IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 22, 2018, accepted May 12, 2018, date of publication May 18, 2018, date of current version June 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2838112

A Fast Algorithm of SLAM Based on Combinatorial Interval Filters

JINGWEN LUO[1,2] AND SHIYIN QIN[1]

[1]School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China
[2]School of Information Science and Technology, Yunnan Normal University, Kunming 650500, China

Corresponding authors: Jingwen Luo (by1503117@buaa.edu.cn) and Shiyin Qin (qsy@buaa.edu.cn)

This work was supported in part by the Beijing Science and Technology Planning Project of China under Grant D161104001300000-D161100001316001 and in part by the National Nature Science Foundation of China under Grants 61731001 and U1435220.

ABSTRACT Current FastSLAM algorithms face challenges such as heavy computing requirements and difficulty in enhancing estimation accuracy. This paper presents a fast algorithm of simultaneous localization and mapping (SLAM) based on combinatorial interval filters coupled with an improved box particle filter (IBPF) and extended interval Kalman filter (EIKF). First, strategies for improving box contracting and resampling are studied in depth via the linear programming contractor and dimension selection subdivision resampling methods. Then, we propose a weighted average based on a time-varying Markov model to increase the estimation accuracy of the EIKF. In this way, a kind of fast SLAM algorithm is designed through combinatorial synthetic integration, in which the IBPF algorithm is employed to realize simultaneous localization and the EIKF is utilized to build a map. A series of simulations and experiments demonstrate the superior performance of our interval filters based SLAM algorithm.

INDEX TERMS Interval analysis, SLAM, box particle filter, extended interval Kalman filter, combinatorial interval filters.

I. INTRODUCTION

Simultaneous localization and mapping (SLAM) is the key to achieving truly autonomous navigation in mobile robots. However, simultaneous localization and map building are mutually dependent on each other, which can make accurate solutions difficult and complex to obtain, especially in highly-dimensional space.

The earliest probability-based method for solving SLAM problems originated from Smith and Cheeseman's [1] extended Kalman filter (EKF). With further development, Paskin [2] presented a low-complexity solution to SLAM problems using thin junction trees. In order to solve the problems of high computational and storage requirements, a sparse extended information filtering-based SLAM approach was proposed [3]. After this, an effective solution to SLAM problems using the Rao-Blackwellized particle filter (RBPF) was proposed, called FastSLAM [4]–[7]. Subsequently, many researchers have developed different methods for improving the original FastSLAM algorithm. In [8] and [9], two techniques were presented to improve FastSLAM by providing a better proposal distribution. An integrated technique combining a genetic algorithm (GA) and particle swarm optimization (PSO) within FastSLAM was presented in [10]. A novel FastSLAM algorithm based on the iterated unscented Kalman filter (IUKF) was proposed in [11]. Zhang *et al.* [12], [13] presented a new decomposition technique to improve the accuracy and reliability of FastSLAM. In order to improve consistency and the diversity of particles, Lv *et al.* [14] proposed a new FastSLAM algorithm based on revised genetic resampling and a square root unscented particle filter (SR-UPF). In [15], a more robust FastSLAM approach was achieved by effective improvement with differential evolution based on square root central difference (SRCD).

Unfortunately, these probabilistic methods have the common drawback of consistency problems. Studies [16], [17] show that the RBPF SLAM can obtain accurate positional estimates, but only in a short time to meet the consistency requirements. This drawback can be overcome using interval analysis (IA) methods rather than probabilistic ones [18]. Indeed, IA provides guaranteed and consistent results, and does not suffer from biased measurements. Such advantages

have been highlighted in localization applications [19], [20] and have provided interesting perspectives for SLAM problems [21]–[23].

In recent years, a particle filter (PF) strategy for mobile robot localization involving interval data was introduced in [24], which proposed the concept of box particle filtering (BPF). A result of the synergy between PF and IA, BPF is an approach that is aimed at solving a general class of nonlinear filtering problems. The key idea is to replace a particle with a multidimensional interval or box of non-zero volume in the state space. The approach is particularly appealing in practical situations involving imprecise stochastic measurements that result in very broad posterior densities. Most recently, various applications [25]–[29] have shown that accurate and reliable performance for several thousand particles can be achieved with just a few dozen boxes.

In order to effectively suppress the noise of time-varying system, a state-of-the-art noise-suppressing method was proposed by Jin *et al.* [30], [31]. Furthermore, to handle systems with uncertain dynamics and measurement noise, Chen *et al.* [32] developed a new interval Kalman filter (IKF) algorithm. In some applications [33]–[36], the IKF has demonstrated better results than those of some existing robust KFs. In terms of interval nonlinear systems, Siouris *et al.* [37] successfully applied the EIKF to track an incoming ballistic missile system, then He and Vik [38] used EIKF in an integrated GPS/INS system to solve tracking problems. Thus, alternative techniques for obtaining more selective and robust solutions to SLAM problems are possible within the IA framework.

The rest of the paper is organized as follows. Section II presents the filtering scenarios and improvement strategies for BPF and introduces the EIKF. The scheme for using combinatorial interval filters in SLAM problems, and its implementation process, are presented in Section III. The procedure for using the proposed fast SLAM algorithm and a performance analysis are presented in Section IV. Section V provides typical experimental results and comparative analyses, while Section VI concludes the paper.

II. FROM ORDINARY NUMERICAL FILTER TO INTERVAL FILTER

In this section, we first present a brief description of the elementary concepts of IA. Then, filtering scenarios for BPF are described, based on which improved strategies for box contracting and resampling are studied in detail. Finally, the EIKF algorithm is introduced.

A. ELEMENTARY CONCEPTS ABOUT INTERVAL ANALYSIS

The main concept of IA is to deal with the intervals of real numbers instead of dealing with the real numbers themselves [39]. A real interval $[x] = [\underline{x}, \bar{x}]$ is defined as a closed and connected subset of R with $\underline{x}$ and $\bar{x}$ denoting the lower and the upper bounds of x, respectively. The center (midpoint) and width are denoted as $mid([x]) = (\underline{x} + \bar{x})/2$ and $wid([x]) = (\bar{x} - \underline{x})/2$ separatively. The set of n-dimensional real intervals is denoted by IR^n. For any interval, the elementary interval operations $(+, -, *, \div)$ etc. are defined by

$$[x] \odot [y] = \{x \odot y \,|x \in [x], y \in [y]\} \tag{1}$$

where $\odot$ denotes any binary operations, and assume $0 \notin [y]$ in case of division.

An interval vector, or a box $[\mathrm{x}] \in R^n$, is a Cartesian product of n intervals, which may be represented as follows:

$$[\mathrm{x}] = [x_1] \times [x_2] \times \cdots \times [x_n] = \times_{i=1}^{n} [x_i] \tag{2}$$

In IA, the size of $[\mathrm{x}]$ is denoted as $|[\mathrm{x}]|$. An interval matrix $[X]$ is a matrix with interval components, and the set of $n \times m$ real interval matrices is denoted by $IR^{n\times m}$. The interval arithmetic and related properties of interval can be naturally extended to the context of interval vector and interval matrix. For more information please refer to [39].

Consider a mapping $f : R^n \to R^m$; then, the interval function $[f]$ from IR^n to IR^m is an *inclusion function* as shown in Fig.1. It is obvious that

$$f([\mathrm{x}]) \subset [f]([\mathrm{x}]), \forall [\mathrm{x}] \in IR^n \tag{3}$$

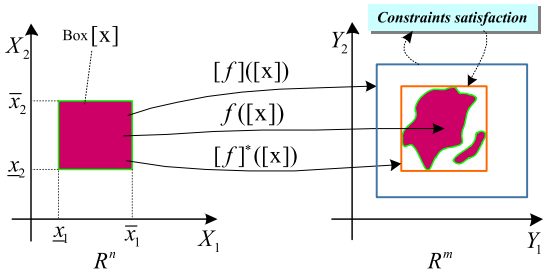

FIGURE 1. Mapping of a box [x] by a vector of function f and its two different inclusion functions $[f]$ and $[f]^*$.

One of the purposes of IA for f is to provide an inclusion function $[f]$ which can be evaluated reasonably such that a proper size of $[f]([\mathrm{x}])$ is achieved. Therefore, we need to solve the CSP commonly expressed as follows:

$$H : (\mathrm{g(x)} = 0, \mathrm{x} \in [\mathrm{x}]) \tag{4}$$

In fact, the connotation of (4) is to find the optimal box enclosure of the set of vectors $\mathrm{x} = (\mathrm{x}_1, \mathrm{x}_2, \cdots, \mathrm{x}_n)$ belonging to a given prior domain $[\mathrm{x}]$ satisfying a set of constraints $\mathrm{g(x)} = (g_1(\mathrm{x}), g_2(\mathrm{x}), \cdots g_m(\mathrm{x}))^{\mathrm{T}}$ for various real function $g_i(\mathrm{x})$, $i = 1, \ldots, m$. The solution set S consists of all of the values of x satisfying $\mathrm{g(x)} = 0$, and can be denoted as follows:

$$S = \{\mathrm{x} \in [\mathrm{x}] \,|\mathrm{g(x)} = 0\} \tag{5}$$

A *contractor* for H is any operator that can be used to contract H, i.e. replacing $[\mathrm{x}]$ by a smaller domain $[\mathrm{x}]'$, such that $S \subseteq [\mathrm{x}]' \subseteq [\mathrm{x}]$.

Various constraints methods named contractors are described in [40], including Gauss elimination, the Krawczyk method, and forward-backward propagation, etc. Each of these methods can be suitable for different types of CSP. It is

remarkable that the consistency conditions be satisfied for contractor, namely, *global consistency* and *local consistency*. *Global consistency* represents the ideal solution for CSP and it is stronger than *local consistency*. However, for most CSPs, the existing methods can only reach *local consistency*.

B. BOX PARTICLE FILTER (BPF) AND ITS IMPROVED STRATEGY

BPF is a nonlinear filtering algorithm which couples a sequential Monte Carlo method and IA. The key idea is to use box particles and a bounded error model, instead of discrete point particles and probabilistic models for the errors and the inputs. Details of the BPF algorithm are provided in [24] and [41]. According to [42], the BPF can be considered an approximation of the Bayesian filter by interpreting each box particle as a uniform probability density function (PDF). In this way, the set of box particles may be interpreted as a mixture of uniform PDFs. Below, we focus on the main steps of Bayesian justification.

In terms of IA, for the sake of quantifying uncertainties, the state vector and the measurement vector become the vectors of intervals. The propagation and the observation functions become inclusion functions, denoted as $[f]$ and $[h]$, respectively. Thus, the system dynamics equations can be defined as follows:

$$\begin{cases} [\mathrm{x}_{k+1}] = [f]([\mathrm{x}_k], [\mathrm{u}_{k+1}]) \\ [\mathrm{z}_{k+1}] = [h]([\mathrm{x}_{k+1}]) \end{cases} \tag{6}$$

where $[\mathrm{x}_k] \in IR^{n_x}$ and $[\mathrm{z}_k] \in IR^{n_y}$ are the state interval vector and the measurement interval vector at time step k, respectively. The control vector $[\mathrm{u}_k]$ is deduced from the proprioceptive sensor data.

1) PREDICTION

Let $U_{[\mathrm{x}^i]}(\mathrm{x})$ denotes a uniform PDF for $[\mathrm{x}^i]$ as its support set in the box $[\mathrm{x}]$, if the current PDF is written as

$$p(\mathrm{x}_k \,|\mathrm{z}_{1,k}) = \sum\nolimits_{i=1}^{N} \omega_k^i U_{[\mathrm{x}_k^i]}(\mathrm{x}_k) \tag{7}$$

where N denotes the number of boxes, the ω_i is a normalized weight for $i = 1, 2, \cdots, N$ and $\sum_{i=1}^{N} \omega_i = 1, \forall i, \omega_i \geq 0$. Then, the next step updating of PDF may be expressed as follows:

$$\begin{aligned} p(\mathrm{x}_{k+1}|\mathrm{z}_{1,k}) &= \int p(\mathrm{x}_{k+1}|\mathrm{x}_k) p(\mathrm{x}_k|\mathrm{z}_{1,k}) d\mathrm{x}_k \\ &= \int p(\mathrm{x}_{k+1}|\mathrm{x}_k) \sum\nolimits_{i=1}^{N} \omega_k^i U_{[\mathrm{x}_k^i]}(\mathrm{x}_k) d\mathrm{x}_k \\ &= \sum\nolimits_{i=1}^{N} \omega_k^i \int_{[\mathrm{x}_k^i]} p(\mathrm{x}_{k+1} \,|\mathrm{x}_k) U_{[\mathrm{x}_k^i]}(\mathrm{x}_k) d\mathrm{x}_k \end{aligned} \tag{8}$$

Assume that the noise v_k at time $k+1$ is bounded in the box $[\mathrm{v}_k]$. Consider an inclusion function $[f], \forall i = 1 \cdots N$, if $x_k \in [\mathrm{x}_k^i]$, then, $x_{k+1} \in [f]([\mathrm{x}_k^i] + [\mathrm{v}_k])$. Thus,

$$p(\mathrm{x}_{k+1} \,|\mathrm{x}_k) U_{[\mathrm{x}_k^i]}(\mathrm{x}_k) = 0, \quad \forall \mathrm{x}_{k+1} \notin [f]([\mathrm{x}_k^i] + [\mathrm{v}_k]) \tag{9}$$

The connotation of (9) is that, for any $[f]$, the support for the PDF terms $\int_{[\mathrm{x}_k^i]} p(\mathrm{x}_{k+1} \,|\mathrm{x}_k) U_{[\mathrm{x}_k^i]}(\mathrm{x}_k) d\mathrm{x}_k$ can be approximated by $[f]([\mathrm{x}_k^i] + [\mathrm{v}_k])$, i.e.

$$\int_{[\mathrm{x}_k^i]} p(\mathrm{x}_{k+1} \,|\mathrm{x}_k) U_{[\mathrm{x}_k^i]}(\mathrm{x}_k) d\mathrm{x}_k \approx [f]([\mathrm{x}_k^i], [\mathrm{v}_k]) \tag{10}$$

According to (8) and (10), the predictive distribution could be expressed as follows:

$$\begin{aligned} p(\mathrm{x}_{k+1} \,|\mathrm{z}_{1,k}) &\approx \sum\nolimits_{i=1}^{N} \omega_k^i U_{[f]([\mathrm{x}_k^i],[\mathrm{v}_k])}(\mathrm{x}_{k+1}) \\ &= \sum\nolimits_{i=1}^{N} \omega_k^i U_{[\mathrm{x}_{k+1}^i]}(\mathrm{x}_{k+1}) \end{aligned} \tag{11}$$

Equation (11) shows that the prediction PDF $p(\mathrm{x}_{k+1} \,|\mathrm{z}_{1,k})$ can be approximated using the weighted sum of N uniform PDFs with $[\mathrm{x}_{k+1|k}^i]$ as the support.

2) CORRECTION

The measurement likelihood function for the BPF is taken to be one component with uniform distribution [42]. A probabilistic model P_w of the bounded measurement noise w_k can be expressed by a single uniform PDF. Such that the box measurement $[\mathrm{z}_{k+1}]$ contains all realisations of $\mathrm{z}_{k+1} = h(\mathrm{x}_{k+1}) + w_k$. Thus, we obtained the following:

$$p(\mathrm{z}_{k+1}|\mathrm{x}_{k+1}) = U_{[\mathrm{z}_{k+1}]}(h(\mathrm{x}_{k+1})) \tag{12}$$

According to (11) and (12), the measurement update step can be performed as follows:

$$\begin{aligned} &p(\mathrm{x}_{k+1}|\mathrm{z}_{1,k+1}) \\ &= \frac{1}{\alpha_{k+1}} p(\mathrm{z}_{k+1}|\mathrm{x}_{k+1}) p(\mathrm{x}_{k+1}|\mathrm{z}_{1,k}) \\ &= \frac{1}{\alpha_{k+1}} \sum\nolimits_{i=1}^{N} \omega_k^i \underbrace{U_{[\mathrm{z}_{k+1}]}(h(\mathrm{x}_{k+1})) U_{[\mathrm{x}_{k+1|k}^i]}(\mathrm{x}_{k+1})}_{\Psi_i} \end{aligned} \tag{13}$$

where $\alpha_{k+1} = \int p(\mathrm{z}_{k+1}|\mathrm{x}_{k+1}) \, p(\mathrm{x}_{k+1}|\mathrm{z}_{1,k}) d\mathrm{x}_{k+1}$ is a normalized coefficient, each of the terms Ψ_i is a constant function and with a support being the set

$$SS_{\psi_i} = \{\mathrm{x}_{k+1} \in [\mathrm{x}_{k+1}^i] \,| h(\mathrm{x}_{k+1}) \in [\mathrm{z}_{k+1}]\} \subset R^{nx} \tag{14}$$

In fact, above set defines a CSP; that is, the predicted box particle $[\mathrm{x}_{k+1|k}^i]$ can be contracted with respect to the relationship between the measurement function h and the measurement $[\mathrm{z}_{k+1}]$. Therefore, a set of new boxes $[\hat{x}_{k+1}^i]$ may be provided to fit $p(\mathrm{z}_{k+1} \,|\mathrm{x}_{k+1})$ after the CSP. According to the contractility and completeness of *contractor* [40], we obtain the following:

$$\begin{aligned} &U_{[\mathrm{z}_{k+1}]}(h(\mathrm{x}_{k+1})) \left|[\mathrm{x}_{k+1|k}^i]\right| U_{[\mathrm{x}_{k+1|k}^i]}(\mathrm{x}_{k+1}) \\ &= U_{[\mathrm{z}_{k+1}]}(h(\mathrm{x}_{k+1})) \left|SS_{\Psi_i}\right| U_{ss_{\Psi_i}}(\mathrm{x}_{k+1}) \\ &\Rightarrow U_{[\mathrm{z}_{k+1}]}(h(\mathrm{x}_{k+1})) U_{[\mathrm{x}_{k+1|k}^i]}(\mathrm{x}_{k+1}) \\ &= U_{[\mathrm{z}_{k+1}]}(h(\mathrm{x}_{k+1})) \frac{1}{\left|[\mathrm{x}_{k+1|k}^i]\right|} \left|SS_{\Psi_i}\right| U_{ss_{\Psi_i}}(\mathrm{x}_{k+1}) \end{aligned} \tag{15}$$

Combining (13) with (15), we defined $[\hat{x}^i_{k+1}]$ as the smallest box containing SS_{Ψ_i}, that is, $[\hat{x}^i_{k+1}] = [SS_{\Psi_i}]$. Thus,

$$p(x_{k+1}|z_{1,k+1}) = \frac{1}{\alpha_{k+1}} \sum_{i=1}^{N} \omega_k^i \frac{1}{|[z_{k+1}]|} \frac{1}{\left|[x^i_{k+1}]\right|} \times \left|SS_{\Psi_i}\right| U_{SS_{\Psi_i}}(x_{k+1}) \propto \frac{1}{\alpha_{k+1}} \sum_{i=1}^{N} \omega_k^i \frac{\left|[\hat{x}^i_{k+1}]\right|}{\left|[x^i_{k+1}]\right|} U_{|[x^i_{k+1}]|}(x_{k+1}) \tag{16}$$

After contracted, each box particle $[x^i_{k+1}]$ turns into $[\hat{x}^i_{k+1}]$. According to (16), the weight is updated by

$$L_k^i = \prod_{j=1}^{p} L_k^i(j), \quad L_k^i(j) = \frac{\left|[\hat{x}^i_{k+1}(j)]\right|}{\left|[x^i_{k+1|k}(j)]\right|} \tag{17}$$

where p represents the dimension of state. Therefore, the posterior distribution $p(x_{k+1}|z_{1,k+1})$ is approximated by $\{\hat{\omega}^i_{k+1}, [\hat{x}^i_{k+1}]\}, i = 1, \cdots, N$, and $\hat{\omega}^i_{k+1} \propto \omega^i_k L^i_k$.

3) RESAMPLING

Compute the effective sample size $N_{eff} = 1/\sum_{i=1}^{N}(\omega_k^i)^2$ and choose a threshold N_{th}. In case of $N_{eff} < N_{th}$ perform resampling using a modified version of the sequential importance resampling (SIR) algorithm, in which the uncertainty regions of the selected box particles are reduced to randomly selected portions of the initial uncertainty regions.

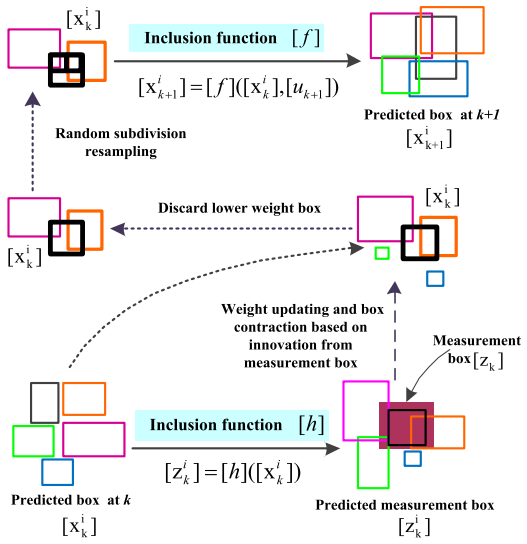

FIGURE 2. Scenarios for the box particle filter.

Although the BPF algorithm has been proven to be efficient in various applications [44], the filtering results presented in [41] are not accurate enough in the context of high nonlinearity and highly ambiguous measurements. As we can see in Fig. 2, box contracting and resampling are vital throughout the process. Thus, our proposed improved BPF (IBPF) schemes mainly involve the following two aspects.

a: LINEAR PROGRAMMING (LP) CONTRACTOR

The original BPF employs the constraint propagation (CP) technique (i.e. forward-backward propagation) to contract boxes [24]. However, a well-known drawback of CP is that decomposition into primitive constraints introduces new variables in the CSP. This hinders efficient domain tightening. Meanwhile, the main limitation of CP is its sensitivity to the multiple occurrence of variables [45]. For some applications, it is only possible to contract box particles for two-dimensional cases [46]. Furthermore, the results are by no means satisfactory as the constraints are accounted for in an arbitrary order [47], and only provide *locally consistent*domains [28]. Below, we will present a strategy, named *linear programming contractor* (*LP-contractor*) that appears to us to be more efficient than others.

According to the observation equation and real measurement z, we can construct the following function

$$g(x) = h(x) - z \tag{18}$$

Then, the function vector g(x) can be bracketed over [x] by the following constraint

$$Ax + \underline{b} \le g(x) \le Ax + \overline{b} \tag{19}$$

Provide that g is differentiable,$x_0 = mid([x])$, and $\eta \in [x]$. Using the *mean-value theorem*,

$$g(x) = g(x_0) + J_g(\eta)(x - x_0) \tag{20}$$

In order to linearize g(x) into the form of (19), we may consider the following two algebraic transformations

a) $g(x) = g(x_0) + J_g(\eta)(x - x_0) + J_g(x_0)x - J_g(x_0)x$

$$\Rightarrow g(x) = \underbrace{J_g(x_0)}_{A} x + \underbrace{g(x_0) - J_g(\eta)x_0 + (J_g(\eta) - J_g(x_0))x}_{b} \tag{21}$$

b) $g(x) = g(x_0) + J_g(\eta)(x - x_0) + J_g(x_0)(x - x_0) - J_g(x_0)(x - x_0)$

$$\Rightarrow g(x) = \underbrace{J_g(x_0)}_{A} x + \underbrace{g(x_0) - J_g(x_0)x_0 + (J_g(\eta) - J_g(x_0))(x - x_0)}_{b} \tag{22}$$

According to IA, we find that the number of time interval variables appearing in (22) is less than in (21), so (22) provides tighter intervals. The CSP, therefore, can be approximated as:

$$H : (g(x) = 0, x \in [x]) \tag{23}$$

where $g(x) = Ax - b$, $[b] = Ax_0 - g(x_0) - (A - [J_g]([x]))([x] - x_0)$, $A = J_g(x_0)$, and $b \in [b]$.

The vector $x \in [x]$ is consistent with H if and only if there exists $b \in [b]$, such that $Ax - b = 0$, i.e.

$$Ax - b = 0 \Leftrightarrow Ax \in [b] \Leftrightarrow Ax \ge b \text{ and } Ax \le \bar{b} \tag{24}$$

Consequently, the smallest box $[x]'$ containing all the vectors x that are consistent with H can be computed by solving the following $2n$ linear programming

problems.

$$\begin{aligned} &Opt\ \mathrm{x}_i, \quad i = 1, \ldots, n \\ &s.t. \begin{pmatrix} -\mathrm{A} \\ \mathrm{A} \end{pmatrix} \mathrm{x} \leq \begin{pmatrix} -\underline{\mathrm{b}} \\ \overline{\mathrm{b}} \end{pmatrix} \ and\ \mathrm{x} \in [\mathrm{x}] \end{aligned} \tag{25}$$

where operator *Opt* is alternatively min and max to obtain *Inf* x_i and *Sup* x_i that hold the corresponding constraint. It is clear that the outstanding advantage of *LP-contractor* lies in its provision of a *globally consistent* feasible domain for CSP by simultaneously processing all constraints.

b: DIRECTION SELECTION SUBDIVISION (DSS) RESAMPLING

The original BPF utilizes multinomial resampling, and then box particles with high weights are subdivided by the corresponding number of realizations along a randomly selective dimension, which obtains smaller box particles around the regions with high likelihoods [24]. The filtering precision of each experiment is different due to the randomness of the subdivision process. In the bounded error areas, the choice of the number of divisions for each dimension is not optimal and remains a subject of research [48]. Thus, a key question involves subdivision, i.e., how to choose the divided dimension. In order to select the divided dimension more efficiently, Merlinge *et al.* [27] presented a deterministic choice of resampling dimension for each box particle. This is done by normalizing the box diameter vector to select the longest dimension along the box particle. Inspired by Rule C, which was investigated in global optimization problems by Ratz and Csendes [49], we propose, here, a dimension selection subdivision resampling method (*DSS-resampling*) for the IBPF algorithm.

Consider the following global optimization problem

$$\min \mathrm{g}(\mathrm{x}) \tag{26}$$

subject to $\mathrm{g}(\mathrm{x}) = h(\mathrm{x}) - \mathrm{z}$ and $\mathrm{x} \in [\mathrm{x}]$.

In order to provide reliable and guaranteed solutions for (26) within the interval framework, interval subdivision methods can be utilized [50]. The criteria for interval subdivision dimension selection rules were defined by [51], namely, each rule selects a dimension k by using a merit function

$$k = \min \left\{ j \,\middle|\, j \in \{1, 2, \cdots, n\}\ and\ D(j) = \max_{i=1}^{n} D(i) \right\} \tag{27}$$

where $D(i)$ is determined by the given rule.

Assume that the inclusion functions of g(x) and its gradient are available [52]. These inclusion functions are therefore used to compute bounds for g(x) on [x]. In our work, the main idea is to minimize the width of the inclusion function, i.e.

$$\begin{aligned} wid([g]([\mathrm{x}])) &= wid\left([g]([\mathrm{x}]) - [g](mid([\mathrm{x}]))\right) \\ &\xrightarrow{\text{mean-value theorem}} \approx wid\left(\nabla[g]([\mathrm{x}]) \cdot ([\mathrm{x}] - mid([\mathrm{x}]))\right) \\ &= wid\left(\sum_{i=1}^{n} \frac{\partial [g]([\mathrm{x}])}{\partial \mathrm{x}_i} \cdot ([\mathrm{x}_i] - mid([\mathrm{x}_i]))\right) \end{aligned} \tag{28}$$

where the inclusion function of the gradient of $g(\mathrm{x})$ is denoted as $\nabla[g]([\mathrm{x}])$. Thus, a significant rule may be determined as:

$$D(i) = \sum_{i=1}^{n} wid\left(\frac{\partial [g]([\mathrm{x}])}{\partial \mathrm{x}_i} \cdot ([\mathrm{x}_i] - mid([\mathrm{x}_i]))\right) \tag{29}$$

That is, the i^{th} component is to be chosen for which $D(i)$ is the largest. Usually, the rules do not specify a certain coordinate direction if the maximum is achieved several times, so we take the smallest one. When selection of the interval dimension is finished, the selection will be subdivided by the corresponding number of realizations in the resampling stage.

C. EXTENDED INTERVAL KALMAN FILTER

The standard KF is no longer applicable when the system is a linear interval system. Chen *et al.* [32] developed a new IKF algorithm based on a well-defined interval conditional mathematical expectation formula. The algorithm has the same recursive structure as the standard KF. As for the nonlinear interval system, the EIKF algorithm can be established based on the IKF and EKF. In what follows, to simplify the notation in EIKF, the superscript I is used to represent the interval, i.e. by denoting $[\mathrm{x}_k]$ as x_k^I, and $[f]$ as f^I, etc. The recursive fashion of EIKF is as follows:

Interval Jacobian matrix $A_k^I = \frac{\partial f_k^I}{\partial \mathrm{x}_k}(\hat{x}_k^I)$, $C_{k+1}^I = \frac{\partial h_{k+1}^I}{\partial \mathrm{x}_{k+1}}(\hat{x}_{k+1,k}^I)$

1) Initial values

$$\hat{x}_0^I = E\{\mathrm{x}_0^I\}, \quad \Sigma_0^I = Cov\{\mathrm{x}_0^I\} \tag{30}$$

2) Prediction

$$\hat{x}_{k+1,k}^I = f_k^I(\hat{x}_k^I) \tag{31}$$

$$\Sigma_{k+1,k}^I = A_k^I \Sigma_k^I (A_k^I)^T + \mathbb{Q}_k^I \tag{32}$$

3) Interval Kalman gain K_k^I and innovation r_k^I

$$K_{k+1}^I = \Sigma_{k+1,k}^I (C_{k+1}^I)^T (C_{k+1}^I \Sigma_{k+1,k}^I (C_{k+1}^I)^T + \mathbb{R}_{k+1}^I)^{-1} \tag{33}$$

$$\mathrm{r}_{k+1}^I = \mathrm{z}_{k+1}^I - h_{k+1}^I(\hat{x}_{k+1,k}^I) \tag{34}$$

4) Updating

$$\hat{x}_{k+1}^I = \hat{x}_{k+1,k}^I + K_{k+1}^I \mathrm{r}_{k+1}^I \tag{35}$$

$$\Sigma_{k+1}^I = (I - K_{k+1}^I C_k^I)\Sigma_{k+1,k}^I \quad k = 1, 2, \ldots \tag{36}$$

where the state transition and observation inclusion functions are denoted as f_k^I and h_k^I, respectively. z_k^I is real measurement. Σ_k^I is the interval covariance matrix. $\mathbb{Q}_k^I$ and $\mathbb{R}_k^I$ are the system noise interval covariance matrix and measurement noise interval covariance matrix separatively.

It should be noted that the interval measurement vector z_k^I shown in the IKF and EIKF is an uncertain interval vector before the data is actually obtained, but is an ordinary constant vector after it has been realized and obtained. The filtering scenario of the EIKF for a 2-dimensional case

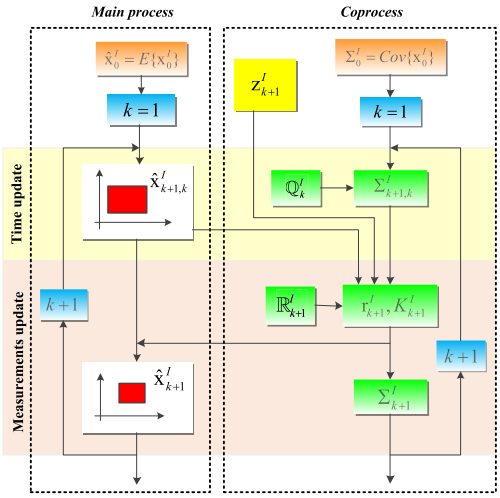

FIGURE 3. Scenarios for the extended interval Kalman filter.

TABLE 1. Calculation of interval inverse matrix.

Algorithm 1: Calculation of Interval Inverse Matrix

Input: $B^I = [(\underline{b}_{ij}, \overline{b}_{ij})]$

Output: $(B^I)^{-1}$

1. Calculate *center matrix* $B_{mid} = [(\underline{b}_{ij} + \overline{b}_{ij})/2]$ and $(B_{mid})^{-1}$.
2. Calculate $E^I = [(\underline{e}_{ij}, \overline{e}_{ij})] = I - B^I B_{mid}^{-1}$ and norm $\|E^I\| = \max_{1\le i\le n} \sum_{j=1}^{n} (|\underline{e}_{ij}|, |\overline{e}_{ij}|)$.
3. *IF* $\|E^I\| \ge 1$, *THEN*, calculate the *upper-boundary matrix* B_s^I and $(B_s^I)^{-1}$, $\Rightarrow (B^I)^{-1} \approx (B_s^I)^{-1}$.
   ELSE, define *interval matrix* Γ^I, each element of Γ^I is $[-\rho, \rho]$ and satisfy $\rho > -\|E\|^{m+1}(1 - \|E\|)$,
   $\Rightarrow (B^I)^{-1} \approx B_{mid}^{-1}(E^I + \Gamma^I) + B_{mid}^{-1}$. *ENDIF*
4. End

is given by Fig. 3. Remarkably, the iterative formula of EIKF contains the calculation of interval matrix inversion, and the Hansen algorithm method [53] can be adopted. In order to reduce the computational requirements, improve the real-time performance and avoid some cases of interval inverse matrix cannot be calculated by the Hansen algorithm. Zhengang [54] proposed a simple and feasible scheme for the calculation of the inverse matrix, which combines the Hansen algorithm with the upper boundary matrix inverse of the interval matrix. We summarize the algorithm as shown in Table 1.

III. SCHEME FOR USING COMBINATORIAL INTERVAL FILTERS TO ACHIEVE SLAM

The aim of this work is to use IBPF to realize simultaneous localization and utilize EIKF to complete map learning. In this way, our approach replaces an ordinary numerical filter with an interval filter within the traditional FastSLAM framework. The combined approach relies on its performance and could provide a better solution for SLAM implementation.

A. SCHEME OF SLAM BASED ON COMBINATORIAL SYNTHETIC INTEGRATION

FastSLAM is a framework using an RBPF, which is based on the following factorization [55]

$$\underbrace{p(x_{1:k}, M \,|z_{1:k}\,, u_{1:k}, n_{1:k})}_{\text{SLAM posterior}} = \underbrace{p(x_{1:k} \,|z_{1:k}\,, u_{1:k-1}, n_{1:k})}_{\text{trajectory posterior}} \cdot \underbrace{\prod_{i=1}^{L} p(m_i \,|x_{1:k}\,, z_{1:k}, n_{1:k})}_{\text{landmark posterior}} \tag{37}$$

where $x_{1:k}$, $z_{1:k}$, $u_{1:k}$ and $n_{1:k}$ are the robot trajectory, observations, controls, and correspondences, respectively, from the start to time k; m_i is a local map of the i^{th} particle and M is a global map. Thus, the posterior probability $p(x_{1:k} \,|z_{1:k}\,, u_{1:k-1}, n_{1:k})$ about the potential trajectory was solved applying PF, which implied that one particle would represent one potential trajectory over one time step while generating its own map. The posterior $\prod_{i=1}^{L} p(m_i \,|x_{1:k}\,, z_{1:k}, n_{1:k})$ was computed analytically when given information on $x_{1:k}$ and $z_{1:k}$.

Under the FastSLAM framework, we attempt to employ IBPF instead of PF to implement simultaneous localization, and EIKF instead of EKF to complete map building. This can be called IBPF-EIKF SLAM for short, and its full scheme is illustrated in Fig. 4. The method is to associate the robot's position with a set of boxes (multi-dimensional intervals) covering a guaranteed area where all acceptable positions definitely exist. Each box has the following form:

$$[x_k^i] = \left\langle \{([x],[y],[\theta])^T\}_k^i, (\hat{m}_1^I, \textstyle\sum_1^I)_k^i, (\hat{m}_2^I, \textstyle\sum_2^I)_k^i, \ldots, (\hat{m}_l^I, \textstyle\sum_l^I)_k^i \right\rangle \tag{38}$$

where the superscript i indicates the index of the box particle. $\{([x], [y], [\theta])^T\}_k^i$ represents the estimated pose of the robot at time k, which consists of a position and heading. $(\hat{m}_j^I)_k^i$ is the positional estimate of thej^{th}landmark conditioned on the i^{th} box particle, $j = 1, 2, \ldots, l$, and the covariance is described by $\sum_j^I$. The EIKF uses a two-dimensional model to describe the landmark feature: $\mathrm{m}_j = (m_{xj}, m_{yj})$. $[\mathrm{z}_k]$ is the real measurement provided by laser radar. As usual, the total number of particles is denoted as N.

In above scheme of combinatorial synthetic integration, the posterior probability at time k is calculated from the one at time $k-1$ and innovation of real measurement, in which a new box-particle set $[\mathrm{x}_k]$ is generated from the box-particle

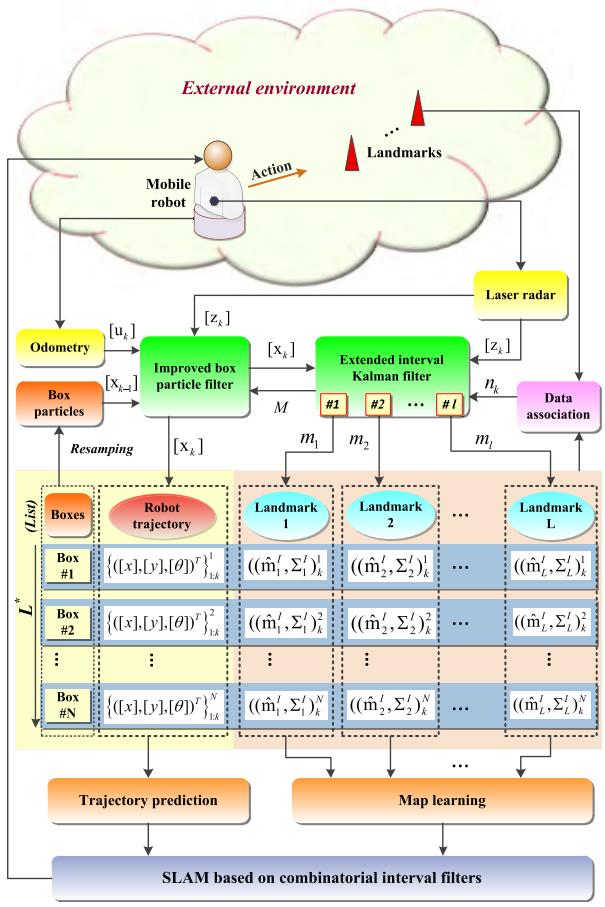

FIGURE 4. Scheme of SLAM based on combinatorial synthetic integration.

set at one time-step earlier, $[\mathrm{x}_{k-1}]$. This new box-particle set incorporates a new control $[\mathrm{u}_k]$ and a measurement $[z_k]$ with associated correspondence n_k. First, the trajectory posterior is extended by sampling new poses, namely, using the control $[\mathrm{u}_k]$ to sample the new robot pose $\{([x], [y], [\theta])^T\}_k^i$ for each box particle in $[\mathrm{x}_{k-1}]$. The resulting sample $[\mathrm{x}_k^i]$ for the i^{th} box particle is then added to a temporary set of box particles, along with the trajectory of previous poses. Then, according to the new measurement $[z_k]$, the EIKF updates the observed feature estimate $(\hat{m}_j^I, \sum_j^I)_k^i$. The updated values are then added to the temporary box-particle set, along with the new pose. Finally, this set of box particles is resampled, and N box particles are drawn (with replacement) from the temporary set according to importance weight. The resulting set of N box particles then forms the new and final box-particle set $[\mathrm{x}_k]$.

During resampling, successive subdivision of the feasible box actually corresponds to branch-and-bound methods [49]. Thus, we need to manage a list L^* which contains intervals whose union includes all states of the considered SLAM problem. In addition, some information on the intervals can be stored in L^* or can be expressed implicitly by the ordering of the list elements. Then, the information will be used to decide which interval is to be chosen.

B. SYNCHRONOUS LOCATION WITH IMPROVED BOX PARTICLE FILTER

The time-discrete kinematic model based on intervals of the robot's displacement is formulated as follows:

$$\begin{cases} [x_{k+1}] = [x_k] + [\Delta d_k] \cos\left([\theta_k] + 1/2[\Delta\theta_k]\right) \\ [y_{k+1}] = [y_k] + [\Delta d_k] \sin\left([\theta_k] + 1/2[\Delta\theta_k]\right) \\ [\theta_{k+1}] = [\theta_k] + [\Delta\theta_k] \end{cases} \tag{39}$$

where input vector $[\mathrm{u}_k] = ([\Delta d_k][\Delta\theta_k])^T$ consists of the elementary displacement and the elementary rotation of the mobile robot at time k. $([x] \times [y])^T$ stands for the position of robot in the global coordinate system, and $[\theta]$ is its orientation with respect to the x axis. More details and elucidation of parameters are presented in [55].

The measurement vector $[z_k] = ([r_k][\phi_k])^T$ is composed of a range of measurements r and bearing measurements ϕ, which are respectively modeled as follows:

$$\begin{cases} [r] = \sqrt{([m_x] - [x_k])^2 + ([m_y] - [y_k])^2} \\ [\varphi] = \arctan(\frac{[m_y]-[y_k]}{[m_x]-[x_k]}) - [\theta_k] \end{cases} \tag{40}$$

In terms of the interval-based uncertainties, the laser range finder provides interval measurements that are expressed as $[z_k] = [z_k - 3\sigma, z_k + 3\sigma]$, where σ denotes the standard deviation.

Assume that at time step k, the box-particle set is $\{\omega_k^i, [\mathrm{x}_k^i]\}, i = 1, \cdots, N$, and has control input $[\mathrm{u}_k]$. The pose estimation is to be performed as

$$[\mathrm{x}_{k+1}^i] = P([\mathrm{x}_{k+1}^i] \,|\, [\mathrm{x}_k^i], [u_k]) \approx [f]([\mathrm{x}_k^i], [u_k]) \tag{41}$$

For each box particle, the predicted box measurement is obtained by

$$[z_{k+1}^i] = P([z_{k+1}^i] \,|\, [\mathrm{x}_{k+1}^i]) = [h]([\mathrm{x}_{k+1}^i]) \tag{42}$$

With the real measurement box $[z_{k+1}]$ obtained from the external sensor, the innovation is given by

$$[\mathrm{r}_{k+1}^i] = [z_{k+1}^i] \cap [z_{k+1}] \tag{43}$$

If $[\mathrm{r}_{k+1}^i] = \phi$, this box particle for which $[z_{k+1}^i]$ has no intersection with $[z_{k+1}]$, should be penalized; if $[\mathrm{r}_{k+1}^i] \neq \phi$, this box particle, for which the predicted value is included in the $[z_{k+1}]$ should be favored. In order to eliminate the inconsistent part of the box particles with respect to $[z_{k+1}]$, *LP-contractor*is used to preserve appropriate box sizes. Thus, the likelihood $P([z_{k+1}] \,|\, [\mathrm{x}_{k+1}^i])$ of the i^{th} box can be calculated as

$$L^i = P([z_{k+1}] \,|\, [\mathrm{x}_{k+1}^i]) = \prod_1^p L^i(j), \quad L^i(j) = \frac{|[\mathrm{r}_{k+1}^i(j)]|}{|[z_{k+1}^i(j)]|} \tag{44}$$

where $|[\mathrm{r}_{k+1}^i(j)]|$ is the length of the innovation for the i^{th} box particle and the j^{th} measurement component, and p

is the dimension of the measurement. After calculating the likelihood, the weight updating is performed according to

$$\omega_{k+1}^{i} = P([z_{k+1}] \left| [\mathrm{x}_{k+1}^{i}]) \omega_{k}^{i} = L^{i} \omega_{k}^{i} \right. \tag{45}$$

Normalization is performed as

$$\omega_{k+1}^{i} = \frac{\omega_{k+1}^{i}}{\sum_{j=1}^{N} \omega_{k+1}^{j}} \tag{46}$$

At time step $k+1$, the state can be extracted by

$$\hat{x}_{k+1} = \sum\nolimits_{i=1}^{N} \omega_{k+1}^{i} mid([\hat{x}_{k+1}]) \tag{47}$$

We might also use a maximum weight estimate, i.e. where the state estimate is the center of the box particle with the larger weight. Given that the BPF estimation $\hat{x}_{k+1}$ is calculated using N vectors $mid([\mathrm{x}_{k+1}^{i}])$, another confidence in the estimation based on the confidence of each $mid([\mathrm{x}_{k+1}^{i}])$ can be calculated by using a Gaussian-like mixture strategy with

$$\begin{aligned} \hat{P}_{k+1} = \sum\nolimits_{i=1}^{N} \omega_{k+1}^{i} ((\left| [\mathrm{x}_{k+1}^{i}] \right| /2) \\ + (\hat{x}_{k+1} - mid([\hat{\mathrm{x}}_{k+1}^{i}]))(\hat{x}_{k+1} - mid([\hat{\mathrm{x}}_{k+1}^{i}]))^{T}) \end{aligned} \tag{48}$$

where $\left| [\mathrm{x}_{k+1}^{i}] \right| /2$ is half width of $[\mathrm{x}_{k+1}^{i}]$ and represents the partial confidence generated when using $mid([\mathrm{x}_{k+1}^{i}])$.

C. MAP BUILDING BASED ON AN EXTENDED INTERVAL KALMAN FILTER

Since the estimations between features are independent of each other, estimation updating for a landmark at time $k+1$ depends on whether the landmark is observed at time k. Therefore, we consider the following cases

1) If a landmark is newly observed, its mean and covariance are initialized as follows:

$$(\hat{\mathrm{m}}_{new}^{I})_{k+1}^{i} = (h^{I})^{-1} \left(\mathrm{z}_{k+1}^{I}, (\mathrm{x}_{k+1}^{I})^{i} \right) \tag{49}$$

$$C_{k+1}^{I} = \frac{\partial h_{k+1}^{I}}{\partial \mathrm{m}_{k+1}^{I}} \left((\hat{x}_{k+1}^{I})^{i}, (\hat{\mathrm{m}}_{new}^{I})_{k+1}^{i} \right) \tag{50}$$

$$(\Sigma_{new}^{I})_{k+1}^{i} = \left((C_{k+1}^{I})^{-1} \right)^{T} \mathbb{R}_{k+1}^{I} (C_{k+1}^{I})^{-1} \tag{51}$$

2) If the latest observation data contains the j^{th} landmark feature in the map, then the mean and covariance of the landmark are updated by the EIKF as follows:

$$(\hat{\mathrm{z}}_{k+1}^{I})^{\mathrm{i}} = h^{I} \left((\hat{\mathrm{m}}_{j}^{I})_{k}^{i}, (\hat{x}_{k+1}^{I})^{i} \right) \tag{52}$$

$$(C_{k+1}^{I})^{i} = \frac{\partial h_{k+1}^{I}}{\partial \mathrm{x}_{k+1}^{I}} \left((\hat{x}_{k+1}^{I})^{i} \right) \tag{53}$$

$$\begin{aligned} (K_{k+1}^{I})^{i} = (\Sigma_{j}^{I})_{k}^{i} \left((C_{k+1}^{I})^{i} \right)^{T} \\ \times \left((C_{k+1}^{I})^{i} (\Sigma_{j}^{I})_{k}^{i} ((C_{k+1}^{I})^{i})^{T} + \mathbb{R}_{k+1}^{I} \right)^{-1} \end{aligned} \tag{54}$$

$$(\mathrm{r}_{k+1}^{I})^{i} = \mathrm{z}_{k+1}^{I} - (\hat{\mathrm{z}}_{k+1}^{I})^{i} \tag{55}$$

$$(\hat{\mathrm{m}}_{j}^{I})_{k+1}^{i} = (\hat{\mathrm{m}}_{j}^{I})_{k}^{i} + (K_{k+1}^{I})^{i} (\mathrm{r}_{k+1}^{I})^{i} \tag{56}$$

$$(\Sigma_{j}^{I})_{k+1}^{i} = \left(I - (K_{k+1}^{I})^{i} (C_{k+1}^{I})^{i} \right) (\Sigma_{j}^{I})_{k}^{i} \tag{57}$$

3) If the latest observation data does not contain the j^{th} landmark feature in the map, then the estimation remains unchanged, i.e.

$$(\hat{\mathrm{m}}_{j}^{I}, \Sigma_{j}^{I})_{k+1}^{i} = (\hat{\mathrm{m}}_{j}^{I}, \Sigma_{j}^{I})_{k}^{i} \tag{58}$$

It is obvious that the EIKF estimation for the j^{th} landmark is an interval, so all estimates of the landmark consist of two boundary curves which contain all possible optimal (or suboptimal) estimates [54]. Here, we propose a time-varying Markov model (TVMM) based weighted average to achieve more reliable results for the EIKF.

Let $(\hat{\mathrm{m}}_{j})_{k}$ be the optimal (or suboptimal) position estimate for the j^{th} landmark at time k, where the filtering results of EIKF correspond to $(\hat{\mathrm{m}}_{j}^{I})_{k} = [(\hat{\mathrm{m}}_{j})_{k}, (\bar{\hat{\mathrm{m}}}_{j})_{k}]$. Thus, $(\hat{\mathrm{m}}_{j})_{k}$ can be approximately represented as

$$(\hat{\mathrm{m}}_{j})_{k} = \alpha_{k} (\hat{\mathrm{m}}_{j})_{k} + (1 - \alpha_{k}) (\bar{\hat{\mathrm{m}}}_{j})_{k} \tag{59}$$

where the weighted coefficients α_{k} satisfy $0 \leq \alpha_{k} \leq 1$.

We can establish the TVMM by discretization the state value of α_{k}. The s discrete state set of α_{k} is denoted as

$$S_{ds} = \{ \alpha^{i} \, | i = 1, \cdots, s \} \tag{60}$$

Meanwhile, let $p_{i}(k)$ denote the probability of $\alpha_{k} = \alpha^{i}$ at time k, so the *state probability vector*of $\alpha_{k}$is

$$P_{vect}(k) = (p_{1}(k), \cdots, p_{s}(k)) \tag{61}$$

Define the *one-step transition probability* as

$$p_{ij}(k) = p(\alpha_{k} = \alpha^{j} \left| \alpha_{k-1} = \alpha^{i} \right.), (i, j = 1, 2, \cdots, s) \tag{62}$$

According to $P_{ij}(k)$, the *transition probability matrix* of α_{k} can be written as

$$P_{tran}(k) = \left(p_{ij}(k) \right)_{s \times s} \tag{63}$$

where $p_{ij}(k) > 0$, and $\sum_{j} p_{ij}(k) = 1$.

Combining (61) with (63), the *probability prediction equation* is established as

$$P_{vect}(k+1) = P_{tran}(k) P_{vect}(k) \tag{64}$$

In fact, we have to dynamically adjust $P_{tran}(k)$ in real time. If we don't, the filtering precision of EIKF will be influenced. As we know in KF, if the state estimation of a system is relatively accurate at time k, then its innovation vector will be smaller at time $k+1$. In the context of EIKF, according to the innovation vector $\mathrm{z}_{k+1} - h_{k+1} \left((\hat{\mathrm{m}}_{j})_{k+1} \right)$ at time $k+1$, and combining (31) with (59), we can establish the following optimization problem:

$$\begin{cases} \min \left\| \mathrm{z}_{k+1} - h_{k+1} \left(f_{k}^{I} \left(\alpha_{k} (\hat{\mathrm{m}}_{j})_{k} + (1 - \alpha_{k}) (\bar{\hat{\mathrm{m}}}_{j})_{k} \right) \right) \right\| \\ 0 \leq \alpha_{k} \leq 1 \end{cases} \tag{65}$$

Consequently, the state value α_k^* is obtained by solving the problem (65), and then calculating the distance between α_k^* and each α^i as

$$d_i = \left\| \alpha_k^* - \alpha^i \right\|, \quad i = 1, \cdots, s \tag{66}$$

Upon the minimum of d_i, the discrete state value $S_{ds}(\alpha_k^*)$ corresponding to α_k^* is obtained, i.e.

$$S_{ds}(\alpha_k^*) = \left\{ \alpha^i \,|\min(d_1, d_2, \cdots, d_s) \right\} \tag{67}$$

If $\alpha_k^* = \alpha^q$ at time $k+1$, the Markov parameter adjustment rule of $P_{tran}(k)$ is given by

$$p_{ij}(k+1) = \begin{cases} p_{ij}(k) + \beta \sum\limits_{j=1, j\neq q}^{s} p_{ij}(k) & j = q \\ (1-\beta) p_{ij}(k) & j \neq q \end{cases} \tag{68}$$

where $0 < \beta < 1$.

With the iterative updating of $p_{ij}(k)$ based on (68), the transition probability matrix $P_{tran}(k)$ can also update dynamically with system changes, and the constraints $p_{ij}(k+1) > 0$ and $\sum_j p_{ij}(k+1) = 1$ can also hold.

Furthermore, we adopt the upper-bound matrix $(\Sigma_j^u)_k^i$ as a measure of the uncertainty of $(\hat{\mathrm{m}}_j)_k$.

IV. FAST ALGORITHM OF SLAM BASED ON COMBINATORIAL INTERVAL FILTERS

Following the scheme of the combined approach presented in Section III, the procedure for our proposed fast algorithm of SLAM and a characteristic analysis is now presented.

A. FAST ALGORITHM OF SLAM AND ITS PROCEDURE

Table 2 presents the full schema for the IBPF-EIKF SLAM. The implementation steps are as follows.

1) The initial box-particle set is generated in a prior-bounded state space region. Suppose that the system noise is known, then a position estimate of each box particle is obtained from the previous box by an inclusion function.

2) According to the latest laser measurement data, the data association technique is executed on a per-box basis, which relates the landmarks observed in the external environment to the landmarks in the map. In our case, the multiple hypothesis tracking (MHT) algorithm [56] is adopted to implement this process.

3) For each observed landmark, incorporate the measurement into the corresponding EIKF by updating the mean and covariance. When the update is finished, the updated values along with the new pose are added to the temporary box-particle sets.

4) Upon updating these landmarks to calculate the predicted measurement, then the innovation and the likelihood are calculated. The new box-particle set is obtained by *LP-contractor* to contract the temporary box-particle set.

5) After several iterations, box particles that have higher weights are more likely to survive, whereas those with lower weights are less likely. Thus, *DSS-resampling* is carried out depending on the threshold value to rescale each box particle weight. The resulting set of box particles then forms the new box-particle set.

TABLE 2. IBPF-EIKF SLAM algorithm.

Algorithm 2: IBPF-EIKF SLAM Algorithm

Input: $\{\omega_k^i, [\mathrm{x}_k^i]\}, [\mathrm{u}_k], [\mathrm{z}_{k+1}]$

Output: $\{\hat{x}_{k+1}, \hat{P}_{k+1}, (\hat{\mathrm{m}}_j)_{k+1}, (\Sigma_j^u)_{k+1}\}$

1. *For* $i = 1$ to N do
2. retrieve N box $[\mathrm{x}_k^i]$ from $[\mathrm{x}_k]$
3. pose propagation: $[\mathrm{x}_{k+1}^i] = [f]([\mathrm{x}_k^i], [\mathrm{u}_k])$
4. MHT data association $\leftarrow [\mathrm{z}_{k+1}]$
5. predicted measurement: $[\mathrm{z}_{k+1}^i] = [h]([\mathrm{x}_{k+1}^i], (\hat{\mathrm{m}}_j^I)_{k+1}^i)$
6. innovation: $[\mathrm{r}_{k+1}^i] = [\mathrm{z}_{k+1}^i] \bigcap [\mathrm{z}_{k+1}]$
7. likelihood: $A^i = \prod_1^p A^i(j), \quad A^i(j) = \left|[\mathrm{r}_{k+1}^i(j)]\right| \Big/ \left|[\mathrm{z}_{k+1}^i(j)]\right|$
8. *If* $[\mathrm{r}_{k+1}^i] \neq \phi$, *Then*
9. LP-contractor $\rightarrow [\mathrm{x}_{k+1}^i]^{new}$, and $[\mathrm{x}_{k+1}^i] \leftarrow [\mathrm{x}_{k+1}^i]^{new}$
10. *Else*
11. $[\mathrm{x}_{k+1}^i]^{new} = [\mathrm{x}_{k+1}^i]$
12. *EndIf*
13. weights update: $\omega_{k+1}^i = A^i \omega_k^i$
14. *EndFor*
15. *For* $i = 1$ to N do
16. $\omega_{k+1}^i \leftarrow \omega_{k+1}^i \Big/ \sum_{j=1}^N \omega_{k+1}^j$
17. *EndFor*
18. pose estimation: $\{\hat{x}_{k+1}, \hat{P}_{k+1}\}$
19. *If* the j^{th} landmark never seen before
20. $(\hat{\mathrm{m}}_j^I, \Sigma_j^I)_{k+1}^i \leftarrow$ initialize $\{(\hat{\mathrm{m}}_{new}^I)_{k+1}^i, (\Sigma_{new}^I)_{k+1}^i, H_{k+1}^I, \omega_0\}$
21. *Else*If $[\mathrm{z}_{k+1}]$ contains the j^{th} landmark in the map
22. $(\hat{\mathrm{m}}_j^I, \Sigma_j^I)_{k+1}^i \leftarrow$ EIKF $\{(\hat{\mathrm{z}}_{k+1}^I)^i, (C_{k+1}^I)^i, (K_{k+1}^I)^i\}$
23. *Else*
24. $(\hat{\mathrm{m}}_j^I, \Sigma_j^I)_{k+1}^i = (\hat{\mathrm{m}}_j^I, \Sigma_j^I)_k^i$
25. *EndIf*
26. TVMM weighted average $\{(\hat{\mathrm{m}}_j)_{k+1}^i, (\Sigma_j^u)_{k+1}^i\}$
27. *If* $N_{eff} < N_{th}$, *Then*
28. DSS-resampling $\rightarrow \{\omega_{k+1}^q = 1/N, [\mathrm{x}_{k+1}^q]\}, \; q = 1, \cdots, N$
29. *EndIf*
30. $k = k+1$, Goto step 1 until $k = k_{end}$

B. PERFORMANCE ANALYSIS FOR FAST ALGORITHM OF SLAM

The proposed combinatorial interval filters based on IA can provide guaranteed and consistent results for the SLAM implementation. On the one hand, use of boxes is more efficient in that it requires a significantly smaller set of particles than the generic PF, thereby reducing computational requirements and improving real-time application. In our work, the original BPF has two important improvements.

1) One is to use *LP-contractor* to solve the CSP, which provides *globally consistent* domains for the SLAM problem by handling all the constraints simultaneously. As can be seen

TABLE 3. Comparison of different rules.

Measure	Rule A			Rule B			Rule C			Rule D		
required	best	mean	worst	best	mean	worst	best	mean	worst	best	mean	worst
CPU time	0.01	487.87	18343.47	0.01	50.20	1379.06	0.01	43.17	1080.00	0.01	5466.22	182203.62
FE number	27	28082.54	971103	27	7848.33	197529	27	7328.95	174009	27	147270.74	4176639
DE number	82	106928.92	3906446	74	64863.63	800050	74	24402.95	708987	82	342616.05	7129277
List length	5	1761.90	68714	4	356.36	13898	4	329.62	12855	5	12471.33	486382

TABLE 4. Values relative to those obtained with rule A.

Measure	(B/A)			(C/A)			(D/A)		
required	best	mean	worst	best	mean	worst	best	mean	worst
CPU time	8%	93%	145%	6%	94%	155%	19%	943%	32620%
FE number	20%	93%	197%	18%	93%	195%	20%	202%	3720%
DE number	20%	93%	205%	18%	93%	203%	19%	198%	3572%
List length	20%	101%	304%	19%	103%	296%	19%	234%	5041%

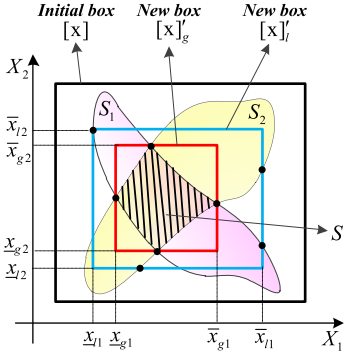

FIGURE 5. Consistent solution in new box after different contractor with local consistency and global consistency.

from Fig. 5, the solution $S_i = \{x \in [x] \,|\, g_i(x) = 0\}$ is provided after CSP $H : (g_i(x) = 0, x \in [x])$, $i = 1, 2$. The initial box [x] is contracted as $[x]'_l$ and $[x]'_g$ with local consistency and global consistency contractors, respectively. It is clear that any value of $[x]'_l$ should have at least one corresponding solution in all the setsS_i taken separately, whereas, any value of $[x]'_g$ corresponds to at least to one solution within S. Thus, the global consistency contractor may provide tighter domains, i.e.

$$S \subseteq [x]'_g \subseteq [x]'_l \subseteq [x] \tag{69}$$

Furthermore, in order to quantify the quality of the *LP-contractor*, we can define the ratio

$$\gamma = \frac{wid([b])}{wid([x])} = \frac{wid\left(\left([J_g]([x]) - [J_g]([x_0])\right)([x] - x_0)\right)}{wid([x])} \tag{70}$$

The ratio $\gamma \to 0$ means that the bracketing becomes more and more accurate, i.e. the box [x] converges to a point.

2) The other improvement is to use *DSS-resampling,* which is more favorable for the choice of subdivision dimensions. The numerical experience with respect to the required CPU time, the number of objective functions evaluation (FE), the derivative evaluation (DE) and the list length for four different rules of global optimization test problems (39 components) have been studied in [48] and [49]. It can be concluded that Rule C is definitely the best choice for the direction selection rule. The most important results are highlighted in Tab. 3 and Tab. 4 to show the superiority of Rule C in our proposed *DSS-resampling* approach.

On the other hand, the EIKF algorithm shows good stability in many applications compared with EKF, so that it can be an alternative solution that provides a robust way to perceive the environment surrounding the mobile robot. In our case, a feasible scheme of a weighted average based on TVMM is proposed to obtain more precise results for the EIKF. Meanwhile, a method with low computational complexity is adopted to calculate the interval inverse matrix to further improve the real-time usability of the algorithm (see Tab. 1)

V. SIMULATION EXPERIMENTS AND COMPARATIVE ANALYSIS

The simulation studies and a case study with mobile robot for the proposed algorithm are given in this section. All of the experiments were performed on a mobile computer with an Intel®Core™i5-5200U CPU @ 2.20 GHz with 8 GB of RAM running under Windows 7. The simulation experiments were implemented using MATLAB (2014a) and based on the INTLAB9.0 toolbox, which contains a number of built-in routines for the interval calculations. The actual experiment was conducted using Ubuntu14.04 and robot operating system (ROS).

A. SIMULATION RESULTS BASED ON BENCHMARK DATASET

In this study, a comparative analysis of FastSLAM2.0, *Unscented FastSLAM* (UFastSLAM) [57], *strong tracking square root central difference FastSLAM* (STSRCDFastSLAM) [58], BPF-EIKF SLAM, and IBPF-EIKF SLAM was conducted using the Car Park dataset [59], which is popular within the SLAM research community. In the simulation,

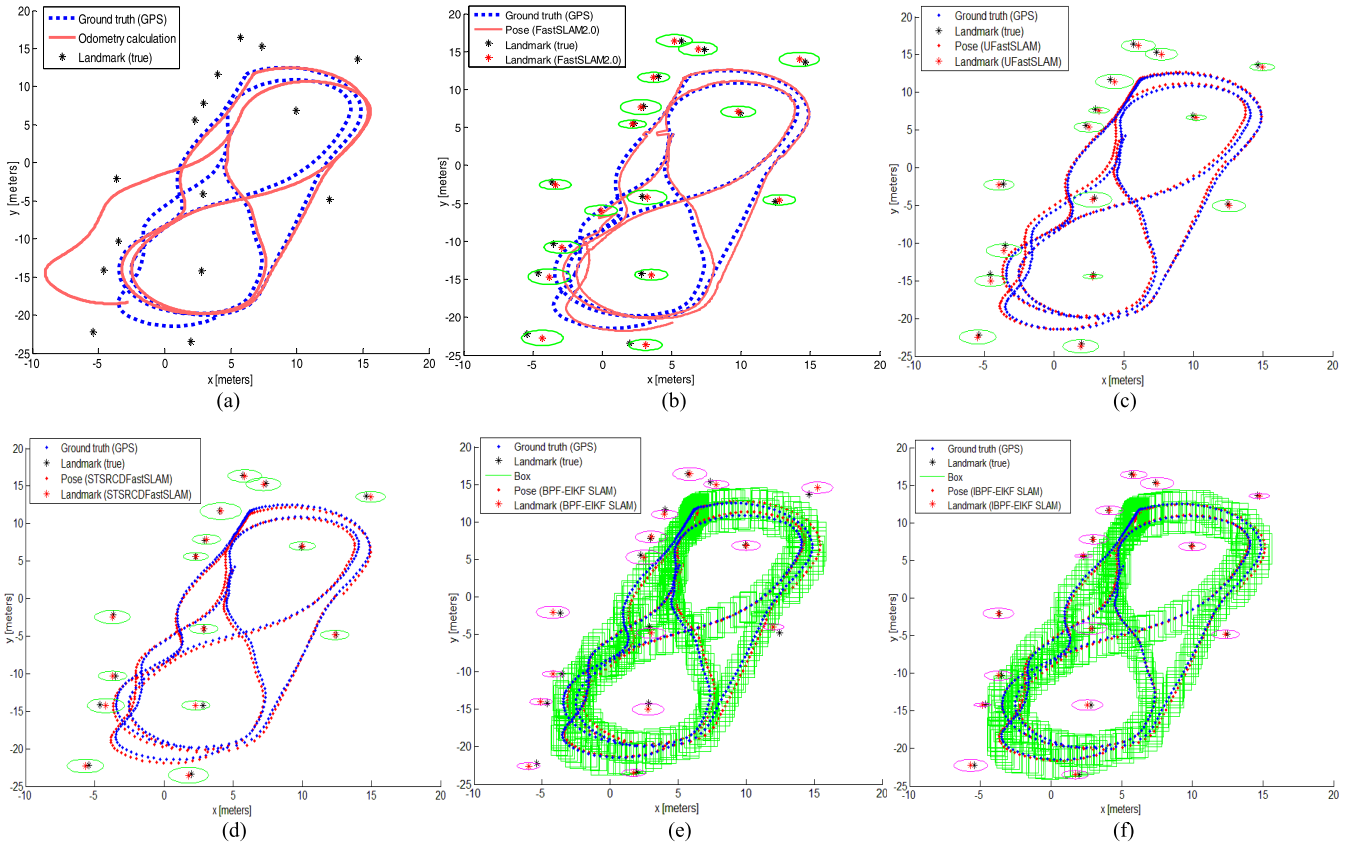

FIGURE 6. Estimated trajectories and landmarks with different algorithms. (a) Odometer trajectories; (b) FastSLAM 2.0 (Np = 100); (c) UFastSLAM (Np = 20); (d) STSRCDFastSLAM (Np = 20); (e) BPF-EIKF SLAM (Nb = 20); (f) IBPF-EIKF SLAM (Nb = 20).

the experimental site used to contain the dataset was located in a 30 × 45 m^2 campus parking area that had a large number of satellites for high-quality GPS information. The experimental platform was a four wheeled vehicle equipped with odometer, GPS, front wheel yaw sensor, and laser range finder with a 180° frontal field of view. Artificial landmarks were used in the car park, consisting of 60 mm steel poles covered with reflective tape. GPS data was used to determine the accuracy ground truth and true position of these landmarks.

In simulations, the numbers of particles and box particles were denoted as Np and Nb, respectively. The vehicle control noise was set as (0.3m/s, 2°), the measurement noise was zero-mean white Gaussian with a covariance $diag[\sigma_r^2, \sigma_\varphi^2]$, where $\sigma_r = 0.4$m, $\sigma_\varphi = 3°$. For the two interval filtering based SLAM methods, we used the 99% interval confidences, i.e. $3\sigma_r$, $3\sigma_\phi$, to model a uniform noise. In TVMM, the discrete state set $S_{ds} = \{0.1, 0.3, 0.5, 0.7, 0.9\}$. Initially, $\alpha_k = 0.5$ and $P_{vect}(k) = (0.1, 0.1, 0.6, 0.1, 0.1)$.

Figure 6 (a) depicts the trajectory calculated from the odometer, which deviates from the ground truth measured by GPS. From Fig.6 (b), the FastSLAM2.0 results show that some parts of the estimated trajectory and landmarks were far from the ground truth and true landmarks. From Fig. 6 (c), we find that UFastSLAM outperform FastSLAM 2.0 in accuracy and robustness. UFastSLAM computes a more accurate mean and more precise uncertainty of the vehicle by applying the *unscented particle filter* (UPF), which takes into account a linear regression of weighted points. Thus, UFastSLAM can effectively reduce the linearization error and use unscented transformation to further reduce the number of particles needed. From Fig. 6 (d), the accuracy of the state estimation with STSRCDFastSLAM has been improved over that with FastSLAM2.0 and UFastSLAM because the proposal distribution and the feature position are calculated using the *strong tracking square root central difference particle filter* (STSRCDPF).

It can be seen from Fig. 6 (e), (f) that the two interval-based SLAM methods present more guaranteed results than those of FastSLAM2.0. Also, the estimated result of the original BPF presents considerable error compared with the reference value. In contrast, the trajectory prediction of IBPF-EIKF SLAM is more consistent with respect to the ground truth, and the corresponding landmark estimates are more accurate than those of the other methods. Thus, equivalent results are expected with smaller numbers of particles with an interval-based method, and the performance of IBPF-EIKF SLAM is better than BPF-EIKF SLAM. Furthermore, it can be concluded that the estimated results of STSRCDFastSLAM and IBPF-EIKF SLAM are almost the same, and slightly better than that of UFastSLAM.

From Fig. 8, we can find the position error of estimated landmark for two interval-based methods are less than that of FastSLAM2.0. In spite of the BPF-EIKF SLAM is worse than FastSLAM2.0 at the tenth landmark, the whole resulting map of BPF-EIKF SLAM is more accurate. The blue line also represents the estimation error for IBPF-EIKF SLAM, but it involves just a general weighted average rather than TVMM. This illustrates the effectiveness of the TVMM based weighted average. Thus, the accuracy of our approach for landmark estimation is better than FastSLAM2.0 and BPF-EIKF SLAM. With the exception of the third and twelfth landmarks, there is little difference in the estimated landmark positions between UFastSLAM, STSRCDFastSLAM and IBPF-EIKF SLAM.

In order to compare the estimation accuracy and stability of above mentioned algorithms under different measurement noise level conditions, we set six groups of measurement noise data for each simulation. For each measurement noise level, the mean and standard deviation of the *root mean square error* (RMSE) were calculated over 30 runs for each SLAM algorithm. The RMSE is calculated as follows:

$$RMSE = \sqrt{\frac{1}{N}\left(\sum_{i=1}^{N} |\hat{x}_k^i - x_k^*|^2\right)} \tag{71}$$

where N is the number of particles in each simulation, x_k^* is the true state, and $\hat{x}_k^i$ is the estimated state of the i^{th} particle or box particle at time k.

Figure 7 compares the results of RMSE for trajectory prediction and landmark estimation. At each measurement noise level, the mean and standard deviation for UFastSLAM, STSRCDFastSLAM and IBPF-EIKF SLAM increase at a slower rate than for the other two algorithms. Hence, the ability to suppress noise of UFastSLAM, STSRCDFastSLAM and IBPF-EIKF SLAM is stronger than that of FastSLAM2.0 and BPF-EIKF SLAM, with respect to increasing measurement of noise levels. The growth rate of UFastSLAM, STSRCDFastSLAM and IBPF-EIKF SLAM is relatively similar, but the accuracy and stability of STSRCDFastSLAM and IBPF-EIKF SLAM outperform UFastSLAM. Compared with STSRCDFastSLAM and IBPF-EIKF SLAM, we find that, with increasing noise level, the influence of measurement noise on STSRCDFastSLAM is relatively large.

According to the inclusion criterion [43], the true value of the pose state vector must be contained in the support of the posterior spatial PDF. In BPF, the credible set $C_k(1)$ is approximated by the union of all of the box particles, that is,

$$C_k(1) = \bigcup_{i=1}^{N} [\mathrm{x}_k^i] \tag{72}$$

Thus, the inclusion criterion ρ_k is calculated as:

$$\rho_k = \begin{cases} 1, & \text{if } x_k^* \in \bigcup_{i=1}^{N} [\mathrm{x}_k^i] \\ 0, & \text{otherwise} \end{cases} \tag{73}$$

The failure to satisfy the inclusion criteria indicated the filter divergence. If the inclusion $\rho_k = 1$, this indicates that the true value of the state is consistently contained within the particle support set. As illustrated in Fig. 9, the IBPF is sufficient to satisfy the inclusion criterion. This is a useful advantage of the IBPF over the original BPF.

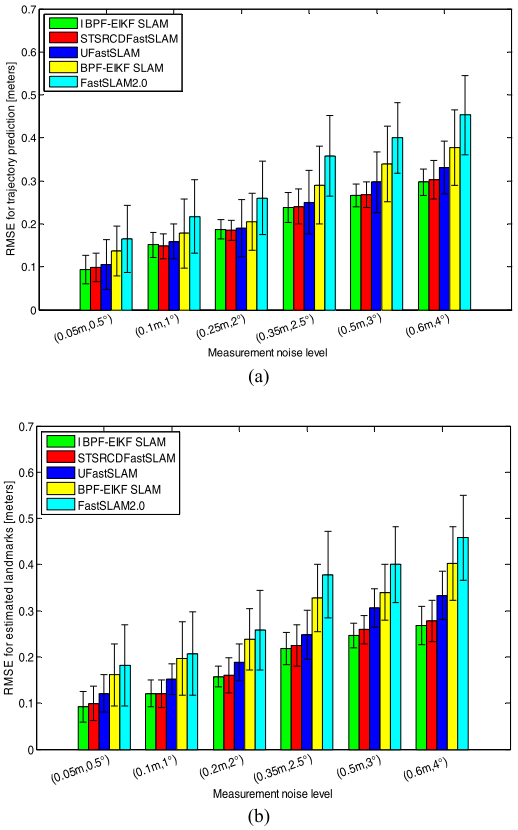

FIGURE 7. RMSE with varying levels of measurement noise. (a) RMSE for trajectory prediction; (b) RMSE for landmark estimation.

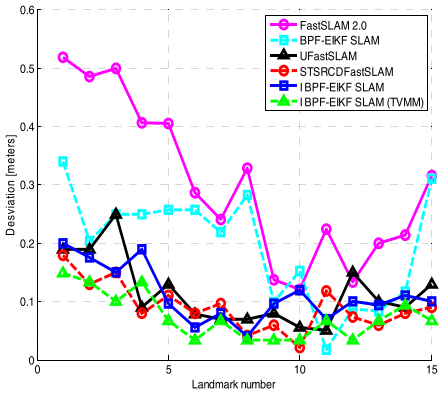

FIGURE 8. Landmark position errors.

Figure 10 illustrates the measurement innovation sequences of the landmarks and their 95% confidence intervals (2 standard deviations) for EKF and EIKF. As can be seen, it remains white and validates the assumed statistic for the model and sensors. The results further show that EIKF has better robustness and accuracy than EKF. We randomly selected 10 sets of

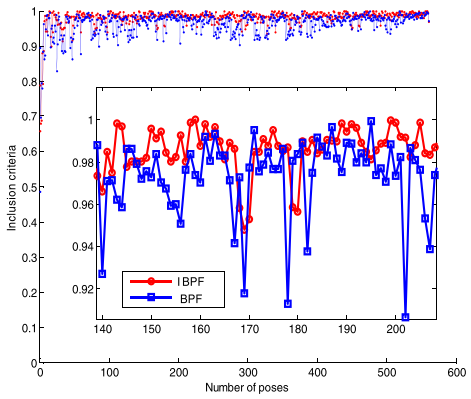

FIGURE 9. Inclusion values.

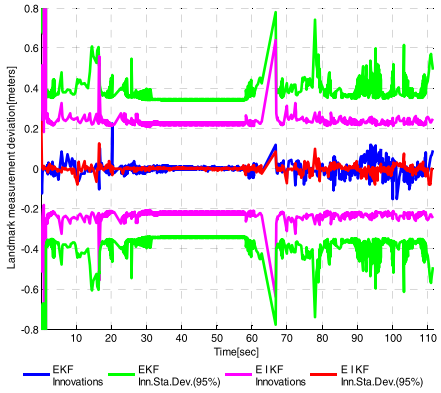

FIGURE 10. Landmark measurement innovation sequence.

TABLE 5. Comparison of the different contractors.

True pose (GPS)	Contractor	Estimated interval
(7.911, 12.384)	CP	([7.856 8.023], [12.301 12.435])
	LP	([7.889 7.987], [12.312 12.395])
(-0.842, -6.815)	CP	([-0.902 -0.783], [-6.904 -6.712])
	LP	([-0.892 -0.788], [-6.883 -6.759])
(1.286, -4.860)	CP	([1.271 1.293], [-4.899 -4.758])
	LP	([1.312 1.290], [-4.881 -4.852])
(-1.895, -17.998)	CP	([-1.975 -1.612], [-18.719 -17.523])
	LP	([-1.933 -1.616], [-18.603 -17.617])
(-3.233, -12.482)	CP	([-3.241 -3.015], [-12.491 -12.465])
	LP	([-3.238 -3.201], [-12.490 -12.470])
(5.751, 10.366)	CP	([5.312 5.793], [10.019 10.523])
	LP	([5.591 5.823], [10.221 10.525])
(6.122, 11.458)	CP	([6.101 6.131], [11.396 11.463])
	LP	([6.102 6.131], [11.421 11.461])
(8.896, 12.499)	CP	([8.842 9.001], [12.451 12.502])
	LP	([8.866 8.923], [12.465 12.500])
(14.337, 9.088)	CP	([13.302 14.349], [8.989 9.100])
	LP	([13.511 14.343], [9.042 9.097])
(13.477, -20.642)	CP	([13.403 13.489], [-20.650 -20.609])
	LP	([13.422 13.487], [-20.650 -20.622])

pose estimation boxes in one trial for comparison of the CP approach and *LP-contractor*. From Tab. 5, it is obvious that *LP-contractor* provides a much more efficient solution than the CP approach.

The CPU time for the SLAM process and the average time taken to run all filtering steps over 30 simulation runs with each SLAM approach are reported in Tab. 6. The average times taken by the two interval-based SLAM methods are longer than those of FastSLAM2.0, UFastSLAM and STSR-CDFastSLAM with the same number of particles. The running time of FastSLAM2.0 is proportional to the number of particles, whereas in the two interval-based SLAM methods, most of the time is consumed for the calculation of intervals. However, as mentioned above, a lower number of particles for the interval-based algorithm are needed to obtain the same estimation accuracy with FastSLAM2.0 (Np $=$ 100, Nb $=$ 20). This indicates that our proposed algorithm outperforms FastSLAM2.0 in computational efficiency while maintaining the same accuracy. Meanwhile, in spite of BPF-EIKF SLAM being slightly faster than IBPF-EIKF SLAM, the results of IBPF-EIKF SLAM are better.

Furthermore, UFastSLAM requires computation of a matrix square root that can be implemented directly using the Cholesky factorization. The covariance matrices have low dimensions and can be expressed recursively. Thus, not only does UFastSLAM outperform FastSLAM in accuracy and robustness, it also does this at no extra computational cost. The increase in dimension results in an increase in computational cost, so that the computational cost of UFastSLAM is slightly higher than that of FastSLAM2.0. There is an adaptive fading factor step in STSRCDFastSLAM, which results in extra computational cost. Compared with the other tested methods, the complexity of STSRCDFastSLAM is moderate.

B. SIMULATION RESULTS BASED ON SIMULATOR

In this study, a simulator developed by Bailey [60] was used to evaluate the performance of different SLAM approach mentioned above. We assumed that the data association was unknown and the individual compatibility nearest neighbor (ICNN) method was used for the association [61]. The simulation environment had an area of 90 m$\times$80 m with 72 landmarks, as illustrated in Fig. 12 (a).

The vehicle has a 0.26 m wheel base and is equipped with a range bearing sensor with a maximum range of 20 m and a 180° frontal field-of-view. Gaussian noise covariances are generated for both the measurement and the motion. The control frequency was 40 Hz and observation scans were obtained every 5 Hz. The control noise and the measurement noise were respectively set to $(0.3m/s, 3°)$ and $(0.2m, 4°)$. The numbers of particles and box particles were set as Np $=$ 10 and Nb $=$ 10, respectively.

As can be seen in Fig. 12 (c), because the feature and the vehicle trajectories were predicted and updated by UPF in UFastSLAM, a more precise variance and robot uncertainty was calculated. Thus, the estimated trajectories and landmarks of UFastSLAM outperform FastSLAM2.0. However, as the feature is continuously added to the state vector, the dimension of SLAM will continue to increase. This leads

TABLE 6. Computational costs of different slam algorithms.

Particle number	FastSLAM2.0		UFastSLAM		STSRCDFastSLAM		BPF-EIKF SLAM		IBPF-EIKF SLAM	
	CPU time(s)	Mean time(s)	CPU time(s)	Mean time(s)	CPU time(s)	Mean time(s)	CPU time(s)	Mean time(s)	CPU time(s)	Mean time(s)
1	0.102	32.719	0.106	34.006	0.107	39.869	0.111	40.671	0.112	42.437
10	0.111	46.133	0.119	50.146	0.123	54.357	0.168	54.944	0.173	58.804
15	0.132	50.904	0.176	58.203	0.181	60.605	0.249	61.902	0.251	65.773
20	0.161	54.459	0.203	63.014	0.291	67.311	0.283	68.014	0.301	70.941
30	0.180	60.875	0.211	71.257	0.304	79.780	0.330	79.378	0.364	85.327
50	0.227	64.482	0.291	75.177	0.312	86.006	0.397	87.132	0.412	93.101
80	0.288	76.620	0.368	88.209	0.385	96.412	0.468	102.245	0.505	111.306
100	0.352	83.186	0.421	97.354	0.473	109.327	0.501	115.029	0.573	124.714

to an increase in the radius of the super ellipsoid bounded by the symmetric *sigma point*. Although the *sigma point* can correctly represent the mean and covariance of the prior distribution of state variables, it does affect the accuracy of the algorithm. We find that the accuracy of UFastSLAM gradually worsens after about half of the trajectory. This is because, as the robot continues to explore new areas, the new features are constantly added to the state vector, which increases the dimension of the scaled unscented transformation (SUT) and affects the accuracy of estimation.

From Fig. 12 (d), STSRCDFastSLAM combines the advantages of *square root central difference Kalman filter* (SRCDKF) and *strong tracking filter* (STF). The square root strategy ensures the symmetry of the covariance matrix, which effectively reduces the error caused by the system mutation. The filter gain matrix is adjusted in real time by introducing the fading factor and by adaptively adjusting the weight of the corresponding data, which leads to more accurate and more robust results. From Fig. 12 (e), we find the overall accuracy of BPF-EIKF SLAM is slightly lower than that of UFastSLAM, and there are large errors in some places. Fig. 12 (f) shows that the estimated positions and landmarks are close to their true positions. The results show that the accuracy of the state estimation with IBPF-EIKF SLAM has been improved over that of FastSLAM2.0, UFastSLAM and BPF-EIKF SLAM. Also, the estimation accuracy of IBPF-EIKF SLAM and STSRCDFastSLAM is very similar.

Figure 11 compare the absolute values of vehicle pose errors in different axes over time of the five tested algorithms. Apparently, the pose error of FastSLAM2.0 is larger than those of the two interval-based algorithms. The results show that the trajectory prediction accuracy of IBPF-EIKF SLAM is better than that of BPF-EIKF SLAM and FastSLAM2.0. In addition, we find that the error of STSRCDFastSLAM and BPF-EIKF SLAM is very similar and slightly smaller than UFastSLAM.

Figure 13 demonstrates the RMSE for the mentioned SLAM algorithms in trajectory prediction when varying the number of particles as 1, 10, 15, 20, 30, 50, 80 or 100 (each run 30 times). It is clear that the RMSE mean values of trajectory prediction for all five algorithms gradually reduce with increasing numbers of particles. However,

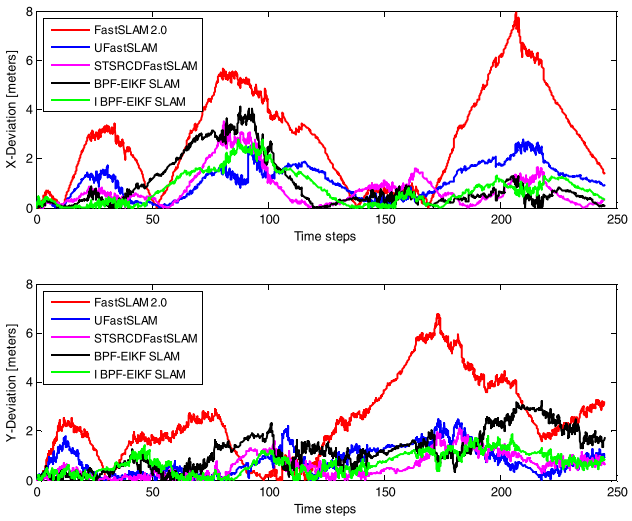

FIGURE 11. Absolute values of errors in different axis.

the two interval-based algorithms reduce the number of particles needed to achieve the same accuracy, so the computational load and real-time usability are improved. Using only 20 particles, IBPF-EIKF SLAM can achieve the same error rate as FastSLAM2.0 with 100 particles. With the same number of particles, the performance in terms of RMSE of UFastSLAM is slightly inferior to that of IBPF-EIKF SLAM and STSRCDFastSLAM. It is important to note that, because the posterior probability is fitted with a uniform distribution in the two interval-based methods, after a certain level of accuracy is achieved it is impossible to significantly improve the accuracy simply by increasing the number of particles. Therefore, when the number of particles exceeds a certain number, the RMSE of UFastSLAM and STSRCDFastSLAM is smaller than that of IBPF-EIKF SLAM.

Furthermore, we employ the percentage of effective particles number N_{Peff} to measure the efficiency of different SLAM methods, i.e.

$$N_{Peff} = \frac{N_{eff}}{N} = \frac{1}{N \cdot \sum_{i=1}^{N} (\omega_k^i)^2} \tag{74}$$

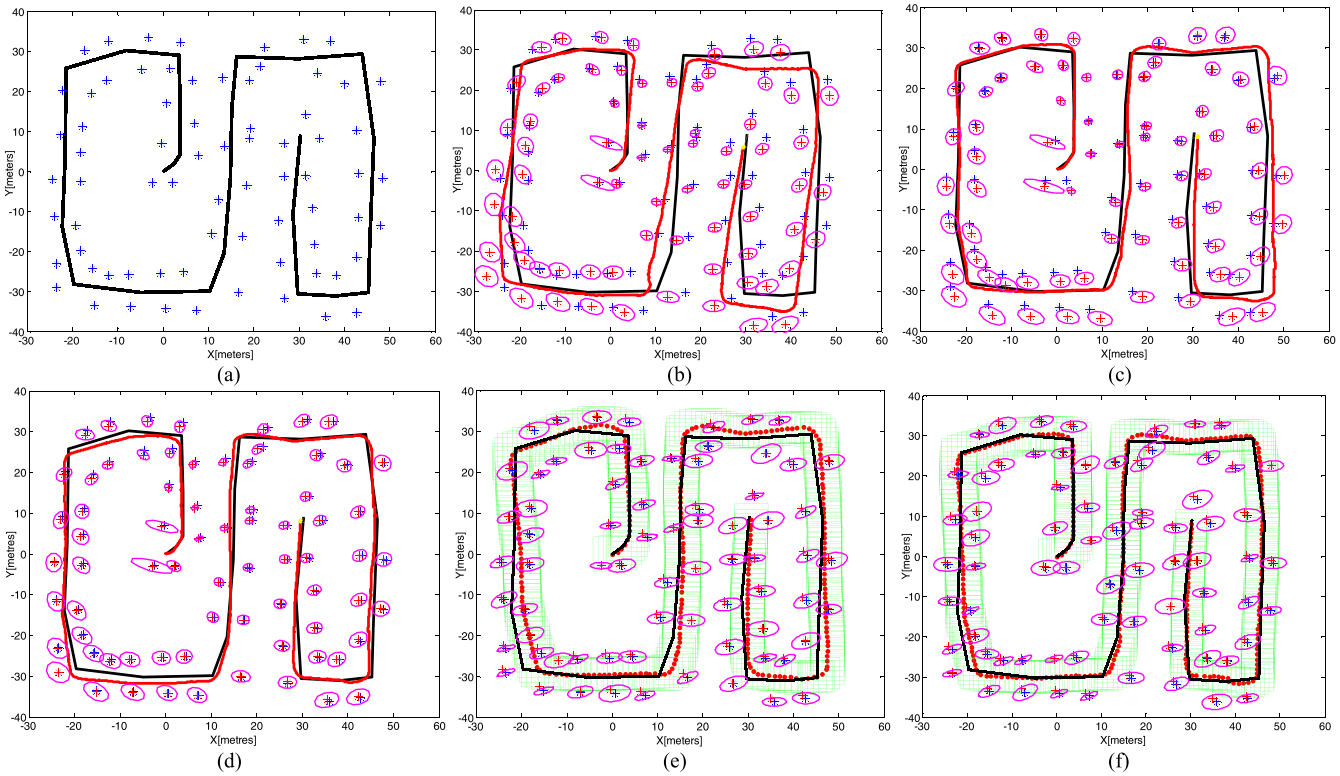

FIGURE 12. (a) Simulation environment; (b) FastSLAM 2.0; (c) UFastSLAM; (d) STSRCDFastSLAM; (e) BPF-EIKF SLAM; (f) IBPF-EIKF SLAM. The black line "-" and the red line "-" denote the true trajectory and the estimated trajectory of vehicle, respectively. The blue "∗" denotes the landmark location, and the red "∗" denotes the estimated landmark. The magenta ellipse is the covariance of the estimated landmarks, and the green rectangular represents box particle.

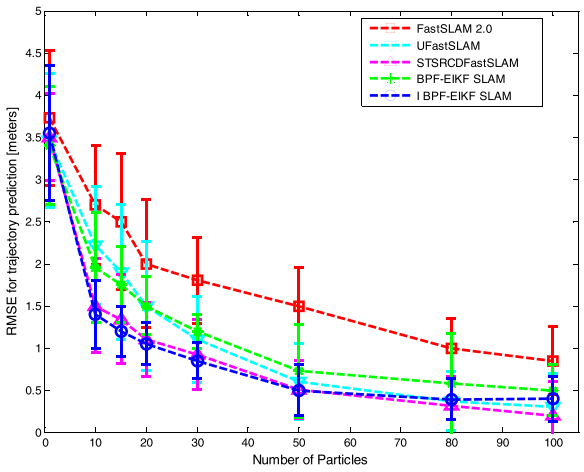

FIGURE 13. RMSE vs. number of particles.

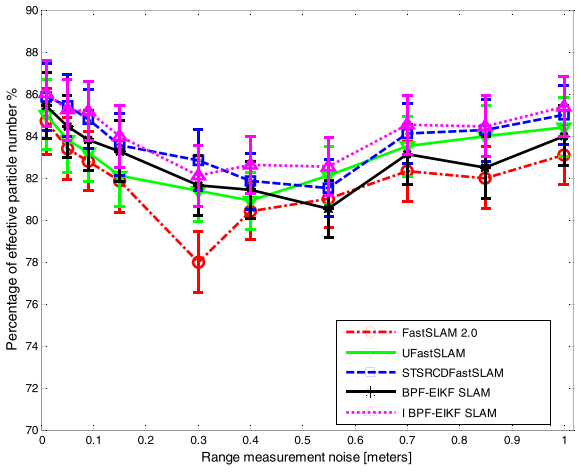

FIGURE 14. Percentage of effective particle number with increasing measurement noise levels in range.

Thus, a series of simulation experiments were carried out under the map conditions shown in Fig. 12 (a). The speed control noise of the vehicle was set at 0.4 m/s, the angular noise of measurement was set at 1.5°, and the range noise of measurement was set at ten group values: 0.01 m, 0.05 m, 0.09 m, 0.15 m, 0.3 m, 0.4 m, 0.55 m, 0.7 m, 0.85 m and 1.0 m (each run 30 times). As shown in Fig. 14, for the ten sets of measurement noise, the effective particle percentage of the proposed algorithm is greater than 82%, which means that only 18% of particles were useless for the SLAM filter. Compared with FastSLAM 2.0, the other three methods also had a higher effective particle number.

To verify the consistency of our proposed algorithms, the average *normalized estimation error squared* (NEES)

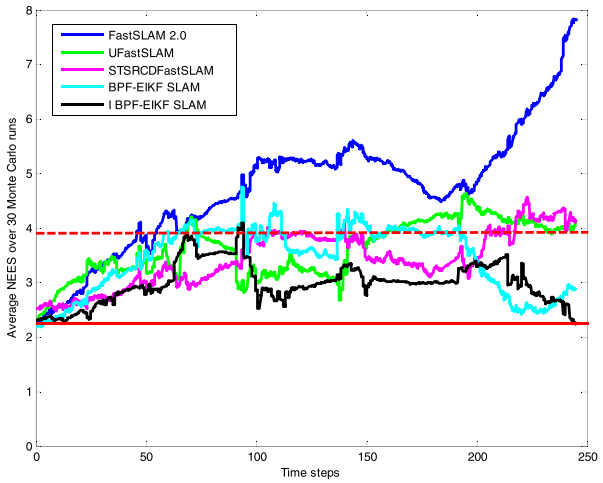

FIGURE 15. Consistency with different SLAM algorithms. The red line "-" and red dotted line "–" denote the lower bound and the upper bound of consistency test, respectively.

is used as a measure [17]. For an available true state x_k^* and the estimated mean and covariance $\{\hat{x}_k, \hat{P}_k\}$, we can use NEES over N_R Monte Carlo runs to characterize the filter performance:

$$\varepsilon_t = (x_k^* - \hat{x}_k)^T \hat{P}_t^{-1} (x_k^* - \hat{x}_k) \tag{75}$$

The average NEES is calculated as follows:

$$\bar{\varepsilon}_t = \frac{1}{N_R} \sum_{i=1}^{N_R} \varepsilon_{it} \tag{76}$$

Thus, 30 Monte Carlo simulations were performed with the two-sided 95% probability concentration region for $\bar{\varepsilon}_t$ bounded by the interval [2.19, 3.93]. Fig. 15 shows that, because a more accurate estimate without accumulating linearization error is achieved in UFastSLAM, many particles contain accurate information repetitively. This induces the possibility of maintaining consistency. As for STSRCDFastSLAM, a better proposal distribution is proposed and the adaptive partial systematic resampling method maintains particle diversity. Consequently, the consistency of STSRCDFastSLAM is prolonged for longer than that of UFastSLAM and FastSLAM2.0. The consistency of IBPF-EIKF SLAM is better than that of the other tested methods. This is because the main advantage of the interval-based approach lies in its ability to provide guaranteed and consistent results. Compared with BPF-EIKF SLAM, we find that the improved strategies in IBPF-EIKF SLAM for box contracting and resampling effectively enhance consistency.

C. CASE STUDY WITH A MOBILE ROBOT

The IBPF-EIKF SLAM algorithm was implemented and tested using a differential drive wheeled robot equipped with a SLAMTEC laser radar (RPLIDAR-A1) with a maximum range of 6m and a 360ř frontal field of view (see Fig. 16). The mobile robot moved at a speed of 0.5 m/s with a maximum

FIGURE 16. Mobile robot platform.

FIGURE 17. Experimental scene.

steering speed of 0.3 rad/s. The control frequency was 40 Hz, and observation scans were obtained at 5 Hz. The experimental site was in a conference room in the New Main Building of Beihang University, Beijing, and the experimental scene is depicted in Fig. 17.

In order to validate the performance of the proposed method, five experiments were carried out with the mobile robot using GridSLAM [62], GMapping [63], UFastSLAM, IBPF-EIKF SLAM, and STSRCDFastSLAM. In this study, we set 27 landmarks, with the resolution of the resulting map set to 25cm^2/cell. The motion of the mobile robot was controlled via a laptop keyboard, which was based on the installed software package *teleop_twist_keyboard* under ROS. Since the mobile robot had horizontal movement, the starting point was set as the coordinate origin (0, 0). We selected the start point, end point and each turning point as measurement points. These measurement points were connected in sequence as the true trajectory to measure the localization accuracy (see Fig. 19).

Figure 18 illustrates typical experimental results for the five tested algorithms. In order to provide a consistent performance comparison, only 10 particles were used in each case. As can be seen in Fig. 18(a), GridSLAM failed to map the environment correctly. Although GMapping can complete the final map learning, it contains clear deviations (Fig. 18(b)). Figure 18(c) shows that the map constructed by IBPF-EIKF SLAM is consistent with the actual environment; the edge of the map is distinct and there is no overlap or distortion. Figure 18 (d) and (e) illustrate the results of UFastSLAM and STSRCDFastSLAM, respectively. We find there are some

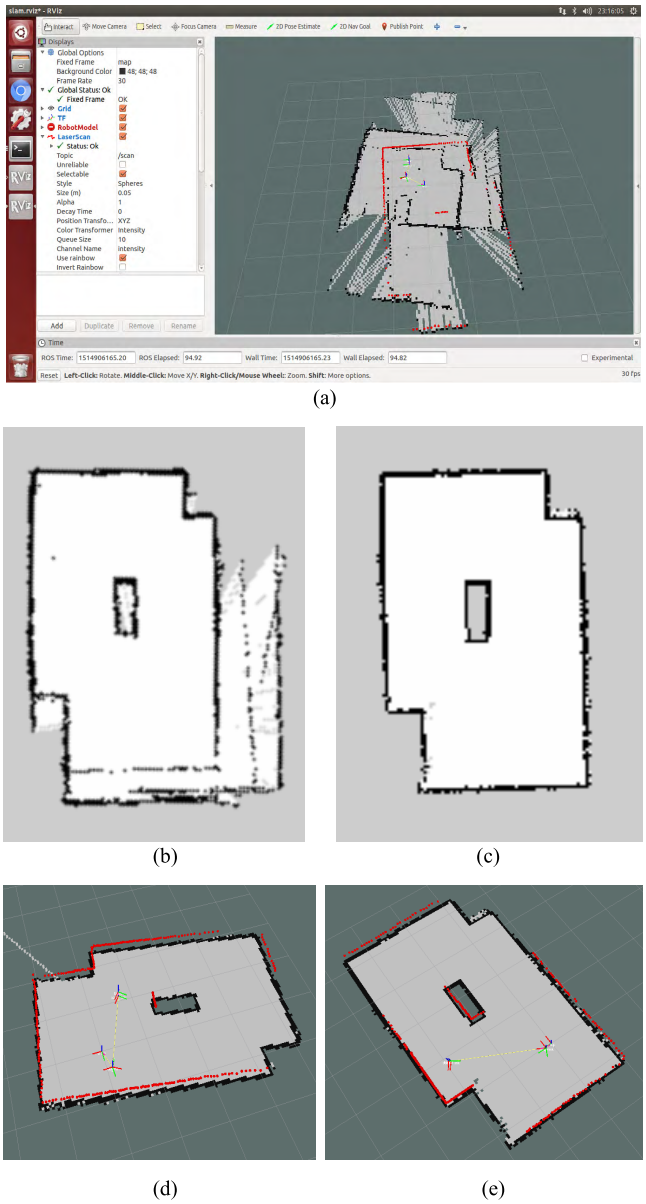

FIGURE 18. The generated 2-dimensional floor plan occupation grid map for different SLAM algorithms. (a) GridSLAM; (b) GMapping; (c) IBPF-EIKF SLAM; (d) UFastSLAM; (e) STSRCDFastSLAM.

TABLE 7. Comparative performance of different algorithms.

Algorithm	Particle number	CPU time (s)	Mean time (s)
GridSLAM	80	0.873	289.10
GMapping	30	0.381	256.37
UFastSLAM	15	0.322	243.94
STSRCDFastSLAM	10	0.303	232.02
IBPF-EIKF SLAM	10	0.316	239.69

serrated edges on the map created by UFastSLAM, whereas STSRCDFastSLAM generates a consistent map that is the same as that generated by IBPF-EIKF SLAM.

Table 7 compares the results of the five algorithms in producing the same precision map. It can be seen that the

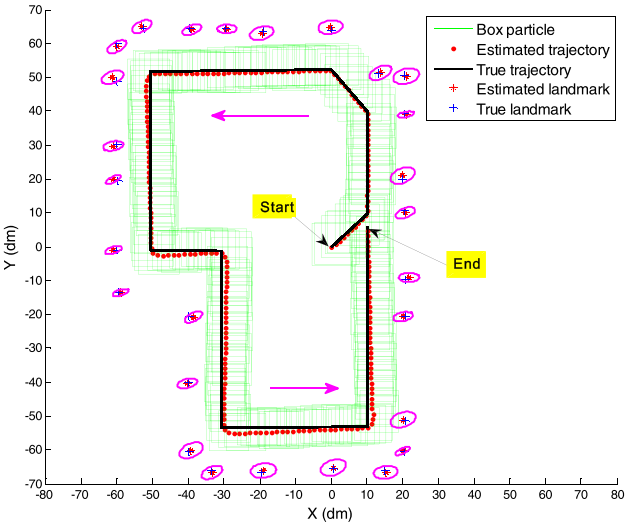

FIGURE 19. Estimated trajectory for IBPF-EIKF SLAM. The mobile robot starts at position "Start" and moves along the black line in the direction of the purple arrow until position "End".

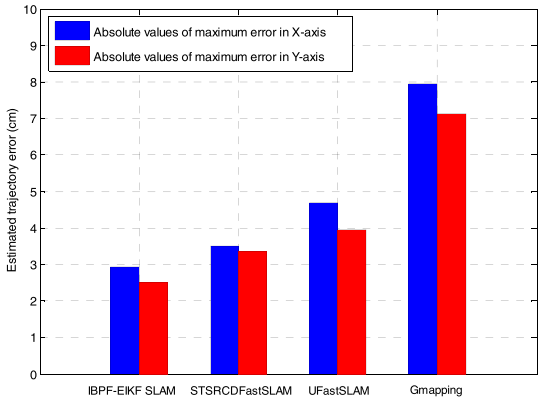

FIGURE 20. Absolute values of maximum error for different SLAM algorithms (Np = 10, Nb = 10).

number of particles required for STSRCDFastSLAM, UFastSLAM, and IBPF-EIKF SLAM are lower than for the other two algorithms. Although GMapping is currently the most effective method for building grid maps, IBPF-EIKF SLAM takes less time to complete the mapping and thus meets real-time requirements well.

From Fig. 19, it can be concluded that the trajectory defined by the location boxes correctly follows the trajectory defined by the reference points, and all the reference points are included in the localization boxes. Consequently, the localization boxes demonstrate the consistency of the proposed IBPF method. In order to quantify the maximum error generated by the whole process for different algorithms, Fig. 20 depicts the maximum error of estimated trajectory in different axes. For IBPF-EIKF SLAM, we find that the maximum error of pose estimation in the x-direction is no more than 3 cm, and in the y-direction is about 2.5 cm. Thus, the experimental results show that our method can accurately estimate the trajectory of the mobile robot, and that the constructed map clearly reflects the real environment,

thereby demonstrating the feasibility and effectiveness of the IBPF-EIKF SLAM method.

VI. CONCLUSION AND REMARKS

This paper describes the development of a fast SLAM algorithm that replaces traditional numerical filters with combinatorial interval filters within a FastSLAM framework. The combinatorial interval filters approach is based on the interval framework which seems to be a good methodology for SLAM applications in the case of non-white and biased measurements. One of the main advantages of interval-based approaches is its ability to provide guaranteed and consistent results. The proposed scheme of SLAM based on combinatorial synthetic integration has two advantages: (1) Instead of point particles and probabilistic models for the errors and inputs, the key concept in BPF is to use box particles and a bounded error model to achieve noise processing. Each weighted box particles is sequential recursive under the IA framework, which effectively reduces the computational complexity and enhances the real-time performance of the algorithm. The improved schemes for box contracting and resampling further enhance the consistency of estimation. (2) EIKF is similar to standard EKF in terms of statistical performance and iterative form, but the EIKF algorithm has strong adaptability and robustness. For the estimated interval of EIKF, the weighted averages based on TVMM were demonstrated to provide more reliable results. A series of simulations and experiments demonstrate the superior performance of our interval-based approach.

In our work, each box particle is seen as approximating a uniform distribution. In fact, simply increasing the number of box particles caused no significant improvement in filtering accuracy. Thus, ways to optimize boxes and select probability density functions should be part of future research work; for example, by using Gaussian distributions. Solutions to this problem will benefit the building of more accurate maps with fewer particles for SLAM systems. Based on the Gaussian components, we plan to employ the Dirichlet process (DP) to determine the number of components and find better ways to determine the number of box particles. In addition, since the unscented Kalman filter (UKF) is better than the EKF in dealing with nonlinear problems, we intend to develop an interval UKF (IUKF), instead of EIKF, for map learning. Another perspective of this research is to adapt the IBPF for visual SLAM problems, and to combine with some deep learning methods [64], [65] and image processing techniques.

REFERENCES

[1] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *Int. J. Robot. Res.*, vol. 5, no. 4, pp. 56–68, 1986.

[2] M. A. Paskin, "Thin junction tree filters for simultaneous localization and mapping," presented at the 18th Int. Joint Conf. Artif. Intell., 2003, pp. 1157–1164.

[3] S. Thrun, D. Koller, and Z. Ghahramani, "Simultaneous localization and mapping with sparse extended information filters: Theory and initial results," *STAR*, vol. 7, no. 1, pp. 363–380, 2002.

[4] K. P. Murphy, "Bayesian map learning in dynamic environments," in *Proc. NIPS*, 2000, pp. 1015–1021.

[5] K. Murphy and S. Russell, "Rao-Blackwellised particle filtering for dynamic Bayesian networks," in *Sequential Monte Carlo Methods Practice*. New York, NY, USA: Springer, 2001, pp. 499–515.

[6] M. Montemerlo, S. Thun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc. AAAI Nat. Conf. Artif. Intell.*, Edmonton, AB, Canada, 2002, pp. 593–598.

[7] M. Montemerlo, S. Thun, D. Koller, and B. Wegbreit, "FastSLAM2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Proc. Int. Conf. Artif. Intell.*, Acapulco, Mexico, 2003, pp. 1151–1156.

[8] A. Diosi and L. Kleeman, "Laser scan matching in polar coordinates with application to SLAM," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, Edmonton, AB, Canada, Aug. 2005, pp. 3317–3322.

[9] G. Grisettiyz, C. Stachniss, and W. Burgard, "Improving grid-based SLAM with rao-blackwellized particle filters by adaptive proposals and selective resampling," in *Proc. IEEE Int. Conf. Robot. Automat.*, Barcelona, Spain, Apr. 2005, pp. 2432–2437.

[10] A. R. Khairuddin, M. S. Talib, H. Haron, and M. Y. C. Abdullah, "GA-PSO-FASTSLAM: A hybrid optimization approach in improving FastSLAM performance," in *Proc. Int. Conf. Intell. Syst. Des. Appl.*, 2016, pp. 57–66.

[11] X. J. Yan, C. X. Zhao, and J. Z. Xiao, "A novel FastSLAM algorithm based on iterated unscented Kalman filter," in *Proc. Int. Conf. Robot. Biometrics*, Dec. 2011, pp. 1906–1911.

[12] J. Zhang, Y. Jiang, and K. Wang, "A modified FastSLAM for an autonomous mobile robot," in *Proc. IEEE Int. Conf. Mechatron. Automat.*, Aug. 2016, pp. 1755–1759.

[13] Y.-F. Zhang, Q.-X. Zhou, J.-Z. Zhang, Y. Jiang, and K. Wang, "A FastSLAM algorithm based on nonlinear adaptive square root unscented Kalman filter," *Math. Problems Eng.*, vol. 2017, Mar. 2017, Art. no. 4197635.

[14] T.-Z. Lv, C.-X. Zhao, and H.-F. Zhang, "An improved FastSLAM algorithm based on revised genetic resampling and SR-UPF," *Int. J. Automat. Comput.*, pp. 1–10, May 2014.

[15] H. Ankişhan, F. Ari, E. Ö. Tartan, and A. G. Pakfiliz, "Square root central difference-based FastSLAM approach improved by differential evolution," *Turkish J. Elect. Eng. Comput. Sci.*, vol. 24, no. 3, pp. 994–1013, 2016.

[16] A. Lambert, D. Gruyer, B. Vincke, and E. Seignez, "Consistent outdoor vehicle localization by bounded-error state estimation," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, Saint Louis, MO, USA, Oct. 2009, pp. 1211–1216.

[17] T. Bailey, J. Nieto, and E. Nebot, "Consistency of the FastSLAM algorithm," in *Proc. IEEE Int. Conf. Robot. Automat.*, Beijing, China, May 2006, pp. 424–429.

[18] B. Vincke, A. Lambert, and A. Elouardi, "Guaranteed simultaneous localization and mapping algorithm using interval analysis," in *Proc. Int. Conf. Control Automat. Robot. Vis.*, Singapore, Dec. 2014, 1409–1414.

[19] B. Vincke and A. Lambert, "Experimental comparison of bounded-error state estimation and constraints propagation," in *Proc. IEEE Int. Conf. Robot. Automat.*, Shanghai, China, May 2011, pp. 4724–4729.

[20] E. Seignez, M. Kieffer, A. Lambert, E. Walter, and T. Maurin, "Experimental vehicle localization by bounded-error state estimation using interval analysis," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, Edmonton, AB, Canada, Aug. 2005, pp. 1084–1089.

[21] M. Di Marco, A. Garulli, S. Lacroix, and A. Vicino, "Set membership localization and mapping for autonomous navigation," *Int. J. Robust Nonlinear Control*, vol. 11, no. 7, pp. 709–734, 2001.

[22] L. Jaulin, F. Dabe, A. Bertholom, and M. Legris, "A set approach to the simultaneous localization and map building-application to underwater robots," in *Proc. 4th Int. Conf. Informat. Control, Autom. Robot., Robot. Autom.*, Angers, France, May 2007, pp. 65–69.

[23] L. Jaulin, "Localization of an underwater robot using interval constraint propagation," in *Proc. Int. Conf. Principles Pract. Constraint Program.*, 2006, pp. 244–255.

[24] F. Abdallah, A. Gning, and P. Bonnifait, "Box particle filtering for nonlinear state estimation using interval analysis," *Automatica*, vol. 44, no. 3, pp. 807–815, 2008.

[25] M. Schikora, A. Gning, L. Mihaylova, D. Cremers, and W. Koch, "Box-particle PHD filter for multi-target tracking," in *Proc. 15th Int. Conf. Inf. Fusion*, Singapore, Jul. 2012, pp. 106–113.

[26] N. Petrov, A. Gning, L. Mihaylova, and D. Angelova, "Box particle filtering for extended object tracking," in *Proc. 15th Int. Conf. Inf. Fusion*, Singapore, Jul. 2012, pp. 82–89.
[27] N. Merlinge, K. Dahia, and H. Piet-Lahanier, "A Box regularized particle filter for terrain navigation with highly non-linear measurements," *IFAC-PapersOnLine*, vol. 49, no. 17, pp. 361–366, Sep. 2016.
[28] M. Mallick, V. Krishnamurthy, and B. N. Vo, "Particle filtering combined with interval methods for tracking applications," in *Integrated Tracking, Classification, and Sensor Management: Theory and Applications*, 1st ed. Hoboken, NJ, USA: Wiley, 2012, pp. 43–74.
[29] A. De Freitas, L. Mihaylova, A. Gning, and V. Kadirkamanathan, "Autonomous crowds tracking with box particle filtering and convolution particle filtering," *Automatica*, vol. 69, pp. 380–394, Jul. 2016.
[30] L. Jin, S. Li, B. Hu, M. Liu, and J. Yu, "Noise-suppressing neural algorithm for solving time-varying system of linear equations: A control-based approach," *IEEE Trans. Ind. Inform.*, to be published, doi: 10.1109/TII.2018.2798642.
[31] L. Jin, S. Li, X. Luo, Y. Li, and B. Qin, "Neural dynamics for cooperative control of redundant robot manipulators," *IEEE Trans. Ind. Inform.*, to be published, doi: 10.1109/TII.2018.2789438.
[32] G. Chen, J. Wang, and L. S. Shieh, "Interval Kalman filtering," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 33, no. 1, pp. 250–259, Jan. 1997.
[33] A. Motwani, W. Liu, S. Sharma, R. Sutton, and R. Bucknall, "An interval Kalman filter–based fuzzy multi-sensor fusion approach for fault-tolerant heading estimation of an autonomous surface vehicle," *Proc. Inst. Mech. Eng. M, J. Eng. Maritime Environ.*, vol. 6, no. 22, pp. 222–230, 2015.
[34] N. Li, H. Ma, and C. Yang, "Interval Kalman filter based RFID indoor positioning," in *Proc. IEEE Control Decision Conf.*, May 2016, pp. 6958–6963.
[35] Y. Le, X. F. He, and R. Y. Xiao, "MEMS IMU and GPS/beidou integration navigation system using interval Kalman filter," *Appl. Mech. Mater.*, vols. 568–570, pp. 970–975, Jun. 2014.
[36] I. Ashokaraj, A. Tsourdos, P. Silson, and B. White, "Sensor based robot localisation and navigation: Using interval analysis and extended Kalman filter," in *Proc. IEEE Conf. Control* vol. 2, Jul. 2004, pp. 1086–1093.
[37] G. M. Siouris, G. Chen, and J. Wang, "Tracking an incoming ballistic missile using an extended interval Kalman filter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 33, no. 1, pp. 232–240, Jan. 1997.
[38] X. F. He and B. Vik, "Use of extended interval Kalman filter on integrated GPS/INS system," in *Proc. 12th Int. Tech. Meeting Satellite Division Inst. Navigat.*, 1999, pp. 1907–1914.
[39] R. E. Moore, M. J. Cloud, and R. B. Kearfott, *Introduction to Interval Analysis*, 1st ed. Philadelphia, PA, USA: SIAM, 2009, pp. 1–234.
[40] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis*. London, U.K.: Springer, 2001.
[41] A. Gning, B. Ristic, L. Mihaylova, and F. Abdallah, "An introduction to box particle filtering," *IEEE Signal. Process. Mag.*, vol. 30, no. 4, pp. 166–171, Jul. 2013.
[42] A. Gning, L. Mihaylova, and F. Abdallah, "Mixture of uniform probability density functions for non linear state estimation using interval analysis," presented at the 13th Int. Conf. Inf. Fusion, Edinburgh, U.K., Jul. 2010, pp. 1–8.
[43] A. Gning, B. Ristic, and L. Mihaylova, "Bernoulli particle/box-particle filters for detection and tracking in the presence of triple measurement uncertainty," *IEEE Trans. Signal Process.*, vol. 60, no. 5, pp. 2138–2151, May 2012.
[44] N. Petrov, L. Mihaylova, A. D. Freitas, and A. Gning, "Crowd tracking with box particle filtering," presented at the 17th Int. Conf. Inf. Fusion, Salamanca, Spain, Jul. 2014.
[45] I. K. Kueviakoe, A. Lambert, and P. Tarroux, "Comparison of interval constraint propagation algorithms for vehicle localization," *J. Softw. Eng. Appl.*, vol. 5, no. 12, pp. 157–162, 2012.
[46] Y. Jie, L. Chang-Yun, and L. Zhi-Hui, "A survey of box particle filter theory," *Electron. Opt. Control*, vol. 22, no. 11, pp. 56–60, 2015.
[47] L. Jaulin, M. Kieffer, I. Braems, and E. Walter, "Guaranteed non-linear estimation using constraint propagation on sets," *Int. J. Control*, vol. 74, no. 18, pp. 1772–1782, 2001.
[48] T. Csendes and D. Ratz, "Subdivision direction selection in interval methods for global optimization," *SIAM. J. Numer. Anal.*, vol. 34, no. 3, pp. 922–938, 1997.
[49] D. Ratz and T. Csendes, "On the selection of subdivision directions in interval branch-and-bound methods for global optimization," *J. Global Optim.*, vol. 7, no. 2, pp. 183–207, 1995.
[50] T. Csendes and D. Ratz, "A review of subdivision direction selection in interval methods for global optimization," *Z. AngewandteMath. Mech.*, vol. 76, no. 6, pp. 319–322,1996.
[51] D. Ratz, "Automatische ergebnisverifikation bei globalen optimierungs problemen," Ph.D. dissertation, Dept. Math., Karlsruhe Inst. Technol., Karlsruhe, Germany, 1992.
[52] G. Alefeld and J. Herzberger, *Introduction to Interval Computations*, J. Rokne, Ed. New York, NY, USA: Academic, 1983.
[53] E. Hansen, "Interval arithmetic in matrix computations, Part I," *J. Soc. Ind. Appl. Math.*, vol. 2, no. 2, pp. 308–320, 1965.
[54] W. Zhengang, "Research on some key techniques of MINS/GPS integrated navigation system," Ph.D. dissertation, Dept. Instrum. Meter Eng., Southeast Univ., Nanjing, China, Aug. 2005.
[55] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, 1st ed. Cambridge, MA, USA: MIT Press, 2005, pp. 52–57.
[56] J. Nieto, J. Guivant, E. Nebot, and S. Thrun, "Real time data association for FastSLAM," in *Proc. IEEE ICRA*, vol. 1, Sep. 2003, pp. 412–418.
[57] 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, Aug. 2008.
[58] D. Liu, J. Duan, and H. Shi, "A strong tracking square root central difference FastSLAM for unmanned intelligent vehicle with adaptive partial systematic resampling," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 11, pp. 3110–3120, Nov. 2016.
[59] E. Nebot, J. Guivant, and J. Nieto. (Jul. 28, 2012). *ACFR, Experimental Outdoor Dataset*. [Online]. Available: http://www.acfr.usyd.edu.au/home-pages/academic/enebot/dataset.htm
[60] T. Bailey. (Mar. 10, 2011). *SLAM Simulations*. [Online]. Available: http://www-personal.acfr.usyd.edu.au/tbailey/software/
[61] D. Ribas, "Towards simultaneous localization and mapping for an auv using an imaging sonar," Ph.D. dissertation, Dept. Electron., Inform. Automat., Univ. Girona, Girona, Spain, 2005.
[62] D. Hahnel, W. Burgard, D. Fox, and S. Thrun, "An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 1, 2003, pp. 206–211.
[63] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 34–46, Feb. 2007.
[64] T. Wang, Y. Chen, M. Zhang, J. Chen, and H. Snoussi, "Internal transfer learning for improving performance in human action recognition for small datasets," *IEEE Access*, vol. 5, pp. 17627–17633, 2017.
[65] T. Wang, Y. Chen, M. Qiao, and H. Snoussi, "A fast and robust convolutional neural network-based defect detection model in product quality control," *Int. J. Adv. Manuf. Technol.*, vol. 94, pp. 3465–3471, Feb. 2018.

JINGWEN LUO received the bachelor's degree in process equipment and control engineering and the master's degree in control theory and control engineering from the Kunming University of Science and Technology, Kunming, China, in 2005 and 2009, respectively. He is currently pursuing the Ph.D. degree with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China.

Since 2009, he has been a Lecturer with the School of Information Science and Technology, Yunnan Normal University.

SHIYIN QIN received the bachelor's and master's degrees for engineering science in automatic controls and industrial systems engineering from Lanzhou Jiaotong University, Lanzhou, China, in 1978 and 1984, respectively, and the Ph.D. degree in industrial control engineering and intelligent automation from Zhejiang University, Zhejiang, China, in 1990. He is currently a Professor with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China.