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

Hybrid-driven Gaussian process online learning for highly maneuvering multi-target tracking*

Qiang GUO†1, Long TENG†‡1,2, Tianxiang YIN†3, Yunfei GUO†3, Xinliang WU2, Wenming SONG2

[1]*College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China*

[2]*China National Aeronautical Radio Electronics Research Institute, Shanghai 200233, China*

[3]*College of Automation, Hangzhou Dianzi University, Hangzhou 310018, China*

†E-mail: guoqiang@hrbeu.edu.cn; tenglong@hrbeu.edu.cn; tianxiangyin@hdu.edu.cn; gyf@hdu.edu.cn

Received May 18, 2023; Revision accepted Oct. 4, 2023; Crosschecked Oct. 31, 2023

Abstract: The performance of existing maneuvering target tracking methods for highly maneuvering targets in cluttered environments is unsatisfactory. This paper proposes a hybrid-driven approach for tracking multiple highly maneuvering targets, leveraging the advantages of both data-driven and model-based algorithms. The time-varying constant velocity model is integrated into the Gaussian process (GP) of online learning to improve the performance of GP prediction. This integration is further combined with a generalized probabilistic data association algorithm to realize multi-target tracking. Through the simulations, it has been demonstrated that the hybrid-driven approach exhibits significant performance improvements in comparison with widely used algorithms such as the interactive multi-model method and the data-driven GP motion tracker.

Key words: Target tracking; Gaussian process; Data-driven; Online learning; Model-driven; Probabilistic data association

https://doi.org/10.1631/FITEE.2300348 **CLC number:** TN953

1 Introduction

Target tracking refers to the estimation of the motion state of the interest target, considering factors such as noisy measurements, false alarm clutter, and imperfect measurements (Da et al., 2021; Zhu et al., 2021). It is widely used in various fields, such as aircraft target tracking (Zhou et al., 2020; Zheng and Cai, 2021) and underwater target tracking (Zhang D et al., 2018). Depending on the nature of the tracked targets, target tracking algorithms can be categorized into point target tracking, extended target tracking, and group target tracking (Guo et al., 2019, 2020a; Wang et al., 2021). The primary focus of this paper is to address the challenge of maneuvering posed by point target tracking. It uses kinematic information to estimate the target's state, which includes parameters such as position, velocity, and acceleration (Li et al., 2019b; Li and Hlawatsch, 2021).

The central concept of maneuvering target tracking focuses on enhancing the degree of correspondence between the tracking model and the real motion model of the target (Li et al., 2019a). Maneuvering target tracking algorithms are classified into two categories, which are the model-driven and data-driven approaches. Model-driven algorithms rely on the prior motion models of the target. The state-space model is modeled as a Markov process by using

‡ Corresponding author

* Project supported by the Technology Foundation for Basic Enhancement Plan, China (No. 2021-JCJQ-JJ-0301), the National Major Research and Development Project of China (No. 2018YFE0206500), the National Natural Science Foundation of China (No. 62071140), and the National Special for International Scientific and Technological Cooperation of China (No. 2015DFR10220)

ORCID: Long TENG, https://orcid.org/0000-0003-3519-7790

prior model information, thereby describing the relationship between the target state vectors at adjacent time slots (Guo et al., 2015; Li et al., 2017). Typical model-driven methods include the interactive multimodel (IMM) method (Wu et al., 2021), where the tracking results of different models are appropriately weighted, and the variable structure IMM (VSIMM), the core of which is the model set selection algorithm. In terms of computational complexity and performance, IMM has been demonstrated as the most cost-efficient approach.

Nevertheless, as the maneuverability of the target continues to advance and maneuvering patterns become more complex, the target trajectory may consist of many models, and the associated maneuvering parameters such as acceleration and angular velocity tend to be larger. Model-driven methods face several application limitations: (1) Poor tracking ability and environmental adaptability. If the target's maneuvering patterns lie beyond the scope of the prior model set, it can result in model mismatch, leading to decrease in the tracking performance of the model-driven algorithm or even causing the filter to diverge. (2) State estimation delay. Because the target motion model is unknown, as well as owing to model switching, model-driven methods need to gather sufficient measurements to precisely estimate the target state. (3) High computational complexity. In the case of highly maneuvering targets, the uncertainty in their maneuvers is significant, and their trajectories may encompass an infinite number of motion models. For model-driven approaches, it becomes impractical to construct an excessively large model set to account for all possible maneuvers.

In response to the constraints of the model-driven approaches, researchers have incorporated neural networks, which possess robust learning capabilities, into the maneuvering target tracking methodology. Zhang XR et al. (2019) and Deng et al. (2020) used long short-term memory (LSTM) neural networks to discern target maneuver parameters, such as the maneuver model, acceleration, and angular velocity. Liu JX et al. (2020) employed neural networks to directly predict the position estimation error of the target, enhancing the overall state estimate. Tian et al. (2022) proposed the multi-model Gaussian mixture probability hypothesis density (MM-GM-PHD) filter based on LSTM. Moreover, in Xiong et al. (2022), an end-to-end method was used to estimate the target state. The central concept is to learn the essential steps of the Bayesian filter, including the state transition matrix, process noise covariance matrix, and measurement noise covariance matrix. Nonetheless, in implementing the aforementioned neural network based maneuvering target tracking methods, we are still faced with certain obstacles, and these are (1) poor interpretability and (2) the fact that for small sample learning, significant challenges continue to be encountered.

In addition to neural networks, data-driven maneuvering target tracking methods encompass Gaussian process (GP), which consists of a series of random variables that follow a Gaussian distribution and fits data to an unknown function using different distributions obtained (Rasmussen and Williams, 2006). Huber (2014) proposed a method based on recursive GPs, enabling the online learning of GP hyperparameters with low computational burden. Aftab and Mihaylova (2021) presented a Gaussian process motion tracker (GPMT) algorithm. This algorithm has the capability to learn an arbitrary motion model online during each tracking period. Subsequently, the target state is estimated using the learned model and measurements. GPMT can flexibly switch between numerous or even unlimited motion models. A maneuvering target tracking method based on GPPF was proposed in Sun et al. (2020), wherein the researchers integrated the particle filter (PF) into GP target tracking. GP was extended to distributed multiple sensors in Liu XC et al. (2022). Guo et al. (2022) proposed a novel approach by combining IMM and GP for track segment association of maneuvering targets, and this approach demonstrated a superior performance in comparison with both the model-driven and data-driven methods. The essence of GP lies in learning unknown models through the covariance kernel function. Therefore, Aftab and Mihaylova (2020) conducted an analysis on the influence of different kernels and training data on the performance of target tracking. Furthermore, GP has found applications in maneuvering ship tracking (Guo et al., 2020b).

However, in the context of a more intricate maneuvering target within a cluttered background, where the target encompasses an infinite spectrum of motion models and its maneuvering parameters are more extensive and exhibit rapid fluctuations, GPMT might, due to the influence of clutter,

encounter substantial state prediction errors, potentially leading to target loss. In this paper we present a novel online learning approach for enhancing the tracking precision of intricate maneuvering targets amidst cluttered environments. The method uses a hybrid-driven GP as the fundamental framework. Building upon this foundation and in conjunction with the generalized probabilistic data association (GPDA) algorithm, robust tracking of highly maneuvering multiple targets is achieved.

2 Background

The standard GPs are presented in this section. The core of a GP that can learn any unknown function is comprised of the mean and covariance (kernel) functions. In most applications, the mean remains constant or zero (Huber, 2014). Thus, the design of the covariance emerges as a crucial aspect within GP. Readers can refer to Huber (2014) for further information regarding various mean and covariance functions. The function $f(\cdot)$ of interest, which is unknown and subject to additive noise, is represented as follows:

$$\boldsymbol{x} = f(\boldsymbol{t}) + \boldsymbol{\tau}, \tag{1}$$

where $\boldsymbol{x}$ represents the measurement vector, $\boldsymbol{t}$ is the input vector, and $\boldsymbol{\tau}$ refers to the Gaussian white noise characterized by a mean of zero and a variance of σ_n^2. Considering the prior data (t_i, x_i), t_i represents the input and x_i represents the corresponding output, $i = 1, 2, \cdots, n$. $f(\cdot)$ can be modeled with GP as

$$f(\boldsymbol{t}) \sim \mathrm{GP}\left(m(\boldsymbol{t}), k\left(\boldsymbol{t}, \boldsymbol{t}'\right)\right), \tag{2}$$

where $m(\cdot)$ is the mean function and $k(\cdot,\cdot)$ is the kernel function.

Let us consider the scenario where we aim to predict the test output $f\left(\boldsymbol{t}^*\right)$. For notational simplicity, $f \triangleq f(\boldsymbol{t})$ and $f^* \triangleq f\left(\boldsymbol{t}^*\right)$. According to Huber (2014), the joint Gaussian distribution of the prior and f^* is

$$\begin{bmatrix} f \\ f^* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} k(\boldsymbol{t},\boldsymbol{t}) + \sigma_n^2 \boldsymbol{I} & k\left(\boldsymbol{t}, \boldsymbol{t}^*\right) \\ k\left(\boldsymbol{t}^*, \boldsymbol{t}\right) & k\left(\boldsymbol{t}^*, \boldsymbol{t}^*\right) \end{bmatrix}\right), \tag{3}$$

where $\boldsymbol{I}$ represents the identity matrix, and $k\left(\boldsymbol{t}, \boldsymbol{t}^*\right)$ is the kernel function between the input vector of test data and the input vector of training data, which is expressed as

$$k\left(\boldsymbol{t}, \boldsymbol{t}^*\right) = \begin{bmatrix} k\left(t_1, t_1^*\right) & k\left(t_1, t_2^*\right) & \cdots & k\left(t_1, t_l^*\right) \\ k\left(t_2, t_1^*\right) & k\left(t_2, t_2^*\right) & \cdots & k\left(t_2, t_l^*\right) \\ \vdots & \vdots & & \vdots \\ k\left(t_n, t_1^*\right) & k\left(t_n, t_2^*\right) & \cdots & k\left(t_n, t_l^*\right) \end{bmatrix}. \tag{4}$$

Here, the number of test data samples is denoted by l, and $k\left(t_n, t_l^*\right)$ is the kernel function between the n^{th} input of training data and the l^{th} input of test data. $\boldsymbol{t}$ and $\boldsymbol{t}^*$ represent the input vectors for f and f^*, respectively. Similarly, we can obtain $k(\boldsymbol{t},\boldsymbol{t})$, $k\left(\boldsymbol{t}^*,\boldsymbol{t}\right)$, and $k\left(\boldsymbol{t}^*,\boldsymbol{t}^*\right)$. According to Eq. (3), the prediction distribution (Huber, 2014) of the test is obtained as

$$f^*|_{\boldsymbol{t},f,\boldsymbol{t}^*} \sim \mathcal{N}\left(m\left(f^*\right), \mathrm{cov}\left(f^*\right)\right), \tag{5}$$

where

$$m\left(f^*\right) = k\left(\boldsymbol{t}^*, \boldsymbol{t}\right)\left[k(\boldsymbol{t},\boldsymbol{t}) + \sigma_n^2 \boldsymbol{I}\right]^{-1} \boldsymbol{x}, \tag{6}$$

$$\begin{aligned} \mathrm{cov}\left(f^*\right) =& k\left(\boldsymbol{t}^*, \boldsymbol{t}^*\right) - k\left(\boldsymbol{t}^*, \boldsymbol{t}\right)\left[k(\boldsymbol{t},\boldsymbol{t})\right. \\ &\left. + \sigma_n^2 \boldsymbol{I}\right]^{-1} k\left(\boldsymbol{t}, \boldsymbol{t}^*\right). \end{aligned} \tag{7}$$

Here, $m\left(f^*\right)$ and $\mathrm{cov}\left(f^*\right)$ represent the predicted mean and covariance, respectively.

3 Hybrid-driven highly maneuvering multi-target tracking

In this section we provide a comprehensive explanation of the proposed approach. The fundamental idea is to integrate the time-varying constant velocity (CV) model into GPMT, thereby enhancing predictive abilities to effectively track extremely maneuverable targets amidst cluttered surroundings. Expanding on this foundation, the methodology integrates GPDA to enable simultaneous tracking of multiple highly maneuvering targets. The method proposed here uses a limited set of previous measurements for target state estimation and operates under the assumption of orthogonal coordinate axes.

First, GP is employed to model the nonlinear function of the x coordinate, and the procedure is repeated for the y coordinate. For more specifics, readers can refer to Section 3.1. Next, the training set is formed by choosing the preceding d-dimensional measurement. The hyperparameters are learned through a maximum likelihood estimator based on the training set, as explained in Section 3.2. Once

again, the time-varying CV model is employed for state prediction, as indicated in Section 3.3. Subsequently, the GPDA algorithm is employed to accomplish the correlation matching between measurements and targets, as explained in Section 3.4 in greater detail. Ultimately, GP is applied for both state and training set updates; further elaboration can be found in Section 3.5. The state update process resembles the state update in the Kalman filter, wherein the target's state estimate is determined by using both the measurement and the state prediction. The training set update entails using GP to optimize the distribution of the prior training set, thus preventing underfitting.

3.1 GP modeling

According to Eqs. (1) and (2), the unknown nonlinear function f and the corresponding observation function (Huber, 2014) can be defined as

$$\boldsymbol{u}=f(\boldsymbol{t}),\quad f\sim \mathrm{GP}\left(0,k\left(\boldsymbol{t},\boldsymbol{t}'\right)\right), \tag{8}$$

$$\boldsymbol{x}=f(\boldsymbol{t})+\boldsymbol{\tau},\quad \boldsymbol{\tau}\sim\mathcal{N}\left(0,\sigma_n^2\boldsymbol{I}\right), \tag{9}$$

where $\boldsymbol{u}$ is the output position variable of the x coordinate, $\boldsymbol{t}$ is the corresponding input time variable, and $\boldsymbol{x}$ is the measurement variable with additive noise. f represents an unknown nonlinear function defined by a GP with zero mean and kernel function $k\left(\boldsymbol{t},\boldsymbol{t}'\right)$ and $\boldsymbol{\tau}$ is Gaussian noise with zero mean and variance $\sigma_n^2\boldsymbol{I}$. In the proposed method, the commonly employed square exponential covariance kernel function (Rasmussen and Williams, 2006) is chosen as the kernel function, denoted as

$$k\left(\boldsymbol{t},\boldsymbol{t}'\right)=\sigma_f^2\exp\left(\frac{1}{2}\left(\boldsymbol{t}-\boldsymbol{t}'\right)^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\left(\boldsymbol{t}-\boldsymbol{t}'\right)\right), \tag{10}$$

where σ_f is the process standard deviation, which controls its vertical scaling, $\boldsymbol{\Lambda}=\mathrm{diag}\left(l_1^2,l_2^2,\cdots,l_d^2\right)$, d is the number of elements, and $\boldsymbol{l}=[l_1,l_2,\cdots,l_d]$ is the input length-scale parameter, which informally can be thought of as roughly the distance for which it is required to travel in the input space before the function value can change significantly (Rasmussen and Williams, 2006). The hyperparameter set $\theta=\left\{\boldsymbol{l},\sigma_f^2,\sigma_n^2\right\}$ consists of the parameters within the kernel function and the measurement noise variance.

During the target tracking process, once the interconnection between adjacent data samples is comprehended, target state estimation can be performed based on this interrelation. The kernel function in GP can capture the relationship between different input points. Hence, the kernel function is employed to comprehend the link among the training data, enabling the modeling of the target's motion pattern. This learning process is conducted at each time step, allowing it to acquire any unknown function from the training data in real time. In the context of target tracking, the unknown function encompasses all potential maneuvering modes of the target.

3.2 Hyperparameter optimization

Taking the prediction at time t as an example, we use the nearest q-dimensional measurement $\boldsymbol{X}_t\triangleq[x_{t-q},x_{t-q+1},\cdots,x_{t-1}]$ as the training data. The input is $\boldsymbol{T}_t\triangleq[t-q,t-q+1,\cdots,t-1]$. The initial distribution of the training data is $p\left(f\mid x_{t-q:t-1}\right)=\mathcal{N}\left(\boldsymbol{\mu}_{t-1}^f,\boldsymbol{C}_{t-1}^f\right)$, with the mean and covariance being $\boldsymbol{\mu}_{t-1}^f=m\left(\boldsymbol{T}_t\right)$ and $\boldsymbol{C}_{t-1}^f=k\left(\boldsymbol{T}_t,\boldsymbol{T}_t\right)$, respectively.

Fixed hyperparameters are suitable only for specific scenarios. Hence, prior to making predictions, it is essential to learn from the training set to acquire the most suitable hyperparameters $\theta=\left\{\boldsymbol{l},\sigma_f^2,\sigma_n^2\right\}$. According to Rasmussen and Williams (2006), the potential hyperparameter based on the marginal likelihood can be written as

$$\begin{aligned}\log\left(\boldsymbol{X}_t\mid\boldsymbol{T}_t,\theta\right)=&-\frac{1}{2}\boldsymbol{X}_t^{\mathrm{T}}\boldsymbol{M}^{-1}\boldsymbol{X}_t-\frac{1}{2}\log|\boldsymbol{M}|\\&-\frac{n}{2}\log(2\pi),\end{aligned} \tag{11}$$

where $\boldsymbol{M}=k\left(\boldsymbol{T}_t,\boldsymbol{T}_t\right)+\sigma_n^2\boldsymbol{I}$ is the covariance matrix, $-\frac{1}{2}\boldsymbol{X}_t^{\mathrm{T}}\boldsymbol{M}^{-1}\boldsymbol{X}_t$ indicates the data fit to the observed target, the complexity penalty $\frac{1}{2}\log|\boldsymbol{M}|$ is determined solely by the covariance function and input, and $\frac{n}{2}\log(2\pi)$ is the normalization constant. "$|\cdot|$" represents the determinant of the matrix. The hyperparameters are learned using the maximum likelihood estimator. This involves calculating the partial derivative of $\log\left(\boldsymbol{X}_t\mid\boldsymbol{T}_t,\theta\right)$ and then using gradient descent to determine the optimal hyperparameters at time t.

3.3 Prediction

During the prediction process, we use the time-varying CV model instead of GP forecasting. The time-varying CV model involves the estimate of the

velocity based on the last two variables (x_{t-1} and x_{t-2}) of the training set $\boldsymbol{X}_t$ before a prediction is made. The velocity estimate $V_{x_{t-1}}$ for the x coordinate can be expressed as

$$V_{x_{t-1}} = (x_{t-1} - x_{t-2}) / T_{\mathrm{d}}, \tag{12}$$

where T_{d} is the time difference between x_{t-1} and x_{t-2}. In the same way, the velocity estimate $V_{y_{t-1}}$ of the y coordinate can be obtained. The state at time $t-1$ is given by $\boldsymbol{S}_{t-1} = \left[x_{t-1}, y_{t-1}, V_{x_{t-1}}, V_{y_{t-1}}\right]^{\mathrm{T}}$. The state and covariance prediction equations are

$$\tilde{\boldsymbol{S}}_t = \boldsymbol{F}\boldsymbol{S}_{t-1}, \tag{13}$$

$$\tilde{\boldsymbol{P}}_t = \boldsymbol{F}\boldsymbol{P}_{t-1}\boldsymbol{F}^{\mathrm{T}} + \boldsymbol{Q}, \tag{14}$$

where $\boldsymbol{F}$ is the state transition matrix, $\tilde{\boldsymbol{S}}_t = \left[\tilde{x}_t, \tilde{y}_t, \tilde{V}_{x_t}, \tilde{V}_{y_t}\right]^{\mathrm{T}}$ is the prediction vector, $\boldsymbol{P}_{t-1}$ is the covariance at time $t-1$, and $\boldsymbol{Q}$ is the process noise covariance.

3.4 Generalized probabilistic data association algorithm

In this subsection, we perform correlation matching between the measurement and the target at time t based on the prediction vector $\tilde{\boldsymbol{S}}_t$. The proposed method introduces the concept of deterministic association measurement of the GPDA algorithm (Pan et al., 2005; Guo et al., 2016). Under this concept, the belief is that both the measurement and target can be reused, and that a generalized joint event is constructed. The association probability is obtained according to the Bayesian rule, to determine the association measurement.

First, a tracking gate is created with the predicted vector $\tilde{\boldsymbol{S}}_t$ as its center, and the permissible measurements are determined based on this gate. This can be formulated as

$$\left(\boldsymbol{Z}_t(i) - \tilde{\boldsymbol{S}}_t\right)^{\mathrm{T}} \boldsymbol{B}_t \left(\boldsymbol{Z}_t(i) - \tilde{\boldsymbol{S}}_t\right) \le \gamma,\ i = 1, 2, \ldots, N, \tag{15}$$

$$\boldsymbol{B}_t = \boldsymbol{H}\tilde{\boldsymbol{P}}_t\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R}, \tag{16}$$

where $\boldsymbol{Z}_t(i)$ represents the i^{th} measurement at time t, $\boldsymbol{R}$ denotes the measurement noise covariance, $\boldsymbol{B}_t$ denotes the residual covariance, N represents the number of measurements, $\boldsymbol{H}$ denotes the observation matrix, and γ is the hypothesis testing threshold of χ^2. If inequality (15) holds true, it indicates that the measurements within the gate collectively contribute to the target.

Then, the association probability matrix $\boldsymbol{G}$ is computed based on the predicted state $\tilde{\boldsymbol{S}}_t$, the predicted covariance $\tilde{\boldsymbol{P}}_t$, the residual covariance $\boldsymbol{B}_t$, and the associated valid measurements (Pan et al., 2005):

$$\boldsymbol{G} = \begin{bmatrix} g_{00} & g_{01} & \cdots & g_{0J} \\ g_{10} & g_{11} & \cdots & g_{1J} \\ \vdots & \vdots & & \vdots \\ g_{m0} & g_{m1} & \cdots & g_{mJ} \end{bmatrix}, \tag{17}$$

where g_{ij} is the probability density between measurement i and target j, $i = 0, 1, \ldots, m$, $j = 0, 1, \ldots, J$, and m and J are the numbers of measurements and targets respectively. The calculation of $\boldsymbol{G}$ is divided into three parts: (1) The first row, $g_{0j} = (nV)^{-1}(1 - P_{\mathrm{D}}P_{\mathrm{G}})$, represents the probability density between measurement 0 and target j ($j \neq 0$), where P_{D} is the detection probability, P_{G} is the gate probability, V is the volume of the gate, and n is the coefficient. (2) The first column, $g_{i0} = \lambda$, represents the probability density between target 0 and measurement i ($i \neq 0$), where λ is the clutter density, target 0 means no target of interest, which may be a new target or a false target, and measurement 0 means no measurement (that is, the target is not detected). (3) The probability density between measurement i and target j ($i \neq 0, j \neq 0$) is

$$g_{ij} = P_{\mathrm{G}}^{-1} \left|2\pi\boldsymbol{B}_t\right|^{-1/2} \exp\left(-\frac{1}{2}\boldsymbol{v}_{ij}^{\mathrm{T}}\boldsymbol{B}_t^{-1}\boldsymbol{v}_{ij}\right), \tag{18}$$

where $\boldsymbol{v}_{ij} = \boldsymbol{Z}_{ij} - \tilde{\boldsymbol{S}}_t$, $\boldsymbol{Z}_{ij}$ represents the i^{th} measurement of target j at time t, and $\tilde{\boldsymbol{S}}_t$ is the prediction of target j at time t. Additionally, $g_{00} = 0$. Based on the target and measurement, the probability matrix $\boldsymbol{G}$ is normalized as

$$\begin{cases} \varepsilon_{ij} = \dfrac{g_{ij}}{\sum_{i=0}^{m} g_{ij}}, \\ \xi_{ij} = \dfrac{g_{ij}}{\sum_{j=0}^{J} g_{ij}}. \end{cases} \tag{19}$$

Finally, the correlation probability β_{ij} of measurement i and target j is calculated based on the normalized ε_{ij} and ξ_{ij}, to obtain the associated measurement $\boldsymbol{Z}_t = \sum_{i=0}^{m} \beta_{ij}\boldsymbol{Z}_{ij}$ of target j, where

$$\beta_{ij} = \frac{1}{c}\left(\rho\varepsilon_{ij}\prod_{\substack{h=0 \\ h\neq j}}^{J}\sum_{\substack{r=0 \\ r\neq i}}^{m}\varepsilon_{rh} + (1-\rho)\xi_{ij}\prod_{\substack{r=0 \\ r\neq i}}^{m}\sum_{\substack{h=0 \\ h\neq j}}^{J}\xi_{rh}\right). \tag{20}$$

Here, c is the normalization coefficient, ρ is the weight factor, h is the target index, and r is the measurement index.

3.5 Update

This subsection consists of two parts: the target state update and the training set update. The state update is similar to the Kalman filter update, and the state and covariance of the x coordinate are updated based on the x-component $\boldsymbol{x}_t$ of the associated measurement $\boldsymbol{Z}_t$. The update equations are

$$\hat{\boldsymbol{\mu}}_t^f = \tilde{\boldsymbol{x}}_t + \boldsymbol{K}_t\left(\boldsymbol{x}_t - \tilde{\boldsymbol{x}}_t\right), \tag{21}$$

$$\hat{\boldsymbol{C}}_t^f = \tilde{\boldsymbol{C}}_t^f - \boldsymbol{K}_t\tilde{\boldsymbol{C}}_t^f, \tag{22}$$

$$\boldsymbol{C}_t = \boldsymbol{C}_{t-1}^f\boldsymbol{J}_t^{\mathrm{T}}\left(\tilde{\boldsymbol{C}}_t^f\right)^{-1}\hat{\boldsymbol{C}}_t^f, \tag{23}$$

where $\boldsymbol{K}_t = \tilde{\boldsymbol{C}}_t^f\left(\tilde{\boldsymbol{C}}_t^f + \sigma_n^2\boldsymbol{I}\right)^{-1}$ is the Kalman gain, $\tilde{\boldsymbol{x}}_t$ is the x-coordinate component of state prediction $\tilde{\boldsymbol{S}}_t$, and $\tilde{\boldsymbol{C}}_t^f$ is the x-coordinate component of prediction covariance $\tilde{\boldsymbol{P}}_t$. $\boldsymbol{J}_t = k\left(t, \boldsymbol{T}_t\right)k\left(\boldsymbol{T}_t, \boldsymbol{T}_t\right)^{-1}$. $\boldsymbol{C}_t$ is the cross-covariance between training set and state estimation vector.

Additionally, the distribution of the training set is updated to obtain $p\left(f \mid x_{t-d+1:t}\right) = \mathcal{N}\left(\boldsymbol{\mu}_t^f, \boldsymbol{C}_t^f\right)$, to prevent GP from underfitting. The mean and covariance of the training set distribution at time t are

$$\boldsymbol{\mu}_t^f = \boldsymbol{\mu}_{t-1}^f + \boldsymbol{L}_t\left(\hat{\boldsymbol{\mu}}_t^f - \tilde{\boldsymbol{x}}_t\right), \tag{24}$$

$$\boldsymbol{C}_t^f = \boldsymbol{C}_{t-1}^f + \boldsymbol{L}_t\left(\hat{\boldsymbol{C}}_t^f - \tilde{\boldsymbol{C}}_t^f\right)\boldsymbol{L}_t^{\mathrm{T}}, \tag{25}$$

where $\boldsymbol{L}_t = \boldsymbol{C}_{t-1}^f\boldsymbol{J}_t^{\mathrm{T}}\left(\tilde{\boldsymbol{C}}_t^f\right)^{-1}$. Substituting $\hat{\boldsymbol{\mu}}_t^f$ and $\hat{\boldsymbol{C}}_t^f$ in Eqs. (21) and (22) into Eqs. (24) and (25) respectively, the mean and variance of the updated distribution are

$$\boldsymbol{\mu}_t^f = \boldsymbol{\mu}_{t-1}^f + \tilde{\boldsymbol{K}}_t\left(\boldsymbol{x}_t - \tilde{\boldsymbol{\mu}}_t^f\right), \tag{26}$$

$$\boldsymbol{C}_t^f = \boldsymbol{C}_{t-1}^f - \tilde{\boldsymbol{K}}_t\boldsymbol{J}_t\boldsymbol{C}_{t-1}^f, \tag{27}$$

$$\tilde{\boldsymbol{K}}_t = \boldsymbol{L}_t\boldsymbol{K}_t = \boldsymbol{C}_{t-1}^f\boldsymbol{J}_t^{\mathrm{T}}\left(\tilde{\boldsymbol{C}}_t^f + \sigma_n^2\boldsymbol{I}\right)^{-1}, \tag{28}$$

where $\tilde{\boldsymbol{K}}_t$ is the gain matrix.

The method proposed above, which combines elements of both the model-driven and data-driven approaches, is called the hybrid-driven GPMT and is summarized in Algorithm 1.

Algorithm 1 Hybrid-driven GPMT

Input: $\boldsymbol{X}_t$, $\boldsymbol{T}_t$, θ

Output: $\hat{\boldsymbol{\mu}}_t^f$, $\hat{\boldsymbol{C}}_t^f$, $\boldsymbol{\mu}_t^f$, $\boldsymbol{C}_t^f$, $\boldsymbol{C}_t$

1: Hyperparameter θ is learned via Eq. (11)
2: Compute the predicted state $\tilde{\boldsymbol{S}}_t$ via Eq. (13) and the predicted covariance $\tilde{\boldsymbol{P}}_t$ via Eq. (14)
3: Compute the associated measurement $\boldsymbol{Z}_t$ according to inequality (15) and Eqs. (16)–(20)
4: Compute mean $\hat{\boldsymbol{\mu}}_t^f$ and covariance $\hat{\boldsymbol{C}}_t^f$ of the estimated state via Eqs. (21) and (22), respectively
5: Compute cross-covariance $\boldsymbol{C}_t$ via Eq. (23)
6: Compute mean $\boldsymbol{\mu}_t^f$ and covariance $\boldsymbol{C}_t^f$ of the training set by Eqs. (26) and (27), respectively

4 Performance validation

In this section, the advantages of the proposed method have been verified under the prevalence of challenging scenarios. Section 4.1 provides an introduction to these scenarios and the configuration of parameters. The comparison methods and their parameter settings are given in Section 4.2, while the simulation results are discussed in Section 4.3.

4.1 Testing scenarios and parameter settings

In a two-dimensional space, the target surveillance region has a size of $V = \pi r^2$, with $r = 50$ km, and the motion state of the target denoted as $\boldsymbol{X} = [x, y, V_x, V_y]^{\mathrm{T}}$ includes the position and velocity components. There are four targets with randomly initialized positions. The maximum acceleration for target maneuvering is 8×9.8 m/s^2. The range of the angular velocity ω of the maneuver is $[-18°, 18°]$. The acceleration and angular velocity for each target maneuver are randomly set. The duration of each maneuver is randomly determined within $[8, 10]$ s. The target velocity is within $[100, 680]$ m/s. The initial target velocity is randomly assigned within $[100, 300]$ m/s. The detection probability is given as $P_{\mathrm{D}} = 0.98$. Clutter returns can be generated using a Poisson process with $\mathcal{K}(z) = \lambda V c(z)$, where the clutter density is $\lambda = 0.5 \times 10^{-7}$ m^{-2}. The clutter is assumed to have a uniform spatial distribution throughout the surveillance region, as represented by $c(z)$. Given the unknown target motion model, the state equation can be formulated as

$$\boldsymbol{X}_t = \begin{cases} \boldsymbol{F}_1\boldsymbol{X}_{t-1} + \boldsymbol{U}\boldsymbol{\varphi}_{t-1}, & \boldsymbol{a} = \boldsymbol{0},\ \omega = 0, \\ \boldsymbol{F}_2\boldsymbol{X}_{t-1} + \boldsymbol{D}\boldsymbol{a} + \boldsymbol{U}\boldsymbol{\varphi}_{t-1}, & \boldsymbol{a} \neq \boldsymbol{0},\ \omega = 0, \\ \boldsymbol{F}_3\boldsymbol{X}_{t-1} + \boldsymbol{U}\boldsymbol{\varphi}_{t-1}, & \boldsymbol{a} = \boldsymbol{0},\ \omega \neq 0, \\ f\left(\boldsymbol{X}_{t-1}\right) + \boldsymbol{U}\boldsymbol{\varphi}_{t-1}, & \boldsymbol{a} \neq \boldsymbol{0},\ \omega \neq 0. \end{cases} \tag{29}$$

The state equation consists of different models represented by the first to the fourth rows, corresponding to the CV, constant acceleration (CA), coordinated turn (CT), and nonlinear models, respectively. $\boldsymbol{a}=[a_x,a_y]^{\mathrm{T}}$ is the acceleration vector, where a_x and a_y represent the acceleration components of the x- and y-coordinate axes respectively. ω is the angular velocity. $\boldsymbol{F}_1$, $\boldsymbol{F}_2$, and $\boldsymbol{F}_3$ are expressed in terms of

$$\boldsymbol{F}_1=\boldsymbol{F}_2=\begin{bmatrix}1&0&T_{\mathrm{data}}&0\\0&1&0&T_{\mathrm{data}}\\0&0&1&0\\0&0&0&1\end{bmatrix}, \tag{30}$$

$$\boldsymbol{F}_3=\begin{bmatrix}1&0&\frac{\sin(\omega T_{\mathrm{data}})}{\omega}&\frac{\cos(\omega T_{\mathrm{data}})-1}{\omega}\\0&1&\frac{1-\cos(\omega T_{\mathrm{data}})}{\omega}&\frac{1-\cos(\omega T_{\mathrm{data}})}{\omega}\\0&0&\cos(\omega T_{\mathrm{data}})&-\sin(\omega T_{\mathrm{data}})\\0&0&\sin(\omega T_{\mathrm{data}})&\cos(\omega T_{\mathrm{data}})\end{bmatrix}. \tag{31}$$

The state transition function of the nonlinear model is denoted by f. $\boldsymbol{X}_{t-1}$ and $\boldsymbol{X}_t$ represent the target states. The control matrix $\boldsymbol{D}$ and process noise matrix $\boldsymbol{U}$ are given by

$$\boldsymbol{D}=\begin{bmatrix}\frac{T_{\mathrm{data}}^2}{2}&0\\0&\frac{T_{\mathrm{data}}^2}{2}\\T_{\mathrm{data}}&0\\0&T_{\mathrm{data}}\end{bmatrix},\boldsymbol{U}=\begin{bmatrix}\frac{T_{\mathrm{data}}^2}{2}&0\\0&\frac{T_{\mathrm{data}}^2}{2}\\T_{\mathrm{data}}&0\\0&T_{\mathrm{data}}\end{bmatrix}. \tag{32}$$

The process noise $\boldsymbol{\varphi}_{t-1}$ has a covariance matrix given by

$$\boldsymbol{Q}_{\boldsymbol{\varphi}}=\begin{bmatrix}1^2&0\\0&1^2\end{bmatrix}. \tag{33}$$

The measurement equation of the target is

$$\boldsymbol{Z}_t=\boldsymbol{H}\boldsymbol{X}_{t-1}+\boldsymbol{\eta}, \tag{34}$$

where

$$\boldsymbol{H}=\begin{bmatrix}1&0&0&0\\0&1&0&0\end{bmatrix}, \tag{35}$$

$\boldsymbol{\eta}$ is the measurement noise, and its covariance matrix is

$$\boldsymbol{R}_{\boldsymbol{\eta}}=\begin{bmatrix}\xi_x^2&0\\0&\xi_y^2\end{bmatrix}. \tag{36}$$

Here, $\xi_x=\xi_y=300$ m.

The simulation time is 100 s. The sampling period $T_{\mathrm{data}}=1$ s. The performance of the tracking method is evaluated through 500 Monte Carlo simulations. The root mean square error (RMSE) is

$$\mathrm{RMSE}_k=\sqrt{\frac{1}{N_{\mathrm{MC}}}\sum_{i=1}^{N_{\mathrm{MC}}}(q_{ik}-\hat{q}_{ik})^2}, \tag{37}$$

where N_{MC} is the number of Monte Carlo simulations, and q_{ik} and $\hat{q}_{ik}$ are the true and estimated values, respectively.

4.2 Compared methods

The popular model-driven IMM and the data-driven GPMT are chosen as the comparison methods. For the proposed method and GPMT, the training set, measurement data, and noise standard deviations are scaled down at the input. The state estimate vector is upscaled at the output. The scaling is $\frac{1}{1000}$. The training set consists of the first $q=8$ measurement data samples. The hyperparameter set $\theta=\left\{\boldsymbol{l},\sigma_f^2,\sigma_n^2\right\}$ is initialized by maximizing the likelihood estimator of the training set, where $\boldsymbol{l}$ and σ_f^2 are the length scale and kernel variance hyperparameters, respectively. $\sigma_n=300$ m is the measurement noise variance hyperparameter. Following each iteration, the oldest data element in the training set is removed and a new state estimate is added to the training set. In GPDA, $\rho=0.5$.

For IMM, as there is no prior knowledge about the target motion, a standard combination of models (CV, CA, and CT) is chosen, which encompasses 31 models (1 CV model, 24 CA models, and 6 CT models). For the CV and CA models, the acceleration vector $\boldsymbol{a}$ satisfies

$$\begin{aligned}\boldsymbol{a}\in\{&[0,0]^{\mathrm{T}},[0,10]^{\mathrm{T}},[0,-10]^{\mathrm{T}},[10,0]^{\mathrm{T}},[-10,0]^{\mathrm{T}},\\&[10,10]^{\mathrm{T}},[10,-10]^{\mathrm{T}},[-10,10]^{\mathrm{T}},[-10,-10]^{\mathrm{T}},\\&[0,20]^{\mathrm{T}},[0,-20]^{\mathrm{T}},[20,0]^{\mathrm{T}},[-20,0]^{\mathrm{T}},\\&[20,20]^{\mathrm{T}},[20,-20]^{\mathrm{T}},[-20,20]^{\mathrm{T}},[-20,-20]^{\mathrm{T}},\\&[0,40]^{\mathrm{T}},[0,-40]^{\mathrm{T}},[40,0]^{\mathrm{T}},[-40,0]^{\mathrm{T}},\\&[40,40]^{\mathrm{T}},[40,-40]^{\mathrm{T}},[-40,40]^{\mathrm{T}},[-40,-40]^{\mathrm{T}}\}.\end{aligned} \tag{38}$$

The first element in the set corresponds to the CV model, and the other elements correspond to the CA models. For the CT model, the set of the angular velocity w is $\{5^\circ,-5^\circ,10^\circ,-10^\circ,20^\circ,-20^\circ\}$. The measurement noise covariance matrix is as presented in Eq. (36), the model probability is $\vartheta=\frac{1}{31}$, and the probability transition matrix is given by

$\boldsymbol{P}_{\text{tran}} = 0.5\boldsymbol{I}_{31} + \frac{0.5}{30}(\boldsymbol{A} - \boldsymbol{I}_{31})$. All elements within matrix $\boldsymbol{A}$ have a value of 1, while matrix $\boldsymbol{I}_{31}$ is a 31^{st}-order identity matrix. The state noise covariance matrix is set as $\boldsymbol{E} = 400^2\boldsymbol{I}_4$, while the process noise matrix is as presented in Eq. (32).

4.3 Results

Fig. 1 presents the tracking results obtained from the proposed method, demonstrating its ability to achieve satisfactory tracking performance for highly maneuvering targets in cluttered environments. RMSE values of the proposed method, GPMT, and IMM are depicted in Fig. 2. The mean RMSE is shown in Table 1. It can be seen that the proposed method exhibits a clear advantage in position estimation compared to the GPMT and IMM methods. Both IMM and GPMT have a poor tracking performance. The reason is that the monitored targets are being highly maneuverable in a cluttered environment. Their maneuvering attributes encompass wider parameter ranges and display rapid changes. The tracking effectiveness of IMM is constrained by the collection of predefined models. The precision of tracking diminishes when the predefined model set fails to encompass the complete motion pattern of the target. IMM is unable to encompass all the maneuvering modes of highly maneuvering targets using a predefined set of models.

Furthermore, another crucial factor contributing to reduced tracking accuracy of IMM and GPMT is the presence of background clutter. Due to clutter, prediction errors are magnified, subsequently affecting data association and the state update process. With the passage of time, errors will progressively accumulate. To sum up, the poor tracking performance of both IMM and GPMT can be primarily attributed to the presence of clutter and the inherent constraints of the algorithms. The proposed method integrates the time-varying CV model into GPMT to improve the prediction error and attain satisfactory tracking performance.

However, there is no significant difference between the three methods in velocity estimation, and this is attributable to the fact that velocity measurement is derived from the difference in position measurements. However, for highly maneuvering targets, this method of obtaining velocity measurements may not yield accurate results. Notably, the proposed method estimates the elements in the state vector (x, y, V_x, V_y) independently.

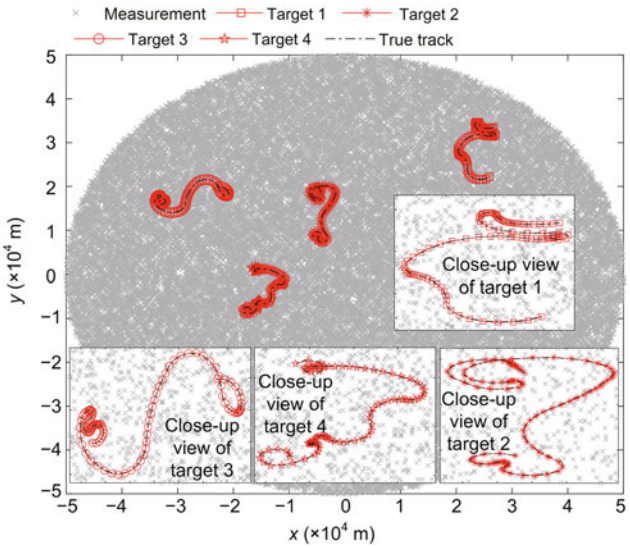

Fig. 1 True target tracks and position estimates

Table 1 Mean root mean square error

Method	x (m)	y (m)	V_x (m/s)	V_y (m/s)
IMM	4622.23	3893.82	242.07	217.04
GPMT	7518.37	7234.40	238.78	243.84
Ours	297.48	297.44	215.36	227.07

Table 2 Processing time

Method	Time (s)
IMM	0.0219
GPMT	0.1829
Ours	0.1816

The code is executed on a Windows 10 computer with an i7-9750H 2.6 GHz CPU. The Matlab version is 2020b. Table 2 displays the processing time involved in the execution of each iteration under each of the methods, that is, IMM, GPMT, and the proposed method, for a total of 500 Monte Carlo simulations. The processing time of the proposed method falls between those of the IMM and GPMT methods.

5 Conclusions

This paper presents a novel approach for highly maneuvering multi-target tracking, combining the strengths of data-driven GP learning and the stability of the time-varying CV model prediction. The proposed hybrid-driven method outperforms existing techniques such as IMM and GPMT under challenging scenarios involving highly maneuvering targets in cluttered environments.

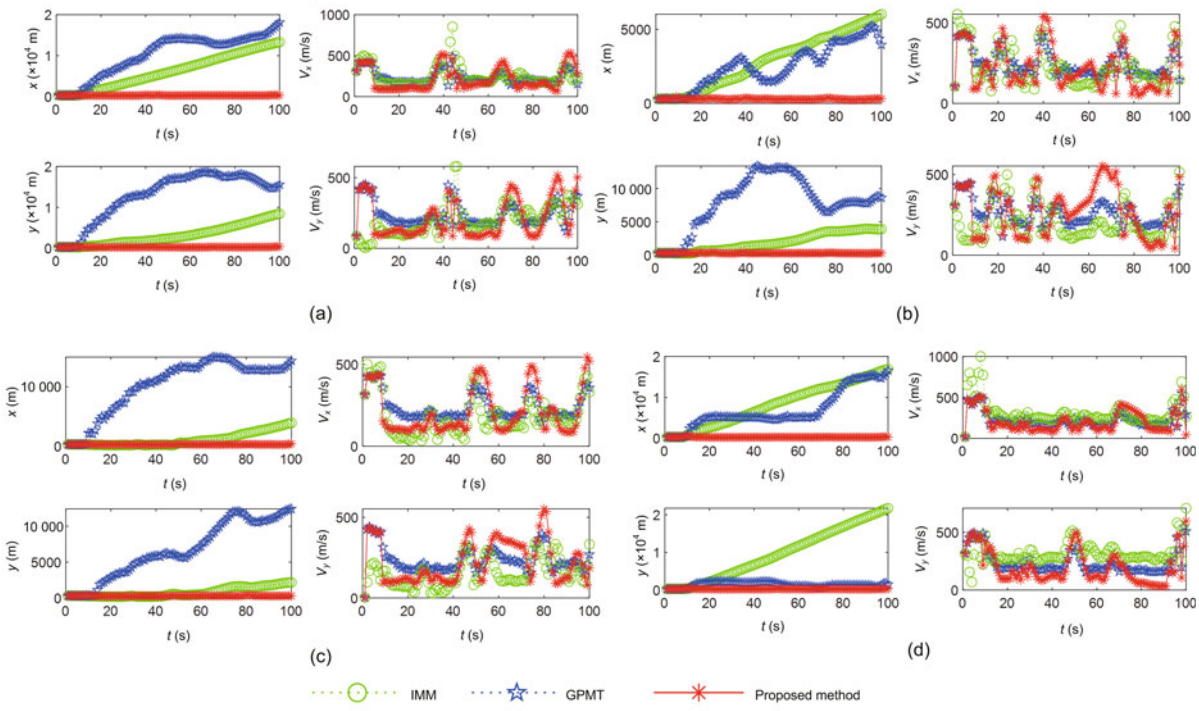

Fig. 2 Root mean square error of the estimated states for targets 1 (a), 2 (b), 3 (c), and 4 (d)

Contributors

Qiang GUO and Long TENG designed the research and addressed the problems. Long TENG processed the data and drafted the paper. Yunfei GUO and Tianxiang YIN helped with the technical information. Xinliang WU and Wenming SONG helped organize the paper and supervised the study. Long TENG revised and finalized the paper.

Compliance with ethics guidelines

Qiang GUO, Long TENG, Tianxiang YIN, Yunfei GUO, Xinliang WU, and Wenming SONG 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

Aftab W, Mihaylova L, 2020. On the impact of different kernels and training data on a Gaussian process approach for target tracking. Proc IEEE 23rd Int Conf on Information Fusion, p.1-6. https://doi.org/10.23919/FUSION45008.2020.9190413

Aftab W, Mihaylova L, 2021. A learning Gaussian process approach for maneuvering target tracking and smoothing. *IEEE Trans Aerosp Electron Syst*, 57(1):278-292. https://doi.org/10.1109/TAES.2020.3021220

Da K, Li TC, Zhu YF, et al., 2021. Recent advances in multisensor multitarget tracking using random finite set. *Front Inform Technol Electron Eng*, 22(1):5-24. https://doi.org/10.1631/FITEE.2000266

Deng LC, Li D, Li RF, 2020. Improved IMM algorithm based on RNNs. *J Phys Conf Ser*, 1518:012055. https://doi.org/10.1088/1742-6596/1518/1/012055

Guo YF, Fan KS, Peng DL, et al., 2015. A modified variable rate particle filter for maneuvering target tracking. *Front Inform Technol Electron Eng*, 16(11):985-994. https://doi.org/10.1631/FITEE.1500149

Guo YF, Tharmarasa R, Rajan S, et al., 2016. Passive tracking in heavy clutter with sensor location uncertainty. *IEEE Trans Aerosp Electron Syst*, 52(4):1536-1554. https://doi.org/10.1109/TAES.2016.140820

Guo YF, Li Y, Tharmarasa R, et al., 2019. GP-PDA filter for extended target tracking with measurement origin uncertainty. *IEEE Trans Aerosp Electron Syst*, 55(4):1725-1742. https://doi.org/10.1109/TAES.2018.2875555

Guo YF, Li Y, Ren X, et al., 2020a. Multiple maneuvering extended target tracking based on Gaussian process. *Acta Autom Sin*, 46(11):2392-2403 (in Chinese). https://doi.org/10.16383/j.aas.c180849

Guo YF, Li Y, Xue AK, et al., 2020b. Simultaneous tracking of a maneuvering ship and its wake using Gaussian processes. *Signal Process*, 172:107547. https://doi.org/10.1016/j.sigpro.2020.107547

Guo YF, Zhu JJ, Zhou S, et al., 2022. A joint model and data driven track segment association algorithm for manoeuvring target tracking. *IET Radar Sonar Nav*, 16(10):1670-1680. https://doi.org/10.1049/rsn2.12288

Huber MF, 2014. Recursive Gaussian process: on-line regression and learning. *Patt Recogn Lett*, 45:85-91. https://doi.org/10.1016/j.patrec.2014.03.004

Li TC, Hlawatsch F, 2021. A distributed particle-PHD filter using arithmetic-average fusion of Gaussian mixture parameters. *Inform Fusion*, 73:111-124. https://doi.org/10.1016/j.inffus.2021.02.020

Li TC, Su JY, Liu W, et al., 2017. Approximate Gaussian conjugacy: parametric recursive filtering under nonlinearity, multimodality, uncertainty, and constraint, and beyond. *Front Inform Technol Electron Eng*, 18(12):1913-1939. https://doi.org/10.1631/FITEE.1700379

Li TC, Liu ZG, Pan Q, 2019a. Distributed Bernoulli filtering for target detection and tracking based on arithmetic average fusion. *IEEE Signal Procss Lett*, 26(12):1812-1816. https://doi.org/10.1109/LSP.2019.2950588

Li TC, Chen HM, Sun SD, et al., 2019b. Joint smoothing and tracking based on continuous-time target trajectory function fitting. *IEEE Trans Autom Sci Eng*, 16(3):1476-1483. https://doi.org/10.1109/TASE.2018.2882641

Liu JX, Wang ZL, Xu M, 2020. DeepMTT: a deep learning maneuvering target-tracking algorithm based on bidirectional LSTM network. *Inform Fusion*, 53:289-304. https://doi.org/10.1016/j.inffus.2019.06.012

Liu XC, Lyu C, George J, et al., 2022. A learning distributed Gaussian process approach for target tracking over sensor networks. Proc 25th Int Conf on Information Fusion, p.1-8. https://doi.org/10.23919/FUSION49751.2022.9841315

Pan Q, Ye XN, Zhang HC, 2005. Generalized probability data association algorithm. *Acta Electron Sin*, 33(3):467-472. https://doi.org/10.3321/j.issn:0372-2112.2005.03.021

Rasmussen CE, Williams CKI, 2006. Gaussian Processes for Machine Learning. MIT Press, Cambridge, USA.

Sun MW, Davies ME, Proudler I, et al., 2020. A Gaussian process based method for multiple model tracking. Proc Sensor Signal Processing for Defence Conf, p.1-5. https://doi.org/10.1109/SSPD47486.2020.9272174

Tian WM, Fang LL, Li WD, et al., 2022. Deep-learning-based multiple model tracking method for targets with complex maneuvering motion. *Remote Sens*, 14(14):3276. https://doi.org/10.3390/rs14143276

Wang LP, Zhan RZ, Huang Y, et al., 2021. Joint tracking and classification of extended targets with complex shapes. *Front Inform Technol Electron Eng*, 22(6):839-861. https://doi.org/10.1631/FITEE.2000061

Wu WH, Cai YC, Jin HB, et al., 2021. Derivation of the multi-model generalized labeled multi-Bernoulli filter: a solution to multi-target hybrid systems. *Front Inform Technol Electron Eng*, 22(1):79-87. https://doi.org/10.1631/FITEE.2000105

Xiong W, Zhu HF, Cui YQ, 2022. Recurrent adaptive maneuvering target tracking algorithm based on online learning. *Acta Aeronaut Astronaut Sin*, 43(5):325250 (in Chinese). https://doi.org/10.7527/S1000-6893.2021.25250

Zhang D, Liu MQ, Zhang SL, et al., 2018. Mutual-information based weighted fusion for target tracking in underwater wireless sensor networks. *Front Inform Technol Electron Eng*, 19(4):544-556. https://doi.org/10.1631/FITEE.1601695

Zhang XR, He FH, Zheng TY, 2019. An LSTM-based trajectory estimation algorithm for non-cooperative maneuvering flight vehicles. Proc Chinese Control Conf, p.8821-8826. https://doi.org/10.23919/ChiCC.2019.8866249

Zheng Z, Cai SC, 2021. A collaborative target tracking algorithm for multiple UAVs with inferior tracking capabilities. *Front Inform Technol Electron Eng*, 22(10):1334-1350. https://doi.org/10.1631/FITEE.2000362

Zhou R, Feng Y, Bin D, et al., 2020. Multi-UAV cooperative target tracking with bounded noise for connectivity preservation. *Front Inform Technol Electron Eng*, 21(10):1494-1503. https://doi.org/10.1631/FITEE.1900617

Zhu Y, Liang S, Wu XJ, et al., 2021. A random finite set based joint probabilistic data association filter with non-homogeneous Markov chain. *Front Inform Technol Electron Eng*, 22(8):1114-1126. https://doi.org/10.1631/FITEE.2000209