

Cooperative SLAM for multiple UGVs navigation using SVSF filter

Fethi Demim, Abdelkrim Nemra, Kahina Louadj, Mustapha Hamerlain & Abdelouahab Bazoula

To cite this article: Fethi Demim, Abdelkrim Nemra, Kahina Louadj, Mustapha Hamerlain & Abdelouahab Bazoula (2017) Cooperative SLAM for multiple UGVs navigation using SVSF filter, *Automatika*, 58:1, 119-129, DOI: [10.1080/00051144.2017.1372123](https://doi.org/10.1080/00051144.2017.1372123)

To link to this article: <https://doi.org/10.1080/00051144.2017.1372123>



© 2017 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group



Published online: 19 Sep 2017.



Submit your article to this journal [↗](#)



Article views: 941



View related articles [↗](#)



View Crossmark data [↗](#)



Cooperative SLAM for multiple UGVs navigation using SVSF filter

Fethi Demim^a, Abdelkrim Nemra^a, Kahina Louadj^b, Mustapha Hamerlain^c and Abdelouahab Bazoula^a

^aLaboratoire Robotique et Productique, Ecole Militaire Polytechnique (EMP), Algiers, Algeria; ^bLaboratoire LIMPAF, Université de Bouira, Algeria; ^cDivision Productique et Robotique, Center for Development of Advanced Technologies (CDTA), Algiers, Algeria

ABSTRACT

The aim of this paper is to present a cooperative simultaneous localization and mapping (CSLAM) solution based on a laser telemeter. The proposed solution gives the opportunity to a group of unmanned ground vehicles (UGVs) to construct a large map and localize themselves without any human intervention. Many solutions proposed to solve this problem, most of them are based on the sequential probabilistic approach, based around Extended Kalman Filter (EKF) or the Rao-Blackwellized particle filter. In our work, we propose a new alternative to avoid these limitations, a novel alternative solution based on the smooth variable structure filter (SVSF) to solve the UGV SLAM problem is proposed. This version of SVSF-SLAM algorithm uses a boundary layer width vector and does not require covariance derivation. The new algorithm has been developed to implement the SVSF filter for CSLAM. Our contribution deals with adapting the SVSF to solve the CSLAM problem for multiple UGVs. The algorithms developed in this work were implemented using a swarm of mobile robots Pioneer 3-AT. Two mapping approaches, point-based and line-based, are implemented and validated experimentally using 2D laser telemeter sensors. Good results are obtained by the Cooperative SVSF-SLAM algorithm compared with the Cooperative EKF-SLAM.

ARTICLE HISTORY

Received 21 September 2016
Accepted 31 July 2017

KEYWORDS

Localization; map building; SVSF filter; sensor fusion; autonomous navigation; cooperative SLAM; mobile robots

1. Introduction

In the last few years, the simultaneous localization and mapping (SLAM) became an important topic of research in the robotics community. Accurate navigation can be achieved by accurate localization within an accurate map. Another technique based on entropy minimization is proposed in [1,2]. The foremost problems of cooperative SLAM (CSLAM) to be solved are in the computational efficiency parts, software/network performance, and robustness to modelling errors. The use of multiple unmanned ground vehicles (UGVs) systems is preferred for their benefits in single-robot systems. The main advantages of the cooperative system are: increasing the efficiency of the overall system, extending the error tolerances, improving reliability and/or performances, rapidity, flexibility and reduction of cost [1]. The CSLAM is crucial in global positioning system. To solve CSLAM problem, we can use inexpensive sensors which is a challenge for researchers. So far, only a little effort has been made in the field of multiple vehicles SLAM. According to Martinelli et al. [3], a number of vehicles can move within their environment and build their map of environment cooperatively using sensor data fusion. Some other solutions for CSLAM, specifically cooperative visual SLAM [4], are based on an Extended Kalman filter. However, the latter is very sensitive to outliers and suffers from linearization. Likewise, the lower bound for the map accuracy is violated

due to errors introduced during EKF linearization process producing inconsistent estimates [5,6]. Solving CSLAM and CVSLAM based on particle filter estimations is presented by Thrun in [7]. Other researches have shown an important drawback related to the computation time which makes it not suitable for hard real-time applications as in airborne navigation [8,9]. Some latest development has been made by proposing filters that try to approximate the nonlinear SLAM problem and recently CSLAM is presented in [10]. The work presented in [11] is a part of research work on autonomous navigation for multiple robots based on local sub-map strategy. In [12], a vision-based position drift minimization method is proposed for cooperative navigation of quad-rotor and ground robot. The recent research seeks for using a multiple small mobile robots which have only limited sensing and computational resources and each robot uses a laser pointer and an event-based vision sensor to compute distances from its surroundings [13]. As an alternative method, another novel algorithm is presented, known as the smooth variable structure filter (SVSF). The SVSF is a relatively new predictor-corrector estimation approach [14]. In 2007, the SVSF was introduced and is based on variable structure theory and sliding-mode concepts [15]. As proved in many researches by Habibi, the SVSF offers a robust and stable estimate to model uncertainties and errors [16]. To solve the SLAM for multiple UGVs, we

can consider two main kinds of solutions: centralized architecture and decentralized architecture. In this paper, multiple UGVs system based on centralized architecture is considered. The work presented in this paper is organized as follows: Section 2 depicts the process model of UGV and the observation model. Section 3 describes the UGV-SLAM algorithm. The CSLAM of multiple UGVs is described in Section 4. Simulation and discussion are presented in Section 5. Experimentation and discussion of results are presented in Section 6. Finally, the conclusion is provided in Section 7.

2. Process model of UGV

The UGV used in our work is the Pioneer 3-AT (Figure 1), the P3-AT is a non-holonomic robot with four wheels. The state vector considered is $[X_r, Y_r, \theta_r]^T$ with (X_r, Y_r) the coordinates of the robot in the global coordinate and θ_r its orientation. The non-holonomic constraint is written as follows [17]:

$$X_r \dot{X}_r \sin(\theta_r) - Y_r \dot{Y}_r \cos(\theta_r) = 0. \quad (1)$$

The state model equations of robot are

$$\begin{cases} \dot{X}_r = v_x \cos(\theta_r) - v_y \sin(\theta_r) \\ \dot{Y}_r = v_y \sin(\theta_r) + v_x \cos(\theta_r) \\ \dot{\theta}_r = w \end{cases} \quad (2)$$

where w is the rotation speed. When $v_y = 0$, and $v = v_x$, the kinematic robot model, by discretizing expression (2), will be

$$\begin{pmatrix} X_{r,k+1} \\ Y_{r,k+1} \\ \theta_{r,k+1} \end{pmatrix} = \begin{pmatrix} X_{r,k} + \Delta T v_k \cos(\theta_{r,k}) \\ Y_{r,k} + \Delta T v_k \sin(\theta_{r,k}) \\ \theta_{r,k} + \Delta T w_k \end{pmatrix} + \begin{pmatrix} \varepsilon_{x_r} \\ \varepsilon_{y_r} \\ \varepsilon_{\theta_r} \end{pmatrix} \quad (3)$$

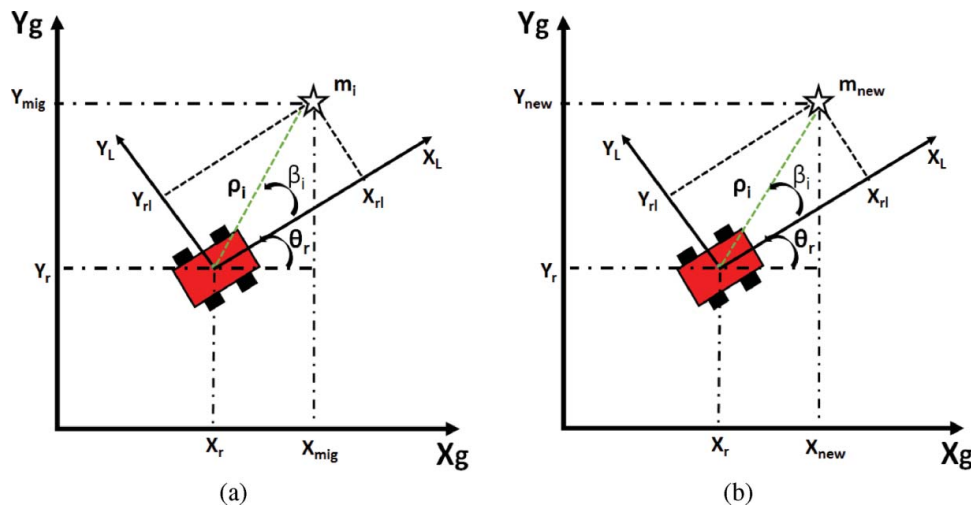


Figure 2. UGV SLAM: (a) direct observation model and (b) inverse observation model.

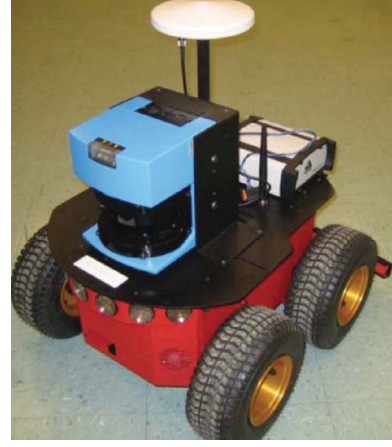


Figure 1. Pioneer 3-AT robot equipped with a 2D SICK laser range finder (LMS-200).

The robot evolution model reflects the relationship between the robot previous states $X_{R,k}$ and its current state $X_{R,k+1}$. ΔT is the time step, $\varepsilon_{x_r, y_r, \theta_r}$ are the noise that arise from the encoder, wheels slipping, etc. The control vector of the robot is the translation and rotation velocity given by $U_k = [v, w]^T$. In SLAM algorithm, the state vector contains two parts, the UGV position given by $X_R = [X_r, Y_r, \theta_r]^T \in \mathbf{R}^3$, and the estimated map given by $m_i = [x_1, y_1, \dots, x_M, y_M]^T \in \mathbf{R}^{2M}$, with M as the total number of landmarks. We can write Equation (3) as follows:

$$X_{R,k+1} = f(X_{R,k}, U_k) + \varepsilon_{x_r, y_r, \theta_r}. \quad (4)$$

2.1. Direct observation point-based model

Perception provides landmark measurements (range and bearing) relative to the UGV frame (Figure 2(a)). We assume that we have M landmarks in the environment. We assume also that the laser range sensor is located

vertically above the centre of the wheel axis. When a landmark m_i is observed, then the UGV will use it as a measurement. The observed quantities are nonlinear functions of the state. $X_{m_{ig}}$ and $Y_{m_{ig}}$ are the coordinates of m_i in the world frame. ρ_i and β_i are the range and bearing of a number of the landmarks in the robot frame. Position and orientation of the UGV in the world frame are given by: $[X_r, Y_r, \theta_r]^T$. Each point of the scan is considered as a landmark and is represented by two parameters in polar coordinates $[\rho_i; \beta_i]$ [16,18].

The representation of point coordinates in the global frame according to its coordinates in the local frame is given by

$$\begin{cases} X_{m_{ig}} = X_{rl}\cos(\theta_r) - Y_{rl}\sin(\theta_r) + X_r \\ Y_{m_{ig}} = X_{rl}\sin(\theta_r) + Y_{rl}\cos(\theta_r) + Y_r \end{cases} \quad (5)$$

The overall transformation to local inverse is given as follows:

$$\begin{cases} X_{rl} = (X_{m_{ig}} - X_r)\cos(\theta_r) + (Y_{m_{ig}} - Y_r)\sin(\theta_r) \\ Y_{rl} = -(X_{m_{ig}} - X_r)\sin(\theta_r) + (Y_{m_{ig}} - Y_r)\cos(\theta_r) \end{cases} \quad (6)$$

where

- $(X_{m_{ig}}; Y_{m_{ig}})$: the coordinates of landmark in the global frame.
- $(X_{rl}; Y_{rl})$: the coordinates of landmark in the local frame.
- (X_r, Y_r) : position of the robot in the global frame.
- θ_r : the orientation of the robot.

The superscript i is the i th sample in landmark sample sets. The observation given by the laser is $Z_k = [\rho_{i,k}, \beta_{i,k}]^T$. The direct observation model $Z_{i,k}$ is expressed as [18]:

$$Z_{i,k} = \begin{bmatrix} \sqrt{(X_{m_{ig}} - X_r)^2 + (Y_{m_{ig}} - Y_r)^2} \\ \tan^{-1}\left(\frac{Y_{m_{ig}} - Y_r}{X_{m_{ig}} - X_r}\right) - \theta_r \end{bmatrix} + \begin{bmatrix} \varepsilon_{\rho_i} \\ \varepsilon_{\beta_i} \end{bmatrix}, \quad (7)$$

where ε_{ρ_i} is the noise of measured range and ε_{β_i} is the noise of bearing in the local frame.

2.2. Inverse observation point-based model

Consider the example shown in Figure 2(b), where the robot is at position $(X_r; Y_r)$ and observing a landmark m_{new} with coordinates $(X_{new}; Y_{new})$ in the world frame using a laser rangefinder. Let $Z_i = [\rho_{il}; \beta_{il}]$ be the observation of landmark m_{new} by the UGV. The landmark mapping model is an inverse observation model $X_{m_{new}}$; using the UGV pose and laser measurement, it can be

written as follows [16,18]:

$$X_{m_{new,k}} = h_i^{-1}(X_k, Z_{i,k}), \quad (8)$$

$$X_{m_{new,k}} = \begin{bmatrix} X_r + \rho_i \cos(\theta_r + \beta_i) \\ Y_r + \rho_i \sin(\theta_r + \beta_i) \end{bmatrix}. \quad (9)$$

2.3. Direct observation line-based model

In the literature, there are several techniques of line fitting. We used Euler representation, where the line is depicted by

$$X_r \cos(\beta_{il}) + Y_r \sin(\beta_{il}) = \rho_{il}. \quad (10)$$

In this case, the direct observation model $[\rho_{il}; \beta_{il}]$ is given by (Figure 3).

$$\begin{bmatrix} \rho_{il} \\ \beta_{il} \end{bmatrix} = \begin{bmatrix} \rho_{ig} - \sqrt{X_r^2 + Y_r^2} \cos\left(\beta_{ig} - \tan^{-1}\left(\frac{Y_r}{X_r}\right)\right) \\ \beta_{ig} - \theta_r \end{bmatrix} + \begin{bmatrix} \varepsilon_{\rho_i} \\ \varepsilon_{\beta_i} \end{bmatrix}. \quad (11)$$

- $[\rho_{il}, \beta_{il}]^T$: polar parameters in the UGV frame of the i th line feature.
- $[\rho_{ig}, \beta_{ig}]^T$: polar parameters in the world frame of the i th line feature.

2.4. Inverse observation line-based model

The mapping model is an inverse observation model $[\rho_{ig}; \beta_{ig}]$ of used features; based on knowledge of the UGV state and the observation, it can be written as follows [18]:

$$\begin{bmatrix} \rho_{ig} \\ \beta_{ig} \end{bmatrix} = \begin{bmatrix} \rho_{il} + \sqrt{X_r^2 + Y_r^2} \cos\left(\beta_{il} - \tan^{-1}\left(\frac{Y_r}{X_r}\right)\right) \\ \beta_{il} + \theta_r \end{bmatrix}. \quad (12)$$

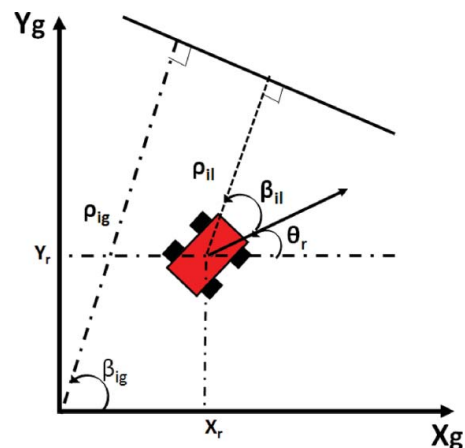


Figure 3. Representation in the local coordinate frame of the i th line feature.

3. UGV-SLAM

On the basis of range and bearing measurements taken with respect to line and pointwise landmarks, the UGV simultaneously builds a map of the environment and localizes itself within it. In this paper, we present two methods to associate landmarks: the first one is point-to-point association. This method is based on minimizing the Euclidean distance. Since the processing time of the “point-to-point” approach is very important and needs a huge memory space, essentially in long-term navigation, we propose another association method which is faster and more robust. This method is based on the association “line-to-line”, which is very efficient for line extraction [19]. The appropriate popular tools to work with a nonlinear dynamic model are EKF-SLAM and the FAST-SLAM. These filters permit to localize and build a map of an unknown environment using only observations relative to extraction features which are detected by its sensors. The extended Kalman filter suffers from the linearization problem, and the FAST-SLAM, which uses Rao-Blackwellised approach for particle filter, is not suitable for real-time implementation. The development of a robust and stable predictor-corrector estimation filter based on sliding-mode theory is proposed to make it suitable for UGV localization and mapping problem. Different approaches are implemented and compared. We start with the popular EKF solution, then we use robust SVSF filter. The latter does not make any assumption on noise characteristics. Furthermore, it is robust against error modelling which is crucial for real-time applications.

3.1. EKF algorithm

System and nonlinear measurement model used for state estimation can be written in discrete form as follows:

$$X_{k+1} = f(X_k, U_k, W_k), \quad (13)$$

$$Z_{k+1} = h(X_k, V_k), \quad (14)$$

where X_{k+1} is the state vector and U_k is the input control vector. The objective of the SLAM problem is to recursively estimate the state X_{k+1} of the landmark according to the measurement Z_{k+1} at a time step. W_k is an uncorrelated zero mean white process which has known covariance Q_k , and V_k is a measured noise with known covariance R_k .

$$\hat{X}_{k+1/k} = F_k \hat{X}_{k/k}, \quad (15)$$

$$P_{k+1/k} = F_k P_{k/k} F_k^T + A_k Q_k A_k^T, \quad (16)$$

$$K_{k+1} = P_{k+1/k} H_k^T [H_{k+1} P_{k+1/k} H_{k+1}^T + R_k]^{-1}, \quad (17)$$

$$\hat{X}_{k+1/k+1} = \hat{X}_{k+1/k} + K_{k+1} [Z_{k+1} - H_{k+1} \hat{X}_{k+1/k}], \quad (18)$$

$$P_{k+1/k+1} = [I - K_{k+1} H_{k+1}] P_{k+1/k}, \quad (19)$$

where $X_{k+1/k}$ represents the predicted state vector and $X_{k+1/k+1}$ denotes the estimated state vector, $(k+1)$ is the Kalman gain matrix, P represents the covariance matrix; A_k and F_k are the Jacobian matrices of the state equation f with respect to the state vector X_{k+1} and the noise variable W_k , respectively; H_{k+1} is the Jacobian matrix of the measurement h with respect to the state vector X_{k+1} and the noise variable V_k , respectively.

3.2. SVSF algorithm

The variable structure filter (VSF) was created in 2003 as a new predictor-corrector estimation approach [15]. The SVSF was introduced in 2007 by Saeid R. Habibi [20]. This filter is based on the sliding-mode control and estimation techniques. It is formulated in a predictor-corrector mode, where gain switching is used to ensure that the estimates converge to true state values [14]. As shown in Figure 4, the SVSF makes use of an existing subspace and of a smoothing boundary layer to keep the estimates bounded within a region of the true state trajectory. The estimation process is summarized in Equations (20)–(26). The SVSF is totally different from EKF which have neither a covariance matrix nor uses it. The SVSF utilizes a switching gain to converge the estimates to within a boundary of the true state values (i.e. existence subspace) [14,15]. The gain is calculated as a function of the error in the predicted output, using the priori and the posteriori measurement errors [21,22]:

$$K_{k+1}^{SVSF} = (\hat{H}_{k+1})^+ \text{diag} \left[\left(|e_{z_{k+1/k}}|_{Abs} + \gamma |e_{z_{k/k}}|_{Abs} \right) \right. \\ \left. \circ \text{Sat} \left(\bar{\varphi}^{-1} e_{z_{k+1/k}} \right) \right] \left[\text{diag} \left(e_{z_{k+1/k}} \right) \right]^{-1}, \quad (20)$$

where

- $e_{z_{k+1/k}}$ is the priori measurement error;
- $e_{z_{k/k}}$ is the posteriori measurement error;
- φ is the smoothing boundary layer widths;
- γ is the convergence rate;

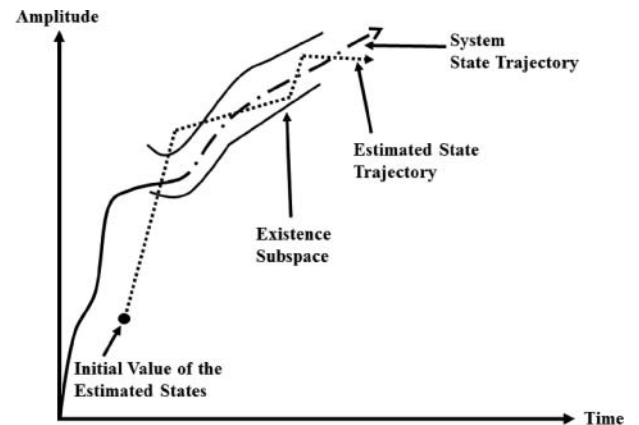


Figure 4. SVSF estimation concept.

- \circ presents “Schur” element-by-element multiplication;
- $^+$ refers to the pseudo-inverse of a matrix;
- $H_{k+1, j} = h_{k+1, j}(F_{X, i})$ is the derivative of h (measurement matrix) with respect to the state vector X_{k+1} . We note that h depends only on the robot pose R_{k+1} and the location of the i th landmark,

where

- i represents the index of its associated landmark in the system state vector X_{k+1} at time k ;
- j defines the index of an individual landmark observation in $Z_{k+1, j}$ at time k .

The matrix $F_{X, i}$ is calculated as follows:

$$F_{X, i} = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & \underbrace{\dots}_{2i-2} & 0 & 0 & 1 & 0 & \underbrace{\dots}_{2M-2i} & 0 \end{bmatrix} \quad (21)$$

We can write $(H_{k+1})^+$ as follows:

$$(H_{k+1})^+ = (F_{X, i})^T (h_{k+1, j})^+. \quad (22)$$

Noting that

$h_{k+1, j} = [\rho_j; \beta_j] = h(X_{k+1, m_i})$ is the observation model of the SVSF-SLAM algorithm.

- $\bar{\varphi}^{-1}$ is established from the smoothing boundary layer vector φ which is formulated as a diagonal matrix

$$\bar{\varphi}^{-1} = [\text{diag}(\varphi)]^{-1} = \begin{pmatrix} \frac{1}{\varphi_1} & 0 & 0 \\ & \ddots & 0 \\ 0 & 0 & \frac{1}{\varphi_{M_i}} \end{pmatrix}, \text{ where } m_M$$

represents the number of measurements.

- $\text{Sat}(\bar{\varphi}^{-1} e_{z_{k+1/k}})$ represents the saturation function

$$\text{Sat}\left(\bar{\varphi}^{-1} e_{z_{k+1/k}}\right) = \begin{cases} +1, & \frac{e_{z_{i,k+1/k}}}{\varphi_i} \geq 1 \\ \frac{e_{z_{i,k+1/k}}}{\varphi_i}, & -1 < \frac{e_{z_{i,k+1/k}}}{\varphi_i} < 1 \\ -1, & \frac{e_{z_{i,k+1/k}}}{\varphi_i} \leq -1 \end{cases} \quad (23)$$

The update of the state estimates can be calculated as follows:

$$\hat{X}_{k+1/k+1} = \hat{X}_{k+1/k} + K_{k+1}^{\text{SVSF}} e_{z_{k+1}}. \quad (24)$$

The priori and posteriori output errors estimates [14] are given by

$$e_{z_{k+1/k}} = Z_{k+1} - \hat{Z}_{k+1/k}, \quad (25)$$

$$e_{z_{k+1/k+1}} = Z_{k+1} - \hat{Z}_{k+1/k+1}. \quad (26)$$

We note that in the SVSF-SLAM algorithm, $Z_{k+1, j}$ estimation process is that, for each observation, only the UGV pose and the observed landmark m_i which is associated, will be updated; in contrary, in the EKF-SLAM algorithm, we update the entire system state vector X_{k+1} . As the SVSF-SLAM, for more details, refer to [16].

4. Cooperative SLAM of multiple UGVs

This work proposes a centralized approach to solve the cooperative multiple UGV-SLAM problem based on laser observation model (Figure 5). A robust nonlinear SVSF filter is implemented to increase the consistency of the CSLAM solution. Each UGV performs a part of computation for such a task, but a central processor is necessary to combine the computations of each UGV to determine the final outcome [6]. The proposed algorithm is more robust, stable and adapted for real-time applications. In multiple vehicles SLAM issue, the estimated state vector becomes the position of the multiple vehicles and the positions of point feature observations in the environment.

This process is known as multiple UGVs SLAM or CSLAM. In the probabilistic approach, the pose of robots UGV_j is $[x_{r_j}, y_{r_j}, \theta_{r_j}]^T$, with $j = 1:N$ and N is the number of used robots. In this case, the state vector X_k contains the poses of the UGVs as well as the M landmarks $m_i = [m_1 \dots m_M]$. The nonlinear discrete-time state transition equation given by Equation (3) can be extended to the multiple UGVs case as follows:

$$X_{k+1} = f(X_k, U_k) + g(X_k) W_k, \quad (27)$$

$$Z_{k+1} = h(X_k, V_k), \quad (28)$$

f is a discrete nonlinear model, X_{k+1} is the state vector at time step $(k+1)$, W_k is some additive process noises, Z_{k+1} is the observation taken at time $k+1$ by all the UGVs, V_k is some additive observation noises. The principle of the filtering method is to estimate X_{k+1} using the observation model Z_{k+1} and h is a

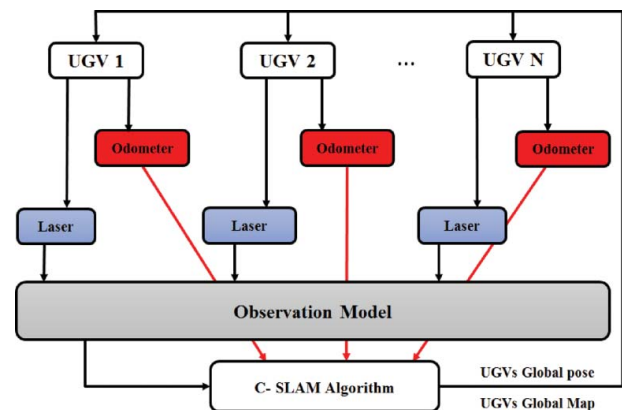


Figure 5. Cooperative SLAM architecture.

continuous observation model. The CSLAM state vector is given by

$$X_{k+1} = [X_{ugv1,k} \dots X_{ugvN,k} \dots m_{1,k} \dots m_{M,k}]^T. \quad (29)$$

The Jacobian matrix of f with respect to X_{k+1} ($F_{sys} = \frac{\partial f}{\partial X_{k+1}}$) is given by

$$F_{sys} = \begin{bmatrix} F_{ugv1,k} & 0_{3 \times 3} & \dots & 0_{3 \times 2M} \\ 0_{3 \times 3} & \ddots & 0_{3 \times 3} & \vdots \\ \vdots & 0_{3 \times 3} & F_{ugvN,k} & 0_{3 \times 2M} \\ 0_{2M \times 3} & \dots & 0_{2M \times 3} & 0_{2M \times 2M} \end{bmatrix}, \quad (30)$$

where F_{ugvj} is the Jacobian matrix of the equation f with respect to the state vector X_{k+1} . The Jacobian is computed at each moment around the estimated current state. This process linearizes the process model around the estimated state. The state vector of each UGV is described as follows:

$$X_{ugvj,k} = [x_{rj,k} \ y_{rj,k} \ \theta_{rj,k}]^T, \quad j = 1, 2, \dots, N. \quad (31)$$

The observation vector by UGVs is given by:

$$X_{m,k} = [x_{mi,k} \ y_{mi,k}]^T, \quad i = 1, 2, \dots, M, \quad (32)$$

$$h = Z_{i,k+1}^j = \begin{bmatrix} \dots & \rho_{i,k}^j & \dots \\ \dots & \beta_{i,k}^j & \dots \end{bmatrix} + \begin{bmatrix} \varepsilon_{\rho_i} \\ \varepsilon_{\beta_i} \end{bmatrix}, \quad (33)$$

$$Z_{i,k+1}^j = \begin{bmatrix} \sqrt{(X_{mi} - X_{rj})^2 + (Y_{mi} - Y_{rj})^2} \\ \tan^{-1} \left(\frac{Y_{mi} - Y_{rj}}{X_{mi} - X_{rj}} \right) - \theta_{rj} \end{bmatrix} + \begin{bmatrix} \varepsilon_{\rho_i} \\ \varepsilon_{\beta_i} \end{bmatrix}. \quad (34)$$

The P_{k+1} is the global covariance matrix estimated at time $(k+1)$ by all the UGVs:

$$P_{k+1} = \begin{bmatrix} P_{ugv1,ugv1} & \dots & P_{ugv1,ugvN} & P_{ugv1,m1} & \dots & P_{ugv1,mM} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ P_{ugvN,ugv1} & \dots & P_{ugvN,ugvN} & P_{ugvN,m1} & \dots & P_{ugvN,mM} \\ P_{ugv1,m1} & \dots & P_{ugvN,mM} & P_{m1,m1} & \dots & P_{m1,mM} \\ \vdots & \ddots & \vdots & \vdots & \dots & \vdots \\ P_{ugv1,mM} & & P_{mM,ugvN} & P_{mM,m1} & & P_{mM,mM} \end{bmatrix}. \quad (35)$$

The Jacobian matrix of f with respect to w_k ($F_{u,sys} = \frac{\partial f}{\partial w_k}$) is given by

$$F_{u,sys} = \begin{bmatrix} F_{u1,ugv} & 0_{3 \times 2} & \dots & 0_{3 \times 2} \\ 0_{3 \times 2} & F_{u2,ugv} & \dots & 0_{3 \times 2} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{3 \times 2} & \dots & \dots & F_{uN,ugv} \\ 0_{2M \times 2} & \dots & \dots & 0_{2M \times 2M} \end{bmatrix}. \quad (36)$$

The Jacobian matrix of h with respect to X_{k+1} for the j th UGV and i th landmark is

$$H_{sys} = \frac{\partial h}{\partial X_{k+1}} = [0_{2 \times 3(j-1)} \ H_{ugv-j} \ 0_{2 \times 3(N-j)} \ 0_{2 \times 2(i-1)} \ H_{(M-i)} \ 0_{2 \times 2(M-i)}]. \quad (37)$$

The Jacobian matrix of f with respect to v_k is

$$V_{sys} = [I_{2 \times 2}]. \quad (38)$$

The off-diagonal elements of the covariance matrix represent the correlation between pairs of state variables. The main drawback of the EKF-SLAM algorithm is the computational complexity due to the dynamical size of the SLAM state vector. This latter increases quadratically with the number of landmarks in the map when new features are observed. In a two-dimensional world, the size of the covariance matrix is $(2M + 3)$ by $(2M + 3)$, where M is the total number of landmarks on the map. As a good alternative, the SVSF represents better performances than the EKF. The SVSF uses two critical variables in its process: the priori and posteriori output error estimates, without using the covariance matrix. A CC-SLAM-based SVSF algorithm is introduced in Section 4.

5. Simulation and discussion

Numerical simulations are reported to illustrate the performance of the proposed approach. In our simulations, we assume that two UGVs are navigating in unknown environment using odometer and laser 2D models. The results of our approach are compared with the cooperative EKF-SLAM algorithm. In this study, the sampling rates used for each sensor and filter are as follows:

$$f_{odom} = f_{laser} = f_{EKF-SLAM} = f_{SVSF-SLAM} = 10 \text{ Hz}.$$

The simulation results provided in Figures 6, 7 and 8 represent the estimated UGVs position obtained using the EKF, and SVSF filters, respectively, with $\sigma_v = 0.1$ m/s, $\sigma_w = 0.25$ rad/s, $\sigma_{\rho_i} = 0.1$ m, $\sigma_{\beta_i} = 0.25$ rad, $\sigma_x = \sigma_y = 10^{-4}$ m, $\sigma_\theta = 10^{-4}$ rad and $\gamma = \text{diag}(0.80, 8)$, $\varphi = \text{diag}(10, 12)$.

Algorithm 1 : Multiple UGV Centralized SVSF-SLAM Algorithm

§At each step time t

◇ **Initialization**

$$U_0 = [U_{ugv_1}(0) \dots U_{ugv_N}(0)]^T$$

$$X_0 = [X_{ugv_1}(0) \dots X_{ugv_N}(0)]^T$$

$$Q_0 = \text{diag}[Q_{ugv_1}(0) \dots Q_{ugv_N}(0)]$$

$$R_t = I_{2 \times 2}, e_{0/0}^Z = [e_0^{Z_1} \dots e_0^{Z_M}]^T$$

◇ **Prediction**

$\hat{X}_{(k+1/k)} = f(\hat{X}_{k/k}, U_k)$ % state prediction using process model.

◇ **Update**

$\hat{Z}_{k+1} = H_{k+1} \hat{X}_{k+1/k}$: % predicted measure.

$$Z_{k+1} = [\dots Z_{k+1}^j \dots] = \begin{bmatrix} \dots & \rho_{i,k}^j & \dots \\ \dots & \beta_{i,k}^j & \dots \end{bmatrix} + \begin{bmatrix} \varepsilon_{\rho_i} \\ \varepsilon_{\beta_i} \end{bmatrix}$$

% coordinates of i^{th} feature observed by j^{th} UGV.

◇ **For $j=1:N$** % numbers of UGVs.

◇ **For $i=1:L$** % number of feature observed by j^{th} UGV $_j$,

-Find the correspond between $Z_{k+1}^{i,j}$ and \hat{Z}_{k+1}

◇ **If** the correspondence is found.

-Compute the Measurement Error

$$e_{Z_{k+1}} = Z_{k+1}^{i,j} - \hat{Z}_{k+1}$$

-Compute the Gain using the priori and the posteriori Measurement Errors.

$$K_{k+1}^{SVSF} = (\hat{H}_{k+1})^+ \text{diag}[(|e_{Z_{k+1}/k}|_{Abs} + \gamma |e_{Z_k/k}|_{Abs})^\circ \text{Sat}(\bar{\varphi}^{-1} e_{Z_{k+1}/k})] [\text{diag}(e_{Z_{k+1}/k})]^{-1}$$

-Update of the state estimates.

$$\hat{X}_{k+1/k+1} = \hat{X}_{k+1/k} + K_{k+1}^{SVSF} e_{Z_{k+1}}$$

-Update the Measurement Errors $e_{Z_{k+1/k+1}}$.

$$\hat{Z}_{k+1} = \hat{H}_{k+1} \hat{X}_{k+1/k+1}$$

$$e_{Z_{k+1/k+1}} = Z_{k+1}^{i,j} - \hat{Z}_{k+1/k+1}$$

◇ **Else** $Z_{k+1}^{i,j}$ is a new feature to be added to the map using the inverse observation model

◇ **End if**

◇ **End for**

◇ **End for**

◇ **Map Management**

◇ **For** all non-associated features

◇- Initial a new feature a_{new} .

$$a_{new} = h^{-1}(\hat{X}_{k+1/k+1}, Z_{k+1/k})$$

◇- Add a_{new} to the state vector $\hat{X}_{k+1/k+1}$.

◇- Use (20) to initialize the posteriori error

$e_{Z_{k+1,new}}$ of the new landmark.

$$e_{Z_{k+1,new}} = Z_{k+1}^{i,j} - h(\hat{X}_{k+1/k+1}, a_{new})$$

◇- Add $e_{Z_{k+1,new}}$ to the posteriori measurement error vector $e_{Z_{k+1}}$.

◇- Increment the state vector $\hat{X}_{Increment}$.

$$\hat{X}_{Increment} = [\hat{X}_{k+1}; a_{new}]$$

$M = M + 1$ % total number of current landmarks.

◇ **End For**

§End

5.1. First experiment: with white-centred Gaussian noise

In the first experiment, we assume a white-centred Gaussian noise, for process and observation model, where

$$Q_k = \begin{bmatrix} (0.1)^2 & 0 \\ 0 & (0.25)^2 \end{bmatrix}, R_k = \begin{bmatrix} (0.1)^2 & 0 \\ 0 & (0.25)^2 \end{bmatrix}.$$

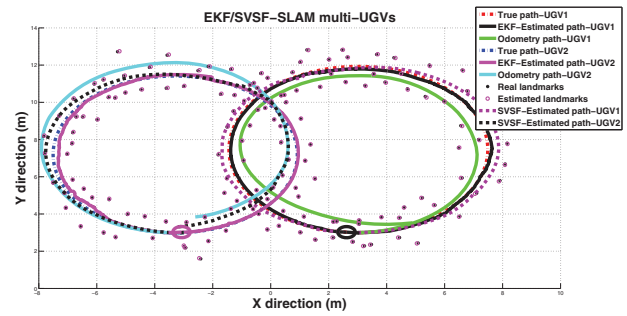


Figure 6. Robots pose using EKF and SVSF-SLAM with white-centred Gaussian noise.

In this case, the scenario represents a simple case that involves two robots moving through a circular trajectory and crossing to measure their relative positions and make the simultaneous localization and mapping with the EKF-SLAM and SVSF-SLAM approaches (Figure 6). Observing a new landmark improves the assessment of the position of the robots. Consequently, they eliminate some parts of the landmark uncertainty previously observed. As can be seen from Figures 6 and 7, when the noise is centred Gaussian, reasonably good EKF-SLAM-based estimation is obtained for x , y and θ , compared to the SVSF-SLAM multi-robots trajectory. Otherwise, we see clearly from Figure 7 that the estimated error increases significantly following x , y and θ . Moreover, from the same figure, we can observe that at the iteration 300, a loop closing is detected. At this moment, we observe a significant improvement of the accuracy of the EKF-SLAM as well the SVSF-SLAM, especially when we use a white Gaussian noise. When the robot closes the loop (revisiting old landmarks or the UGVs observe common feature), UGVs will also correct the positions of landmarks (the map).

The RMSE (root mean square error) for the UGVs position is given in Figure 8. In this figure, the RMSE of SVSF-SLAM and EKF-SLAM is compared in the presence of white Gaussian noise. In this case, the EKF-SLAM algorithm is more accurate than SVSF-SLAM algorithm. This can be interpreted by the optimality of the EKF when the process and observation models are accurately known.

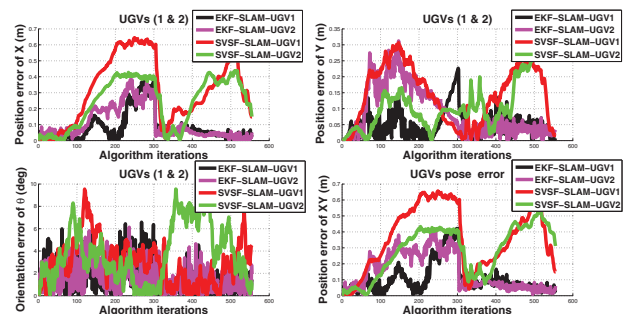


Figure 7. The position of the robot with white-centred Gaussian noise.

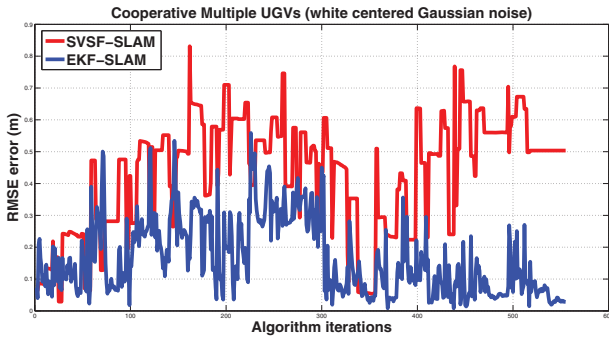


Figure 8. RMSE results for the EKF and SVSF-SLAM algorithm with white-centred Gaussian noise.

5.2. Second experiment: with non-centred Gaussian noise

In this experiment, we assume a white noise with bias. We use the following variance covariance matrix for process and observation models, respectively.

$$Q_k = \begin{bmatrix} (0.1)^2 & 0 \\ 0 & (0.25)^2 \end{bmatrix}, R_k = \begin{bmatrix} (0.1)^2 & 0 \\ 0 & (0.15)^2 \end{bmatrix}$$

When the process and observation noises are non-centred, the performance of the EKF-SLAM estimator decreases significantly as shown in Figure 9. In this case, the corrected values of x , y and θ of the cooperative SVSF-SLAM are more accurate than those estimated by the cooperative EKF-SLAM. This can be explained by the efficiency of the SVSF algorithm compared to the EKF algorithm. SLAM by SVSF shows much better navigation and mapping performances than EKF-SLAM and provides an accurate position of the robots. Comparing the results of this experiment, we can say that the SVSF may perform much better than the EKF (see Figure 10), because this new filter is robust and more suitable for real-time implementation. The RMSE for the UGVs position in this experiment is given in Figure 11. We can see that the cooperative SVSF-SLAM provides the best RMSE in comparison with the cooperative EKF-SLAM.

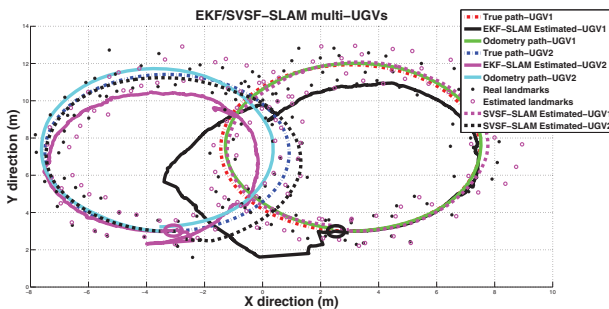


Figure 9. Robots pose using EKF and SVSF-SLAM algorithm with non-centred Gaussian noise.

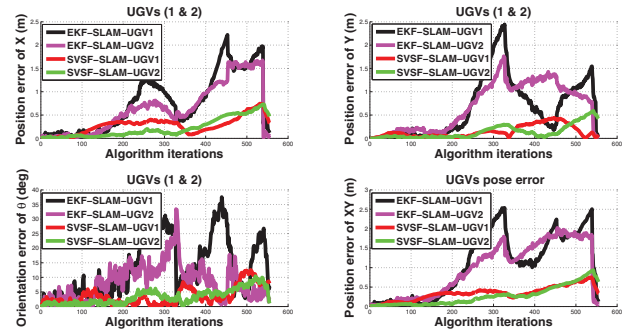


Figure 10. The position of the robot with non-centred Gaussian noise.

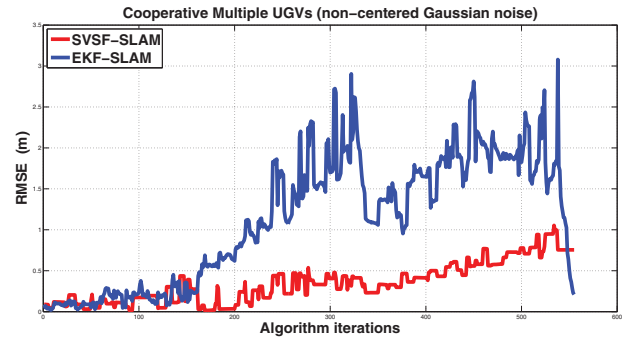


Figure 11. RMSE results for the EKF and SVSF-SLAM algorithm with non-centred Gaussian noise.

5.3. Third experiment: with coloured noise

In this experiment, in order to see the robustness of the SVSF-SLAM algorithm compared to the EKF-SLAM, we assume a coloured noise, for process and observation model, where

$$Q_k = \begin{bmatrix} (0.2)^2 & (0.1)^2 \\ (0.1)^2 & (0.15)^2 \end{bmatrix}, R_k = \begin{bmatrix} (0.02)^2 & (0.03)^2 \\ (0.03)^2 & (0.02)^2 \end{bmatrix}$$

Figure 12 shows the pose of the two robots following x , y and θ axes. As can be seen, from this Figure, x position and θ orientation estimate with significant accuracy when using cooperative SVSF-SLAM. This can be explained by the fact that SVSF filter can

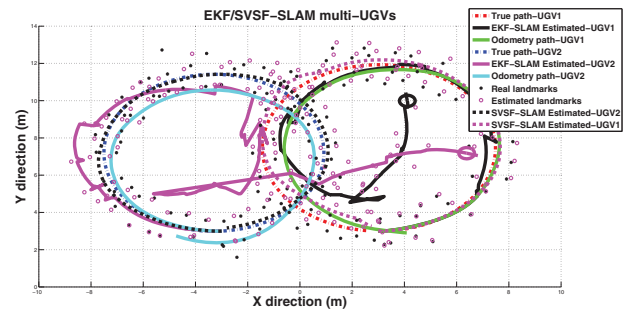


Figure 12. Robots pose using EKF and SVSF-SLAM algorithm with coloured noise.

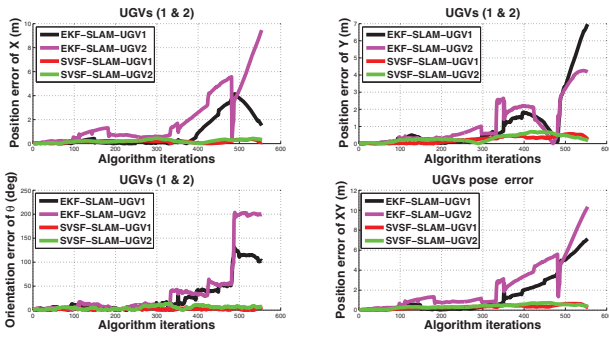


Figure 13. The position of the robot with coloured noise.

provide precision and robustness against uncertain parameters and error modelling. Figure 13 compares the position estimation errors of the EKF and SVSF-SLAM. As can be seen from this figure, good performances are obtained by the new filter for both robots. The SVSF-SLAM provides the best RMSE in comparison with the EKF-SLAM, when we use coloured noise as shown in Figure 14. From the second experiment, we can conclude that the SVSF-SLAM algorithm is more robust and more accurate, even in the presence of coloured noise in the process and observation model. Furthermore, when a loop closure is detected, significant decrease of the RMSE is obtained for SVSF algorithm. The SVSF-SLAM provides the best RMSE in comparison with the EKF-SLAM, when we use non-centred Gaussian noise as shown in Figures 10 and 13. These results clearly validate the advantages of the SVSF-SLAM over the EKF-SLAM, especially when the system or observation models are not accurately known and the process and observation noises are not white or non-centred noise. This result is confirmed by the third experiment, when the process and observation models are coloured noise.

5.4. Computational time evaluation

The structure of cooperative SVSF-SLAM algorithm has been shown to reduce the computational complexity of the algorithm. The EKF-SLAM use of the full size of the state vector to update the pose and the map

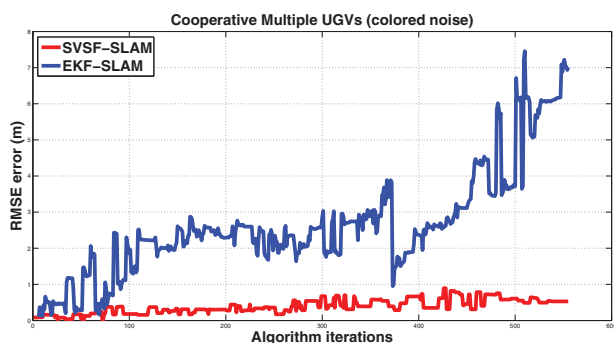


Figure 14. RMSE results for the EKF and SVSF-SLAM algorithm with coloured noise.

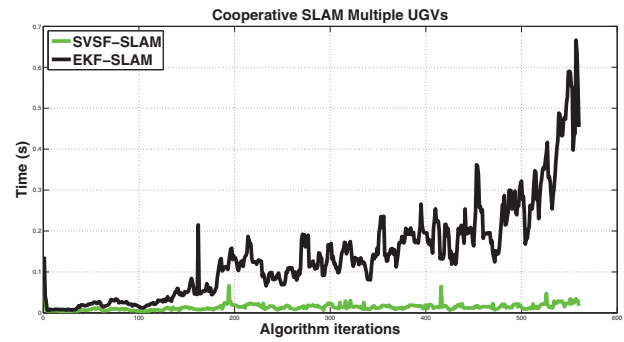


Figure 15. Comparison of computational time per iteration by different algorithms.

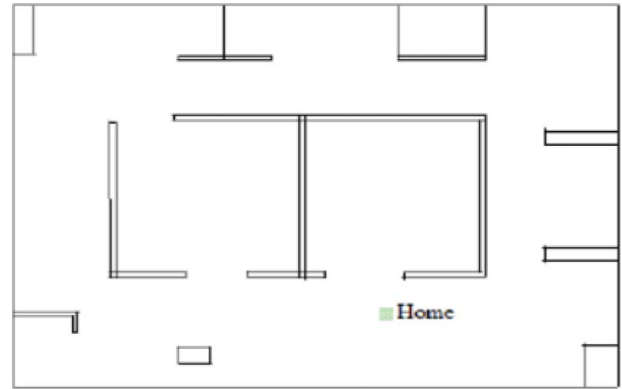


Figure 16. The simulation environment with MobileSim simulator.

estimation is considered as inconvenient, because in this case the computation time will increase with the size of the state vector. On the other hand, the SVSF-SLAM uses only the pose of UGV and the current feature to update the map and the pose estimation which makes the computational time independent of the size of the state vector. The cooperative SVSF-SLAM remains almost constant per iteration depending on the number of visible features (Figure 15). However, the computational time of the cooperative EKF-SLAM grows quadratically over time.

5.5. Experiment MobileSim simulator

In this case, we try to evaluate our algorithms using MobileSim simulator. The scene dimensions are $20\text{ m} \times 16\text{ m}$ (Figure 16). The multi-UGV SVSF/EKF-SLAM with two approaches such as point and line segments features is validated and the environment maps are issued from the simulation results (see Figure 17).

6. Experimentation and discussion of the results

In this experiment, we use two robots Pioneer P3 – AT, which are equipped with two lasers mounted on top of each one (see Figure 19). The P3–AT robots are operated in real time using an ARIA platform and a

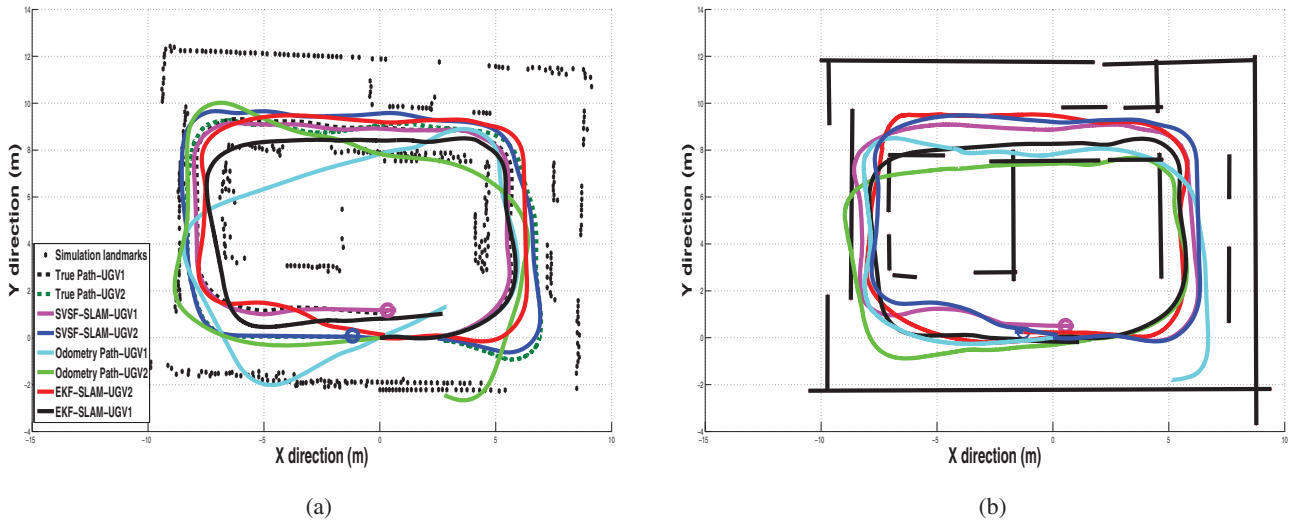


Figure 17. Localization and mapping results with two UGVs: (a) point-based method and (b) line-based method.

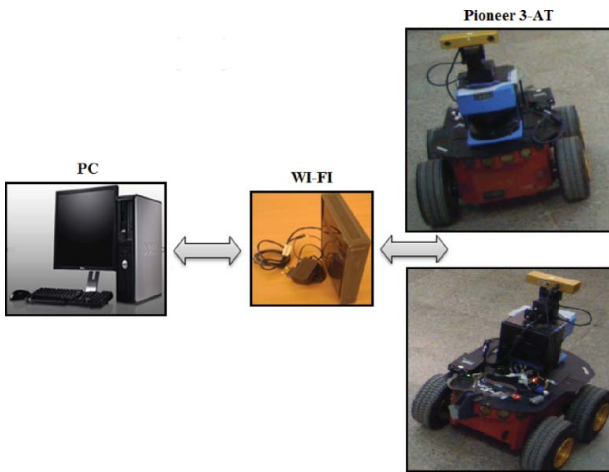


Figure 18. Robots platform test.



Figure 19. Shots of visited places by two robots.

Wi-Fi network socket to keep a data link between a fixed station PC and the mobile robots via vncviewer.exe program (Virtual Network Computing), as can be seen in Figure 18. When the robots move on the map, they use the laser to detect landmarks such as points or lines. In this experiment, we assume that the speed of mobile robot is 0.4 m/s, the sampling time is 0.02 s, and the maximum range of the SICK laser is 15 m. In this section, the experimental data used are collected from experiment locations.

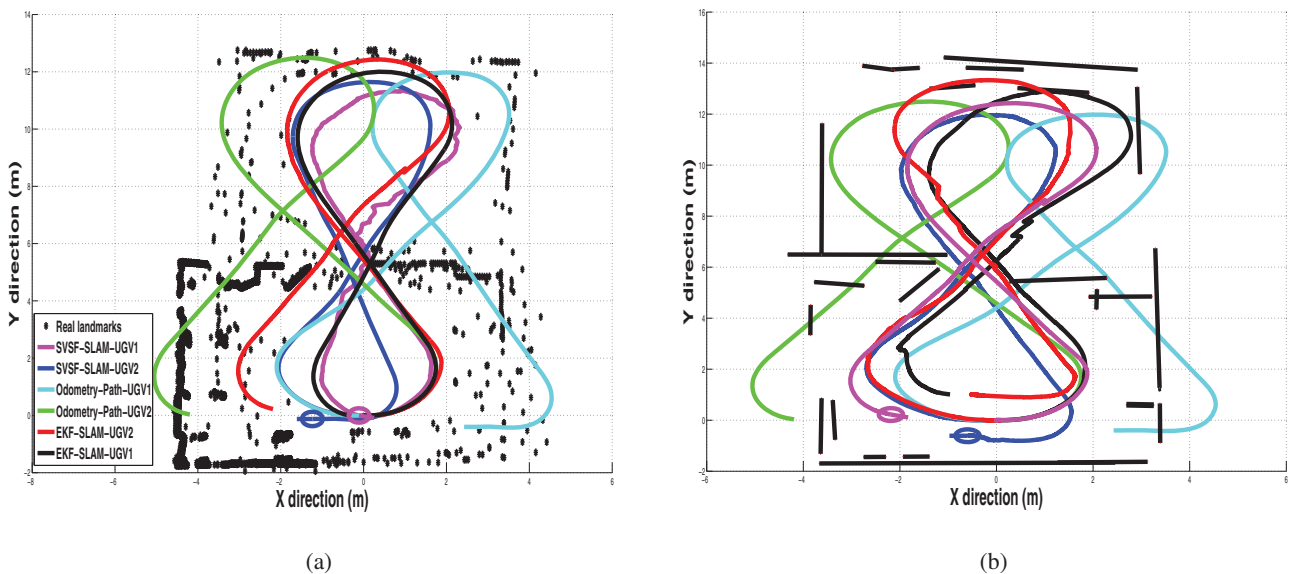


Figure 20. Robots pose estimation using the cooperative SLAM with real data: (a) point-based method and (b) line-based method.

In this experiment, the UGV will navigate in rectangular area with dimension approximately $14\text{ m} \times 8\text{ m}$ (see Figure 19). The maps provided by the cooperative SVSF/EKF-SLAM, shown in Figure 20, is more accurate and descriptive for the navigation environment. Besides, we can see that the borders of the two rooms are conserved and clearly shown. The feature association becomes more complicated by using line approaches in comparison with the points since line segments have to take the best and correct map. For each part, simulation and experimental results have been presented to verify the effectiveness of the cooperative SVSF-SLAM for multiple UGVs algorithms.

7. Conclusion

In this work, we studied a number of technical problems that are necessary to be solved in order to increase UGVs autonomy. First, UGVs localization has been investigated. Then, the UGV map building was presented, followed by an implementation of a SLAM solution using 2D laser telemetry observation. Finally, the proposed solution is validated by the CSLAM case with two types of primitive, point-based approach and line-based approach. Our contribution was the proposition of a new solution for CSLAM for multiple UGVs navigation. The proposed algorithms were validated by simulation and real data on two mobile robots Pioneer P3-AT, and satisfactory results (with a good accuracy and robustness) were obtained with SVSF filter without any assumption on the process or/and observation model accuracy. A real-time experiment illustrates the effectiveness of cooperative SVSF-SLAM algorithm to resolve multiple UGVs localization and mapping problems.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

- [1] Yamashita A, Arai T, Ota J, et al. Motion planning of multiple mobile robots for cooperative manipulation and transportation. *J IEEE Trans Robot Automat.* 2003; 19: 223–237.
- [2] Burgard W, Fox D, Thrun S. Active mobile robot localization by entropy minimisation. In: *IEEE Conference, Second Euro-micro Workshop on Advanced Mobile Robots*; Brescia, Italy; 1997. p. 155–162.
- [3] Martinelli A, Pont F, Siegwart R. Multi-robot localization using relative observations. In: *IEEE Conference on Robotics and Automation*; Barcelona, Spain; 2005. p. 2797–2797.
- [4] Ming Wu, Feifei H, Long W. Cooperative multi-robot monocular-SLAM using salient landmarks. In: *IEEE Conference, International Asia Conference on Informatics in Control*; Bangkok, Thailand; 2009. p. 151–151.
- [5] Dissanayake MWMG, Newman P, Clark S, et al. A solution to the simultaneous localization and map building (SLAM) problem. *J IEEE Trans Robot Automat.* 2001; 17: 229–241.
- [6] Nemra A, Aouf N. Robust airborne 3D visual simultaneous localization and mapping with observability and consistency analysis. *J Int Robot Syst.* 2009; 55: 345–376.
- [7] Thrun S. A probabilistic online mapping algorithm for teams of mobile robots. *J Int Robot Syst.* 2001; 20: 335–363.
- [8] Gil A, Reinoso I, Ballesta M, et al. Multi-robot visual SLAM using a Rao-Blackwellized particle filter. *J Robot Autonom Syst.* 2010; 58: 68–80.
- [9] Ergin Ozkucur N, Levent Akin H. Cooperative multi-robot map merging using Fast-SLAM. In: *Baltes J, Lagoudakis MG, Naruse T, editors. Robo cup*; Berlin Heidelberg: Springer-Verlag; 2009. p. 68–68.
- [10] Andersson LAA, Jonas N. Multi-robot SLAM using square root information smoothing. In: *IEEE Conference, IEEE International Conference on Robotics and Automation*; Pasadena, CA; 2008. p. 2798–2798.
- [11] Xiaolin W, Jing Y, Fengchi S, et al. An approach to multi-robot cooperative SLAM. In: *IEEE Conference, 31st Chinese Control Conference*; Hefei, China; 2012. p. 4904–4909.
- [12] Nasir AK, Hsino A, Roth H, et al. Aerial robot localization using ground robot tracking - towards cooperative SLAM. In: *19th IFAC Symposium on Automatic Control in Aerospace*. Vol. 46; Germany: Elsevier Ltd; 2013. p. 313–318.
- [13] Waniek N, Biedermann J, Conradt J. Cooperative SLAM on small mobile robots. In: *IEEE Conference on Robotics and Biomimetics*. 2015. p. 6–6.
- [14] Gadsden SA, Habibi SR, Kirubarajan T. Kalman and smooth variable structure filters for robust estimation. *IEEE J Trans Aerospace Electron Syst.* 2014; 50: 1038–1050.
- [15] Habibi SR, Burton R. Variable structure filter. *ASME J Dyn Syst Meas Control.* 2003; 125: 287–293.
- [16] Demim F, Nemra A, Louadj K, et al. Simultaneous localization and mapping algorithm for unmanned ground vehicle with SVSF filter. In: *IEEE Conference, 8th International Conference on Modelling, Identification and Control*; Algiers, Algeria; 2016. p. 155–162.
- [17] Sfeir J. Navigation d'un robot mobile en environnement inconnu utilisant les champs de potentiels artificiels, maîtrise en génie de la production automatisée. Montréal: École de Technologie Supérieure; 2009.
- [18] Demim F, Nemra A, Louadj K. Robust SVSF-SLAM for unmanned vehicle in unknown environment. *J IFAC-Papers OnLine Sci Direct.* 2016; 49: 386–394.
- [19] Arras KO, Siegwart R. Feature extraction and scene interpretation for map-based navigation and map building. *ASME J Dyn Syst Meas Control.* 1997; 125: 42–53.
- [20] Habibi SR. The smooth variable structure filter. In: *IEEE Conference*. Vol. 95. 2007. p. 1026–1026.
- [21] Al-Shabi M. The general Toeplitz/observability SVSF [master's thesis]. Vol. 125. Department of Mechanical Engineering, Jordan University of Science & Technology; 2011. p. 42–53.
- [22] Demim F, Boucheloukh A, Nemra A, et al. A new adaptive smooth variable structure filter SLAM algorithm for unmanned vehicle. In: *IEEE Conference, 6th International Conference on Systems and Control*; Batna, Algeria; 2017. p. 6–13.