Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

# Distributed kernel mean embedding Gaussian belief propagation for underwater multi-sensor multi-target passive tracking*

Dengpeng YANG, Yunfei GUO†‡, Yanbo XUE, Anke XUE, Yun CHEN

*School of Automation, Hangzhou Dianzi University, Hangzhou 310018, China*

†E-mail: gyf@hdu.edu.cn

Received Apr. 2, 2025; Revision accepted Aug. 14, 2025; Crosschecked

**Abstract:** To address the problem of underwater multi-sensor multi-target passive tracking in clutter, a distributed kernel mean embedding-based Gaussian belief propagation (DKME-GaBP) algorithm is proposed. First, the joint posterior probability density function (pdf) is established and factorized, and it is represented by the corresponding factor graph. Then, the GaBP algorithm is run on this factor graph to reduce the computational complexity of data association. The factor graph of GaBP consists of inner and outer loops. The inner loop is responsible for local track estimation and data association. The outer loop fuses information from different sensors. For the inner loop, the kernel mean embedding (KME) with a Gaussian kernel is designed to transform the strong nonlinear problem of local estimation into a linear problem in a high-dimensional reproducing kernel Hilbert space (RKHS). For the outer loop, a multi-sensor distributed fusion method based on KME is proposed to improve fusion accuracy by accounting for the distance among different pdfs in RKHS. The effectiveness and robustness of DKME-GaBP are validated in simulation.

Unedited

**Key words:** Kernel mean embedding; Belief propagation; Multi-sensor multi-target tracking; Underwater passive tracking

https://doi.org/10.1631/FITEE.2500204 **CLC number:**

## 1 Introduction

Underwater passive tracking (UPT) is a key technology in ocean detection and is widely applied in underwater navigation, maritime security, resource exploration, and environmental monitoring (Chen et al., 2018; Tian et al., 2024). UPT can achieve silent tracking by passively receiving the target's bearing information, providing higher safety and concealment compared to active tracking (Northardt and Nardone, 2019; Wolek et al., 2022). However, UPT also faces numerous challenges, such as incomplete observability, strong nonlinearity and poor real-time performance (Badriasl et al., 2020; Jiang et al., 2022), which often result in unsatisfactory tracking performance.

To solve the problem of strong nonlinearity in UPT, traditional studies have developed some nonlinear filters including the extended Kalman filter (EKF) (Xiao and Hill, 1996) and its variants such as the modified polar coordinate EKF (MPCEKF) (Aidala and Hammel, 1983) and the modified gain EKF (MG-EKF) (Rao, 2005). The unscented Kalman filter (UKF) (Li et al., 2016b) and the particle filter (PF) (Li et al., 2016a) are presented to estimate the target state by a weighted set of sigma points or random samples, thereby obtaining a more accurate estimation. The adaptive kernel

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 62371173, U22A2047, and U22A2044) and the Stable Supporting Fund of Acoustic Science and Technology Laboratory (No. JCKYS2024604SSJS009)

ORCID: Yunfei GUO, https://orcid.org/0000-0001-7887-4312

© Zhejiang University Press 2025

Kalman filter (AKKF) is proposed in (Sun et al., 2023, 2022). It transforms strong nonlinearity into a linear problem in a high-dimensional feature space through kernel mean embedding (KME) (Song et al., 2013). However, the employed quadratic or quartic kernel fails to capture all the features of the target of the strong nonlinear problem in UPT.

In UPT, multi-sensor fusion is an effective method to address the problem of incomplete observability. Additional information about the target is provided by the spatial distribution of sensors. Multi-sensor fusion methods can be categorized into two types centralized fusion and distributed fusion. Because the centralized fusion methods such as triangulation techniques are prone to issues such as ghost points(Choi et al., 2014), distributed fusion is used in this paper. In distributed fusion, local tracks are formed by each sensor and then fused at the center, offering strong fault tolerance and scalability(Cao et al., 2024). However, in distributed tracking, especially in underwater multi-target passive tracking with heavy clutter, the computation burden is greatly increased, leading to poor real-time performance(Xu et al., 2024). Reducing the computational complexity of data association is therefore considered essential. Traditional underwater multi-target passive tracking methods, such as the joint probability data association filter (JPDAF) (Tian et al., 2023), the probability hypothesis density (PHD) filter, the cardinalized PHD (CPHD) filter (Kim, 2024), and the multi-Bernoulli (MB) filter and its variants (Gunes and Guldogan, 2016) are well-established, offering reliable solutions for data association. However, they all have high complexity(Williams and Lau, 2014), leading to a poor real-time performance. In (Meyer et al., 2017), marginal posterior probabilities are computed efficiently by brief propagation (BP), whose computational complexity increases linearly in the number of sensors and quadratically in the number of targets (Meyer et al., 2018). Furthermore, a Gaussian version of BP is proposed by our research group (Xue et al., 2025). In (Yan et al., 2024), GaBP achieves a closed-form Gaussian message passing on the factor graph, further reducing memory requirements and computational complexity.

To address the problems of poor real-time performance, strong nonlinearity, and poor observability in UPT, a distributed kernel mean embedding-based Gaussian belief propagation (DKME-GaBP) multi-sensor multi-target passive tracking algorithm is proposed. First, the association problem between targets and measurements is solved by a closed-form Gaussian message passing on the factor graph with inner and outer loops. Next, in the inner loop, the strong nonlinearity between measurements and target states is transformed into a linear problem in a high-dimensional feature space by using KME with a Gaussian kernel, enabling the efficient estimation of target states. Finally, in the outer loop, within the distributed fusion framework of GaBP, information obtained from different sensors is fused by KME in reproducing kernel Hilbert space (RKHS), thereby addressing the issue of partially unobservable target states, and estimation accuracy is improved. In DKME-GaBP, the tracking accuracy and real-time performance are improved. The main contributions of this paper are as follows

- To solve inherent data association problems in underwater multi-target passive tracking while maintaining low computational complexity in registering measurements with targets, a DKME-GaBP algorithm is developed. This method factorizes the joint probability density function (pdf) and represents it using a factor graph. Messages are passed among the nodes in the factor graph using the mean and covariance of the Gaussian distribution. The covariance of the Gaussian distribution replaces the covariance sampled from the original KME, leading to more accurate state estimates.

- A Gaussian kernel-based KME method is developed to address strong nonlinear problems in local estimation within the inner loop. Since the choice of kernel is crucial for KME, the infinite-dimensional form of the Gaussian kernel is provided to capture more characteristics of the target. Additionally, the hyperparameter of the Gaussian kernel used in DKME-GaBP is designed.

- A distributed multi-sensor fusion solution is provided by the DKME-GaBP. A passive multi-sensor fusion approach based on KME is developed by minimizing the sum of squared distances between the fused probability density and each local probability density obtained from

Unedited

DKME-GaBP. The limitations of single-sensor systems are addressed through the integration of data from multiple sensors using DKME-GaBP.

The structure of this paper is as follows. In Section 2, the dynamic model and measurement model are described. In Section 3, the factor graph of the proposed algorithm is established, and the proposed DKME-GaBP algorithm is developed. The simulation results of the proposed algorithm are presented in Section 4, and conclusions are drawn in Section 5.

## 2 Problem formulation

### 2.1 Dynamic model

Assume that there are $K$ targets appearing in the surveillance region, each with time-varying states. Owing to the possible appearance or disappearance of targets between two successive measurement instants, the number of targets $K$ is time-varying and unknown to the tracker. The state of target $k$, $k \in \mathcal{K} \triangleq \{1,\ldots,K\}$, denoted as $\mathbf{x}_{n,k} \triangleq [x_{n,k}, y_{n,k}, \dot{x}_{n,k}, \dot{y}_{n,k}]^{\mathrm{T}}$, consists of the position and velocity at time $n$. The joint state of all targets at time $n$ is denoted as $\mathbf{x}_n \triangleq [\mathbf{x}_{n,1}^{\mathrm{T}}, \cdots, \mathbf{x}_{n,K}^{\mathrm{T}}]^{\mathrm{T}}$. Assuming that all target states evolve independently and follow a Markov dynamic model, the joint state transition function for all targets at time $n$ can be factored as

$$f(\mathbf{x}_n|\mathbf{x}_{n-1}) = \prod_{k=1}^{K} f(\mathbf{x}_{n,k}|\mathbf{x}_{n-1,k}). \tag{1}$$

where $f(\mathbf{x}_{n,k}|\mathbf{x}_{n-1,k})$ is the state transition function at time $n$. A prior pdf $f(\mathbf{x}_0)$ is given for each target at $n=0$; the joint state pdf $f(\mathbf{x})$ can be denoted as

$$f(\mathbf{x}) = \prod_{k=1}^{K} f(\mathbf{x}_{0,k}) \prod_{n=1}^{N} f(\mathbf{x}_{n,k}|\mathbf{x}_{n-1,k}). \tag{2}$$

### 2.2 Measurement model

Assume that there are $S$ sensors with index $s$ in the surveillance region. Each target $k$ is assumed to generate at most one measurement at sensor $s$, $s \in \{1,\ldots,S\}$, and each measurement can be generated by at most one target. Thus, each measurement is either from a target or a false alarm. Assume that sensor $s$ generates $M_n^{(s)}$ bearing measurement $\mathbf{z}_{n,m}^{(s)}$ at time $n$, $m \in \{1,\ldots,M_n^{(s)}\}$. The $m^{\mathrm{th}}$ measurement $\mathbf{z}_{n,m}^{(s)}$ is modeled as

$$\mathbf{z}_{n,m}^{(s)} = \begin{cases} h^{(s)}(\mathbf{x}_n, w_n^{(s)}), & \text{detection} \\ u_{n,m}^{(s)}, & \text{false alarm} \end{cases} \tag{3}$$

$$h^{(s)}(\mathbf{x}_n, w_n^{(s)}) = \tan^{-1}(x_{n,k}, y_{n,k}) + w_n^{(s)}, \tag{4}$$

where $w_n^{(s)}$ represents the measurement noise. Then, all measurements are stacked as $\mathrm{z}_n^{(s)} \triangleq [\mathrm{z}_{n,1}^{(s)\mathrm{T}}, \cdots, \mathrm{z}_{n,M_n^{(s)}}^{(s)\mathrm{T}}]^{\mathrm{T}}$ and $\mathrm{z}_n \triangleq [\mathrm{z}_n^{(1)\mathrm{T}}, \cdots, \mathrm{z}_n^{(\mathrm{S})\mathrm{T}}]^{\mathrm{T}}$. Assuming that the false alarm follows a uniform distribution in space with pdf $f_{\mathrm{FA}}(\mathbf{z}_{n,m}^{(s)})$, and follows a Poisson distribution in cardinality with pdf $\mu^{(s)}$.

The objective of this paper is to estimate the target states $\mathbf{x}_{n,k}$ from $\mathrm{z}_n$, in the sense of minimum mean square error (MMSE), which is denoted as

$$\hat{\mathbf{x}}_{n,k}^{\mathrm{MMSE}} = \int \mathbf{x}_{n,k} f(\mathbf{x}_{n,k}|r_{n,k}=1,\mathbf{z}) \mathrm{d}\mathbf{x}_{\mathrm{n,k}}. \tag{5}$$

where $r_{n,k}$ is a bigram and is used to denote the target existence probability, with $r_{n,k}=1$ indicating that the target exists, $f(\mathbf{x}_{n,k}|r_{n,k}=1,\mathbf{z})$ is the marginal posterior pdf of target $k$ when it presents(Meyer et al., 2017). Other issues are described in Section 3.

## 3 DKME-GaBP algorithm

This section presents multi-sensor multi-target DKME-GaBP algorithm is presented in this section. Subsections 3.1-3.4 introduce the BP algorithm. Subsection 3.5 describes the factor graph of DKME-GaBP, and the final Subsection3.6 derives the DKME-GaBP algorithm. The flowchart of DKME-GaBP is shown in Fig. 1, and it describes the specific process of the DKME-GaBP algorithm. The inner loop in the diagram performs data association and local track estimation, while the outer loop conducts information fusion.

### 3.1 Association variable

To describe the association uncertainty between the targets and the measurements, two association variables $\mathbf{a}$ and $\mathbf{b}$ are defined. The target-oriented association vectors for sensor $s$ at time $n$ are described by a random vector $\mathbf{a}_n^{(s)} = [a_{n,1}^{(s)}, \cdots, a_{n,K}^{(s)}]^{\mathrm{T}}$,

Unedited

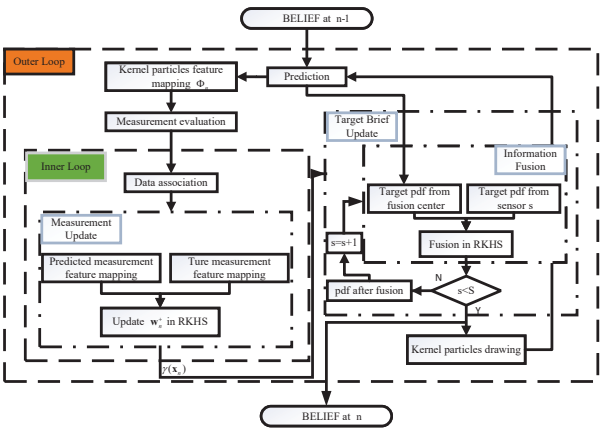


**Fig. 1 Flowchart of DKME-GaBP algorithm.**

where the $k$-th element is defined as

$$a_{n,k}^{(s)} \triangleq \begin{cases} m \in \{1,\ldots,M_n^{(s)}\}, \text{target } k \text{ yields a} \\ \quad \text{measurement } m \text{ at time } n, \\ 0, \text{target } k \text{ is not detected} \\ \quad \text{by sensor } s \text{ at time } n. \end{cases} \tag{6}$$

It can be stacked as $\mathbf{a}_n \triangleq [\mathbf{a}_n^{(1)\mathrm{T}}, \cdots, \mathbf{a}_n^{(S)\mathrm{T}}]^{\mathrm{T}}$. Also, the measurement-oriented association vector is defined as $\mathbf{b}_n^{(s)} = [b_{n,1}^{(s)}, \cdots, b_{n,M_n^{(s)}}^{(s)}]^{\mathrm{T}}$, where the $k$-th term is

$$b_{n,m}^{(s)} \triangleq \begin{cases} k \in \{1,\ldots,K\}, \text{measurement } m \text{ from} \\ \quad \text{sensor yielded by target } k \text{ at time } n \\ 0, \text{measurement } m \text{ from sensor} \\ \quad \text{is a false alarm at time } n \end{cases} \tag{7}$$

### 3.2 Augmented target states

The number of targets to the tracker is unknown, and all possible surviving targets are referred to as potential targets (PTs). We define a vector $\mathbf{r}_n = [r_{n,1}, \cdots, r_{n,K}]^{\mathrm{T}}$ to indicate the existence of PTs. $r_{n,k} = 1$ indicates target exists and $r_{n,k} = 0$ means target disappears. For convenience, the state of the PTs is augmented and defined as $\mathcal{A}_{n,k} \triangleq [\mathbf{x}_{n,k}^{\mathrm{T}}, r_{n,k}]^{\mathrm{T}}$, and the joint augmented state is defined as $\mathcal{A}_n \triangleq \left[\mathcal{A}_{n,1}^{\mathrm{T}}, \cdots, \mathcal{A}_{n,k}^{\mathrm{T}}\right]^{\mathrm{T}}$.

According to Bayes' rule and the independence assumption, given the prior pdfs $f(\mathcal{A}_{0,k}) = f(\mathbf{x}_{0,k}, r_{0,k})$, the pdf of the augmented state $\mathcal{A}$ can be factorized as

$$f(\mathcal{A}) = \prod_{k=1}^{K} f(\mathcal{A}_{0,k}) \prod_{n'=1}^{n} f(\mathcal{A}_{n',k}|\mathcal{A}_{n'-1,k}). \tag{8}$$

thus, $f(\mathcal{A}_{n,k}|\mathcal{A}_{n-1,k}) = f(\mathbf{x}_{n,k}, r_{n,k}|\mathbf{x}_{n-1,k}, r_{n-1,k})$ is given as

$$f(\mathbf{x}_{n,k}, r_{n,k}|\mathbf{x}_{n-1,k}, r_{n-1,k}) = \begin{cases} (1-p_{n,k}^{\mathrm{b}}) f_{\mathrm{D}}(\mathbf{x}_{n,k}), & r_{n,k}=0,\ r_{n-1,k}=0 \\ p_{n,k}^{\mathrm{b}} f_{\mathrm{b}}(\mathbf{x}_{n,k}), & r_{n,k}=1,\ r_{n-1,k}=0 \\ (1-p_{n,k}^{\mathbf{s}}) f_{\mathrm{D}}(\mathbf{x}_{n,k}), & r_{n,k}=0,\ r_{n-1,k}=1 \\ p_{n,k}^{\mathbf{s}} f(\mathbf{x}_{n,k}|\mathbf{x}_{n-1,k}), & r_{n,k}=1,\ r_{n-1,k}=1 \end{cases} \tag{9}$$

where $p_{n,k}^{\mathbf{b}}$ and $p_{n,k}^{\mathbf{s}}$ are the birth and the survival probability, respectively. $f_{\mathbf{b}}(\mathbf{x}_{n,k})$ is birth pdf.

### 3.3 Prior distributions

In Subsection 3.1, the association variables $\mathbf{a}_n^{(s)}$ and $\mathbf{b}_n^{(s)}$ are introduced. They can fully and clearly represent the events occurring at time $n$ at sensor $s$, and thus both $\mathbf{a}$ and $\mathbf{b}$ can be used to represent association events. Assume that a prior pdf $f(\mathbf{x}_{0,k})$ is available for each PT state $\mathbf{x}_{0,k}$ at time $n$. Next, the prior probability mass function (pmf) of the association variable $\mathbf{a}_n^s$ can be modeled as

$$p(\mathbf{a}_n^{(s)}) \propto \psi(\mathbf{a}_n^{(s)}) \prod_{k=1}^{K} \varphi(\mathbf{x}_{n,k}, r_{n,k}, a_{n,k}^{(s)}). \tag{10}$$

where $\psi(\,\cdot\,)$ is the exclusion indicator function, denoting that each target can generate at most one measurement at each sensor, and each sensor measurement can be generated by at most one target. Its formula is expressed as

$$\psi(\mathbf{a}_n^{(s)}) = \begin{cases} 0, & \exists k, k' \in \{1,\ldots,K\} \quad \text{s.t.} a_{k,n}^{(s)} = a_{k',n}^{(s)} \\ 1, & \text{others.} \end{cases} \tag{11}$$

The remaining terms in Eq.(10) are

$$\varphi(\mathbf{x}_{n,k}, 1, a_{n,k}^{(s)}) = \begin{cases} \frac{P_d^{(s)}(\mathbf{x}_{n,k})}{\mu^{(s)}}, & a_{n,k}^{(s)} \in M_n^{(s)} \\ 1 - P_d^{(s)}(\mathbf{x}_{n,k}), & a_{n,k}^{(s)} = 0 \end{cases} \tag{12}$$

$$\varphi(\mathbf{x}_{n,k}, 0, a_{n,k}^{(s)}) = 1(a_{n,k}^{(s)}).$$

where $1(\cdot)$ is the indicator for $a = 0$ (see Eq.(6)). Since there is a one-to-one relationship between $\mathbf{a}_n^{(s)}$ and $\mathbf{b}_n^{(s)}$, $p(\mathbf{a}_n^{(s)})$ can be replaced with $p(\mathbf{a}_n^{(s)}, \mathbf{b}_n^{(s)})$ and be rewritten as(Zhang and Meyer, 2024)

$$\psi(\mathbf{a}_n^{(s)}, \mathbf{b}_n^{(s)}) = \prod_{k=1}^{K} \prod_{m=1}^{M_n^{(s)}} \Psi(a_{n,k}^{(s)}, b_{n,m}^{(s)}), \tag{13}$$

$$\Psi(a_{n,k}^{(s)}, b_{n,m}^{(s)}) = \begin{cases} 0, & a_{n,k}^{(s)} = m, b_{n,m}^{(s)} \neq k \\ & \text{or } b_{n,m}^{(s)} = k, a_{n,k}^{(s)} \neq m \\ 1, & \text{others.} \end{cases} \tag{14}$$

Unedited

thus, Eq.(10) can be rewritten as

$$\begin{aligned} p(a_{n,k}^{(s)},b_{n,m}^{(s)}) &\propto \psi(\mathbf{a}_n^{(s)},\mathbf{b}_n^{(s)}) \prod_{k=1}^{K} \varphi(a_{n,k}^{(s)}) \\ &= \prod_{k=1}^{K} \varphi(\mathbf{x}_{n,k}, r_{n,k}, a_{n,k}^{(s)}) \prod_{m=1}^{M_n^s} \Psi(a_{n,k}^{(s)}, b_{n,m}^{(s)}). \end{aligned} \tag{15}$$

### 3.4 Global likelihood function

According to the independence assumption, the global likelihood function can be decomposed as

$$\begin{aligned} f(\mathbf{z}_n|\mathcal{A}_n,\mathbf{a}_n) = &\prod_{s=1}^{S} \left( \prod_{m=1}^{M_n^{(s)}} f_{\mathrm{FA}}\left(\mathbf{z}_{n,m}^{(s)}\right) \right) \\ &\times \prod_{k\in\mathcal{D}_{a_n,r_n}^{(s)}} \frac{f\left(\mathbf{z}_{n,a_{n,k}^{(s)}}^{(s)} \middle| \mathbf{x}_{n,k}\right)}{f_{\mathrm{FA}}\left(\mathbf{z}_{n,a_{n,k}^{(s)}}^{(s)}\right)} \end{aligned} \tag{16}$$

where $\mathcal{D}_{a_n,r_n}^{(s)} \triangleq \{k \in \{1,\ldots,K\} : r_{n,k} = 1, a_{n,k}^{(s)} \neq 0\}$ represents the set of surviving targets detected by sensor $s$, $f\left(\mathbf{z}_{n,m}^{(s)}|\mathbf{x}_{n,k}\right)$ is the likelihood function, and $f_{\mathrm{FA}}\left(\mathbf{z}_{n,m}^{(s)}\right)$ represents the false alarm distribution within the detection range of sensor $s$. Then, the global likelihood function can be rewritten as

$$f(\mathbf{z}_n|\mathcal{A}_n,\mathbf{a}_n) \propto \prod_{s=1}^{S}\prod_{k=1}^{K} g\left(\mathbf{x}_{n,k}, r_{n,k}, a_{n,k}^{(s)}; \mathbf{z}_n^{(s)}\right), \tag{17}$$

where

$$\begin{aligned} g\left(\mathbf{x}_{n,k}, 1, a_{n,k}^{(s)}; \mathbf{z}_n^{(s)}\right) &= \begin{cases} \frac{f\left(\mathbf{z}_{n,m}^{(s)}\middle|\mathbf{x}_{n,k}\right)}{f_{\mathrm{FA}}\left(\mathbf{z}_{n,m}\right)}, & a_{n,k}^{(s)} \in \mathcal{M}_n^{(s)} \\ 1, & a_{n,k}^{(s)} = 0 \end{cases} \\ g\left(\mathbf{x}_{n,k}, 0, a_{n,k}^{(s)}; \mathbf{z}_n^{(s)}\right) &= 1. \end{aligned} \tag{18}$$

### 3.5 The factor graph of DKME-GaBP

The marginal posterior pdf $f(\mathbf{x}_{n,k}|r_{n,k}=1,\mathbf{z})$ involved in Eq.(5) can be directly marginalized from $f(\mathcal{A},\mathbf{a}|\mathbf{z})$ of the total target vector $\mathbf{x}$. By passing messages in a factor graph, an approximate marginalization can be obtained, with total complexity growing linearly with time (Zhang and Meyer, 2024). Then, $f(\mathcal{A},\mathbf{a}|\mathbf{z})$ can be expressed as

$$\begin{aligned} f(\mathcal{A},\mathbf{a}|\mathbf{z}) &\propto f(\mathbf{z}|\mathcal{A},\mathbf{a}) f(\mathcal{A},\mathbf{a}) \\ &= f(\mathcal{A}_0) \prod_{n'=1}^{n} f(\mathbf{z}_{n'}|\mathcal{A}_{n'},\mathbf{a}_{n'}) f(\mathcal{A}_{n'}|\mathcal{A}_{n'-1}) \\ &\times p(\mathbf{a}_{n'}) \propto \left( \prod_{k'=1}^{K} f(\mathcal{A}_{0,k'}) \right) \prod_{n'=1}^{n} p(\mathbf{a}_{n'}) \times \\ &\prod_{k=1}^{K} g(\mathcal{A}_{n',k}, a_{n',k}; \mathbf{z}_{n'}) f(\mathcal{A}_{n',k}|\mathcal{A}_{n'-1,k}). \end{aligned} \tag{19}$$

According to Eq.(15), Eq.(19) can be replaced with $f(\mathcal{A},\mathbf{a},\mathbf{b}|\mathbf{z})$, specifically by substituting $p(a_n,b_n)$ for $p(a_n)$. Therefore, utilizing Bayes' rule and independence assumptions, the joint posterior pdfs can be expressed as

$$\begin{aligned} f(\mathcal{A},\mathbf{a},\mathbf{b}|\mathbf{z}) &\propto f(\mathbf{z}|\mathcal{A},\mathbf{a},\mathbf{b}) f(\mathcal{A},\mathbf{a},\mathbf{b}) \\ &\propto \left( \prod_{k'=1}^{K} f(\mathcal{A}_{0,k'}) \right) \prod_{n'=1}^{n}\prod_{k=1}^{K} f(\mathcal{A}_{n',k}|\mathcal{A}_{n'-1,k}) \\ &\times \prod_{s=1}^{S} g\left(\mathbf{x}_{n,k}, r_{n,k}, a_{n,k}^{(s)}; \mathbf{z}_n^{(s)}\right) \\ &\times \varphi(\mathbf{x}_{n,k}, r_{n,k}, a_{n,k}^{(s)}) \times \prod_{m=1}^{M_{n'}^{(s)}} \Psi\left(a_{n',k}^{(s)}, b_{n',m}^{(s)}\right), \end{aligned} \tag{20}$$

the factor graph of the above expression is given in Fig. 2. For simplicity, some shorthand notations are described in Subsection 3.6.

### 3.6 The DKME-GaBP algorithm

This section presents the DKME-GaBP algorithm. The first four subsections are dedicated to data association and local track estimation in the inner loop, while the final subsection is focused on multi-sensor fusion in the outer loop. Consider the following model

$$f(\mathbf{x}_{n,k}|\mathbf{x}_{n-1,k},\mathbf{z}_{n-1,k}) \to \mu_{n,k}^{-} = \Phi_{n,k}\mathbf{w}_{n,k}^{-}, \tag{21}$$

$$f(\mathbf{x}_{n,k}|\mathbf{x}_{n-1,k},\mathbf{z}_{n,k}) \to \mu_{n,k}^{+} = \Phi_{n,k}\mathbf{w}_{n,k}^{+}, \tag{22}$$

$$f(\mathbf{z}_{n,k}|\mathbf{x}_{n,k}) = \mathcal{N}(\mathbf{h}_n(\mathbf{x}_{n,k}), \mathbf{R}) \tag{23}$$

where $\mathbf{F}$ is the state transition matrix, the likelihood function is approximated by a first-order Taylor series expansion. $\mathbf{G}$ is the process noise matrix, $\mu_{n,k}$ indicates the importance distribution of that target, $\mathbf{w}_{n,k}$ represents the weights of the particles in KME, and $\Phi_{n,k} = \left[\phi_x(x_{n,k}^{\{1\}}),\ldots,\phi_x(x_{n,k}^{\{M\}})\right]$

Unedited

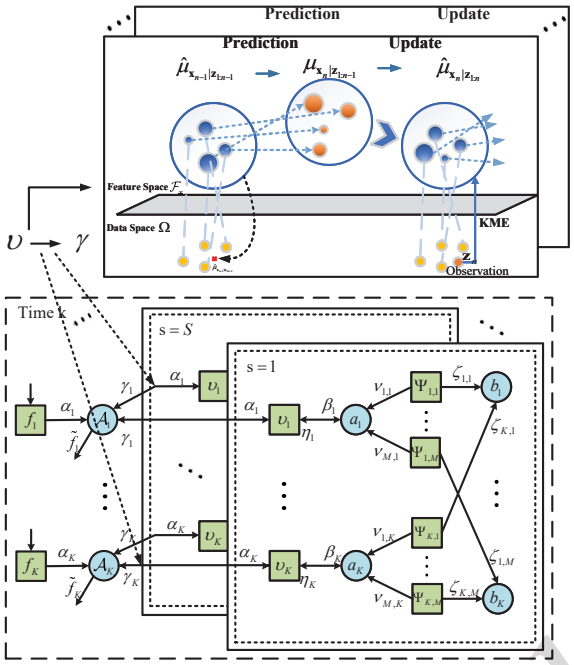


**Fig. 2 Factor graph representing the factorization of the DKME-GaBP joint posterior pdf in Eq.**(20)

denotes the set of feature mappings of the KME particles. Where $M$ is the number of kernel particles. Here, $\phi_{\mathbf{x}}(\mathbf{x}) = k(\mathbf{x}, \cdot)$ represents the feature mapping of each $\mathbf{x} \in \Omega$ in the RKHS using kernel. Common kernels include the polynomial kernel $k(\mathbf{x}, \mathbf{x}') = (\langle \mathbf{x}, \mathbf{x}' \rangle + c)^d$, the Gaussian kernel $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right)$, and so on(Gebhardt et al., 2019).

$$\mu_X = \mathbb{E}_X[\phi(X)] = \int_{\Omega} \phi(\mathbf{x}) dP(\mathbf{x}), \tag{24}$$

where $X$ denotes a random variable with domain $\Omega$ and distribution $P(X)$, and $\mathbf{x}$ represents its instance.

### 3.6.1 Target state prediction

Given the target belief at the previous step, the prior information of target can be obtained as

$$\alpha(\mathbf{x}_{n,k}, r_{n,k}) = \sum_{r_{n-1,k} \in \{0,1\}} \int f(\mathbf{x}_{n,k}, r_{n,k} | \mathbf{x}_{n-1,k}, r_{n-1,k}) \times \tilde{f}(\mathbf{x}_{n-1,k}, r_{n-1,k}) \mathrm{d}\mathbf{x}_{n-1,k}, \tag{25}$$

where $\tilde{f}(\mathbf{x}_{n-1,k}, r_{n-1,k})$ is the final result of the outer loop at time $n-1$. By substituting Eq.(9) into $f(\mathbf{x}_{n,k}, r_{n,k} | \mathbf{x}_{n-1,k}, r_{n-1,k})$, we can obtain the representation of $\alpha(\mathbf{x}_{n,k}, r_{n,k})$

$$\alpha(\mathbf{x}_{n,k}, 1) = p^{\mathrm{b}}_{n,k} f_{\mathrm{b}}(\mathbf{x}_{n,k}) \tilde{f}_{n-1,k} + p^{\mathrm{s}}_{n,k} \times \int f(\mathbf{x}_{n,k} | \mathbf{x}_{n-1,k}) \tilde{f}(\mathbf{x}_{n-1,k}, 1) \, \mathrm{d}\mathbf{x}_{n-1,k}, \tag{26}$$

$$\alpha(\mathbf{x}_{n,k}, 0) = \left(1 - p^{\mathrm{b}}_{n,k}\right) \tilde{f}_{n-1,k} + \left(1 - p^{\mathrm{s}}_{n,k}\right) \times \int \tilde{f}(\mathbf{x}_{n-1,k}, 1) \, \mathrm{d}\mathbf{x}_{n-1,k}. \tag{27}$$

Under the Gaussian condition and kernel mean condition, the above formula can be rewritten as

$$\alpha(\mathbf{x}_{n,k}, 1) == \mathcal{N}(\mathbf{x}_{n|n-1,k}, \mathbf{P}_{n|n-1,k}) = \Phi_{n,k} \mathbf{w}^{-}_{n,k} \tag{28}$$

where $\mathbf{x}_{n|n-1,k}$ and $\mathbf{P}_{n|n-1,k}$ can be predicted from Eqs(1) and (2).

In the framework of KME, the prediction step from time $n-1$ to time $n$ can be expressed as

$$\mathbf{x}^{\{i\}}_{n,k} = f_k(\tilde{\mathbf{x}}^{\{i\}}_{n-1,k}, \mathbf{u}^{\{i\}}_{n-1,k}), i = 1, \ldots, M \tag{29}$$

$$\mathbf{w}^{-}_{n,k} = \mathcal{M} \tilde{\mathbf{w}}^{+}_{n-1,k} \tag{30}$$

$$S^{-}_{n,k} = \mathcal{M} \tilde{S}^{+}_{n-1,k} \mathcal{M} + V_{n,k}. \tag{31}$$

where $\tilde{\mathbf{x}}^{\{i=1:M\}}_{n-1|n-1,k} \sim \mathcal{N}(\mathbf{x}_{n-1|n-1,k}, \mathbf{P}_{n-1|n-1,k})$ is the KME proposal particle at the previous time step, sampled from a Gaussian distribution. $\Phi_{n,k} = [\phi_{\mathbf{x}}(\mathbf{x}^{\{1\}}_{n,k}), \ldots, \phi_{\mathbf{x}}(\mathbf{x}^{\{M\}}_{n,k})]$ is the feature mapping in the feature space. $\mathbf{w}^{+}_{n-1,k}$ and $\mathbf{w}^{-}_{n,k}$ represent the posterior kernel weight vector and the prior predicted kernel weight vector for the target $k$, respectively. $\mathcal{M} = (K_{\underline{x}\underline{x}} + \kappa I)^{-1} K_{\underline{x}x}$ is the kernel transition matrices derived from the state after training at the previous time and the current training state, $K_{\underline{x}\underline{x}} = \Psi^{\mathrm{T}}_{n-1,k} \Phi_{n-1,k}$ and $K_{\underline{x}x} = \Psi^{\mathrm{T}}_{n-1,k} \Phi_{n-1,k}$ are the transition Gram matrices, and $\Psi_{n-1,k}$ is the feature mapping of the previous state after training. $S^{-}_{n,k}$ is the prior weight covariance matrix, and $V_{n,k}$ is the transition residual matrix(Sun et al., 2023).

Additionally, if the target $k$ is newborn at time $n$, the initial predicted weight is given by $w_{n,k} = \frac{1}{M} I_{M \times 1}$, the initial predicted covariance is $S^{-}_{n,k} = I_{M \times M}$, and the initial predicted state of the target follows the birth pdf $f_b(\mathbf{x}_{n,k})$.

The kernel selection is crucial for mapping features in UPT. Gaussian kernel handles nonlinear data well and captures subtle feature differences, while the polynomial kernel is better at capturing feature interactions(Shen et al., 2010). Additionally, the Gaussian kernel has lower computational

Unedited

complexity compared to polynomial kernels of degree higher than two. Thus, the Gaussian kernel is more suitable for UPT. The Gaussian kernel feature mapping and the Gaussian kernel with $\sigma$ are given as

$$\phi(x)=\exp\left(-\frac{x^2}{2\sigma^2}\right)\cdot\left[1,\frac{x}{\sigma},\frac{x^2}{\sqrt{2!}\sigma^2},\ \ldots\right]^T, \tag{32}$$

$$k(x,x')=\exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right). \tag{33}$$

Thus, Eq.(24) can be rewritten as

$$\mu_X=\mathbb{E}_X\left(\exp\left(-\frac{x^2}{2\sigma^2}\right)\right)\cdot\left[1,\frac{x}{\sigma},\frac{x^2}{\sqrt{2!}\sigma^2},\ldots\right]^T. \tag{34}$$

where the hyperparameter $\sigma\in[0,1]$ plays a crucial role. Sensitivity to sample variations is reduced with larger values, while smaller values cause the kernel function to change more rapidly, allowing for the capture of more detailed information. Therefore, in complex underwater environments, smaller hyperparameter values are more suitable.

### 3.6.2 Measurement evaluation

Following the message passing protocol of the factor graph, the measurement evaluation message can be given as

$$\begin{aligned}&\beta(a_{n,k}^{(s)})=\\&\sum_{r_{n,k}\in\{0,1\}}\int v\big(\mathbf{x}_{n,k},r_{n,k},a_{n,k}^{(s)};\mathbf{z}_n^{(s)}\big)\times\alpha(\mathbf{x}_{n,k},r_{n,k})\mathrm{d}\mathbf{x}_{n,k}\\&=\int v\big(\mathbf{x}_{n,k},1,a_{n,k}^{(s)};\mathbf{z}_n^{(s)}\big)\alpha(\mathbf{x}_{n,k},1)\mathrm{d}\mathbf{x}_{n,k}+1\big(a_{n,k}^{(s)}\big)\alpha_{n,k}.\end{aligned} \tag{35}$$

where $\beta\big(a_{n,k}^{(s)}\big)$ is the message from the factor node $g_{n,k}^{(s)}$ to the variable node $a_{n,k}^{(s)}$(Meyer et al., 2018). The term $v\big(\mathbf{x}_{n,k},1,a_{n,k}^{(s)};\mathbf{z}_n^{(s)}\big)$ is given as

$$\begin{aligned}v\big(\mathcal{A}_{n,k},a_{n,k}^{(s)};\mathbf{z}_n^{(s)}\big)\triangleq\ &g\big(\mathbf{x}_{n,k},r_{n,k},a_{n,k}^{(s)};\mathbf{z}_n^{(s)}\big)\\&\times\varphi\big(\mathbf{x}_{n,k},r_{n,k},a_{n,k}^{(s)};M_n^{(s)}\big).\end{aligned} \tag{36}$$

$v\big(\mathbf{x}_{n,k},0,a_{n,k}^{(s)};\mathbf{z}_n^{(s)}\big)=1(a_{n,k}^{(s)})$ is given in Eqs(12) , (18), and (36).

Under the Gaussian condition, Eq.(35) can be rewritten as

$$\beta(a_{n,k}^{(s)},1)\propto\begin{cases}\mathcal{N}(\mathbf{z}_n^{(s)};\mathbf{h}_n^{(s)}(\mathbf{x}_{n|n-1,k}),\mathbf{S}_{n,k}^{(s)}), & a_{n,k}^{(s)}=m\\1, & a_{n,k}^{(s)}=0\end{cases} \tag{37}$$

where

$$\mathbf{S}_{n,k}^{(s)}=\mathbf{H}_{n,k}^{(s)}\mathbf{P}_{n|n-1,k}{\mathbf{H}_{n,k}^{(s)}}^{\mathrm{T}}+\mathbf{R}^{(s)} \tag{38}$$

$\mathbf{H}_{n,k}^{(s)}$ is the Jacobian matrix.

### 3.6.3 Data association

According to the factor graph of DKME-GaBP, the messages from target to measurement can be calculated as

$$\eta\big(a_{n,k}^{(s)}\big)=\prod_{m=1}^{M_n^{(s)}}\nu_{m\to k}^{(P)}\big(a_{n,k}^{(s)}\big), \tag{39}$$

where $\nu_{m\to k}^{(P)}(a_{n,k}^{(s)})$ is the result of the iteration loop in BP, executed for all $M_n^{(s)}$ bearing measurements in the internal loop $p\in\{1,\ldots,P\}$.

$$\begin{aligned}\nu_{m\to k}^{(p)}\big(a_{n,k}^{(s)}\big)=&\sum_{b_{n,m}^{(s)}=0}^{K}\Psi\big(a_{n,k}^{(s)},b_{n,m}^{(s)}\big)\\&\times\prod_{k'\in\mathcal{K}\setminus\{k\}}\zeta_{k'\to m}^{(p-1)}\big(b_{n,m}^{(s)}\big),\end{aligned} \tag{40}$$

where $\nu_{m\to k}^{(p)}(a_{n,k}^{(s)})$ is the message from measurement $m$ to target $k$, and $\zeta_{k'\to m}^{(p)}(b_{n,m}^{(s)})$ is the message from other targets $k'\neq k$ to the same measurement $m$.

$$\begin{aligned}\zeta_{k\to m}^{(p)}\left(b_{n,m}^{(s)}\right)=&\sum_{a_{n,k}^{(s)}=0}^{M_n^{(s)}}\beta\big(a_{n,k}^{(s)}\big)\Psi\big(a_{n,k}^{(s)},b_{n,m}^{(s)}\big)\\&\times\prod_{m'\in\mathcal{M}_n^{(s)}\setminus\{m\}}\nu_{m'\to k}^{(p)}\big(a_{n,k}^{(s)}\big).\end{aligned} \tag{41}$$

The above equation constitutes an iteration loop, which is initialized ($p=0$) as

$$\zeta_{k\to m}^{(0)}\left(b_{n,m}^{(s)}\right)=\sum_{a_{n,k}^{(s)}=0}^{M_n^{(s)}}\beta\big(a_{n,k}^{(s)}\big)\Psi\big(a_{n,k}^{(s)},b_{n,m}^{(s)}\big). \tag{42}$$

### 3.6.4 Measurement update

The measurement update messages are given as

$$\begin{aligned}\gamma^{(s)}(\mathbf{x}_{n,k},1)=&\sum_{a_{n,k}^{(s)}=0}^{M_n^{(s)}}v\big(\mathbf{x}_{n,k},1,a_{n,k}^{(s)};\mathbf{z}_n^{(s)}\big)\eta\big(a_{n,k}^{(s)}\big)\\\gamma_{n,k}^{(s)}=&\eta\big(a_{n,k}^{(s)}=0\big).\end{aligned} \tag{43}$$

According to Eqs(36) and (39) and the Gaussian condition, Eq.(43) can be rewritten as

$$\begin{aligned}&\gamma^{(s)}(\mathbf{x}_{n,k},1)\propto\\&\eta(a_{n,k}^{(s)}=0)\mathcal{N}\Big(\mathbf{x}_{n,k};\,\mathbf{h}_n^{(s),-1}(\mathbf{z}_{n,0}^{(s)}),\,\mathbf{H}_{n,k}^{(s),-1}\mathbf{S}_{n,k}^{(s)}(\mathbf{H}_{n,k}^{(s),-1})^{\mathrm{T}}\Big)\\&+\sum_{a_{n,k}^{(s)}=1}^{M_n^{(s)}}\eta(a_{n,k}^{(s)})\mathcal{N}\Big(\mathbf{x}_{n,k};\,\mathbf{h}_n^{(s),-1}(\mathbf{z}_{n,a_{nk}^{(s)}}^{(s)}),\,\mathbf{H}_{n,k}^{(s),-1}\mathbf{R}^{(s)}(\mathbf{H}_{n,k}^{(s),-1})^{\mathrm{T}}\Big)\end{aligned} \tag{44}$$

Unedited

in the case of $a_{n,k}^{(s)}=0$, $\mathbf{z}_{n,0}^{(s)}=\mathbf{h}_n^{(s)}(\mathbf{x}_{n|n-1,k})$ is replaced by the predictive measurement,

Then, Eq.(44) is rewritten as

$$\gamma^{(s)}(\mathbf{x}_{n,k},1)\propto\eta(a_{n,k}^{(s)}=0)\Phi_{n,k}\mathbf{w}_{n,k}^{-}+\sum_{a_{n,k}^{(s)}=1}^{M_n^{(s)}}\eta(a_{n,k}^{(s)})\Phi_{n,k}\mathbf{w}_{n,a_{n,k}^{(s)}}^{+} \tag{45}$$

The calculation pdf for $\mathbf{w}_{n,a_{n,k}^{(s)}}^{+}$ and $S_{n,a_{n,k}^{(s)}}^{+}$ are given as

$$\mathbf{w}_{n,a_{n,k}^{(s)}}^{+}=\mathbf{w}_{n,k}^{-}+Q_n\left(Z_{:,\mathbf{y}_{a_{n,k}^{(s)}}}-Z_{\mathbf{yy}}\mathbf{w}_{n,k}^{-}\right), \tag{46}$$

$$S_{n,a_{n,k}^{(s)}}^{+}=S_{n,k}^{-}-Q_nZ_{\mathbf{yy}}S_{n,k}^{-}, \tag{47}$$

$$Q_n=S_{n,k}^{-}\left(Z_{\mathbf{yy}}S_{n,k}^{-}+\kappa I\right)^{-1}, \tag{48}$$

where $Z_{\mathbf{yy}}=\Upsilon^{\mathrm{T}}\Upsilon$ is the Gram matrix of the observed values from the kernel particle states, $\Upsilon=\left[\phi_y(\mathbf{y}_{n,k}^{\{1\}}),\ldots,\phi_y(\mathbf{y}_{n,k}^{\{M\}})\right]$, observed particles $\mathbf{y}_{n,k}=h^{(s)}(\mathbf{x}_{n,k},w_n^{(s)})$ are drawn from kernel particles. $Z_{:,\mathbf{y}_n}=\left[k_{\mathbf{y}}(\mathbf{y}_{n,k}^{\{1\}},\mathbf{z}_{n,a_{n,k}^{(s)}}),\ldots,k_{\mathbf{y}}(\mathbf{y}_{n,k}^{\{M\}},\mathbf{z}_{n,a_{n,k}^{(s)}})\right]^{\mathrm{T}}$ is the kernel vector of the measurement, where $\mathbf{z}_{n,a_{n,k}^{(s)}}$ represents the actual bearing. Then, Eq.(44) can be given as

$$\gamma^{(s)}(\mathbf{x}_{n,k})\propto\mathcal{N}(\mathbf{x}_{n,k}^{\gamma^{(s)}},\mathbf{P}_{n,k}^{\gamma^{(s)}{}^{-1}}) \tag{49}$$

where,

$$\mathbf{x}_{n,k}^{\gamma^{(s)}}=\sum_{a_{n,k}^{(s)}=0}^{M_n^{(s)}}\eta(a_{n,k}^{(s)})\mathbf{x}_{n,k}^{\{i=1:M\}}\mathbf{w}_{n,a_{n,k}^{(s)}}^{+} \tag{50}$$

$$\mathbf{P}_{n,k}^{\gamma^{(s)}}=\mathbf{H}_{n,k}^{(s)\mathrm{T}}\Big(\eta(a_{n,k}^{(s)}=0)\mathbf{S}_{n,k}^{(s)}+\sum_{a_{n,k}^{(s)}=1}^{M_n^{(s)}}\eta(a_{n,k}^{(s)})\hat{\Sigma}_{n,a_{n,k}^{(s)}}\Big)^{-1}\mathbf{H}_{n,k}^{(s)}. \tag{51}$$

The KME covariance $\hat{\Sigma}_{n,a_{n,k}^{(s)}}$ is given as

$$\hat{\Sigma}_{n,a_{n,k}^{(s)}}=\mathbf{x}_{n,k}^{\{i=1:M\}}S_{n,a_{n,k}^{(s)}}^{+}\mathbf{x}_{n,k}^{\{i=1:M\}}. \tag{52}$$

### 3.6.5 Target belief calculation

After the sensors perform the inner loop of the DKME-GaBP, the new confidence $\tilde{f}(\mathbf{x}_{n,k})$ for the participating targets is computed using the sum-product algorithm in the outer loop, resulting in an approximation of the marginal posterior pdf$f(\mathbf{x}_{n,k}|\mathbf{z})$, which is normalized as

$$\tilde{f}(\mathbf{x}_{n,k},1)\propto\alpha(\mathbf{x}_{n,k},1)\prod_{s=1}^{S}\gamma^{(s)}(\mathbf{x}_{n,k},1) \tag{53}$$

$$\tilde{f}_{n,k}\propto\alpha_{n,k}\prod_{s=1}^{S}\gamma_{n,k}^{(s)} \tag{54}$$

Under the Gaussian condition, the distribution of the target at time $n$ is given as

$$\widetilde{f}(\mathbf{x}_{n,k},1)\propto\mathcal{N}(\mathbf{x}_{n|n,k},\mathbf{P}_{n|n,k}) \tag{55}$$

where

$$\mathbf{P}_{n|n,k}=\left(\mathbf{P}_{n|n-1,k}^{-1}+\sum_{s\in\mathbb{S}}\mathbf{P}_{n,k}^{\gamma^{(s)}}\right)^{-1}, \tag{56}$$

$$\mathbf{x}_{n|n,k}=\mathbf{P}_{n|n,k}\Big(\mathbf{P}_{n|n-1,k}^{-1}\mathbf{x}_{n|n-1,k}+\sum_{s\in\mathbb{S}}\mathbf{P}_{n,k}^{\gamma^{(s)}}\mathbf{x}_{n,k}^{\gamma^{(s)}}\Big). \tag{57}$$

To enhance the effectiveness of multi-sensor data fusion, a KME-based fusion method is designed. In RKHS, the fusion of pdfs estimated by different sensors considers both the distance between the two pdfs and bearing information. Thus, under the KME-based fusion framework using a quadratic kernel, Eq.(56) can be expressed as

$$\mathbf{x}_{n|n,k}=\sum_{s=1}^{S}w_{n,k}^{(s)}\mathbf{x}_{n,k}^{\gamma^{(s)}}, \tag{58}$$

$$\mathbf{P}_{n|n,k}=\sum_{s=1}^{S}w_{n,k}^{(s)}\Big(\mathbf{x}_{n,k}^{\gamma^{(s)}}\left(\mathbf{x}_{n,k}^{\gamma^{(s)}}\right)^{\mathrm{T}}+\left(\mathbf{P}_{n,k}^{\gamma^{(s)}}\right)^{2}\Big)-\Big(\sum_{s=1}^{S}w_{n,k}^{(s)}\mathbf{x}_{n|n-1,k}\Big)\Big(\sum_{s=1}^{S}w_{n,k}^{(s)}\mathbf{x}_{n|n-1,k}\Big)^{\mathrm{T}}. \tag{59}$$

where $w_{n,k}^{(s)}$ is the weight calculated based on the estimated bearing information from each sensor and the predicted bearing information from the fusion center. The fusion method is implemented by minimizing the sum of squared distances between the fused probability density and each local probability density. Specifically, let$\frac{\partial f(\mathbf{x}_{n,k}^{\gamma^{(s)}},\mathbf{P}_{n,k}^{\gamma^{(s)}})}{\partial(\mathbf{x}_{n,k}^{\gamma^{(s)}},\mathbf{P}_{n,k}^{\gamma^{(s)}})}=0$, where$f(\mathbf{x}_{n,k}^{\gamma^{(s)}},\mathbf{P}_{n,k}^{\gamma^{(s)}})=\langle\mathcal{N}(\mathbf{x}_{n,k}^{\gamma^{(s)}},\mathbf{P}_{n,k}^{\gamma^{(s)}}),\mathcal{N}(\mathbf{x}_{n,k}^{\gamma^{(s)}},\mathbf{P}_{n,k}^{\gamma^{(s)}})\rangle_{\mathcal{F}}$(Luo et al., 2024).

After calculating the target confidence, KME proposal particles are drawn from Eq.(60). The

Unedited

mean and variance are derived from the calculated Gaussian distribution and are expressed as

$$\tilde{\mathbf{x}}_{n|n,k}^{\{i=1:M\}} \sim \mathcal{N}(\mathbf{x}_{n|n,k}, \mathbf{P}_{n|n,k}), \tag{60}$$

$$\Psi_{n,k} = [\phi_{\mathbf{x}}(\tilde{\mathbf{x}}_{n,k}^{\{1\}}), \ldots, \phi_{\mathbf{x}}(\tilde{\mathbf{x}}_{n,k}^{\{M\}})]. \tag{61}$$

KME is better suited for GaBP compared to traditional BP algorithms, as it is not constrained to approximate the state pdf of the target as Gaussian, which makes state updates and error propagation easier to handle.

## 4 Simulation

The proposed algorithm is tested to track multiple underwater targets moving at a nearly constant velocity (CV) with four sensors moving at constant turn (CT) in the presence of clutter. To validate its tracking performance, DKME-GaBP is compared with GaBP (Yan et al., 2024), PFBP (Meyer et al., 2017) and LMB-EKF (Vo et al., 2014) over 100 Monte Carlo runs, each consisting of 400 scans. The simulations are conducted on a PC equipped with an Intel Core i5-12400F CPU and 32 GB of RAM, with all filters implemented in MATLAB R2024a. Performance is evaluated using the optimal sub-pattern assignment (OSPA), the average OSPA(AOSPA) metrics, and runtime per frame(RPF). The definitions of OSPA and AOSPA can be found in (Ristic et al., 2011). In this study, the OSPA parameters are set as $p = 2$ to control the distance and $c = 100$ to truncate the maximum contribution of an individual target to the overall error.

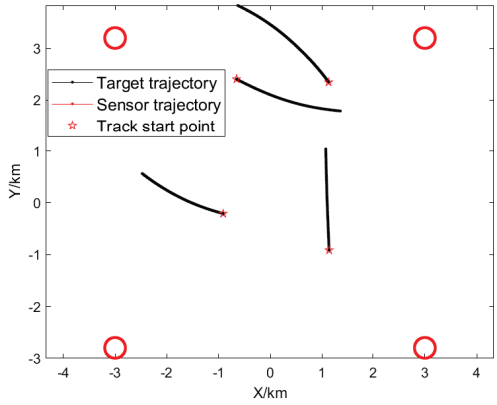


**Fig. 3 Tracking scenario with four targets and four sensors**

The scenario is shown in Fig. 3, with the following scenario parameters: the sampling interval $T = 1$ s, the average number of false alarms is $\mu^{(s)} = 5$, the targets' initial states are shown in Table 1, and the sensors initial states are shown in Table 2. The number of particles in PFBP and DKME-GaBP is 50. The detection probability $P_d$ of all sensors is 0.9, the gating probability $P_g = 0.98$, and the angle measurement standard deviation is $\sigma_\theta = 1°$. The Gaussian kernel is used in DKME-GaBP with hyperparameter $\sigma = 0.1$.

**Table 1 Motion parameters for targets**

| Target | Survival time | Initial state |
|---|---|---|
| 1 | 1—393 s | $[-653\text{m}, 5\text{m/s}, 2407\text{m}, 2\text{m/s}]$ |
| 2 | 1—363 s | $[-911\text{m}, -5\text{m/s}, -208\text{m}, -1\text{m/s}]$ |
| 3 | 1—397 s | $[1128\text{m}, -3\text{m/s}, 2347\text{m}, 5\text{m/s}]$ |
| 4 | 1—331 s | $[1137\text{m}, -0.2\text{m/s}, -914\text{m}, 5\text{m/s}]$ |

**Table 2 Motion parameters for sensors**

| Sensor | Initial state | $\omega$ |
|---|---|---|
| 1 | $[-3000\text{m}\ , 10\text{m/s}\ , 3000\text{m}\ , 5\text{m/s}\ ]$ | 0.9 °/s |
| 2 | $[3000\text{m}\ , 10\text{m/s}, 3000\text{m}\ , 5\text{m/s}\ ]$ | 0.9 °/s |
| 3 | $[3000\text{m}\ , 10\text{m/s}\ , -3000\text{m}\ , 5\text{m/s}\ ]$ | 0.9 °/s |
| 4 | $[-3000\text{m}\ , 10\text{m/s}\ , -3000\text{m}\ , 5\text{m/s}\ ]$ | 0.9 °/s |

### 4.1 Tracking performance comparison

#### 4.1.1 Tracking accuracy

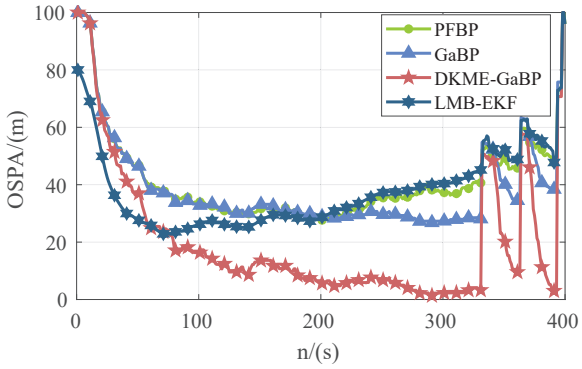


**Fig. 4 OSPAs of different methods.**

Fig. 4 shows the OSPAs of three algorithms. It can be seen that DKME-GaBP exhibits the smallest estimation error and responds more quickly, outperforming PFBP and GaBP. Due to its relatively accurate initialization region, LMB-EKF responds quickly in the initial stage. There are three peaks

Unedited

on the OSPA curves during the surveillance, indicating the extinction of targets. DKME-GaBP demonstrated the best performance, which is attributed to the KME algorithm that alleviates the strong nonlinear issues associated with passive tracking. LMB-EKF gradually diverges in the later stages due to the EKF's weak ability to track highly nonlinear motion, resulting in a decrease in OSPA.

Tables 3 and 4 show the AOSPAs of DKME-GaBP, GaBP, PFBP, and LMB-EKF under varying $P_d$ and $\mu^{(s)}$. It can be seen that as $P_d$ and $\mu^{(s)}$ increase, the AOSPAs of all algorithms decrease, with DKME-GaBP consistently outperforming the others. This indicates that DKME-GaBP exhibits significant robustness in handling high detection probability and clutter scenarios, maintaining superior tracking accuracy under challenging conditions.

**Table 3 AOSPAs of different methods**

| Metrics | $P_d$ | DKME-GaBP | GaBP | PFBP | LME-EKF |
|---|---|---|---|---|---|
| AOSPA | 0.9 | 20.8 | 38.9 | 41.7 | 38.0 |
| | 0.8 | 27.0 | 41.3 | 42.8 | 40.7 |
| | 0.7 | 27.3 | 42.0 | 44.2 | 42.3 |
| | 0.6 | 27.8 | 42.8 | 44.6 | 42.8 |

**Table 4 AOSPAs of different methods**

| Metrics | $\mu^{(s)}$ | DKME-GaBP | GaBP | PFBP | LME-EKF |
|---|---|---|---|---|---|
| AOSPA | 4 | 20.2 | 38.6 | 41.5 | 37.5 |
| | 6 | 26.4 | 41.2 | 43.8 | 41.4 |
| | 8 | 27.7 | 42.3 | 44.1 | 43.3 |
| | 10 | 28.8 | 43.7 | 45.7 | 43.8 |

### 4.1.2 Computational complexity

This section discusses the computational complexity of the DKME-GaBP algorithm. The baseline values for the number of sensors $S$, targets $K$, and measurements $M$ are first set as $S = 4$, $K = 4$, and $M = 5$, respectively. Subsequently, the computational complexity of the algorithm is tested by varying each parameter while keeping the others constant.

First, as illustrated in Fig. 5, the effect on the runtime of the number of measurements $M$ from each sensor is investigated. Then the effect of the number of sensors $S$ is shown in Fig. 6. Figs5 and 6 show that the runtimes of all algorithms exhibit a nearly linear expansion, while PFBP has a longer runtime than the DKME-GaBP and GaBP. The RPF of DKME-GaBP is shorter than PFBP but longer than GaBP. The shorter runtime compared to PFBP is due to the lower computational complexity, as the resampling step is avoided by KME. The longer runtime compared to GaBP is due to the addition of the KME method, implemented with particles, to improve estimation accuracy. This increase in computational time is considered a reasonable trade-off, as the improved tracking performance justifies the extra cost.

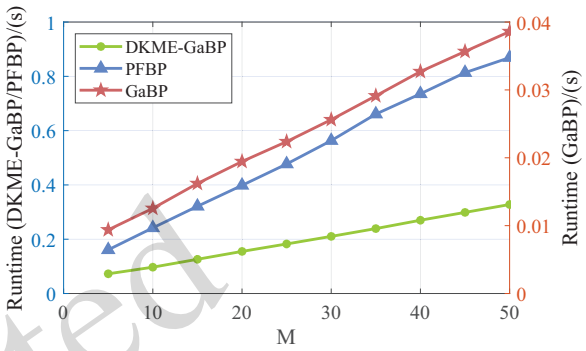


**Fig. 5 RPF of different methods against M**

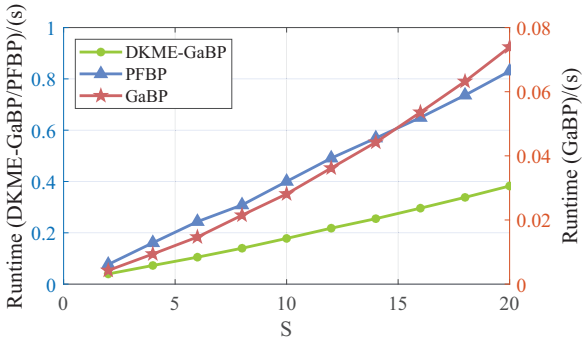


**Fig. 6 RPF of different methods against S**

Finally, the effect of the number of targets $K$ on computational complexity is shown in Fig. 7. PFBP and GaBP exhibit a nearly quadratic growth in runtime, while DKME-GaBP has a shorter runtime compared to PFBP. The theoretical complexity of DKME-GaBP is analyzed as follows. For each sensor, the computational complexity of the local data association is $O(K \cdot M)$. In the outer fusion loop, each sensor contributes to the overall complexity linearly, resulting in a total complexity of $O(S \cdot K \cdot M)$. This efficiency makes DKME-GaBP particularly well-suited for real-time applications in underwater environments, where multiple PTs need to be tracked simultaneously.

Unedited

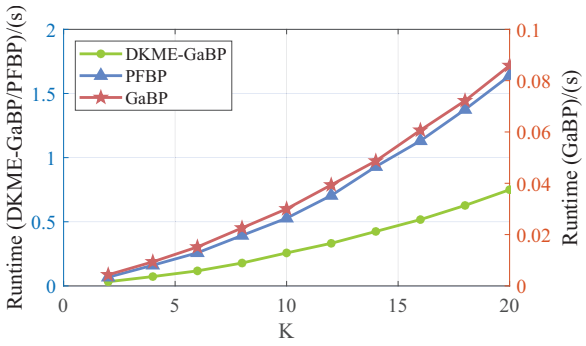


**Fig. 7 RPF of different methods against K**

## 4.2 Effect of kernels

The effect of different kernels is evaluated in this subsection. First, Gaussian, linear, Laplacian, quadratic, quartic, and sixth-order kernels are used to calculate the OSPAs of DKME-GaBP. Fig. 8 illustrates the effect of the kernels on the algorithm.

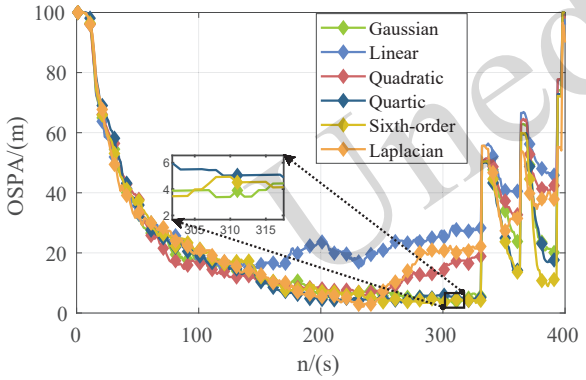


**Fig. 8 OSPAs of DKME-GaBP with different kernels**

**Table 5 RPF and AOSPAs of DKME-GaBP with different kernels**

| Kernel | Time | AOSPA |
|---|---|---|
| Quadratic | 0.0600 | 25.8332 |
| Quartic | 0.0754 | 22.3609 |
| Sixth-order | 0.0850 | 22.2819 |
| Gaussian | 0.0739 | 22.9440 |
| Laplacian | 0.0725 | 25.8469 |
| Linear | 0.0547 | 32.4319 |

As shown in Fig. 8, the quartic, sixth-order, and Gaussian kernels exhibit the smallest estimation errors, whereas the linear kernel has the largest one. The Laplacian and Quadratic kernels exhibit the comparable performance. However, the computational complexity of the polynomial kernel is $O(n^d)$, and that of the Gaussian kernel is $O(n^2)$, where $d$ is the degree of the polynomial kernel. The results of the runtime are shown in Table 5. For Gaussian kernels, the effect of hyperparameters on tracking performance is considered crucial. Different hyperparameters $\sigma$ are compared and analyzed.

**Table 6 AOSPAs of DKME-GaBP with a Gaussian kernel against different $\sigma$**

| Metrics | $\sigma$ | DKME-GaBP |
|---|---|---|
| AOSPA | 0.005 | 25.3143 |
| | 0.010 | 23.8745 |
| | 0.100 | 22.9440 |
| | 0.200 | 23.1900 |
| | 0.500 | 23.2075 |
| | 1.000 | 23.4914 |

From Table 6, it can be seen that when $\sigma = 0.1$, the AOSPA is smallest. This is because the kernel width is moderate, allowing it to effectively capture the local similarities between target state features while avoiding the effects of over-smoothing or excessive sensitivity. In practice, $\sigma$ can be chosen based on the target's motion model. For CV targets, it is typically set a relatively small $\sigma$ is preferred to reflect the smooth motion. For maneuvering targets, a slightly larger $\sigma$ is often used to better capture wider state variations.

## 4.3 Heavy-tailed noise

The DKME-GaBP's ability to handle heavy-tailed noise is demonstrated.

### 4.3.1 Heavy-tailed noise modeling

Construct the heavy-tailed process noise $\mathbf{u}_n$ and measurement noise $\boldsymbol{w}_n$. According to (Wang et al., 2025), the process noise covariance $Q_n$ and the measurement noise covariance $R_n$ are slowly varying

$$\boldsymbol{Q}_n = [15 + 1.5\cos(\pi n/N)] \times q \times \begin{bmatrix} \frac{T^3}{3}\boldsymbol{I}_2 & \frac{T^2}{2}\boldsymbol{I}_2 \\ \frac{T^2}{2}\boldsymbol{I}_2 & T\boldsymbol{I}_2 \end{bmatrix}, \quad (62)$$

$$\boldsymbol{R}_n = [0.1 + 0.05\cos(\pi n/N)] \times g \times 0.5, \quad (63)$$

where $\boldsymbol{I}_2$ is a 2-dimensional identity matrix with $q = 0.5$ and $g = 100$.

Set the heavy-tailed distribution for the system

Unedited

process noise and measurement noise as

$$\mathbf{u}_n \sim \begin{cases} N(0, \boldsymbol{Q}_n), & w.p \quad 0.95 \\ N(0, a\boldsymbol{Q}_n), & w.p \quad 0.05 \end{cases} \tag{64}$$

$$\boldsymbol{w}_n \sim \begin{cases} N(0, \boldsymbol{R}_n), & w.p \quad 0.95 \\ N(0, b\boldsymbol{R}_k), & w.p \quad 0.05 \end{cases} \tag{65}$$

where $a$ and $b$ are the adjustment parameters for the size of the outliers, and $w.p$ (with probability) represents the outlier probability parameter.

#### 4.3.2 Tracking performance against a and b

$a$ is set to 100, 300 and 500. Similarly, $b$ is also set to 100, 300 and 500. The simulation results are shown in Fig. 9.

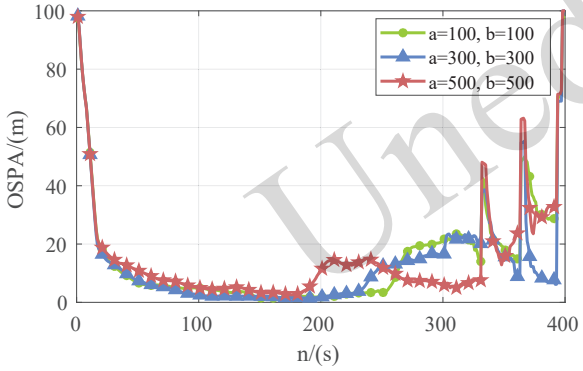


**Fig. 9 OSPAs of DKME-GaBP against different a and b**

As shown in Fig. 9, the tracking performance of DKME-GaBP under heavy-tailed noise is illustrated. Although a slight degradation is observed in the latter part of the OSPA curve due to increased target maneuverability caused by heavy-tailed noise, DKME-GaBP still maintains excellent and stable performance. This robustness is attributed to the ability of KME to approximate complex noise distributions and handle nonlinearities in the RKHS (Song et al., 2013). In addition, data characteristics are better captured and state estimates are improved through the use of a Gaussian kernel. Estimation accuracy and robustness are further enhanced by the distributed fusion framework, in which aggregates information from multiple sensors to mitigate the impact of local outliers and heavy-tailed interference.

## 5 Conclusions

The DKME-GaBP algorithm presented in this paper is an effective solution for complex underwater multi-sensor multi-target passive tracking. By addressing nonlinearity and flexibly handling unknown and time-varying target numbers in the presence of clutter, the algorithm significantly enhances tracking accuracy. Performance metric OSPA demonstrates that DKME-GaBP outperforms traditional methods such as GaBP, PFBP, and LMB-EKF, providing more accurate target estimations. Additionally, its lower computational complexity, which increases with the number of targets and sensors, highlights its practicality for real-time applications. Overall, DKME-GaBP proves to be a robust and efficient approach for underwater multi-target tracking.

Unedited

### Contributors

Yunfei GUO designed the research, Dengpeng YANG drafted this paper. Yanbo XUE, Anke XUE, and Yun CHEN helped organize the paper. Yunfei GUO and Dengpeng YANG revised and finalized the paper.

### Conflict of interest

All the authors declare that they have no conflict of interest.

### Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

### References

Aidala V, Hammel S, 1983. Utilization of modified polar coordinates for bearings-only tracking. *IEEE Trans Autom Control*, 28(3):283-294. https://doi.org/10.1109/TAC.1983.1103230

Badriasl L, Arulampalam S, Nguyen NH, et al., 2020. An algebraic closed-form solution for bearings-only maneuvering target motion analysis from a nonmaneuvering platform. *IEEE Trans Signal Process*, 68:4672-4687. https://doi.org/10.1109/TSP.2020.3012004

Cao JK, Zhang XF, Hao HH, et al., 2024. Noncircular signal tracking with distributed passive arrays: Combining data fusion and extended Kalman filter. *IEEE Sens J*, 24(1):757-768. https://doi.org/10.1109/JSEN.2023.3333863

Chen HY, Liu MQ, Zhang SL, 2018. Energy-efficient localization and target tracking via underwater mobile sensor networks. *Front Inf Technol Electron Eng*, 19(8):999-1012. https://doi.org/10.1631/FITEE.1700598

Choi S, Crouse DF, Willett P, et al., 2014. Approaches to cartesian data association passive radar tracking in a

dab/dvb network. *IEEE Trans Aerosp Electron Syst*, 50(1):649-663. https://doi.org/10.1109/TAES.2013.120431

Gebhardt GHW, Kupcsik A, Neumann G, 2019. The kernel Kalman rule. *Mach Learn*, 108(12):2113-2157. https://doi.org/10.1007/s10994-019-05816-z

Gunes A, Guldogan MB, 2016. Joint underwater target detection and tracking with the Bernoulli filter using an acoustic vector sensor. *Digit Signal Process*, 48:246-258. https://doi.org/10.1016/j.dsp.2015.09.020

Jiang HN, Cai YL, Yu ZH, 2022. Observability metrics for single-target tracking with bearings-only measurements. *IEEE Trans Syst Man Cybern Syst*, 52(2):1065-1077. https://doi.org/10.1109/TSMC.2020.3012485

Kim J, 2024. Tracking multiple underwater targets using adaptive Gaussian mixture probability hypothesis density filter with unknown clutter rate. *IEEE Trans Aerosp Electron Syst*, 60(6):9154-9162. https://doi.org/10.1109/TAES.2024.3437343

Li LQ, Xie WX, Liu ZX, 2016a. Auxiliary truncated particle filtering with least-square method for bearings-only maneuvering target tracking. *IEEE Trans Aerosp Electron Syst*, 52(5):2562-2567. https://doi.org/10.1109/TAES.2016.150048

Li XH, Willett P, Baum M, et al., 2016b. PMHT approach for underwater bearing-only multisensor-multitarget tracking in clutter. *IEEE J Ocean Eng*, 41(4):831-839. https://doi.org/10.1109/JOE.2015.2506220

Luo MJ, Zhou J, Zou QK, 2024. Multisensor estimation fusion based on kernel mean embedding. Proc 27th Int Conf on Information Fusion, p.1-7. https://doi.org/10.23919/FUSION59988.2024.10706487

Meyer F, Braca P, Willett P, et al., 2017. A scalable algorithm for tracking an unknown number of targets using multiple sensors. *IEEE Trans Signal Process*, 65(13):3478-3493. https://doi.org/10.1109/TSP.2017.2688966

Meyer F, Kropfreiter T, Williams JL, et al., 2018. Message passing algorithms for scalable multitarget tracking. *Proc IEEE*, 106(2):221-259. https://doi.org/10.1109/JPROC.2018.2789427

Northardt T, Nardone SC, 2019. Track-before-detect bearings-only localization performance in complex passive sonar scenarios: A case study. *IEEE J Ocean Eng*, 44(2):482-491. https://doi.org/10.1109/JOE.2028.2811419

Rao SK, 2005. Modified gain extended Kalman filter with application to bearings-only passive manoeuvring target tracking. *IEE Proc Radar Sonar Navig*, 152(4):239-244. https://doi.org/10.1049/ip-rsn:20045074

Ristic B, Vo BN, Clark D, et al., 2011. A metric for performance evaluation of multi-target tracking algorithms. *IEEE Trans Signal Process*, 59(7):3452-3457. https://doi.org/10.1109/TSP.2011.2140111

Shen CH, Kim J, Wang HZ, 2010. Generalized kernel-based visual tracking. *IEEE Trans Circuits Syst Video Technol*, 20(1):119-130. https://doi.org/10.1109/TCSVT.2009.2031393

Song L, Fukumizu K, Gretton A, 2013. Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models. *IEEE Signal Process Mag*, 30(4):98-111. https://doi.org/10.1109/MSP.2013.2252713

Sun MW, Davies ME, Proudler IK, et al., 2022. Adaptive kernel Kalman filter based belief propagation algorithm for maneuvering multi-target tracking. *IEEE Signal Process Lett*, 29:1452-1456. https://doi.org/10.1109/LSP.2022.3184534

Sun MW, Davies ME, Proudler IK, et al., 2023. Adaptive kernel Kalman filter. *IEEE Trans Signal Process*, 71:713-726. https://doi.org/10.1109/TSP.2023.3250829

Tian YW, Liu MQ, Zhang SL, et al., 2023. Feature-aided passive tracking of noncooperative multiple targets based on the underwater sensor networks. *IEEE Internet Things J*, 10(5):4579-4591. https://doi.org/10.1109/JIOT.2022.3218733

Tian YW, Liu MQ, Zhang SL, et al., 2024. Underwater target tracking based on the feature-aided GM-PHD method. *IEEE Trans Instrum Meas*, 73:5500412. https://doi.org/10.1109/TIM.2023.3336455

Vo BN, Vo BT, Phung D, 2014. Labeled random finite sets and the bayes multi-target tracking filter. *IEEE Trans Signal Process*, 62(24):6554-6567. https://doi.org/10.1109/TSP.2014.2364014

Wang L, Chen H, Lian F, et al., 2025. Robust Bayesian recursive ensemble Kalman filter under the nonstationary heavy-tailed noise. *IEEE Sens J*, 25(1):749-762. https://doi.org/10.1109/JSEN.2024.3477421

Williams J, Lau R, 2014. Approximate evaluation of marginal association probabilities with belief propagation. *IEEE Trans Aerosp Electron Syst*, 50(4):2942-2959. https://doi.org/10.1109/TAES.2014.120568

Wolek A, McMahon J, Dzikowicz BR, et al., 2022. Tracking multiple surface vessels with an autonomous underwater vehicle: Field results. *IEEE J Ocean Eng*, 47(1):32-45. https://doi.org/10.1109/JOE.2020.3015415

Xiao CS, Hill DJ, 1996. Stability and absence of overflow oscillations for 2-D discrete-time systems. *IEEE Trans Signal Process*, 44(8):2108-2110. https://doi.org/10.1109/78.533735

Xu ZQ, Guo YF, Kuang Y, et al., 2024. Multi-sensor distributed fusion based on cross-location for passive tracking. *Signal Image Video Process*, 18(12):9441-9449. https://doi.org/10.1007/s11760-024-03558-7

Xue YB, Guo YF, Yang DS, et al., 2025. Distributed multi-sensor multi-target tracking with fault detection and exclusion using belief propagation. *Digit Signal Process*, 156:104797. https://doi.org/10.1016/j.dsp.2024.104797

Yan L, Guo YF, Lin BT, et al., 2024. Scalable multitarget tracking using PCL in SFN with 3D data association uncertainty. *Digit Signal Process*, 146:104355. https://doi.org/10.1016/j.dsp.2023.104355

Zhang WY, Meyer F, 2024. Multisensor multiobject tracking with improved sampling efficiency. *IEEE Trans Signal Process*, 72:2036-2053. https://doi.org/10.1109/TSP.2024.3374047

Unedited