2019 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, OCT. 13–16, 2019, PITTSBURGH, PA, USA

JUMP MODELS FOR INTENT PREDICTION IN HIGHLY PERTURBED SCENARIOS

Runze Gan, Jiaming Liang, Bashar I. Ahmad and Simon Godsill

Signal Processing and Communications (SigProC) Laboratory
Engineering Department, University of Cambridge
Trumpington Street, Cambridge, UK, CB2 1PZ
Emails:{rg605, jl809, bia23, sjg30}@cam.ac.uk

ABSTRACT

The motion of a tracked object often has long term underlying dependencies due to premeditated actions dictated by intent, such as destination. Revealing this intent, as early as possible, can enable advanced intelligent system functionalities for conflict/opportunity detection and automated decision making, for instance in surveillance and human computer interaction. This paper presents a novel Bayesian intent inference framework that utilises sequential Monte Carlo (SMC) methods to determine the destination of a tracked object exhibiting unknown jump behaviour. The latter can arise from the object undertaking fast maneuvers (e.g. for obstacle avoidance) and/or due to external uncontrollable environmental perturbations. Suitable intent-driven stochastic models and inference routines are introduced. The effectiveness of the proposed approach is demonstrated using synthetic and real data.

Index Terms— Bayesian intent prediction, sequential Monte Carlo, variable rate particle filter, Rao-Blackwellisation

1. INTRODUCTION

Unlike conventional sensor-level tracking algorithms devised for estimating the state of an object (e.g. position, velocity, etc.), a meta-level tracker aims to infer the intent driving the object motion, for instance its final destination. Such intent prediction techniques facilitate automated decision making and early detection of conflict or opportunities, thereby introducing smart predictive functuinalities. They have been used in various application areas, such as human computer interaction (e.g. to robustly recognise symbolic gestures [1] and enable touch-free displays [2]), maritime surveillance (e.g. for vessel trajectory or activity forecasting [3, 4, 5]), robotics (e.g. intelligent navigation in the presence of other agents [6, 7]), and advanced driver assistance systems [8, 9], to name a few.

1.1. Problem Statement and Contributions

In this paper, we tackle the meta-level problem of predicting the destination (e.g. harbour or on-display selectable item) of a tracked object (e.g. vessel or pointing apparatus) whose motion contains erratic, unknown, jumps. They can be due to fast maneouver(s) undertaken by the target, for instance to avoid unforeseen obstacles or conflicts. The present jumps or jolts can also be induced by environmental perturbations, such as the impact of severe vibrations/forces on tracked freehand gesture movements in a moving vehicle, or weather conditions (e.g. wind) on small unmanned air systems. Let $\mathbb{D}$ be a set of $N_{\mathcal{D}}$ possible destinations $\{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_{N_{\mathcal{D}}}\}$. The objective is to evaluate the probability of each destination $\mathcal{D}_i$ being the intended endpoint $\mathcal{D}_I$, from the available measurements $\boldsymbol{y}_{1:n} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_n\}$ at time instant t_n,

$$p(\mathcal{D}_I = \mathcal{D}_i|\boldsymbol{y}_{1:n}) \propto p(\boldsymbol{y}_{1:n}|\mathcal{D}_I = \mathcal{D}_i)p(\mathcal{D}_I = \mathcal{D}_i). \quad (1)$$

The observation received at t_n is denoted by $\boldsymbol{y}_n$ and it is assumed to be derived from a true, but unknown, underlying target state $\boldsymbol{x_n}$.

Here, new stochastic nonlinear motion models, based on Ornstein-Uhlenbeck (OU) processes, are proposed to capture the influence of intent on the target motion. A novel Bayesian intent inference framework is then introduced. It utilises the *variable rate particle filter* (VRPF) [10, 11] and *Rao-Blackwellisation* [12]. The developed approach in this paper can not only reliably and efficiently predicts the destination of the target, but also it can provide accurate estimates of the tracked object trajectory and times of the detected jumps. The efficacy of this technique is demonstrated using synthetic maritime trajectories and real freehand pointing data.

1.2. Related Work

We recall that there are numerous well-established sensor-level tracking algorithms for estimating the posterior of the target state $p(\boldsymbol{x}_n|\boldsymbol{y}_{1:n})$. In particular, several SMC methods exist for inferring $p(\boldsymbol{x}_n|\boldsymbol{y}_{1:n})$ assuming nonlinear motion behaviour including jumps/jolts [13, 10, 14]. For example, in the field of financial data modelling, a Langevin dynamics based jump-diffusion model is shown in [13] to reliably account for the effects of sudden changes in the price/trend. In a more closely related application [15], a non-destination-aware dynamic model plus a jump diffusion process are used to model hand movements whilst interacting with an in-vehicle display, i.e. pointing motion can be subjected to severe perturbations due to road/driving conditions. Detected jumps are removed during a pre-processing phase and the filtered pointing gesture data is used for predicting the intended on-display destination. Contrary to such prior work, jumps/jolts are here considered as an integrated part of a destination-dependent model and not decoupled as in [15]. Hence, a unified treatment of the intent prediction task is introduced in this paper with a computationally efficient inference routine based on the Rao-Blackwellised VRPF (R-BVRPF).

Various data driven prediction-classification methods rely on a dynamical model and/or *pattern of life* learnt from previously recorded data, e.g. [16, 17, 18, 19]. Whilst such techniques typically require substantial parameters training from extensive data sets (not always available), a probabilistic model-based framework is adopted here. It uses known dynamical and measurements models, with a few unknown parameters [20]. Subsequently, an efficient inference approach, which requires minimal training, is introduced.

978-1-7281-0824-7/19/$31.00

Previous work on Bayesian intent prediction within an object tracking framework exclusively addressed linear Gaussian set-ups [2, 21, 22], whereas here linear motion models with jump components are considered. Nonetheless, it has been shown in [3, 4, 5, 21] that linear and Gaussian OU processes can accurately model intent-driven vessel and pointing hand movements in 3D, leading to improved state estimates and future trajectory predictions. Such processes are utilised in this paper to capture the intent-driven aspect of the target motion and a new model is proposed (see Section 2.1).

Intent inference methods that utilise stochastic context-free grammars and reciprocal process were proposed in [23, 24] and extended to apply to nonlinear systems with particle-filter-based schemes in [1] and [25]. The underlying premise of these techniques is that the tracked object can follow a set of predefined trajectories within a discritised state space formulation. This limitation can render them impractical for scenarios where the object motion is unconstrained (e.g. freehand pointing motion). On the other hand, the proposed approach in this paper uses continuous-time state space models without imposing any constraints on the path the target has to follow en-route to its intended endpoint.

2. PROPOSED SYSTEM MODEL

Mean reverting models, which are based on OU processes, can capture the influence of premeditated actions or destination on the object motion via the mean term, e.g. as in [21, 4, 5, 3]. Here, a new mean (destination) reverting model, denoted the *equilibrium reverting acceleration* (ERA), is first introduced, representing the linear intent-driven component of the motion behaviour. The ERA and the *equilibrium reverting velocity* (ERV) model in [21] are then extended to include the jump/jolt components to account for the impact of severe perturbations and/or fast maneuvering of the object.

2.1. Linear Gaussian Component of the Dynamic Models

A continuous time OU-based model with state $\boldsymbol{x}_t$ is given by following *stochastic differential equation* (SDE)

$$d\boldsymbol{x}_t = \boldsymbol{A}(\boldsymbol{\mu}_d - \boldsymbol{x}_t)dt + \boldsymbol{\sigma}d\boldsymbol{\beta}_t, \tag{2}$$

where $\boldsymbol{\beta}_t$ is a Brownian motion; other parameters will be defined separately. The proposed ERA model state vector incorporates acceleration and in one-dimension (for simplicity) it is governed by the SDE

$$d\ddot{x}_t = \eta(p^d - x_t)dt - \rho\dot{x}_t dt - \gamma\ddot{x}dt + \sigma d\beta, \tag{3}$$

where η (> 0) is the reversion factor, coefficients ρ and γ are the positive damping factor set for the velocity and acceleration, respectively. The known location of the intended destination is denoted by p^d. In this model, since the motion is controlled by a reversion term proportional to the difference between the object and its intended destination positions, the target will gravitate towards the endpoint with increasing acceleration at larger separations. The damping terms aim to revert both velocity and acceleration to zeros as the object approaches its endpoint, i.e. reaches the equilibrium state. To ensure wide-sense stationarity of the system in (3), the model parameters should have the following relationship

$$0 < \eta < \rho\gamma. \tag{4}$$

This can be proved by applying *Routh-Hurwitz* stability criterion for a third-order system.

From (2) the ERA model in a s-dimensional spatial system (i.e. its state is of dimensions $3 \times s$) is given by

$$\begin{aligned} \boldsymbol{x}_t &= [x_{1,t}, \dot{x}_{1,t}, \ddot{x}_{1,t}, \ldots, x_{s,t}, \dot{x}_{s,t}, \ddot{x}_{s,t}]', \\ \boldsymbol{\mu}_d &= [p_1^d, 0, 0, \ldots, p_s^d, 0, 0]', \\ \boldsymbol{\sigma} &= [0, 0, \sigma_1, \ldots, 0, 0, \sigma_s]', \end{aligned} \tag{5}$$

where σ_m specifies the standard deviation of noise driving the velocity component on the m^{th} axis. Note that the form of $\boldsymbol{\sigma}$ assumes noises in different coordinate axes are uncorrelated. The $3s \times 3s$ block diagonal Matrix $\boldsymbol{A}$ is defined by $diag\{\boldsymbol{A_1}, ..., \boldsymbol{A_s}\}$, such that

$$\boldsymbol{A}_m = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ \eta_m & \rho_m & \gamma_m \end{bmatrix}. \tag{6}$$

If the state only includes the position and velocity information, the resulting destination-reverting model is the ERV model in [21]. It is described by the one-dimensional SDE

$$d\dot{x}_t = \eta(p^d - x_t)dt - \rho\dot{x}_t dt + \sigma d\beta \tag{7}$$

with the restoring force parameter η, and the damping factor ρ, the target position x_t will gradually revert to destination position p^d with zero velocity. Similar to (5), the ERV model has

$$\begin{aligned} \boldsymbol{x}_t &= [x_{1,t}, \dot{x}_{1,t}, \ldots, x_{s,t}, \dot{x}_{s,t}]', \\ \boldsymbol{\mu}_d &= [p_1^d, 0, \ldots, p_s^d, 0]', \\ \boldsymbol{\sigma} &= [0, \sigma_1, \ldots, 0, \sigma_s]', \\ \boldsymbol{A} &= diag\{\boldsymbol{A_1}, ..., \boldsymbol{A_s}\}, \\ \boldsymbol{A}_m &= \begin{bmatrix} 0 & -1 \\ \eta_m & \rho_m \end{bmatrix}. \end{aligned} \tag{8}$$

Subsequently, integrating the SDE in (2), which covers both the ERA and ERV models, with respect to the multi-dimensional state $\boldsymbol{x}_t$ over the time interval $[t, t+h]$, yields

$$\boldsymbol{x}_{t+h} = \boldsymbol{F}_{i,h}\boldsymbol{x}_t + \boldsymbol{M}_{i,h} + \boldsymbol{w}_t. \tag{9}$$

This represents the linear Gaussian component of the adopted, intrinsically destination-dependent, models (i.e excluding jumps) where $\boldsymbol{x}_{t+h}$ and $\boldsymbol{x}_t$ are the latent state at times t and $t+h$, respectively. $\boldsymbol{w}_t \sim \mathcal{N}(0, \boldsymbol{Q}_{i,h})$ is the noise term and matrices $\boldsymbol{F}_{i,h}$, $\boldsymbol{Q}_{i,h}$ and vector $\boldsymbol{M}_{i,h}$ are determined by the time interval h and $\boldsymbol{\mu_d}$ pertaining to destination $\mathcal{D}_i \in \mathbb{D}$. They are given by

$$\begin{aligned} \boldsymbol{F}_{i,h} &= e^{-\boldsymbol{A}h}, \\ \boldsymbol{M}_{i,h} &= (\boldsymbol{I} - e^{-\boldsymbol{A}h})\boldsymbol{\mu_d}, \\ \boldsymbol{Q}_{i,h} &= \int_t^{t+h} e^{\boldsymbol{A}(t+h-v)}\boldsymbol{\sigma}\boldsymbol{\sigma}'e^{\boldsymbol{A}'(t+h-v)}dv, \end{aligned} \tag{10}$$

where $\boldsymbol{A}$, μ_d, and $\boldsymbol{\sigma}$ have been specified above for the ERV and ERA models; $\boldsymbol{I}$ is $2s \times 2s$ identity matrix for the ERV model and $3s \times 3s$ identity matrix for the ERA model. The calculation of the covariance matrix $\boldsymbol{Q_{i,h}}$ can be simplified by *matrix fraction decomposition* [26].

2.2. Destination-dependent Dynamic Models with Jump Terms

Here, jump processes are added to the destination reverting models in (9), e.g. ERA and ERV, to improve the intent prediction accuracy as well as the tracking performance under the influence of severe external perturbations on the target motion and/or fast manoeuvring behaviour. As the jumps do not necessarily arrive synchronously with the observations, we use $\{\tau_0, \tau_1, \ldots, \tau_k, \ldots\}$ to denote the unknown jump times. The general SDE with jumps can be then formalised as

$$d\boldsymbol{x}_t = \boldsymbol{A}(\boldsymbol{\mu}_d - \boldsymbol{x}_t)dt + \boldsymbol{\sigma}d\beta_t + \boldsymbol{B}dJ_t \tag{11}$$

where $\boldsymbol{B} = [\boldsymbol{B_1}, ..., \boldsymbol{B_s}]'$ such that $\boldsymbol{B_i}$ is $[0, 1]'$ for the ERV model and $[0, 0, 1]'$ for the ERA model. Component J_t is a pure jump process with $J_t = \sum_{\tau_k < t} S_k$ and S_k the jump size. Here, we assume that the jumps follow a Gauss-Poisson process, thus $S_k \sim \mathcal{N}(S_k|\mu_J, \sigma_J^2)$ and $\tau_k - \tau_{k-1} \sim \exp_{\lambda_J}(\cdot)$; μ_J and σ_J are the mean and covariance of the jump size respectively. Whereas, λ_J^{-1} is the mean value of the jump interarrival time.

When jumps arrive within the time interval $(t, t + h)$, the state transition density between t and $t + h$ can be expressed by

$$p(\boldsymbol{x}_{t+h}|\boldsymbol{x}_t, \tau_{(K_t+1):K_{t+h}}) = \mathcal{N}(\boldsymbol{x}_{t+h} \mid \boldsymbol{\mu}^*_{t+h}, \boldsymbol{\Sigma}^*_{t+h}) \tag{12}$$

where K_t is the count of jumps up to time t, i.e., $K_t = \#\{k, \forall \tau_k < t\}$, $\tau_{K_t+1:K_{t+h}}$ is the sequence of jump times in the interval $[t, t + h)$ and

$$\boldsymbol{\mu}^*_{t+h} = \boldsymbol{F}_h\boldsymbol{x}_t + \boldsymbol{M}_h + \sum_{k=K_t+1}^{K_{t+h}} \boldsymbol{F}_{t+h-\tau_k}\boldsymbol{B}\mu_J, \tag{13}$$

$$\boldsymbol{\Sigma}^*_{t+h} = \boldsymbol{Q}_h + \sum_{k=K_t+1}^{K_{t+h}} \boldsymbol{F}_{t+h-\tau_k}\boldsymbol{B}\sigma_J^2\boldsymbol{B}'\boldsymbol{F}'_{t+h-\tau_k}. \tag{14}$$

with matrices/vectors $\boldsymbol{F}_h$, $\boldsymbol{M}_h$ and $\boldsymbol{Q}_h$ specified in (10). Detailed derivation of (12)-(14) can be found in [27]. For simplicity, we assume below that the jumps only occur at key driving component of the target state, i.e. velocity and acceleration components for the ERV and ERA models, respectively. However, jumps could also be included in other state components by simply modifying matrix $\boldsymbol{B}$.

In scenarios where there are no jumps, i.e. $K_t + 1 > K_{t+h}$, we define $\tau_{K_t+1:K_{t+h}} = \tau_{K_{t+h}}$, and the transition density stays in the linear Gaussian form $\mathcal{N}(\boldsymbol{x}_{t+h}|\boldsymbol{F}_h\boldsymbol{x}_t + \boldsymbol{M}_h, \boldsymbol{Q}_h)$. It is noted that since $N_{\mathcal{D}}$ nominal destinations are considered, one destination-dependent dynamic model per endpoint is constructed with $\boldsymbol{\mu}_d$ corresponding to a particular $\mathcal{D}_i$.

2.3. Observation Models

Assuming that only position measurements of the target are available for simplicity, a linear observation model with an additive Gaussian noise is adopted,

$$\boldsymbol{y_t} = \boldsymbol{H}\boldsymbol{x}_t + \boldsymbol{v_t} \tag{15}$$

where $\boldsymbol{v_t} \sim \mathcal{N}(0, \boldsymbol{R})$ and $\boldsymbol{H} = diag\{\boldsymbol{H}_1, ..., \boldsymbol{H}_s\}$; $\boldsymbol{H}_m = diag\{1, 0\}$ for ERV and $\boldsymbol{H}_m = diag\{1, 0, 0\}$ for ERA. Other such linear Gaussian observation models can be applied.

3. INTENT INFERENCE

3.1. Variable Rate Particle Filter for Intent Prediction

In this section, the Bayesian inference framework is presented to infer sequentially the intended destination of the target, whose motion can exhibit sudden jumps at any instant. Additionally, the kinematics (e.g. position, velocity and acceleration) and the jump information (e.g. jump times and sizes) can be estimated. According to (1), we require

$$p(\boldsymbol{y}_{1:n}|\mathcal{D}_i) = p(\boldsymbol{y}_n|\boldsymbol{y}_{1:n-1}, \mathcal{D}_i)p(\boldsymbol{y}_{1:n-1}|\mathcal{D}_i) \tag{16}$$

with the condition $\mathcal{D}_I = \mathcal{D}_i$ abbreviated as $\mathcal{D}_i$ henceforth. The prior $p(\mathcal{D}_I = \mathcal{D}_i)$, which is independent of the sensory observations, is assumed to be known, e.g. from contextual information; otherwise a uniform uninformative prior $p(\mathcal{D}_I = \mathcal{D}_i) = 1/N$ can be applied. Specifically, $p(\boldsymbol{y}_n|\boldsymbol{y}_{1:n-1}, \mathcal{D}_i)$ at t_n is sought since $p(\boldsymbol{y}_{1:n-1}|\mathcal{D}_i)$ is available from the previous time step t_{n-1}.

In our case, $p(\boldsymbol{y}_n|\boldsymbol{y}_{1:n-1}, \mathcal{D}_i)$ cannot be obtained in closed form because of the random unknown jumps. Therefore, we employ an RBVRPF to compute this for each destination, in a Monte Carlo sense. The target distribution of the RBVRPF operating on the unknown jumps up to time t_n can then be factorised as

$$\begin{aligned} p(\tau_{1:K_{t_n}}|\boldsymbol{y}_{1:n}, \mathcal{D}_i) &\propto p(\boldsymbol{y}_n|\boldsymbol{y}_{1:n-1}, \tau_{1:K_{t_n}}, \mathcal{D}_i) \\ &\times p(\tau_{(K_{t_{n-1}}+1):K_{t_n}}|\tau_{1:K_{t_{n-1}}})p(\tau_{1:K_{t_{n-1}}}|\boldsymbol{y}_{1:n-1}, \mathcal{D}_i), \end{aligned} \tag{17}$$

with $p(\boldsymbol{y}_n|\boldsymbol{y}_{1:n-1}, \tau_{1:K_{t_n}}, \mathcal{D}_i)$ defined in (20), the second term is the prior distribution for jump times and the last one is the posterior obtained at t_{n-1}. If the interarrival time between jumps is designed as stated in Section 2.2, the memoryless property of the exponential distribution will allow simulating a jump time sequence from $p(\tau_{(K_{t_{n-1}}+1):K_{t_n}}|\tau_{1:K_{t_{n-1}}})$ in the following way: 1) sample the jump time $\tau_{K_{t_{n-1}}+1}$ from $\exp_{\lambda_J}(\cdot|\mathrm{t_{n-1}})$, 2) consecutively sample the next jump time conditioned on the last sample until the k^{th} sample $\tau_{K_{t_{n-1}}+k} > t_n$ is obtained and finally, 3) the previous $k - 1$ samples, i.e. $\tau_{(K_{t_{n-1}}+1):(K_{t_{n-1}}+k-1)}$, are kept as the desired sequence.

The *unnormalised* importance weight $\tilde{\omega}$ in the VRPF for each destination is expressed by

$$\begin{aligned} \tilde{\omega}_n^{(p,i)} &= \frac{p(\tau^{(p)}_{1:K_{t_n}}|\boldsymbol{y}_{1:n}, \mathcal{D}_i)}{q(\tau^{(p)}_{1:K_{t_n}}|\boldsymbol{y}_{1:n}, \mathcal{D}_i)} \propto \frac{p(\tau^{(p)}_{1:K_{t_{n-1}}}|\boldsymbol{y}_{1:n}, \mathcal{D}_i)}{q(\tau^{(p)}_{1:K_{t_{n-1}}}|\boldsymbol{y}_{1:n}, \mathcal{D}_i))} \\ &\times \frac{p(\boldsymbol{y}_n|\boldsymbol{y}_{1:n-1}, \tau^{(p)}_{1:K_{t_n}}, \mathcal{D}_i)p(\tau^{(p)}_{(K_{t_{n-1}}+1):K_{t_n}}|\tau^{(p)}_{1:K_{t_{n-1}}})}{q(\tau^{(p)}_{(K_{t_{n-1}}+1):K_{t_n}}|\tau^{(p)}_{1:K_{t_n}}, \boldsymbol{y}_{1:n}, \mathcal{D}_i)} \\ &= \frac{\omega_{n-1}^{(p,i)}}{\nu_{n-1}^{(p,i)}} \frac{p(\boldsymbol{y}_n|\boldsymbol{y}_{1:n-1}, \tau^{(p)}_{1:K_{t_n}}, \mathcal{D}_i)p(\tau^{(p)}_{(K_{t_{n-1}}+1):K_{t_n}}|\tau^{(p)}_{1:K_{t_{n-1}}})}{q(\tau^{(p)}_{(K_{t_{n-1}}+1):K_{t_n}}|\tau^{(p)}_{1:K_{t_n}}, \boldsymbol{y}_{1:n}, \mathcal{D}_i)} \end{aligned} \tag{18}$$

where the superscript (p) denotes the p^{th} particle, while (i) is for the i^{th} destination. When the total number of particles is $N_{\mathcal{P}}$, $\{\omega_n^{p,i}\}_{p=1}^{N_{\mathcal{P}}}$ are the *normalised* importance weights while $\{\nu_t^{(p,i)}\}_{p=1}^{N_{\mathcal{P}}}$ can be viewed as the weights of a discrete distribution from which past paths $\tau_{1:K_{t_{n-1}}}$ may be drawn. The selection weight ν offers a way to incorporate more sophisticated proposals which can lead to improved performance of the filter. Here, we choose $\nu = \omega$ when *effective sample size* (ESS) based resampling criterion is met and $\nu = const.$ when it is not. This corresponds to the standard resampling scheme in the *bootstrap* filter [28, 29]. Nevertheless, other choices such as the *auxiliary particle filter* [30] and the bias sampling scheme in [13] may be applied. When the *bootstrap* filter is chosen, in which the jump times are drawn from the prior, the weight update equation (18) becomes

$$\tilde{\omega}_n^{(p,i)} = \omega_{n-1}^{(p,i)}p(\boldsymbol{y}_n|\boldsymbol{y}_{1:n-1}, \tau^{(p)}_{1:K_{t_n}}, \mathcal{D}_i). \tag{19}$$

It can be seen from (19) that to update the particle weights we only need to evaluate the destination-dependent likelihood function $p(\boldsymbol{y}_n|\boldsymbol{y}_{1:n-1}, \tau^{(p)}_{1:K_{t_n}}, \mathcal{D}_i)$ which can be computed via the Kalman filter *prediction error decomposition* (PED) [31], summarised as follows:

$$p(\boldsymbol{y}_n|\boldsymbol{y}_{1:n-1}, \tau^{(p)}_{1:K_{t_n}}, \mathcal{D}_i) = \int p(\boldsymbol{y}_n|\boldsymbol{x}_n)p(\boldsymbol{x}_n|\tau^{(p)}_{1:K_{t_n}}, \boldsymbol{y}_{1:n-1}, \mathcal{D}_i)d\boldsymbol{x}_n, \tag{20}$$

where $p(\boldsymbol{x}_n|\tau^{(p)}_{1:K_{t_n}}, \boldsymbol{y}_{1:n-1}, \mathcal{D}_i)$ is calculated via the prediction step of the Kalman filter given (12). The superscript p indicated that $N_{\mathcal{P}}$ Kalman filters are needed in order to evaluate this likelihood for each particle filter. Subsequently, it can be shown that the predictive likelihood required for inferring the intended destination in (16) is given by

$$p(\boldsymbol{y}_n|\boldsymbol{y}_{1:n-1}, \mathcal{D}_i) = \int p(\boldsymbol{y}_n|\boldsymbol{y}_{1:n-1}, \tau_{1:K_{t_n}}, \mathcal{D}_i) \times p(\tau_{(K_{t_{n-1}}+1):K_{t_n}}|\tau_{1:K_{t_{n-1}}})p(\tau_{1:K_{t_{n-1}}}|\boldsymbol{y}_{1:n-1}, \mathcal{D}_i)d\tau_{1:K_{t_n}}. \tag{21}$$

Supposing that the weighted posterior given by the *bootstrap* VRPF associated with $\mathcal{D}_i$ at time t_{n-1} is

$$p(\tau_{1:K_{t_{n-1}}}|\boldsymbol{y}_{1:n-1}, \mathcal{D}_i) \approx \sum_{p=1}^{N_{\mathcal{P}}} \omega^{(p,i)}_{n-1}\delta(\tau^{(p)}_{1:K_{t_{n-1}}}), \tag{22}$$

the integral in (21) can be approximated by

$$\begin{aligned} p(\boldsymbol{y}_n|\boldsymbol{y}_{1:n-1}, \mathcal{D}_i) &\approx \sum_{p=1}^{N_{\mathcal{P}}} \omega^{(p,i)}_{n-1} p(\boldsymbol{y}_n|\boldsymbol{y}_{1:n-1}, \tau^{(p)}_{1:K_{t_n}}, \mathcal{D}_i) \\ &= \sum_{p=1}^{N_{\mathcal{P}}} \tilde{\omega}^{(p,i)}_n \end{aligned} \tag{23}$$

3.2. Filtering Implementation and Jump Time Detection

One computationally intensive task is to sample the jumps. This can be addressed by only sampling $N_{\mathcal{P}}$ jump times once at each time instant for all $N_{\mathcal{D}}$ RBVRPFs. Other programming strategies such as stacking matrices (i.e. parallel Kalman filters) could be considered. A pseudo-code of the proposed inference procedure is given in Algorithm 1. Note that the implementation each particle stores the following data: current weight, mean and covariance of the hidden state, as well as the collection of jump times. One may estimate of the target state at t_n within the same framework using

$$p(\boldsymbol{x}_n|\boldsymbol{y}_{1:n}, \mathcal{D}_i) \approx \sum_{p=1}^{N_{\mathcal{P}}} \omega^{(p,i)}_n p(\boldsymbol{x}_n|\boldsymbol{y}_{1:n}, \tau^{(p)}_{1:K_{t_n}}, \mathcal{D}_i) \tag{24}$$

where the distribution in the summation is given by the Kalman filter running on the p^{th} particle of the i^{th} filter. Besides, jump times can also be detected according to (22). Specifically, a rough estimation of jump time can be carried out by calculating the weighted percentage of jump particles for each observation interval. Furthermore, the probability density of the jump time (22) can be obtained using suitable *kernel density estimation* (KDE), which can indicate the most likely jump times.

Algorithm 1: Intent Inference with RBVRPFs

```
Initialisation: Create N_D RBVRPFs, each with N_P particles;
for each observations n = 1 : N captured at t_n do
    for particles p = 1 : N_P do
        Sampling the jump time sequence from prior
          τ^(p)_{K_{t_{n-1}}+1:K_{t_n}} ~ p(τ_{(K_{t_{n-1}}+1):K_{t_n}} | τ_{1:K_{t_{n-1}}});
        Obtain summed jump means (Σ μ_J)_p and
          covariances (Σ Q_J)_p in (13)(14);
    end
    for destinations i = 1 : N_D do
        if Resample then
            Resample particles and set weights ω = 1/D;
        else
            for particles p = 1 : N_P do
                Set jump times
                  τ^(p,i)_{1:K_{t_n}} = [τ^p_{K_{t_{n-1}}+1:K_{t_n}}, τ^(p,i)_{K_{t_1:K_{t_{n-1}}}}];
                Update the mean and covariance with
                  (Σ μ_J)_p and (Σ Q_J)_p via (12) and the
                  Kalman filter
                Calculate unnormalised weight ω̃^(p,i)_n
                  according to (19)(20);
            end
            Produce the predictive likelihood
              p(y_n | y_{1:n-1}, D_i) = Σ_{p=1}^{N_P} ω̃^(p,i)_n;
            For all particles p = 1 : N_P normalize the
              weight ω^(p,i)_n = ω̃^(p,i)_n / Σ_{p=1}^{N_P} ω̃^(p,i)_n;
        end
        Calculate likelihood p(y_{1:n} | D_i) in (16);
    end
    Determine endpoint probability:p(D_I | y_{1:n}) in (1);
end
```

4. RESULTS

The introduced Bayesian framework is tested on both synthetic maritime data and real freehand pointing gesture data in this section. Its performance is evaluated in terms of the jump time detection and the average intent inference success rate (defined below). In the maritime case, the jump represents a sudden fast maneuver which may be caused by strong wind/currents or obstacle avoidance. In Fig. 1, four synthetic vessel trajectory data are generated. The first two tracks (from left to right), i.e. marked with destinations $\mathcal{D}_1$ and $\mathcal{D}_2$, and the other two trajectories are simulated from ERV and ERA models with jumps, respectively. The color along the trajectory, which is corresponding to the KDE result, shows the probability density of the jump. For example, weighted jump particle percentage of the trajectory with destination $\mathcal{D}_2$ in Fig. 2 identifies several distinct jumps, while the KDE result presents the density plot of jump times. It can be clearly noticed either from the color of the trajectory or the KDE result that the proposed inference algorithm can effectively capture most of the present jumps. This is albeit being uncertain about the time of small size jumps (e.g. the first jump in the track with endpoint $\mathcal{D}_2$). It is noted that the jump time detection for ERA model (e.g. $\mathcal{D}_3$ and $\mathcal{D}_4$ trajectories) is more challenging since the influence of an acceleration jump takes longer to reflect on the position observation.

To assess the intent inference performance, we used 10 severely perturbed hand pointing gestures datasets that were collected in an

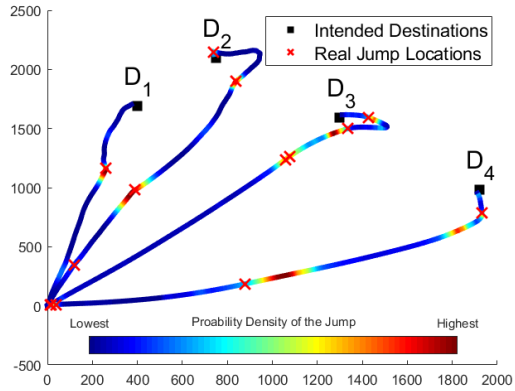

Fig. 1. Synthetic trajectories and particles jump PDF. The left two trajectories (with endpoints $\mathcal{D}_1$ and $\mathcal{D}_2$) were generated from jump-included ERV model, while $\mathcal{D}_3$ and $\mathcal{D}_4$ from jump-ERA model

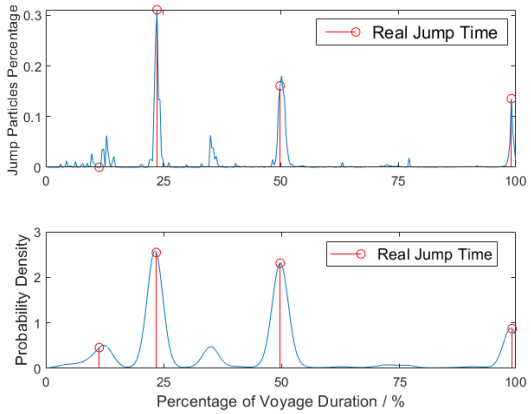

Fig. 2. Weighted jump particles percentage (top) and the jump time KDE (bottom) for trajectory with endpoint $\mathcal{D}_2$.

instrumented vehicle during off-road driving conditions (i.e. high vibrations/forces were experienced by the user interacting with the in-car display mounted to the dashboard). One example of such highly perturbed pointing movement, as tracked by a vision-based system, is shown in Fig. 3. In each pointing task the participant intended to touch one out of 37 selectable on-display icons. The aim is to predict the intended icon as early as possible and facilitate-expedite the selection task. The evaluated success rate is defined as the fraction of the track duration for which the true endpoint is predicted; it is averaged over 10 datasets. Fig. 4 depicts the intent prediction results across the five models, MRD [21], ERV, ERA, ERV with jump process and ERA with jump process. The former three are defined by the linear Gaussian model in (9), i.e. without a jump component. The parameters are hand-tuned and specified in Table. 1.

Table 1. Parameters Set

Model	Dynamics	Jumps
Jump-ERV	$\eta = 60, \rho = 15, \sigma = 450$	$\mu_J = 0, \sigma_J = 866, \lambda_J^{-1} = 1$
Jump-ERA	$\eta = 435, \rho = 170, \gamma = 16, \sigma = 5500$	$\mu_J = 0, \sigma_J = 8670, \lambda_J^{-1} = 0.8$

Compared with the results in [15], which employ the same

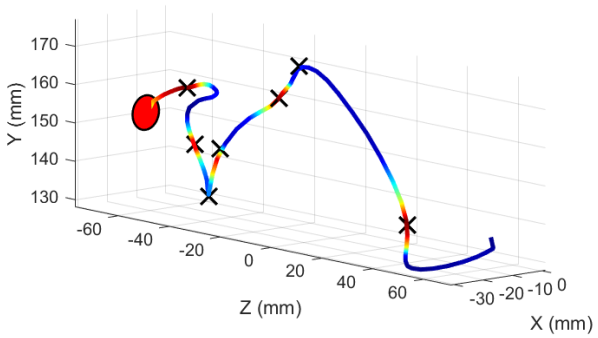

Fig. 3. Highly perturbed 3D pointing track with ERA-jump model. The red circle is the endpoint, color of the trajectory is the estimated jump PDF from KDE and crosses are PDF peaks.

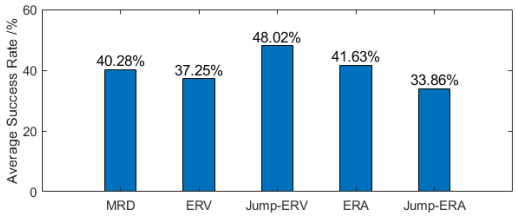

Fig. 4. Average Success Rate of Five Algorithms

dataset, the highest success rate is achieved by the ERV with jumps using the introduced VRPF intent predictor. Although the proposed ERA model with jumps performs less well on this type of data in terms of intent prediction accuracy, we include it here because we envisage it will give benefits in other tracking scenarios. With the same hand-tuned parameters set for intent inference We also note that jump-ERA is capable of accurately capturing the jump times. This can be seen in Fig. 3 where the color of the recorded track gives the jump probability density function (PDF) from KDE ; the black crosses mark the peaks of the PDF. Visually, the predicted jump times in Fig. 3 are consistent with human intuition.

5. CONCLUSION

This paper introduces an efficient VRPF-based Bayesian intent prediction framework for sequentially inferring the intended destination of an object whose motion may include erratic behaviour, namely jumps due to perturbations or fast maneuvers. This approach also facilitates estimating the object's kinetics as well as the jump times. The intent prediction results show the jump-ERV model outperforms previous schemes. Although the proposed novel jump-ERA motion model is not suitable for perturbed freehand pointing data, it still presents a new modelling capability for motion behaviour with jumps.

6. REFERENCES

[1] M. Fanaswala and V. Krishnamurthy, "Meta-level tracking for gestural intent recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 5600–5604.

[2] B. I. Ahmad, J. K. Murphy, S. Godsill, P. M. Langdon, and R. Hardy, "Intelligent interactive displays in vehicles with intent prediction: A bayesian framework," *IEEE Signal Processing Magazine*, vol. 34, no. 2, pp. 82–94, March 2017.

[3] M. Üney, L. M. Millefiori, and P. Braca, "Data driven vessel trajectory forecasting using stochastic generative models," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 8459–8463.

[4] M. Üney, L. M. Millefiori, and P. Braca, "Prediction of rendezvous in maritime situational awareness," in *2018 21st International Conference on Information Fusion (FUSION)*, July 2018, pp. 622–628.

[5] L. M. Millefiori, P. Braca, K. Bryan, and P. Willett, "Modeling vessel kinematics using a stochastic mean-reverting process for long-term prediction," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 5, pp. 2313–2330, October 2016.

[6] G. Best and R. Fitch, "Bayesian intention inference for trajectory prediction with an unknown goal destination," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 5817–5823.

[7] H. Chiang, N. Rackley, and L. Tapia, "Stochastic ensemble simulation motion planning in stochastic dynamic environments," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 3836–3843.

[8] T. Bando, K. Takenaka, S. Nagasaka, and T. Taniguchi, "Unsupervised drive topic finding from driving behavioral data," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, June 2013, pp. 177–182.

[9] B. Ahmad, P. M. Langdon, and S. Godsill, "A meta-tracking approach for predicting the driver or passenger intent," 07 2018, pp. 1–5.

[10] S. J. Godsill, J. Vermaak, W. Ng, and J. F. Li, "Models and algorithms for tracking of maneuvering objects using variable rate particle filters," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 925–952, May 2007.

[11] P. Bunch and S. Godsill, "Particle smoothing algorithms for variable rate models," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1663–1675, April 2013.

[12] C. P. Robert and G. Casella, "Monte Carlo statistical methods," 2004.

[13] H. L. Christensen, J. Murphy, and S. J. Godsill, "Forecasting high-frequency futures returns using online langevin dynamics," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 4, pp. 366–380, Aug 2012.

[14] A. Golightly, "Bayesian filtering for jump-diffusions with application to stochastic volatility," *Journal of Computational and Graphical Statistics*, vol. 18, no. 2, pp. 384–400, 2009.

[15] B. I. Ahmad, J. Murphy, P. M. Langdon, and S. J. Godsill, "Filtering perturbed in-vehicle pointing gesture trajectories: Improving the reliability of intent inference," in *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, Sep. 2014, pp. 1–6.

[16] H.-T. Chiang, N. Rackley, and L. Tapia, "Stochastic ensemble simulation motion planning in stochastic dynamic environments," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2015.

[17] K. Kitani, B. Ziebart, J. Bagnell, and M. Hebert, "Activity forecasting," in *Proc. of European Conf. on Computer Vision*, 2012, pp. 201–214.

[18] T. Bando, K. Takenaka, S. Nagasaka, and T. Taniguchi, "Unsupervised drive topic finding from driving behavioral data," in *Proc. of IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 177–182.

[19] B. Völz, H. Mielenz, I. Gilitschenski, R. Siegwart, and J. Nieto, "Inferring pedestrian motions at urban crosswalks," *IEEE Transactions on Intelligent Transportation Systems*, 2018.

[20] Y. Bar-Shalom, P. Willett, and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithms*, YBS Publishing, 2011.

[21] B. I. Ahmad, J. K. Murphy, P. M. Langdon, S. J. Godsill, R. Hardy, and L. Skrypchuk, "Intent inference for hand pointing gesture-based interactions in vehicles," *IEEE Transactions on Cybernetics*, vol. 46, no. 4, pp. 878–889, April 2016.

[22] B. I. Ahmad, J. K. Murphy, P. M. Langdon, and S. J. Godsill, "Bayesian intent prediction in object tracking using bridging distributions," *IEEE Trans. on Cybernetics*, vol. 48, no. 1, pp. 215–227, 2018.

[23] M. Fanaswala and V. Krishnamurthy, "Spatiotemporal trajectory models for metalevel target tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 30, no. 1, pp. 16–31, Jan 2015.

[24] M. Fanaswala, V. Krishnamurthy, and L. White, "Destination-aware target tracking via syntactic signal processing," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011, pp. 3692–3695.

[25] V. Krishnamurthy and S. Gao, "Syntactic enhancement to vsimm for roadmap based anomalous trajectory detection: A natural language processing approach," *IEEE Transactions on Signal Processing*, vol. 66, no. 20, pp. 5212–5227, Oct 2018.

[26] S. Särkkä, *Recursive Bayesian inference on stochastic differential equations*, Helsinki University of Technology, 2006.

[27] S. Godsill, "Particle filters for continuous-time jump models in tracking applications," *ESAIM: Proc.*, vol. 19, pp. 39–52, 2007.

[28] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," *IEE Proceedings F - Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, April 1993.

[29] S. Godsill, "Particle filtering: the first 25 years and beyond," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 7760–7764.

[30] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of the American Statistical Association*, vol. 94, no. 446, pp. 590–599, 1999.

[31] A. C. Harvey, *Forecasting, structural time series models and the Kalman filter*, Cambridge university press, 1990.