Journal of Network Intelligence ©2024 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **9**, Number **2**, May **2024**

A Motion Tracker Base on Sparse Gaussian Process Regression and Gradient Descent Method

Hai-Hua Xie
School of Mechanical Engineering and Electronic Information
China University of Geosciences
Wuhan, 430070, P.R.China
xie_haihua@edu.cug.cn

Yi Yuan*
School of Mechanical Engineering and Electronic Information
China University of Geosciences
Wuhan, 430070, P.R.China
School of Electromechanical
Beijing Industry and Trade Technician College
Beijing, 100080, P.R.China
yuan0387@126.com

San-You Zeng
School of Mechanical Engineering and Electronic Information
China University of Geosciences
Wuhan, 430070, P.R.China
sanyouzeng@gmail.com

*Corresponding author: Yi Yuan

Received July 17, 2023, revised October 21, 2023, accepted January 14, 2024.

Abstract. *In practical tracking scenarios, the ability of model-based trackers to estimate the potential target trajectories is constrained by their predetermined target tracking models. A model-free tracker with the ability to learn unknown parameters online is presented in this paper, named Gaussian process regression tracker. Different from the model-based trackers, the proposed tracker online learns the highly nonlinear motion model using sparse Gaussian process regression. Firstly, sparse Gaussian process regression combined with gradient descent method was used to estimate the motion model online, and then incorporated into the Kalman filter framework to achieve estimation and prediction of the target motion state. Monte Carlo experiments show that the proposed tracker provides an accuracy of the predicted and estimated target states in six challenging maneuver test scenarios.*

Keywords: Target Tracking, Gaussian Process, Stochastic Gradient Descent, Recursive Estimation.

1. **Introduction.** Target tracking (TT) [1, 2] has been at the forefront task, such as radar tracking, navigation control, and driverless. Its goals are to determine unknown number trajectories of the target from sensor measurement and predict the state of the targets of interest in a noisy environment. Considering target characteristics, TT can be mainly categorized into three groups, point target [3], extended target, and group target [4, 5, 6]. Generally, point target tracking only involves target kinematics estimation, whereas extended target tracking and grouped target tracking focus on target kinematics and shape

estimation. Point target tracking models for kinematics estimation provide a solution for two major problems, i.e., data association and state estimation problems [7]. The data association methods is necessary to assign measurements to corresponding targets. The estimation updates the target state by utilizing the corresponding measurement. This paper presents an approach that specifically focuses on tracking and state estimation.

The state estimation problem is commonly addressed using a Bayesian filter framework. This framework consists of two components: the motion model, which represents the motion space, and the measurement model, which represents the measurement space. In the field of motion analysis, several motion models have been created to estimate the dynamic path of the target. The motion models are divided into the linear model and the non-linear model. For the linear model, such as constant velocity (CV) model and constant acceleration (CA) model, the widely known Kalman filter (KF) [8] is the optimal recursive solution. For the non-linear model, such as coordinate turn (CT) model, extended KF [9], unscented KF [10], and interactive multiple model (IMM) [11] have been proposed to address this problem. The IMM (Interacting Multiple Model) is a cost-effective nonlinear filter [12]. When dealing with sophisticated motion behaviors of the target, such as time-varying and doubly-stochastic movements, it becomes necessary to use a diverse set of models with varying parameters to accurately represent the potential motion patterns. This, in turn, results in a significant increase in computational complexity. To overcome this downside, a formula for estimating the motion parameters of the target was established in [13], and the method is applied to a multi-target Bayes filter, developing a filter with adaptive estimation of the parameters. In [14], applying the adaptive estimation method to Cardinality-balanced multi-target multi-Bernoulli filter, the model with an unknown parameter can be modified adaptively by using the selected parameter particles.

All of the above motion model-based filters require to operate in the known motion model. In practical scenarios, there may be a significant mismatch between the motion model and the desired motion behavior. If this happen, the tracking performance would degrade or even unacceptable. In [15], the first model-free filter was proposed, named Gaussian process (GP) motion tracker. Motion models can be represents by GP, a powerful kernelized technique. In existing historical measurement data, the GP can predict and update the position and velocity of the target. However, the tracker is a non-recursive method, which can be caused high computational complexity costs. To this end, in [16], the recursive and learning GP motion tracker was proposed by exploiting the recursive GP. Replace historical measurement data with a sparse GP induced point set. In the many open publication, GP approaches has been applied to many areas, such as system identification [17] nonlinear ARX models [18] ,and evolutionary algorithm [19]. Time series prediction and estimation are the weak points of GP. This is because a new sample are added to existing set, the joint probability density of the set is calculated repeatedly. Similarly, target state estimation is a time series application. Currently, GP-based methods are commonly used for extended target tracking to estimate the shape, while model-based methods are employed for target kinematics estimation. However, GP-based approaches have not been extensively studied for kinematics estimation due to two main challenges: 1) Difficulty in incorporating new measurements and 2) Difficulty in online learning of hyperparameters in a recursive version.

In GP community, the sparse approximation methods are commonly used to construct recursive GP in a wealth of research to solve the former [20, 21], such as Fully Independent Training Conditional (FITC), Power Expectation Propagation (PEP) [21], and Deterministic Training Conditional (DTC) [23] .et.al. Such approximation is based on inducing points methods, where the unknown function is represented by its values at a

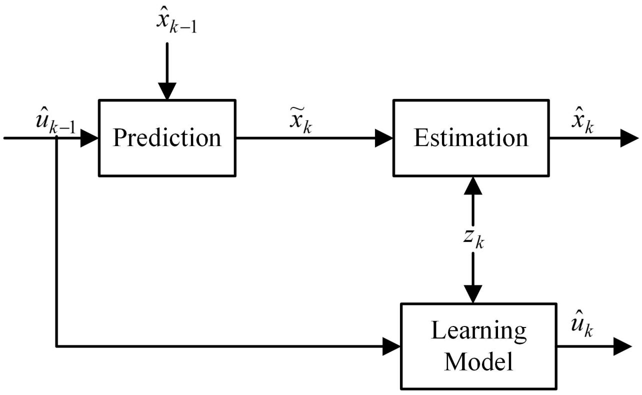


Figure 1. The proposed tracker block diagram. $\boldsymbol{u}$ denotes the inducing point or state-space model, $\boldsymbol{x}$ represents the target state, $\boldsymbol{z}$ is the measurement, k is the time index. $\tilde{}$ and $\hat{}$ represent, respectively, the predicted and estimated states. The learning model process uses $\hat{\boldsymbol{u}}_{k-1}$ as a priori and incorporate measurement $\boldsymbol{z}_k$ to learn the state-space model.

set of $M \ll N$ pseudoinputs (called inducing points), where M denotes the number of inducing point, N is the number of samples set. Instead, hyperparameters online learning methods are rare, but also contain state space framework and Stochastic gradient descent (SGD). We integrate sparse recursion GP and SDG to build the target tracker in this paper, the block diagram as shown in figure 1.

2. Background.

2.1. Problem Formulation.

Consider a 2-D horizontal motion model with additive Gaussian noise. The following gives the motion and measurement models of discrete-time targets:

$$\boldsymbol{x}_k = f_{k|k-1}(\boldsymbol{x}_{k-1}) + \boldsymbol{v}_k, \tag{1}$$

$$\boldsymbol{z}_k = h(\boldsymbol{x}_k) + \boldsymbol{n}_k. \tag{2}$$

Here, $\boldsymbol{x}_k = [p_x^k, p_y^k, v_x^k, v_y^k]^T$ represents the target state at time index k, $\boldsymbol{z}_k$ is the measurement at time index k detected by the sensor. The $\boldsymbol{v}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q})$ and $\boldsymbol{n}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R})$ are the additive process and measurement Gaussian noise vectors, respectively, where $\boldsymbol{Q} = q \cdot \boldsymbol{I}_{4\times4}$ and $\boldsymbol{R} = r \cdot \boldsymbol{I}_{4\times4}$, where q, r, and $\boldsymbol{I}_{4\times4}$ denote the corresponding noise variance and identity matrix. The $f(\cdot)$ and $h(\cdot)$ represent the state motion model and the measurement model, respectively. The measurement model is assumed to be a linear model in this paper, given as follows:

$$\boldsymbol{z}_k = H\boldsymbol{x}_k + \boldsymbol{n}_k, \tag{3}$$

where,

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4}$$

According to Bayes' rule, the $\boldsymbol{x}_k$ under the conditioned on the measurements $\boldsymbol{z}_{1:k}$ is,

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) = \frac{1}{c} g(\boldsymbol{z}_k|\boldsymbol{x}_k) \cdot p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}) \tag{5}$$

where c is the normalization constant, $g(\boldsymbol{z}_k|\boldsymbol{x}_k) = \mathcal{N}(H\boldsymbol{x}_k, R)$ denotes the likelihood function of the sensor under the state $\boldsymbol{x}_k$, $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1})$ is the predictive probability density function.

In real-world situations, the success of target tracking is primarily dependent on how well the motion model matches the unidentified target dynamics. As a result, the following will introduce the GP, which is utilized to represent the target's motion process by applying GP priors to the unknown functions $f_{k|k-1}(\cdot)$.

2.2. **Regular and Sparse GP Regression.** GP is a set of infinite Gaussian random variables, any realization that existing data fit the unknown functions of which are jointly Gaussian distributed. As a non-parametric method, the Gaussian Process (GP) is used to solve regression and classification problems by mapping from the input to the output space. The GP is characterized by its mean function and covariance kernel function, which serves as a prior. In many applications, the mean functions consist of two types: those with nonzero constants and those with zero constants. Therefore, the covariance kernel function is a crucial parameter in GP-based models.

A GP is used to fit a unknown function $f(\cdot)$ given as follows

$$\boldsymbol{z} = f(\boldsymbol{x}) + \boldsymbol{\epsilon}, \tag{6}$$

$$f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), K(\boldsymbol{x}, \boldsymbol{x}')), \tag{7}$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}), \tag{8}$$

where $f(\boldsymbol{x})$ and $\boldsymbol{z}$ denote, respectively, the true function values and noisy observation of the function at the input $\boldsymbol{x}$, $\boldsymbol{\epsilon}$ denote the observation noise, $\mathcal{GP}(m(\boldsymbol{x}), K(\boldsymbol{x}, \boldsymbol{x}'))$ represents a GP with a mean and a covariance kernel.

Given a training data set with N pair of training points $\mathcal{D}_N\left\{\boldsymbol{x}, \boldsymbol{z}\right\} = \{(x_1, z_1), (x_2, z_2), ..., (x_N, x_N)\}$. The goal of GP is to predict the latent distribution of the function $\boldsymbol{f}^* = [f_1^*, ..., f_L^*]^T$ at the test inputs $\boldsymbol{x}^* = [x_1^*, \dots, x_L^*]^T$, conditioned on the training data set. The latent distribution at test input are

$$\boldsymbol{f}^*|\mathcal{D} \sim \mathcal{N}(\mathbb{E}(\boldsymbol{f}^*), \mathbb{D}(\boldsymbol{f}^*)) \tag{9}$$

$$\mathbb{E}(\boldsymbol{f}^*) = m(\boldsymbol{x}^*) + K_{*x} \cdot (K + \sigma^2 \boldsymbol{I})^{-1}(\boldsymbol{f} - m(\boldsymbol{x})) \tag{10}$$

$$\mathbb{D}(\boldsymbol{f}^*) = K_{**} - K_{*x} \cdot (K + \sigma^2 \boldsymbol{I})^{-1} K_{*x}^T \tag{11}$$

where $\mathbb{E}(\boldsymbol{f}^*)$ and $\mathbb{D}(\boldsymbol{f}^*)$ are, respectively, the predicted mean and the covariance at test input point $\boldsymbol{x}^*$, K_{pq} represents the GP covariance matrix between the input $\boldsymbol{p}$ and $\boldsymbol{q}$, $\cdot^{-1}$ and $\cdot^T$ denote the matrix inverse and transpose, respectively. $\boldsymbol{I}$ represents the identity matrix.

An important downside of GP regression is its computational costs of $\mathcal{O}(N^3)$, with N the size of training data set. A common solution so-called sparse GP regression methods are used to solve this. To apply sparse method, an additional set of $M \ll N$ latent random variables $\boldsymbol{u} \in \mathbb{R}^M$ at the input $\boldsymbol{u_f} = \left\{u_f^1, u_f^2, ..., u_f^M\right\}$ are introduced, which are called the inducing variables or inducing points. The inducing input is in the one-dimension domain, and the output $\boldsymbol{u} = f(\boldsymbol{u_f})$ is the corresponding GP function value. The joint prior $\boldsymbol{f}$ and $\boldsymbol{f}^*$ is augmented with the inducing variables $\boldsymbol{u}$. By marginalizing out the inducing variables, the original prior $p(\boldsymbol{f}, \boldsymbol{f}^*) = \int p(\boldsymbol{f}, \boldsymbol{f}^*|\boldsymbol{u})p(\boldsymbol{u})d\boldsymbol{u}$ is recovered. Mathematically, all sparse models suppose that $\boldsymbol{f}$ and $\boldsymbol{f}^*$ are conditionally independent, given $\boldsymbol{u}$. Consequently, the computational costs of these sparse models can be done $\mathcal{O}(MN^2)$.

2.3. **SGD.** SGD is a simple and very effective optimization method, which is often used for parameter learning of linear systems under convex functions such as support vector machines and neural networks. Take the minimization objective function $\boldsymbol{\psi}(\boldsymbol{\theta})$ as an example, stochastic gradient descent methods involve finding the parameters which minimize a mathematical function. The learning process of the parameter can be described as follows:

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \alpha \frac{\partial}{\partial \boldsymbol{\theta}} \boldsymbol{\psi}(\boldsymbol{\theta}) \tag{12}$$

where α denotes the learning rate.

3. **The Improved Method.** In this section, the recursive and online learning Gaussian process motion tracker (GPMT) is presented. The tracker consists of two blocks, namely, learning of state-space motion model and the target state estimating and predicting online. The former is represented by the *inducing points* of the sparse GP regression. The proposed tracker contains two assumptions, which will be integrated into the KF framework.

1) the target motion stochastic process, in the x and y axis, are assumed to be independent, and they have the same common GP distribution function.

2) The correlation between two samples is determined by the distance between input points. The farther apart these two points are, the weaker the correlation.

The trackers are described in detail below. Considering the assumption (1), the GPMT formulation of position for x direction and y direction is given below.

$$\boldsymbol{x} = f^x(\boldsymbol{t}), \quad f^x \sim \mathcal{GP}^x(\mathbf{0}, \boldsymbol{K}^x(\boldsymbol{t}, \boldsymbol{t}')) \tag{13}$$

$$\boldsymbol{z}_t^x = f^x(\boldsymbol{t}) + \boldsymbol{\epsilon}_t^x, \quad \boldsymbol{\epsilon}_t^x \sim \mathcal{N}(0, \sigma_x^2 \boldsymbol{I}) \tag{14}$$

$$\boldsymbol{y} = f^y(\boldsymbol{t}), \quad f^y \sim \mathcal{GP}^y(\mathbf{0}, \boldsymbol{K}^y(\boldsymbol{t}, \boldsymbol{t}')) \tag{15}$$

$$\boldsymbol{z}_t^y = f^y(\boldsymbol{t}) + \boldsymbol{\epsilon}_t^y, \quad \boldsymbol{\epsilon}_t^y \sim \mathcal{N}(\mathbf{0}, \sigma_y^2 \boldsymbol{I}) \tag{16}$$

where f^x and f^y are respectively the corresponding latent functions in x and y Cartesian coordinate, $\boldsymbol{t}$ is the input parameter. $\boldsymbol{z}_t$ denotes the measurement corresponding to input $\boldsymbol{t}$, $\boldsymbol{\epsilon}_t$ represents an the Gaussian observation noise, $\sigma_x^2 = \sigma_y^2$, $\boldsymbol{K}$ represents the covariance kernel function.

Considering the assumption (2), the squared exponential (SE) covariance kernel is applied in proposed tracker.

$$k(t, t') = \sigma_0^2 e^{\frac{-\|t-t'\|^2}{2l^2}} \tag{17}$$

where l denotes the scale parameter, σ_0^2 is the variance under the same input points.

Suppose that all hyperparameters $\boldsymbol{\theta} = [\sigma_0^2, l, \sigma_x^2, \sigma_y^2]$ are known. Consider the sparse GP method, the unknown function $\boldsymbol{f}$ is represented by N inducing Gaussian variables $\boldsymbol{u} = f(\boldsymbol{u_f})$ with an initial distribution $\boldsymbol{u} = \mathcal{N}(\mathbf{0}, \boldsymbol{C}_0)$. At new measurement, the inducing variables distribution is updated using the corresponding samples and the prior distribution. By applying Bayes law consecutively, the required posterior distribution $p(\boldsymbol{u}|\boldsymbol{z}_{1:N})$ can be obtained

$$p(\boldsymbol{u}|\boldsymbol{z}_{1:N}) \propto p(z_N|\boldsymbol{u}, \boldsymbol{z}_{1:N-1}) \cdot p(\boldsymbol{u}|\boldsymbol{z}_{1:N-1}) \tag{18}$$

Suppose that $\boldsymbol{u}$ will be the sufficient statistic of all the past measurement $\boldsymbol{z}_{1:N-1}$. The posterior distribution can be approximately in the following recursion.

$$p(\boldsymbol{u}|\boldsymbol{z}_{1:N}) \propto p(z_N|\boldsymbol{u}) \cdot p(\boldsymbol{u}|\boldsymbol{z}_{1:N-1}) \tag{19}$$

The measurement z_N and the function values $\boldsymbol{u}$ are jointly Gaussian

$$\begin{bmatrix} z_N \\ \boldsymbol{u} \end{bmatrix} \sim N\left(\begin{bmatrix} m(x_N) \\ m(\boldsymbol{u_f}) \end{bmatrix}, \begin{bmatrix} \boldsymbol{K}(x_N, x_N) + \sigma_0^2 & \boldsymbol{K}(x_N, \boldsymbol{u_f}) \\ \boldsymbol{K}(\boldsymbol{u_f}, x_N) & \boldsymbol{K}(\boldsymbol{u_f}, \boldsymbol{u_f}) \end{bmatrix} \right) \tag{20}$$

By marginalizing out $\boldsymbol{u}$, the likelihood function and the inducing variables prior are given by

$$p(z_N|\boldsymbol{u}) = \mathcal{N}(z_N; \boldsymbol{H}_N\boldsymbol{u}, \boldsymbol{R}_N) \tag{21}$$
$$p(\boldsymbol{u}) = \mathcal{N}(0, \boldsymbol{C}_0) \tag{22}$$

with

$$\boldsymbol{H}_N = \boldsymbol{K}(x_N, \boldsymbol{u_f}) \cdot \boldsymbol{K}(\boldsymbol{u_f}, \boldsymbol{u_f})^{-1}, \tag{23}$$
$$\boldsymbol{R}_N = \boldsymbol{K}(x_N, x_N) + \sigma^2 - \boldsymbol{K}(x_N, \boldsymbol{u_f}) \cdot \boldsymbol{K}(\boldsymbol{u_f}, \boldsymbol{u_f})^{-1}\boldsymbol{K}(\boldsymbol{u_f}, x_N) \tag{24}$$
$$\boldsymbol{C}_0 = \boldsymbol{K}(\boldsymbol{u_f}, \boldsymbol{u_f}). \tag{25}$$

The latent function $f(\cdot)$ of target motion is modeled by the sparse Gaussian process model in this letter. Therefor, the measurements from sensor has two functions, namely, updating the latent function modeled by *inducing points* and the estimating the target state.

3.1. **Learning the State-Space Motion Model.** In this block, learning of motion model at each time step requires to complete two stages. At each time step, the *inducing points* are updated first. Then, the motion model's hyper-parameter is updated using a learning method and the stochastic gradient descent technique. For all steps $N = 0$ to $N = N_{max}$, it assumed that the *inducing points* are fixed number and equidistant in location in this paper. Since $\boldsymbol{u}$ is assume to be a GP, the initial distribution $\boldsymbol{u}$ is Gaussian with mean 0 and covariance $K(\boldsymbol{u_f}, \boldsymbol{u_f})$, i.e.,$\boldsymbol{u} = \mathcal{N}(\boldsymbol{0}, \boldsymbol{K}(\boldsymbol{u_f}, \boldsymbol{u_f}))$. The goal is now to obtain the posterior distribution $\boldsymbol{u}$ with $\boldsymbol{z}_{1:N}$, recursively by updating the prior distribution of $\boldsymbol{u}$ from the previous step $N-1$.

According to the KF framework, the posterior distribution $\boldsymbol{u}$ is given by

$$\hat{\boldsymbol{u}}_N = \hat{\boldsymbol{u}}_{N-1} + \boldsymbol{G}_N \cdot (z_N - \tilde{f}(x_N)) \tag{26}$$
$$\hat{\boldsymbol{C}}_N = \hat{\boldsymbol{C}}_{N-1} - \boldsymbol{G}_N\boldsymbol{H}\hat{\boldsymbol{C}}_{N-1} \tag{27}$$

where

$$\boldsymbol{G}_N = \hat{\boldsymbol{C}}_{N-1}\boldsymbol{H}^T(\mathbb{D}\left[\tilde{f}(x_N)\right] + \sigma^2\boldsymbol{I})^{-1}, \tag{28}$$

$\tilde{\cdot}$ and $\hat{\cdot}$ denote the predicted and updated variables. Equation (26) and equation (27) denote the mean and covriance matrix of the state-space motion model.

For unknown hyper-parameters, online learning hyper-parameter is crucial in tracking system. we summarizes its online learning procedure proposed, which the goal to learn hyper-parameters $\boldsymbol{\theta}$ simultaneously with estimating the values of the latent function $f(\cdot)$ at the *inducing points*. By marginalizing out $\boldsymbol{u}$, one can obtain the log marginal function $\log p(\boldsymbol{z})$.

$$\begin{aligned} log\ p(\boldsymbol{z}_{1:N}) &= log \prod_{n=1}^{N} p(z_N|\boldsymbol{z}_{1:N-1}) \\ &= log\mathcal{N}(\boldsymbol{z}; \boldsymbol{H}_N\hat{\boldsymbol{u}}_{N-1}, \boldsymbol{H}_N\hat{\boldsymbol{C}}_{N-1}\boldsymbol{H}_N^T + \boldsymbol{R}_N - \sigma^2\boldsymbol{I}). \end{aligned} \tag{29}$$

According to equation (29), the log likelihood function is given by

$$\begin{aligned} \psi(\boldsymbol{\theta}) &= log\ p(\boldsymbol{z}_{1:N}) = \Sigma_{k=1}^{N}\mathcal{N}(\boldsymbol{r}_N^\theta; \boldsymbol{0}, \boldsymbol{S}_N^\theta) \\ &= -\frac{N}{2}log2\pi - \frac{1}{2}\Sigma_{n=1}^{N}log|\boldsymbol{S}_n| + \boldsymbol{r}_n^T\boldsymbol{S}_n^{-1}\boldsymbol{r}_n. \end{aligned} \tag{30}$$

where $\boldsymbol{r}_N^\theta = z_N - \boldsymbol{H}_N\hat{\boldsymbol{u}}_{N-1}$ and $S_N^\theta = \boldsymbol{H}_N\hat{\boldsymbol{C}}_{N-1}\boldsymbol{H}_N^T + \boldsymbol{R}_N - \sigma^2\boldsymbol{I}$.

In this paper, we utilize stochastic gradient descent to identify the maximizer $\boldsymbol{\theta} \in \Theta$ of the objective function $\psi(\boldsymbol{\theta})$ by employing the update method.

$$\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)} - \gamma^{(t-1)} \frac{\partial \psi_N}{\partial \boldsymbol{\theta}}_{|\boldsymbol{\theta}=\boldsymbol{\theta}^{(t-1)}} \tag{31}$$

where $\gamma^{(t-1)}$ denotes the search step. The derivative of $\psi(\theta)$, we have

$$\frac{\partial \psi(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -\frac{1}{2}\frac{\partial log|\boldsymbol{S}_N^{\theta}|}{\partial \boldsymbol{\theta}} - \frac{1}{2}\frac{\partial (\boldsymbol{r}_N^{\theta})(\boldsymbol{S}_N^{\theta})^{-1}(\boldsymbol{r}_N^{\theta})}{\partial \boldsymbol{\theta}} \tag{32}$$

The gradients of the mean and the covariance across time are recursively propagated using the chain method for derivatives of $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{C}}$, that is.

$$\frac{\partial \hat{\boldsymbol{u}}_N}{\partial \boldsymbol{\theta}} = \frac{\partial \hat{\boldsymbol{u}}_{N-1}}{\partial \boldsymbol{\theta}} + \frac{\partial \boldsymbol{G}_N}{\partial \boldsymbol{\theta}} \boldsymbol{r}_N + \boldsymbol{G}_N \frac{\partial \boldsymbol{r}_N}{\partial \boldsymbol{\theta}} \tag{33}$$

$$\begin{aligned} \frac{\partial \hat{\boldsymbol{C}}_N}{\partial \boldsymbol{\theta}} = \frac{\partial \hat{\boldsymbol{C}}_{N-1}}{\partial \boldsymbol{\theta}} - \frac{\partial \boldsymbol{G}_N}{\partial \boldsymbol{\theta}} \boldsymbol{S}_N \boldsymbol{G}_N^T - \boldsymbol{G}_N \frac{\partial \boldsymbol{S}_N}{\partial \boldsymbol{\theta}} \boldsymbol{G}_N \\ - \boldsymbol{G}_N \boldsymbol{S}_N \frac{\partial \boldsymbol{G}_N^T}{\partial \boldsymbol{\theta}} \end{aligned} \tag{34}$$

where$\frac{\partial \boldsymbol{G}_N}{\partial \boldsymbol{\theta}}$, $\frac{\partial \boldsymbol{r}_N}{\partial \boldsymbol{\theta}}$ and $\frac{\partial \boldsymbol{S}_N}{\partial \boldsymbol{\theta}}$ are computed recursively according to equation (26) to equation (28). Computing the derivatives of ψ as explained in equation (32) to equation (34), the stochastic gradient can be computed for single measurement.

3.2. **State Prediction and Estimation.** when the measurement z_N obtained at N, the inducing point are learned, then the target state at time N is updated by learning *inducing points*, and further predict the state at time N+1.

To incorporate the new measurements z_N at N, This requires determination of the posterior distribution of the state, given the *inducing points*. The density of this distribution have

$$p(\hat{f}(x_N)|\hat{\boldsymbol{u}}_N) = \mathcal{N}(\hat{f}(x_N); \mathbb{E}\left[\hat{f}(x_N)\right], \mathbb{D}\left[\tilde{f}(x_N)\right]) \tag{35}$$

with

$$\mathbb{E}(\hat{f}(x_N)) = \mathbb{E}(\tilde{f}(x_N)) + \hat{\boldsymbol{G}}_N(z_N - \mathbb{E}\left[\tilde{f}(x_N)\right]) \tag{36}$$

$$\mathbb{D}(\hat{f}(x_N)) = (\boldsymbol{I} - \hat{\boldsymbol{G}}_N)\mathbb{D}\left[\tilde{f}(x_N)\right] \tag{37}$$

where

$$\hat{\boldsymbol{G}} = \mathbb{D}\left[\tilde{f}(x_N)\right](\mathbb{D}\left[\tilde{f}(x_N)\right] + \sigma^2 \boldsymbol{I})^{-1} \tag{38}$$

For known prior distribution $p(\boldsymbol{u}|\boldsymbol{z}_{1:N})$, the predictive nonliear function $f(\cdot)$ at $\boldsymbol{u_f}$ is decribed as follows.

$$p(\tilde{f}(x_{N+1})|\hat{\boldsymbol{u}}_N) = \mathcal{N}(\tilde{f}(x_{N+1}); \mathbb{E}\left[\tilde{f}(x_{N+1})\right], \mathbb{D}\left[\tilde{f}(x_{N+1})\right]) \tag{39}$$

with

$$\mathbb{E}\left[\tilde{f}(x_{N+1})\right] = m(x_{N+1}) + \boldsymbol{H}_N(\hat{\boldsymbol{u}}_N - m(\boldsymbol{u_f})) \tag{40}$$

$$\begin{aligned} \mathbb{D}\left[\tilde{f}(x_{N+1})\right] = \boldsymbol{H}_N \hat{\boldsymbol{C}}_N \boldsymbol{H}_N^T + K(x_{N+1}, x_{N+1}) - \\ \boldsymbol{H}_N \boldsymbol{K}(\boldsymbol{u_f}, X_{N+1}) \end{aligned} \tag{41}$$

Fig 2. presents the schematic diagrams of the proposed tracker.

4. **Experiments.** We present the simulation results to demonstrate the performance of the sparse GP regression approach for TT. In this section, it are compared to existing tracker: KF, FGIMM, and Singer et.al.

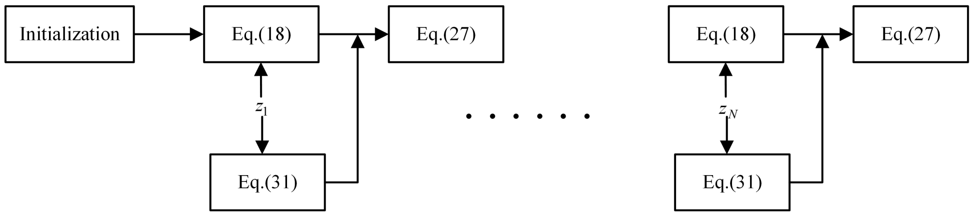


FIGURE 2. Schematic diagrams of the proposed tracker

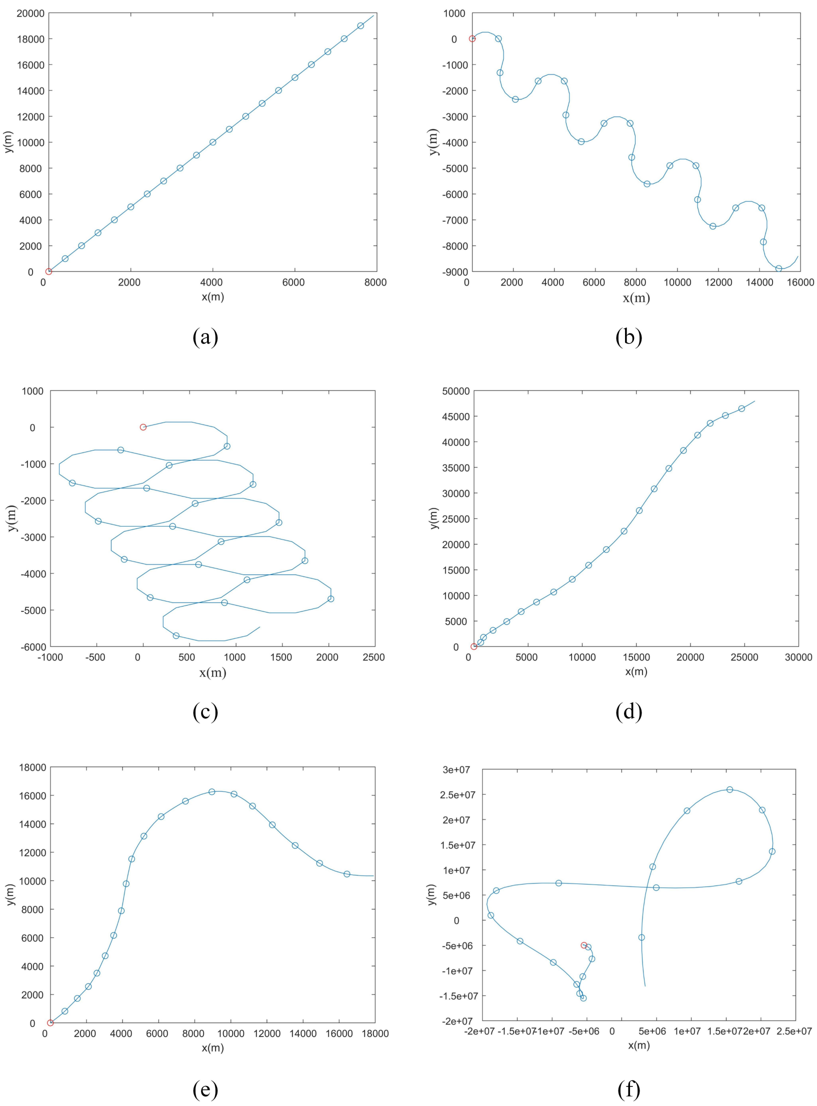


FIGURE 3. The figure shows a test trajectory of each scenario. The initial position is indicated by a red circle. (a) S1: CV. (b) S2: Gradual CT. (c) S3: Sharp CT. (d) S4: Singer lazy. (e) S5: Singer Agile. (f) S6: GP

4.1. **Compared Methods.** Three model-based trackers are compared with the suggested technique. The following is a detailed description of the three related trackers:

$\boldsymbol{KF}_{CV}$. A constant velocity (CV) model for a Kalman Filter (KF) with state transition. The process noise variance is set as $Q_{CV} = 10\ m^2/s^4$ $\boldsymbol{KF}_{CV}$. a constant velocity (CV) model for a Kalman Filter (KF) with state transition. The process noise variance is set as $Q_{CV} = 10\ m^2/s^4$.

Fixed grid interacting multiple model (FGIMM). An FGIMM [24] consists of three KFs, with a transition model that uses a CV model and two Coordinated Turn models. The rate of turns is set to $\{-15, 15\}^\circ/s$. The Markov transition probability for staying in the same mode is set to 0.9, and for changing the mode is 0.05. The initial model probability vector is $\{0.15, 0.7, 0.15\}$, and the process noise variance is set to 26 m^2/s^4 for each model. This process noise variance is considered optimal for a target moving at a speed of 200 m/s.

Singer. A Singer model was utilized to depict a KF with a state transition. The model's parameters were assigned the following values: $P_0 = 0.4$ for the probability of CV, $A_{max} = 8m/s^2$ for the maximum potential acceleration, the probability is $P_{max} = 0.1$, and $\tau_m = 8s$ for the time constant of the oeuvre.

4.2. **Test Scenarios.** The targets' trajectories in the six test scenarios below are produced by the aforementioned approaches.

S1): *Uniform motion*. Uniform motion is widely used as an ideal model, and As with the motion estimation of light, sound, and airplanes, the target velocity is constant. The situations match the filters in the CV and FGIMM.

S2): *Gradual Coordinated Turns*. The motion model for the left and right coordinated turns ($15^\circ/s$ for 10 s) is utilized to simulate the motion trajectory. The target alternates between turning left and right. First, it makes a 180° left turn and then follows it with a right turn.

S3): *Sharp Coordinated Turns*. Similar to S2, but with turn rates set at $30^\circ/s$ for 10 s. The scenarios is not match to any filters, and represents highly maneuverable targets dynamics.

S4): *Singer Lazy*. The state transition is the singer acceleration model. The parameter of this scenarios are same as those of the S3.

S5): *Singer Agile*. $A^2_{max} = 50m^2/s^4$ is the maximum acceleration, while the target motion is similar to S4. In this test scenario, an agile target is simulated compared d to S4.

S6): *GP*. The GP with zero mean and the SE covariance kernel, respectively, model the x and y coordinates of the motion trajectory. The GPs' hyperparameters are fixed constants. The length-scale parameter is $l = 10s$, and the variance magnitude is set to $\sigma^2 = 1e7m^2$.

The trajectory lasts for a total of 100 seconds, with each step lasting 1 second. The measurement noise covariance is $\sigma^2 = 25^2 m$. The performance of all approaches is evaluated using the root mean square errors (RMSE) of the target's state. Figure 3 illustrates an example trajectory for each scenario.

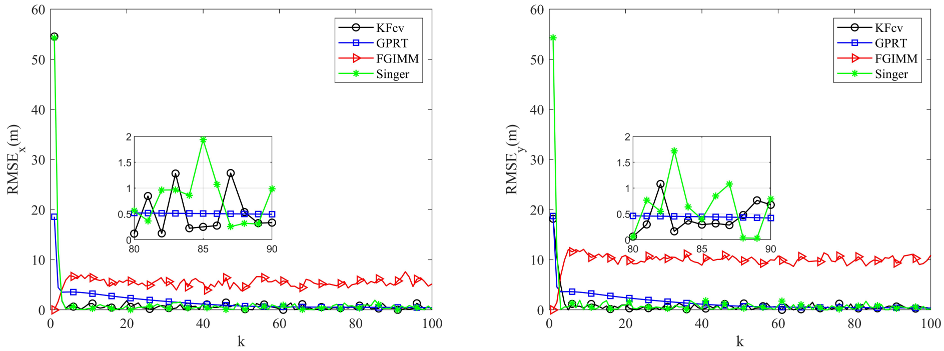
$$RMSE = sqrt\frac{1}{N_{mc}}\sum_{i=1}^{N_{mc}}(\hat{f}(\boldsymbol{x}) - f(\boldsymbol{x}))^2 \tag{42}$$

where, N_{mc} represents the Monte Carlo number, $\hat{f}(\boldsymbol{x})$is the tracker output, $f(\boldsymbol{x})$ is the true value.

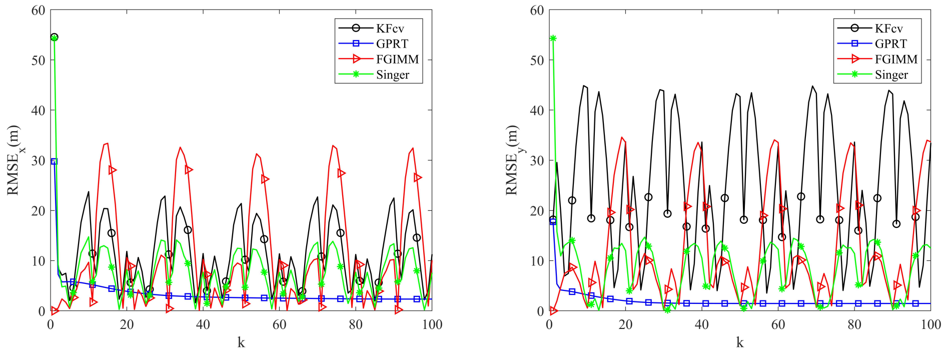
4.3. **Experimental Results.** We perform simulation experiments on the proposed tracker in six test scenarios in this section. Monte Carlo methods, with 10,000 runs, are used to simulation the performance of all tracker in each test scenario. The result, graphical comparison of the estimation procedure are obtained. The result of the procedure are illustrated in figure 4.

Figure 4(a) demonstrates that in the S1 constant velocity motion scene, the KF_{CV} method outperforms the other three methods. This is because KF_{CV} is the optimal state estimation algorithm in linear systems. The Singer method performs the worst in S1 scenes. For figure 4(b) and figure 4(c), the proposed method, GPMT, exhibits significant robustness and outperforms the predefined model in test scenarios. For methods based on KF, improvements in estimation performance can be observed. The improvement in the estimate process based on preset model approaches owe to the KF, when compared to their respective prediction procedures.

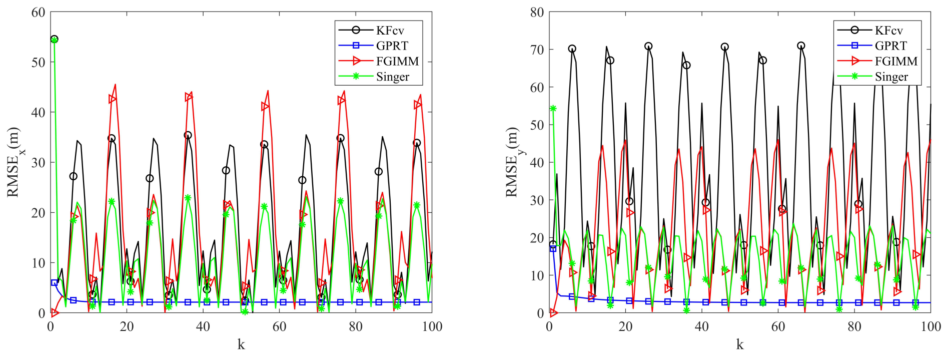
In the two CT scenarios of S2 and S3, The proposed trackers showed the most stable performance and RMSE values remained below 10. The RMSE values of the other three schemes exhibit "periodicity", which is related to the motion attributes of the simulation target on the one hand, and to the predefined motion models, namely KF_{CV}, FGIMM, and Singer, which have insufficient response to turns. The poor performance of FGIMM in the S4 scenario can be attributed to the mismatching between the predefined motion and the actual scene, leading to a loss in performance. On the one hand, the trajectory of S4 is generated based on the Singer model and exhibits a certain degree of linearity, so KF_{CV} and Singer trackers achieve optimal performance in this scenario. The proposed trackers did not achieve optimal performance in this scenario. Compared to the other four motion scenarios, due to the high randomness of the S5 scenario, all trackers obtained higher RMSE values. However, the RMSE value of The proposed trackers gradually decreased, and the lowest RMSE value was obtained at the end of the iteration.


KFcv
GPRT
FGIMM
Singer
RMSE$_x$ (m)
RMSE$_y$ (m)
k

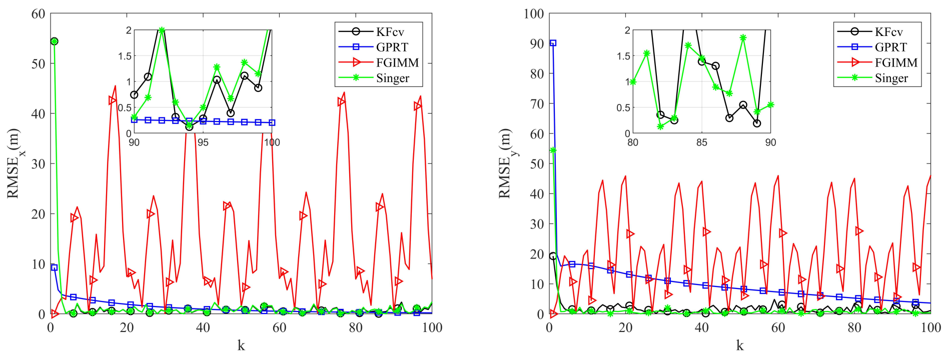
(a)


KFcv
GPRT
FGIMM
Singer
RMSE$_x$ (m)
RMSE$_y$ (m)
k

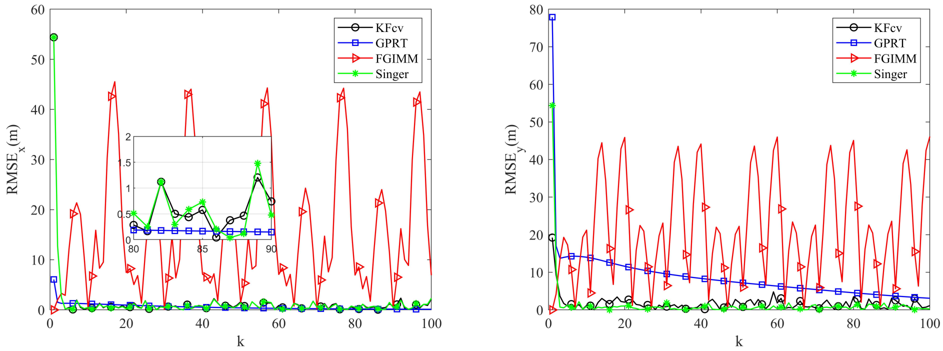
(b)


KFcv
GPRT
FGIMM
Singer
RMSE$_x$ (m)
RMSE$_y$ (m)
k

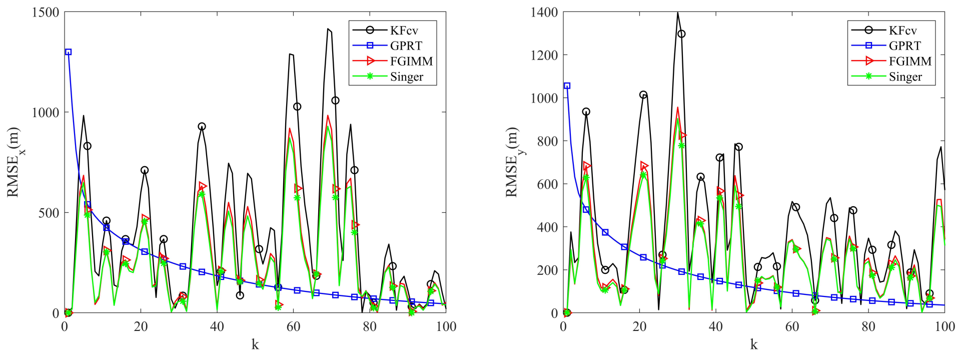
(c)


KFcv
GPRT
FGIMM
Singer
RMSE$_x$ (m)
RMSE$_y$ (m)
k

(d)



(e)



(f)

FIGURE 4. RMSE. The figure shows the estimation performance in 10, 000 Monte Carlo runs based on RMSE. (a) S1: CV. (b) S2: Gradual CT. (c) S3: Sharp CT. (d) S4: Singer lazy. (e) S5: Singer Agile. (f) S6: GP.

5. **Conclusion.** This paper proposes a model-free technique for filtering and predicting single target trajectories in cases of mixed and uncertain motion. The approach involves building a recursive and online learning Gaussian process regression tracker based on sparse Gaussian process regression and stochastic gradient descent. This tracker is capable of providing estimates of the target positions in the present time index as well as predictions for the next time. In challenging environments, we have demonstrated the superiority and reliability of the proposed tracker by comparing it to alternative online model-based trackers. The experimental results show that the proposed method gives more accurate than the model-based result. The performance of the propose tracker in addressing multi-target tracking issues and the data association between the target and sensor will be the main topics of future research.

REFERENCES

[1] M. Sun, M.-E. Davies, I. Proudler, "A Gaussian process based method for multiple model tracking," In Proceeding of the Sensor Signal Processing for Defence Conference (SSPD), South Hall, Pollock Halls, Edinburgh, September 15 - September 16, 2020.

[2] J. Zhao, W.-P. Chen, R.-B. Xiao, J. Ye, "FA with division of roles for complex optimal scheduling," *Frontiers of Information Technology & Electronic Engineering*, vol. 22, no. 10, pp.1311-1333, 2021.

[3] B. Han, H. Huang,; L. Lei, "An improved IMM algorithm based on STSRCKF for maneuvering target tracking", *IEEE Access*, vol. 7, pp. 57795-57804, 2019.

[4] J. Zhao, J. Tang, A. Shi, "Improved Density Peaks clustering based on Firefly Algorithm", *International Journal of Bio-Inspired Computation*, vol. 15, no.1, pp. 24-42, 2020.

[5] L. Mihaylova, A.-Y. Carmi, F. Septier, "Overview of Bayesian sequential Monte Carlo methods for group and extended object tracking", *Digital Signal Process*, vol. 25, pp. 1-16, 2014.

[6] J.-W. Koch, "Bayesian approach to extended object and cluster tracking using random matrices", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, pp. 1042-1059, 2008.

[7] Y. Bar-Shalom, P.-K. Willett,X. Tian, *Tracking and data fusion.* Storrs, CT, USA, YBS publishing, 2011.

[8] R.-E. Kalman, "A new approach to linear filtering and prediction problems", *Journal of Basic Engineering*, vol. 82, pp. 35-45, 1960.

[9] H.-W. Sorenson, *Kalman Filtering: Theory and Application.* Piscataway, NJ, USA: IEEE Press, 1985.

[10] A. WanE, R.-V.-D. Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceeding of the Adaptive Systems for Signal Processing, Communications, and Control Symposium.* IEEE, 2000, pp. 153-158.

[11] H.-A. Blom, Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Transactions on Automatic Control*, vol. 33, pp. 780-783, 1988.

[12] J. Zhao, G. Wang, J. Pan, "Density peaks clustering algorithm based on fuzzy and weighted shared neighbor for uneven density datasets," *Pattern Recognition*, vol. 139, 109406, 2023.

[13] J. Yang, L. Yang, Z. Xiao, "Adaptive parameter particle CBMeMBer tracker for multiple maneuvering target tracking," *EURASIP Journal on Advances in Signal Processing*, vol. 1, pp. 1-11, 2016.

[14] W. Aftab, L. Mihaylova, "A Gaussian process regression approach for point target tracking", in *Proceedings of the 22th International Conference on Information Fusion (FUSION)*, IEEE, 2019, pp. 1-9.

[15] W. Aftab, L. Mihaylova, "A learning Gaussian process approach for maneuvering target tracking and smoothing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, pp. 278-292, 2020.

[16] T. Chen, H. Ohlsson, L. Ljung, "On the estimation of transfer functions, regularizations and Gaussian processes—Revisited," *Automacation*, vol. 8, pp. 1525-1535, 2012.

[17] C. He, P. Kang P, Q. Li, "Firefly Algorithm with Combination of Multi-strategiey," *Journal of Nanchang Institute of Technology*, vol. 91, pp. 37-47, 2019.

[18] J. Zhao, Z.-F. Xie, L. Lv, H. Wang, "Firefly Algorithm with Deep Learning,". *Acta Electronica Sinica*, vol. 46, no. 11, pp. 2633-2641, 2018.

[19] M. Schürch, D. Azzimonti, A. Benavoli, "Recursive estimation for sparse Gaussian process regression," *Automacation*, vol. 120, 109127, 2020.

[20] J. Zhao, D. Chen, R.-B Xiao, Z.-H. Cui, "Multi-strategy ensemble firefly algorithm with equilibrium of convergence and diversity," *Applied Soft Computing*, vol. 123, 108938, 2022.

[21] T.-D. Bui, J. Yan, R.-E. Turner, "A Unifying Framework for Gaussian Process Pseudo-Point Approximations using Power Expectation Propagation". *Journal of Machine Learing Research*, vol. 18, pp. 1-72, 2017.

[22] L. Csató, M. Opper, "Sparse online Gaussian processes", *Neural Computing*, vol. 14, pp.641-668, 2002.

[23] X.-R. Li, V.-P. Jilkov, "Survey of maneuvering target tracking. Part V. Multiple-model methods", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, pp. 1255–1321, 2005.

[24] L. Lv, J. Zhao, J.-J. Wang, T.-H. Fan, "Multi-objective Firefly Algorithm Based on Compensation Factor and Elite Learning", *Future Generation Computer Systems*, vol. 91, pp. 37-47, 2019.