Bayesian Intent Prediction in Object Tracking Using Bridging Distributions

Bashar I. Ahmad[†], James K. Murphy[†], Patrick M. Langdon and Simon J. Godsill

Abstract—In several application areas, such as human computer interaction, surveillance and defence, determining the intent of a tracked object enables systems to aid the user/operator and facilitate effective, possibly automated, decision making. In this paper, we propose a probabilistic inference approach that permits the prediction, well in advance, of the intended destination of a tracked object and its future trajectory. Within the framework introduced here, the observed partial track of the object is modeled as being part of a Markov bridge terminating at its destination, since the target path, albeit random, must end at the intended endpoint. This captures the underlying long term dependencies in the trajectory, as dictated by the object intent. By determining the likelihood of the partial track being drawn from a particular constructed bridge, the probability of each of a number of possible destinations is evaluated. These bridges can also be employed to produce refined estimates of the latent system state (e.g. object position, velocity, etc.), predict its future values (up until reaching the designated endpoint) and estimate the time of arrival. This is shown to lead to a low complexity Kalman-filter-based implementation of the inference routine, where any linear Gaussian motion model, including the destination reverting ones, can be applied. Free hand pointing gestures data collected in an instrumented vehicle and synthetic trajectories of a vessel heading towards multiple possible harbours are utilised to demonstrate the effectiveness of the proposed approach.

Index Terms—Bayesian inference, Kalman filtering, tracking, maritime surveillance, Human computer interactions.

I. INTRODUCTION

IN several application areas such as surveillance, defence and human computer interaction (HCI), the trajectory of a tracked object, e.g. a vessel, jet, pedestrian or pointing apparatus, is driven by its final destination. Thus, *a priori* knowledge of the object endpoint can not only offer vital information on intent, unveil potential conflict or threat and enable task facilitation strategies, but can also produce more accurate tracking routines [1]–[9]. In this paper, we address the problem of predicting the intended destination of a tracked object from a finite set of possible endpoints and the future values of its hidden *state* (e.g. object position, velocity, etc.), given the available noisy observations. This can be viewed as a means to assist or automate timely decision making, planning and resources allocation at a higher system level, compared with a conventional sensor-level tracker. The latter typically focuses on inferring the current value of the latent *state* X_t, with several well-established algorithms [10]–[12].

B. I. Ahmad*, J. K. Murphy, P. M. Langdon, S. J. Godsill are with the Engineering Department, University of Cambridge, Trumpington Street, Cambridge, UK, CB2 1PZ. Emails:{bia23, jm362, pml24, sjg30}@cam.ac.uk.
[†]The first two authors made an equal contribution to this work.

To motivate the work presented here, consider the following two examples:

1) Maritime Surveillance: analysing the route and determining the destination of vessels in a given geographical area is necessary for maintaining maritime situational awareness (MSA), critical for maritime safety. This permits identification of potential threats, opportunities or malicious behaviour, allowing the protection of assets or other reactive actions [1], [2], [8], [9]. Given the complex nature of typical maritime traffic as well as the vast amounts of available information on tracked targets, there is a growing interest in increasing the degree of automation in MSA systems by unveiling the intent of object(s) of interest from available low level tracking data. Thus, there is a notable demand for low-complexity and reliable destination and trajectory prediction techniques. Similar challenges can be found in general surveillance applications, including aerospace, with aircrafts in lieu of vessels in MSA.

2) Interacting with touchscreen: touchscreens are becoming an integrated part of modern vehicles due to their ability to present large quantities of in-vehicle infotainment system data and offering additional design flexibility through a combined display-input-feedback module [13], [14]. Using these displays entails undertaking a free hand pointing gesture and dedicating a considerable amount of attention that would otherwise be available for driving. Hence, such interactions can act as a distractor from the primary task of driving and have safety implications [15]. The early inference of the intended on-screen item of the free hand pointing gesture can simplify and expedite the selection task, thus improving the usability of in-vehicle touchscreens by reducing distractions (see [16] for an overview of the intent-aware display concept). For example, assuming that the endpoint prediction certainty meets a set criterion, the user need not touch the display surface to select an item, allowing mid-air selection. This can significantly reduce the effort (attention) associated with interacting with in-car displays as per the user study in [17].

There is a wide range of other applications that can benefit from knowing the intent of a target of interest. For instance, predicting the destination of a pedestrian [18], e.g. an intruder in a perimeter, intelligent robot navigation in general or in the presence of other moving agents such as people [19]–[24], and advanced driver assistance systems [25], to name a few.

A. Contributions

The main contribution of this paper is the development of a simple and robust Bayesian intent inference approach that models the available partial trajectory of a tracked object as a random bridge terminating at a nominal destination. It can

employ any linear Gaussian motion model, including *Linear Destination Reverting* (LDR) models, such as those detailed in Sections III-A and III-B, which are intrinsically driven by the endpoint of the tracked object [16]. The bridging framework introduced here capitalises on the premise that the path of the object, albeit random, must end at the intended destination. Since the endpoint is unknown *a priori*, a bridge for each possible destination is constructed. This encapsulates the long term dependencies in the object trajectory due to premeditated actions guided by intent. By determining the likelihood of the observed partial track being drawn from a particular bridge, the probability of each nominal endpoint, along with a refined estimate of the current object state X_t and its future values X_s (for $s > t$), can be evaluated. This is accomplished via a Kalman-filter-based inference, amenable to parallelisation.

Notably, the proposed approach in this paper does not impose prior knowledge of the arrival time T at the intended destination D. A conservatively chosen prior distribution of the possible times of arrival for each destination suffices. This allows the introduction of a technique to sequentially estimate the posterior distribution $p(T|D)$ of the arrival time T from the available partial trajectory of the tracked object. Within the formulation adopted here, possible destinations may also be specified as (Gaussian) distributions, with a mean and covariance, in order to model endpoints with a non-zero spatial extent; point destinations can be modelled by setting the covariance to zero. Finally, we conduct simulations to illustrate the inference capability of the bridging-distribution-based predictors using real pointing data (HCI) and synthetic vessel tracks (MSA).

B. Related Work

One of the first techniques to incorporate predictive information on the target endpoint to improve the accuracy of the tracking results was proposed in [1] and motivated various subsequent studies such as [3]. It assumes prior knowledge of the time of arrival at destination to devise a destination-aware tracker. In this paper, the objective is to predict the intended destination of the tracked object, using motion models that are also inherently dependent on the endpoint. This can be inferred using a multiple Kalman-filters-based solution, even without imposing knowledge of T. It is noted that unlike the widely used interacting multiple models (IMM) [11], [26] and generalised pseudo-Bayesian [27] approaches for manoeuvering targets, here we construct a bridge model per nominal destination and no interaction or switching among models is applied. This is based on the premise that the tracked object intent is set well in advance of reaching its endpoint, leading to simple and low complexity algorithms.

More recently, destination-aware trackers that facilitate inference of the object state X_t followed by an additional mechanism to determine its endpoint are presented in [5]–[7], succeeding the work in [28]. The object trajectory is modelled as a discrete stochastic reciprocal or context-free grammar process, which can be regarded as non-causal generalisations of Markov processes. The state space is discretised within predefined regions, for example, the spatial dimensions are divided into finite grids. The target can accordingly pass through a finite number of zones. In this paper, in contrast, we adopt continuous state space models with bridging distributions that do not impose any discretisation/restrictions on the path the tracked object has to follow to reach its endpoint. This formulation is particularly important in applications where discretisation of the spatial area is burdensome, e.g. tracking objects in 3D as with free hand pointing gestures or surveying a large geographical area for MSA; it can also easily handle noisy asynchronous measurements. The method proposed here provides a simple, effective solution to the intent prediction problem compared with those in [6], [7]; it also combines the destination prediction and tracking operations.

Intent inference is often treated within the context of anomaly detection, e.g. [2], [4], [6]–[9], [29]. A common approach is to categorise each trajectory of the tracked object(s) as either normal or anomalous, i.e. classification techniques. This follows defining (or learning from recorded data) a *pattern of life* that constitutes ordinary behaviour. Deviations from this are considered to indicate an anomaly, assuming adequate data association algorithms [2], [8]. For example, a support vector machine (SVM) based method is introduced in [5] to classify deviant trajectories. A model-based technique, using a tuned hidden Markov model, is utilised in [9] to characterise a *pattern of life* and predict the future position of a conforming target. Similarly, in [6] and related work, 'tracklets', which are sub-patterns comprising a trajectory, are defined as a means to capture a semantic interpretation of complex patterns such as anomaly or intent. In this paper, a probabilistic model-based formulation is proposed to tackle the intent inference problem rather than anomaly detection, with the posterior probabilities of various intents sequentially calculated using bridging distributions.

The benefits of predicting the intended item on a Graphical User Interface (GUI) early in a pointing task are widely recognised in HCI, e.g. [30]–[33]. Most existing algorithms focus on pointing via a mechanical device, such as a mouse, on a computer screen in a 2D set-up. In [16], 2D–based predictors are shown to be unsuitable for pointing tasks in 3D. For instance, the linear-regression methods in [31] and [32] assume that the the destination is always located along the path followed by the pointing object, which is rarely true in free hand pointing gestures. Endpoint inference based on modelling the pointing movements as a linear destination reverting process is considered in [16]. Compared to [16], the bridging-based-solution developed here is more general and more robust to variability in the target behaviour, leading to superior prediction results.

Finally, data driven prediction/classification techniques, such as in [18]–[23], [25], [33], rely on a dynamical model and/or a representation of the environment (e.g. including physical constraints) learnt from recorded tracks/behaviour. For example, predictors based on inverse-optimal control are introduced in [18], [33]. Such methods often entail a high computational cost and necessitates substantial parameters training from complete data sets (not always available). In this paper, as is common in tracking applications [10]–[12], known dynamical and sensor models (e.g. due to practical physical limitations), albeit with unknown parameters, are presumed,

i.e. a state-space-modelling approach. The solution proposed here requires minimal training, is amenable to parallelisation and is computationally efficient, yet delivers a competitive performance.

C. Paper Outline

The remainder of this paper is organised as follows. In Section II, we formulate the tackled problem and define the objectives. A range of possible motion models are outlined in Section III and the prior of the state value at destination, i.e. X_T, is addressed. In Section IV, the bridging-based-prediction is introduced and pseudo-code for the proposed algorithms is provided. The performance of the proposed techniques is evaluated in Section V using real free hand pointing gesture data and synthetic vessel tracks. Finally, conclusions are drawn in Section VI.

II. PROBLEM STATEMENT

Let $\mathbb{D} = \{d : d = 1, ..., N\}$ be the set of N nominal destinations, e.g. harbours where a vessel can dock or selectable icons displayed on a touchscreen. The time instant the tracked object reaches the *a priori* unknown intended destination D is denoted by T. Whilst no assumptions are made about the layout of $\mathbb{D}$, each endpoint is modelled as an extended region, e.g. GUI icons or harbour, rather than a single point as in [16]. Hence, the dth destination is defined by the (Gaussian) distribution $d \backsim \mathcal{N}(a_d, \Sigma_d)$; see Section III-C.

The objective here is to dynamically determine the probability of each possible endpoint being the intended destination,

$$\mathcal{P}(t_n) = \{p(D = d \mid y_{1:n}) : d = 1, 2, ..., N\} \quad (1)$$

where $y_{1:n} \triangleq \{y_1, y_2, ..., y_n\}$ is the available partial trajectory or observations of the tracked object at the time instant $t_n \leqslant T$. For example, as in [16], we have $y_k = [\hat{x}_{t_k} \quad \hat{y}_{t_k} \quad \hat{z}_{t_k}]'$ is the Cartesian coordinates of the pointing finger at t_k as captured by the pointing-gesture tracking device. Each observation is assumed to be derived from a true, but unknown, underlying target state X_{t_n} at time t_n, which can include position, velocity and higher order kinematics. Given $\mathcal{P}(t_n)$, the Maximum *a Posteriori* (MAP) estimate

$$\hat{D}(t_n) = \underset{d=1,2,...,N}{\arg\max} \; p(D = d \mid y_{1:n}), \quad (2)$$

is an intuitive approach to determine the intended endpoint D following the arrival of the new observation y_n. It should be noted that other decision criteria can be adopted, although this is not treated here. Therefore, the proposed probabilistic endpoint prediction relies on a belief-based inference, i.e. calculating $\mathcal{P}(t_n)$ in (1), followed by a classifier, e.g. (2).

As well as establishing D at time t_n, a refined estimate of the system latent state X_{t_n} and its predicted future values, i.e. X_{t^*} for $t^* > t_n$, and time of arrival at D are sought. A successful prediction at t_n of the tracked object's final destination and/or its future trajectory can reveal its intention, alerting an operator $T - t_n$ in advance of any potential conflict or reducing the pointing time by $T - t_n$ in an HCI context.

III. SYSTEM MODELS

A linear and Gaussian motion model for the evolution of the physical state X_t of the tracked object is assumed throughout this paper. Whilst the system governing the target dynamics does not change over time, it does depend on the object eventually reaching its destination $D \in \mathbb{D}$. Conditioned on knowing the endpoint $D = d$, this leads to a linear time-invariant Gaussian system such that the relationship between the system state at times t and $t + h$ can be written as

$$X_{t+h} = F(h, d)X_t + M(h, d) + \varepsilon_t \quad (3)$$

with $\varepsilon_t \sim \mathcal{N}(0, Q(h, d))$. The matrices F and Q as well as the vector M, which together define the state transition from one time to another, are functions of the time step h and the destination d.

The nth observation y_n is modelled as a linear function of the time t_n state perturbed by additive Gaussian noise,

$$y_n = GX_{t_n} + \nu_n \quad (4)$$

where $\nu_n \sim \mathcal{N}(0, V_n)$. No assumption is made about the observation arrival times t_n and irregular, asynchronous observations can naturally be incorporated within the framework presented here. Based on (3) and (4), a Kalman filter can be utilised to calculate both the posterior distribution of the latent state and the observation likelihood for the current set of measurements $y_{1:n}$ [34], conditioned on knowing d. The computationally efficient Kalman filter is particularly desirable since running, concurrently, multiple Kalman filters, in real-time, is plausible, even in settings where limited computing power is available, such as on a vehicle touchscreen or a portable battery-powered system.

To condition on the destination, the inference framework requires that the transition density of such models can be calculated both from one observation to the next (i.e. from time instant t_{n-1} to t_n) and from the current observation time to the arrival time at the intended endpoint (i.e. from t_n to T). Continuous-time motion models are therefore a natural choice, where the tracked object dynamics is represented by a continuous-time stochastic differential equation (SDE). This SDE can be integrated to obtain a transition density over any time interval. For Gaussian linear time invariant (LTI) models, this integration is analytically tractable, giving transition functions of the form in (3). This class of models includes many widely used ones, e.g. (near) constant velocity (CV) and (near) constant acceleration (CA) models, as well as the linear destination reverting models given below.

The early work in [16] employs LDR models in a forward sense, without using the destination state as conditioning information. In this paper, by contrast, a prior probability distribution on the object state at its destination X_T is used directly as conditioning information in the inference procedure. This is achieved by using bridging distributions to introduce the longer term dependencies in the target trajectory (see Section III-C). A similar idea is explored in the preliminary study in [35] using the CV model, in which the endpoint information does not feature in the motion model. That work also specifies each destination as a single point (rather than
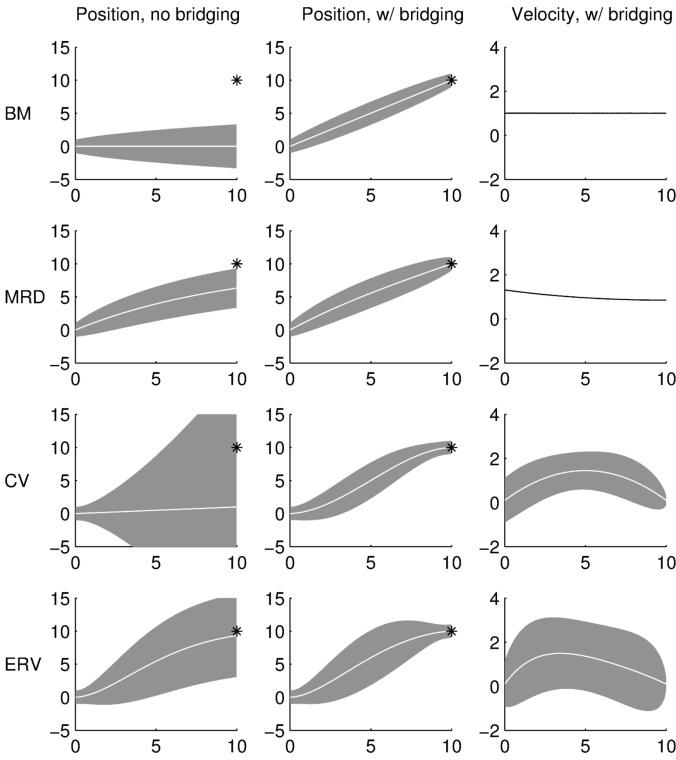
Figure 1: Predictive distribution calculated at $t = 0$ of position without bridging (first column), predictive position with bridging (second column), and predictive velocity with bridging (third column) for the BM, MRD, CV and ERV models in 1D. The x-axis shows time t and y-axis depicts position/velocity. The destination (black star), is located at position 10 at time $T = 10$. Grey shading shows one standard deviation, white line depicts the mean (except for BM and MRD velocity, for which the black line shows the implied mean of velocity only). Parameters for the models are $\sigma = 1$ (all models), $\lambda = 0.3$ (MRD), $\eta = 0.1, \rho = 0.5$ (ERV).

a prior distribution) and applies a two step modified Kalman filter to infer D, which is more computationally demanding and obscure compared with the solution presented here.

In the remainder of this section, we describe the LDR models of [16] for completion, other related motion processes and the prior on X_T. The predictive position and velocity distributions of selected motion models, with and without the use of bridging (i.e. conditioning on X_T) are depicted in Figure 1 in the one dimensional case. The figure demonstrates the substantial effect of introducing the bridging assumptions on the prediction results of various motion models. It clearly shows how these distributions more accurately model the predicted state at the endpoint. It is worth noting that any continuous-time LTI model could be used directly within the proposed inference framework, as long as the system can be expressed by equations (3) and (4).

A. Mean Reverting Diffusion (MRD)

The position of the tracked object for the Mean Reverting Diffusion (MRD) model follows an Ornstein-Uhlenbeck process [36], with its mean being the destination. The intuition behind MRD is that a target heading towards a particular endpoint will revert towards it. The reversion strength is

stronger the further the target is from its destination. This produces the predictive position and velocity dynamics of the form displayed in the second row of Figure 1.

The state X_t of the mean reverting diffusion model consists of only the target position in each of the s spatial dimensions. For endpoint d (located at position p_d), the SDE is given by

$$dX_t = \Lambda \left(p_d - X_t \right) dt + \sigma dw_t, \tag{5}$$

where Λ is a diagonal matrix with diagonal elements $\{\lambda_i\}_{i=1}^{s}$ that set the reversion coefficient in each dimension. The diagonal matrix σ specifies the standard deviation of the dynamic noise and w_t is a standard (unit variance) s-dimensional Wiener process. The diagonal structure of σ implies that the noise is independent in each spatial dimension, which is a common assumption in tracking [10]. However, this can be relaxed as shown in equation (7) below.

As in [16], by integrating (5) from t to $t + h$, we obtain a state transition function in the form of equation (3) with

$$
\begin{aligned}
F_{\text{MRD}}(h, d) &= e^{-\Lambda h}, \\
M_{\text{MRD}}(h, d) &= (I_s - e^{-\Lambda h}) p_d, \\
Q_{\text{MRD}}(h, d) &= \frac{1}{2} \left[I_s - e^{-2\Lambda h} \right] \Lambda^{-1} \sigma^2,
\end{aligned}
\tag{6}
$$

where I_s is the $s \times s$ identity matrix. For non-diagonal σ, the $(i, j)^{\text{th}}$ element of Q_{MRD} is given by

$$Q_{\text{MRD,ij}} = \frac{(\boldsymbol{\sigma}\boldsymbol{\sigma}')_{ij}}{\Lambda_{ii} + \Lambda_{jj}} \left[1 - e^{-(\Lambda_{ii} + \Lambda_{jj})h} \right], \tag{7}$$

where z' is the transpose of the vector/matrix z. A special case of the MRD model occurs when $\Lambda = 0_s$ (with 0_s being the $s \times s$ matrix of zeros), in which case the dynamics of the target position follow a Brownian motion (BM). In this case, the F, M and Q matrices in (3) become

$$
\begin{aligned}
F_{\text{BM}}(h, d) &= h I_s, \\
M_{\text{BM}}(h, d) &= 0_{s \times 1}, \\
Q_{\text{BM}}(h, d) &= h \sigma^2,
\end{aligned}
\tag{8}
$$

where $0_{s \times 1}$ is an $s \times 1$ column vector of zeros.

If measurements y_n are direct noisy observations of the tracked object position, the observation matrix G in equation (4) is simply the identity matrix, $G = I_s$.

B. Equilibrium Reverting Velocity (ERV)

In the ERV model, introduced in [16] for tracking and destination inference for pointing tasks on in-car displays, the system state X_t at time t is a $2s \times 1$ vector, arranged as $[x_1, ..., x_s, \dot{x}_1, ..., \dot{x}_s]'$, where x_i is the position in spatial dimension i, and $\dot{x}_i$ is the velocity in that dimension. Under this model, the SDE governing the evolution of the tracked object state is

$$dX_t = A \left(\mu_d - X_t \right) dt + \sigma dw_t, \tag{9}$$

where the mean $\mu_d = [p_d', 0_{s \times 1}']'$ contains the position p_d of destination d; w_t is a standard s dimensional Wiener process and $B = [0_s, \sigma]$ is a $2s \times s$ matrix that controls the noise which affects the velocity components of the process (modelling

random forces acting on the target). In this case, σ can be given by the $s \times s$ Cholesky decomposition of the velocity noise covariance Σ, such that $\Sigma = \sigma\sigma'$. As with the MRD model, a common choice is a diagonal σ, i.e. independent noise in each spatial dimension. The matrix A is given by

$$A = \begin{bmatrix} 0_s & -I_s \\ \eta & \rho \end{bmatrix},$$

where ρ is an $s \times s$ diagonal matrix of the drag coefficients (can be assumed to be the same across all dimensions) and η is a $s \times s$ diagonal matrix of the mean reversion strengths in each spatial dimension.

A physical interpretation of the ERV model is that the destination d exerts an attractive force on the tracked object, of strength proportional to the distance separating them. This can be viewed as a linear spring of zero natural length connecting the target to its endpoint. A drag term proportional to the velocity of the target is also included, allowing the velocity profile of the tracked object (e.g. pointing-finger-tip in a free hand pointing gesture) to be correctly modelled; see Figure 1 (fourth row) and [16].

Integrating the SDE in (9) from t to $t+h$ allows the system evolution to be expressed in the form in equation (3) with

$$\begin{aligned} F_{\text{ERV}}(h, d) &= e^{-Ah}, \\ M_{\text{ERV}}(h, d) &= (I_{2s} - e^{-Ah})\mu_d, \\ Q_{\text{ERV}}(h, d) &= \int_t^{t+h} e^{-A(t+h-v)}\sigma\sigma' e^{-A'(t+h-v)}dv. \end{aligned} \quad (10)$$

The covariance matrix $Q_{\text{ERV}}(h, d)$ can be calculated by Matrix Fraction Decomposition [37] where

$$Q_{\text{ERV}}(h, d) = JK^{-1},$$

$$\begin{bmatrix} J \\ K \end{bmatrix} = \exp\left(\begin{bmatrix} -A & \sigma\sigma' \\ 0_{2s} & A' \end{bmatrix} h\right) \begin{bmatrix} 0_{2s} \\ I_{2s} \end{bmatrix}.$$

A special case of the ERV model occurs when η and ρ are both zero. In this scenario, the model reduces to the widely used (near) constant velocity model with

$$\begin{aligned} F_{\text{CV}}(h, d) &= \begin{bmatrix} I_s & hI_s \\ 0_s & I_s \end{bmatrix}, \\ M_{\text{CV}}(h, d) &= 0_{2s \times 1}, \\ Q_{\text{CV}}(h, d) &= \begin{bmatrix} \sigma\sigma'\frac{h^3}{3} & \sigma\sigma'\frac{h^2}{2} \\ \sigma\sigma'\frac{h^2}{2} & \sigma\sigma'h \end{bmatrix}. \end{aligned} \quad (11)$$

If observations y_n are direct noisy measurements of the target position, the observation matrix G in equation (4) is given by $G = \begin{bmatrix} I_s & 0_s \end{bmatrix}$.

C. State Value Prior at Destination

Knowing the intended destination D of a tracked object gives information about the system state at some future time T. This can be modelled by a prior probability distribution for X_T corresponding to the geometry of the destination, since most endpoints are extended regions (e.g. GUI buttons or harbours), rather than single points. In order to maintain the

linear Gaussian structure of the system, the following Gaussian prior on the object state upon arrival at destination at time T:

$$p(X_T \mid D = d) = \mathcal{N}(X_T; a_d, \Sigma_d), \quad (12)$$

is assumed[1]. This effectively models the destination as ellipsoidal, since the iso-probability surfaces of the Gaussian distribution are elliptical. The mean vector a_d specifies the location/centre of the destination, for example, $a_d = p_d$ for the MRD model and $a_d = \mu_d$ from equation (9) for the ERV model. Whereas, Σ_d is a covariance matrix of the appropriate dimension, which sets the extent and orientation of the endpoint. In the case of the ERV model, defining the destination also entails specifying a distribution of the tracked object velocity at the destination. If this is unknown, a large prior variance can be used to model this uncertainty.

IV. INTENT INFERENCE USING BRIDGING DISTRIBUTIONS

For motion models of the form in equation (3), and conditioning on a given destination $d \in \mathbb{D}$ as well as arrival time T, the posterior of the system state can be expressed by $p(X_{t_n} \mid y_{1:n}, T, D = d)$ and the observation likelihood is $p(y_{1:n} \mid T, D = d)$ after n measurements. The graphical structure of this system is depicted in Figure 2. The intended destination D influences the state at all times via the reversion built into the LDR motion models (for non-destination reverting ones, the endpoint only affects the final state). Nonetheless, the inclusion of the prior on X_T in (12) changes the system dynamics, even for the MRD and ERV models. Assuming that the destination is known, the posterior distribution of the state changes from a random walk to a bridging distribution. This is clearly visible in Figure 1, especially in terms of producing consistent and meaningful predictions. Consequently, all Gaussian linear models, including the non-destination reverting ones, whose dynamic models are not intrinsically dependent on an endpoint, e.g. BM and CV, can be used for destination prediction within the proposed formulation.

A. Filtering and Likelihood Calculation

An elegant way to filter for X_t with a destination prior is to extend the latent system state to incorporate X_T, thus, it becomes $Z_t = [X_t' \ X_T']'$. Filtering is then carried out for $p(Z_{t_n} \mid y_{1:n}, D = d, T)$, permitting the calculation of the observation likelihood $p(y_{1:n} \mid D = d, T)$. The key to developing this filter is to calculate the transition density $p(Z_{t+h} \mid Z_t, D = d, T)$, given by

$$\begin{aligned} p(Z_{t+h} \mid Z_t, D = d, T) &= p(X_T \mid X_{t+h}, X_T, D = d, T) \\ &\quad \times p(X_{t+h} \mid X_t, X_T, D = d, T) \\ &= p(X_{t+h} \mid X_t, X_T, D = d, T). \end{aligned} \quad (13)$$

---

[1] Note that this *artificial* prior encourages the state X_T to be close to one of the desired destinations d, as shown in Figure 1. Whilst this can be viewed as constraining the system terminal state conditioned on a known endpoint, it is in general inconsistent with the underlying dynamical model in (3) since (3) implies that $p(X_T|D = d) = \int p(X_{t_1}|D = d)p(X_T|X_{t_1}, D = d)dx_{t_1}$. For further details on this construct, see the last part of Section IV-A, below equation (21).
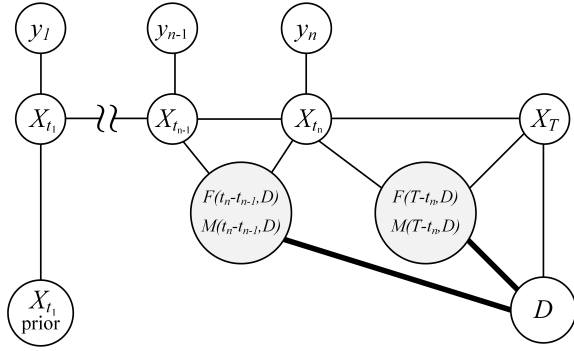
Figure 2: The graphical structure of the system after n observations. The destination plays a similar role to the prior distribution of X_{t_1} in addition to affecting the state transition function. Heavy lines indicate a deterministic relationship (early transition matrices are not shown).

This follows from $Pr(X_T = x \mid X_{t+h}, X_T, D = d, T) = 1$, for any value of X_T, due to the fact that the state X_T is included in the conditioning information via Z_t. The distribution $p(X_{t+h} \mid X_t, X_T, D = d, T)$ is given by

$$p(X_{t+h} \mid X_t, X_T, D = d, T) \propto p(X_T \mid X_{t+h}, D = d, T)$$
$$\times p(X_{t+h} \mid X_t, D = d, T)$$
$$= \mathcal{N}\left(X_T; F_x X_{t+h} + M_x, Q_x\right) \mathcal{N}\left(X_{t+h}; F_h X_t + M_h, Q_h\right)$$
$$(14)$$

where the F, M and Q matrices are taken from the motion models in Section III, with

$$
\begin{aligned}
F_x &= F(T - t - h, D) & F_h &= F(h, D), \\
M_x &= M(T - t - h, D) & M_h &= M(h, D), \\
Q_x &= Q(T - t - h, D) & Q_h &= Q(h, D).
\end{aligned}
\tag{15}
$$

This comes from the system structure in Figure 2, and the fact that for the continuous-time integrable models used here, the state transition density can be calculated over any time period.

The following Gaussian identity

$$\mathcal{N}\left(x; \mu_1, \Sigma_1\right) \mathcal{N}\left(\mu_2; Lx, \Sigma_2\right) = z \mathcal{N}\left(x; \mu_*, \Sigma_*\right), \tag{16}$$

simplifies the state transition density in equation (14) where L is a matrix of appropriate size, z is a normalizing constant that does not depend on x, and

$$
\begin{aligned}
\Sigma_* &= \left(\Sigma_1^{-1} + L' \Sigma_2^{-1} L\right)^{-1}, \\
\mu_* &= \Sigma_* \left(\Sigma_1^{-1} \mu_1 + L' \Sigma_2^{-1} \mu_2\right).
\end{aligned}
$$

This leads to

$$p(X_{t+h} \mid X_t, X_T, T, D = d) = \mathcal{N}\left(X_{t+h}; c_t, C_t\right),$$

where

$$
\begin{aligned}
C_t &= (Q_h^{-1} + F_x' Q_x^{-1} F_x)^{-1}, \\
c_t &= C_t \left[Q_h^{-1}(F_h X_t + M_h) + F_x' Q_x^{-1}(X_T - M_x)\right], \quad (17) \\
&= H_t Z_t + m_t,
\end{aligned}
$$

with, for an r-dimensional state vector X_t, H_t a $r \times 2r$ matrix

and m_t a $r \times 1$ vector such that

$$
\begin{aligned}
H_t &= [C_t Q_h^{-1} F_h, \ C_t F_x' Q_x^{-1}], \\
m_t &= C_t (Q_h^{-1} M_h - F_x' Q_x^{-1} M_x).
\end{aligned}
$$

This allows the state transition with respect to Z_t to be written as a linear Gaussian transition of the form

$$Z_{t+h} = R_t Z_t + \tilde{m}_t + \gamma_t, \tag{18}$$
$$\gamma_t \sim \mathcal{N}\left(0, U_t\right).$$

where

$$R_t = \begin{bmatrix} H_t \\ P_T \end{bmatrix}, \quad \tilde{m}_t = \begin{bmatrix} m_t \\ 0_r \end{bmatrix}, \quad U_t = \begin{bmatrix} C_t & 0_r \\ 0_r & 0_r \end{bmatrix}, \tag{19}$$

and $P_T = \begin{bmatrix} 0_r & I_r \end{bmatrix}$. With respect to this extended system, the k-dimensional observation vector at each observation time is given by

$$y_n = \tilde{G} Z_{t_n} + \nu_n \tag{20}$$

with $\tilde{G} = [G, \ 0_{k \times r}]$ and where G and ν_n are as those in equation (4).

Algorithm 1 Conditioned Prediction Error Decomposition and State Inference - Kalman Filter (single iteration)

Notation: $\{\ell_n, \hat{Z}_n, \Sigma_n\} = \mathrm{KF}(y_n, \hat{Z}_{n-1}, \Sigma_{n-1}, R_t, U_t, \tilde{G})$
Input: Observation y_n; previous posterior state mean estimate† $\hat{Z}_{n-1}$; previous posterior state covariance† Σ_{n-1}; state transition matrix R_t and covariance U_t from equation (18); observation matrix $\tilde{G}$ from equation (20)
Predict†:
    $\hat{Z}_{n|n-1} = R_t \hat{Z}_{n-1} + \tilde{m}_t$
    $\Sigma_{n|n-1} = R_t \Sigma_{n-1} R_t' + U_t$
PED Calculation:
    $\ell_n = \mathcal{N}\left(y_n; \tilde{G}\hat{Z}_{n|n-1}, \tilde{G}\Sigma_{n|n-1}\tilde{G}' + V_n\right)$
Correct:
    $K = \Sigma_{n|n-1}\tilde{G}'(\tilde{G}\Sigma_{n|n-1}\tilde{G}' + V_n)^{-1}$
    $\hat{Z}_n = \hat{Z}_{n|n-1} + K(y_n - \tilde{G}\hat{Z}_{n|n-1})$
    $\Sigma_n = (I_r - K\tilde{G})\Sigma_{n|n-1}$
Output: PED $\ell_n = p(y_n \mid y_{1:n-1}, D = d, T)$; posterior mean of state at time t_n, $\hat{Z}_n$; posterior state covariance Σ_{n-1}.
† For first observation y_1, skip predict step and use prior mean and covariance for Z_{t_1} from equation (21) in place of $\hat{Z}_{n|n-1}$ and $\Sigma_{n|n-1}$ in likelihood calculation and correction steps.

Equations (18) and (20) form a standard linear Gaussian system, albeit with a degenerate state transition covariance matrix. Thus, a standard Kalman filter can be applied to calculate the conditioned posterior filtering distribution of the state and the conditioned prediction error decomposition (PED), $p(y_n \mid y_{1:n-1}, D = d, T)$. This latter is shown in Sections IV-C to IV-E to be key to destination inference, because it allows the destination and end-time conditioned observation likelihood of the partially observed track to be

calculated recursively as:

$$p(y_{1:n} \mid D = d, T) = p(y_n \mid y_{1:n-1}, D = d, T)$$
$$\times p(y_{1:n-1} \mid D = d, T).$$

The algorithm for a single iteration of the Kalman filter is given in Algorithm 1. It requires the prior $Z_{t_1} = [X'_{t_1}, X'_T]'$ at the first observation time. Whilst the prior on X_{t_1} is the standard prior on the initial state, the prior on X_T is derived from the destination as described in Section III-C, and can be neatly incorporated into the Kalman filter. It should be noted that technically the prior on X_T is linked to X_{t_1} according to the transition density $p(X_T|X_{t_1}, D = d, T)$. However, in our formulation, we introduce an *artificial* prior distribution on X_T that guides the trajectory towards a particular destination $D = d$ at time T. This leads to

$$p(Z_{t_1} \mid T, D = d) = \mathcal{N}\left(\begin{bmatrix} X_{t_1} \\ X_T \end{bmatrix}; \begin{bmatrix} \mu_1 \\ a_d \end{bmatrix}, \begin{bmatrix} \Sigma_1 & 0_r \\ 0_r & \Sigma_d \end{bmatrix} \right), \quad (21)$$

where μ_1 and Σ_1 specify the initial prior on X_{t_1}, thereby $p(X_{t_1} \mid D = d) = \mathcal{N}(X_{t_1}; \mu_1, \Sigma_1)$; a_d and Σ_d come from the endpoint prior in equation (12). In this form, we assume that X_T and X_{t_1} are independent *a priori*, which is approximately true in many of the dynamical models we consider. Nonetheless, if we wished to capture the prior dependence structure between X_T and X_{t_1}, because of a more "informative" dynamical model, we could instead use

$$p(Z_{t_1} \mid T, D = d) \propto \mathcal{N}\left(\begin{bmatrix} X_{t_1} \\ X_T \end{bmatrix}; \begin{bmatrix} \mu_1 \\ a_d \end{bmatrix}, \begin{bmatrix} \Sigma_1 & 0_r \\ 0_r & \Sigma_d \end{bmatrix} \right)$$
$$\times p(X_T|X_{t_1}, D, T). \quad (22)$$

Although incorporation of (22) is straightforward for the linear Gaussian models studied in this paper, it is not explored here.

An alternative interpretation of our construct as per equation (12) is to consider the possible destinations at a_d as *pseudo-observations* rather than prior means, where $d = 1, 2, ..., N$. In this case, we write an observation density for the pseudo-observation $\tilde{y}_T$ as $p(\tilde{y}_T = a_d|X_T) = \mathcal{N}(a_d; X_T, \Sigma_d)$ and proceed to compute the likelihoods $p(y_{1:k}|\tilde{y}_T = a_d, D = d, T)$ via the Kalman filter. These likelihood functions then serve to determine which pseudo-observation is most consistent with the data seen, i.e. $y_{1:n}$. The calculations are exactly the same as under the prior modelling formulation adopted here; for further discussion of this point, see [38].

B. Unknown Arrival Times

So far, it has been assumed that the arrival time at the destination T is known *a priori*. Often, this is not a realistic assumption and we are interested in the probability of the tracked object arriving at a destination at *any* time within some time interval. In this case, the unknown arrival time T is treated as a random variable, which must be integrated over in order to infer the destination of the tracked object.

The observation likelihood with unknown arrival time is given by $p(y_{1:n} \mid D)$. This can be calculated by integrating the arrival-time-conditioned likelihood calculated by the Kalman filter in the preceding section (see Algorithm 1) over all possible arrival times according to

$$p(y_{1:n} \mid D) = \int_{T \in \mathcal{T}} p(y_{1:n} \mid T, D)p(T \mid D)dT, \quad (23)$$

where $p(T \mid D)$ is the a prior distribution of arrival times for destination D and $\mathcal{T}$ is the time interval of possible arrival times T ($D = d$ is here replaced by D for notational brevity). For example, arrivals might be expected uniformly within some time period $[t_a, t_b]$, giving $p(T \mid D) = \mathcal{U}(t_a, t_b)$.

In most cases, $p(y_{1:n} \mid T, D)$ after n observations is a nontrivial function of T resulting in intractable integrals. A numerical approximation to the integral in equation (23) can be obtained via numerical quadrature, which is viable since the arrival time is a one-dimensional quantity. The approximation requires multiple evaluations of the arrival-time-conditioned-observation-likelihood (for various arrival times T) for each of the N nominal endpoints. Therefore, in time-sensitive applications, it is likely that only a few quadrature points q can be used.

Whilst adaptive quadrature schemes might seem appealing for their efficiency and accuracy, they are not easily applied to this problem. For a given arrival time, only a single iteration of the Kalman filter as in Algorithm 1 need be run (per destination $d \in \mathbb{D}$), following the arrival of a new observation. Thus, if a fixed quadrature is utilised with q quadrature points, Nq Kalman filter iterations of the form in Algorithm 1 are required per observation arrival. For adaptive quadrature, it will, in general, be necessary to calculate the observation likelihood for a different set of arrival times at each step. This, however, requires the Kalman filter be re-run from scratch for all n available observations, imposing nNq iterations. Hence, simple fixed grid quadrature schemes are employed here. A Kalman filter iteration as in Algorithm 1 is not computationally intensive and each of the Nq iterations, required per observation, can be performed in parallel. For arrival times before the current observation time $T_i < t_n$, the arrival-time-conditioned observation likelihood, i.e. $p(y_{1:n} \mid T = T_i < t_n, D)$, is taken to have zero values.

Here, we apply a Simpson's rule quadrature scheme, with an odd number q of evenly spaced quadrature points, $T_1, ..., T_q$. This approximates the integral in equation (23) by

$$p(y_{1:n} \mid D) \approx \frac{T_q - T_1}{3(q-1)} \Bigg[p(y_{1:n} \mid T = T_1, D)$$
$$+ p(y_{1:n} \mid T = T_q, D) + 4 \sum_{i=1}^{(q-1)/2} p(y_{1:n} \mid T = T_{2i}, D)$$
$$+ 2 \sum_{i=1}^{(q-1)/2-1} p(y_{1:n} \mid T = T_{2i+1}, D) \Bigg]. \quad (24)$$

Other numerical integration schemes such as Gaussian quadrature can be employed. It may also be desirable to use unequal intervals between quadrature points, e.g. if the distribution of arrival times is heavily peaked in particular regions.

C. Destination Inference

The posterior distribution of the nominal destinations, i.e. $p(D = d \mid y_{1:n}), d \in \mathbb{D}$ in equation (1), at the time instant t_n can be expressed via Bayes' theorem by

$$p(D = d \mid y_{1:n}) \propto p(y_{1:n} \mid D = d)p(D = d). \qquad (25)$$

The discrete probability distribution $p(D = d)$ defines a prior over all possible destinations; it is independent of the current track and can be obtained from contextual information, historical data, etc. Alternatively, a non-informative prior can be used where $Pr(D = d) = 1/N$ for all $d \in \mathbb{D}$.

Algorithm 2 shows how the posterior distribution over destinations $p(D = d \mid y_{1:n})$ in (1) can be inferred sequentially given a series of observations $y_{1:n}$. It begins by initializing the running likelihood estimate $L_0^{(d)}$ for each $d \in \mathbb{D}$ and the current posterior state mean $\hat{Z}_0^{(d,i)}$ as well as covariance $\Sigma_0^{(d,i)}$ for each destination d and quadrature point i (corresponding to arrival time T_i) to their priors. After each observation, the Kalman filter iteration in Algorithm 1 is utilised to calculate the one step arrival-time-conditioned-observation PED $\ell_n^{(d,i)} = p(y_n \mid y_{1:n-1}, T = T_i, D = d)$ for each destination $d \in \mathbb{D}$ and quadrature point i. This is then used to calculate the overall arrival-time-conditioned observation likelihood

$$L_n^{(d,i)} = p(y_{1:n} \mid T = T_i, D = d)$$
$$= p(y_n \mid y_{1:n-1}, T = T_i, D = d)p(y_{1:n-1} \mid T = T_i, D = d)$$
$$= \ell_n^{(d,i)} \times L_{n-1}^{(d,i)}. \qquad (26)$$

The Kalman iteration also determines the corresponding updated posterior state mean $\hat{Z}_n^{(d,i)}$ and covariance $\Sigma_n^{(d,i)}$, which are necessary for the next steps.

After running a Kalman iteration for all quadrature points, the quadrature function (and arrival time prior) is used on the likelihood $L_n^{(d,i)}$ calculated at each of these points for a destination d in order to approximate the integral in (23). This gives the observation likelihood $p(y_{1:n} \mid D = d)$ for each destination $d \in \mathbb{D}$,

$$p(y_{1:n} \mid D = d) \approx \text{quad}(L_n^{(d,1)}, L_n^{(d,2)}, ..., L_n^{(d,q)}). \qquad (27)$$

For a finite set $\mathbb{D}$, the probability of any given destination as in (1) and (2) can be determined by evaluating the expression in equation (25) for each $d \in \mathbb{D}$, followed by the normalisation

$$p(D = d \mid y_{1:n}) = \frac{p(y_{1:n} \mid D = d)p(D = d)}{\sum_{j \in \mathbb{D}} p(y_{1:n} \mid D = j)p(D = j)}, \qquad (28)$$

which ensures that the total probability over all possible destinations sums to 1. In Algorithm 2, this is approximated (due to numerical quadrature) by $u_d \approx p(D = d \mid y_{1:n})$. For a fixed set of quadrature points, this estimated posterior can be updated sequentially after each observation, making it tractable for a large numbers of possible destinations (N) and quadrature points (q).

D. Arrival Time Inference

In addition to inferring the intended destination D of the tracked object, it is also possible to infer a posterior distribution of the time at which the target is expected to reach

---

Algorithm 2 Destination Inference

Input: Observations $y_{1:N}$
Initialize: Set $L_0^{(d,i)} = 1$ and set $\hat{Z}_0^{(d,i)}$, $\Sigma_0^{(d,i)}$ to priors from equation (21) for all $d \in \mathbb{D}$, $i = 1, ..., q$
for observations $n = 1, ..., N$ **do**
  for destination $d \in \mathbb{D}$ **do**
    for quadrature point $i \in 1, ..., q$ **do**
      Calculate $R_t^{(d,i)}$, $U_t^{(d,i)}$ in equation (18) for observation time t_n, destination d and arrival time T_i
      Run Kalman filter iteration:
      $\{\ell_n^{(d,i)}, \hat{Z}_n^{(d,i)}, \Sigma_n^{(d,i)}\} =$
          $\text{KF}(y_n, \hat{Z}_{n-1}^{(d,i)}, \Sigma_{n-1}^{(d,i)}, R_t^{(d,i)}, U_t^{(d,i)}, \tilde{G})$
      $\{\ell_n^{(d,i)}$ is the PED for a known arrival time, i.e. $\ell_n^{(d,i)} = p(y_n \mid y_{1:n-1}, D = d, T = T_i)\}$
      Update likelihood: $L_n^{(d,i)} = L_{n-1}^{(d,i)} \times \ell_n^{(d,i)}$
    end for
    Calculate likelihood approximation:
      $P_n^{(d)} = \text{quad}(L_n^{(d,1)}, L_n^{(d,2)}, ..., L_n^{(d,q)})$
    $\{P_n^{(d)} \approx p(y_{1:n} \mid D = d)$; quad is quadrature function$\}$
  end for
  for destination $d \in \mathbb{D}$ **do**
    $u_d = \frac{p(D=d)P_n^{(d)}}{\sum_{d \in \mathbb{D}} p(D=d)L_n^{(d)}}$
  end for
  Destination posterior after n^{th} observation:
    $p(D = d \mid y_{1:n}) \approx u_d$
end for

---

its intended (unknown) endpoint. For a specific destination $D = d$, this is given by

$$p(T \mid D = d, y_{1:t}) \propto p(y_{1:t} \mid T, D = d)p(T \mid D = d). \quad (29)$$

Since the quadrature procedure used in Section IV-A requires calculation of the arrival-time-conditioned-likelihood, i.e. $p(y_{1:t} \mid T = T_i, D = d)$, for a number of quadrature points T_i, a discrete approximation of the overall posterior can be obtained almost without additional calculations via

$$p(T \mid D = d, y_{1:t}) \approx \sum_{i=1}^{q} w_i \delta_{\{T_i\}}, \qquad (30)$$

where $\delta_{\{T_i\}}$ is a Dirac delta located at T_i. This is a weighted distribution over the quadrature points with

$$w_i = \frac{p(y_{1:t} \mid T = T_i, D = d)p(T_i \mid D = d)}{\sum_{i=1}^{q} p(y_{1:t} \mid T = T_i, D = d)p(T_i \mid D = d)}.$$

These weights are normalized evaluations of the expression in equation (29), calculated at each T_i. Normalization ensures that the approximate posterior distribution in (30) is a valid probability distribution that integrates to 1.

The posterior distribution of the arrival time at *any* destination can be calculated without significant further calculations by integrating over all endpoints (since $\mathbb{D}$ is a discrete set) as

$$p(T \mid y_{1:t}) = \sum_{d \in \mathbb{D}} p(T, D = d \mid y_{1:t})$$
$$\propto \sum_{d \in \mathbb{D}} p(y_{1:t} \mid T, D = d)p(T \mid D = d)p(D = d). \qquad (31)$$

Since the conditioned likelihood $p(y_{1:t} \mid T, D = d)$ is determined for each $d \in \mathbb{D}$ and quadrature point $T = T_i$, $i = 1, 2, ..., q$, the distribution in (31) can be approximated by

$$p(T \mid y_{1:t}) \approx \sum_{i=1}^{q} v_i \delta_{\{T_i\}},$$

with the weights defined by

$$v_i = \frac{\sum_{d \in \mathbb{D}} p(y_{1:t} \mid T_i, D = d) p(T_i \mid D = d) p(D = d)}{\sum_{i=1}^{q} \sum_{d \in \mathbb{D}} p(y_{1:t} \mid T_i, D = d) p(T_i \mid D = d) p(D = d)}.$$

This relies on the assumption that the set of arrival time quadrature points are the same for each destination $d \in \mathbb{D}$.

E. State Inference and Trajectory Prediction

Estimating the current state of the tracked object (e.g. position) and predicting its future state can also be readily preformed within the framework presented. The posterior estimate of the target state $p(X_{t_n} \mid y_{1:t})$ is given by integrating over all possible destinations and arrival times, i.e.

$$p(X_{t_n} \mid y_{1:t}) = \int \left[\sum_{d \in \mathbb{D}} p(X_{t_n} \mid y_{1:t}, T, D = d) \right.$$
$$\left. \times p(T \mid D = d) p(D = d) \right] dT. \quad (32)$$

The estimated state at the time instant t_n conditioned on the arrival time T_i and endpoint d is $p(X_{t_n} \mid y_{1:t}, T = T_i, D = d)$. It has a Gaussian distribution with mean and variance given by $P_t \hat{Z}_n^{(d,i)}$ and $P_t \Sigma_n^{(d,i)} P_t'$, respectively, where, $P_t = \begin{bmatrix} I_r & 0_r \end{bmatrix}$, i.e. the components of $\hat{Z}_n^{(d,i)}$ and $\Sigma_n^{(d,i)}$ in Algorithm 2 corresponding to X_{t_n} rather than X_{T_i}.

The distribution in (32) can be approximated from the calculated arrival time and destination conditioned state estimates by the Gaussian mixture

$$p(X_{t_n} \mid y_{1:t}) \approx \sum_{i=1}^{q} \sum_{d \in \mathbb{D}} u_{i,d} \mathcal{N} \left(X_{t_n}; P_t \hat{Z}_n^{(d,i)}, P_t \Sigma_n^{(d,i)} P_t' \right)$$

where weights in the above summation are given by

$$u_{i,d} = \frac{p(y_{1:t} \mid T_i, D = d) p(T_i \mid D = d) p(D = d)}{\sum_{i=1}^{q} \sum_{d \in \mathbb{D}} p(y_{1:t} \mid T_i, D = d) p(T_i \mid D = d) p(D = d)}. \quad (33)$$

The future state of the tracked object can be estimated using its dynamics model, applied to each arrival time and destination. This is identical to the 'predict' step of the Kalman filter in Algorithm 1 for the future time instant of interest $t^* > t_n$. For a given arrival time $T = T_i$ and destination $D = d$, the prediction of the object future state is Gaussian with mean and covariance specified by, respectively,

$$\hat{Z}_{t^*|n}^{(d,i)} = R_{t^*}^{(d,i)} \hat{Z}_n^{(d,i)} + \tilde{m}_{t^*}^{(d,i)}$$
$$\Sigma_{t^*|n}^{(d,i)} = R_{t^*}^{(d,i)} \Sigma_n^{(d,i)} (R_{t^*}^{(d,i)})' + U_{t^*}^{(d,i)}.$$

Each of $R_{t^*}^{(d,i)}$, $m_{t^*}^{(d,i)}$ and $U_{t^*}^{(d,i)}$ are calculated in the same way as R_t, m_t and U_t in equation (19) for a time step h, corresponding to the required prediction time, i.e. $h = t^* - t_n$. The predicted distribution of the target state is a Gaussian

mixture with the same component weights $u_{i,d}$ as for state estimate at the current time instant t_n in (33) such that

$$p(X_{t^*} \mid y_{1:t}) \approx \sum_{i=1}^{q} \sum_{d \in \mathbb{D}} u_{i,d} \mathcal{N} \left(X_{t^*}; P_t \hat{Z}_{t^*|n}^{(d,i)}, P_t \Sigma_{t^*|n}^{(d,i)} P_t' \right). \quad (34)$$

The inference algorithms for the posteriors of the destination, arrival time, and current and future state of the tracked object can be readily parallelised, with calculations for each quadrature point and/or nominal destination able to be run largely independently on an independent processor. Only the weight normalization step as in e.g. (33) requires results from all calculations to be available, fitting naturally into e.g. a map-reduce programming paradigm for parallel implementation.

V. RESULTS

In this section, we evaluate the performance of the proposed bridging-distributions-based intent inference approach in two application areas, namely HCI[2] and maritime surveillance[3]. Maximising the likelihood function $\prod_{j=1}^{J} p(y_{1:T}^j \mid D = d, \Omega)$ for a sample of J typical trajectories (constituting the training set) is the criterion adopted below to set the motion model parameters Ω. For example, for the MRD and ERV models, we have $\Omega = \{\Lambda, A, \sigma\}$. The learnt (fixed) values are then applied to all of the tested out-of-sample trajectories. This parameter estimation procedure is suitable for an operational real-time system where parameter training is an off-line process based on historical data.

A. Intent-aware Interactive Displays

In the results presented here, 50 free hand pointing gestures pertaining to four participants interacting with an in-vehicle touchscreen are used ($J = 5$ tracks are utilised for training). This data is collected in a system mounted to the dashboard of an instrumented car (identical to the prototype used in [16]). It consists of an 11" Windows tablet and a gesture-tracker, namely the Leap Motion (LM) Controller [39]. The LM produces, in real-time, the 3D cartesian coordinates of the pointing finger/hand, i.e. $y_n = [\hat{x}_{t_n} \ \hat{y}_{t_n} \ \hat{z}_{t_n}]'$, at an average frame rate of ≈ 30Hz. An experimental GUI of a circular layout is displayed on the tablet screen; it has 21 selectable circular icons that are ≈ 2 cm apart. Participants are asked to select a highlighted on-screen GUI item, giving a known ground truth intention D^+. Nevertheless, in all experiments, the predictor is unaware of the track end time T and the intended destination, when making decisions. To demonstrate the possible range of times of arrival at destination, Figure 3 depicts the distribution of T for 4,000 recorded in-vehicle free hand pointing gestures aimed at selecting on-screen icons. This empirical $p(T \mid D)$ is used as the arrival time prior for all destinations within the bridging-distribution-based predictors.

---

[2]Please refer to the attached video demonstrating an intent-aware display operating in real-time on a sample of typical in-car free hand pointing gestures; alternatively, follow the link: *https://youtu.be/lq4Xolpqbuc*.

[3]Please refer to the attached videos demonstrating the destination and track prediction for a vessel approaching a coast with multiple possible ports; alternatively, follow the link: *https://youtu.be/ElyFh-xwMWs*.
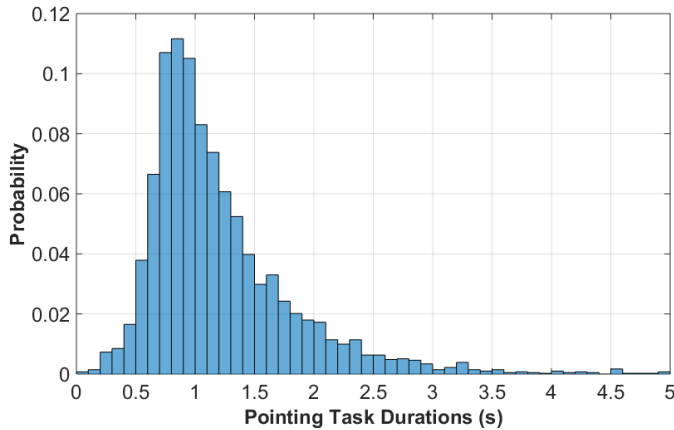
Figure 3: Distribution of the pointing times, $p(T|D)$, from over 4000 in-vehicle free hand pointing gestures to select on-screen icons.
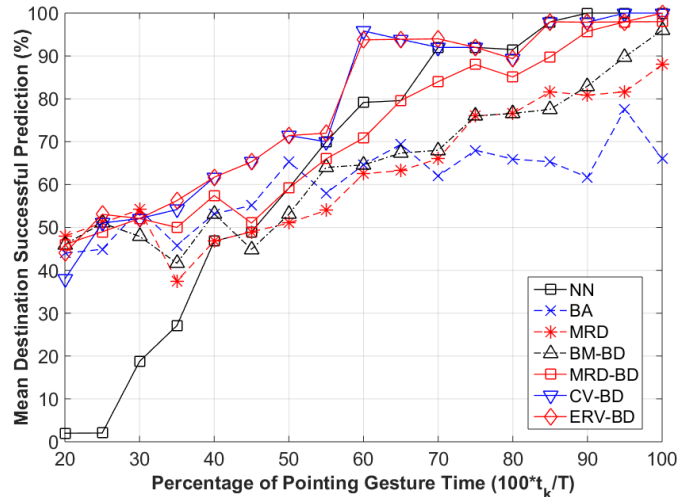

Figure 4: Mean percentage of successful destination inference as a function of pointing time.
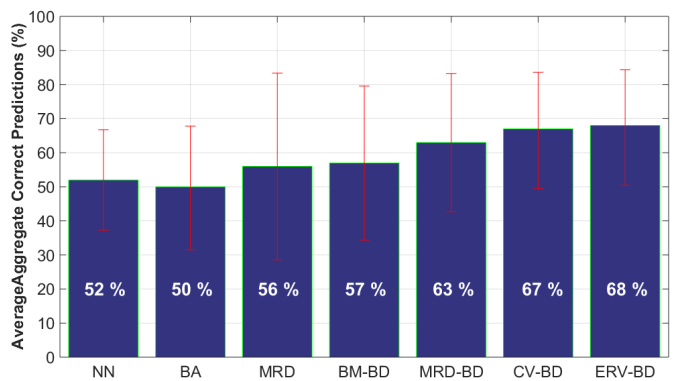

Figure 5: Gesture portion (in time) with successful prediction (error bars are one standard deviation).

The inference performance is evaluated in terms of the ability of the predictor to successfully establish the intended icon D via the MAP estimator in equation (2), i.e. how early in the pointing gesture the predictor assigns the highest probability to D. The prediction success is defined by $S(t_n) = 1$ if $\hat{D}(t_n) = D^+$ and $\mathcal{S}(t_n) = 0$ otherwise, for observations at times $t_n \in \{t_1, t_2, ..., T\}$. This is depicted in Figure 4 versus the percentage of completed pointing gesture (in time), i.e. $t_p = 100 \times t_n/T$, and averaged over all considered pointing tasks. Figure 5 shows the proportion of the total pointing gesture (in time) for which the predictor correctly establishes the intended destination.

In Figures 4 and 5, we assess the linear destination reverting, BM and CV models with the bridging prior notated here as MRD-BD, ERV-BD, BM-BD and CV-BD. A mean reverting diffusion model without bridging (MRD) is also shown to illustrate the gain attained by incorporating the prior on X_T. Additionally, the benchmark Nearest Neighbour (NN) and Bearing Angle (BA) methods are examined (see [16] for more details). In the former, the destination closest to the pointing finger-tip position is assigned the highest probability and vice versa; i.e. $p(y_n|D = d) = \mathcal{N}(y_n; p_d, \sigma_{\text{NN}}^2)$ where σ_{NN}^2 is covariance of the multivariate normal distribution. In BA, $p(y_n|y_{n-1}, D = d) = \mathcal{N}(\theta_n; 0, \sigma_{\text{BA}}^2)$ where the angle to destination d is $\theta_n = \angle (y_n - y_{n-1}, d)$ and σ_{BA}^2 is a design parameter. It assumes that the cumulative angle to the intended destination is minimal.

Figure 4 shows that the introduced bridging-distributions-based inference schemes CV-BD and ERV-BD, achieve the earliest successful intent predictions. This is particularly visible in the first 70% of the pointing gesture where notable reductions in the pointing time can be obtained and pointing facilitation regimes can be most effective (e.g. expanding icon(s) size, mid-air selection, etc.). Destination prediction towards the end of the pointing gesture, e.g. in the last third of the pointing time, has limited benefit, since by that stage the user has already dedicated the necessary visual, cognitive and manual efforts to execute the task. In general, the performance of all evaluated predictors improves as the pointing hand/finger is closer to the display. This is particulary visible

for the nearest neighbour model, which exhibits very poor performance early in the pointing task and gradually matches other techniques as the observed track length increases as the pointing finger becomes close to the endpoint. An exception is the BA model, where the reliability of the heading angle as a measure of intent declines as the pointing finger approaches the destination.

The gains from combining the MRD motion model with the bridging technique (MRD-BD) are clearly visible in Figure 4 compared to the MRD without bridging. This can be attributed to the ability of bridging models to reduce the sensitivity of linear destination reverting models to variability in the processed trajectories, which reduces the system sensitivity to parameter estimates and thus reduces the parameter training requirements. MRD performance versus that of the NN has deteriorated compared with that in [16] since less parameter training is performed here; only $J = 5$ out of the 50 tested tracks are used for training to illustrate the low training requirement of the applied bridging-distributions-based inference approach. Similar observations are made for the ERV which has even more parameters than MRD; the quality of its predictions without bridging (not shown here) is very poor.

Figure 5 demonstrates that the proposed bridging-

distribution-based inference delivers the highest overall correct destination predictions across the pointing trajectories. The highest aggregate successes are achieved by the constant velocity and equilibrium reverting velocity models with relatively tight error bounds. This is due to the importance of the velocity component in the pointing task, which is only captured by these two models. MRD without bridging has the largest variance, highlighting its lack of robustness without the bridging element. NN and BA performances are similar over the considered data set.

It is important to note that small improvements in pointing task efficiency (effort reduction), even reducing pointing times by few milliseconds, will have substantial aggregate benefits on overall user experience since interactions with displays are very prevalent in typical scenarios, e.g. using a touchscreen in a modern vehicle environment to control the car infotainment system [13], [14]. In a developed initial prototype of a predictive display system (an optimised C# implementation on a typical automotive computing platform), prediction with Kalman filtering was tested with up to $N = 64$ destinations and an observations data rate ≈ 30Hz without any noticeable delays in the system response.

B. Maritime Situational Awareness

Figure 6 shows the results of the proposed destination inference algorithm, via the MAP estimate in (2), for a two-dimensional vessel tracking problem. The aim is to predict the endpoint of each vessel, from a set of $N = 6$ possible harbours, based on noisy observations $y_{1:n}$ of its trajectory up to the present time t_n. The trajectory data is generated from a bridged constant velocity model, such that tracks begin at a random point in the middle of the bay (around 20 km from the shore). It is conditioned on arriving at a chosen destination port (all equally likely) at a uniformly distributed random arrival time between 50 and 250 minutes later. Velocity at arrival is assumed to have a Gaussian distribution with a zero mean and standard deviation 10m min^{-1} (i.e. relatively slow). The dynamic noise parameter is $\sigma = 20$m min$^{-3/2}$ in both dimensions. Observations are direct measurements of the current vessel position, corrupted by Gaussian noise with standard deviation of 1m.

Destination inference in Figure 6 is performed using a bridged constant velocity model and the arrival time prior is uniform over the interval [50, 250] minutes, i.e. $p(T \mid D) = \mathcal{U}(50, 250)$. Quadrature was based on $q = 15$ evenly spaced arrival times over this time interval, employing Simpson's rule integration as in (24).

For most of the tracks depicted in Figure 6, the correct intended destination is inferred early in the observed trajectory. When the algorithm fails to achieve such early predictions, e.g. vessels heading to harbours 2 and 4, visual inspection of these tracks shows that the vessel in question makes a notable sharp manoeuvre at some point in its trajectory. Prior to these sharp changes in direction of travel, the vessel appears to be heading towards a different endpoint. Nevertheless, after the manoeuvre is completed, the correct destination is quickly inferred. This figure clearly highlights the potential of the proposed bridging-distribution-based destination inference.

Figure 6: Destination inference for ten ship trajectories, heading to one of six numbered destinations. Thick green lines show the portion of each trajectory for which the true intended destination is correctly inferred via MAP estimate in (2); thin red lines show portions of the trajectory for which this was not the case.
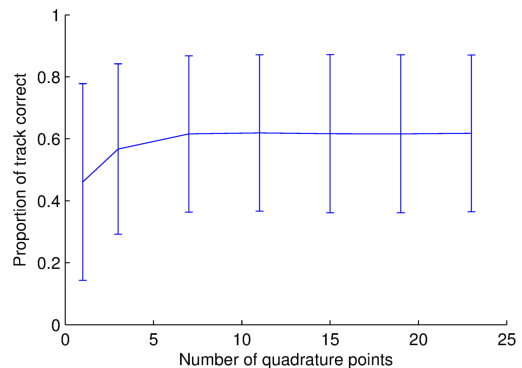

Figure 7: Proportion of track for which the true destination was inferred for different numbers of quadrature points, using Simpson's rule as in equation (24), averaged over 100 tracks (error bars show $\pm$ one standard deviation). End times are uniformly distributed in the interval [50, 250] min, within which the quadrature points are evenly spaced. For one quadrature point $T = 250$ is assumed for all tracks.

Figure 7 examines the effect of the number of quadrature points q on the endpoint inference outcome for the above vessel tracking scenario. This figure shows the average and standard deviation of the aggregate prediction successes for 100 randomly generated vessel trajectories using Simpson's rule quadrature as in equation (24) with varying numbers of quadrature points q. Figure 7 illustrates that increasing the number of quadrature points (up to $q = 9$) improves the inference performance, after which the success rate levels off. The result for one quadrature point is that for assuming $T = 250$ minutes for all tracks. The results in this figure demonstrate that assuming a distribution of arrival times is beneficial for destination inference compared to guessing a
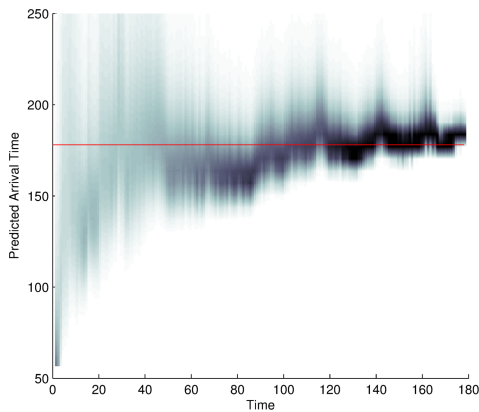
Figure 8: Arrival time estimate (across all possible destinations) for a track similar to those shown in Figure 6. The red line shows the true arrival time and shading shows the posterior density at each time.
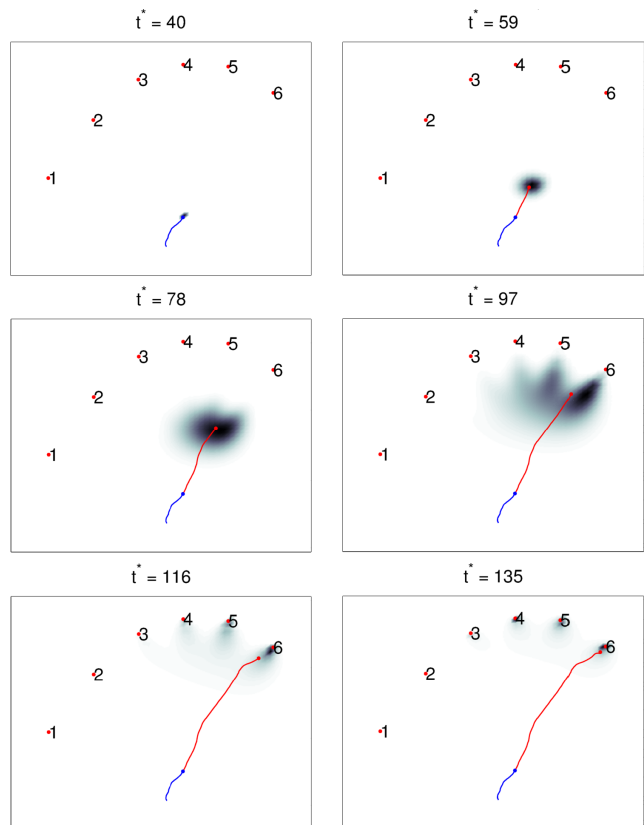

Figure 9: Predicted target position. Shading shows the posterior distribution of the tracked object position at a range of future time instants t^*. The available trajectory $X_{1:40}$, i.e. at $t_n = 40$, is depicted by the solid blue track. Top left panel shows the current position posterior at $t_n = 40$ and subsequent panels exhibit the predicted position posterior at times $t^* > 40$, up to $t^* \approx T$. Red line indicates the true trajectory values at t^* (red dot is the true target position at t^*). The arrival time and destination are unknown.

single arrival time. Most importantly, it shows that only a few quadrature points (e.g. 9 in this case) are sufficient to leverage this benefit.

Figure 8 shows the arrival time estimations for a track similar to those considered in Figure 6, applying the technique described in Section IV-D with $q = 31$ quadrature points. Initially, the arrival time is uncertain as shown by the diffuse shading. However, as more trajectory data becomes available, the posterior steadily becomes more concentrated in the region of the true arrival time (red line).

Finally, the predicted target position at a number of future time instants $t^* > t_n$ are displayed in Figure 9, using the prediction method in Section IV-E. At the time shown t_n, 40 observations have been made. The tracked object's true position up to each of the assessed future times is shown in red. Initially, the inference is unimodal, dominated by the object's current motion. As predictions are made further into the future, the possible destinations of the target become visibly influential and the predicted position becomes multimodal (e.g. see the last row in Figure 9). Each of these modes corresponds to a destination $d \in \mathbb{D}$ and the prediction is dominated by the endpoint(s) deemed to be more probable by the inference algorithm. For example, the available trajectory (blue line) leads to more weight being assigned to endpoints 4, 5 and 6 compared to the other possible destinations (noting that $d = 6$ is the true intended destination).

The effect of the unknown arrival time T (estimated using $q = 25$ quadrature points) is reflected in Figure 9 by the shape of the predicted densities, which resemble a "finger" shape pointing to each destination (particularly in the fourth panel at $t^* = 97$). These 'fingers' result from an increased uncertainty about where the target is in its approach to each destination due to the unknown arrival time. As more observations are made, this effect diminishes due to the increased confidence in the arrival time, as per Figure 8.

Defining a normal trajectory template (pattern of life) for vessels moving towards potential docking points (e.g. over relatively short distances as in Section V-B) or protected assets can be challenging. This is due to the fact that the tracked object approach to the intended endpoint can significantly vary and might not necessarily conform to a known pattern (especially if the intent is malicious). Besides, building a complete training data set might not be possible for such scenarios. The proposed inference algorithms in this paper, which do not treat intent inference as an anomaly detection problem, are able to handle such a setting effectively with consistent predictions (destination, position and time of arrival) through the use of bridging distributions.

VI. CONCLUSION

This paper sets out a probabilistic framework for a simple, low complexity intent inference that demands minimal training and is amenable to parallelisation. Utilising the bridging approach presented here not only permits earlier predictions, but also significantly improves the robustness of destination reverting models against variability in the tracked object behaviour. The early inference of the destination, future position and time of arrival of the tracked object, e.g. pointing finger or vessel, can bring notable benefits such as reducing the attention required to interact with in-vehicle displays, enhancing the ability of maritime surveillance systems and facilitating supervised or unsupervised automated warning/assistive functions. The inference results from the two applications considered in

this paper testify to the effectiveness and usefulness of the introduced prediction algorithms. Whilst linear Gaussian motion and observation models are considered here, extending the formulation to a more generic settings (nonlinear and/or non-Gaussian), could broaden its applicability to other scenarios, especially when targets undertake sharp manoeuvres prior to reaching their intended endpoint. This would require the use of Bayesian filtering techniques such as sequential Monte Carlo algorithms [40], which will entail substantial additional computational cost. Other extensions include incorporating constraints into the path followed by the tracked object, modelling endpoints with complicated shapes that cannot be approximated by Gaussian distributions, consider interaction between multiple tracked agents in a scene (e.g. collision avoidance) and others.

ACKNOWLEDGMENT

REFERENCES

[1] D. A. Castanon, B. C. Levy, and A. S. Willsky, "Algorithms for the incorporation of predictive information in surveillance theory," *International Journal of Systems Science*, vol. 16, no. 3, pp. 367–382, 1985.

[2] B. Ristic, B. La Scala, M. Morelande, and N. Gordon, "Statistical analysis of motion patterns in AIS data: Anomaly detection and motion prediction," in *11 Int. Conf. on Info. Fusion (FUSION)*, 2008, pp. 1–7.

[3] E. Baccarelli and R. Cusani, "Recursive filtering and smoothing for reciprocal Gaussian processes with Dirichlet boundary conditions," *IEEE Trans. on Signal Processing*, vol. 46, no. 3, pp. 790–795, 1998.

[4] C. Piciarelli, C. Micheloni, and G. L. Foresti, "Trajectory-based anomalous event detection," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1544–1554, 2008.

[5] A. Wang, V. Krishnamurthy, and B. Balaji, "Intent inference and syntactic tracking with GMTI measurements," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 47, no. 4, pp. 2824–2843, 2011.

[6] M. Fanaswala and V. Krishnamurthy, "Detection of anomalous trajectory patterns in target tracking via stochastic context-free grammars and reciprocal process models," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 1, pp. 76–90, 2013.

[7] R. C. Nunez, B. Samarakoon, K. Premaratne, and M. N. Murthi, "Hard and soft data fusion for joint tracking and classification/intent-detection," in *16th Int. Conf. on Information Fusion (FUSION)*, 2013, pp. 661–668.

[8] G. Pallotta, M. Vespe, and K. Bryan, "Vessel pattern knowledge discovery from AIS data: A framework for anomaly detection and route prediction," *Entropy*, vol. 15, no. 6, pp. 2218–2245, 2013.

[9] G. Pallotta, S. Horn, P. Braca, and K. Bryan, "Context-enhanced vessel prediction based on Ornstein-Uhlenbeck processes using historical AIS traffic patterns: Real-world experimental results," in *17th Int. Conf. on Information Fusion (FUSION '14)*, 2014, pp. 1–7.

[10] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. Part I. Dynamic models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, 2003.

[11] ——, "Survey of maneuvering target tracking. part v. multiple-model methods," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1255–1321, 2005.

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

[13] C. Harvey and N. A. Stanton, *Usability Evaluation for In-vehicle Systems*. CRC Press, 2013.

[14] G. E. Burnett and J. Mark Porter, "Ubiquitous computing within cars: designing controls for non-visual use," *International Journal of Human-Computer Studies*, vol. 55, no. 4, pp. 521–531, 2001.

[15] S. G. Klauer, T. A. Dingus, V. L. Neale, J. D. Sudweeks, and D. J. Ramsey, "The impact of driver inattention on near-crash/crash risk: An analysis using the 100-car naturalistic driving study data," National Highway Traffic Safety Administration, DOT HS 810 5942006, 2006.

[16] B. I. Ahmad, J. K. Murphy, P. Langdon, S. Godsill, R. Hardy, and L. Skrypchuk, "Intent inference for pointing gesture based interactions in vehicles," *IEEE Trans. on Cybernetics*, vol. 46, pp. 878–889, 2016.

[17] B. I. Ahmad, P. M. Langdon, S. J. Godsill, R. Donkor, R. Wilde, and L. Skrypchuk, "You do not have to touch to select: A study on predictive in-car touchscreen with mid-air selection," in *Proc. of the Int. Conf. on Automotive User Interfaces and Interactive Veh. Apps. (AutomotiveUI)*, 2016.

[18] K. Kitani, B. Ziebart, J. Bagnell, and M. Hebert, "Activity forecasting," in *Proc. of 12th European Conference on Computer Vision (ECCV '12)*, 2012, pp. 201–214.

[19] G. Ferrer and A. Sanfeliu, "Bayesian human motion intentionality prediction in urban environments," *Pattern Recognition Letters*, vol. 44, pp. 134–140, 2014.

[20] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa, "Planning-based prediction for pedestrians," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 3931–3936.

[21] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: the case for cooperation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 2153–2160.

[22] A. Hüntemann, E. Demeester, E. Vander Poorten, H. Van Brussel, and J. De Schutter, "Probabilistic approach to recognize local navigation plans by fusing past driving information with a personalized user model," in *IEEE Int. Conf. on Robotics and Automation*, 2013, pp. 4376–4383.

[23] 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 (IROS)*, 2015, pp. 3836–3843.

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

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

[26] H. A. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Transactions on Automatic Control*, vol. 33, no. 8, pp. 780–783, 1988.

[27] J. K. Tugnait, "Adaptive estimation and identification for discrete systems with Markov jump parameters," *IEEE Trans. on Automatic Control*, vol. 27, no. 5, pp. 1054–1065, 1982.

[28] A. F. Bobick, Y. Ivanov *et al.*, "Action recognition using probabilistic parsing," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 1998, pp. 196–202.

[29] K. G. Derpanis, M. Sizintsev, K. J. Cannons, and R. P. Wildes, "Action spotting and recognition based on a spatiotemporal orientation analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 3, pp. 527–540, 2013.

[30] M. J. McGuffin and R. Balakrishnan, "Fitts' law and expanding targets: Experimental studies and designs for user interfaces," *ACM Trans. on Computer-Human Interaction*, vol. 12, no. 4, pp. 388–422, 2005.

[31] T. Asano, E. Sharlin, Y. Kitamura, K. Takashima, and F. Kishino, "Predictive interaction using the Delphian desktop," in *Proc. of the ACM Symp. on User Interface Software and Technology*, 2005, pp. 133–141.

[32] E. Lank, Y.-C. N. Cheng, and J. Ruiz, "Endpoint prediction using motion kinematics," in *Proc. of the SIGCHI Conf. on Human factors in Computing Systems*, 2007, pp. 637–646.

[33] B. Ziebart, A. Dey, and J. A. Bagnell, "Probabilistic pointing target prediction via inverse optimal control," in *Proc. of the 2012 ACM Int. Conf. on Intelligent User Interfaces (IUI)*, 2012, pp. 1–10.

[34] A. J. Haug, *Bayesian Estimation and Tracking: A Practical Guide*. John Wiley & Sons, 2012.

[35] B. I. Ahmad, J. K. Murphy, P. M. Langdon, S. J. Godsill, and R. Hardy, "Destination inference using bridging distributions," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP '15)*, Brisbane, 2015, pp. 5585–5589.

[36] D. T. Gillespie, "Exact numerical simulation of the Ornstein-Uhlenbeck process and its integral," *Physical review E*, vol. 54, p. 2084, 1996.

[37] S. Särkkä, "Recursive Bayesian inference on stochastic differential equations," Ph.D. dissertation, Helsinki University of Technology, Department of Electrical and Communications Engineering, 2006.

[38] T. Ardeshiri, B. I. Ahmad, P. M. Langdon, and S. J. Godsill, "Efficient bridging-based destination inference in object tracking," in *submitted to IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2017.

[39] Leap Motion Website: https://www.leapmotion.com/.

[40] 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, pp. 925–952, 2007.