

A Gaussian Process based Method for Multiple Model Tracking

Mengwei Sun, Mike E. Davies, Ian Proudler, James R. Hopgood

Abstract—Manoeuvring target tracking faces the challenge caused by the target motion model uncertainty, i.e., unknown model types or uncertain model parameters. Multiple-model (MM) methods have been generally considered to deal with this challenge, in which a bank of elemental filters is run simultaneously to provide a joint decision and estimation of motion model and localisation. However, if the uncertainty of the target trajectory increases, such as the target moves under mixed manoeuvring behaviours with time-varying parameters, more filters with different model assumptions have to be taken into account to match the motion of the target, which may lead to prohibitive computational complexity. To address this problem, we establish a training based algorithm which can learn the actual motion model as a Gaussian process (GP) regression. Then, by integrating the trained GP into the particle filter (PF), a GP-PF based tracking method is developed to track the manoeuvring targets in real-time. Monte Carlo experiments show that the proposed method had much lower tracking root mean square error (RMSE) and robustness compared with the most commonly used MM methods.

Index Terms—Gaussian process, manoeuvring target tracking, mixing manoeuvres, particle filtering

I. INTRODUCTION

TARGET tracking is a fundamental task in sensor-based applications, such as radar, sonar, and navigation [1]. Numerous mathematical models have been developed to approximate the motion trajectory of the target [2], such as the commonly used non-manoevring model, i.e., constant velocity (CV) model and highly manoeuvring model, i.e., coordinated turn (CT) model. The application of Bayesian filters for target tracking is based on the actual state-space model of objects, therefore being conditional on a motion model. However, in reality, there may be significant motion-model uncertainty when targets undergo unknown or mixed manoeuvring behaviours [3][4]: i.e., the evolution of the target state is too complex to be approximated as one specific mathematical model. Uncertainty can also be caused due to various parameters not being known a priori, or if they change with time. Examples of such parameters include: the turn rate when CT model is considered [5], or the process noise level. If incorrect models or parameters are applied,

the tracking performance of traditional Bayesian filters would degrade and even become unacceptable.

To deal with this problem, multiple-model (MM) methods are commonly used [3]. The basic idea of MM methods is to assume a set of motion models as possible candidates for the manoeuvring target, and then use a bank of elemental filters with these different models to capture the mixed motion behaviours of the target, and generate the overall estimation based on the results achieved by each elemental filter [3]. MM based methods can achieve satisfactory tracking performance when the uncertainty is low. However, in some cases, such as when the rotation is a time-varying or doubly-stochastic process, a sufficiently large set of models with different parameters is required to cover the range of possible motion models, which leads to a high computation complexity [4]. To address this aspect, [4] and [5] incorporated adaptive parameter estimation methods into the Bayesian filter framework, where the unknown parameter is estimated by approximating its distribution with particles and corresponding weights. However, as parametric models are not always able to capture all aspects of the motion behaviours, those methods might fail if the chosen motion model set is incapable of modelling the ground truth trajectory.

As an alternative to traditional Bayesian methods, machine learning based tracking algorithms have been proposed recently [6]-[8]. In [6], a quadruplet convolutional neural network (CNN) based algorithm was designed for multi-target tracking. A long short-term memory (LSTM) neural network was used in [7] to perform the prediction step of single-target localisation, which can improve the tracking accuracy and facilitate the use of computationally efficient low-dimensional state spaces. Wahlstrom [8] proposed an extended target tracking method using a Gaussian process (GP) to estimate the shape of the object whose moving trajectory followed a linear CV model. These methods achieved better performance and higher flexibility, but are limited to a fixed but unknown motion behaviour and did not take into consideration the case of mixed motion behaviours with uncertain parameters.

To overcome the limitations of existing methods regarding to mixed motion behaviours, in this paper, Gaussian process regression [9]-[11] is introduced to learn the motion model of the moving target. The key advantage of GP is the flexibility for modelling the uncertain systems, and the ability to learn noise and parameters from training data. The contributions of this paper can be summarised into two aspects. First, we develop a GP-PF based tracking method, showing how

M. W. Sun, M. E. Davies and J. R. Hopgood are with Institute of Digital Communications, University of Edinburgh, Edinburgh, EH9 3FG, U.K. E-mail: (msun; James.Hopgood; Mike.Davies)@ed.ac.uk.

I. Proudler is with the Centre for Signal Image Processing (CeSIP), Department of Electronic Electrical Engineering, University of Strathclyde, Glasgow, G1 1XW, U.K. E-mail: ian.proudler@strath.ac.uk.

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/S000631/1; and the MOD University Defence Research Collaboration (UDRC) in Signal Processing.

to integrate the trained GP prediction motion model into a particle filter (PF) tracking framework. Second, the GP-PF based method is compared to interacting MM (IMM) methods with different number of elemental filters. The simulation results show the improved tracking performance and higher robustness of the proposed method.

The remainder of this paper is organised as follows. The description of the formulated problem and GP regression are presented in Section II. The proposed algorithm is developed in Section III. Simulation results and performance comparisons are presented in Section IV. Finally, conclusions are provided in Section V.

II. BACKGROUND

A. Problem Formulation

For simplicity and clarity, a 2-D horizontal model is considered but it is easily extended to 3-D. The discrete time state-space model for tracking a single moving target is described by:

$$\mathbf{x}_{t+1} = f_t(\mathbf{x}_t) + \mathbf{v}_t, \quad (1)$$

$$\mathbf{z}_t = h(\mathbf{x}_t) + \mathbf{e}_t. \quad (2)$$

Here, $\mathbf{x}_t = [\xi_t, \dot{\xi}_t, \eta_t, \dot{\eta}_t]^T$ denotes the target state at time t in a 2-D plane, and (ξ_t, η_t) and $(\dot{\xi}_t, \dot{\eta}_t)$ represent the target position and the corresponding velocity in X-axis and Y-axis, respectively. The velocity is a vector with its magnitude as the speed. \mathbf{z}_t denotes the measurement detected by the sensor. The state motion functions defined as $f_t(\mathbf{x}_t) : \mathbb{R}^4 \rightarrow \mathbb{R}^4$ dynamically follow different motion models, and two typical motion models are considered in this paper, the non-maneuvring CV model, and the highly manoeuvring CT model with unknown turn rate. Note, the motion models assumed are not a given which means that more sophisticated state space models can also be used equally. The measurement function $h(\mathbf{x}_t) : \mathbb{R}^4 \rightarrow \mathbb{R}^2$ is defined as:

$$\mathbf{z}_t = H\mathbf{x}_t + \mathbf{e}_t, \quad (3)$$

where,

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (4)$$

$\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, Q)$ and $\mathbf{e}_t \sim \mathcal{N}(\mathbf{0}, R)$ denote the process noise and the measurement noise, respectively. The matrix $Q = qI_{4 \times 4}$ and $R = rI_{2 \times 2}$, where q and r are the corresponding noise variance and are assumed to be static in this paper.

Based on Bayes' rule, the posterior probability density function (PDF) of the target state vector conditioned on the measurements is,

$$p(\mathbf{x}_{1:t} | \mathbf{z}_{1:t}) \propto p(\mathbf{z}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{x}_{1:t-1}) p(\mathbf{x}_{1:t-1} | \mathbf{z}_{1:t-1}), \quad (5)$$

where $p(\mathbf{z}_t | \mathbf{x}_t) = \mathcal{N}(\mathbf{z}_t | h(\mathbf{x}_t), R)$ describes the likelihood of the measurement given the state. The quantity $p(\mathbf{x}_t | \mathbf{x}_{1:t-1})$ is the prediction probability which can be calculated based on the real-time motion model $f_{t-1}(\mathbf{x}_{t-1})$, i.e., $p(\mathbf{x}_t | \mathbf{x}_{1:t-1}) = \mathcal{N}(\mathbf{x}_t | f_{t-1}(\mathbf{x}_{t-1}), Q)$. Typically, the motion model is a

parametric description for the underlying evolution process. While in practice, motions of the target may not follow parametric models. Therefore, GP is used to identify the motion models by placing GP priors on the unknown functions and will be introduced in the following.

B. Gaussian Process Regression

GP regression is a popular means of non-parametric nonlinear modelling that can directly capture model uncertainty. It attempts to model a function $g(\mathbf{x})$ by providing both the mean function $m(\mathbf{x})$ and covariance/kernel function $k(\mathbf{x}, \mathbf{x}')$ [11]:

$$g(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \quad (6)$$

$$m(\mathbf{x}) = \mathbb{E}[g(\mathbf{x})], \quad (7)$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(g(\mathbf{x}) - m(\mathbf{x}))(g(\mathbf{x}') - m(\mathbf{x}'))]. \quad (8)$$

Let us consider a general GP regression problem with noisy observations from an unknown function described as:

$$y = g(\mathbf{x}) + \mathbf{v}, \quad \mathbf{v} \sim \mathcal{N}(0, \sigma_v^2 I). \quad (9)$$

The training inputs are denoted as $X = [\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{N-1}]$ and the outputs are $\mathbf{y} = [y_0, y_1, \dots, y_{N-1}]^T$. The purpose of GP is to derive the latent distribution of the function $\mathbf{g}^* = [g_0^*, \dots, g_{T-1}^*]^T$ at the test inputs $X^* = [\mathbf{x}_0^*, \dots, \mathbf{x}_{T-1}^*]^T$, conditioned on the training data set $\mathcal{D} = \{X, \mathbf{y}\}$. The joint distribution of the training measurement values \mathbf{y} and the test function value at one test point, i.e., $g_t^* = g(\mathbf{x}_t^*)$, is given as,

$$\begin{bmatrix} \mathbf{y} \\ g_t^* \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \mathbf{m}(X) \\ m(\mathbf{x}_t^*) \end{bmatrix}, \begin{bmatrix} K(X, X) + \sigma_v^2 I & \mathbf{k}(X, \mathbf{x}_t^*) \\ \mathbf{k}(\mathbf{x}_t^*, X) & k(\mathbf{x}_t^*, \mathbf{x}_t^*) \end{bmatrix} \right). \quad (10)$$

where $\mathbf{k}(X, \mathbf{x}_t^*) = [k(\mathbf{x}_0, \mathbf{x}_t^*), \dots, k(\mathbf{x}_{N-1}, \mathbf{x}_t^*)]^T$, $k(\mathbf{x}_t^*, \mathbf{x}_t^*)$ is the covariance of $g(\mathbf{x}_t^*)$, and $K(X, X)$ denotes the covariance matrix for the training input data:

$$K(X, X) = \begin{bmatrix} k(\mathbf{x}_0, \mathbf{x}_0), & \dots, & k(\mathbf{x}_0, \mathbf{x}_{N-1}) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_{N-1}, \mathbf{x}_0), & \dots, & k(\mathbf{x}_{N-1}, \mathbf{x}_{N-1}) \end{bmatrix}. \quad (11)$$

From (10), the conditional distribution of the test function is derived as,

$$(g_t^* | \mathcal{D}) \sim \mathcal{N}(\bar{g}_t^*, \mathbb{V}[g_t^*]), \quad (12)$$

$$\bar{g}_t^* = m(\mathbf{x}_t^*) + \mathbf{k}_t^{*T} [K + \sigma_v^2 I]^{-1} (\mathbf{y} - \mathbf{m}(X)), \quad (13)$$

$$\mathbb{V}[g_t^*] = k(\mathbf{x}_t^*, \mathbf{x}_t^*) - \mathbf{k}_t^{*T} [K + \sigma_v^2 I]^{-1} \mathbf{k}_t^*. \quad (14)$$

III. GP-PF BASED TRACKING ALGORITHM

In this section a tracking algorithm is proposed where the uncertainty and mixing behaviour of the motion model is first learned by non-parametric stationary and GP regression off-line, i.e., the training process, and then the learned model is integrated into the PF framework to estimate the target states on-line, i.e., the test process. The notations used in this section are listed in Table I. The motion model $f(\mathbf{x})$ is

TABLE I: The meaning of notations used in Section III.

Notation	Meaning
$\mathbf{x}_{0:N}$	Target state of training data sequence
$\mathbf{z}_{0:N}$	Observation of training data sequence
$\mathbf{x}_{0:T}^*$	Target state of test data sequence
$\mathbf{z}_{0:T}^*$	Observation of test data sequence
(ξ_t, η_t)	Position of target at 2-D plane
$(\dot{\xi}_t, \dot{\eta}_t)$	Velocity of target at 2-D plane
Q	Covariance matrix of process noise
R	Covariance matrix of measurement noise
\mathcal{D}	Training data set for GP
\mathcal{X}	State space
\mathcal{Z}	Observation space
$\mathbf{f}_{0:N}(\mathbf{x}_{0:N})$	Motion function of training process
$\mathbf{f}_{0:T}^*(\mathbf{x}_{0:T}^*)$	Motion function of test process
$h(\mathbf{x}_t)$	Measurement function

learned using the non-parametric GP, while the measurement function $h(\mathbf{x})$ is deterministic by different sensor type with no correlation over time and is given in arbitrary parametric form in (3). Hence, the state-space model shown in (1)-(2) can be graphically illustrated in Fig. 1. Our goal is to integrate the learned GP regression into the PF framework to estimate the target states. Specifically, the motion function with mixed behaviours is approximated by a prediction Gaussian process, which is then applied into the PF framework for drawing particles. Next, the correction step is performed by updating the weights of particles and achieving the estimations of target states based on the maximum a posterior probability (MAP) criterion.

To train the state transition function, we take training data set $\mathcal{D} := \{X_I, \mathbf{x}_O\} \subset \mathcal{X} \times \mathcal{X}$ as the input and output respectively, where $X_I = [\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{N-1}]$ and $\mathbf{x}_O = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]^T$, and $\mathbf{x}_n = [\xi_n, \dot{\xi}_n, \eta_n, \dot{\eta}_n]^T$. The test sequence are represented as $\{\mathbf{x}_{0:T}^*, \mathbf{z}_{0:T}^*\} \subset \mathcal{X} \times \mathcal{Z}$, where $\mathbf{z}_{0:T}^* := (z_0^*, \dots, z_T^*)$ is the sequence of test observations while the target states $\mathbf{x}_{0:T}^* := (\mathbf{x}_0^*, \dots, \mathbf{x}_T^*)$ are unknown and need to be estimated sequentially.

A. Learning the Motion Model based on GP Regression

The joint conditional distribution of the states for the test process can be partitioned as [11],

$$\begin{aligned} p(\mathbf{x}_{0:T}^* | \mathcal{D}) &= \int p(\mathbf{x}_{0:T}^*, \mathbf{f}_{0:T}^* | \mathcal{D}) d\mathbf{f}_{0:T}^*, \\ &= \int p(\mathbf{x}_{0:T}^* | \mathbf{f}_{0:T}^*, \mathcal{D}) p(\mathbf{f}_{0:T}^* | \mathcal{D}) d\mathbf{f}_{0:T}^*. \end{aligned} \quad (15)$$

Based on the GP regression (12), the distribution of motion function can be written as,

$$p(\mathbf{f}_{0:T}^* | \mathcal{D}) = \mathcal{N}(\mathbf{f}_{0:T}^* | \bar{\mathbf{f}}^*(\mathbf{x}_{0:T}^*), \mathbb{V}_{\mathbf{f}^*}(\mathbf{x}_{0:T}^*)), \quad (16)$$

where $\bar{\mathbf{f}}^*(\mathbf{x}_{0:T}^*) = [\bar{\mathbf{f}}_0^*, \dots, \bar{\mathbf{f}}_T^*]$ and $\mathbb{V}_{\mathbf{f}^*}(\mathbf{x}_{0:T}^*) = [\mathbb{V}_{\mathbf{f}_0^*}, \dots, \mathbb{V}_{\mathbf{f}_T^*}]$, where $\bar{\mathbf{f}}_t^*$ and $\mathbb{V}_{\mathbf{f}_t^*}$ can be calculated by (13) and (14) respectively. We have that the test input \mathbf{x}_t^* is independent from the training data set X but relates to the

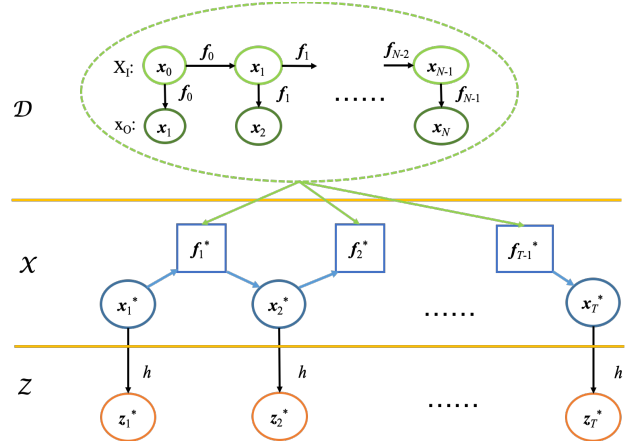


Fig. 1: State-space models: GP-based state transition and parametric measurement functions. Circle represents the variable node and square is the non-parametric function node. \mathcal{D} , \mathcal{X} and \mathcal{Z} represent the training data space, test state space and test observation space respectively.

motion function \mathbf{f}_{t-1}^* and the state \mathbf{x}_{t-1}^* in the previous time slot, thus yielding,

$$\begin{aligned} p(\mathbf{x}_{0:T}^* | \mathbf{f}_{0:T}^*, \mathcal{D}) &= p(\mathbf{x}_{0:T}^* | \mathbf{f}_{0:T}^*), \\ &= p(\mathbf{x}_0^*) \prod_{t=1}^T p(\mathbf{x}_t^* | \mathbf{f}_{t-1}^*, \mathbf{x}_{t-1}^*), \\ &= p(\mathbf{x}_0^*) \prod_{t=1}^T \mathcal{N}(\mathbf{x}_t^* | \mathbf{f}_{t-1}^*, \mathbf{Q}). \end{aligned} \quad (17)$$

For the tracking problem, the Bayesian filter is implemented by realising (15) in an iterative way as shown in Fig. 1.

$$p(\mathbf{x}_t^* | \mathcal{D}) = \int p(\mathbf{x}_t^* | \mathbf{f}_{t-1}^*) p(\mathbf{f}_{t-1}^* | \mathcal{D}) d\mathbf{f}_{t-1}^*. \quad (18)$$

As we mainly focus on the four variables of the target states in this paper, i.e., $\mathbf{x}_t^* = [\xi_t^*, \dot{\xi}_t^*, \eta_t^*, \dot{\eta}_t^*]^T$, the motion function \mathbf{f}_{t-1}^* is decomposed into $\mathbf{f}_{t-1}^* = [f_{\xi,t-1}^*, f_{\dot{\xi},t-1}^*, f_{\eta,t-1}^*, f_{\dot{\eta},t-1}^*]$ and is learned and modelled as four Gaussian processes jointly, as shown in Fig. 2. Therefore, the prediction distribution in (18) can be decomposed as:

$$\begin{aligned} p(\mathbf{x}_t^* | \mathcal{D}) &= \prod_{s_t = \{\xi_t^*, \dot{\xi}_t^*, \eta_t^*, \dot{\eta}_t^*\}} \int p(s_t | f_{s,t-1}^*) p(f_{s,t-1}^* | \mathcal{D}) df_{s,t-1}^*. \end{aligned} \quad (19)$$

Here,

$$p(s_t | f_{s,t-1}^*) = \mathcal{N}(s_t | f_{s,t-1}^*, q), \quad (20)$$

$$\begin{aligned} p(f_{s,t-1}^* | \mathcal{D}) &= \mathcal{N}(f_{s,t-1}^* | \bar{f}_{s,t-1}^*(\mathbf{x}_{t-1}^*), \mathbb{V}_{f_{s,t-1}^*}(\mathbf{x}_{t-1}^*)), \\ &\sim \mathcal{N}(\bar{f}_{s,t-1}^*, \mathbb{V}[f_{s,t-1}^*]), \end{aligned} \quad (21)$$

where the mean and covariance function can be calculated as (13)-(14). By substituting (20) and (21) into (19), we obtain

$$\begin{aligned} p(\mathbf{x}_t^* | \mathcal{D}) &= \prod_{s_t = \{\xi_t^*, \dot{\xi}_t^*, \eta_t^*, \dot{\eta}_t^*\}} \mathcal{N}(s_t | \bar{f}_{s,t-1}^*, \mathbb{V}[f_{s,t-1}^*] + q). \end{aligned} \quad (22)$$

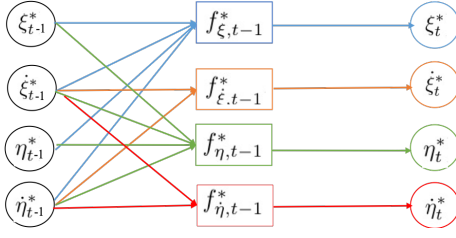


Fig. 2: Joint GPs for state prediction.

B. Data Test based on Particle Filter

In order to estimate the states for the test data iteratively, a PF is used to estimate the posterior distribution over the state $p(\mathbf{x}_t^* | \mathbf{z}_{1:T}^*, \mathcal{D})$ for each test time $t = 1, \dots, T$ [12] [13]. Based on the PF, the posterior probability can be approximated by the particles with corresponding weights as,

$$\begin{aligned} p(\mathbf{x}_t^* | \mathbf{z}_t^*, \mathcal{D}) &= \frac{p(\mathbf{z}_t^* | \mathbf{x}_t^*) p(\mathbf{x}_t^* | \mathcal{D})}{\int p(\mathbf{z}_t^* | \mathbf{x}_t^*) p(\mathbf{x}_t^* | \mathcal{D}) d\mathbf{x}_t^*} \\ &\approx \sum_{m=1}^M w_{m,t} \delta(\mathbf{x}_{m,t}^\dagger - \mathbf{x}_t^*), \\ &\approx \sum_{m=1}^M w_{m,t} \delta(\xi_{m,t}^\dagger - \xi_t^*) \delta(\dot{\xi}_{m,t}^\dagger - \dot{\xi}_t^*) \\ &\quad \times \delta(\eta_{m,t}^\dagger - \eta_t^*) \delta(\dot{\eta}_{m,t}^\dagger - \dot{\eta}_t^*). \end{aligned} \quad (23)$$

Here, m is the index of the particles, and $m = 1, \dots, M$. $\mathbf{x}_{m,t}^\dagger = [\xi_{m,t}^\dagger, \dot{\xi}_{m,t}^\dagger, \eta_{m,t}^\dagger, \dot{\eta}_{m,t}^\dagger]^T$ represents the particles and $w_{m,t}$ is the corresponding weight. The particles are drawn from the importance density which is chosen to be the prior probability [14] derived in (22), and weights are then calculated based on the likelihood function and observed measurement as (28). The pseudo-code of the proposed GP-PF based filter for tracking is illustrated in Algorithm 1.

IV. SIMULATION RESULTS

We first evaluate the derived distribution of the motion function $\mathbf{f}_{1:T}^*$ based on GP with a synthetic data-set. Then, the tracking performance is evaluated and compared with different existing tracking methods. The data for both training and test processes are synthetic, i.e., the generated target states evolve following motion models which switch between the CV or CT models as specified by the transition probability matrix:

$$\Gamma = \begin{bmatrix} p_{CV \rightarrow CV} & p_{CV \rightarrow CT} \\ p_{CT \rightarrow CV} & p_{CT \rightarrow CT} \end{bmatrix} = \begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix}. \quad (30)$$

For all experiments, the following parameters are used: the model parameters are updated every 10 samples; the rotation rate of the CT model is distributed as $\omega_t \sim \mathcal{N}(0.1, 0.4)$; the process noise variance of both CT and CV models is $q = 1$. The length of the training data set is $N = 2000$.

A. Synthetic Data Set Evaluations

Before we evaluate the tracking performance of the proposed method, it is worth to verify the prediction

Algorithm 1

1) *Initialisation*: Draw M samples $\mathbf{x}_{m,0}^\dagger \sim p(\mathbf{x}_0)$, and the initial weights are set as $w_{m,0} = 1/M$, $m = 1, \dots, M$.

2) For $t = 1, \dots, T$, do:

For $m = 1, \dots, M$:

2.1) Draw particles $\mathbf{x}_{m,t}^\dagger = [\xi_{m,t}^\dagger, \dot{\xi}_{m,t}^\dagger, \eta_{m,t}^\dagger, \dot{\eta}_{m,t}^\dagger]^T$ using the distribution in (22),

$$\xi_{m,t}^\dagger \sim \mathcal{N}(\bar{f}_{\xi,t-1}^*, \mathbb{V}[f_{\xi,t-1}^*] + q), \quad (24)$$

$$\dot{\xi}_{m,t}^\dagger \sim \mathcal{N}(\bar{f}_{\dot{\xi},t-1}^*, \mathbb{V}[f_{\dot{\xi},t-1}^*] + q), \quad (25)$$

$$\eta_{m,t}^\dagger \sim \mathcal{N}(\bar{f}_{\eta,t-1}^*, \mathbb{V}[f_{\eta,t-1}^*] + q), \quad (26)$$

$$\dot{\eta}_{m,t}^\dagger \sim \mathcal{N}(\bar{f}_{\dot{\eta},t-1}^*, \mathbb{V}[f_{\dot{\eta},t-1}^*] + q), \quad (27)$$

2.2) Update the weight according to [14] as,

$$w_{m,t} \propto w_{m,t-1} \times \mathcal{N}(\mathbf{z}_t | h(\mathbf{x}_{m,t}^\dagger), R), \quad (28)$$

End For

2.3) The weight is then normalised and MAP estimation of the state is achieved according to,

$$[\hat{\mathbf{x}}_t^*]^{\text{MAP}} = \arg \max_{\mathbf{x}_t^*} p(\mathbf{x}_t^* | \mathbf{z}_{1:T}^*, \mathbf{x}_{1:t-1}^*). \quad (29)$$

2.4) Resampling: Replace particles with negligible weight by new particles in proximity of higher weighted particles.

End For

performance of the trained GP models for positions and velocities in 2D plane. The root-mean-square error (RMSE) of the prediction for each state within time are shown in Fig. 3. From this figure, we can conclude that the motion model with mixed behaviours and random rotation can be learned and approximated accurately using non-parametric GP regression.

B. Tracking Performance

This experiment compares and analyses the performance of different tracking methods. The performance of the target tracking is evaluated using the RMSE of the target positions. The proposed algorithm is compared with two other methods: The traditional PF with full information and the interacting multiple model-particle filter (IMM-PF) method with partial prior information. Specifically, for the traditional PF method, the dynamic motion mode in real-time and the turn rate for CT model ω_t are fully-known when operating the filter. This gives us measure of the optimum tracking performance. While for the IMM-PF method, the statistical distribution of the turn rate, i.e., $\omega_t \sim \mathcal{N}(\mu_\omega, \Sigma_\omega)$, are known as a *prior* information. The motion model is a mixture by CV and CT, therefore, two different multiple model (MM) set are used. The first denoted as IMM-PF1 is set to be $\{\text{CV}, \text{CT}_{\omega=\mu_\omega}\}$. The second denoted as IMM-PF2 is designed based on minimum distribution-mismatch design rule and

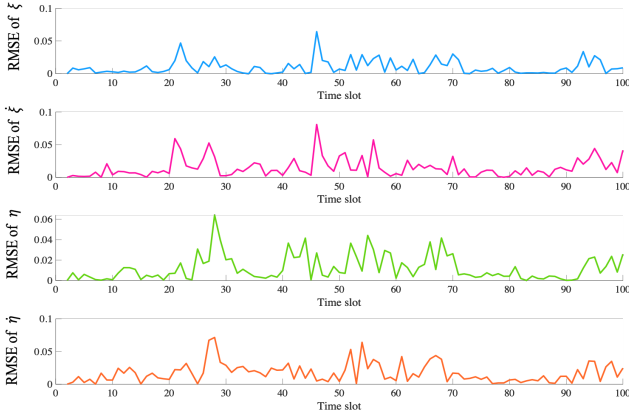


Fig. 3: Prediction performance of GP models: RMSE of the prediction for each state.

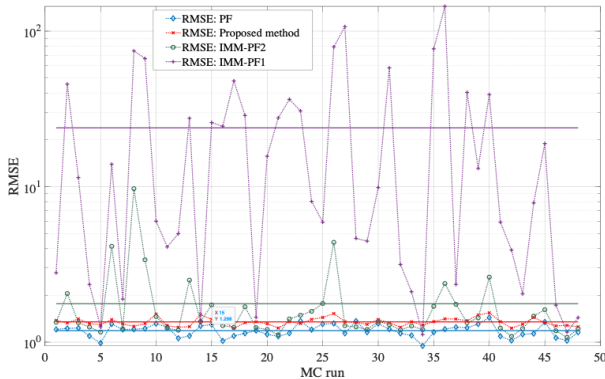


Fig. 4: Comparison of RMSE tracking performance achieved by different methods, and the straight lines show the mean value of the RMSE.

denoted as $\{CV, CT_{\omega=\omega(1)}, \dots, CT_{\omega=\omega(r)}\}$, where Υ is the number of the equal intervals that the cumulative distribution function (CDF) of ω is divided into. The comparisons of tracking performance achieved by the proposed GP-PF based method with others are shown in Fig. 4 and Fig. 5. In Fig. 4, $r = 1$, RMSE performance comparison is evaluated by different Monte Carlo realisations, i.e., different random target trajectories. In Fig. 5, RMSE performance with different measurement process noise variance is evaluated, where $r = \{0.1, 1, 2, 5, 10, 20\}$. From these two figures, we see that the proposed method outperforms the IMM-PF methods. Furthermore, the proposed method shows better robustness for different random trajectories of the target. Compared with the traditional PF with full information, the performance of GP-PF is very good considering it has no prior knowledge of the turn rate.

V. CONCLUSION

In this study, we propose a new method for single target tracking for mixed and uncertain motion. The proposed method is based on Gaussian process regression. The experiments show that this method has clear advantages to IMM methods when dealing with model uncertainty, and shows great robustness for random target trajectories and

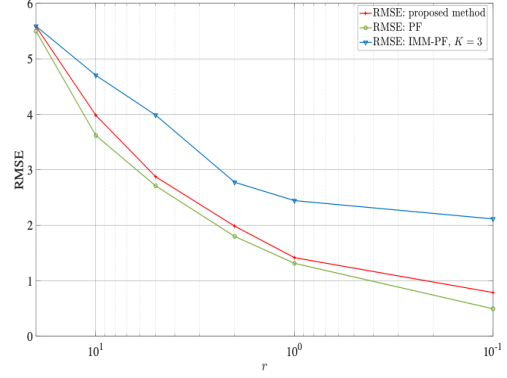


Fig. 5: RMSE tracking performance comparison within various measurement noise density, averaged over 100 Monte Carlo realisations.

system parameters. Further work will explore the proposed filter's performance on real world data and the performance when dealing with the multi-target tracking problem.

REFERENCES

- [1] Z. X. Liu, Q. Q. Zhang, L. Q. Li, W. X. Xie, "Tracking multiple maneuvering targets using a sequential multiple target Bayes filter with jump Markov system models," *Neurocomputing*, vol. 216, pp. 183–191, 2016.
- [2] X. Rong Li and V. P. Jilkov, "Survey of maneuvering target tracking. Part I. Dynamic models," in *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1333-1364, Oct. 2003.
- [3] X. Rong Li and V. P. Jilkov, "Survey of maneuvering target tracking. Part V. Multiple-model methods," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1255-1321, Oct. 2005.
- [4] J. L. Yang, L. Yang, Z. Y. Xiao, and J. J. Liu, "Adaptive parameter particle CBMeMBer tracker for multiple maneuvering target tracking," *IEURASIP J ADV SIG PR*, Vol. 2016, pp. 1-11, May 2016).
- [5] Z. X. Liu, D. H. Wu, W. X. Xie and L. Q. Li, "Tracking the Turn Maneuvering Target Using the Multi-Target Bayes Filter with an Adaptive Estimation of Turn Rate", *Sensors*, vol. 17, no. 373, pp. 1-14, Feb. 2017).
- [6] J. Son, M. Baek, M. Cho, and B. Han, "Multi-object tracking with quadruplet convolutional neural networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2017, pp. 5620–5629.
- [7] S. Jung, I. Schlangen, and A. Charlish, "Sequential Monte Carlo Filtering with Long Short-Term Memory Prediction," in *Proc. of 22nd Int. Conf. on Info. Fusion*, Jul. 2019.
- [8] N. Wahlström and E. Özkan, "Extended Target Tracking Using Gaussian Processes," in *IEEE Trans. Signal Process.*, vol. 63, no. 16, pp. 4165-4178, Aug. 2015.
- [9] J. Wagberg, D. Zachariah, T. Schön, and P. Stoica, "Prediction performance after learning in Gaussian process regression," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, vol. 54, pp. 1264–1272, Apr. 2017.
- [10] J. Ko and D. Fox, "GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models," in *Proc. Int. Conf. Intell. Robots Syst.*, Nice, France, Sep. 2008, pp. 3471–3476.
- [11] Y. Zhao, C. Fritsche, G. Hendeby, F. Yin, T. Chen and F. Gunnarsson, "Cramér–Rao Bounds for Filtering Based on Gaussian Process State-Space Models," in *IEEE Trans. Signal Process.*, vol. 67, no. 23, pp. 5936-5951, Dec. 2019.
- [12] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEEE Proc. F—Radar Signal Process.*, vol. 140, no. 2, pp. 107–113, Apr. 1993.
- [13] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," in *Nonlinear Filtering Handbook*. London, U.K.: Oxford Univ. Press, 2011, pp. 656–704.
- [14] M. S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," in *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174-188, Feb. 2002.