Research Article

Advances in Mechanical Engineering

Advances in Mechanical Engineering
2018, Vol. 10(1) 1–25
© The Author(s) 2018
DOI: 10.1177/1687814017736653
journals.sagepub.com/home/ade
SAGE

Simultaneous localization, mapping, and path planning for unmanned vehicle using optimal control

Demim Fethi[1], Abdelkrim Nemra[1], Kahina Louadj[2] and Mustapha Hamerlain[3]

Abstract

Among the huge number of functionalities that are required for autonomous navigation, the most important are localization, mapping, and path planning. In this article, investigation of the path planning problem of unmanned ground vehicle is based on optimal control theory and simultaneous localization and mapping. A new approach of optimal simultaneous localization, mapping, and path planning is proposed. Our approach is mainly affected by vehicle's kinematics and environment constraints. Simultaneous localization, mapping, and path planning algorithm requires two main stages. First, the simultaneous localization and mapping algorithm depends on the robust smooth variable structure filter estimate accurate positions of the unmanned ground vehicle. Then, an optimal path is planned using the aforementioned positions. The aim of the simultaneous localization, mapping, and path planning algorithm is to find an optimal path planning using the Shooting and Bellman methods which minimizes the final time of the unmanned ground vehicle path tracking. The simultaneous localization, mapping, and path planning algorithm has been approved in simulation, experiments, and including real data employing the mobile robot Pioneer $\mathrm{P3-AT}$. The obtained results using smooth variable structure filter–simultaneous localization and mapping positions and the Bellman approach show path generation improvements in terms of accuracy, smoothness, and continuity compared to extended Kalman filter–simultaneous localization and mapping positions.

Keywords

Localization, map building, unmanned ground vehicle, path planning, optimal control, Shooting's method, Pontryagin's principle, Bellman's method

Date received: 21 December 2016; accepted: 23 August 2017

Handling Editor: Francesco Massi

Introduction

In this article, an optimal path planning problem for unmanned vehicle navigation is investigated. This topic is once in a lifetime of diverse categories of control theory; however, it is one of serious researches. Path planning for unmanned ground vehicle (UGV) is very interesting for many applications (monitoring, reconnaissance, mapping, etc.). Moreover, optimal path planning requires accurate and robust UGV localization. For this purpose, we propose in this article a new approach for simultaneous localization, mapping, and

[1]Laboratoire Robotique et Productique, Ecole Militaire Polytechnique (EMP), Algiers, Algeria
[2]Laboratoire d'Informatique, de Mathématiques, et de Physique pour l'Agriculture et les Forêts (LIMPAF), University of Bouira, Bouira, Algeria
[3]Division Productique et Robotique, Center for Development of Advanced Technologies (CDTA), Algiers, Algeria

Corresponding author:
Demim Fethi, Laboratoire Robotique et Productique, Ecole Militaire Polytechnique (EMP), BP 17, Bordj El Bahri, Algiers 16111, Algeria.
Email: demifethi@gmail.com

path planning (SLAMPP). This system contains two important parts, accurate localization and optimal path planning.

First, a new solution of UGV localization is considered, based on the simultaneous localization and mapping (SLAM) utilizing the smooth variable structure filter (SVSF). SLAM algorithm is the mechanism that helps a UGV to localize itself and construct a map where globally accurate position data (e.g. global positioning system (GPS)) are not accessible. In this respect, UGV-SLAM solution is crucial and has shown significant promise for remote exploration.[1,2]

The SLAM framework based on the stochastic map approach was introduced in a seminal paper by Smith and Cheeseman in 1986. The SLAM techniques use in formation from onboard robot sensors to provide feature relative localization give an apriori map of the environment using the extended (EKF) which is presented Moutarlier and Chatila.[3] This approach focuses on stochastic estimation techniques "extended EKF" to estimate the position of the robot and to construct the map. The work on the EKF-SLAM algorithm implementations can be found in different environments, such as indoors, outdoors, aerial space, and underwater submarine.[4,5] In addition, the update of the EKF covariance matrix increases quadratically in the size of the number of detecting landmarks.[6–8] A nonlinear version of the Kalman filter proposed in Davison[9] is the unscented Kalman filter (UKF) that employs a particular representation of a Gaussian random variable in (N) dimensions using ($2N + 1$) samples, labeled sigma points. Among the advantages of the UKF, noise can be considered in a nonlinear mode to account for non-Gaussian noise. On the other hand, it suffers less from linearization, although it is not exempt. Like EKF, the UKF does not fully recover from poor landmarks.

Another algorithm of estimation is known as Fast-SLAM. This algorithm uses a Rao-Blackwellized Particle Filter to estimate the robot's pose and map. Each particle represents a possible robot's position. The uncertainty about the knowledge of this position is represented by the distribution of these particles in space. The implementation of the Fast-SLAM algorithm is resolved successfully with thousands of landmarks and compared to the EKF-SLAM algorithm which can use only a few hundreds of landmarks. The Fast-SLAM is demonstrated that it is not suitable for real-time implementation.[10,11] In 2007, the progress of a new predictor–corrector estimator was established on the variable structure theory and sliding mode concepts.[11–13]

In this article, we investigate this new approach, known as the SVSF. It is used to solve SLAM problems. It provides accurate and stable solution without any assumptions about noise characteristics. The SVSF can compromise with uncertainties and modeling errors of odometer/laser sensor system of UGV. It is robust and stable to modeling uncertainties making it suitable for UGV localization and mapping problem. The strategy of SVSF-SLAM algorithm is presented, this new concept ensures the stability and robustness faculties in modeling uncertainties.

In the second part of this work, a path planning algorithm using two methods (direct and indirect methods) is investigated. Indirect methods based on the Pontryagin principle of maximum[14–16] are known for their speed and precision in the treatment of optimal control problems. The Shooting method[16] is an indirect method of optimization which attempts to solve the true optimization problem by solving the two-point boundary value problem (TPBVP).[16] The initial conditions sensitive to the co-state equations cause a big problem with indirect methods. Therefore, direct methods are often used to initialize indirect methods when solving function optimization problems.[17] Dynamic programming (DP) based on the Bellman principle is the best which solves the direct method problem, by reducing the initial hypothesis, and bringing a discrete time approximation to the optimal function of time, since DP algorithms are of order (N) in the parameter optimization.[17,18]

In this article, a robust system for localization, mapping, path planning, and optimal control (see Figure 1) is developed. The local navigator is a susceptible process that depends on the current sensor data to keep vehicle safe and stable during movement quick as possible. The path planning with optimal control using the Shooting and Bellman method based on different approaches of localization, as reviewed in this work, is a deliberative technique which regards forward to the future and uses information about the world to produce a safe path for the robot.

This article is organized as follows. Section "Process model of UGV" depicts the process model of UGV. In section "EKF-SLAM algorithm," the EKF-SLAM algorithm is detailed and compared with the proposed SVSF-SLAM algorithm. The implementation of this latter is presented in section "SVSF-SLAM algorithm." In section "Optimal control," we describe path planning with optimal control. Finally, results and discussion are provided in section "Results and discussion" followed by conclusion and perspectives.

Process model of UGV

The Pioneer P3-AT which used in this work is a non-holonomic robot with small four-wheel (see Figure 2). It is a differential drive robot, offers an installed *PC* alternative, opening the path for locally available vision handling, Ethernet-based interchanges, laser (see Figure

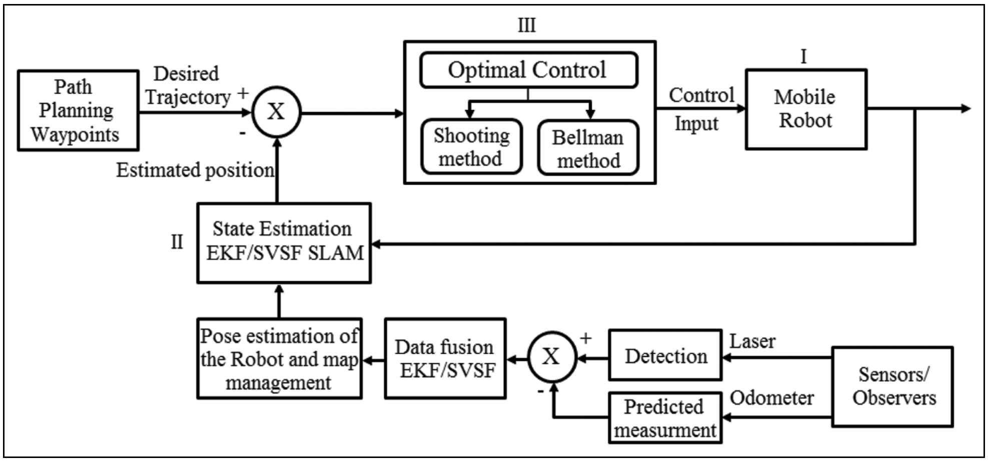

Figure 1. State estimation EKF/SVSF-SLAM and optimal control for mobile robot.

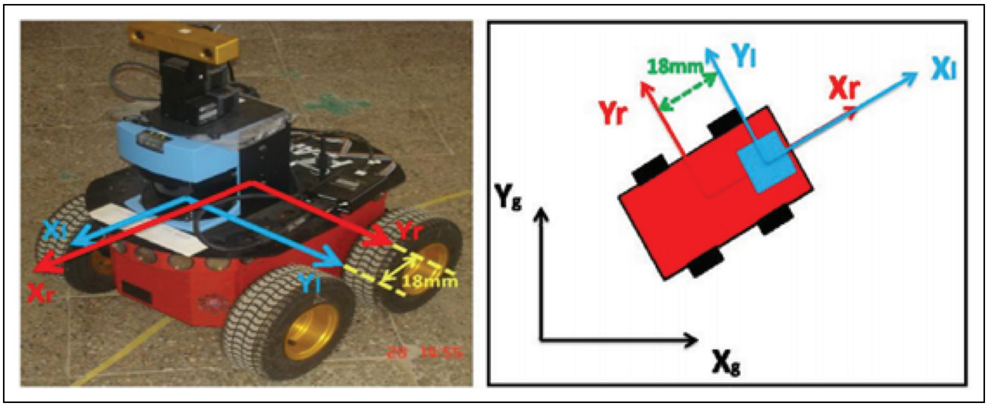

Figure 2. Mobile robot (P3-AT) Robotic Laboratory—EMP.

3), DGPS (differential global positioning system), and different independent capacities. P3-AT's powerful motors and can reach speeds of 0.8 m/s and carry a payload of up to 12 kg. Let the vector $[X_r; Y_r; \theta_r]^T$ 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[15]

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

From Figure 4 and Sfeir,[15] the state transition equation of the robot is given by

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

where (v_x, v_y) are translation velocities of the robot following X_L axes and Y_L axes, respectively, and w is the rotation velocity. Mobile robot's model and measurement observation model are used to describe the motion and the status of the robot.

Figure 3. SICK 2D laser range finder (LMS-200).

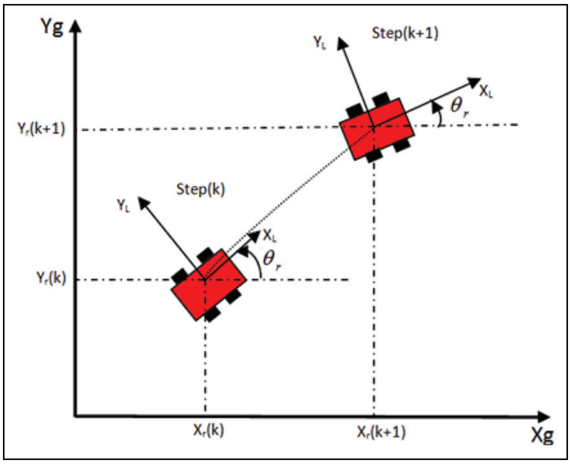

Figure 4. Kinematic model of the UGV.

When $v_y = 0$, the kinematic model of the robot will be

$$\begin{pmatrix} \dot{X}_r \\ \dot{Y}_r \\ \dot{\theta}_r \end{pmatrix} = \begin{pmatrix} v\,cos(\theta_r) \\ v\,sin(\theta_r) \\ w \end{pmatrix} + \begin{pmatrix} \varepsilon_{x_r} \\ \varepsilon_{y_r} \\ \varepsilon_{\theta_r} \end{pmatrix} \tag{3}$$

The odometry depends on the assumption which is only true if there is no slipping in the robot's wheels. The angular velocity delivered by the gyroscope can be integrated to provide the orientation $(\theta_r(k+1) = \theta_r(k) + w\Delta T)$. In this case, w is obtained from the gyroscope which has a large drift error. It can be modeled by an exponential curve and determined experimentally as done by Barshan and Durrant-Whyte.[19]

When $v = v_x$, and by discretizing equation (3), the model will be written as follows

$$\begin{pmatrix} X_r(k+1) \\ Y_r(k+1) \\ \theta_r(k+1) \end{pmatrix} = \begin{pmatrix} X_r(k) + \Delta T v_k\,cos(\theta_r(k)) \\ Y_r(k) + \Delta T v_k\,sin(\theta_r(k)) \\ \theta_r(k) + \Delta T w(k) \end{pmatrix} + \begin{pmatrix} \varepsilon_{x_r} \\ \varepsilon_{y_r} \\ \varepsilon_{\theta_r} \end{pmatrix} \tag{4}$$

ΔT is time step and $\varepsilon_{x_r y_r \theta_r}$ is the noise that arises from the encoder and wheels slipping. The robot evolution model reflects the relationship between the robot previous states $X_R(k)$ and its current state $X_R(k+1)$. Let $U_k = [v, w]^T$ be control vector, which is assigned with the robot (translation and rotation velocity). During navigation, the robot moves on a 2D plane. In SLAM, the system state vector has a position of the UGV (X_R). It is represented by $X_R = [X_r, Y_r, \theta_r]^T \in \mathbf{R}^3$ and the map is presented by $m_i = [x_1, y_1; \ldots; x_M, y_M]^T \in \mathbf{R}^{2M}$ with M is the total number of landmarks. We can write equation (4) as follows

$$X_R(k+1) = f(X_R(k), U(k)) + \varepsilon_{x_r y_r \theta_r} \tag{5}$$

Direct observation point–based model

m_i presents the landmark which is detected in the environment of the map. If a landmark m_i is detected for the first time, it will be initialized and added to the state vector. If it is already observed, then the UGV uses it to update the robot pose state.[20,21] In the case of point-based landmark representation, each point of the scan is considered as 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[20]

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

The overall transformation to local inverse model is given as follows

$$\begin{cases} X_{rl} = (X_{m_{ig}} - X_r)cos(\theta_r) + (Y_{m_{ig}} - Y_r)sin(\theta_r) \\ Y_{rl} = -(X_{m_{ig}} - X_r)sin(\theta_r) + (Y_{m_{ig}} - Y_r)cos(\theta_r) \end{cases} \tag{7}$$

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

The subscript i is the ith sample in landmark sample sets. The observation is given by $Z_{ik} = [\rho_{ik}, \beta_{ik}]^T$. It contains the range and bearing of a landmark in the local frame. The direct observation model is expressed as

$$Z_i = \begin{bmatrix} \sqrt{(X_{m_{ig}} - X_r)^2 + (Y_{m_{ig}} - Y_r)^2} \\ tan^{-1}\left(\frac{Y_{m_{ig}} - Y_r}{X_{m_{ig}} - X_r}\right) - \theta_r \end{bmatrix} + \begin{bmatrix} \varepsilon_{\rho_i} \\ \varepsilon_{\beta_i} \end{bmatrix} \tag{8}$$

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

Inverse observation point–based model

The landmark mapping model is an inverse observation model. Knowing the robot state and the landmark coordinate, the inverse observation model $X_{m_{new}}$ can be given by[20,22]

$$X_{m_{new},k} = h_i^{-1}(X_k, Z_{ik}) \tag{9}$$

$$X_{m_{new}}(k) = \begin{bmatrix} X_r + \rho_i\,cos(\theta_r)cos(\beta_i) - \rho_i\,sin(\theta_r)sin(\beta_i) \\ Y_r + \rho_i\,sin(\theta_r)cos(\beta_i) + \rho_i\,cos(\theta_r)sin(\beta_i) \end{bmatrix} \tag{10}$$

where h represents a continuous observation model.

Direct observation line–based model

The direct observation model is given by[20,23]

$$Z_{i,\,line} = \begin{bmatrix} \rho_{il} \\ \beta_{il} \end{bmatrix} = \begin{bmatrix} \rho_{ig} - \sqrt{X_r^2 + Y_r^2} cos\left(\beta_{ig} - tan^{-1}\left(\frac{Y_r}{X_r}\right)\right) \\ \beta_{ig} - \theta_r \end{bmatrix} \tag{11}$$

$[\rho_{il}, \beta_{il}]^T$ is the polar parameters in the local coordinate frame of the ith line feature and $[\rho_{ig}, \beta_{ig}]^T$ the polar parameters in the global coordinate frame of the ith line feature.

Inverse observation line–based model

The mapping model is an inverse observation model $m_{new,\,line}$ which can be written as follows

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

There are many types of features association “line to line,”[24,25] “point to line,”[26] and “point to point.”[24,27] In this article, we will do a comparative’s survey between two types of association “point to point” and “line to line.” Since the processing time of the EKF/SVSF-SLAM approach “point to point” is very important and allocates a large memory space, thus, in the association step, we propose to use a method based on “line to line” that is quicker than the standard “point-to-point” approach. In our work, we used the method “Split and Merge” which is largely used for line extraction.[24,25]

EKF-SLAM algorithm

The discretization form of the system and nonlinear model measurement used for state estimation can be written as follows

$$X_{k+1} = f(X_k, U_k) + \varepsilon_{x_r, y_r, \theta_r} \tag{13}$$

$$Z_{k+1} = h(X_k) + \varepsilon_{\rho_i, \beta_i} \tag{14}$$

The EKF provides an approximation of the optimal estimate. X_{k+1} is the state of the vector and U_k is the input control vector. The objective of the EKF-SLAM problem is to recursively estimate the state X_k of the landmark as stated by the measurement Z_{k+1}. The noise that arises from the encoder, wheels slipping is represented by $\varepsilon_{x_r, y_r, \theta_r}$ and $\varepsilon_{\rho_i, \beta_i}$ are the noise of measured range and the noise of bearing in the local frame which are having recognized by the covariances Q_k and R_k, respectively. The EKF Steps are presented as follows:

- Initial estimates for

Project the state $\hat{X}_{k+1}$ ahead

$$\hat{X}_{k+1} = f(\hat{X}_k, U_{k+1}) + \nabla F_X(X_k - \hat{X}_{k/k}) + \varepsilon_{x_r, y_r, \theta_r} \tag{15}$$

- Prediction

Project the error covariance ahead. The prediction covariance matrix $P_{k+1/k}$ is written as follows

$$P_{k+1/k} = \nabla F_X P_{k/k} \nabla F_X{}^T + \nabla F_U Q_k \nabla F_U{}^T \tag{16}$$

where ∇F_X and ∇F_U be the Jacobian matrices of $f(.)$ with respect to X_{k+1} evaluated at an elsewhere specified point denoted by

$$\nabla F_X = \begin{bmatrix} J_1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}, \quad \nabla F_U = \begin{bmatrix} J_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

where
$$J_1 = \left[\frac{\partial f(.)}{\partial x_r}; \frac{\partial f(.)}{\partial y_r}; \frac{\partial f(.)}{\partial \Theta_r}\right] = \begin{bmatrix} 1 & 0 & -v\,sin(\theta_r)\Delta T \\ 0 & 1 & v\,cos(\theta_r)\Delta T \\ 0 & 0 & 1 \end{bmatrix}$$

$$J_2 = \left[\frac{\partial f(.)}{\partial v}; \frac{\partial f(.)}{\partial w}\right] = \begin{bmatrix} cos(\theta_r)\Delta T & 0 \\ sin(\theta_r)\Delta T & 0 \\ 0 & \Delta T \end{bmatrix}$$

$$Q_k = \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_w^2 \end{bmatrix}, R_k = \begin{bmatrix} \sigma_{\rho_i}^2 & 0 \\ 0 & \beta_i^2 \end{bmatrix}$$

- Observation and update

Linearization of the observation model $\hat{Z}_{k+1}$ around $\hat{X}_{k+1/k}$

$$\hat{Z}_{k+1} = h(\hat{X}_{k+1/k}) + H_{k+1}(\hat{X}_{k+1/k} - X_{k+1}) + \varepsilon_{\rho_i, \beta_i} \tag{17}$$

The update cycle of the Kalman filter is applied to $\hat{X}_{k+1/k+1}$ and $P_{k+1/k+1}$

Compute the Kalman gain K_{k+1}

$$K_{k+1} = P_{k+1/k} H_{k+1}^T [H_{k+1} P_{k+1/k} H_{k+1}^T + R_k]^{-1} \tag{18}$$

Update estimate with measurement Z_{k+1}

$$\hat{X}_{k+1/k+1} = \hat{X}_{k+1/k} + K_{k+1}[Z_{k+1} - H_{k+1}\hat{X}_{k+1/k}] \tag{19}$$

Update error covariance $P_{k+1/k+1}$

$$P_{k+1/k+1} = [I - K_{k+1} H_{k+1}] P_{k+1/k} \tag{20}$$

Even as the environment is investigated, new features are detected and should be put on the accumulation map. In this case, the state vector and the output error estimate matrix are calculated from the new observation.

- Map management

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

Initialization of the covariance matrix $P_{new} \in R^{2*2}$ of a_{new}

$$P_{new} = \nabla h_R^{-1} P_R (\nabla h_R^{-1})^T + \nabla h_Z^{-1} R_k (\nabla h_Z^{-1})^T$$

where ∇h_R^{-1} and ∇h_Z^{-1} are the Jacobian matrices, denoted by

$$\nabla h_R^{-1} = \begin{bmatrix} 1 & 0 & -\rho_i\, sin(\theta_r + \beta_i) \\ 0 & 1 & \rho_i\, cos(\theta_r + \beta_i) \end{bmatrix}$$

and

$$\nabla h_Z^{-1} = \begin{bmatrix} cos(\theta_r + \beta_i) & -\rho_i\, sin(\theta_r + \beta_i) \\ sin(\theta_r + \beta_i) & \rho_i\, cos(\theta_r + \beta_i) \end{bmatrix}$$

Initialization of the inter-covariance matrix $P_{new,X} \in R^{2*3M}$ of a_{new}

$$P_{new,X} = \nabla h_R^{-1} P_{R,X}$$

Increment the state vector $\hat{X}_{Incremt}$

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

Increment the covariance matrix $P_{Incremt}$

$$P_{Incremt} = \begin{bmatrix} P & P_{X,new} \\ P_{new,X} & P_{new} \end{bmatrix}$$

$M = M + 1$ is the total number of current landmarks. The EKF diverge if the linearization consecutive is not good linearized.[26,28,29]

SVSF-SLAM algorithm

The variable structure filter (VSF) was created in 2003, as a new estimator using a sliding mode concept where switching gain is used to ensure that the estimates converge to true state values.[30] As shown in Figure 5, the SVSF utilizes a switching gain to converge the estimates to within a boundary of the true state values (i.e. existence subspace).[31,32]

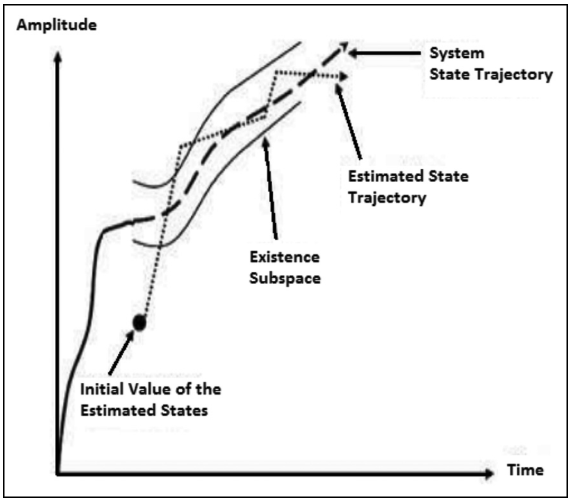
Amplitude
System State Trajectory
Estimated State Trajectory
Existence Subspace
Initial Value of the Estimated States
Time

Figure 5. SVSF estimation concept.

A corrective term, referred to as the SVSF gain, is computed as a function of the error in the predicted output, as well as a gain matrix and the smoothing boundary layer width.[30,32,33] The corrected term calculated in equation (21) is then used in equation (24) to find a posteriori state estimate. The priori and posteriori output error estimates are considered as critical variables which are defined by equations (25) and (26), respectively[32,33] The estimation progression makes a summary of equations (21)–(28) and is repeated iteratively. U_k is the input control vector. The gain is computed[32] as follows

$$\begin{aligned} K_{k+1}^{SVSF} = (\hat{H}_{k+1})^{+} diag[(|e_{z_{k+1/k}}|_{Abs} + \gamma |e_{z_{k/k}}|_{Abs})^{\circ} \\ Sat(\bar{\varphi}^{-1} e_{z_{k+1/k}})][diag(e_{z_{k+1/k}})]^{-1} \end{aligned} \tag{21}$$

where ° represents "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 measurement matrix that is derivative of h 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 ith landmark, where i is the detected landmark index at time k and j is the index of an individual landmark observation in the $h_{k+1,j}$. $F_{X,i}$ is calculated as follows

$$F_{X,i} = \begin{bmatrix} 1 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & 0 & \ldots & 0 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 0 & \underbrace{\ldots}_{2i-2} & 0 & 0 & 1 & 0 & \underbrace{\ldots}_{2M-2i} & 0 \end{bmatrix} \tag{22}$$

This equation $(H_{k+1})^{+}$ can be written as follows

$(H_{k+1})^{+} = (F_{X,i})^{T}(h_{k+1,j})^{+}$, with the knowledge that

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

- $\bar{\varphi}^{-1}$ is a diagonal matrix constructed from the smoothing boundary layer vector φ, such that

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

with M_i represents the number of measurements.

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

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

The update of the state estimates can be calculated as follows

$$\hat{X}_{k+1/k+1} = \hat{X}_{k+1/k} + K_{k+1}^{SVSF}\hat{e}_{z_{k+1/k}} \tag{24}$$

The priori and posteriori output error estimates are defined as follows

$$e_{z_{k+1/k}} = Z_{k+1} - \hat{Z}_{k+1/k} \tag{25}$$

$$e_{z_{k+1/k+1}} = Z_{k+1} - \hat{Z}_{k+1/k+1} \tag{26}$$

The SVSF filter provides a robust and stable estimate to modeling uncertainties and errors.[31,33]

The calculation of the SVSF gain is needed to use two main parameters. The first parameter is (γ) which control the speed of convergence, where the second (φ) refers to the boundary layer width which is used to smooth out the switching action.[30,32] These parameters must be suitably selected. γ is a diagonal matrix whose elements were set to the following[21,23]

$$0 < \gamma_i \leq 1 \tag{27}$$

As maintained by Sfeir,[15] the estimation process is stable and converges to the existence subspace if the following circumstances are contented[31]

$$|e_{k/k}|_{Abs} > |e_{k+1/k+1}|_{Abs} \tag{28}$$

Two association approaches are proposed: "point to point" and "line to line." The steps of SVSF-SLAM algorithm is illustrated in the algorithm 1.[20,22] In this work, we also treat the problem of optimal control, focusing on two main methods which have proven to be effective on mobile robots: the Shooting and Bellman methods.

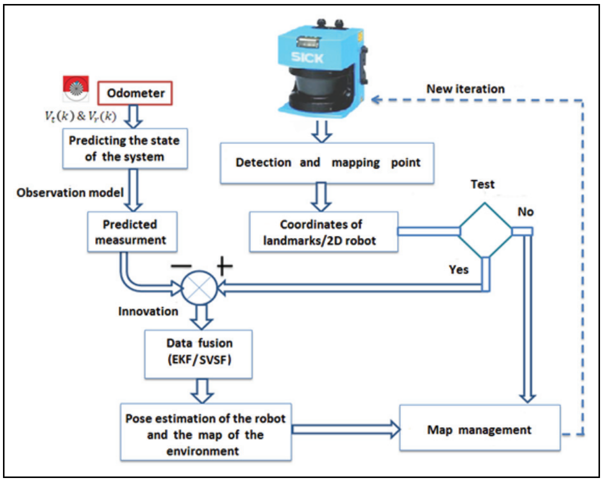
Odometer
$V_t(k)$ & $V_r(k)$
Predicting the state of the system
Observation model
Predicted measurment
Innovation
Data fusion (EKF/SVSF)
Pose estimation of the robot and the map of the environment
Map management
SICK
New iteration
Detection and mapping point
Coordinates of landmarks/2D robot
Test
No
Yes
−
+

Figure 6. Steps of the EKF/SVSF-SLAM principle.

Optimal control

In this section, we provide a path planning algorithm based on optimal control theory. In the beginning, we use the Shooting methods based on the Pontryagin's principle.[16,34] In this principle, we determine an optimality equation, that is, a differential algebraic equation system is provided with an initial condition and a final condition and the adjoint equations. In other words, we should note that in the adjoint equation, derived from the Pontryagin's principle, no information is given concerning the initial or the final conditions. Consequently, this equation is hard to be used numerically. Thus, in order to determine the initial condition of the adjoint state, we use the Shooting indirect method for the numerical procedure.[34] After this, we present the results of numerical experiments implemented using MATLAB facilities. For classical method problems, the solution can be obtained analytically or numerically. On the other hand, we use the Bellman principle which is the direct method and gives an appropriate condition to the optimality solution.[35–37]

Numerical solution with the Shooting method

The optimal control problem of the robot is stated as follows: Find the control $u = (v, w)$ that minimizes the final time (t_f)

$$J = t_f \rightarrow \min \tag{29}$$

subject to the differential constraints

$$\begin{cases} \dot{x}_r = v\cos\theta_r \\ \dot{y}_r = v\sin\theta_r \\ \dot{\theta}_r = w \end{cases} \tag{30}$$

The prescribed boundary conditions

Algorithm 1: SVSF-SLAM Algorithm

```
§ At each step time t
  ◇ Initial Estimate
    $\hat{X}_0$ and $e_{z_0}$
  ◇ Prediction Time Update
    ◇− Project state Ahead
    $\hat{X}_{k+1/k} = f(\hat{X}_{k/k}, U_k)$ % state prediction using process
    model
  ◇ Observation and Update
    ◇ For all features observation $Z_{k+1,j}$.
      ◇ if the correspondence is founded before seen the
        feature
    ◇− Calculate a posteriori measurements error
      $\hat{e}_{z_{k+1,i}} = Z_{k+1,j} - \hat{Z}_{k+1/k,i}$
      $(H_{k+1,j})^{+} = (F_{X,i})^{T}(h_{k+1,j})^{+}$
      ◇− Compute the SVSF Gain $K^{SVSF}_{k+1}$.
      $K^{SVSF}_{k+1} = (\hat{H}_{k+1,j})^{+} diag[(|\hat{e}_{z_{k+1,i}}|_{Abs} +$
      $\gamma|e_{z_{k,i}}|_{Abs})^{\circ} Sat(\bar{\varphi}^{-1}\hat{e}_{z_{k+1,i}})][diag(e_{z_{k+1,i}})]^{-1}$
      ◇− Update of the State estimate vector $\hat{X}_{k+1/k+1}$.
      $\hat{X}_{k+1/k+1} = \hat{X}_{k+1/k} + K^{SVSF}_{k+1}\hat{e}_{z_{k+1,i}}$
      ◇− calculate a priori measurements error vector
      $e_{z_{k+1,i}} = z_{k+1,j} - h(\hat{R}_{k+1}, m_i)$
      ◇End if
    ◇End For
  ◇ Map Management
    ◇For all non-associated features.
    ◇− Initial a new feature $a_{new}$.
    $a_{new} = h^{-1}(\hat{X}_{k+1/k+1}, Z_{k+1/k})$
    ◇− Add $a_{new}$ to the state vector $\hat{X}_{k+1/k+1}$.
    ◇− Use equation (28) to initialize a posteriori
        measurements error
    $e_{Z_{k+1,new}}$ of the new landmark.
    $e_{Z_{k+1,new}} = Z_{k+1} - h(\hat{X}_{k+1/k+1}, a_{new})$
    ◇− Add $e_{Z_{k+1,new}}$ to a posteriori measurement error
    vector $e_{Z_{k+1,i}}$.
    ◇− Increment the state vector $\hat{X}_{Incremt}$
    $\hat{X}_{Incremt} = [\hat{X}_{k+1}; a_{new}]$
    $M = M + 1$ % total number of current landmarks.
    ◇ End For
§ End
```

$$\begin{aligned} t_0 = 0, x_{r0} = 0, y_{r0} = 0, \theta_{r0} = 0 \\ x_{rf} = 0, y_{rf} = 0, \theta_{rf} = 0 \end{aligned} \tag{31}$$

$$U = \{(v, w)/-1 \le v \le 1, \ -1 \le w \le 1\} \tag{32}$$

Equation (29) is equivalent to

$$\int_0^{t_f} dt \rightarrow min \tag{33}$$

The Hamiltonian is given by

$$H(t, x(t), p(t), u(t)) = p_1(v\,cos\,\theta_r) + p_2(v\,sin\,\theta_r) + p_3 w - 1$$

The Euler–Lagrange equations are given by

$$\begin{cases} \dot{x}_r = \dfrac{\partial H}{\partial p_1} \\ \dot{y}_r = \dfrac{\partial H}{\partial p_2} \\ \dot{\theta}_r = \dfrac{\partial H}{\partial p_3} \\ \dot{p}_1 = -\dfrac{\partial H}{\partial x_r} \\ \dot{p}_2 = -\dfrac{\partial H}{\partial y_r} \\ \dot{p}_3 = -\dfrac{\partial H}{\partial \theta_r} \\ 0 = -\dfrac{\partial H}{\partial v} \\ 0 = -\dfrac{\partial H}{\partial w} \end{cases}$$

$$\begin{cases} \dot{x}_r = v\,cos\,\theta_r \\ \dot{y}_r = v\,sin\,\theta_r \\ \dot{\theta}_r = w \\ \dot{p}_1 = 0 \\ \dot{p}_2 = 0 \\ \dot{p}_3 = -[-p_1 v\,sin\,\theta_r + p_2 v\,cos\,\theta_r] \end{cases}$$

From Hamiltonian H, a control $u = (v, w)$ verifies

$$\begin{aligned} H(t, x(t), p(t), u(t)) &= \min_{u \in U} H(t, x(t), p(t), \\ u(t)) &= \min_{\{-1 \le v \le 1, -1 \le w \le 1\}} H(t, x(t), p(t), u(t)) \end{aligned}$$

The following equation is given to find a control U

$$\frac{\partial H}{\partial v} = p_1\,cos\,\theta_r + p_2\,sin\,\theta_r, \ \frac{\partial H}{\partial w} = p_3$$

then

$$\begin{aligned} p_1(t) cos\,\theta_r + p_2(t) sin\,\theta_r &= \min_{\{-1 \le v \le 1\}} \\ (p_1(t) cos\,\theta_r + p_2(t) sin\,\theta_r), p_3(t) &= \min_{\{-1 \le w \le 1\}} p_3(t) \end{aligned}$$

Consequently, we deduce

$$v(t) = sign(p_1\,cos\,\theta_r + p_2\,sin\,\theta_r), w(t) = sign(p_3(t))$$

In the numerical solution, we used the Shooting indirect method to determine a final time. Then we have to solve the following system

$$\begin{cases} \dot{v}_1 = v\,cos v_3 \\ \dot{v}_2 = v\,sin v_3 \\ \dot{v}_3 = w \\ \dot{v}_4 = 0 \\ \dot{v}_5 = 0 \\ \dot{v}_6 = v_4\,v\,sin v_3 - v_5\,v\,cos v_3 \\ v = sign(p_1 \cos\theta_r + p_2 \sin\theta) \\ w = sign(p_3(t)) \\ v_1(0) \in \mathbf{R}, v_2(0) \in \mathbf{R}, v_3(0) \in \mathbf{R} \\ v_4(0) \in \mathbf{R}, v_5(0) \in \mathbf{R}, v_6(0) \in \mathbf{R} \end{cases}$$

Let $v(t, x_{r0}, y_{r0}, \theta_{r0}, p_1, p_2, p_3)$ be the solution of the previous system at time t with the initial condition $(v_1(0), v_2(0), v_3(0), v_4(0), v_5(0), v_6(0))$. We construct a Shooting function which is a nonlinear algebraic equation of the variable ψ at time $t = 0$. The equation of Shooting method is computed by integrating the differential equation (using a numerical solution for example Euler method, Runge–Kutta method). The Shooting function is defined by

$$\varphi(p) = \begin{pmatrix} v_1(t_f,\ x_{r0},\ y_{r0},\ \theta_{r0},\ p_1,\ p_2,\ p_3) - x_{rf} \\ v_2(t_f,\ x_{r0},\ y_{r0},\ \theta_{r0},\ p_1,\ p_2,\ p_3) - y_{rf} \\ v_3(t_f,\ x_{r0},\ y_{r0},\ \theta_{r0},\ p_2,\ p_3,\ p_4) - \theta_{rf} \end{pmatrix}$$

Solving the problem, we can write: Find $p(0)$ such that $\varphi(p(0))$ gives the desired value of $x(t_f) = (x_{rf}, y_{rf}, \theta_{rf})$. The algorithm for numerical solution of this problem will then be completely defined if one gives oneself:

1. The integration algorithm of a differential system with initial condition (i.e. the Euler or Runge–Kutta procedure) to compute the Shooting function φ (implemented in "ode45" of MATLAB which is a method of Runge–Kutta 4/5 with variable pitch).
2. The solution algorithm $\varphi(v) = 0$ which in our case uses the quasi-Newton method (implemented in "fsolve" of MATLAB).

Numerical solution with the Bellman method

In this section, we use the Bellman principle to solve the same system with free final time which is the direct method. We consider the control of a discrete time dynamical system

$$\begin{cases} x_r(i+1) = x_r(i) + t_f/Nv(i)cos(\theta_r(i)) \\ y_r(i+1) = y_r(i) + t_f/Nv(i)sin(\theta_r(i)) \\ \theta_r(i+1) = \theta_r(i) + tf/Nw(i) \end{cases} \tag{34}$$

where $i = \overline{1, N}$, N is a number of steps. The values function is defined by

$$t_f = inf\{t | x(.) = (x_r(.), y_r(.), theta_r(.))\} \tag{35}$$

where $x(.)$ is solution of equation (34). To determine a control (v, w), we use the Pontryagin's principle.

$$\begin{cases} p_1(j+1) = p_1(j) + t_f/N \times 0 \\ p_2(j+1) = p_2(j) + t_f/N \times 0 \\ p_3(j+1) = p_3(j) + t_f/N \times (p_1(j+1)v(i)sin(\theta(i)) \\ \qquad\qquad + p_2(j+1)v(i)cos(\theta(i))) \end{cases} \tag{36}$$

where $i = \overline{1, N}, j = \overline{1, N-i+1}$, and t_f is the final time established by each method (SVSF-SLAM, EKF-SLAM, and the odometer). The initial condition of adjoint system is the values founded by the Shooting method. Then, the control (v, w) is defined as follows

$$\begin{cases} v(i) = sign(p_1(i)cos(\theta_r(i)) + p_2(i)sin(\theta(i))) \\ w(i) = sign(p_3(i)) \end{cases} \tag{37}$$

where $i = \overline{1, N}$.

Results and discussion

SLAM results

Simulation results. The simulation results of the proposed SVSF-SLAM algorithm for UGV localization problem are presented. The results of our proposed algorithm are compared with EKF-SLAM algorithm. The sampling rates used for each filter and sensors used in this study are as follows[20]

$$f_{odometer} = f_{laser} = f_{EKF-SLAM} = f_{SVSF-SLAM} = 10\,\text{Hz}$$

Figure results which are presented show the estimated position of the UGV using EKF and SVSF, with $\sigma_x = \sigma_y = 10^{-4}\,\text{m}, \sigma_\theta = 10^{-4}\,\text{rad}$, the convergence rate matrix $\gamma = diag(0.8, 0.8)$, and the width of the smoothing boundary layer vector used is $\varphi = [10; 12]$.

- Test 1: zero-mean Gaussian noise

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

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

- Test 2: colored noise

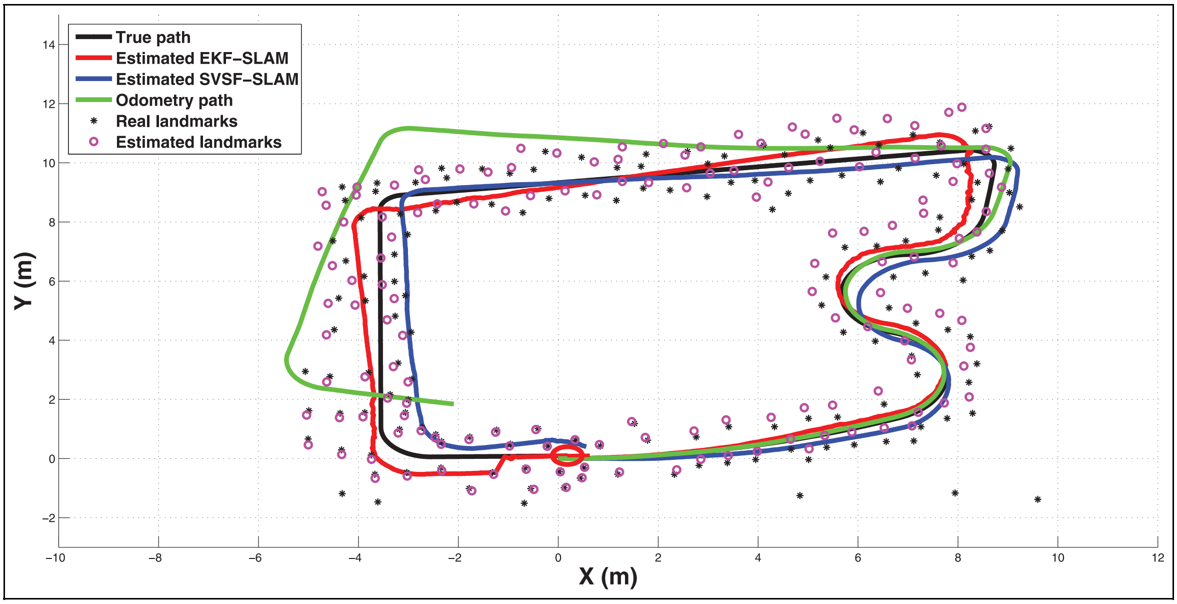

Figure 7. Pose estimation with zero-mean Gaussian noise.

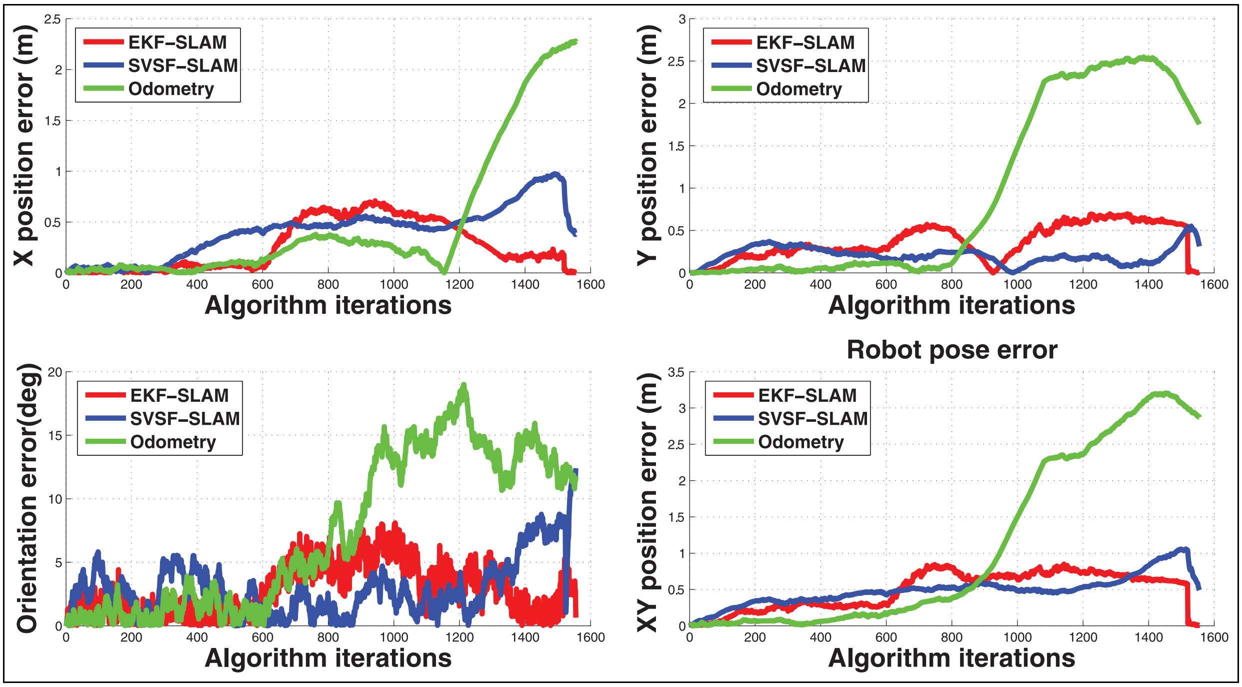

Figure 8. Error position of the robot with zero-mean Gaussian noise.

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

$$Q_k = \begin{bmatrix} (0.1)^2 & (0.045)^2 \\ (0.045)^2 & (0.08)^2 \end{bmatrix}, R_k \begin{bmatrix} (0.045)^2 & (0.03)^2 \\ (0.03)^2 & (0.045)^2 \end{bmatrix}$$

Figure 7 shows proposed algorithms and provide accurate positions when process and measurement noise are zero-mean and Gaussian. The position errors are shown in Figure 8. We note that the EKF-SLAM algorithm gives higher quality in estimation error terms compared with SVSF-SLAM algorithm. At 1550 iteration, we note significant increase of the SVSF-SLAM

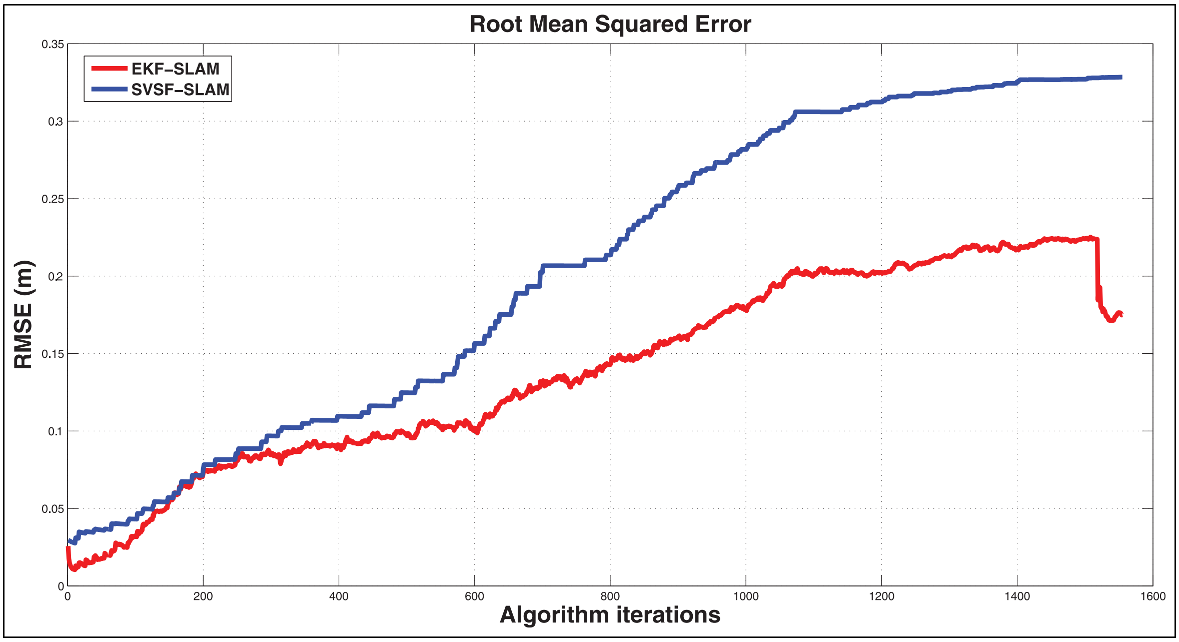

Figure 9. Position RMSE under zero-mean Gaussian noise.

algorithm consistency in comparison to EKF-SLAM algorithm caused by closing-loop detection. The robot closes the loop making the error in the environment to decrease at 1550 iteration when it frequently moves into pre-visited regions.

In the first test, by looking at the root mean square error (RMSE) of EKF/SVSF-SLAM algorithms, within the sight of zero-mean Gaussian noise, we can show that the EKF notices better results with zero-mean Gaussian noise. This can be found in Figure 9. EKF-SLAM algorithm gives the best position results and is more accurate than SVSF-SLAM algorithm. This can be translated as follows: In this case, the EKF gives a good accuracy when the process and observation noise are uncorrelated zero-mean Gaussian with known covariance. On other hand, when the control noise is colored, the EKF-SLAM algorithm is unsuccessful with high error in comparison to the SVSF-SLAM algorithm. For this situation, the map created by the proposed SVSF-SLAM algorithm with colored noise is not debased in quality as can be seen in Figure 10.

The position errors are appeared in Figure 11, we note that the SVSF-SLAM algorithm performs altogether superior to the EKF-SLAM algorithm in terms of estimation error. The SVSF is in a position to get the best of information in a short time and come up with a relatively estimate.

Figure 12 puts in view the evaluation results of an investigation in comparison to the convergence RMSE of both algorithms. As for the results of the second experiment in comparison to the RMSE of EKF/SVSF-SLAM algorithms, in appearance of colored noise, we note that the RMSE showed the best outcomes with colored noise as can be seen from Figure 12. SVSF-SLAM algorithm is more accurate than EKF-SLAM algorithm in terms of position.

The structure of SVSF-SLAM algorithm reveals that it reduces the computational complexity of the algorithm. One inconvenient of the EKF-SLAM algorithm is the use of the full dimension of the state vector to update the pose and the map estimation. In this way, the computation time will increment with the size of the state vector. On the other hand, the SVSF-SLAM algorithm uses just the posture 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 SVSF-SLAM algorithm remains nearly constant per iteration depending on the number of visible landmarks (Figure 13).

Experimental results. Figure 14 shows the visited places (three (03) rooms (see (a), (b), and (c)) + corridor (see (d)) of the building (indoor environment). We made a trip over a distance of 100 m with a speed of 0.5 m/s.

When the robot moves in the map, it uses a laser to detect landmarks such as points and lines. In this experiment, the maximum range of the SICK laser is 15 m. Each scan provides 361 points which are represented in polar coordinates. They are spread over a range from 0° to 180°. The experimental used data are collected from experimented locations. This

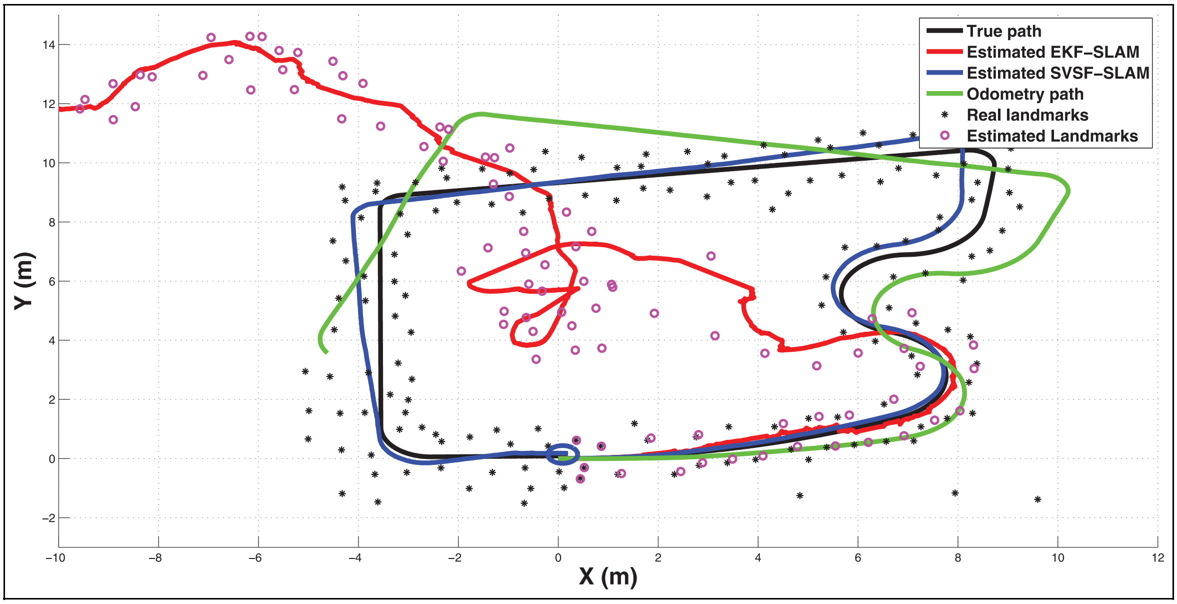

Figure 10. Pose estimation with colored noise.

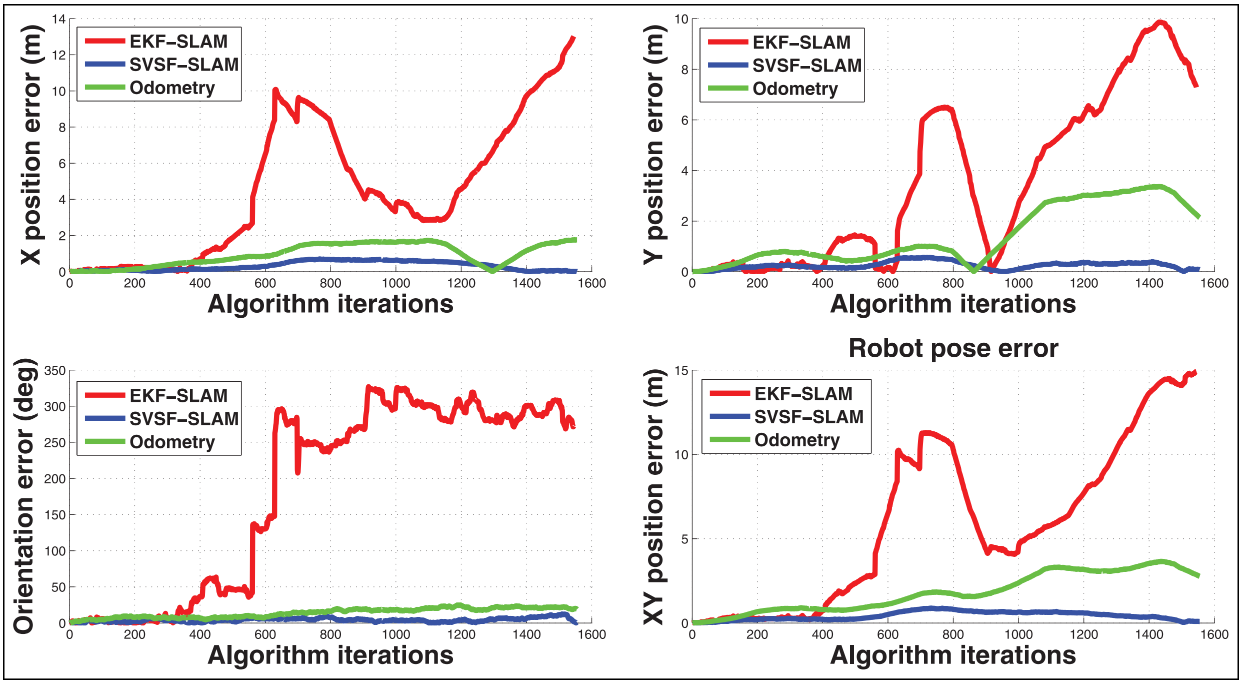

Figure 11. Error position of the robot with colored noise.

environment is used for the algorithm application SLAM UGV with metric maps.

Trajectory estimated by the SVSF-SLAM algorithm, as shown in Figure 15, is more accurate in comparison to the EKF-SLAM algorithm. The SVSF-SLAM algorithm provides the best position in comparison to the EKF-SLAM algorithm. The SVSF-SLAM algorithm shows that our results are consistent and realistic in comparison to the EKF-SLAM algorithm. We see clearly as shown in Figure 15(a) that the pose errors of the EKF-SLAM algorithm increase significantly and the robot is so far and does not arrive to the initial position because of the previous errors made through its trajectory. Moreover, from Figure 15(b), at the final

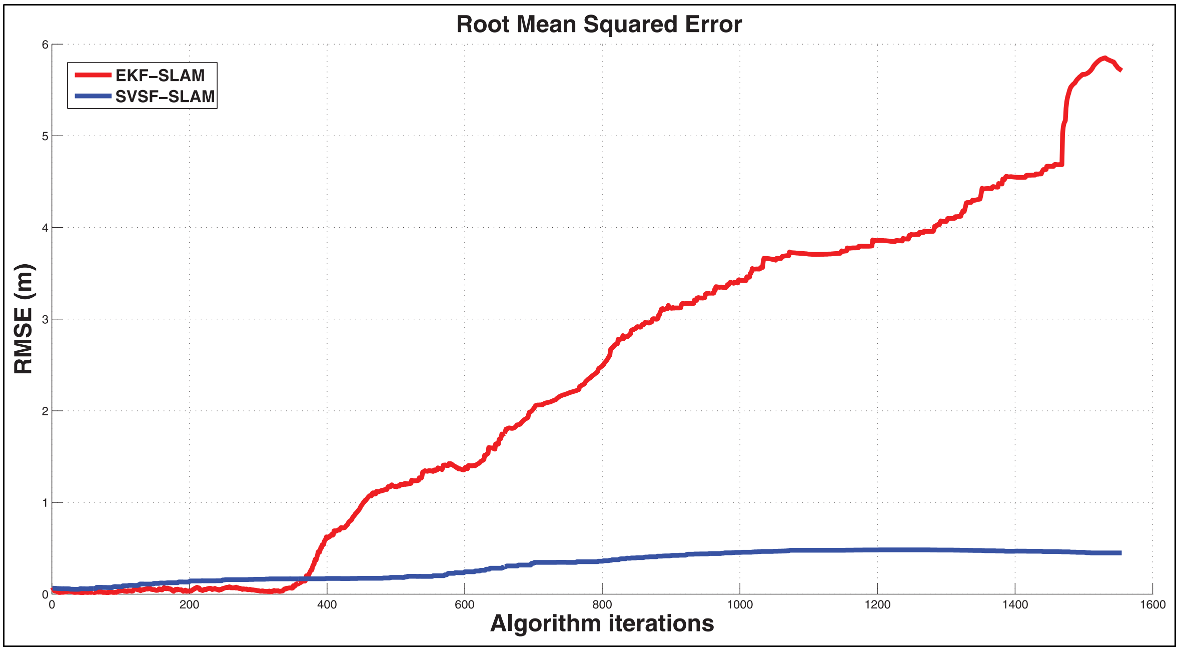

Figure 12. Position RMSE under colored noise.

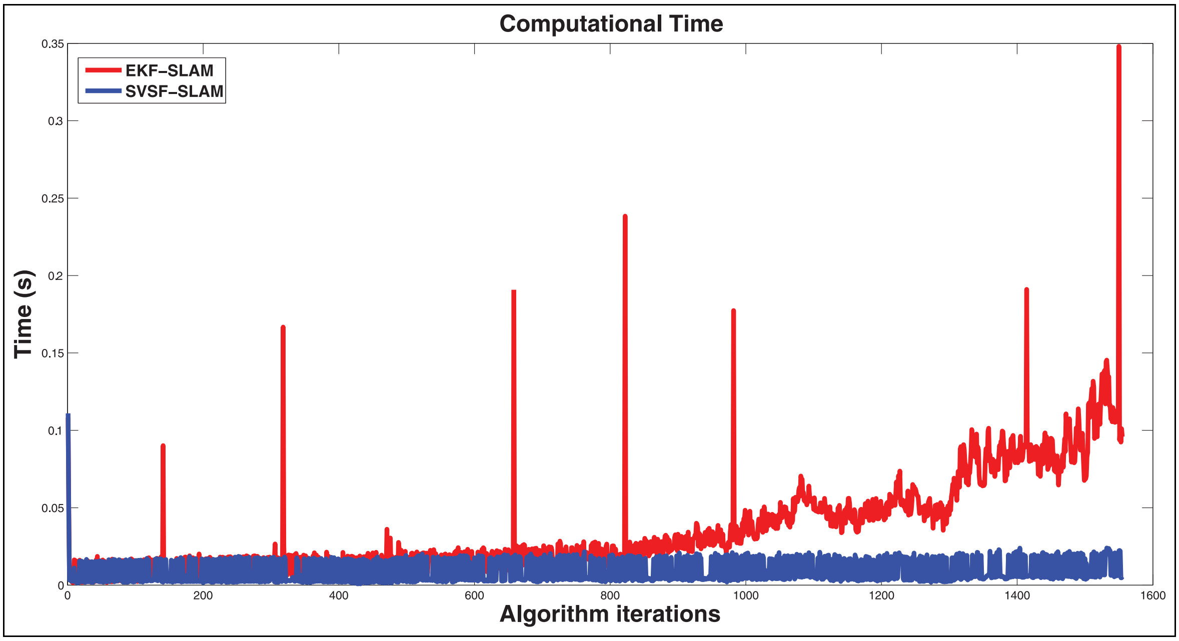

Figure 13. Computational time comparison.

time, we can remark that the robot detects a common feature which improves the consistency of the SVSF-SLAM as well as the EKF-SLAM algorithm.

From the starting position to the final position (loop closure at $(0, 0)$), the SVSF-SLAM algorithm converges into a very accurate solution in comparison to the EKF-SLAM algorithm either with feature association–based point or with feature association–based line.

The structure of the SLAM algorithm–based line has been shown to decrease the computational time in comparison to the SLAM algorithm–based point which is proportional to the state vector length. EKF-SLAM algorithm allocates a large memory space and requires heavy significant computational time (see Figure 16).

In this environment, we note that the path given by the odometer is completely drift and this is due to

Figure 14. Shots of visited places.

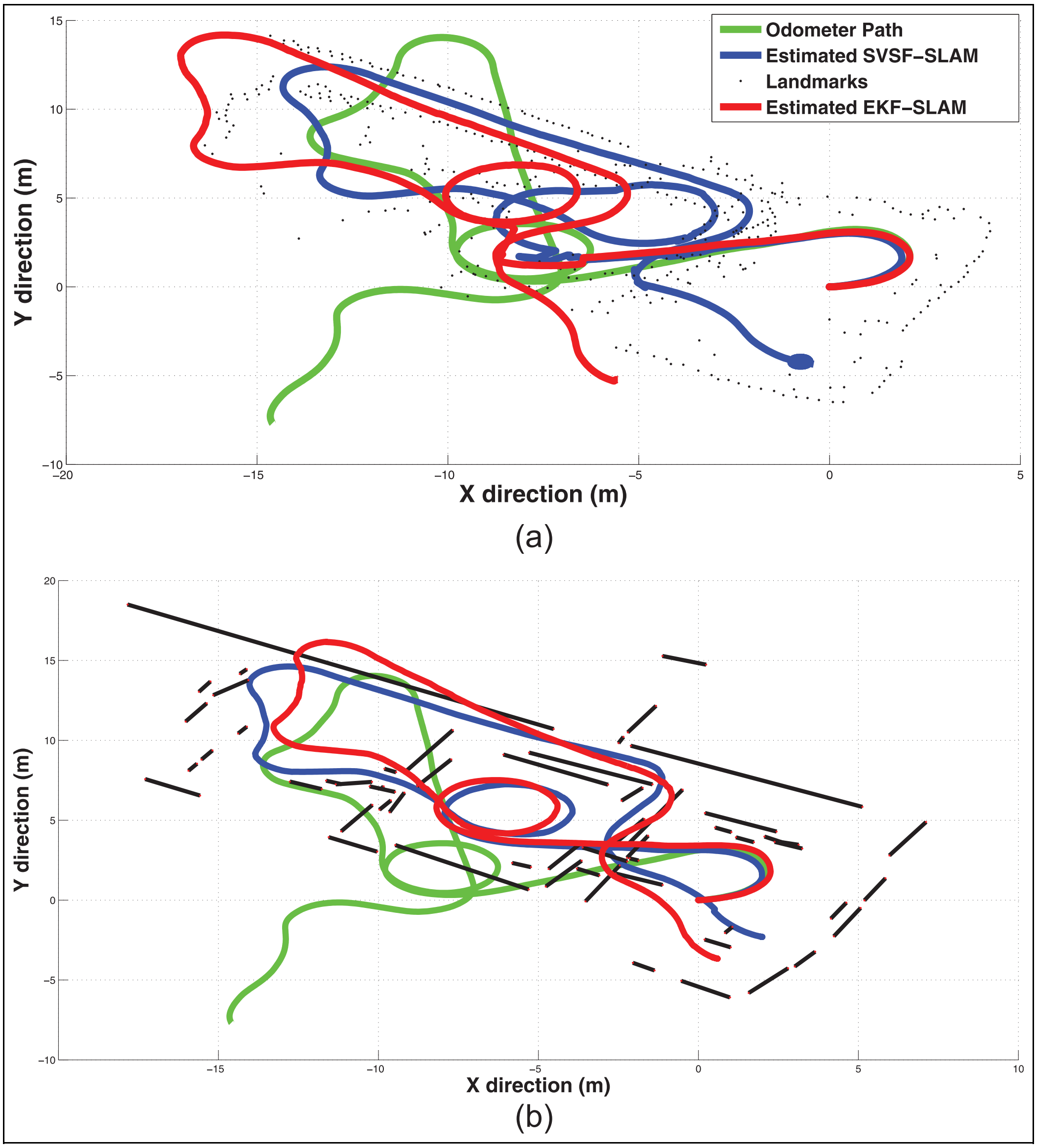

Figure 15. Result of SVSF/EKF-SLAM algorithms based on point and line approaches in real time.

the great range done by the robot. Besides, the SVSF-SLAM algorithm is better than EKF-SLAM algorithm. According to these results, we can confirm how a better localization of the robot makes the possibility of a coherent environment map building (see Figure 15).

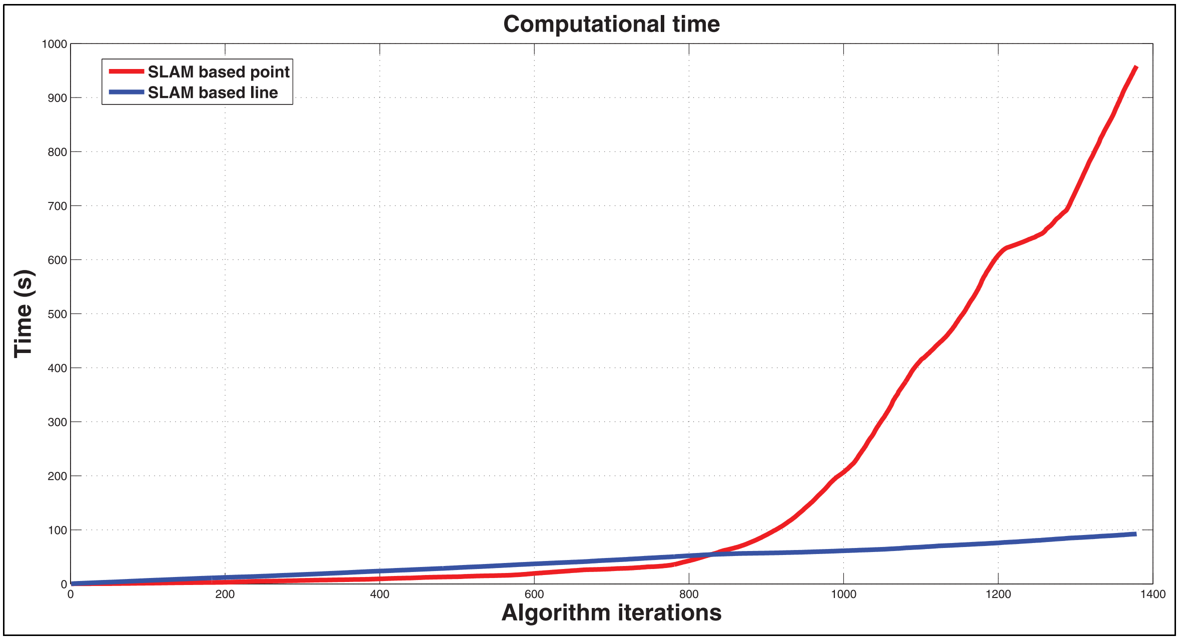

Figure 16. Computational time of EKF/SVSF-SLAM algorithms based on point and line approaches in real time.

Table 1. True and estimated positions for all different trajectories (EKF/SVSF-SLAM and odometer).

True position	Estimated position $(x_r(m), y_r(m), \theta_r(\text{rad}))$		
(x_r, y_r, θ_r)	SVSF-SLAM	EKF-SLAM	Odometer
(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)
V = 0 (m/s), w = 0 (rad/s), t = 0 (s)			
(4.90;0.70;0.29)	(4.87;0.85;0.30)	(4.84;0.75;0.32)	(4.92;0.67;0.27)
V = 0.35 (m/s), w = 0.02 (rad/s), t = 14.2 (s)			
(7.20;1.61;0.70)	(7.16;1.81;0.70)	(7.04;1.98;0.84)	(7.30;1.50;0.59)
V = 0.2 (m/s), w = 0.13 (rad/s), t = 22.3 (s)			
(7.77;2.80;1.58)	(7.77;3.03;1.571)	(7.43;3.22;1.77)	(8.02;2.61;1.42)
V = 0.2 (m/s), w = 0.13 (rad/s), t = 29.1 (s)			
(6.13;4.54;2.43)	(6.19;4.77;2.37)	(5.43;4.57;2.64)	(6.72;4.69;2.21)
V = 0.2 (m/s), w = −0.15 (rad/s), t = 41.9 (s)			
(5.80;6.16;1.10)	(5.93;6.33;1.05)	(4.73;6.01;1.38)	(6.80;6.32;0.81)
V = 0.2 (m/s), w = −0.15 (rad/s), t = 50.8 (s)			
(8.52;8.27;1.42)	(8.67;8.41;1.39)	(7.15;8.62;1.59)	(10.06;7.47;1.03)
V = 0.3 (m/s), w = 0.01 (rad/s), t = 65.1 (s)			
(8.53;10.33;2.52)	(8.75;10.39;2.48)	(6.64;10.73;2.69)	(10.97;9.34;2.00)
V = 0.1 (m/s), w = 0.2 (rad/s), t = 75.7 (s)			
(−3.35;8.82;3.75)	(−3.17;9.28;3.71)	(−4.73;6.52;3.99)	(−0.50;13.42;3.24)
V = 0.1 (m/s), w = 0.2 (rad/s), t = 110.4 (s)			
(−3.55;2.36;4.72)	(−3.92;2.73;4.62)	(−4.28;0.37;4.76)	(−3.07;7.58;4.42)
V = 0.3 (m/s), w = 0.2 (rad/s), t = 135.4 (s)			
(−3.36;0.47;5.36)	(−3.85;0.74;5.28)	(−3.86; −1.53;5.41)	(−3.44;5.71;5.05)
V = 0.2 (m/s), w = 0.2 (rad/s), t = 142.9 (s)			
(0.08;0.09;6.30)	(−0.38;0.26;6.25)	(0.09;0.02;6.42)	(−0.19;4.28;6.02)
V = 0.2 (m/s), w = 0.2 (rad/s), t = 155.1 (s)			

Optimal control results

The path planning with optimal control. We have implemented the methods (Shooting and Bellman) using EKF/SVSF-SLAM localization. Table 1 depicts the results of the two methods we used to solve the SLAM localization. Numerical solution provides the following results with the Shooting and Bellman methods. To improve the position accuracy of the numerical solution of optimal control problems, we use the Bellman method compared to the Shooting method which has low accuracy and decreases the convergence of the robot position. We compare the true positions (Table 1) with the results given by the Shooting and Bellman methods (Tables 2 and 3), (Tables 4 and 5), and (Tables 6 and 7), respectively.

Table 2. The Shooting method (SVSF-SLAM).

x_r(m)	y_r(m)	θ_r(rad)
0	0	0
0.0104	0.0445	−0.0089
4.9620	1.8240	−0.0412
7.0336	2.1621	0.7743
8.6238	9.9804	−0.5499
6.5188	7.9142	2.2464
5.4368	12.5223	0.2323
7.7459	14.9626	0.0502
8.3090	21.4742	3.1229
−2.3921	16.0587	2.8101
−3.4041	2.4282	4.5972
−2.2606	1.8989	5.3109
−0.3574	0.7865	6.1960

Table 3. The Bellman method (SVSF-SLAM).

x_r(m)	y_r(m)	θ_r(rad)
0	0	0
0.2347	−0.0126	14.0000
4.8701	0.8434	0.3049
7.1620	1.8106	0.7046
7.7710	3.0310	1.5713
6.1909	4.7703	2.3716
5.9325	6.3291	1.0534
8.6694	8.4055	1.3912
8.7541	10.3931	2.4790
−3.1722	9.2848	3.7126
−3.9216	2.7342	4.6181
−3.8461	0.7447	5.2782
−0.3823	0.2597	6.2493

Table 4. The Shooting method (EKF-SLAM).

x_r(m)	y_r(m)	θ_r(rad)
0	0	0
0.0107	0.0358	−0.0094
5.0729	1.8867	0.0305
7.3223	1.6664	0.7740
9.4357	9.5588	2.8671
5.4607	6.0168	2.8377
3.9588	15.2038	15.6792
7.3412	14.7279	2.2658
6.7042	21.6882	3.2895
−3.4118	17.6088	2.3423
−3.3100	−0.0507	4.7511
0.5717	1.2401	5.2740
0.1316	0.8493	6.0910

Table 5. The Bellman method (EKF-SLAM).

x_r(m)	y_r(m)	θ_r(rad)
0	0	0
−0.0241	0.1255	14.0000
4.8487	0.7518	0.3204
7.0405	1.9790	0.8427
7.4374	3.2203	1.7680
5.4386	4.5743	2.6401
4.7271	6.0134	1.3792
7.1512	8.6169	1.5855
6.6441	10.7262	2.6917
−4.7272	6.5180	3.9861
−4.2812	0.3738	4.7561
−3.8639	−1.5264	5.4088
0.0871	0.0242	6.4186

Table 6. The Shooting method (odometer).

x_r(m)	y_r(m)	θ_r(rad)
0	0	0
0.0114	0.0412	−0.0094
5.0751	1.4653	0.0511
7.8570	0.6748	0.5899
9.9409	10.5034	0.4596
7.1834	7.8435	2.5400
7.4595	11.9491	−0.0820
9.3109	12.3822	−0.2362
10.5109	24.7952	3.1537
1.8898	28.5927	−15.8027
−2.3851	7.3407	4.6600
−2.3799	6.0224	5.2756
0.0734	5.2376	5.9845

Table 7. The Bellman method (odometer).

x_r(m)	y_r(m)	θ_r(rad)
0	0	0
0.5441	0.1360	14.0000
4.9155	0.6565	0.2612
7.3070	1.4974	0.5892
8.0209	2.6101	1.4267
6.7214	4.6862	2.2099
6.8020	6.3225	0.8069
10.0647	7.4671	1.0277
10.9708	9.3438	2.0048
−0.5026	13.4171	3.2365
−3.0654	7.5815	4.4208
−3.4404	5.7083	5.0539
−0.1935	4.2843	6.0231

Table 8. Final time with the Shooting and Bellman methods for three methods (EKF-SLAM, SVSF-SLAM, and odometer).

Method	Final time (s)	
	Shooting method	Bellman method
SVSF-SLAM	158.6	156.5
EKF-SLAM	161.7	158
Odometer	163	162.4

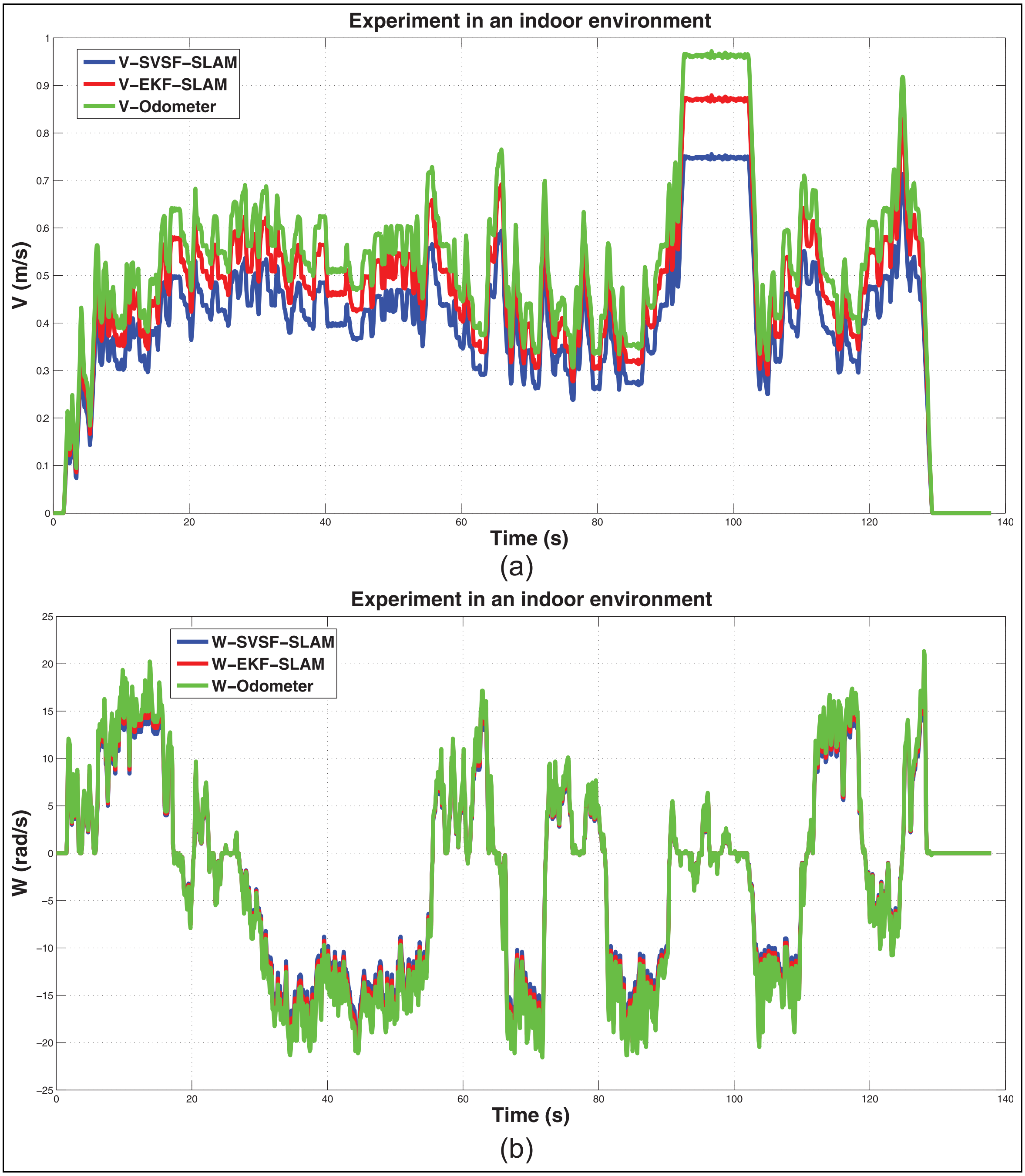

Figure 17. Control of V and W in an indoor environment–based point (SVSF-SLAM/EKF-SLAM/odometer).

Table 9. Final time with the Shooting and Bellman methods for three methods (EKF-SLAM, SVSF-SLAM, and odometer).

Shooting method (EKF-SLAM)	Bellman method (EKF-SLAM)	Shooting method (SVSF-SLAM)	Bellman method (SVSF-SLAM).
327.5330 s	327.5330 s	111.3860 s	315.8968 s

Through the two methods, best results are obtained by SVSF-SLAM compared to EKF-SLAM and the odometer; moreover, short time is required by the proposed approach (see Table 8). We can say that the result sets using the Bellman method are much better than the Shooting method. The Shooting method is not

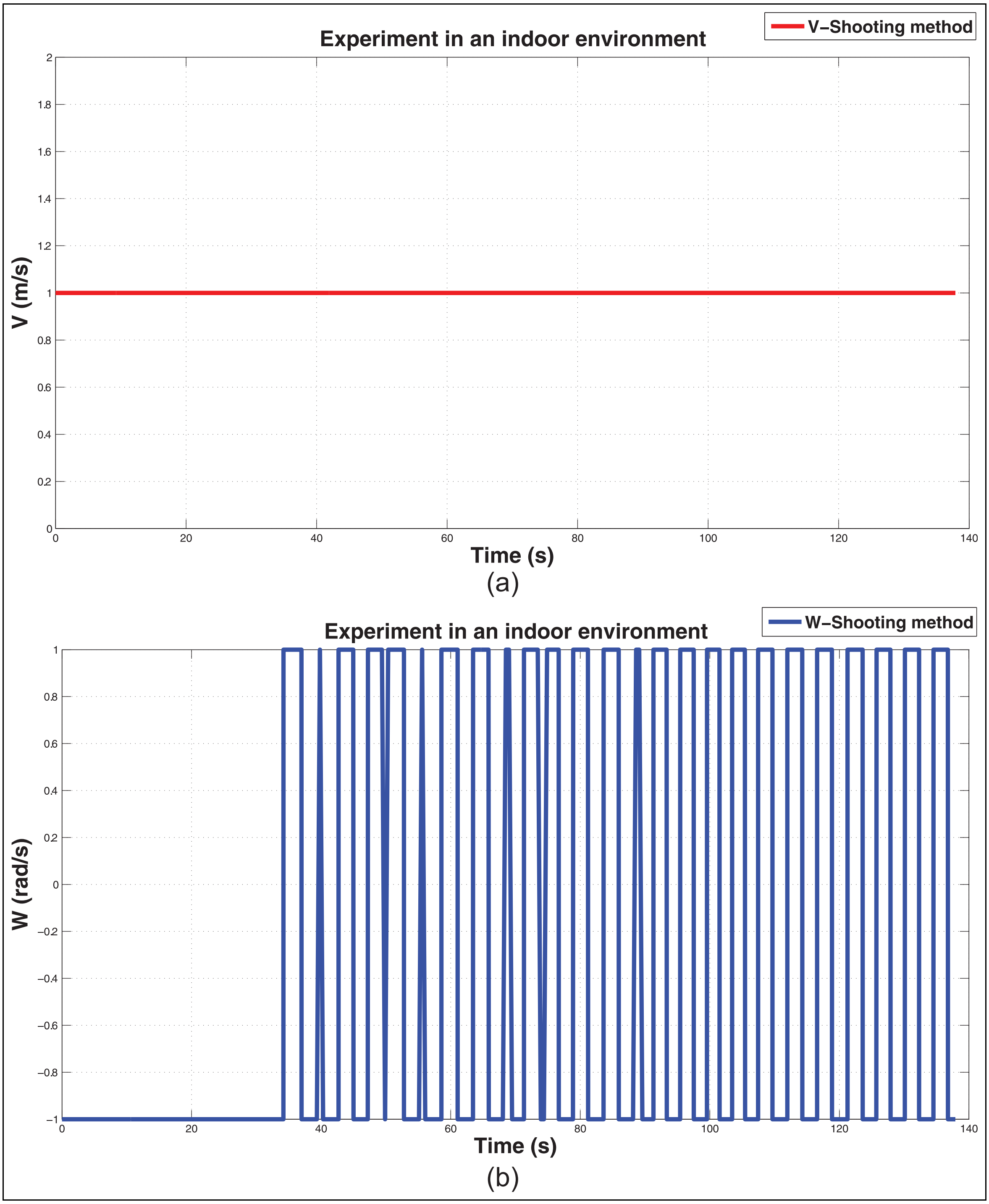

Figure 18. Control of V and W with the Shooting method in an indoor environment–based point (SVSF-SLAM).

accurate compared to the Bellman method, because the initial conditions of adjoint state are sensible, that make us away from the correct positions. The solution for SLAM problem and the accuracy of the result are controlled by the accuracy of the Bellman method used as opposed to the Shooting method with the use of the Euler–Lagrange equations. The following figures show the results of optimal control with the Shooting and Bellman methods.

The final time is the minimum time to come to the last time point corresponding at 155.1 s (final time) to the final position (0,0,0). From Table 9, note that the final time found by the Bellman method is better than the Shooting method. As can be seen from Table 8, the

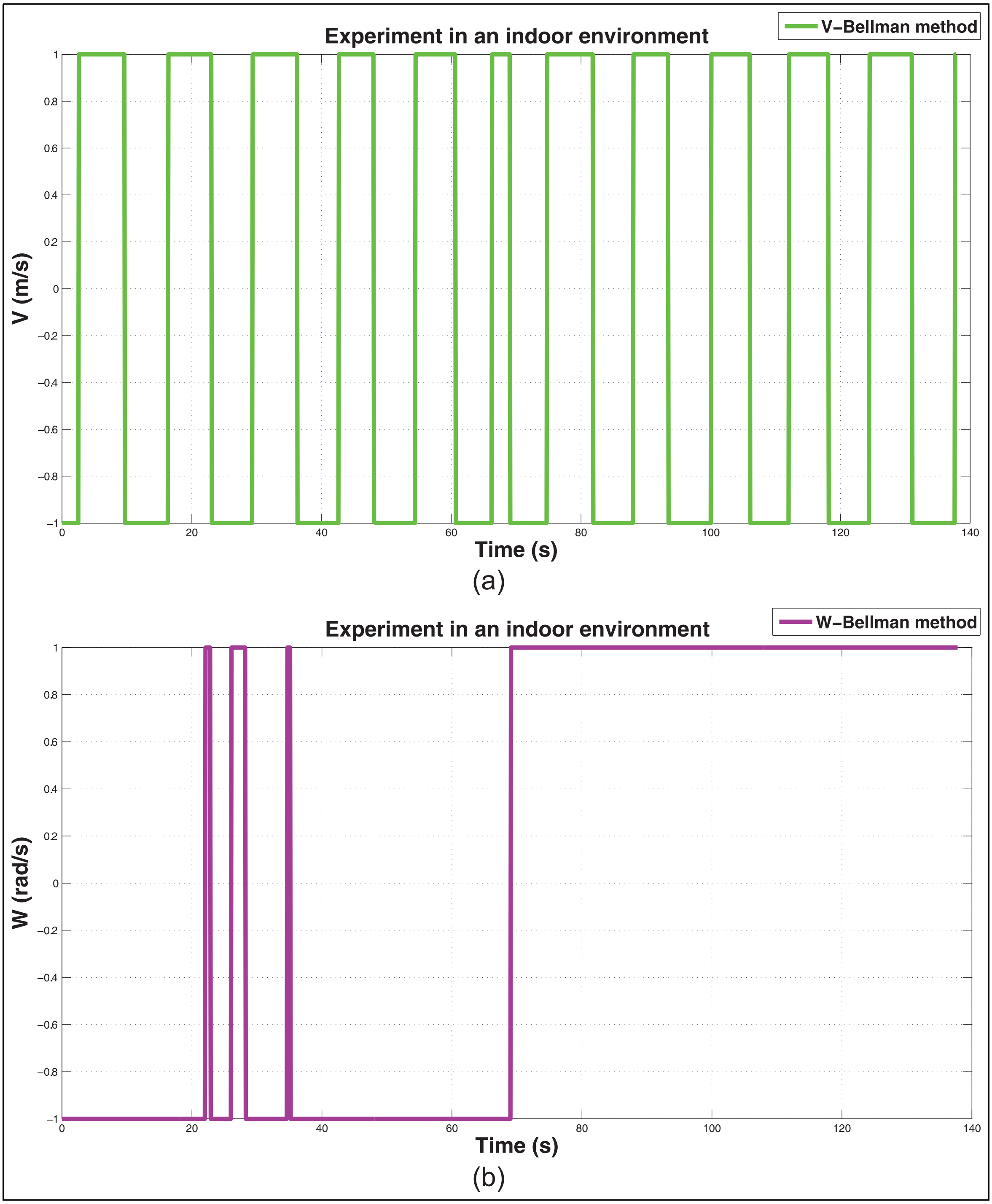

Figure 19. Control of V and W with the Bellman method in an indoor environment–based point (SVSF-SLAM).

minimal final time is given with SVSF-SLAM localization. Then, we compared the true positions of our experiments for different positions of localization such as SVSF, EKF, and odometer. We determined the robot controls for each localization of two methods and we made a comparison between them. We notice that the robot control parameters using the Bellman method is better than Shooting method that the SVSF-SLAM localization is used. The results conform with our data (see Figures 17–19).

From Figures 20 and 21, we noted that the results found using the Bellman principle are much better than the Shooting method, and the points (x_r, y_r) given by SVSF-SLAM are very close to the true points of the

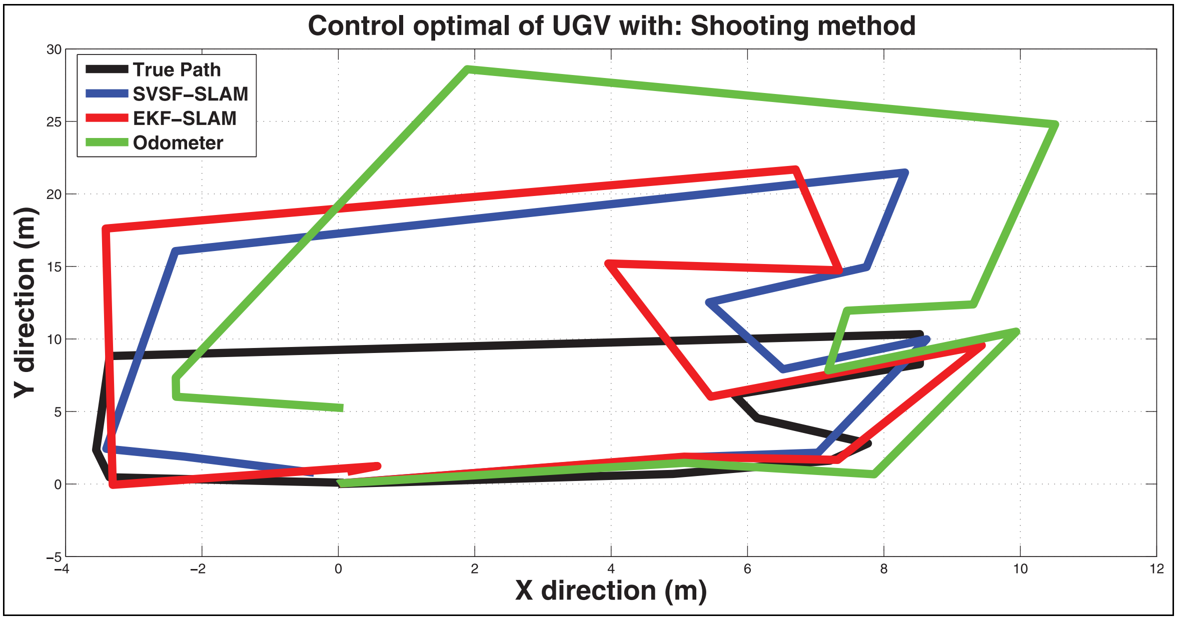

Figure 20. Control optimal of UGV with the Shooting method: SVSF-SLAM, EKF-SLAM, and odometer.

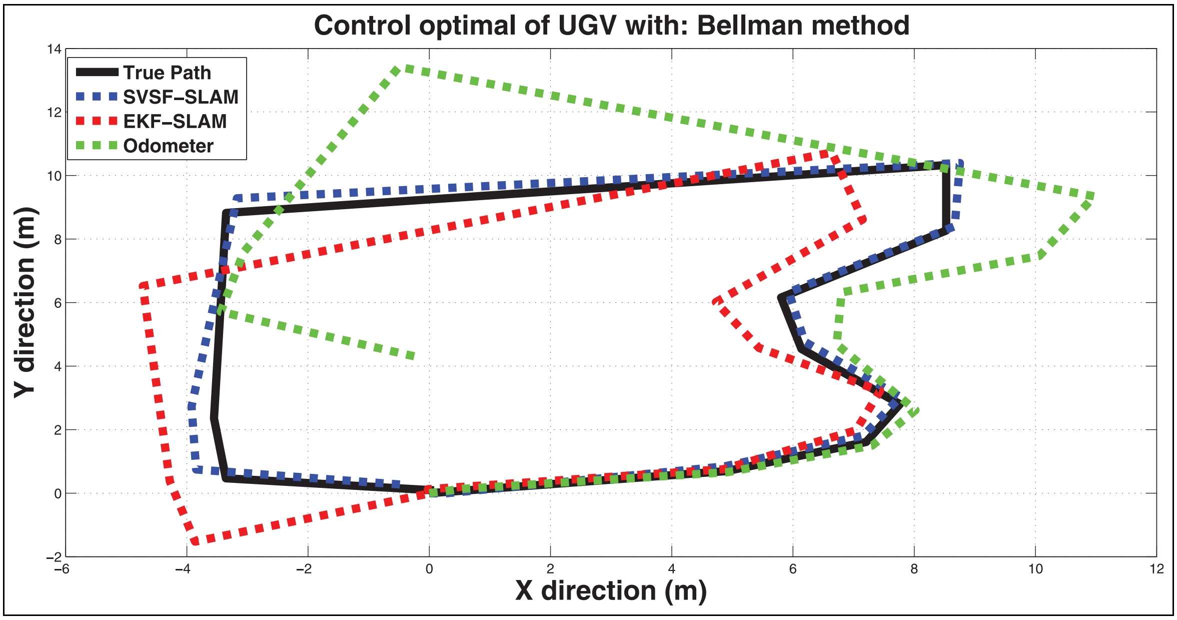

Figure 21. Control optimal of UGV with the Bellman method: SVSF-SLAM, EKF-SLAM, and odometer.

polygons. Consequently, SVSF-SLAM gives good results with the Bellman method (see Figure 22). For our experiment where the robot begins on the path, the trajectories produced by EKF-SLAM, SVSF-SLAM, and odometer were the same. Then on the contrary, the trajectories of EKF-SLAM and odometer bring the robot on the wrong path. Figures 23–28 show the control of V and W with the Shooting and Bellman methods of EKF-SLAM, SVSF-SLAM, and odometer.

Table 9 shows the energy of the robot orientation $E = (\sum \theta_i^2)$ with different approaches. Smoothest trajectories present less energy (less maneuverability). As shown in this table, SVSF-SLAM presents less energy in both methods which confirm the obtained results.

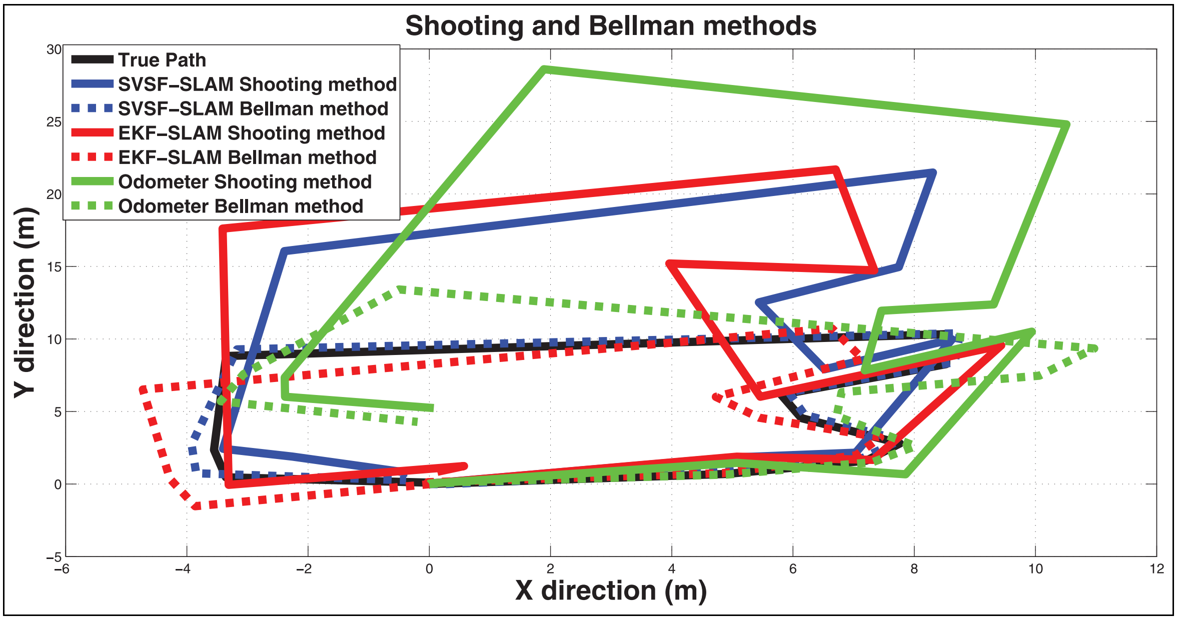

Figure 22. Comparison of the Shooting and Bellman methods.

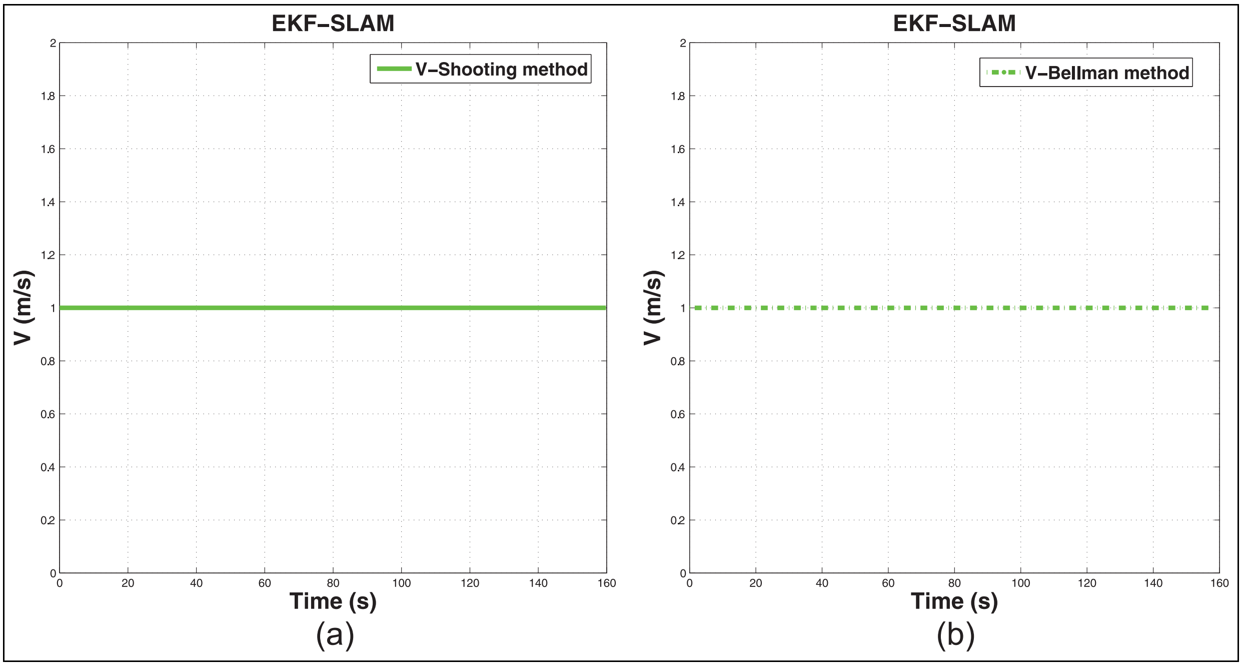

Figure 23. Control of V with the Shooting and Bellman methods (EKF-SLAM).

Conclusion

In this article, SLAMPP algorithm have been proposed for UGV navigation. As this UGV operates at low speed, path planning and optimization strategies are developed without considering vehicle dynamics. Only the kinematics of the vehicle and the environment constraints are considered.

Based on SVSF-SLAM algorithm localization and optimal control theory, the proposed algorithm (SLAMPP) improves the path in terms of continuity by

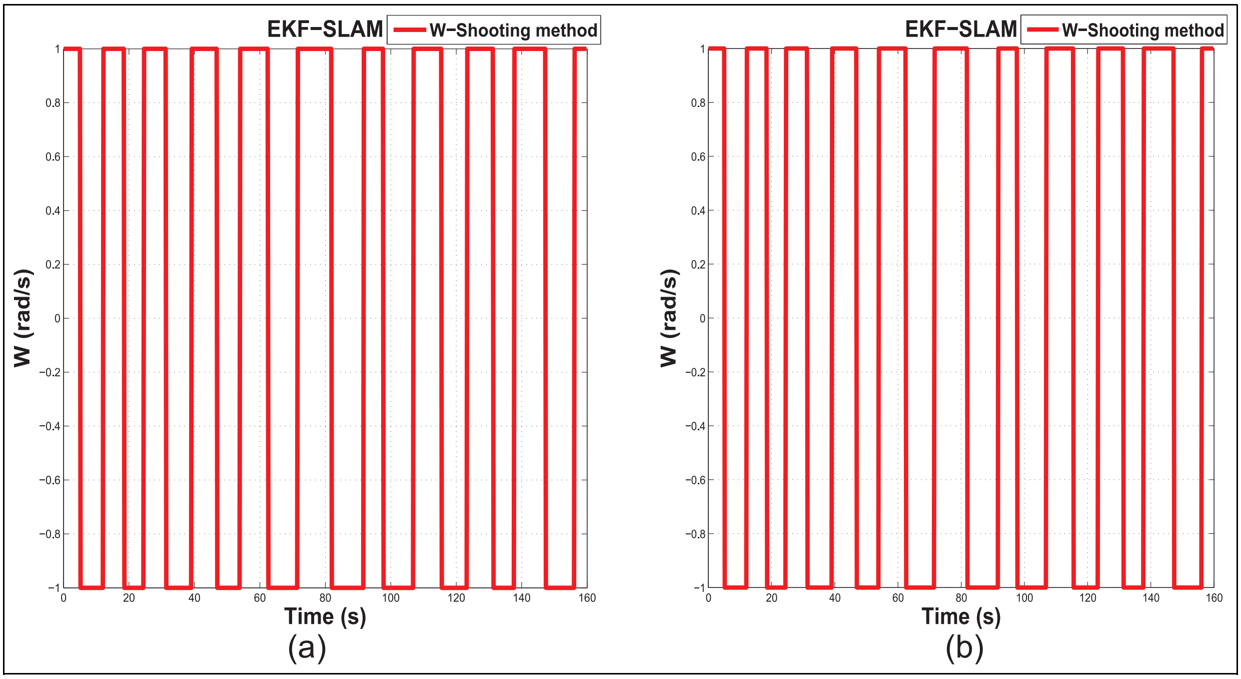

Figure 24. Control of W with the Shooting and Bellman methods (EKF-SLAM).

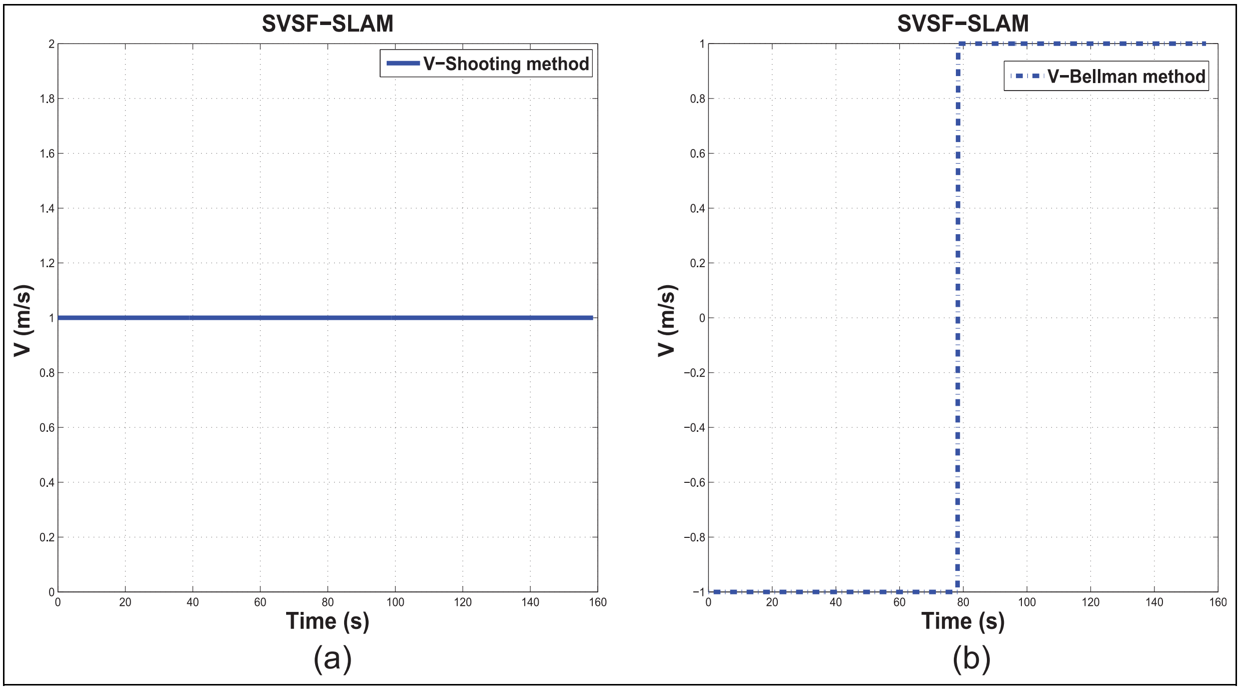

Figure 25. Control of V with the Shooting and Bellman methods (SVSF-SLAM).

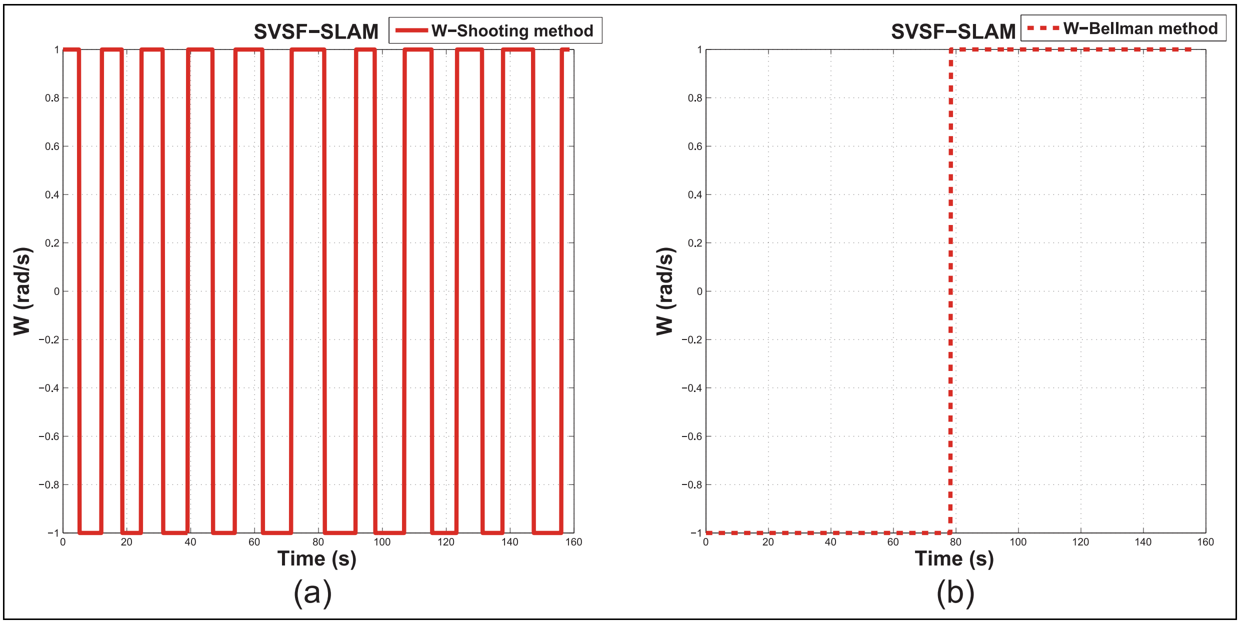

Figure 26. Control of W with the Shooting and Bellman methods (SVSF-SLAM).

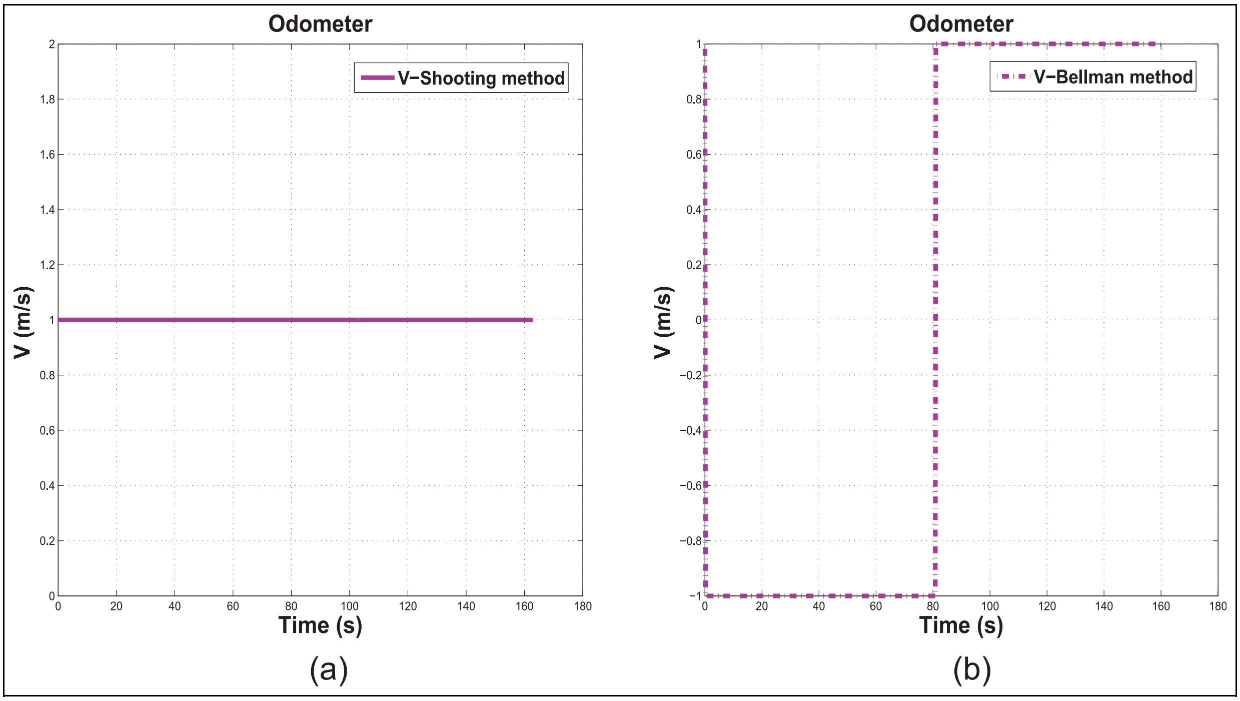

Figure 27. Control of V with the Shooting and Bellman methods (odometer).

providing a smoother and more comfortable trajectory tracking. In addition, the proposed algorithm requires less computation time to achieve the improvement. The proposed approach is validated using simulation and experimental data. Several experiments are tested under realistic conditions and good performances are obtained by the SVSF-SLAM compared to the EKF-SLAM algorithm. As perspectives, we propose to extend the proposed approach for multiple robots (UGVs, UAVs, and AUVs).

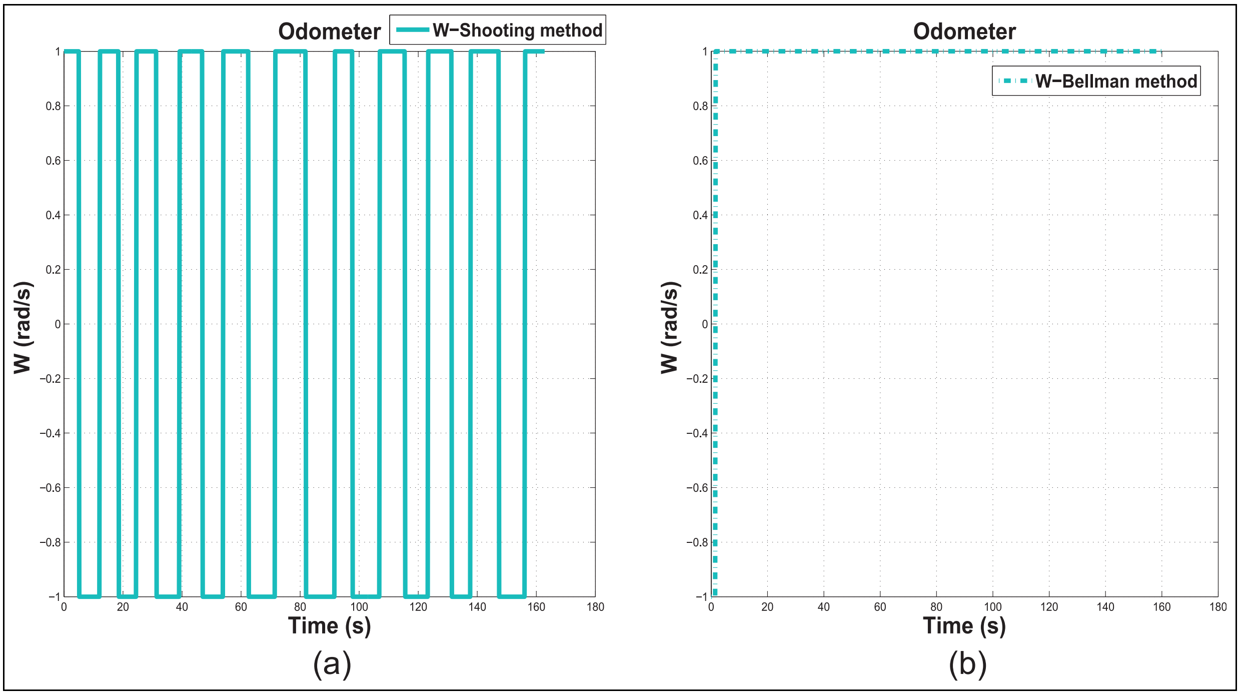

Figure 28. Control of W with the Shooting and Bellman methods (odometer).

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

References

1. Golombek MP, Cook RA, Economou T, et al. Overview of the mars pathfinder mission and assessment of landing site predictions. *Science* 1997; 278: 1743–1748.
2. Thrun S, Ferguson D, Haehnel D, et al. A system for volumetric robotic mapping of abandoned mines. In: *Proceedings of the 2003 IEEE international conference on robotics and automation*, Taipei, Taiwan, 14–19 September 2003. New York: IEEE.
3. Moutarlier P and Chatila R. Stochastic multisensory data fusion for mobile robot location and environment modeling. In: *Proceedings of the 5th international symposium of robotics research*, Tokyo, Japan, 28–31 August 1989. MIT Press Cambridge.
4. Leonard JJ, Newman PM, Rikoski RJ, et al. Towards robust data association and feature modelling for concurrent mapping and localization. In: Jarvis RA and Zelinsky A (eds) *Robotics research*, vol. 6 (Springer tracts in advanced robotics). Berlin; Heidelberg: Springer, 2003, pp.7–20.
5. Williams SB, Dissanayake G and Durrant-Whyte H. Constrained initialization of the simultaneous localization and mapping algorithm. In: *Proceedings of the 3rd international conference on field and service robotics*, Espoo, 10–13 June 2001. Helsinki, Finland: Yleissjhaljennhos-Painophorssi.
6. Guivant JE and Nebot EM. Optimization of the simultaneous localization and map-building algorithm for real time implementation. *IEEE T Robotic Autom* 2001; 17: 242–257.
7. Durrant-Whyte H and Bailey T. Simultaneous localisation and mapping: part I. *IEEE Robot Autom Mag* 2006; 13: 99–110.
8. Jensfelt P, Kragic D, Folkesson J, et al. A framework for vision based bearing only 3D SLAM. In: *Proceedings of the IEEE international conference on robotics and automation*, Orlando, FL, 15–19 May 2006. New York: IEEE.
9. Davison AJ. Real-time simultaneous localisation and mapping with a single camera. In: *Proceedings of the 9th IEEE international conference on computer vision*, Nice, 13–16 October 2003. Washington, DC: IEEE.
10. Montemerlo M, Thrun S, Koller D, et al. FastSLAM 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In: *Proceedings of the 6th international joint conference on artificial intelligence*, Acapulco, Mexico, 9–15 August 2003. San Francisco, CA: Morgan Kaufmann Publishers.
11. Montemerlo M, Thrun S, Koller D, et al. FastSLAM: a factored solution to the simultaneous localization and mapping problem. In: *Proceedings of the AAAI 18th national conference on artificial intelligence*, Edmonton, AB, Canada, 28 July–1 August 2002. Menlo Park, CA: American Association for Artificial Intelligence.
12. Doucet A, Freitas NDE and Gordon N. *Sequential Monte Carlo methods in practice* (Statistics for engineering and information science). New York: Springer, 2001.

13. Lemaire T, Berger C, Jung IK, et al. Vision-based SLAM: stereo and monocular approaches. *Int J Comput Vision* 2007; 74: 343–364.
14. Arras KO and Siegwart RY. Feature extraction and scene interpretation for map-based navigation and map building. In: *Proceedings of the SPIE 3210, mobile robotics XII*, Pittsburgh, PA, 25 January 1998. Bellingham, WA: SPIE.
15. Sfeir J. *Navigation D'un Robot Mobile En Environnement Inconnu Utilisant Les Champs De Potentiels Artificiels*. Mémoire de maîtrise électronique, Génie de la production automatisée, École de Technologie Supérieure, Montréal, QC, Canada, 2009.
16. Hull DG. *Optimal control theory for applications* (Mechanical engineering series). New York: Springer-Verlag, 2003.
17. Habibi S, Burton R and Chinniah Y. Estimation using a new variable structure filter. In: *Proceedings of the IEEE American control conference*, Anchorage, AK, 8–10 May 2002. New York: IEEE.
18. Robinett RD III, Eisler GR and Hurtado JE. *Applied dynamic programming for optimization of dynamical systems* (Advances in design and control, Book Code: DC09, xviii-255). Philadelphia, PA: Society for Industrial and Applied Mathematics, 2005.
19. Barshan B and Durrant-Whyte HF. Inertial navigation systems for mobile robots. *IEEE T Robotic Autom* 1995; 11: 328–342.
20. Demim F, Nemra A and Louadj K. Robust SVSF-SLAM for unmanned vehicle in unknown environment. *IFAC PapersOnLine* 2016; 49: 386–394.
21. Demim F, Nemra A, Louadj K, et al. A new approach to improve the success and solving the UGVs Cooperation for SLAM Problem, using a SVSF Filter. In: *Proceedings of the Mediterranean conference on pattern recognition and artificial intelligence*, Tebessa, Algeria, 22–23 November 2016, pp.56–63. New York: ACM.
22. Demim F, Nemra A, Louadj K, et al. Simultaneous localization and mapping algorithm for unmanned ground vehicle with SVSF filter. In: *Proceedings of the IEEE 8th international conference on modelling, identification and control*, Algiers, Algeria, 15–17 November 2016. New York: IEEE.
23. Demim F, Boucheloukh A, Nemra A, et al. A new adaptive smooth variable structure filter SLAM algorithm for unmanned vehicle. In: *Proceedings of the 6th international conference on systems and control*, Batna, Algeria, 7–9 May 2017. New York: IEEE.
24. Gutmanand JS and Schlegel C. AMOS: comparison of scan matching approaches for self-localization in indoor environments. In: *Proceedings of the 1st Euromicro workshop on advanced mobile robot*, Kaiserslautern, 9–11 October 1996. New York: IEEE.
25. Yaqub T, Tordon MJ and Katubitiya J. Line segment based scan matching for concurrent mapping and localization of a mobile robot. In: *Proceedings of the 9th international conference on control, automation, robotics and vision*, Singapore, 5–8 December 2006. New York: IEEE.
26. Diosi A and Kleeman L. Laser scan matching in polar coordinates with application to SLAM. In: *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS 2005)*, Edmonton, AB, Canada, 2–6 August 2005. New York: IEEE.
27. Dissanayake G, Newman PM, Durrant-Whyte HF, et al. An experimental and theoretical investigation into simultaneous localisation and map building. In: Corke PI and Trevelyan J (eds) *Experimental robotics VI*, vol. 250. 2nd ed. London: Springer, 2000, pp.265–274.
28. Orderud F. Comparison of Kalman filter estimation approaches for state space models with nonlinear measurements. In: *Proceedings of the Scandinavian conference on simulation and modeling*, vol. 7491, Trondheim, 2005, pp. 7–9.
29. Grewal MS and Andrews AP. *Kalman filtering: theory and practice using MATLAB*. 2nd ed. New York: John Wiley & Sons, 1993.
30. Habibi SR and Burton R. The variable structure filter. *J Dyn Syst: T ASME* 2003; 125: 287–293.
31. Habibi SR. The smooth variable structure filter. *P IEEE* 2007; 95: 1026–1059.
32. Al-Shabi M. *The general Toeplitz/observability SVSF*. PhD Thesis, Department of Mechanical Engineering, McMaster University, Hamilton, ON, Canada, 2011.
33. Gadsden SA, Dunne D, Habibi SR, et al. Comparison of extended and unscented Kalman, particle, and smooth variable structure filters on a bearing-only target tracking problem. In: *Proceedings of the SPIE 7445, signal and data processing of small targets*, San Diego, CA, 3 September 2009, paper no. 74450B-74451. Bellingham, WA: SPIE.
34. Trelat E. *Contrôle Optimal: Théorie et Applications*. Paris: Vuibert, Collection Mathématiques Concrètes, 2005.
35. Bellman RE and Dreyfus SE. *Applied dynamic programming*. Santa Monica, CA: The Rand Corporation, 1962.
36. Mauch H. *Dynamic programming*. Berlin; Heidelberg: Springer, 2007.
37. Dreyfus E. Some types of optimal control of stochastic systems. *J Soc Ind Appl Math* 1964; 2: 120–134.