The CSI Journal on
Computer Science and Engineering
Vol. 19, No. 2, 2025
Pages 52-65
doi.org/10.22034/jcse.2025.181976
Regular Paper

A Novel approach for the Navigation Issues using Soft Computing

Amir Panah[1], Homayun Motameni[2], Ali Ebrahimnejad[3]

[1] Department of Computer Engineering, Babol Branch, Islamic Azad University, Babol, Iran
[2] Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran
[3] Department of Mathematics, Qaemshahr Branch, Islamic Azad University, Qaemshahr, Iran

Abstract

The use of robots that accomplish varied procedures in numerous applications in an area are thriving nowadays. robots can search the area, create a fit map, and localize itself into this map, via interpreting the area autonomously. An approach for simultaneous localization and mapping (SLAM) for robots is the UFastSLAM. The accurate a previous knowledge of the control and measurement noise covariance matrices are needed to the correctness and efficiency of the estimation of the UFastSLAM depend often on. Also, inexplicit past data may seriously go down their efficiency. One of the significant causes of missing particle multifold is sample insignificant in the UFastSLAM. This paper presents a novel heuristic method to solve these issues called Hybrid filter SLAM (Hf SLAM). In the proposed method, for tuning the measurement noise covariance matrices for increase of consistency and correctness are exploited in the Intuitionistic Fuzzy Logic System (IFLS). Moreover, we exploit to improve the efficiency of importance weight from Cuckoo Search Optimization (CSO). The Hf SLAM is more efficient than the UFastSLAM and FastSLAM that has fewer computations and good performance for the larger area based on the results of the simulation and experiment demonstrated.
Keywords: Robots. SLAM. IFLS. CSO. Hybrid filter

1. introduction

One of the significant issues for robots as the robots keeps track of its location by retaining a map of areas and an estimate of its localization is the navigation.
Two significant limitations of the FastSLAM are the approximations of nonlinear procedures and the Jacobian matrices. These can create the filter inconsistently. Another significant challenge is to diminish the number of searches whenever maintaining estimation exactitude. The results in huge areas with a dataset are offered, representing the superiority of the UFastSLAM [1].
A robust controller proposed in attendance of actuator and sensor faults for an actuators helicopter control. In this approach to create a controller used from the neural networks, interval type-2 fuzzy logic control, sliding manner control and exploitation. This avoids rigid modeling, weakening the whisper efficacy of the sliding manner control, reducing rules for the fuzzy controller, and assuring the stability of the system. This method could greatly diminish the whispering efficiency, tracking good in sensor errors and the presence of actuator [2].
Using three existent gene expression datasets, the fuzzy normalization methods were compared with two standard normalizations also a raw gene. The k-nearest-neighbor, support vector machine classifiers, and random forest were applied to identify the exactitude of selected features. The results demonstrate that the genes selected via the FS and IFS normalized datasets give high classification exactitude and the IFS better than the FS normalization [3].
The UFastSLAM will commonly flop to block large loops when concerned with harsh surroundings. The significant innovation of the method is to use the 5th-order conjugate unscented transform, which to detect local features and landmarks and a better the particle filter distribution used to

compute the SLAM transition agglomeration to the 5th term, [4].

For the mobile vehicle pose estimation use a novel approach using particle swarm optimization and the feature location's estimation. The results demonstrate that the estimated and efficiency exactitude of the proposed method is perfect for FastSLAM, and for achieving better consistency utilize form the adaptive Neuro-fuzzy square root central difference Kalman filter for the feature location's estimation. Also, it will decline the number of particles and computational complexity [5].

This approach is a combination of the Rao-Blackwellized particle filter and FastSLAM 2.0 method, and a medium of the local posterior possibility participatory SLAM method combined with a M-posterior distributed estimation approach. Then, was corrected local posterior possibility participatory SLAM time medium of the method pursuant to time discrepancy optimization is proposed pursuant to the exactitude privilege of the early landmarks compared with the next landmarks, that improves optimizes the local estimation

weights and the global estimation exactitude. The method is feasible and efficient [6].

Dual two-dimensional grid maps explain a fuzzy threedimensional grid, and a syntactic preprocess is proposed to accomplish localization via replacement of the weighted average localization approach. The presented method has better properties and fewer calculations than in navigation [7]. An improved significance sampling to improve the efficiency of the FastSLAM is presented via the transformed UKF. The improvement is combined with a novel fuzzy noise estimator, that can regulate observation noises online and the state pursuant to the residual and related covariance, and so reduce the faults caused by the model in exactitude, generally. Totally, the FastSLAM method suffers from the indigence issue so it is basically a particle filter. An adaptive genetic resampling retrieved from genetic optimization is presented to the resampling phase to prevail these defects [8].

The target is to estimate the measurement noise covariance matrix at every step appropriate values. The proposed approach pursuant to a neuro fuzzy to make it least the difference between the innovation consequence covariance matrices and the actual value. The parameters of an adaptive EKF pursuant to a neuro fuzzy is then learned offline via a particle swarm optimization approach. So, more efficient acquisition and the benefits of large scale search areas can be extracted. The results have demonstrated that the proposed approach provides the improvement in both performance efficiency and computational cost [9].

The particle specifications in computing the importance weight and holding a particle conformation. Exclusively calculated weights are avoided to achieve a more accurate posterior distribution considering correlation among particles. Particles with an inappropriate weight are update updated their weights and elected pursuant to an adaptive weight restitution plan. A triangular mesh manufacture that was originally designed vehicles is applied for particle conformation renovation to remove the particle reduction issue that appears in the resampling step [10].

The presented algorithm skips the procedure of updating the landmarks. Significant recuperation on computational performance is thus acquired as a result. They exploited measurements to update the status of particles before updating the landmark estimates. The wrapped computation of all the existent landmark data can be compensated to achieve the realtime design objective [11].

Computation of the movement approach on a route explained via a superficial recognizable curve is an important issue. The hardness occurs in accordance with great-exactitude moving. The method of computation pursuant to the inherent attributes of the area is presented. The method is pursuant to algebraic relations and linguistics of mathematics. Classic computation and novel approaches for movement point on the planar route in order that comparison is shown. For Jordan curves can be achieved in some items using analytical formulas [12].

An efficacious act of robots is necessary to optimize route planning. The robot's navigation in different surroundings populated by a diversity of obstacles has been exploited as a new strategy. This approach is pursuant to brood parasitic action of cuckoos and Lévy flight action. A new target function has been formulated among the robots and obstructions and target, that fulfilled purpose-seeking behavior of robots and the conditions of obstruction avoidance proposed in the area. There is efficaciously guiding the robot tracking from the begin point to the self-assertive goal point with optimum route length and ability of avoiding obstacles. Contingent on the objective function value of every nest, the robots proceed towards the target and avoid obstacles. When the robots reach their target, the optimum route is collected with this method [13].

An unknown area is involved in certain static obstacles and also various mobile obstacles whose amount and position change via the time, randomly. A shortest possible route from an optional starting location into an eligible target point that needs to be smooth and safe, also feasible in an unknown area, is discovered via robot's exploration. To receive this target, a satisfactory set of some constraints involving the smooth shortest and collision free route is needed, simultaneously. Thus, this subject can be an improvement issue, so dissolved via improvement algorithms. In this research, a new method pursuant to CSO for dissolving the issue of exploration in an unknown area via robots is proposed. The simulation results showed efficiency of the proposed method in discovering a short, smooth, collision free route and safety in various unknown areas [14].

The EKF was improved pursuant to the Maximum Likelihood Estimation (MLE) and Expectation-Maximization (EM) Creation. It aims to provide the EKF with capability of its covariance matrices and estimating noise statistics. In addition, EKF was improved and modified to adapt the estimated values given via EM and MLE creation. However, it results in the best solution. So, an addition of Innovation Covariance Estimation was exploited to depreciate this possibility. The proposed approach has better consistency and exactitude compared to the common EKF [15]. Some related methods are expressed in Section 2, and the Hf SLAM is proposed in Section 3. Section 4 demonstrates the simulation and experimental results of the FastSLAM, UFastSLAM and Hf SLAM. Section 5 discussed Concluding.

2. Preliminaries

In this section the basic definitions about the formulation of the proposed method are given.

2.1. Intuitionistic Fuzzy Logic System (IFLS)

In most cases of decisions, appraisal is accomplished by human beings wherever there are limitations of knowledge, subjective functionaries or availability of information because of some uncontrollable parameters. Naturally, every decision maker hesitates more or lower on every selection activity. This is the IFS theory concept explained via Atanassov [16]. The significant gain of the IFS on a fuzzy set is that the IFS separates the rate of membership and non-membership of a component within the set. Also, they get the best tools for optimization issues formulation and, also, the IFLS solution can assure the objective(s) with a bigger degree than the crisp one and the analogous fuzzy optimization issue. The IFS theory is one of the distributions of the fuzzy set theory presented via Zadeh. Due to this generalization, the IF concept has a much wider do significance than the usual standard FS concept in solving various forms of very physical issues [17].

Definition 1. The IFS structure A^* in X is:

$A^* = \{(x, \mu_{A^*}(x), v_{A^*}(x)) : x \in X)\}$, Wherein $\mu_{A^*}(x): X -> 0 \leq \mu_{A^*}(x) + v_{A^*}(x) \leq 1$

[0,1] is the Membership Function (MF) rate and $v_{A^*}(x)$: X-> [0,1] is the Non-Membership Function (NMF) rate of component $x \in X$ confined by $0 \leq \mu_{A^*}(x) + v_{A^*}(x) \leq 1$. [15]

A traditional fuzzy set A is achieved if $v_{A^*}(x) = 1 - \mu_{A^*}(x)$ for every $x \in X$.

Definition 2. Given An intuitionistic fuzzy set A* an IF-index, meant via π and explained inside in [0,1] is the supplement of the summation of the rate of NMF and MF of a component x to When $\pi_{A^*}(x) = 1 - (\mu_{A^*}(x) + v_{A^*}(x))$. Obviously, $0 \leq \pi_{A^*}(x) \leq 1$ [15].

Definition 3. A Total Form-2 Intuitionistic Fuzzy Set (TF2IFS) by $\tilde{A}^*$ is make of a Form-2 MF(F2MF)$\mu_{\tilde{A}^*}(x,u)$, and a Form-2 NMF(F2NMF) $v_{\tilde{A}^*}(x,u)$.

$\tilde{A}^* = \{(x,u), \mu_{\tilde{A}^*}(x,u), v_{\tilde{A}^*}(x,u) \mid \forall x \in X, \forall u \in J_x^{\mu}, \forall u \in J_x^{v}\}$ (1)

Such that $0 \leq \mu_{\tilde{A}^*}(x,u), v_{\tilde{A}^*}(x,u) \leq 1$ and $0 \leq \mu_{\tilde{A}^*}(x) + v_{\tilde{A}^*}(x) \leq 1$.

Wherein $\forall u \in J_x^{\mu}$ and $\forall u \in J_x^{v}$:

$$J_x^{\mu} = \left\{(x,u) : u \in \left[\underline{\mu}_{\tilde{A}^*}(x), \bar{\mu}_{\tilde{A}^*}(x)\right]\right\} \quad (2)$$

$$J_x^{v} = \left\{(x,u) : u \in \left[\underline{v}_{\tilde{A}^*}(x), \bar{v}_{\tilde{A}^*}(x)\right]\right\} \quad (3)$$

Equations. (2)-(3) for TF2IFS constitute the supports of the second MF and second NMF element quantities $x \in X$ on the third dimension, respectively. A Form-2 IFS(F2IFS) must crash to an IFS. A F2IFS can also be described as,

$$\int_{x \in X} \left[\int_{u \in J_x^{\mu}} \int_{u \in J_x^{v}} \{\mu_{\tilde{A}^*}(x,u), v_{\tilde{A}^*}(x,u)\} \right] / (x,u)$$

Wherein $\iint$ represents union over all allowable quantities of u and x upon a continuous discourse universe and $\sum$ as a successor for discrete discourse universe. When $\mu_{\tilde{A}^*}(x,u) = 1$ and $v_{\tilde{A}^*}(x,u) = 1$.

Definition 4. An interval form-2 intuitionistic fuzzy system (IF2IFS), $\tilde{A}^*$ consist of a F2MF and a F2NMF demonstrated as $\left[\underline{\mu}_{\tilde{A}^*}(x), \bar{\mu}_{\tilde{A}^*}(x)\right]$ and $\left[\underline{v}_{\tilde{A}^*}(x), \bar{v}_{\tilde{A}^*}(x)\right]$ respectively for all $x \in X$ with constraints: $0 \leq \bar{\mu}_{\tilde{A}^*}(x) + \underline{v}_{\tilde{A}^*}(x) \leq 1$ and $0 \leq \underline{\mu}_{\tilde{A}^*}(x) + \bar{v}_{\tilde{A}^*}(x) \leq 1$.

An IF2IFS can be demonstrated as:

$$\tilde{A}^* = \int_{x \in X} \int_{u \in J_x^{\mu}} \int_{u \in J_x^{v}} 1/(x,u) = \int_{x \in X} \left[\int_{u \in J_x^{\mu}} \int_{u \in J_x^{v}} 1/(u) \right] / x \quad (4)$$

wherein x is the primitive item, and u is the second variable. For IF2IFS, J_x^{μ} and J_x^{v} constitute the primitive MF and primitive NMF component quantities $x \in X$. The traces of uncertainty (TOU) of an IF2IFS comprise MF-TOU in Equation. (5) and NMF-TOU in Equation. (6).

$$\mathrm{TOU}_{\mu}(\tilde{A}^*) = \bigcup_{\forall x \in X} \left[\underline{\mu}_{\tilde{A}^*}(x), \bar{\mu}_{\tilde{A}^*}(x)\right] \quad (5)$$

$$\mathrm{TOU}_{v}(\tilde{A}^*) = \bigcup_{\forall x \in X} \left[\underline{v}_{A^*}(x), \bar{v}_{A^*}(x)\right] \quad (6)$$

The TOUs concertedly with the IF-indices totally explain the uncertainty of an IF2IFS. The IF-indices exploited for the description of the IF2IFS are as per the following:

$$\pi_c(x) = \max\left(0, \left(1 - (\mu_{\tilde{A}^*}(x) + v_{\tilde{A}^*}(x))\right)\right) \quad (7)$$

$$\bar{\pi}_{var}(x) = \max\left(0, \left(1 - \left(\bar{\mu}_{\tilde{A}^*}(x) + \underline{v}_{\tilde{A}^*}(x)\right)\right)\right) \quad (8)$$

$$\underline{\pi}_{var}(x) = \max\left(0, \left(1 - \left(\underline{\mu}_{\tilde{A}^*}(x) + \bar{v}_{\tilde{A}^*}(x)\right)\right)\right) \quad (9)$$

such that: $0 \leq \pi_c(x) \leq 1$ and $0 \leq \pi_{\mathrm{var}}(x) \leq 1$.

2.2. CUCKOO SEARCH ALGORITHM

Among optimization algorithms, CSO algorithms have advantages, including fewer basic solutions, strong search optimization competence and rapid convergence speed, making it an efficient tool for solving the nonlinear optimization issue.

Mature cuckoos can compensate for the host nests eggs to augment the possibility of success hatching of their own eggs. The gate of cuckoo eggs grows faster than the host's and cuckoo chicks grow faster, which augments the opportunity in their survival and makes the cuckoo species more competitive, in the nests of the host. But several host birds can have straight battles with the disturb cuckoos. If a bird of host finds that the eggs are not its own, it will both simply abandon its nest or these unknown eggs approach and make a novel nest elsewhere [18].

A Lévy flight is a random walk in that step-sizes are retrieved from a definite probability distribution and step directions are random and convergence. It is proven that it can explain animals' hunting patterns. Also, when searching for food by Birds and other animals seem to follow it [19].

Therefore, it is usually exploited to create novel solutions. The Lévy's flights have definite probability distributions with the directions of the steps being convergence and random. The size of random step is recovered from a Lévy distribution as per the following:

$$\text{Lévy} : u = t^{-\lambda}, (1 < \lambda \leq 3) \quad (10)$$

Wherein the number of iterations is t. We are required to create a step size obeying next a random direction and a given Lévy distribution, to create the Lévy flights random numbers. The size of step is calculated as per the following:

$$\mu = \mathrm{u}/\mathrm{v}|^{1/\beta} \quad (11)$$

Wherein β is a constant; and a normal distribution for both u and v as per the following: $u: N(0, \sigma_u^2)$ and $v: N(0, \sigma_v^2)$

$$\sigma_u = \left(\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma((1+\beta)/2)2^{(\beta-1)/2}\beta}\right)^{1/\beta}, \sigma_v = 1 \quad (12)$$

Wherein $\Gamma(z)$ is a gamma function as per the following:

$$\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt \quad (14)$$

For the k th host nest, the d th dimension position update at the iteration is:

$x_k^d(t+1) = x_k^d(t) + c_1\mu\left(x_k^d(t) - \text{bnest }^d\right)r \quad (15)$

Wherein a random number is r following with variance 1 and mean 0 for a normal distribution, the step size coefficient is c_1 that is usually set to be 0.01 , the global best nest location is *bnest* ^{d} in the d th dimension [20].

Every egg of cuckoo in a nest of host has a possibility to be found via its bird of host as marked via p_a. Once found, its bird of host will generate a new nest close to the old one. The interval between two nests is selected randomly from those among all nests [21].

$$x_k^d(t) = x_k^d(t) + r_1\left(x_i^d(t) - x_j^d(t)\right) \quad (16)$$

Where a nest's position is updated by Equation. (14) or Equation. (15), the nest may be fourth of a space of solution. In this state, we use parameters $X_{\min}^d$ and $X_{\max}^d$ to restrict it.

$$x_{t,k}^d = \begin{cases} X_{\min}^d, & \text{if } x_{t,k}^d < X_{\min}^d \\ X_{\max}^d, & \text{if } x_{t,k}^d > X_{\max}^d \\ x_{t,k}^d, & \text{else} \end{cases} \quad (17)$$

Respectively, the d th dimension minimum and maximum in the space of solution are $X_{\min}^d$ and $X_{\max}^d$.

2.3. The UFastSLAM

We mark the robot pose at time t via x_t and the map by θ. The map Included a set of features, every of which will be marked via θ_n and the stationary features total number will be mentioned via N. The SLAM issue can be formulated in a Bayesian probabilistic type as per the following [22]:

$$p(x^t, \theta \mid z^t, u^t, n^t) = p(x^t \mid z^t, u^t, n^t)\prod_{n=1}^{N} p(\theta_n \mid x^t, z^t, u^t, n^t) \quad (18)$$

Wherein the data association is $n^t = \{n_1, \dots, n_t\}$, the robot route is $x^t = \{x_1, \dots, x_t\}$, the control input $u^t = \{u_1, \dots, u_t\}$ and the observation is $z^t = \{z_1, \dots, z_t\}$

The FastSLAM method using a PF demonstrates the robot route posterior $p(x^t \mid z^t, u^t, n^t)$ and N the landmark posteriors $p(\theta_n \mid x^t, z^t, u^t, n^t)$ are realized via a UKF. Every particle in the FastSLAM consists of a robot route and a feature location set with their covariance as per the following:

$$X_t^m = \left[x^{t,m}, \mu_{1,t}^m, \Sigma_{1,t}^m, \dots, \mu_{N,t}^m, \Sigma_{N,t}^m\right] \quad (19)$$

where m is the particles index, $x^{t,m}$ is the m th particle's route estimate, $\Sigma_{N,t}^m$ and $\mu_{N,t}^m$ are the covariance and mean of the Gaussian distribution deputation the Nth location of feature, connected to the mth particle. In the UFastSLAM, a new probable structure is proposed to dominate the mistakes effected via linearization in FastSLAM. The UFastSLAM calculates the proposal distribution via the UKF measurement updates in the sampling step and updates every feature state via the UKF without computing the observation model Jacobian matrix [23]. The method of the UFastSLAM composed of the importance weights calculation, robot pose estimation, feature initialization, and feature estimation.

2.3.1. Estimation of robot pose

The pose of robot X_t is sampled as:

$$x_t: q(x_t \mid x^{-1,m}, z^t, u^t, n^t) \quad (20)$$

wherein the proposal distribution mark q. augmented covariance matrix and an augmented state and is formed via supplementary the covariance and mean of process noise vector to create the proposal distribution. The process noise mean is assuming zero, the covariance matrix and augmented state is as per the following:

$$xa_{t-1}^m = \begin{bmatrix} x_{t-1}^m \\ 0 \\ 0 \end{bmatrix}, Pa_{t-1}^m = \begin{bmatrix} P_{t-1}^m & 0 & 0 \\ 0 & Q_t & 0 \\ 0 & 0 & R_t \end{bmatrix} \quad (21)$$

wherein the augmented state mark xa_{t-1}^m, the augmented covariance matrix mark Pa_{t-1}^m, the covariance of control noise mark Q_t, the mean mark x_{t-1}^m and the covariance of the robot pose at time $t-1$ mark P_{t-1}^m. A collection of $2n+1$ sigma points is sampled from the augmented state as per the following:

$$\begin{aligned} \delta a_{t-1}^{0,m} &= xa_{t-1}^m \\ \delta a_{t-1}^{i,m} &= xa_{t-1}^m + (\sqrt{(n+\lambda)Pa_{t-1}^m})_i \\ \delta a_{t-1}^{j,m} &= xa_{t-1}^m - (\sqrt{(n+\lambda)Pa_{t-1}^m})i = 1, \dots, n \end{aligned} \quad (22)$$

wherein the term $\left(\sqrt{(n+\lambda)Pa_{t-1}^m}\right)_i$ demonstrates *ith* column of the matrix $\sqrt{(n+\lambda)Pa_{t-1}^m}, n$ is the dimension of augmented state and λ is a scaling parameter. Each sigma point $\delta a_{t-1}^{i,m}$ contains the robot pose and a control noise, which can be represented as follows:

$$\delta a_{t-1}^{i,m} = \left[\delta_{t-1}^{i,m}\delta u_{t-1}^{i,m}\right]^T \quad (23)$$

The collection of sigma points $\delta a_{t-1}^{i,m}$ is transformed via the motion model as per the following:

$$\bar{x}_t^{-i,m} = f\left(\delta_{t-1}^{i,m}, u_t + \delta u_{t-1}^{i,m}\right) \quad (24)$$

Wherein $\bar{X}_t^{-i,m}$ is its transformed sigma point and f is the nonlinear motion function. The transformed points are exploited to calculate the predictive covariance P_t^m and mean x_t^m of the robot pose as per the following:

$$x_t^m = \sum_{i=0}^{2n} \omega_g^i \bar{X}_t^{i,m} \quad (25)$$

$P_t^m = \sum_{i=0}^{2n} \omega_c^i\left(x_t^{i,m} - x_t^m\right)\left(x_t^{i,m} - x_t^m\right)^T$

wherein the weights ω_g^i and ω_c^i are as per the following:

$$\begin{aligned} \omega_\xi^0 &= \frac{\lambda}{n+\lambda} \\ \omega_c^0 &= \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta) \\ \omega_\xi^i &= \omega_c^i = \frac{1}{2(n+\lambda)} \quad (i = 1, \dots, 2n) \end{aligned} \quad (26)$$

The Knowledge of the posterior distribution demonstrates with β and the constant α is usually collected to a small positive value ($10^{-4} \le \alpha \le 1$). The Kalmangain K_t^m and predicted measurement z_t^m are computed as per the following:

$$\zeta_t^{i,m} = h(\bar{x}_t^{i,m}, \mu_k^m)$$
$$\overline{z}_t^m = \sum_{i=0}^{2n} w_g^i \zeta_t^{i,m} \quad (27)$$

$$P_{vv}^m = \sum_{i=0}^{2n} w_c^i (\zeta_t^{i,m} - \overline{z}_t^m)(\zeta_t^{i,m} - \overline{z}_t^m)^T + R_t$$
$$P_{\delta v}^m = \sum_{i=0}^{2n} \omega_c^i (\overline{x}_t^{i,m} - x_t^m)(\zeta_t^{i,m} - \overline{z}_t^m)^T \quad (28)$$
$$K_t^m = P_{\delta v}^m (P_{vv}^m)^{-1}$$

wherein the observation model marked with $h(.).The$ measurement is exploited to update the covariance P_t^m and predicted pose mean x_t^m as per the following:

$$x_t^m = x_t^m + K_t^m (z_t - \bar{z}_t^m)$$
$$P_t^m = P_t^m - K_t^m P_{vv}^m (K_t^m)^T \quad (29)$$

wherein the true measurement marked with Z_t. From the Gaussian distribution created via the mean and estimated covariance of the vehicle, the state of every particle is sampled as per the following:

$$x_t : N(x_t; x_t^m, P_t^m) \quad (30)$$

The robot pose is predicted without the measurement update using Equation. (25) when there is no observation. Equation. (29) is replicated for every observed landmark, the covariance and mean of the robot pose are updated pursuant to the previously updated one If many features are observed at the same time.

2.3.2. Feature update

This step for updating the location of the landmark n_t uses the observations. By the observed feature extract a set of sigma points as per the following:

$$\chi 0, m = \mu_{n_t,t-1}^m$$
$$\chi i, m = \mu_{n_t,t-1}^m + \left(\sqrt{(n+\lambda) \sum_{n_t,t-1}^{m}} \right)_i \ (i = 1, \dots, n) \quad (31)$$
$$\chi i, m = \mu_{n_t,t-1}^m - \left(\sqrt{(n+\lambda) \sum_{n_t,t-1}^m} \right)_i \ (i = n+1, \dots, 2n)$$

wherein $\sum_{n_t,t-1}^m$ and $\mu_{n_t,t-1}^m$ are the covariance and mean matrix of the nth feature in feature initialization step is registered. The Kalman gain $\bar{K}_t^m$ and predicted measurement $\hat{z}_t^m$ are computed as per the following:

$$\bar{Z}_t^{i,m} = h(\chi^{i,m}, x_t^m) \ (i = 0, \dots, 2n)$$
$$\widehat{Z_t^m} = \sum_{i=0}^{2n} \omega_g^i Z_t^{i,m} \quad (32)$$

$$\overline{X}_t^m = \sum_{i=0}^{2n} \omega_c^i \left(\overline{Z}_t^{i,m} - \widehat{Z_t^m} \right) \left(\overline{Z}_t^{i,m} - \overline{Z_t^m} \right)^T + R_t$$
$$\overline{\Sigma}_t^m = \sum_{i=0}^{2n} \omega_c^i \left(\chi^{i,m} - \mu_{n_t,t-1}^m \right) \left(\overline{Z}_t^{i,m} - \widehat{Z_t^m} \right)^T \quad (33)$$
$$\overline{K}_t^m = \overline{\Sigma}_t^m \left(\overline{X}_t^m \right)^{-1}$$

Finally, the covariance $\Sigma_{n_t,t}^m$ and mean $\mu_{n_t,t}^m$ of the n_t th feature are updated as per the following:

$$\mu_{n_t,t}^m = \mu_{n_t,t-1}^m + \bar{K}_t^m (z_t - \hat{z}_t^m) \quad (34)$$
$$\Sigma_{n_t,t}^m = \Sigma_{n_t,t-1}^m - \bar{K}_t^m S_t^m (\bar{K}_t^m)^T \quad (35)$$

2.3.3. Feature initialization

The feature covariance $\Sigma_{n,t}^m$ and feature mean $\mu_{n,t}^m$ are initialized as a measurement function z_t and robot pose $x_t^{[m]}$ as per the following [23]:

$$\psi = \left[z_t, z_t + \sqrt{(n+\lambda) R_t}, z_t - \sqrt{(n+\lambda) R_t} \right]$$
$$\overline{M}_t^{i,m} = h^{-1}(\psi^{i,m}, x_t^m) \quad (i = 0, \dots, 2l)$$
$$\mu_{n_t,t}^m = \sum_{i=0}^{2n} \omega_g^i \overline{M}_t^{i,m} \quad (36)$$
$$\Sigma_{n_t,t}^m = \sum_{i=0}^{2n} \omega_c^i \left(\overline{M}_t^{i,m} - \mu_{n_t,t}^m \right) \left(\overline{M}_t^{i,m} - \mu_{n_t,t}^m \right)^T$$

2.3.4. The importance weight Computation

The importance weight for every particle is calculated via:

$$w_t^m = w_{t-1}^m \frac{p(z_t \mid x_t^m) p(x_t^m \mid x_{t-1}^m, u_t)}{N(x_t; x_t^m, P_t^m)} \quad (37)$$

Since the importance of weights variance increases during the time particles are resampled based on their weights. The particles with particles with high weights are multiplied and low weights are eliminated, in the resampling procedure [24].

3. The Hf SLAM

In the UFastSLAM, an incorrect a priori information of measurement noise covariance matrix (MNCM) that demonstrates probable cause to performance inconsistency and degradation. Also, it can cause divergence between particles [25]. The extra exactitude of the knowledge of the MNCM causes the extra performance. Using an adaptive algorithm is one of the useful methods to overcome the above defects [25].

Though the implementation of these methods is distinct. In the Hf SLAM, we adjusted the MNCM by IFLS, adaptively. There are two EKF in the FastSLAM, one is for updating of the proposal distribution and other is for updating landmark's position estimation. When the landmark position is updated, the MNCM is adaptively adjusted, and landmarks are static. Also, the MNCM is adjusted when the landmark position is estimated.

For achieving a better importance sampling distribution, the proposal distribution is constructed via adaptive IFLS, in the proposed method.

To improve sampling, the IFLS before sampling to movement particles to the goal of the state space, the CSO is merged in.

It can overcome the poverty of particles. It can improve the effect of every particle, when the optimization procedure causes the particles that are more far from the correct state to lead to the district where the correct state has a bigger emergence possibility. The proposed method consists of sampling, feature update, computing resampling and importance weight similar to the FastSLAM.

3.1. Adjustment MNCM

The $MNCM$ represents the exactitude of the measurement instrument. We estimated this data less and more at the prediction for the enlargement of the MNCM for estimated data means.

When the landmark position is updated, since landmarks are static, we adjust the MNCM. We need to adapt the MNCM

only because features are static. The adjustment is to make the actual content of the covariance of the residual consistent with its theoretical content. If the theoretical covariance and its actual covariance are contradictory, an IFLS will adjust the $MNCM$ online to reduce the difference. Therefore, the method for adjustment of the MNCM can be derived. The significant idea of this method is to create the actual data of the residual covariance to be in accordance with its theoretic data. The innovation sequence is $r_t = (z_t - \hat{z}_{n_t,t})$. Through averaging in an animated window of size N can be estimate the actual residual covariance $\hat{C}_t$ via its sample covariance as per the following:

$$\hat{C}_t = \frac{1}{N} \sum_{i=k-N+1}^{t} (r_i^T r_i) \quad (38)$$

If the actual content of covariance $\hat{C}_t$ has differences with its theoretic content, then the diagonal components of the MNCM pursuant to this discrepancy size can be adjusted. The purpose of these adjustments is to exact this mismatch as much as possible. The size of the mentioned discrepancy is given by a variable called the mismatch degree (M_t), as per the following:

$$M_t = X_t - \hat{C}_t \quad (39)$$

The significant notion exploited to adjust the MNCM is from Equation. (20) an enhancement in the MNCM will inverse and

augment X_t. So, the MNCM can be exploited to differentiate X_t in association with the content to M_t, for decline the conformity of the MNCM is made pursuant to the (i,i) component of M_t. differences between $\hat{C}_t$ and X_t. The (i,i) component

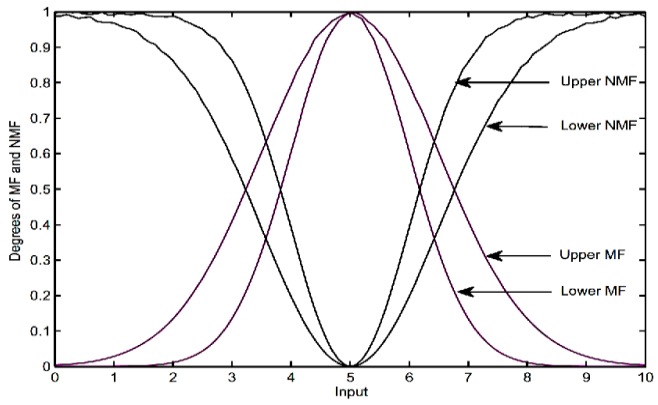


Figure. 1 Input parameter for the MF and NMF of an intuitionistic Gaussian

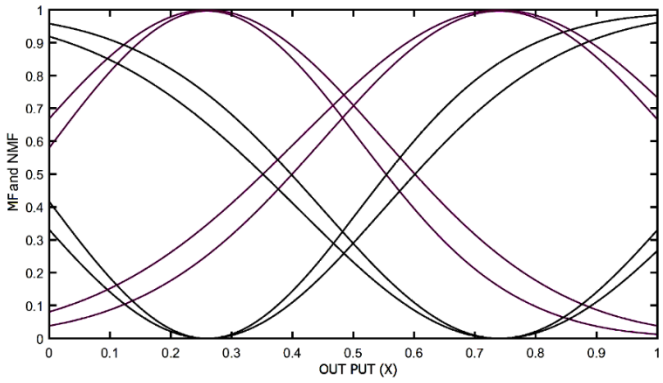


Figure. 2 Output parameter for the MF and NMF of an intuitionistic Gaussian

The IFLS is proposed to adjust MNCM. The IFLS exploits a set of subsystems where every subsystem exploits a twoinput and single-output IFLS. This is because the dimensions of M_t and $MNCM$ are both 2×2. These two-input and singleoutput IFLS are exploited to tune every MNCM component diagonal. Inputs of the IFLS are the M_t and ΔM_t. The ΔM_t is introduced as per the following:

$$\Delta M_t = M_t - M_{t-1} \quad (40)$$

Figure. 1 present the MFs and NMFs for M_t and ΔM_t. The Adjustment of the MNCM is accomplished that marked with R_t as per the following:

$$R_t = R_t + \Delta R_t \quad (41)$$

Figure. 2 is shows the MFs and NMFs of output wherein ΔR_t is the IFLS output. Figure. 3 shows the IF-THEN rule formation of IFLS follows up the fuzzy logic rule syntax that the IFLS is a six layers' network. For IFLS, the prior IFLSs are the inputs of linear compositions when the consequent sections. A rule structure of IFLS as per the following:

$$R_k: IF x_1 \text{ is } \tilde{A}_{1k}^* \text{ and } \cdots \text{ and } x_n \text{ is } \tilde{A}_{nk}^* \text{ THEN } F_k = \sum_{i=1}^{n} w_{ik} x_i + b_k \quad (42)$$

The rules are defined for the MFs in Equation. (43) and the NMFs in Equation. (44) as per the following:

$$R_k^{\mu}: \text{IF} x_1 \text{ is } \tilde{A}_{1k}^{\mu} \text{ and } \cdots \text{ and } x_n \text{ is } \tilde{A}_{nk}^{\mu} \text{THEN} F_k^{\mu} = \sum_{i=1}^{n} w_{ik}^{\mu} x_i + b_k^{\mu} \quad (43)$$

$$R_k^{v}: IF x_1 \text{ is } \tilde{A}_{1k}^{v} \text{ and } \cdots \text{ and } x_n \text{ is } \tilde{A}_{nk}^{v} \text{ THEN } F_k^{v} = \sum_{i=1}^{n} w_{ik}^{v} x_i + b_k^{v} \quad (44)$$

Wherein $\tilde{A}_{1k}^*, \tilde{A}_{2k}^*, \cdots, \tilde{A}_{ik}^*, \cdots, \tilde{A}_{nk}^*$ are the IT2IFS, the NMFs and MFs outputs of the k th rule are $F_k{}^{v}$ and $F_k{}^{\mu}$, w's are the coefficients of the independent variables of function parameters plus a constant term b 's known as the bias, and utilizes the "prod" t-norm to represent the firing strength so that:

$$\underline{f_k^{\mu}}(x) = \prod_{i=1}^{n} \underline{\mu}_{\tilde{A}_{ik}^*}(x_i), \overline{f_k^{\mu}}(x) = \prod_{i=1}^{n} \bar{\mu}_{\tilde{A}_{ik}^*}(x_i) \quad (45)$$

$$\underline{f_k^{v}}(x) = \prod_{i=1}^{n} \underline{v}_{\tilde{A}_{ik}^*}(x_i), \overline{f^{v}}_k(x) = \prod_{i=1}^{n} \bar{v}_{\tilde{A}_{ik}^*}(x_i) \quad (46)$$

The outputs of the kth rule regarding the NMFs and MFs are Wherein y_k^v and y_k^{μ}, respectively. the upper and lower firing strengths defined for both the NMFs and MFs are, $\bar{f}_k$ and $\underline{f}_k$, respectively.

The function of the consequent layer in closed form is:

$$F() =. F_k^v + F_k^{\mu} \quad (47)$$

The IFLS inference engine is defined in Equation. (48):

$$F = \frac{(1-\beta)\sum_{k=1}^{M} \left(\underline{f_k^{\mu}} + \bar{f}_k^{\mu}\right) F_k^{\mu}}{\sum_{k=1}^{M} \underline{f_k^{\mu}} + \sum_{k=1}^{M} \bar{f}_k^{\mu}} + \frac{\beta \sum_{k=1}^{M} \left(\underline{f_k^{v}} + \bar{f}_k^{v}\right) F_k^{v}}{\sum_{k=1}^{M} \underline{f_k^{v}} + \sum_{k=1}^{M} \bar{f}_k^{v}} \quad (48)$$

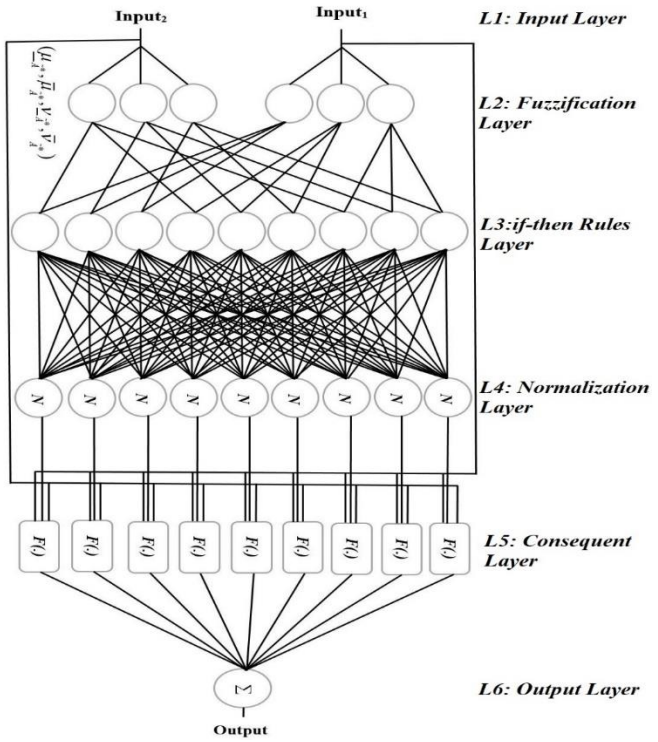


Figure. 3 The Structure of IFLS for tuning the *MNCM*

The IFLS final output is a weighted average of every IFTHEN rule's output. The parameter weighs the contribution of the MFs and NMFs values in the final output is $\beta(0 \le \beta \le 1)$.

The IFLS is trained using data from the similar area that the SLAM will accomplish in the area. The training approach adjusts the network weights via the following cost function minimization:

$$E_t = \frac{1}{2}\mathrm{tr}(e_t^2) \tag{49}$$

Wherein

$$e_t = X_t - \hat{C}_t \tag{50}$$

In order that the error defined in (48) is less than a qualified threshold value after a given amount of learning cycles is adjusted the IFLS weighting vector, by the gradient descent (GD) learning approach. The well-known GD approach perhaps written as per the following:

$$W_{t+1} = W_t - \eta\frac{\partial E_t}{\partial W_t} \tag{51}$$

Wherein is the learning rate is η and is the IFLS adjusting parameters is $W_t = \left[m_{t,MF}, m_{t,NMF}, \sigma_{t,MF}, \sigma_{t,NMF}, \omega_t\right]^T$. Wherein the width and center of the Gaussian MF are $\sigma_{t,MF}$ and $m_{t,MF}$, the width and center of the Gaussian NMF are $\sigma_{t,NMF}$ and $m_{t,NMF}$, respectively. Also, ω_t the link weights in six layers. The gradient of E due to the optional weighting vector W_t is as per the following:

$$\frac{\partial E_t}{\partial W_t} = -e_t\frac{\partial \Delta R_t}{\partial W_t} \tag{52}$$

The error situation for every layer is calculated first, and then another parameter in the relevant layers, via the chain rule recursive are adjusted as per the following:

$$\begin{aligned}
m_{t+1,\mathrm{MF}} &= m_{t,\mathrm{MF}} - \eta\frac{\partial E}{\partial m_{t,\mathrm{MF}}}\\
m_{t+1,\mathrm{NMF}} &= m_{t,\mathrm{NMF}} - \eta\frac{\partial E}{\partial m_{t,\mathrm{NMF}}}\\
\sigma_{t+1,\mathrm{MF}} &= \sigma_{t,\mathrm{MF}} - \eta\frac{\partial E}{\partial \sigma_{t,\mathrm{MF}}}\\
\sigma_{t+1,\mathrm{NMF}} &= \sigma_{t,\mathrm{NMF}} - \eta\frac{\partial E}{\partial \sigma_{t,\mathrm{NMF}}}\\
\omega_{t+1} &= \omega_t - \eta\frac{\partial E}{\partial \omega_t}
\end{aligned} \tag{53}$$

3.2. Improvement of importance weight using CSO

A cost function to appraise the nests' positions in the CSO algorithm is defined. Also, it is necessary for the combination of PF and CSO, and introduced as per the following:

$$f(x_t^m) = e^{-\frac{1}{2R}\left(z_t^m - z_{t,obs}\right)^2} \tag{54}$$

Wherein the noise covariance of observation is R, the predicted value of x_t^m is z_t^m, and the observation at time t is $z_{t,obs}$. The d th dimension position update for the k th host nest at the iteration as per the following:

$$x_{t+1,k}^d = x_{t,k}^d + c_1\mu\left(x_{t,k}^d - \text{bnest }^d\right)r \tag{55}$$

Wherein coefficient of the step size that is common equal to be 0.01 is c_1, a random number sequacious a normal distribution with covariance 1 and mean 0 is r, and the global best nest situation in the d th dimension is bnest ^{d}. All the nests move to the optimal solution, pursuant to a CSO algorithm.

Finally, the procedure of the CSO is demanding all nests to move to the area with the top likelihood possibility for them to the optimum. The nests have formerly been distributed near-optimally or optimally, when the algorithm execution achieves the maximum number of iterations or bnest cost achieves a given threshold ε.

Therefore, the combination of the PF and CSO overcomes the particle impoverishment issue. The weights of particles are updated and as per the following:

$$\omega_t^m = \omega_{t-1}^m\frac{p(z_t \mid x_t^m)p(x_t^m \mid x_{t-1}^m)}{q(x_t^m \mid x_{t-1}^m, z_t)} = \omega_{t-1}^m p(z_t \mid x_t^m) = \omega_{t-1}^m e^{-\frac{1}{2R}(z_t^m - z_{id+1})^2} \tag{56}$$

$$\omega_t^m = \omega_t^m / \sum_{m=1}^{N}\omega_t^m \tag{57}$$

It is required to resample particles to the same as basic PF does, after the weights normalization every time, propagate good ones and to discard the bad particles acquired via the CSO. Then the state can be estimated as per the following:

$$\hat{x}_t = \sum_{m=1}^{N}\omega_t^m\hat{x}_t^m \tag{58}$$

The CSO transmits all particles towards the topmost likelihood areas that is the universal best of CSO. The optimized sampling procedure is stopped, when the best adaptable value. The sampling procedure with this particle collection will be done on the proposal distribution basis. Finally, Algorithm 1 is shown the pseudo code of the Hf SLAM.

Algorithm 1 the Hf SLAM Pseudocode

(1) Initialization

Initialize the *MNCM,* the particle number *N,* and the particle collection $X_t=\phi$

(2) Accomplish the Hf SLAM

For *t= 1, 2, 3, ...* **do** // for every time instant

  For *m= 1, 2, 3, ..., N* **do** // for every particle

Calculation of robot route posterior $p(x^t \mid z^t, u^t, n^t)$

- Retrieve previous particle X_{t-1}^m from the particle collection

- Predict robot state $X_{t|t-1}^m$ and its covariance $P_{t|t-1}^m$

-. Do data-association for all measurements to make a distinction between the revisited landmarks and new visited landmarks

- Accomplish the state robot updating step, adjustment the (using the IFLS)

- Sample from proposal distribution to acquire new robot state$x_t^m$and the compute its importance weightw_t^m (using the CSO)

- Add current robot state $x_t^m$to the previous robot route x_{t-1}^m

*** Calculation of map posterior** $p(\theta_k \mid x^t, z^t, u^t, n^t)$

-Update the revisited features

- Initialize the new features, and the add then add the Gaussian densities of the new landmarks to the particles X_t

End For

Resample particle set$X_t$based on its importance weight collection w_t^m

Add new particles to X_t

Return current particle collectionX_t

End For

4. Experiments and Analysis

The Python code, to demonstrate the efficiency of the proposed method expanded by Atsushi, mutated [26]. There are restrictions on system and observation noise, steering angle, and velocity, for robots with maximum speeds of 3[m/sec] and a wheel diameter of 1[m]. The steering angle and maximum speed are 25[°] and 15[°/sec], respectively.

A zero mean Gaussian is supposed to the control input noise with $\sigma_V(= 3[\,°])$ and $\sigma_V(= 0.2[\,\mathrm{m/s}])$. For observation, the number of optional features around waypoints was exploited. An observation model and range bearing sensor model were exploited to appraise the position of feature and pose of the robot, that includes a noise with 1[°] in bearing and a level of 0.1[m] in span. The sensor span is limited to 15[m] for small areas and 30[m] for large areas. Two navigation cases are surveyed: the floor navigation, and the Victoria Park navigation as explained in Table 1.

Table 1 Fundamental specs for navigation

Table 1 Fundamental specs for navigation

Item	Floor	Victoria Park
Feature	17	73
Waypoint	18	79
Area[m]	16*17	250*300

4.1. Navigation Results in the floor map

In Figure. 4, navigation pursuant to the FastSLAM, UFastSLAM and Hf SLAM are demonstrated wherever more deviations are shown on the edges with a larger angle during navigation procedure. The proposed method efficiency is compared with the FastSLAM and UFastSLAM. The proposed method uses wrong contents and then adapts the MNCM in the UKF via the IFLS and tries to minimize the mismatch between the actual and theoretical contents of the sequence of innovation in the UKF.

The IFLS parameters are automatically learned by GD during training. The blue line, are demonstrant the routes of robots must cover, the red line is the route of the robot and green line is laser rays for see the landmarks, pursuant to data explained via the real odometry. The plus points (+) represents the landmarks location.

In Figure. 5, the green, red and blue lines the X, Y, and Θ errors the FastSLAM, UFastSLAM and Hf SLAM, respectively. The inference procedure decreases errors and is more accurate over time, for small areas.

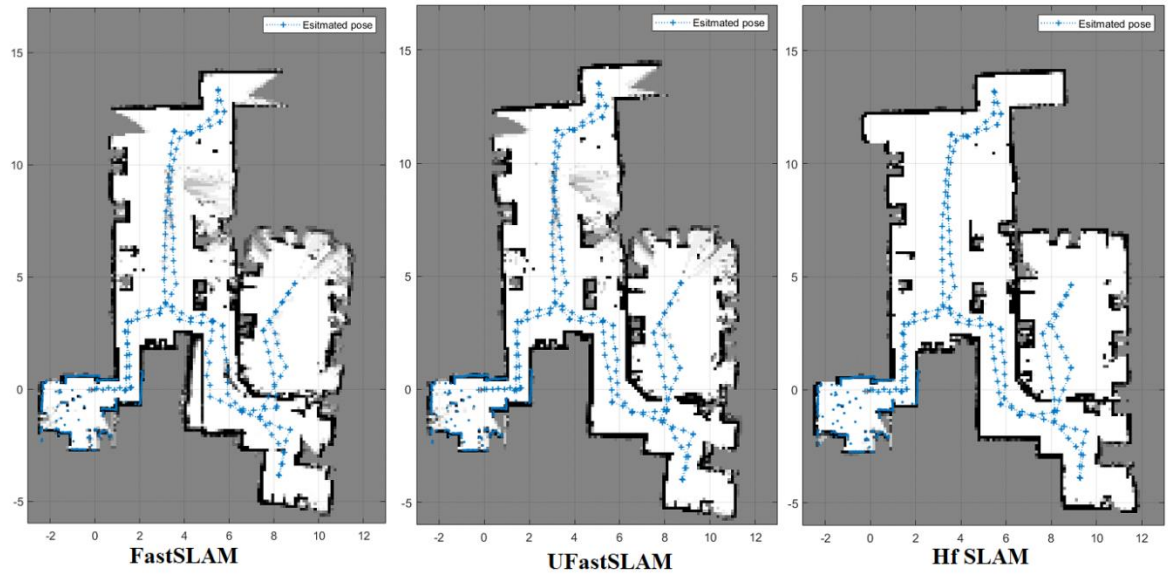


Figure. 4 Results of navigation for 30 Particles in the floor map

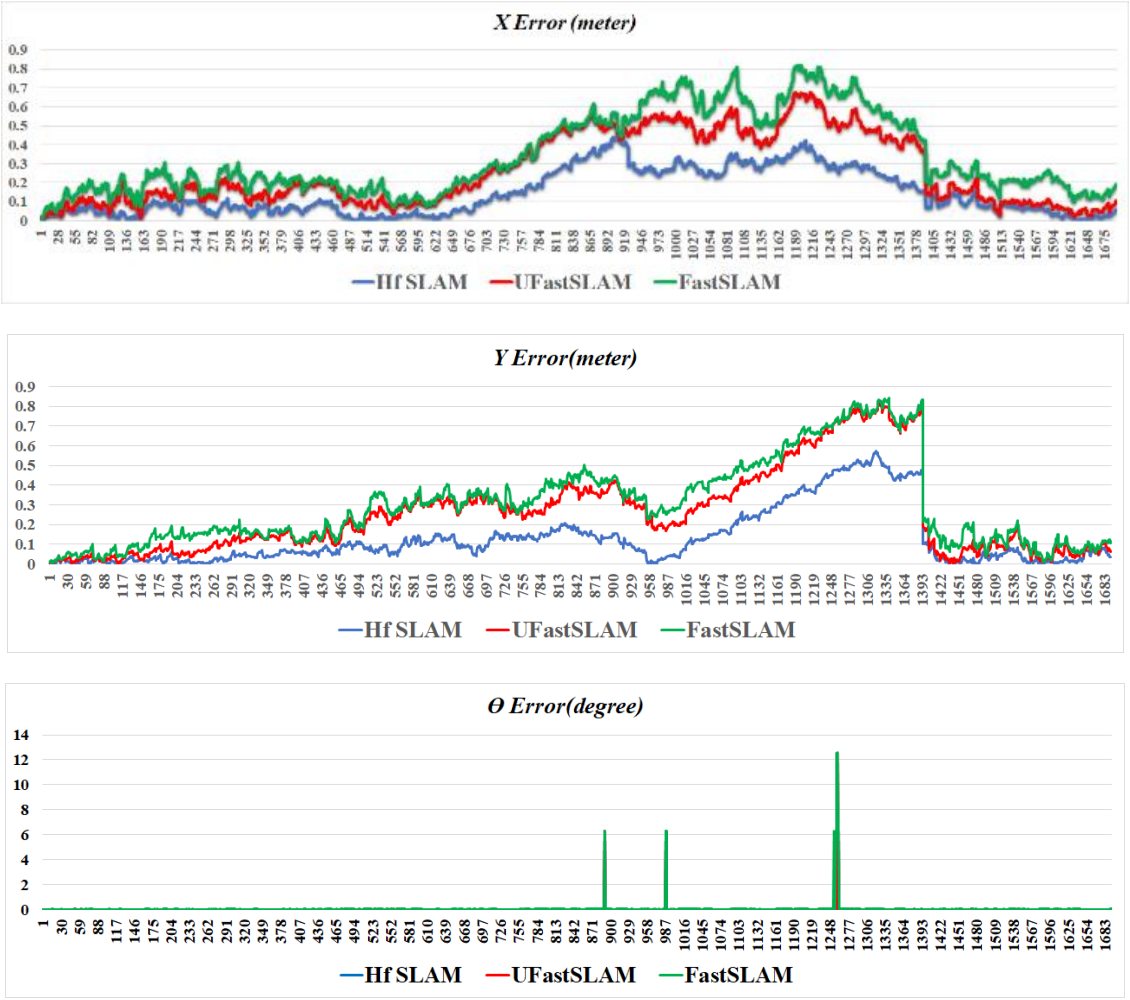
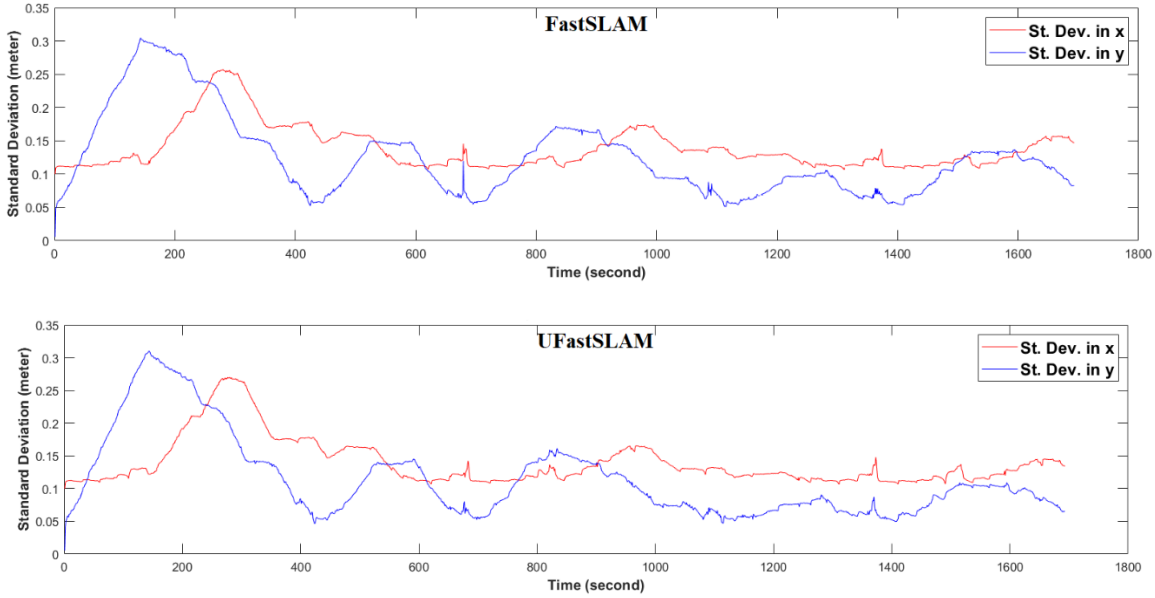


Figure. 5 Errors of navigation for 30 Particles in the floor map



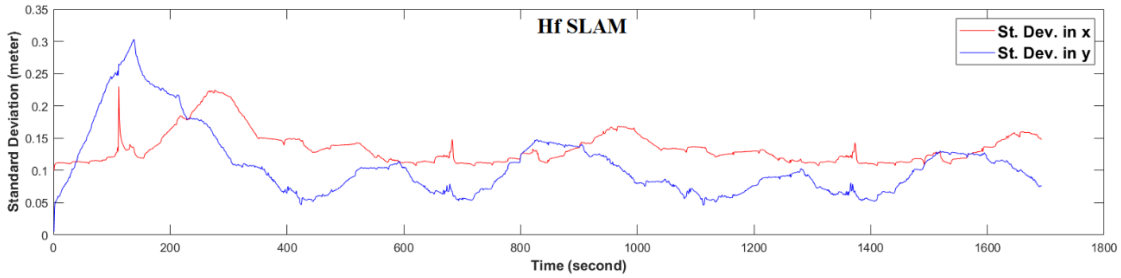


Figure. 6 Position Standard deviation for 30 Particles in the floor map

In Figure. 6, by comparing the final estimation of the position, the standard deviation curve of the position deviation, the real position deviation, and the state amount of x, y are demonstrated for the UFastSLAM, FastSLAM, and Hf SLAM.

The all over position deviation acquired via the Hf SLAM is less than that of the UFastSLAM and FastSLAM deviation. These deviations may demonstrate that there is unfit deviation control to calculate the vehicle's turns. So the IFLS has a more appropriate effect on positioning exactitude via adjusting the noise model in real time. During the all navigation process, the stability of the filter is effectively improved and the positioning error ever swings within a low range.

Table 2 provides the RMSE of the vehicle position and running time of the proposed method compared to all methods. The result demonstrates that the Hf SLAM improves the positioning exactitude of robots, because the proposed method adaptively tuned the *MNCM*. These matrices converge to the actual *MNCM* when the matrix *MNCM* in the UFastSLAM and FastSLAM are kept constant all time.

The proposed method offered the CSO method into the adaptive UKF to progress the distribution samples, and accelerate the convergence of the particle set. Also, the proposed method consumes less running time. Therefore, the proposed method has better precesion and computational efficiency than that of the UFastSLAM and FastSLAM.

Table 2 Robots position RMSE and running time of methods in the floor map.

Number Particles	Algorithms	RMSE (m)	Average Runtime (s)
50	*FastSLAM*	1.452	58.1
	UFastSLAM	1.214	49.4
	Hf SLAM	0.564	47.6
30	*FastSLAM*	0.735	38.2
	UFastSLAM	0.519	36.7
	Hf SLAM	0.198	35.2
10	*FastSLAM*	0.341	31.4
	UFastSLAM	0.175	30.5
	Hf SLAM	0.088	30.1

4.2. Navigation experimental Results in the 'Victoria Park dataset'

The experimental is carried out in the Victoria Park dataset until validation of the efficiency of the proposed method demonstrated for solving SLAM issue [27]. The vehicle provided with different sensors is shown in Figure. 7a.

For supply ground truth data, steering angle and vehicle velocity was exploited a GPS were gathered with an inertial sensor. A laser range finder was exploited to landmarks bearing and measure range with the vehicle.

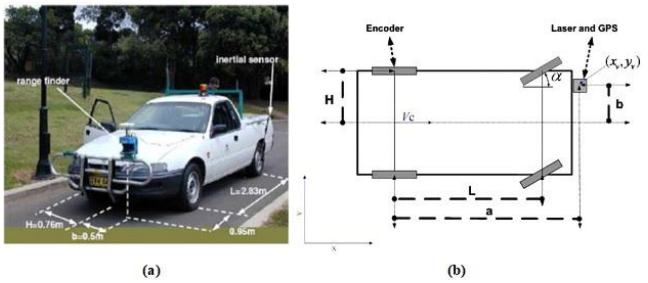


Figure. 7 (a) The vehicle for data gathering. (b) The motion model of vehicle [28]

The type of car-like robot is considered for its motion model [28]. The motion model is demonstrates as per the following:

$$x_v = v\cos(\theta)\,, \quad y_v = v\sin(\theta) \quad \text{and} \quad \theta_v = \frac{v}{L\tan(\alpha)} \qquad (59)$$

The vehicle motion model is shown in Figure. 7b and Equation. (60) demonstrates the pose of the center of the back axle, but a laser range finder and GPS are installed at the front of the vehicle. After coordinate transformation, discrete motion model is demonstrated as per the following [29]:

$$\begin{bmatrix} x_{k,v} \\ y_{k,v} \\ \theta_{k,v} \end{bmatrix} = \begin{bmatrix} x_{k-1,v} + \Delta t\left(v_{k-1}\cos\left(\theta_{k-1,v}\right) - \frac{v_{k-1}}{L}\tan\left(\alpha_{k-1}\right)\left(a\sin(\theta_{k-1,v}) + b\cos\left(\theta_{k-1,v}\right)\right)\right) \\ y_{k-1,v} + \Delta t\left(v_{k-1}\sin\left(\theta_{k-1,v}\right) + \frac{v_{k-1}}{L}\tan\left(\alpha_{k-1}\right)\left(a\cos(\theta_{k-1,v}) - b\sin\left(\theta_{k-1,v}\right)\right)\right) \\ \theta_{k-1,v} + \Delta t\frac{v_{k-1}}{L}\tan\left(\alpha_{k-1}\right) \end{bmatrix} + W_{k-1} \qquad (60)$$

$$v_{k-1} = \frac{v_{k-1,e}}{1-\frac{H}{L}\tan\left(\alpha_{k-1}\right)} \qquad (61)$$

Wherein the velocity of the back axle is v and the sampling time is t, but v_e get from inertial sensor demonstrates the the back left wheel velocity. Navigation pursuant to the Hf SLAM, FastSLAM and UFastSLAM are demonstrated in Figure. 8, wherein more diversions are shown on edges with larger angles in navigation procedure.

The vehicle determines a direction for the navigation pursuant to the data from the landmarks detected locations, due to unpredictable alterations in incoming information but it does not right away turn in the edges. The green line, are shown the routes of vehicle should cover with the blue line is route of vehicle and the black line is GPS, pursuant to information explained via the Hf SLAM.The red circle (o) depict the landmarks location.

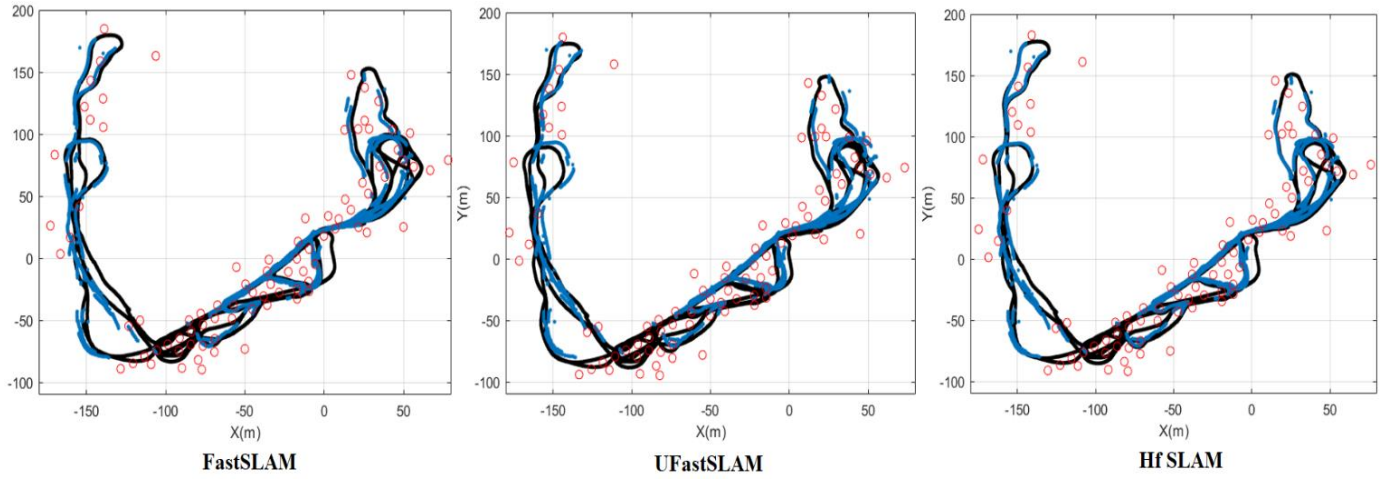


Figure. 8 Experimental results for 30 Particles in Victoria Park map

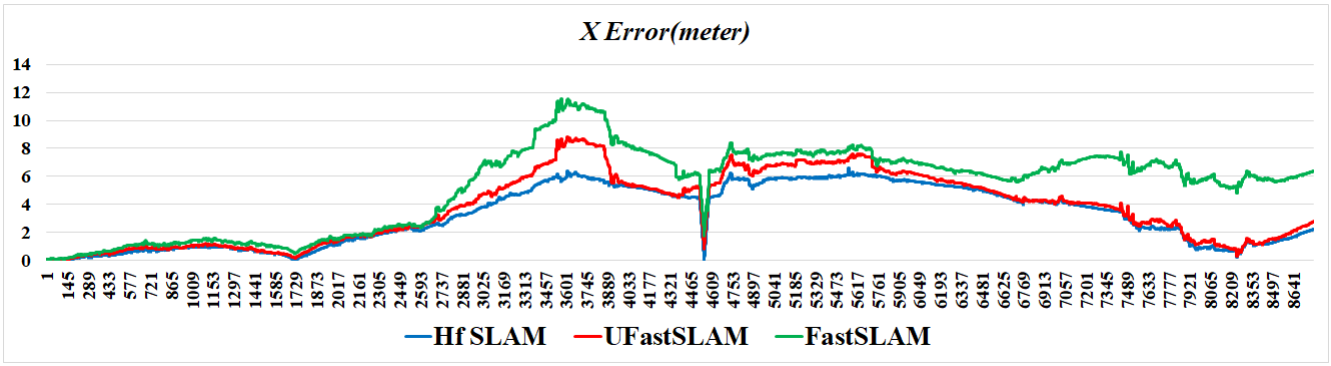
The results showed that efficiency of the proposed method is more efficient than both other methods.

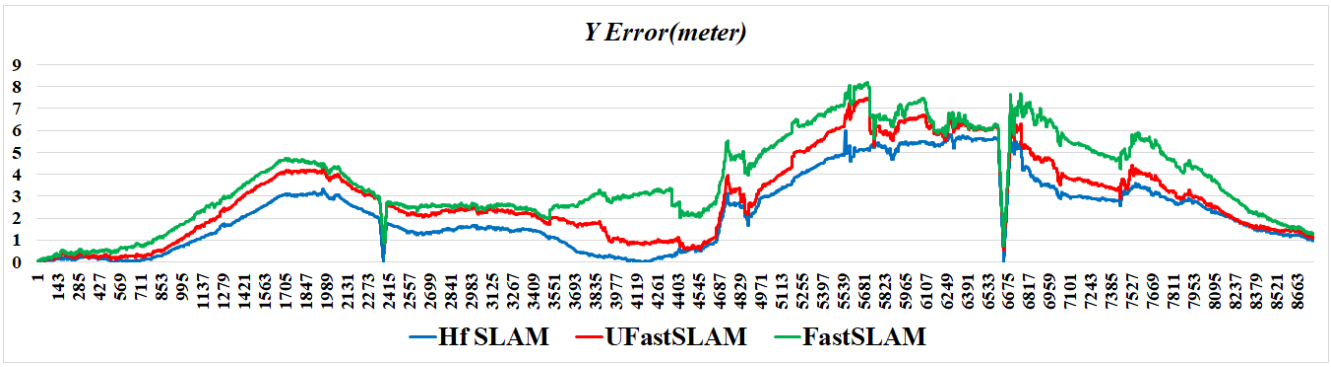
Also, the performance of all methods dependent on incrementing the number of loops and the number of hypothetical Jacobians will create these methods more complex and accurate.

Also, when all methods are exploited to solve a different issue with more complexity may be augmented.

Thus, a suitable value for loops should be selected to create a balance between computational complexity and estimation exactitude. As the update and predict phase are optimumed on the proposed method, the uncertainties are spread well, and the exactitude of the state estimation has been improved over that of the both another method.

In Figure. 9, the green, red and blue lines the *x, y*, and Θ errors.





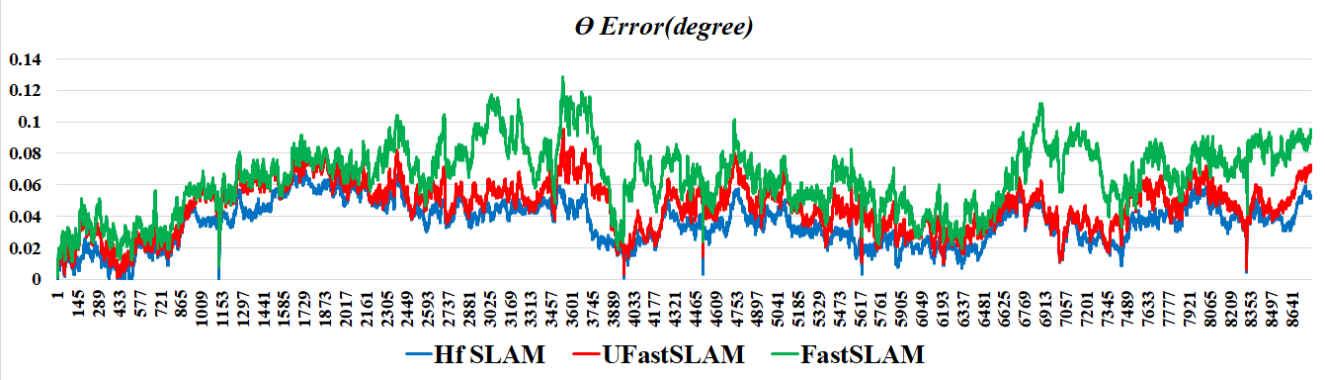


Figure. 9 Navigation errors for 30 Particles in the Victoria Park map

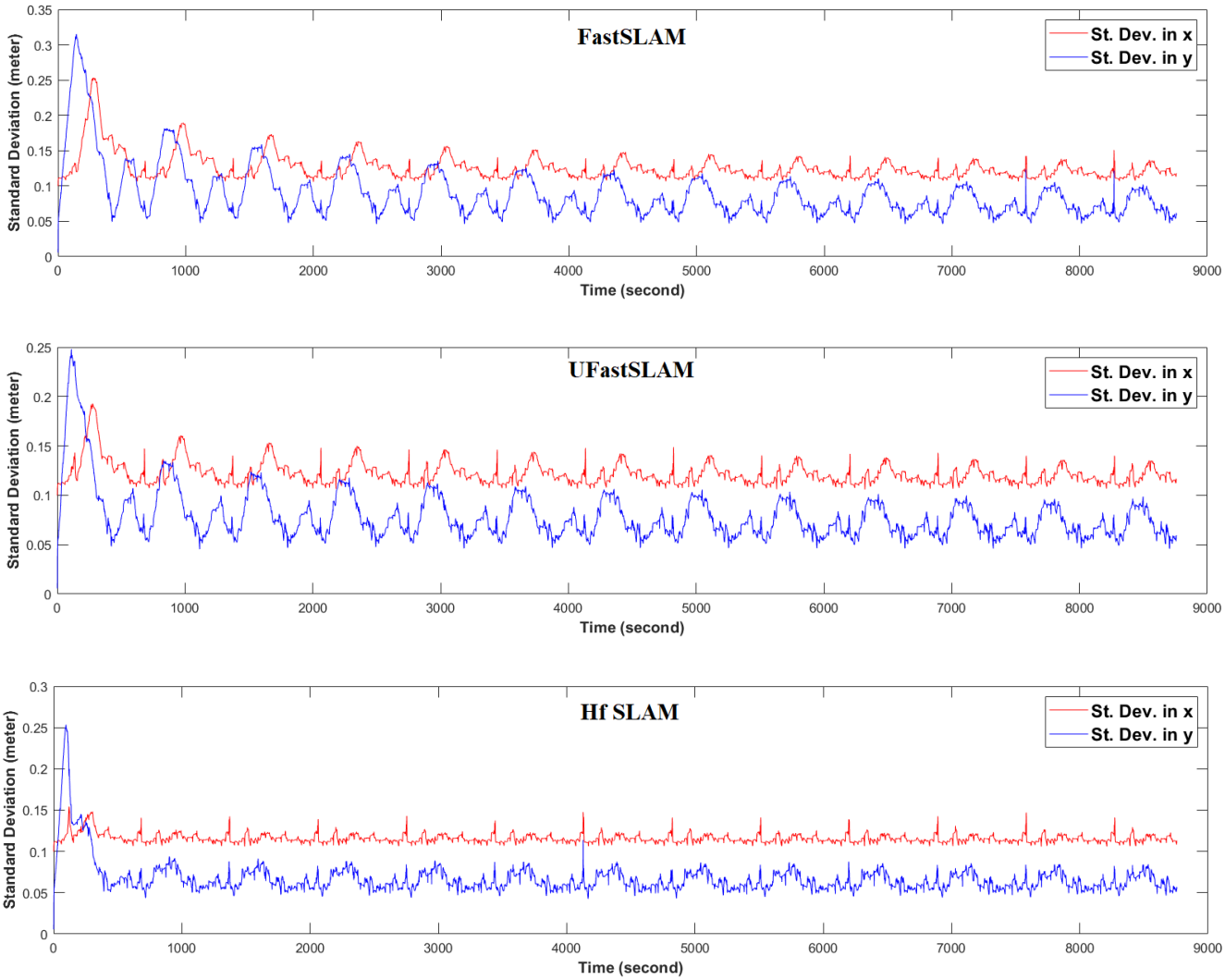


Figure. 10 Position Standard deviation for 30 Particles in the Victoria Park map

In Figure. 10, by comparing the final estimation of the position, the standard deviation curve of the position deviation, the real position deviation, and the state amount of x, y are demonstrated for the UFastSLAM, FastSLAM, and Hf SLAM.

The all over position deviation acquired via the Hf SLAM is less than that of the UFastSLAM and FastSLAM deviation. These deviations may demonstrate that there is unfit deviation control to calculate the vehicle's turns. So the IFLS has a more appropriate effect on positioning exactitude via adjusting the noise model in real time. During the all navigation process, the stability of the filter is effectively improved and the positioning error ever swings within a low range.

Table 3 provides the RMSE of the vehicle position and running time of the proposed method compared to all methods. The result demonstrates that the Hf SLAM improves the positioning exactitude of robots, because the proposed method adaptively tuned the *MNCM*. These matrices converge to the actual *MNCM* when matrix the

MNCM in the UFastSLAM and FastSLAM are kept constant all time.

The proposed method offered the CSO method into the adaptive UKF to progress the distribution samples, and accelerate the convergence of the particle set. Also, the proposed algorithm consumes more running time. Therefore, the proposed method has better computational efficiency and exactitude than the UFastSLAM and FastSLAM.

Table 3 Vehicle position RMSE and running time of three methods in the Victoria Park map

Number Particles	Algorithms	RMSE (*m*)	Average Runtime (*s*)
50	*FastSLAM*	7.954	143.5
	UFastSLAM	6.214	167.4
	Hf SLAM	3.564	194.6
30	*FastSLAM*	6.523	93.6
	UFastSLAM	4.418	97.3
	Hf SLAM	2.712	105.8
10	*FastSLAM*	5.329	66.4
	UFastSLAM	3.675	71.7
	Hf SLAM	1.788	87.4

5 Conclusions

The SLAM is one of the most significant issues for robots as the robots follow its position by preserving a map of areas and an estimate of its position in that map.

In this paper for navigation procedure of an robots proposed the Hf SLAM method, in order to reduce the UFastSLAM procedure errors. The incorrect past knowledge about the robots may very well decline the performance of UFastSLAM.

In order to decline the efficacy of the cumulative. It is concluded that correcting the formula exploited to compute the perversion of the linear approximation and Jacobian matrix of observation function can decline the linearization error. The Hf SLAM does not use the production of the linear approximations and Jacobian matrices to the nonlinear functions in the UFastSLAM and FastSLAM is the significant superiority and updates the covariance and mean of the feature state via IFLS.

Also, the procedure of resampling is done pursuant to the CSO to augment diversity. The CSO is exploited to optimize performance of sampling in the SLAM. The CSO causes the set of particles to push on the great likelihood area before the weight is updated, as an outcome the particle's impoverishment can be overcome.

The Hf SLAM implemented in IFLS provides better proposal distribution and the prediction step of the robots state, in the localization procedure. The IFLS is used for dynamically adjusting the procedure noise covariance, and it will be exploited via monitoring the innovation data. The proposed method, via the filter model lower order can be exploited, therefore, less calculation effort will be enough.

When the filter parameters are calmly changing with time or When a designer does not have equated information to extend the complete filter models, the IFLS can be occupied to improve the UFastSLAM and FastSLAM performance. The proposed method, when significant training the estimation exactitude, declining the amount of particles has the additional advantage.

The results of the experimental and simulation for three diverse navigation cases showed the efficiency of the proposed method as compared with the both other methods, also, shown that the proposed method has efficient results in the wider area but we require to use of long range sensors. Pursuant to results, the Hf SLAM has fewer errors than the both other methods and can improve the exactitude of estimation clearly and significantly train the diversity. In addition, the results accept that the Hf SLAM is more resistant for the robot's navigation procedure and the consistency of it is higher than that of the both other methods.

It is significant to make a trade off between estimation exactitude and computational complexity.

References

[1] C. Kim, R. Sakthivel, and W. K. Chung, "Unscented FastSLAM: A robust and efficient solution to the SLAM problem," *IEEE Trans. Robot.*, vol. 24, no. 4, pp. 808–820, 2008.

[2] S. Zeghlache, D. Saigaa, and K. Kara, "Fault tolerant control based on neural network interval type-2 fuzzy sliding mode controller for octorotor UAV," *Front. Comput. Sci.*, vol. 10, pp. 657–672, 2016.

[3] P. Ramasamy and P. Kandhasamy, "Effect of intuitionistic fuzzy normalization in microarray gene selection," *Turk. J. Electr. Eng. Comput. Sci.*, vol. 26, no. 3, pp. 1141–1152, 2018.

[4] Y. Song, Q. L. Li, and Y. F. Kang, "Conjugate Unscented FastSLAM for autonomous mobile robots in large-scale areas," *Cogn. Comput.*, vol. 6, no. 3, pp. 496–509, 2014.

[5] R. Havangi, "A mutated FastSLAM using soft computing," *Ind. Robot Int. J.*, vol. 44, no. 4, pp. 416–427, 2017.

[6] S. Wen, J. Chen, X. Lv, and Y. Tong, "Cooperative simultaneous localization and mapping algorithm based on distributed particle filter," *Int. J. Adv. Robot. Syst.*, vol. 16, no. 1, pp. 1–8, 2019.

[7] H. Shi, X. Li, W. Pan, K. S. Hwang, and Z. Li, "A novel fuzzy three-dimensional grid navigation method for mobile robots," *Int. J. Adv. Robot. Syst.*, vol. 14, pp. 1–16, 2017.

[8] M. Lin, C. Yang, and D. Li, "An improved transformed Unscented FastSLAM with adaptive genetic resampling," *IEEE Trans. Ind. Electron.*, vol. 66, no. 5, pp. 3583–3594, 2018.

[9] C. H. Do and H. Y. Lin, "Incorporating neuro-fuzzy with extended Kalman filter for simultaneous localization and mapping," *Int. J. Adv. Robot. Syst.*, vol. 16, no. 5, pp. 1–13, 2019.

[10] S. H. Lee, G. Eyoh, and B. H. Lee, "Relational FastSLAM: An improved Rao-Blackwellized particle filtering framework using particle swarm characteristics," *Robotica*, vol. 34, no. 6, pp. 1282–1296, 2016.

[11] C. C. Hsu, W. Y. Wan, and T. Y. Lin, "Enhanced simultaneous localization and mapping (ESLAM) for mobile robots," *Int. J. Humanoid Robot.*, vol. 14, no. 2, pp. 1–17, 2017.

[12] P. Bozek, P. Pokorný, J. Svetlík, A. Lozhkin, and I. Arkhipov, "The calculations of Jordan curves trajectory of

the robot movement," *Int. J. Adv. Robot. Syst.*, vol. 13, no. 5, pp. 1–7, 2016.
[13] P. K. Mohanty and D. R. Parhi, "Optimal path planning for a mobile robot using cuckoo search algorithm," *J. Exp. Theor. Artif. Intell.*, vol. 28, no. 1, pp. 1–18, 2014.
[14] S. Hosseininejad and C. Dadkhah, "Mobile robot path planning in dynamic environment based on cuckoo optimization algorithm," *Int. J. Adv. Robot. Syst.*, vol. 16, no. 2, pp. 1–13, 2019.
[15] Y. Tian, H. Suwoyo, W. Wang, D. Mbemba, and L. Li, "An AEKF-SLAM algorithm with recursive noise statistic based on MLE and EM," *J. Intell. Robot. Syst.*, vol. 97, pp. 339–355, 2019.
[16] K. T. Atanassov, "Intuitionistic fuzzy sets," *Fuzzy Sets Syst.*, vol. 20, no. 1, pp. 87–96, 1986.
[17] J. Li and X. Xu, "Unified forms of the CDR method of approximate reasoning on Atanassov's intuitionistic fuzzy sets and its property analysis," *Comput. Intell.*, vol. 34, pp. 1–21, 2018.
[18] X. S. Yang and S. Deb, "Multi-objective cuckoo search for design optimization," *Comput. Oper. Res.*, vol. 40, no. 6, pp. 1616–1624, 2013.
[19] L. Wang, Y. Yin, and Y. Zhong, "Erratum to: Cuckoo search with varied scaling factor," *Front. Comput. Sci.*, vol. 9, no. 4, pp. 623–635, 2015.
[20] A. M. Reynolds and F. Bartumeus, "Optimizing the success of random destructive searches: Levy walks can outperform ballistic motions," *J. Theor. Biol.*, vol. 260, no. 1, pp. 98–103, 2009.
[21] A. Wang, H. Gong, and F. Chen, "Fault estimation and compensation for hypersonic flight vehicle via type-2 fuzzy technique and cuckoo search algorithm," *Int. J. Adv. Robot. Syst.*, vol. 17, no. 1, pp. 1–12, 2020.
[22] S. Thrun *et al.*, "FastSLAM: An efficient solution to the simultaneous localization and mapping issue with unknown data association," *J. Mach. Learn. Res.*, 2004.
[23] C. C. Hsu, C. C. Wong, H. C. Teng, and C. Y. Ho, "Localization of mobile robots via an enhanced particle filter incorporating tournament selection and Nelder–Mead simplex search," *Int. J. Innov. Comput. Inf. Control*, vol. 7, no. 7, pp. 3725–3737, 2011.
[24] L. Carlone, N. M. Kaouk, J. Du *et al.*, "Simultaneous localization and mapping using Rao-Blackwellized particle filters in multi-robot systems," *J. Intell. Robot. Syst.*, vol. 63, no. 2, pp. 283–307, 2011.
[25] R. K. Mehra, "On the identification of variances and adaptive Kalman filtering," *IEEE Trans. Autom. Control*, vol. 15, no. 2, pp. 175–184, 1970.
[26] A. Sakai, "PythonRobotics project," https://atsushisakai.github.io, 2018.
[27] E. Nebot, "Victoria Park dataset," [Online]. Available: http://www-personal.acfr.usyd.edu.au/nebot/dataset.htm, 2008.
[28] F. F. Yadkuri and M. J. Khosrowjerdi, "Methods for improving the linearization issue of extended Kalman filter," *J. Intell. Robot. Syst.*, vol. 78, no. 4, pp. 485–497, 2015.
[29] J. Guivant, E. Nebot, and S. Baiker, "Autonomous navigation and map building using laser range sensors in outdoor applications," *J. Robot. Syst.*, vol. 17, no. 10, pp. 565–583, 2000.

Amir Panah received his B.S. degree from Iran University of Science and Technology, Iran in 2008, his M.S. degree from Qazvin Branch, Islamic Azad University, Iran in 2011. His research interests include navigation, Fuzzy Logic, Neural Network, SLAM, mobile robot, rescue robot, software engineering and artificial intelligence.
Email: amir.panah2020@gmail.com

Homayun Motameni received M.S. degree from Science and Research Branch, Islamic Azad University, Tehran, Iran in 1998 and Ph.D. degree from Science and Research Branch, Islamic Azad University, Tehran, Iran in 2007. He is currently a faculty of the department computer engineering at Sari Branch, Islamic Azad University, Sari, Iran. His current research interests include machine learning, software engineering and robotic.
Email: motameni@iausari.ac.ir

Ali Ebrahimnejad received M.S. from Tehran University, Iran and Ph.D. degree from Science and Research Branch, Islamic Azad University, Tehran, Iran. He is currently a faculty of the department of mathematics Islamic Azad University, Qaemshahr Branch, Iran. His research interests include data analysis, fuzzy logic, mathematics and robotic.
Email: a.ebrahimnejad@qaemiau.ac.ir