# Simultaneous Localization and Mapping Algorithm for Unmanned Ground Vehicle with SVSF Filter

Fethi DEMIM
dept. UER Automatique
Laboratoire Robotique et Productique, EMP
Bordj EI Bahri, Algiers, Algeria
demifethi@gmail.com

Abdelkrim NEMRA
dept. UER Automatique
Laboratoire Robotique et Productique, EMP
Bordj EI Bahri, Algiers, Algeria
karim nemra@yahoo.fr

Kahina LOUADJ
Laboratoire d'Informatique de Mathématiques
et de Physique pour l'Agriculture et les Forets
LIMPAF, Université de Bouira, Algérie
louadj kahina@yahoo.fr

Zakaria MEHAL
dept. UER Automatique
Laboratoire Robotique et Productique, EMP
Bordj EI Bahri, Algiers, Algeria
mhzaki@outlook.fr

Mustapha HAMERLAIN
Center for Development of Advanced
Technologies, CDTA
Baba Hassen, Algiers, Algeria
mhamerlain@cdta.dz

Abdelouahab BAZOULA
dept. UER Automatique
Laboratoire Robotique et Productique, EMP
Bordj EI Bahri, Algiers, Algeria
abdelouahab.bazoula@gmail.com

***Abstract*— Filtering strategies play an important role in estimation theory, and are used to extract knowledge of the true states typically from noisy measurements or observations made of the system. This paper describes a novel approach that combines the information given by an odometer and a laser range finder sensors to efficiently solve the Simultaneous Localization and Mapping (SLAM) problem of the Unmanned Ground Vehicle (UGV) and reconstruct a $2D$ representation of the environment. In recent years, to solve the SLAM problem, many solutions have been proposed. To resolve this problem, the most commonly used approaches are the EKF-SLAM and the FAST-SLAM. An accurate process and a model of observation are needed for the first approach, which is suffering the linearization problem. While the second one is not convenient and is not suitable for real time implementation. Therefore, a new state and parameter estimation method is introduced based on the smooth variable structure filter (SVSF) is proposed in this paper to solve the UGV SLAM problem. The SVSF is a relatively new estimation strategy based on sliding mode concepts, formulated in a predictor-corrector format. It has been shown to be very robust to modeling errors and uncertainties. In this work the SVSF-SLAM algorithm is implemented to construct a map of the environment and localize the UGV within this map. The proposed algorithm is validated and compared to the EKF-SLAM algorithm. Good results are obtained.**

***Keywords— localization; map building; autonomous navigation; sensor fusion; unmanned ground vehicle***

## I. INTRODUCTION

Simultaneous localization and mapping (SLAM) is the process that enables the Unmanned Ground Vehicle to localize and build a map of an unknown environment using only observations relative to the most relevant features detected by its sensors. SLAM is an essential capability for our UGV traveling in unknown environments where globally accurate position data (e.g. GPS) is not available. In particular, mobile robots have shown significant promise for remote exploration, going places that are too distant [1], too dangerous [2], or simply too costly to allow humanitarian access. If robots are to operate autonomously in the external environment under the sea, underground, and on the surfaces of other planets, they must be capable of building maps and navigating reliably according to these maps. The solution to the SLAM problem is considered by many as a key prerequisite for making a robot fully autonomous [3] [4]. The problem of SLAM has attracted immense attention in the robotics literature. SLAM addresses the problem of the Unmanned Ground Vehicle moving through an environment of which no map is available a priori.

The dominant approach to the SLAM problem was introduced in a seminal paper by Smith and Cheeseman in 1986, and first developed into an implemented system by Moutarlier and Chatila [5]. This approach uses the Extended Kalman Filter (EKF) to estimate the posterior over robot poses and maps. There have been several implementations of the EKF-SLAM in different environments, such as indoors [6], underwater [7] and outdoors [8]. The EKF approximates the SLAM posterior as a high-dimensional Gaussian over all features in the map and the robot pose [9]. There is another type of filter started to rise and take place in estimation utilizing the principles of the Unscented Kalman Filter (UKF) uses a unique representation of a Gaussian random variable in N dimensions using 2N+1 samples, called sigma points [10]. The UKF suffers less from linearization, though it is not

978-0-9567157-6-0 

exempt. Finally, the UKF does not fully recover from poor landmarks, just as with the EKF.

There is another algorithm which uses the multi hypothesis data association and logarithmic complexity instead of quadratic. This approach, known as Fast-SLAM utilizes Rao-Blackwellised particle filter to solve the SLAM problem efficiently. Using Fast-SLAM algorithm, the posterior estimation will be over the robot's pose and landmark locations. The Fast-SLAM algorithm has been implemented successfully over thousands of landmarks and compare to EKF-SLAM that can only handle a few hundreds of landmarks, it has appeared with considerable advantages [11][12][13].

As an alternative approach, there is another novel algorithm, known as the smooth variable structure filter (SVSF). The SVSF is a relatively new predictor-correct estimation method based on sliding mode theory [15]. In 2007, the smooth variable structure filter (SVSF) was introduced and is based on variable structure theory and sliding mode concepts [15].

The development of a new predictor-corrector filter based on sliding mode theory is proposed for state and parameter estimation known as the Smooth Variable Structure Filter (SVSF) which is robust and stable to modeling uncertainties making it suitable for Unmanned Ground Vehicle (UGV) localization and mapping problem. Moreover, the SVSF is a robust recursive predictor-corrector estimation method that can effectively deal with uncertainties associated with initial conditions and modeling errors of Odometer/Laser system. In parameter estimation, the EKF-SLAM strategy is very sensitive to modeling uncertainties and is susceptible to instability. The concept of SVSF-SLAM is reviewed, this new strategy retains the near optimal performance of the SVSF when applied to an uncertain system, it has the added benefit of presenting a considerable improvement in the robustness of the estimation process.

The paper is organized as follows. Section 2 depicts the process model of UGV and the observation model. Section 3 describes the EKF filter. The Section 4 describes also the SVSF filter in detail. Section 5 proposes our version of the implementation of the SVSF-SLAM algorithm. Finally, results, discussion and conclusion are provided in Section 6.

## II. PROCESS MODEL OF UGV

The Pioneer 3-AT robot (Fig. 1) used in this work is a non-holonomic robot with four wheels. Let the vector $[X_r; Y_r; \theta_r]$ with $(X_r; Y_r)$ be the coordinates of the robot in the global coordinate and $\theta_r$ it's an orientation. The non-holonomic constraint is written as follows [18]

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

From the [18] and as can be seen in Fig. 2, the state transition equation of the robot is

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

Mobile robot's motion model and measurement output model are used to describe the motion and the status of the robot and landmarks at the next time step, with: $\omega$ present the rotation speed. When $v_x = 0, v = v_x$, and by discretizing expression "(2)," the model will be written as follows

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

Fig. 1. Representation the UGV (Pioneer 3-AT) equipped with a SICK 2D laser range finder (LMS-200) for 2D mapping

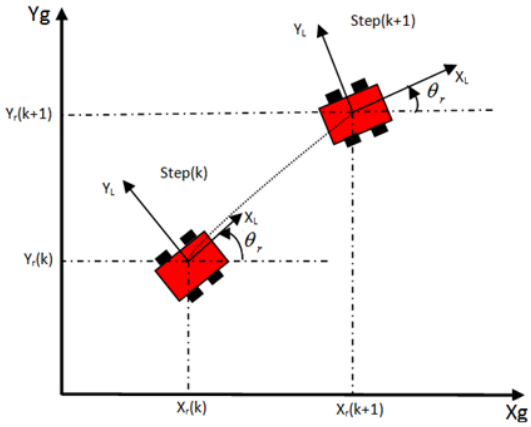


Fig. 2. UGV modeling and representation in the global coordinate system

The robot evolution model reflects the relationship between the robot previous states $X_{R,k}$ and its current state $X_{R,k+1}$. Let control vector be $U_k = [v, \omega]^T$, when $U_k$ is put on the robot.
The robot moves on a plane and receives direction and point landmarks range on the same plane. In SLAM, the system state vector has a position of the UGV $(X_R)$. It is represented by $X_R = [X_r; Y_r; \theta_r]^T \in R^3$, and the map is presented by $m_i = [x_1, y_1, \ldots, x_M, y_M]^T \in R^{2M}$, with $M$ is the total number of landmarks, $\Delta T$ is the time step, $\varepsilon_{x_r, y_r \theta_r}$ are the noise that arises from the encoder and wheels slipping, etc.
We can write "(1)," as follows

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

### A. *Direct Observation Point based Model*

Perceptions provides measurements of range and angle of direction of a landmark known position, measured relative to the robot [14]. We assume that we have $m_i$ numbers of

landmark in the environment located at known points. We also assume that the laser range sensor is located vertically above the center of the axle wheels, because this assumption simplifies the equations and study the algorithm. If $m_i$ is known, then the UGV will be used as the location for observation. The observed quantities are nonlinear functions of the state.

When at the moment $k$ landmark $m_i$ coordinated $X_{m_{ig}}$ and $Y_{m_{ig}}$ is detected, an angle $\beta_i$ is measured from the $X-$axis is the line connecting the center with the landmark benchmark UGV and a range $\rho_i$ between landmark and the origin of the coordinate of the UGV connected to the state $[X_r; Y_r; \theta_r]^T$ [14].

The UGV used in our experiment is equipped with a laser range finder (see Fig. 1), returning 361 range measurements in a single sweep, with a range resolution of $0.05\,meters$, an angular resolution of $0.5^{\circ}$, a maximum range of $15\,meters$ and a scanning angle of $180^{\circ}$. In the case of the representation of landmark by point, each point of the scan is considered a landmark and is represented by two parameters $[\rho_i; \beta_i]$. The representation of point coordinates in the global frame according to its coordinates in the local frame is given by [14].

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

The overall transformation to local inverse is given as follows

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

where

- $\left(X_{m_{ig}}; Y_{m_{ig}}\right)$: The coordinates of landmark in the global frame.
- $\left(X_{rl}; Y_{rl}\right)$: The coordinates of landmark in the local frame.
- $\left(X_r; Y_r\right)$: Position of the robot in the global frame.
- $\theta_r$: The orientation of the robot.

The superscript $i$ is the $i^{th}$ sample in landmark sample sets. The observation is given by $Z_k = \left[\rho_{i,k}, \beta_{i,k}\right]^T$ are the range and bearing of a sensor relative to a landmark. The direct observation models are expressed as

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

Where $\varepsilon_{\rho_i}$ is the noise of measured range and $\varepsilon_{\beta_i} is\ the$ bearing in the local frame.

### *B. Inverse Observation Point based Model*

We consider the example shown in [14]. Where the robot state $\left(X_r; Y_r\right)$ and watching a landmark $m_{new}$ with coordinates $\left(X_{new}; Y_{new}\right)$ using a laser range finder; let $Z_i = [\rho_{il}; \beta_{il}]$ is the observation of landmark $m_{new}$ by the robot. The landmark mapping model is an inverse observation model, knowing the state of the robot and observation, it can be written as follows

$$X_{m_i,k} = h_i^{-1}\left(X_k, Z_{i,k}\right) \quad (8)$$

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

## III. EKF FILTER

The Kalman Filter (KF), provides an elegant and statistically optimal solution for linear dynamic systems in the presence of Gaussian white noise. It is a method that utilizes measurements linearly related to the states, and error covariance matrices, to generate a gain referred to as the Kalman gain. This gain is applied to the a priori state estimate, thus creating an a posteriori estimate. The following two equations describe the system dynamic model and the measurement model used in general for linear state estimation [12].

$$X_{k+1} = f(X_k, U_k, W_k) \quad (10)$$

$$Z_{k+1} = h(X_k, V_k) \quad (11)$$

The next five equations from the KF algorithm, and are used in an iterative fashion, in conjunction with "(10)," and "(11)," . The "(12)," extrapolates the a priori state estimate, and "(13)," is the corresponding state error covariance. The Kalman gain may be calculated by "(14)," and is used to update the state estimate and error covariance, described by "(15)," and "(16)," respectively.

$$\hat{X}_{k+1/k} = F_k \hat{X}_{k/k} \quad (12)$$

$$P_{k+1/k} = F_k P_{k/k} F_k^T + A_k Q_k A_k^T \quad (13)$$

$$K_{k+1} = P_{k+1/k} H_k^T \left[H_{k+1} P_{k+1/k} H_{k+1}^T + R_k\right]^{-1} \quad (14)$$

$$\hat{X}_{k+1/k+1} = \hat{X}_{k+1/k} + K_{k+1}\left[Z_{k+1} - H_{k+1}\hat{X}_{k+1/k}\right] \quad (15)$$

$$P_{k+1/k+1} = [I - K_{k+1}H_{k+1}]P_{k+1/k} \quad (16)$$

The EKF may be used for nonlinear systems. It is conceptually similar to the iterative KF process. The nonlinear system $F_k$ and measurement $H_{k+1}$ matrices are linearized according to its corresponding Jacobian, which is a first-order partial derive. This linearization can sometimes cause instabilities when implementing the EKF [12]. The main steps of the EKF algorithm can be defined as follows (see Fig. 3):

978-0-9567157-6-0 © IEEE 2016

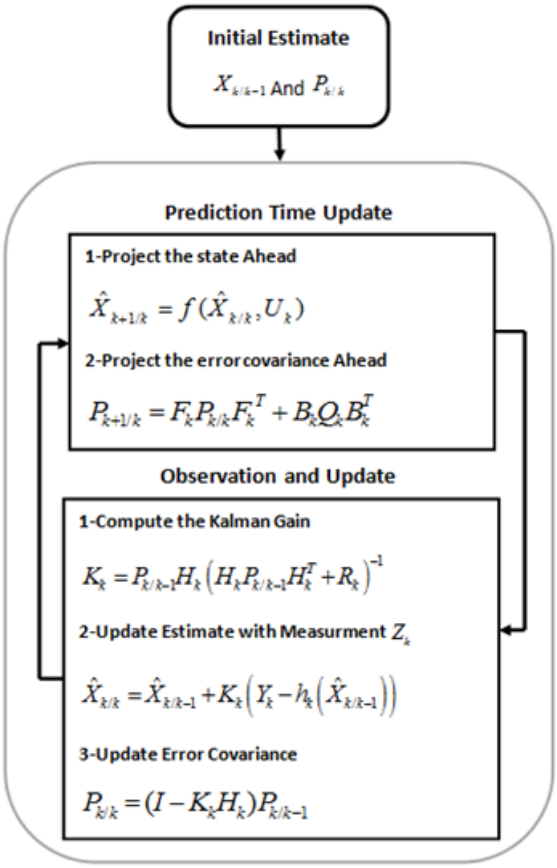


Fig. 3. Algorithm of the EKF filter

## IV. SVSF FILTER

In 2007, the smooth variable structure filter (SVSF) was introduced. This filter is based on the sliding mode control and estimation techniques, and is formulated in a predictor-corrector fashion. As shown in the following figure 4, the SVSF utilizes a switching gain to converge the estimates to within a boundary of the true state values (i.e., existence subspace), this creates a robust and stable estimation strategy [15][17]. The SVSF has been shown to be stable and robust to modeling uncertainties and noise, when given an upper bound on the level of unmodeled dynamics and noise [16].

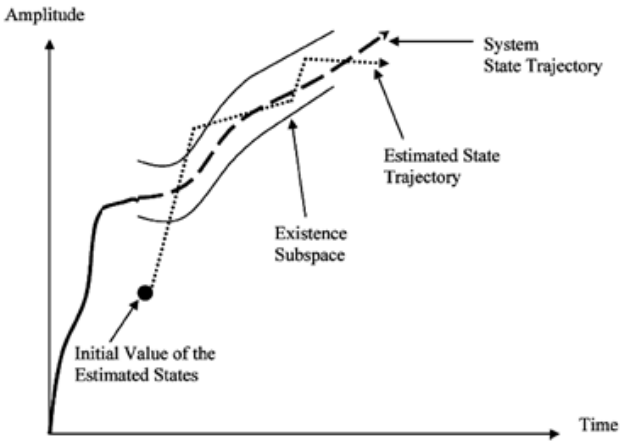


Fig. 4. SVSF Estimation Concept

The SVSF method is model based and applies to smooth nonlinear dynamic equations. The estimation process may be summarized by "(17)," to "(23)," and is repeated iteratively. An a priori state estimate is calculated using an estimated model of the system [15]. This a priori value is then used to calculate an a priori estimate of the measurement, defined by "(18)". A corrective term, referred to as the SVSF gain, is calculated as a function of the error in the predicted output, as well as a gain matrix and the smoothing boundary layer width. The correct term calculated in "(19)," is then used in "(23)," to find the a posteriori state estimate. Two critical variables in this process are the a priori and a posteriori output error estimate, defined by "(21)," and "(22)," respectively [15]. Note that "(21)," is the output error estimate from the previous time step, and is used only in the gain calculation.

$$\hat{X}_{k+1} = f(\hat{X}_k, U_k) \tag{17}$$

$$\hat{Z}_{k+1} = h(\hat{X}_k) \tag{18}$$

$U_k$ is the input control vector. The SVSF guarantees bounded-input bounded-output stability and the convergence of the estimation process by using the Lyapunov stability condition [15]. The derivation of the SVSF gain and its stability conditions can be found in [17]. The gain is computed using the a priori, the a posteriori measurement errors, the smoothing boundary layer widths $\varphi$, convergence rate $\gamma$ and measurement matrix $H_{k+1}$ as follows

$$K_{k+1}^{SVSF} = \left(\hat{H}_{k+1}\right)^{+} diag\left[\left(\left|e_{Z_{k+1/k}}\right|_{Abs} + \gamma\left|e_{Z_{k/k}}\right|_{Abs}\right)^{o} Sat\left(\bar{\varphi}^{-1}\, e_{Z_{k+1/k}}\right)\right]\left[diag\left(e_{Z_{k+1/k}}\right)\right]^{-1} \tag{19}$$

where

- $^{o}$ presents "Schur" multiplication element-by-element;
- $^{+}$ refers to the pseudo inverse of a matrix;
- $\bar{\varphi}^{-1}$ is a diagonal matrix constructed from the smoothing boundary layer vector $\varphi$, such that

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

with $M_i$ presents the number of measurements.

- $Sat\left(\bar{\varphi}^{-1}\, e_{Z_{k+1/k}}\right)$ presents the saturation function

$$Sat\left(\bar{\varphi}^{-1}\, e_{Z_{k+1/k}}\right) = \begin{cases} +1, & \frac{e_{Z_{i,k+1/k}}}{\varphi_i} \geq 1 \\ \frac{e_{Z_{i,k+1/k}}}{\varphi_i}, & -1 < \frac{e_{Z_{i,k+1/k}}}{\varphi_i} < 1 \\ -1, & \frac{e_{Z_{i,k+\frac{1}{k}}}}{\varphi_i} \leq -1 \end{cases} \tag{20}$$

$$e_{Z_{k+1/k}} = Z_{k+1} - \hat{Z}_{k+1/k} \tag{21}$$

$$e_{Z_{k+1/k+1}} = Z_{k+1} - \hat{Z}_{k+1/k+1} \tag{22}$$

The update of the state estimates can be calculated as follows: $\hat{X}_{k+1/k+1} = \hat{X}_{k+1/k} + K_{k+1}^{SVSF} e_{Z_{k+1/k}}$ (23)

## V. SVSF-SLAM ALGORITHM

Simultaneous Localization and Mapping (SLAM) is the problem of constructing a model of the environment being traversed with on board sensors, while at the same time maintaining an estimate of the vehicle location within the model. As an alternative approach, there is another novel filter, known as the smooth variable structure filter (SVSF). This filter provides a robust and stable estimate to modeling uncertainties and errors [17].

In this section we investigate the SVSF-SLAM algorithm proposed as an alternative to solve the Unmanned Ground Vehicle SLAM problem. The initial conditions used by the SVSF-SLAM was the same as those used by the EKF-SLAM. There are two main SVSF design parameters. The first parameter $\gamma$ which control the speed of convergence, where the second $\varphi$ refers to the boundary layer width which is used to smooth out the switching action. These parameters should be chosen carefully. $\gamma$ is a diagonal matrix with elements were set to the following

$$0 < \gamma_i \leq 1 \quad (24)$$

According to [15], the estimation process is stable and converges to the existence subspace if the following condition is satisfied

$$\left|e_{k/k}\right|_{Abs} > \left|e_{k+1/k+1}\right|_{Abs} \quad (25)$$

In this section, a framework for feature map SLAM based on the SVSF are presented. Like EKF-SLAM, SVSF-SLAM is a kind of stochastic SLAM algorithm, which is performed by storing the vehicle pose and map features in a single state vector. It consists of following stages: initialization, prediction, data association and update, finally the management of the map [14].

**SVSF-SLAM Algorithm**

***At each step time t***

- **Initialization**

The process estimation needs the initialization:

$$\hat{X}_{0/0} = \left[\hat{X}_0, \hat{m}_1, \dots, \hat{m}_M\right]^T;\ e_{0/0}^{Z} = \left[e_0^{Z_1}, \dots, e_0^{Z_M}\right]^T$$

- **Prediction Time Update**

The prediction stage is a process, which deals with vehicle motion based on incremental dead reckoning estimates and increases the uncertainty of the vehicle pose estimate. First, the state vector is augmented with a control input $U_k$. Consider the following process for the SVSF estimation strategy, as applied to a nonlinear system with a linear measurement equation. The predicted state estimates $\hat{X}_{k+1/k}$ are first calculated as follows

$1-\ \hat{X}_{k+1/k} = f\left(\hat{X}_{k/k}, U_k\right)$

This motion model is generally nonlinear in its arguments.

- **Data association and update**

$2-$ ***For*** all features observations

$3-\ \hat{Z}_{k+\frac{1}{k}} = h\left(\hat{X}_{k+1}\right) = h\left(\hat{R}_{k+1}, m_i\right)$

$4-$ ***if*** the correspondence is founded.

The a posteriori measurements error vector $\hat{e}_{z_{k+1/k+1}} \in R^{2*1}$ may be calculated:

$5-\ \hat{e}_{z_{k+1/k+1}} = Z_{k+1} - \hat{Z}_{k+1/k}$

$6-$ Use "(19)," to calculate SVSF gain $K_{k+1}^{SVSF}$.

$7-$ Use "(23)," to calculate the state vector $\hat{X}_{k+1/k+1}$.

The a priori measurements error vector $e_{Z_{k+1/k}} \in R^{2*1}$ may be calculated:

$8-\ e_{z_{k+1/k}} = z_{k+1} - h(\hat{R}_{k+1}, m_i)$

$9-$ ***End if***

$10-$ ***End For***

$11-\ \hat{X}_{k+1/k+1} = \hat{X}_{k+1/k}$

Note that the matrix $H_{k+1}$ is a Jacobian matrix with partial derivatives of the system $h(.)$.

$H_{k+1} = \left[H_{R,k+1}\ 0 \dots\ H_{m_i,k+1} \dots 0\right]$ Such that:

$H_{R,k+1} = \frac{\partial h(X_{k+1/k})}{\partial X_{R,k+1}}$ and $H_{m_i,k+1} = \frac{\partial h(X_{k+1/k})}{\partial X_{m_i}}$

The update measurement estimate $Z_{k+1/k+1}$ and measurement errors $e_{z_{k+1/k+1}}$ are used to calculate for the next iterations. The $K_{k+1}^{SVSF}$ is calculated by the a priori and posteriori state estimate and the smoothing boundary layer widths $\varphi$.

The estimation equation of SVSF-SLAM is computationally simpler than EKF-SLAM and the updating form of SVSF is more suitable than EKF. The SVSF-SLAM performed significantly better than the EKF-SLAM, in terms of estimation error. In order to be processed by SVSF algorithm as shows in Algorithm 1, each measurement extracted from the sensor data must be associated with the corresponding either line or point present in the map. Only the matching features can be used to update the robot pose and the map.

- **Management of the map**

As the environment is explored, new features are observed and should be added to the stored map. In this case, the state vector and the output error estimate matrix are calculated of the new observation.

$12-$ **For** all non-associated features.

Initial a new feature $a_{new}$:

$13-\ a_{new} = h^{-1}\left(\hat{X}_{k+1/k+1}, Z_{k+1/k}\right)$

$14-$ Added $a_{new}$ to the state vector $\hat{X}_{k+1/k+1}$.

Use "(22)," to initialize a posteriori measurement error $e_{Z,new}$ of the new landmark $a_{new}$.

$15-\ e_{Z,new} = z_{k+1} - h(\hat{X}_{k+1/k+1}, a_{new})$

Incremented of the state vector $\hat{X}_{Incremt}$

$16-\ \hat{X}_{Incremt} = [\hat{X}_{k+1}; a_{new}]$

$17-\ M = M + 1$ % total number of landmarks

18 – ***End For***
***End***

**Algorithm 1:** SVSF-SLAM Algorithm

§ **At each step time t**
♦ **Initial Estimate**
$\hat{X}_0$ and $e_{e0}$
♦ **Prediction Time Update**
◊– Project state Ahead
$\hat{X}_{k+1/k} = f(\hat{X}_{k/k}, U_k)$ % state prediction.
♦ **Observation and Update**
◊ **For** all features observations.
◊ **If** the correspondence is founded.
◊– Calculate the a posteriori measurement error
$\hat{e}_{z_{k+1/k+1}} = Z_{k+1} - \hat{Z}_{k+1/k}$
◊– Compute the SVSF Gain $K_{k+1}^{SVSF}$
$K_{k+1}^{SVSF} = (\hat{H}_{k+1})^{+} diag\left[\left(\left|e_{Z_{k+1/k}}\right|_{Abs} + \gamma\left|e_{Z_{k/k}}\right|_{Abs}\right)^{o} Sat\left(\bar{\varphi}^{-1}\, e_{Z_{k+1/k}}\right)\right]\left[diag\left(e_{Z_{k+1/k}}\right)\right]^{-1}$
◊– Update on the State estimates vector $\hat{X}_{k+1/k+1}$
$\hat{X}_{k+1/k+1} = \hat{X}_{k+1/k} + K_{k+1}^{SVSF}\hat{e}_{z_{k+1/k}}$
◊– Calculate the a priori measurement error vector
$e_{z_{k+1/k}} = z_{k+1} - h(\hat{R}_{k+1}, m_i)$
◊**End if**
◊**End For**
♦ **Map Management**
◊ **For** all non-associated features.
◊– Initial a new feature $a_{new}$.
$a_{new} = h^{-1}(\hat{X}_{k+1/k+1}, Z_{k+1/k})$
◊– Added $a_{new}$ to the state vector $\hat{X}_{k+1/k+1}$.
◊– Use "(22)," to initialize a posteriori measurement
$e_{Z,new}$ of the new landmark $a_{new}$.
$e_{Z,new} = z_{k+1} - h(\hat{X}_{k+1/k+1}, a_{new})$
◊– Incremented of 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**

## VI. SIMULATION, EXPERIMENTS AND DISCUSSION

We present our simulation results to validate the proposed SVSF-SLAM for Unmanned Ground Vehicle localization problem. The results of our approach are compared with EKF-SLAM navigation filtering. The sampling rates used for each filter and sensors used in this study are as follows:

$$f_{Odometer} = f_{Laser} = f_{EKF-SLAM} = f_{SVSF-SLAM} = 10Hz$$

The simulation results provided in the following figures represent the estimated UGV position obtained using the EKF, and SVSF filters respectively with $\sigma_x = \sigma_y = 10^{-4}m$, $\sigma_\theta = 10^{-4}rad$, the convergence rate matrix $\gamma = diag(0.8, 0.8)$, and the width of the smoothing boundary layer vector used is $\varphi = [10; 12]$

### *A. First experiment: with white centered Gaussian noise*

In the first experiment we assume a white centered Gaussian noise, for process and observation model where $\sigma_v = 0.1\,m/s, \sigma_\omega = 0.25\,rad/s, \sigma_\rho = 0.1m, \sigma_\beta = 0.25rad$

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

The EKF-SLAM and SVSF-SLAM given low and high levels of the noise, such as centered Gaussian noise, non-centered Gaussian noise or colored and affected noise by a variable bias are shown in figures Fig. 5, Fig. 8 and Fig. 11.While both algorithms generate accurate maps when control noise is white centered Gaussian as shows in Fig. 5. But when control noise is not centered Gaussian, the EKF-SLAM fails catastrophically with high error and can be affected comparing to the SVSF-SLAM algorithm. In this case, the map generated by SVSF-SLAM with non-centered Gaussian noise is not degraded in quality.

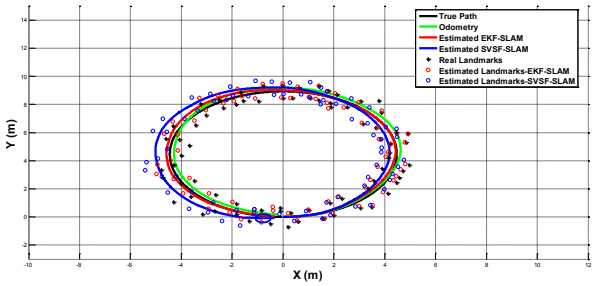


Fig. 5. Estimated robot trajectory using EKF/SVSF-SLAM algorithms with white centered Gaussian noise

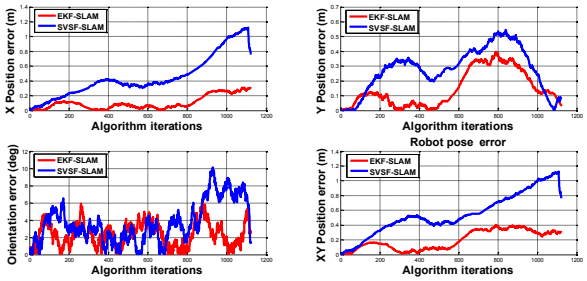


Fig. 6. Estimation error of SLAM by EKF/SVSF-SLAM algorithms with white centered Gaussian noise

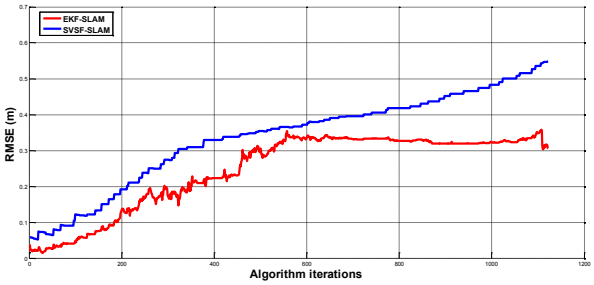


Fig. 7. RMSE for the EKF/SVSF-SLAM algorithms with white centered Gaussian noise

The position errors are shown in figures Fig. 6, Fig. 9 and Fig. 12, were found that the SVSF-SLAM performed significantly better than the EKF-SLAM, in terms of estimation error. The SVSF was able to overcome the lack of information after a few time steps, and provide a relatively estimate. The figures (Fig. 6, Fig. 9 and Fig. 12) show the estimation error of SLAM by both approaches along the axes $X$, $Y$ and $\theta$. It is clear, that the estimated uncertainty is very small of SVSF-SLAM compared to the true uncertainty specifically for the $X$, $Y$ and $\theta$ states of the EKF-SLAM in the second and in the third experiments. At the, we observe the increase of the SVSF-SLAM consistency compared to EKF-SLAM, because of the loop closure detection. When the robot closes the loop at the end, revisiting old landmarks should affect the positions of landmarks all around the loop, causing the overall error in the map to decrease. It is easy to see that the inconsistency problem is well solved, even robot frequently moves into pre-visited regions.

### *B. Second experiment: with non-centered Gaussian noise*

In this experiment we assume a white noise with bias, for process and observation model where $\sigma_v = 0.1\,m/s, \sigma_\omega = 0.08\,rad/s, \sigma_\rho = 0.045m, \sigma_\beta = 0.045rad$

$$Q_k = \begin{bmatrix} (0.1)^2 & 0 \\ 0 & (0.08)^2 \end{bmatrix}, R_k = \begin{bmatrix} (0.045)^2 & 0 \\ 0 & (0.045)^2 \end{bmatrix}$$

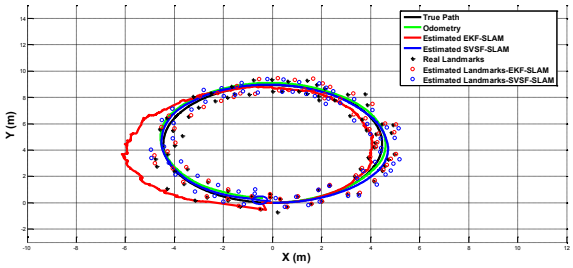


Fig. 8. Estimated robot trajectory using EKF/SVSF-SLAM algorithms with non centered Gaussian noise

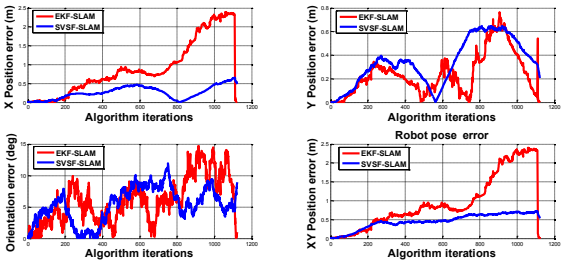


Fig. 9. Estimation error of SLAM by EKF/SVSF-SLAM algorithms with non centered Gaussian noise

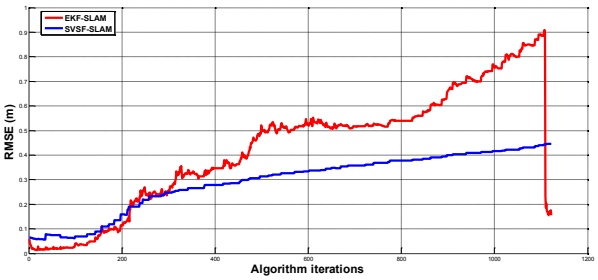


Fig. 10. RMSE for the EKF/SVSF-SLAM algorithms with non centered Gaussian noise

### *C. Third experiment: with colored noise*

In this experiment and in order to see the robustness of the SVSF-SLAM algorithm, we assume a colored noise, for process and observation model where $\sigma_v = 0.2\,m/s, \sigma_\omega = 0.15\,rad/s, \sigma_\rho = 0.02m, \sigma_\beta = 0.02rad$

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

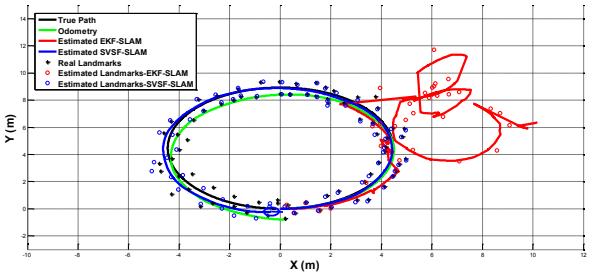


Fig. 11. Estimated robot trajectory using EKF/SVSF-SLAM algorithms with colored noise

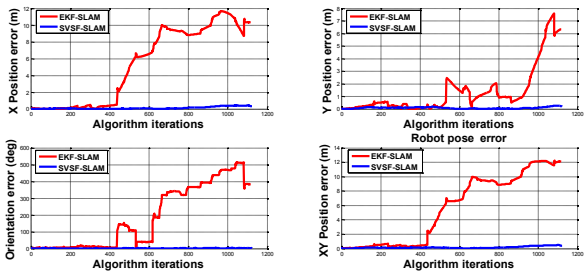


Fig. 12. Estimation error of SLAM by EKF/SVSF-SLAM algorithms with colored noise

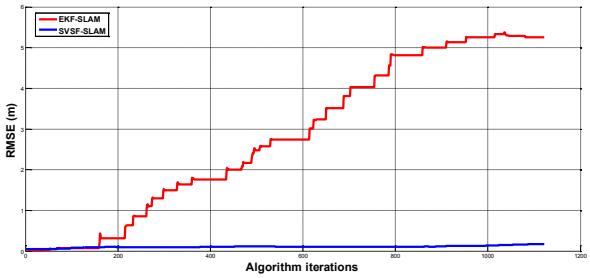


Fig. 13. RMSE for the EKF/SVSF-SLAM algorithms with colored noise

978-0-9567157-6-0 © IEEE 2016

Figure 7, Fig. 10 and Fig. 13, show the results of an experiment comparing the convergence RMSE of SVSF-SLAM and EKF-SLAM. At the robot goes around the loop, error should gradually build up in the map. The results of the first experiment comparing the RMSE of SVSF-SLAM and EKF-SLAM, in the presence of white Gaussian noise, we see that the RMSE was shown the best result, in this case, with white centered Gaussian noise, as can be seen from Fig. 6, EKF-SLAM algorithm gives the best results position and is more accurate than SVSF-SLAM algorithm. This can interpreted as when the system and observation models are accurate beside when the process and observation noises are uncorrelated zero mean Gaussian with known covariance the EKF gives a good accuracy for position estimation. However the SVSF-SLAM provides an accurate estimate than EKF-SLAM, when we use non centered Gaussian noise as shown in the Fig. 10 and Fig. 13. These results clearly validate the advantage of the SVSF-SLAM over the EKF-SLAM especially when the system or observation models are not accurate enough and the process and observation noises are non-centered Gaussian noise.

We see that after many time steps the uncertainty grows, in the almost experiment, Both algorithms were able to successfully detect Loop closing. When revisiting known landmarks, uncertainty decreases and the whole map is re-estimated.

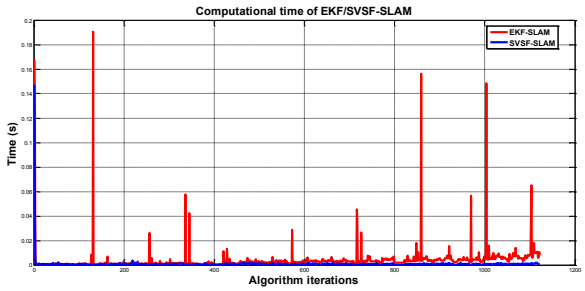


Fig. 14. Computational time of EKF/SVSF-SLAM algorithms

Figure 14 shows the computational time of the EKF/SVSF-SLAM approaches. The computational time of the EKF-SLAM 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 nearly constant and independent of the size of the state vector.

## VII. CONCLUSION

In this paper the development and validation of an alternative robust Odometer/Laser filtering scheme based on nonlinear SVSF for UGV localization and mapping problem are proposed. The proposed SVSF-SLAM algorithm is validated and compared to the EKF-SLAM algorithm. Good results are obtained by the SVSF-SLAM compares to the EKF-SLAM especially when the noise is colored or affected by a variable bias. The comparison proves the efficiency and robustness of the UGV localization and mapping process using the SVSF.

As future work, to solve the SLAM problem, we propose to use the Adaptive SVSF (ASVSF) filter in new form using matrix covariance to evaluate the uncertainty of the estimation.

## REFERENCES

[1] M. P. Golombek, R. A. Cook, T. Economou, W. M. Folkner, A. F. Haldemann and al. "Overview of the mars pathfinder mission and assessment of landing site predictions" Science 278 (5344): 1743-1748, December 1997.

[2] S. Thrun, D. Ferguson, D. Haehnel, M. Montemerlo, and W. Burgard R. Triebel., "A system for volumetric robotic mapping of abandoned mines", In proceedings of the IEEE International Conference on Robotics and Automation (ICRA'03), 2003.

[3] J. Leonard, J.D. Tardos, S. Thrun, and H. Choset, Eds., Workshop Notes of the ICRA Workshop on Concurrent Mapping and Localization for Autonomous Mobile Robots (W4),ICRA Conference, Washington, 2002.

[4] C. Thorpe and H. Durrant-Whyte. "Field robots." ISRR-2001.

[5] P. Moutarlier and R. Chatila. "An experimental system for incremental environment modeling by an autonomous mobile robot." In 1st International Symposium On Experimental Robotics, Montreal, 1989a.

[6] Williams SB, G Dissanayake and H Durrant-Whyte. "Constrained Initialisation of the Simultaneous Localization and Mapping Algorithm". 3rd International Conference on Field and Service Robotics." Service Robotics (FSR 2001) June 10-13, Espoo, Finland, pp 315-320.

[7] J. Guivant and E. Nebot. "Optimization of the simultaneous localization and map building algorithm for real time implementation." IEEE Transaction of Robotic and Automation, June 2001.

[8] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse, "Mono SLAM: Real-Time Single camera SLAM," In Conference on Pattern Analysis and Macine Intelligence, vol. 29, 2007.

[9] Hugh Durrant-Whyte and Time Bailey, "Simultaneous Localisation and Mapping: Part I," Robotics & Automation Magazine, vol. 13, pp. 99-110, june 2006.

[10] S. Holmes, G. Klein, and D. W. Murray, " A square root unscented Kalman filter for visual monoSLAM," . In Proc. of the IEEE International Conference on Robotics and Automation, p. 3710-3716, Pasadena, CA, May 2008.

[11] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit, "FastSLAM 2.0 an improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that provably Converges," In proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, Acapulco, 2003.

[12] Fredrik Orderud, "Comparison of Kalman Filter estimation approaches for state space models with Nonlinear measurements," In Scandinavian Conference on Simulation and Modeling, 2005.

[13] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. "FastSlam: A Factored Solution to the Simultaneous Localization and Mapping Problem." AAAI, 2002.

[14] F. Demim, A. Nemra, K. Louadj," Robust SVSF-SLAM for Unmanned Vehicle in Unknown Environment," In Proc. of the 7th IFAC Symposium on Mechatronic Systems, Loughborough University, UK, pp. 386-394, September 5-8, 2016.

[15] S. R. Habibi, "The Smooth Variable Structure Filter," Proceedings of the IEEE, vol. 95, no. 5, pp. 1026-1059, 2007.

[16] S. R. Habibi and R. Burton, "The Variable Structure Filter," Journal of Dynamic Systems, Measurement, and Control (ASME), vol. 125, pp. 287-293, September 2003.

[17] S. A. Gadsden, D. Dunne, S. R. Habibi and T. Kirubarajan, "Comparison of Extebded and Unscented Kalman, Particle, and Smooth Variable Structure Filters on a bearing-Only Target Tracking Problem," Proc. Of SPIE Vol. 7445, 74450B-1, 2013.

J. Sfeir., Navigation d'un Robot Mobile en Environnement Inconnu Utilisant les Champs de Potentiels Artificiels, Maıtrise en génie de la production automatisée, ´Ecole de Technologie Supérieure, Montréal, Canda, 2009.

978-0-9567157-6-0 © IEEE 2016