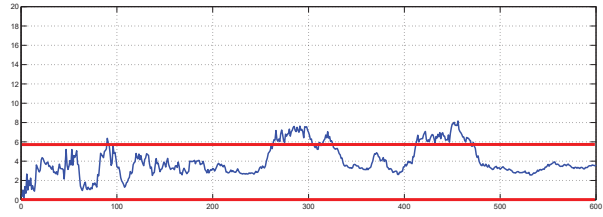


Figure 16. Consistency region for SLAM algorithm

If $\bar{\epsilon}_k$ rises higher than the upper bound, the filter is considered too optimistic and if it tends below the lower bound, the filter is too conservative or too pessimistic. In both ways, the filter is counted inconsistent.

As Figure 16 shows, we conclude that 95% of the average NEES is found in the tolerance region for 100 runs. Nevertheless, when observing the whole curve described by Figure 14, it is clear that the EKF SLAM algorithm is inconsistent as it rises in a significant way higher the upper bound value beyond the iteration 700 besides the presence of an important spike.

In fact, the inconsistency of EKF SLAM is a frequent problem especially for large scale maps. Measurements and observation processes are main factors besides of existing noisy data.

4.5 Comparison Between EKF SLAM Consistency for Both Small and Large Scaled Maps

In order to evaluate more the SLAM algorithm, we realize the consistency test for the case of the small map. Figure 17 presents the average NEES for 100 runs. The maximum value doesn't rise upper than the tolerance region defined by $[0, \chi^2(0.095)] = [0, 5.99]$.

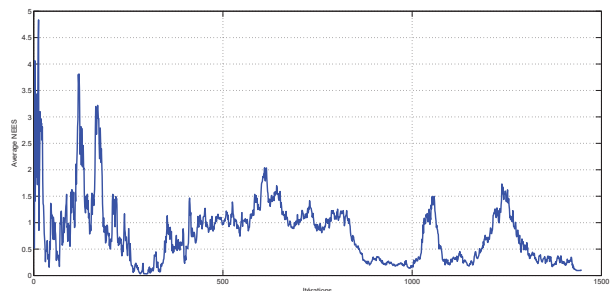


Figure 17. Average NEES for a small scaled map

In the other hand, the orientation uncertainty is presented in Figure 18. The maximal value of the orientation difference doesn't exceed 0.025° .

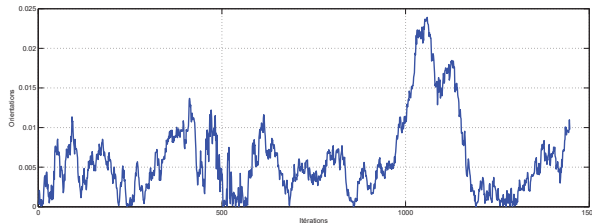


Figure 18. Orientation uncertainty for a small scaled map

The curves show clearly that the filter in this case is more consistent for longer iterations. As a result, the robot navigates in a more stable way and the EKF SLAM will be more performant and robust.

5 Conclusion

In this paper, we have come to explicit a SLAM algorithm based on the Extended Kalman Filter. We have simulated the EKF SLAM on a small scale map. Then, we have introduced an experimental method to explore a Core Slam unit package integrated in a mobile robot. The unit is no longer a black box and its use becomes easier. Obtained large scale map was subject of evaluating performances of SLAM algorithm. Results show that the proposed method is very useful in tracking a defined trajectory although the difference between real and estimated data indicates inaccuracy of the algorithm. For consistent behaviour, the chi-square distribution is being used to test the algorithm performance. The average Normalized Estimation Error Squared besides heading curves show disturbances in addition of several spikes. We conclude that the consistency of the EKF SLAM is unavoidable especially in the case of large scale maps. As a future work, we should consider an optimization approach to make the SLAM algorithm more consistent and thereafter, to implement it to be tested on a real mobile robot.

References

- [1] G. Bresson, R. Aufre, R. Chapuis, A general consistent decentralized Simultaneous Localization And Mapping solution, *Robotics and Autonomous Systems*, Vol. 74, Part A, 2015, pp. 128–147.
- [2] G. Tuna, K. Gulez, V.C. Gungor and T.V. Mumcu, Evaluations of Different Simultaneous Localization and Mapping (SLAM) Algorithms, *Conference on IEEE Industrial Electronics Society*, 2012, pp. 2693–2698.
- [3] H.P. LI, D.M. XU, F.B. ZHANG and Y. YAO, Consistency Analysis of EKF-based SLAM by Measurement Noise and Observation Times, *Acta Automatica Sinica*, Vol. 35, No. 9, 2009, pp. 1177–1184.
- [4] J. Cheng, J. Kim, J. Shao and W. Zhang, Robust linear pose graph-based SLAM, *Robotics and Autonomous Systems*, Vol. 72, 2015, pp. 71–82.
- [5] J.G. Kang, W.S. Choi, S.Y. An and S.Y. Oh, Augmented EKF based SLAM method for Improving the Accuracy of the Feature Map, *International Conference on Intelligent Robots and Systems*, 2010, pp. 3725–3731.
- [6] J.J. Leonard and H.F. Durrant-Whyte, Mobile robot localization by tracking geometric beacons, *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 3, 1991, pp. 376–382.
- [7] J. Woo and N. Kubota, Simultaneous Localization and Mapping using a Robot Partner in Dynamic Environment, *SICE Annual Conference*, 2011, pp. 524–529.
- [8] L. Zhao, S. Huang and G. Dissanayake, Linear SLAM: A Linear Solution to the Feature-based and Pose Graph SLAM based on Submap Joining, *International Conference on Intelligent Robots and Systems*, 2013, pp. 24–30.
- [9] L.D. Rodriguez, F. Matia, L. Pedraza, A. Jimenez and R. Galan, Consistency of SLAM-EKF Algorithms for Indoor Environments, *Journal of Intelligent and Robotic Systems*, Vol. 50, No. 4, 2007, pp. 375–397.
- [10] L. D'Alfonso, W. Lucia, P. Muraca and P. Pugliese, Mobile robot localization via EKF and UKF: A comparison based on real data, *Robotics and Autonomous Systems*, Vol. 74, Part A, 2015, pp. 122–127.
- [11] M. Begum, G.K.I Mann and R.G. Gosine, An Evolutionary SLAM Algorithm for Mobile Robots, *International Conference on Intelligent Robots and Systems*, 2006, pp. 4066–4071.

- [12] M. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte and M. Csorba, A Solution to the Simultaneous Localization and Map Building (SLAM) Problem, *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 3, 2001, pp. 229–241.
- [13] M. Montemerlo, S. Thrun, D. Koller and B. Wegbreit, FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem, *8th National Conference on Artificial Intelligence*, 2002, pp. 593–598.
- [14] 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, *18th international joint conference on Artificial intelligence*, 2003, pp. 1151–1156.
- [15] N. Kubota, K. Yuki and N. Baba, Integration of Intelligent Technologies for Simultaneous Localization and Mapping, *International Joint Conference (ICCS-SICE)*, 2009, pp. 4981–4986.
- [16] O. Hamzaoui and B. Steux, SLAM Algorithm with Parallel Localization Loops: tinySLAM 1.1, *International Conference on Automation and Logistics*, 2001, pp. 137–142.
- [17] R. Smith, M. Self and P. Cheesemans, P, Estimating Uncertain Spatial Relationships in Robotics, *Proceedings of the Second Conference on Uncertainty in Artificial Intelligence (UAI)*, 1986, pp. 267–288.
- [18] S.M. Lee, J. Jung, S. Kim, I.J. Kim and H. Myung, DV-SLAM (Dual-Sensor-Based Vector-Field SLAM) and Observability Analysis, *IEEE Transactions on Industrial Electronics*, Vol. 62, No. 2, 2015.
- [19] S. Wen, X. Chen, C. Ma, H.K. Lam, S. Hua, The Q-learning obstacle avoidance algorithm based on EKF-SLAM for NAO autonomous walking under unknown environments, *Robotics and Autonomous Systems*, Vol. 72, 2015, pp. 29–36.
- [20] T. Bailey, J. Nieto, J. Guivant, M. Stevens and E. Nebot, Consistency of the EKF-SLAM Algorithm, *International Conference on Intelligent Robots and Systems*, 2006, pp. 3562–3568.
- [21] T. Bailey and H. Durrant-Whyte, Simultaneous Localisation and Mapping (SLAM): Part II State of the Art, *IEEE Robotics and Automation Magazine*, 2006, pp. 108–117.
- [22] Y. Nakamura and Y. Fujimoto, Validation of SLAM without Odometry in Outdoor Environment, *IEEE 13th International Workshop on Advanced Motion Control (AMC)*, 2014, pp. 278–283.



Noura Ayadi received the Electrical Engineering Degree from the National School of Engineers of Sfax, Tunisia in 2009, the Research Master Degree in control and industrial computer science in 2010 and the Doctorate on electrical engineering in 2016 at the same Engineering School. Currently, she is a researcher in the Control and Energy

Management Laboratory in the Electrical Department of the National School of Engineers of Sfax. Her research interests include wheeled robots control, autonomous navigation and focused on mapping and localization.



Nabil Derbel received the engineering Diploma from the National School of Engineers of Sfax in 1986, the Research Master Degree in automatic control from the Institut National des Sciences Appliquées de Toulouse in 1986, the Doctorate from the National School of Engineers of Tunis in 1989. He joined the Tunisian University,

where he held different position involved in research and education. Currently, he is a full Professor of Automatic Control at the National School of Engineers of Sfax. He is an IEEE Senior member. His current interests include: Optimal Control, Complex Systems, Fuzzy Logic, Neural Networks, Genetic Algorithm. He is the author and the co-author of more

than 50 papers published in international Journals and of more than 300 papers published in international conferences.



Nicolas Morette worked in the Laboratoire PRISME of the University of Orleans (France) between 2005 and 2016. He received a PhD in robotics of the University of Orleans in 2009. His main researches focus is on navigation for mobiles robots, trajectory generation though direct kinematics model and predictive control approaches. He

participated in the PROTEUS project for the design of a robotic development platform and in medical robotics projects for tele-echography (PROSIT) and needle insertion guidance (DANIEAL).



Cyril Novales received an Engineer diploma in Electronics from ISIM Institute (Montpellier) in 1989, and a PhD degree in 1994 in robotics from the University of Montpellier II (France). From 1995 to 1997, he works on autonomous vehicles in INRIA of Grenoble. In 1997, he joined the University of Orleans (France), where he

is associate professor in the laboratory PRISME (previously LVR). Since this date, he has been involved in the design and the development of mobiles and medical robots. He worked

on tele-operated robotics project (OTELO, TERESA), and works actually on PROSIT and PROTEUS national projects, mainly on autonomous robots and navigation in unknown environment. With Danical 1 and 2 projects, he focusses on robot architecture for autonomous and/or teleoperated robots.



Gérard Poisson Gérard Poisson is Professor in the PRISME Lab, Orleans University, France. His teaching experience includes automatics, mechanical engineering, robotics. His research interests concern the robotics design and control, on the topics of mechatronics, perception of the environment, mechanical design and optimisation.

Specifically involved in robotized tele-echography, he has participated in projects in the fields of industrial, mobile or medical robotics.