Data-Centric Engineering (2020), 1: e12
doi:10.1017/dce.2020.11

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Modeling intent and destination prediction within a Bayesian framework: Predictive touch as a usecase

Runze Gan[1,2,*], Jiaming Liang[1,2], Bashar I. Ahmad[1] and Simon Godsill[1]

[1]Engineering Department, University of Cambridge, Cambridge, United Kingdom
[2]Equal Contribution
*Corresponding author. E-mail: rg605@cam.ac.uk

Received: 30 June 2020; **Revised:** 02 September 2020; **Accepted:** 04 September 2020

Keywords: Destination prediction; human–computer interaction; intent modeling; Kalman filter; object tracking; particle filter; sequential Monte Carlo; touchless interaction

Abstract

In various scenarios, the motion of a tracked object, for example, a pointing apparatus, pedestrian, animal, vehicle, and others, is driven by achieving a premeditated goal such as reaching a destination. This is albeit the various possible trajectories to this endpoint. This paper presents a generic Bayesian framework that utilizes stochastic models that can capture the influence of intent (viz., destination) on the object behavior. It leads to simple algorithms to infer, as early as possible, the intended endpoint from noisy sensory observations, with relatively low computational and training data requirements. This framework is introduced in the context of the novel predictive touch technology for intelligent user interfaces and touchless interactions. It can determine, early in the interaction task or pointing gesture, the interface item the user intends to select on the display (e.g., touchscreen) and accordingly simplify as well as expedite the selection task. This is shown to significantly improve the usability of displays in vehicles, especially under the influence of perturbations due to road and driving conditions, and enable intuitive contact-free interactions. Data collected in instrumented vehicles are shown to demonstrate the effectiveness of the proposed intent prediction approach.

Impact Statement

The presented Bayesian framework facilitates automated decision-making, resource allocation and future action planning with applications in various fields, such as in human–computer interaction (HCI), surveillance, robotics, to name a few. It led to the introduction of the patented HCI technology *predictive touch*, developed as part of a collaboration with Jaguar Land Rover and is set for commercialization; it won a Jaguar Land Rover TATA Innovista Award 2020 ("Dare To Try" category). Predictive touch does not only offer an intuitive approach to touchless interactions (i.e., no physical contact with the display is required), but also it can significantly improve the usability of interactive displays in vehicles or any moving platform, reduce the attention they require and enhance the input accuracy, including under the influence of perturbations due to road and driving conditions. This has been demonstrated in various on-road trials. This touchless interaction technology can have widespread applications in a post COVID-19 world by minimizing the risk of transmission of pathogens via touch surfaces, for instance, when using ticketing or self checkout machines, control panels, and interactive displays in public spaces, kiosks, or workplaces, and so on. It also offers a means to easily interact with emerging display technologies that do not have a physical surface, such as 2D/3D projections and in virtual or augmented reality, and offers additional design flexibility to support inclusive design practices.

1. Introduction

In conventional *sensor-level* tracking, the objective is typically to estimate the hidden state $\mathbf{x}_t$ of an object of interest (e.g., pointing apparatus, pedestrian, vehicle, vessel, airplane, etc.), where $\mathbf{x}_t$ is the target location, orientation, velocity, higher order kinematics, or other spatio-temporal characteristics. This state is assumed to be related to the available noisy sensory measurements (e.g., from camera, radar, inertial measurement units, radio frequency transmissions, global navigation satellite system, acoustic signals, etc.) as per a defined observation model. Plethora of well-established algorithms for estimating $\mathbf{x}_t$ exist, including from multiple data sources, see Bar-Shalom et al. (2011) and Haug (2012). They often implicitly assume that the object moves in an unpremeditated manner and suitable motion models are accordingly employed.

In this paper, the aim is not to estimate the state $\mathbf{x}_t$, but instead to infer the underlying intent that is driving the object motion, namely its destination. This capitalizes on the premise that the target motion (e.g., the trajectory followed by a pointing finger while interacting with a display) is dictated by the intended endpoint (e.g., the sought interface item), and that the destination influence on the target movements can be modeled. Therefore, the sought probabilistic modeling and destination predictor(s) belong to a higher system level compared with the sensor-level tracking techniques, hence dubbed *meta-level tracking* algorithms. They have several applications, such as in surveillance, human–computer interaction (HCI), robotics, and others, since such meta-level approaches can facilitate automated decision-making, resources allocation and informed future action planning. They offer a more integrated viewpoint of a scene where intents can be automatically learnt and conflict or opportunities can be identified in a timely manner. The HCI technology, dubbed predictive touch, is used here as an application or motivation for the proposed Bayesian meta-level inference framework. Nonetheless, this approach can be applied in numerous other areas and scenarios.

1.1. Predictive touch

Predictive touch is an emerging HCI technology for intelligent displays and touchless interactions that can predict the interface component the user intends to select (e.g., a selectable graphical user interface [GUI] displayed on a touch screen), notably early in the pointing-selection task (Ahmad et al., 2017). This is based on the available freehand pointing movements in 3D, for example, provided by gesture trackers, and potentially other available sensory data such as eye-gaze. The pointing-selection task is then simplified and expedited by the predictive touch solution via applying a suitable selection facilitation scheme. This can significantly reduce the effort and distractions associated with using in-vehicle displays while driving (Jæger et al., 2008), including under the influence of perturbations, for example, vibrations and accelerations due to the road and driving conditions. Such perturbations can have a detrimental impact on the usability of displays in moving platforms, such as in-vehicle touch screens (Goode et al., 2012; Ahmad et al. 2015), which often act as the gateway to control in-vehicle infotainment systems. For instance, pointing time can be reduced by over 30% and effort/workload halved with predictive touch, see Ahmad et al. (2017). It is noted that gesture trackers are increasingly becoming commonplace in automotive, gaming, infotainment applications in general and more recently in smartphones, see Quinn et al. (2019), due to recent advancements in sensing and computer-vision systems. Thus, predictive touch system typically assumes the presence of a gesture tracker (including integrated into the display, e.g., computer-vision solution with several built-in cameras on a touch screen), which it can utilize.

Figure 1 depicts the system block diagram which comprises of the following four main modules:

- *Pointing gesture tracker*: provides, in real-time, the pointing hand/finger(s) location in 3D, for example, $\mathbf{y}_{0:n}$ is the partial (filtered) pointing trajectory pertaining to the time instants $\{t_1, t_2, \ldots, t_n\}$ at time t_n.
- *Intent predictor*: for a set of $N_{\mathcal{D}}$ selectable interface icons, $\{\mathcal{D}_i : i = 1, 2, \ldots, N_{\mathcal{D}}\}$, this module calculates the likelihood of each of $\mathcal{D}_i$ being the intended destination at t_n, from the available $\mathbf{y}_{1:n}$.

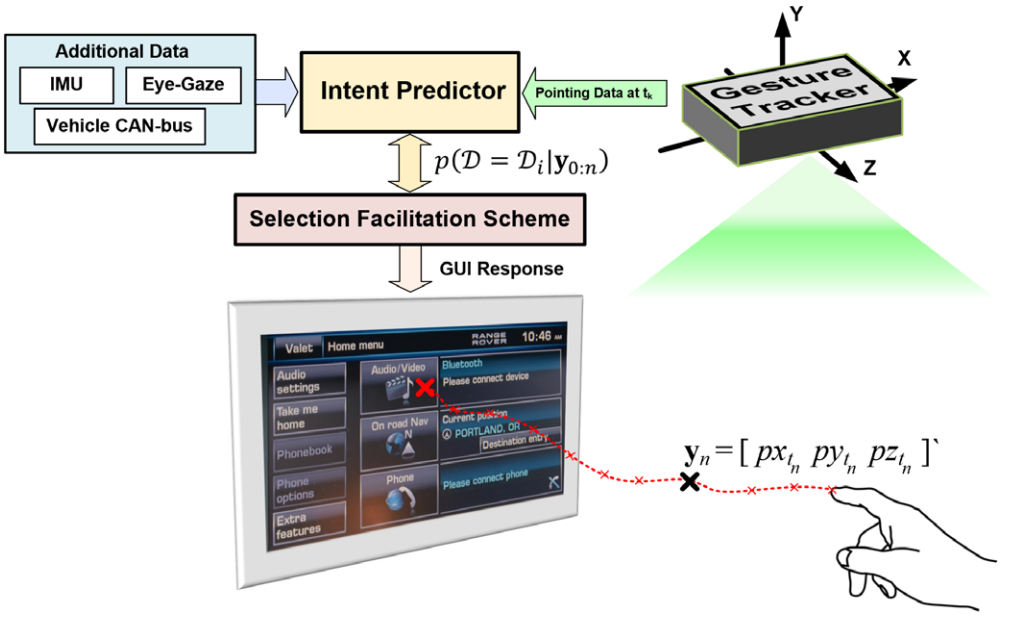


Figure 1. *Block diagram of an in-vehicle predictive touch system. The dotted line is a recorded full in-car pointing trajectory.The gesture tracker (sensor is facing downwards to increase the region of coverage and minimize occlusions) provides at time t_n the pointing finger/hand Cartesian coordinates along the* x, y, *and* z *axes, denoted by* $\mathbf{y}_n$.

- *Selection facilitation*: based on the prediction results, the system simplifies-expedites the selection task. Various such facilitation schemes can be applied (e.g., expand or highlight/fade or drag the item closer to the pointing location, etc.) and were the subject of the studies in Ahmad et al. (2019a) for automotive applications. It was reported that the system autonomously selecting the predicted GUI item on behalf of the user, thus immediate mid-air selection, is an effective facilitation scheme leading to touchless or contact-free interactions.
- *Additional data*: available additional sensory data, such as inertial (accelerometer/gyroscope), eye-gaze measurements, environmental data can be utilized to improve the prediction results. For instance, vehicle CAN-bus data (e.g., suspensions and speed signals) can indicate the level of experienced perturbations due to road-driving conditions.

Therefore, it is software-based touchless technology where the user does not need to physically touch a display to select an interface component. Predictive touch can not only improve the usability and performance of interactive displays, but it also provides the means to interact with new display technologies that do not have a physical surface such as head-up displays, holograms and 3D projections (Bark et al., 2014; Broy et al. 2015). This novel HCI solution uses the intuitive free hand pointing gestures and intrinsically relies on predicting the user intent, rather than using the pointing finger/arm location or orientation as a pointing apparatus as in Roider and Gross (2018). Thereby, predictive touch is not a *mid-air* pointing or ray-casting approach (Plaumann et al., 2018), and it is fundamentally distinct from gesture-recognition-based interactions that require the user to pre-learn particular "symbolic" gesture shapes to trigger certain interface responses (May et al. 2017). It also offers several design flexibility in terms of the display placement and GUI design which is otherwise limited by the reach and motor capabilities of the user. This can promote inclusive design practices by tailoring the display operation to the user requirements via configuring the prediction algorithms and facilitation schemes.

1.2. Related work and contributions

The Bayesian framework for intent prediction presented in this article was introduced in Ahmad et al. (2016b) and Ahmad et al. (2018) for predictive touch and other applications; see Ahmad et al. (2019b) for a short overview. It treats the problem within an object tracking formulation, albeit not necessarily seeking state estimation, such that the influence of intended destination is captured by utilizing suitable stochastic motion model with a few unknown parameters. The latter parameters can be estimated from a small number of example motion patterns or trajectories. Linear Time-Invariant Gaussian systems were considered in the aforementioned papers and more recently nonlinear behavior due to external forces (e.g., jumps and jolts in the pointing movements due to the road/driving conditions) was briefly addressed in Gan et al. (2019). Here and compared with previous work, we

1. present an overview and unified treatment of the intent prediction task for linear as well as nonlinear (albeit within a conditionally linear formulation) motion models and systems,
2. propose a new approach to the *bridging distributions* (BD) class of intent-driven models, which have a moderate computational requirement and a clear stochastic interpretation. In this context, the previously unconsidered bridged (nearly) constant acceleration dynamic model is shown to deliver the highest prediction performance for a predictive touch system, and
3. benchmark various prediction models using significantly larger data set of pointing gestures recorded in instrumented vehicles under various road-driving conditions.,

In the tracking area, incorporating *known* predictive information to improve the accuracy of state estimation has a long history, for example Castanon et al. (1985) and Baccarelli and Cusani (1998). Additionally, mean-driven models such as those derived from an Ornstein–Uhlenbeck (OU) process, with known means, were to better estimate behavior of certain objects, for example vessel Millefiori et al. (2016) or financial time series data in Christensen et al. (2012). Also, the use of stochastic context-free grammar (SCFG) and conditionally Markov process/reciprocal process has been proposed to predict intent as in Fanaswala and Krishnamurthy (2015) and Rezaie and Li (2019a,b). In this paper, the destination (i.e., intent) is assumed to be unknown and predictors are developed to infer it. The adopted formulation here leads to significantly simpler algorithms with no constraints on the trajectory followed by the object (e.g., freehand pointing finger), unlike those using SCGF which discretizes the state space. The employed continuous state space models within the introduced Bayesian framework, such as OU-type process and bridging distributions (both are detailed in the next Section), enable treating asynchronous sensory measurements. A noteworthy fact is that the bridging distribution can be viewed as a special case of conditionally Markov models in Rezaie and Li (2019a) under certain assumptions.

On the other hand, modeling and inferring complex intentions, such as drivers behaviors at junctions, pedestrians at crosswalks, and human daily activities, can be tackled with data-driven or classification approaches, possibly combined with an *a priori* learnt pattern of life. They assume the availability of sufficiently complete and diverse training data sets with several well-established such prediction techniques, for example Bando et al. (2013), Völz et al. (2018), and Gaurav and Ziebart (2019). However, in this paper, the objective is to develop a simple and computationally efficient destination prediction algorithm where limited training data are available. For example, it can be very challenging and expensive to collect data sets of 3D freehand pointing gestures that sufficiently sample possible paths/trajectories to the display, starting locations of the gesture (e.g., steering wheel, armrest and others), road/driving conditions, context of use, user interface design, screen size/reach, etc. Instead, suitable state space models are employed here, albeit with a few unknown parameters, as is common in object tracking. They enable modeling and robustly inferring the intended endpoint of a tracked object, especially that the possible intentions are a finite set of nominal destinations, for example selectable interface items. Subsequently, the introduced Bayesian intent predictors have minimal training requirements.

1.3. Paper layout

The remainder of the paper is organized as follows. The overall inference framework, various approaches to modeling intent, and the system model are described in Section 2. Destination predictors for linear and nonlinear settings are then outlined in Section 3. Results using real pointing data, recorded by in-vehicle predictive touch prototypes under various road conditions, are presented in Section 4, and conclusions are drawn in Section 5.

2. Bayesian Framework: Modeling Intent and Overall System

Here, the destination inference problem is treated within a Bayesian framework. Let $\mathbb{D} = \{\mathcal{D}_i : i = 1,2,\ldots,N_{\mathcal{D}}\}$ be the set of $N_{\mathcal{D}}$ nominal endpoints (e.g., selectable on-display interface icons) of a tracked object (e.g., a pointing finger-tip). The objective is to *sequentially* calculate the probability of each destination (i.e., selectable interface components) $\mathcal{D}_i \in \mathbb{D}$ being the intended endpoint at the current/latest time instant t_n, thus $p(\mathcal{D} = \mathcal{D}_i|\mathbf{y}_{0:n})$, $i = 1,2,\ldots,N_{\mathcal{D}}$, from the available sensory measurements $\mathbf{y}_{0:n} = \{\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_n\}$. We recall that in a predictive touch system observations $\mathbf{y}_{0:n}$ are provided by the gesture tracker and other sensors at the successive time instants $\{t_0, t_1, \ldots, t_n\}$, for instance, $\mathbf{y}_n$ is the 3D Cartesian coordinates of the pointing finger/hand at t_n as in Figure 1. For each $\mathcal{D}_i \in \mathbb{D}$ and per Bayes' rule, we have

$$p(\mathcal{D} = \mathcal{D}_i|\mathbf{y}_{0:n}) \propto p(\mathbf{y}_{0:n}|\mathcal{D} = \mathcal{D}_i)p(\mathcal{D} = \mathcal{D}_i), \quad i = 1,\ldots,N_{\mathcal{D}}, \tag{1}$$

where $p(\mathcal{D} = \mathcal{D}_i)$ is the prior on the ith possible destination. In predictive touch this prior can be attained from semantic data, frequency of use, interface design, other sensory data, etc. The task of the inference module (i.e., intent predictor in Figure 1) at t_n is hence to estimate the likelihoods $p(\mathbf{y}_{0:n}|\mathcal{D} = \mathcal{D}_i)$, $i = 1,2,\ldots,N_{\mathcal{D}}$. This makes the Bayesian formulation particularly appealing since additional contextual information can be easily incorporated, whenever available.

2.1. Destination-driven motion models

A key challenge within the introduced Bayesian approach is employing suitable motion models that represent the influence of intent on the object motion and devising inference algorithms to reveal it. The object motion (e.g., pointing gesture movement) towards an intended item on a display is not deterministic or necessarily takes the shortest path to the endpoint. This is because this movement is driven by a very complex sensorimotor system, capable of autonomous action based on various modalities (e.g., vision and can utilize feedback on the action) and is also subjected to various constraints (e.g., to optimize action required to deliver/predict smooth movement trajectories and minimize the variance of the eye or arm's position, in the presence of biological noise due to mechanical properties of muscles) and possibly perturbed by external forces such as due to road/driving conditions or walking, see Harris and Wolpert (1998). Thereby, models of such motion are intrinsically uncertain and any prediction of the object movements at a future time instant should not be a single point following a particular deterministic path. Instead, it should be expressed as a probability distribution in space.

Stochastic processes can adequately capture the aforementioned motion uncertainties, where state $\mathbf{x}_n$ (e.g., pointing finger true position in 3D) at t_n is related to its position at the previous time step t_{n-1}, according to a given probability distribution defined by the following evolution of the state over time

$$\mathbf{x}_{n,i} = \mathbf{f}_{i,h}(\mathbf{x}_{n-1,i}) + \boldsymbol{\varepsilon}_{n-1} \tag{2}$$

where $\mathbf{f}_{i,h}(.)$ is the state transition function between t_{n-1} and t_n and $h = t_n - t_{n-1}$. Here, this function can be nonlinear and it is assumed to be dependent on the intended endpoint $\mathcal{D}_i$; thus the subscript index i. Whereas, $\boldsymbol{\varepsilon}_{n-1}$ is the process noise, which is often assumed to be independently and identically distributed (i.i.d) and represents the uncertainty in motion. For example, a zero-mean Gaussian process noise with covariance $\mathbf{Q}_{i,h}$ and a linear time-invariant transition function, for example $\mathbf{x}_{n,i} = \mathbf{F}_{i,h}\mathbf{x}_{n-1,i} + \boldsymbol{\mu}_{i,h} + \boldsymbol{\varepsilon}_{n-1}$, lead to a transition density of the state at t_n described by a multivariate Gaussian distribution. It is given by:

$p(\mathbf{x}_{n,i}|\mathbf{x}_{n-1,i}) = \mathcal{N}\left(\mathbf{x}_{n,i}|\mathbf{F}_{i,h}\mathbf{x}_{n-1,i} + \boldsymbol{\mu}_{i,h}, \mathbf{Q}_{i,h}\right)$ where its mean is dependent on the previous position $\mathbf{x}_{n-1,i}$, input term $\boldsymbol{\mu}_{i,h}$ and covariance $\mathbf{Q}_{i,h}$. The latter represents the potential level of uncertainty between successive movements.

2.1.1. Linear Gaussian motion models

Approximate motion models that enable inferring intent, that is not necessarily the exact modeling of the object motion, can suffice for the task of destination prediction. Under this assertion, Gaussian Linear Time Invariant (LTI) models can be particularly favorable since they can be easily formulated and lead to computationally efficient prediction algorithms, compared with nonlinear non-Gaussian models (Godsill, 2007; Haug, 2012). Next, two classes of Gaussian LTI intent-driven models, namely mean-reverting and bridging distributions, are introduced.

2.1.1.1. Linear Gaussian mean reverting models. The OU process with mean reverting property offers an effective way to model the destination-driven behavior. By setting the mean term of the underlying model according to the destination information, the target would revert to the premeditated endpoint and finally arrive somewhere nearby. Denote the continuous-time destination dependent target state as vector $\mathbf{x}_{t,i}$, then the OU-based models can be described in continuous time by the following stochastic differential equation (SDE),

$$d\mathbf{x}_{t,i} = \mathbf{A}(\boldsymbol{\mu}_i - \mathbf{x}_{t,i})dt + \boldsymbol{\sigma}d\boldsymbol{\beta}_t, \tag{3}$$

where β_t is a multivariate standard Wiener process. For a 3D pointing movement, $\mathbf{x}_{t,i} = \left[\mathbf{x}'_{t,i,1}, \mathbf{x}'_{t,i,2}, \mathbf{x}'_{t,i,3}\right]'$, with $\mathbf{x}_{t,i,s} \in \mathbb{R}^2$ (position and velocity) or $\mathbb{R}^3$ (position, velocity and acceleration), $s = \{1,2,3\}$, and

$$\mathbf{A} = \mathrm{diag}\{\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3\}, \quad \boldsymbol{\mu}_i = \left[\boldsymbol{\mu}_{i,1}, \boldsymbol{\mu}_{i,2}, \boldsymbol{\mu}_{i,3}\right]', \quad \boldsymbol{\sigma} = \begin{bmatrix} \boldsymbol{\sigma}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\sigma}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\sigma}_3 \end{bmatrix}, \boldsymbol{\beta}_t = \left[\beta_{t,1}, \beta_{t,2}, \beta_{t,3}\right]'. \tag{4}$$

Different orders of kinematics included in each "substate" $\mathbf{x}_{t,i,s}$ along with the corresponding parameters lead to distinct SDEs as per (3), for instance: (a) the mean reverting diffusion (MRD) model which only includes position in the state (Ahmad et al., 2016b), (b) equilibrium reverting velocity (ERV) that model position and velocity (Ahmad et al., 2016b), and (c) equilibrium reverting acceleration (ERA) representing position, velocity, and acceleration (Gan et al., 2019). These three models have similar mean reverting behavior, that is, the state will revert to the mean term $\boldsymbol{\mu}_i$, for example set as the destination position for MRD and with (nearly) zero velocity and acceleration for ERV and ERA, respectively.

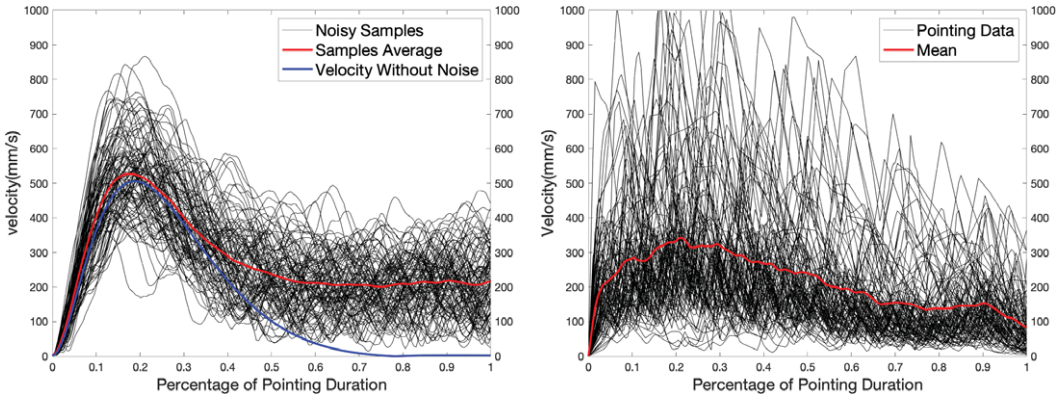
Here we only discuss the set up for ERA model for simplicity, while other models follow the similar rationale, refer to Ahmad et al. (2016b) for further details. For ERA, the submatrices and vectors in Equation (4) for the sth dimension are

$$\mathbf{A}_s = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ \eta & \rho & \gamma \end{bmatrix}, \quad \boldsymbol{\mu}_{i,s} = \left[p_{i,s}, 0, 0\right], \quad \mathbf{x}_{t,i,s} = \left[x_{t,i,s}, \dot{x}_{t,i,s}, \ddot{x}_{t,i,s}\right]', \quad \boldsymbol{\sigma}_s = [0,0,\sigma]', \tag{5}$$

where $p_{i,s}$ is the position of destination $\mathcal{D}_i$ in the sth dimension, and $\dot{x}_{t,i,s}, \ddot{x}_{t,i,s}$ denote the second and third derivative (velocity and acceleration) of $x_{t,i,s}$. The above setup assumes independent transitions for each coordinate, specifically, it can be specified by the following SDE,

$$d\ddot{x}_{t,i,s} = \eta\left(p_{i,s} - x_{t,i,s}\right)dt - \rho\dot{x}_{t,i,s}dt - \gamma\ddot{x}_{t,i,s}dt + \sigma d\beta_{t,s}. \tag{6}$$

One can see that the object motion governed by such an SDE will initially gravitate to the destination position (i.e., $p_{i,s}$ prescribed in the mean vector $\boldsymbol{\mu}_{i,s}$ of this OU process) with increasing acceleration due to the positive reversion factor η, then the positive damping factor ρ and γ would guarantee the target slows



(a) ERA Velocity Samples (b) Real Pointing Data

Figure 2. *The 3D norm velocity profile generated by the equilibrium reverting acceleration (ERA) model is shown in (a), where the black lines are 100 random realizations, the red line is the mean of them, and the blue line shows the deterministic transition of the norm velocity of the same ERA model. (b) Shows the velocity profile from 95 real pointing data, where the red line is the mean trajectory.*

down and arrives the destination in an equilibrium state, with nearly zero velocity and acceleration. This velocity behavior can be demonstrated as the blue line in Figure 2a, which is the deterministic transition (i.e., with σ as zero) of the norm velocity of the ERA model. The norm velocities of an ERA model depicted in Figure 2a, that is sample realizations as well as their mean, are generated from the parameters manually tuned to maximize the intent prediction accuracy. They noticeably capture, on average, an overall profile similar to that exhibited by the real pointing gesture data shown in Figure 2b.

Solving (3) yields the general discrete LTI transition function for all three models (MRD, ERV and ERA) as per,

$$p(\mathbf{x}_{n+1,i}|\mathbf{x}_{n,i}) = \mathcal{N}\left(\mathbf{x}_{n+1,i}|\mathbf{F}_{i,t_{n+1}-t_n}\mathbf{x}_{n,i} + \mathbf{M}_{i,t_{n+1}-t_n}, \mathbf{Q}_{i,t_{n+1}-t_n}\right) \tag{7}$$

such as

$$\begin{aligned} \mathbf{F}_{i,t_{n+1}-t_n} &= e^{-\mathbf{A}(t_{n+1}-t_n)}, \\ \mathbf{M}_{i,t_{n+1}-t_n} &= \left(\mathbf{I} - e^{-\mathbf{A}(t_{n+1}-t_n)}\right)\boldsymbol{\mu}_i, \\ \mathbf{Q}_{i,t_{n+1}-t_n} &= \int_0^{t_{n+1}-t_n} e^{-\mathbf{A}(t_{n+1}-t_n-v)}\boldsymbol{\sigma}\boldsymbol{\sigma}' e^{-\mathbf{A}'(t_{n+1}-t_n-v)} dv. \end{aligned} \tag{8}$$

Whereas, $\mathbf{A}$, $\boldsymbol{\mu}_i$, and $\boldsymbol{\sigma}$ are parameters set for the specific model, $\mathbf{I}$ is the identity matrix with the corresponding size. The derivation of this solution and calculation for $\mathbf{Q}_{i,h}$ can be found in Ahmad et al. (2016b) and references therein. Note that the $\mathbf{x}_{t,i}$ in such models is constructed to revert to the destination $\mathcal{D}_i$, and thus the transition function (7) can be equivalently described as the destination-conditioned transition density, that is, $p(\mathbf{x}_{n+1}|\mathbf{x}_n, \mathcal{D}_i) = p(\mathbf{x}_{n+1,i}|\mathbf{x}_{n,i})$ where $\mathbf{x}_n$ describes the general state (without conditioned-destination information), and the condition $\mathcal{D}_i$ can be further introduced by the destination reverting construction.

2.1.1.2. Bridging distributions. While the destination information is modeled above by the mean of the OU process, another approach to incorporate such knowledge can be provided by the bridging distributions method. This is particularly relevant if we use a known or legacy motion model, which does not encapsulate the influence of intent on the object motion as with numerous models in the tracking literature, for instance the nearly constant velocity (CV) and acceleration (CA) models; see Li and Jilkov (2003) for a

comprehensive overview. Additionally, in some scenarios, an OU process might not accurately characterize the destination reverting behavior of the tracked object. In such cases, BD permits more free underlying motion dynamics, and at the same time, ensures the object arrival at/near its endpoint.

Bridging distributions capture the destination influence on the target behavior by constructing a Markov bridge between the intended endpoint and the target current state at t_n. This capitalizes on the premise that the trajectory followed by the object (e.g., pointing finger) must terminate at the endpoint (on-display selectable interface item), at arrival time $\mathcal{T}$, despite the random behavior between the current time step t_n and $\mathcal{T}$. BD accordingly introduces this knowledge into a motion model via a prior and facilitates destination-aware behavior modeling without requiring the development of specialized stochastic processes that are intrinsically intent-driven. Nonetheless, BD may be applied to OU-type models for means dictated by a destination or not, for endpoint-driven OU process BD can reduce their sensitivity to parameterization as discussed in Ahmad et al. (2018).

Assuming that the target will reach the destination at time $t_N = \mathcal{T}$, a terminal state is defined as $\mathbf{x}_N$. A *bridged* state transition distribution in a Markovian system, which conditions on the destination and the arrival time, can be expressed as the conditional distribution $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathcal{D}_i, \mathcal{T})$. There exists several ways of finding this conditional density and they may differ based on the made assumption(s). For example, Ahmad et al. (2018) assumes the terminal state $\mathbf{x}_N$ has exactly the same position as the destination $\mathcal{D}_i$, and the destination-related information is introduced via a Gaussian prior at t_0, $p(\mathbf{x}_N|\mathcal{D}_i, \mathcal{T}) = \mathcal{N}(\mathbf{x}_N|\mathbf{a}_i, \Sigma_i)$ with $\mathbf{a}_i$ being the mean, Σ_i the covariance matrix and $i = 1, 2, \ldots, N_{\mathcal{D}}$. This covariance can model the size-orientation of the endpoint and hence with BD destinations can be regions rather than single spatial points as with OU-type models. Based on this assumption, the sought transition density $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathcal{D}_i, \mathcal{T})$ is a Markov transition density for the current state ($\mathbf{x}_n$), conditioning on its terminal state ($\mathbf{x}_N$), i.e.,

$$p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{x}_N, \mathcal{T}) \propto p(\mathbf{x}_n|\mathbf{x}_{n-1}) p(\mathbf{x}_N|\mathbf{x}_n, \mathcal{T}). \tag{9}$$

Given the fact that the terminal state $\mathbf{x}_N$ is fixed, one can construct a joint state vector $\mathbf{z}_n = [\mathbf{x}_n, \mathbf{x}_N]'$ and obtain the transition density for $\mathbf{z}_n$ accordingly. The joint state transition will ultimately lead $\mathbf{x}_n$ to its terminal state $\mathbf{x}_N$ which follows the prior $p(\mathbf{x}_N|\mathcal{D}_i, \mathcal{T})$. When observations are available, such a construction of $\mathbf{z}_n$ permits a joint estimation on destination and kinematic state.

An alternative formulation of BD can be found in Liang et al. (2019), in which the destination information is interpreted as a "pseudo-observation" instead of as a state prior. Specifically, a linear and Gaussian pseudo-observation model,

$$p\left(\widetilde{\mathbf{y}}_N^i = \mathbf{a}_i|\mathbf{x}_N\right) = \mathcal{N}\left(\mathbf{a}_i|\widetilde{\mathbf{G}}\mathbf{x}_N, \Sigma_i\right), \tag{10}$$

was considered with $\widetilde{\mathbf{G}}$ being the mapping matrix. It was shown in Liang et al. (2019), Algorithm 2, that this interpretation leads to the following destination-conditioned state transition density,

$$p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathcal{D}_i, \mathcal{T}) \propto \int p\left(\widetilde{\mathbf{y}}_N^i = \mathbf{a}_i|\mathbf{x}_N\right) p(\mathbf{x}_N|\mathbf{x}_n, \mathcal{T}) p(\mathbf{x}_n|\mathbf{x}_{n-1}) d\mathbf{x}_N, \tag{11}$$

where the Markovian assumption is preserved between the terminal state and the initial state.

Motivated by the pseudo-observation–based formulation of BD, in this paper we introduce a new intent prediction algorithm which utilizes (11) as its main ingredient. Similar to Ahmad et al. (2018) and Liang et al. (2019), we will focus on linear Gaussian models because they lead to analytically tractable results. First, consider the following LTI SDE, where $\mathbf{x}_t = [x_t, \dot{x}_t, \ddot{x}_t, y_t, \dot{y}_t, \ddot{y}_t, z_t, \dot{z}_t, \ddot{z}_t]'$ has the same physical meaning as in Section 2.1.1 (i.e., position, velocity and acceleration in 3D Cartesian coordinates),

$$d\mathbf{x}_t = \mathbf{A}\mathbf{x}_t dt + \boldsymbol{\sigma} d\boldsymbol{\beta}_t, \tag{12}$$

with $\mathbf{A}_s = [0,1,0;0,0,1;0,0,0]$ (see again Section 2.1.1 for further details related to the noise components). It can be shown that the transition density resulting from this SDE is of the form:

$$p(\mathbf{x}_n|\mathbf{x}_{n-1}) = \mathcal{N}(\mathbf{x}_n|\mathbf{F}_h\mathbf{x}_{n-1}, \mathbf{Q}_h), \tag{13}$$

with $\mathbf{F}_h$ being the state transition matrix, $\mathbf{Q}_h$ the process noise covariance and $h = t_n - t_{n-1}$. In comparison to (7), this transition density has no dependency on a destination. When the process noise level is relatively low, (13) corresponds to the nearly CA model (also known as the Wiener-process acceleration model). Substituting (13) into (11) yields

$$\begin{aligned} p(\mathbf{x}_n|\mathbf{x}_{n-1},\mathcal{D}_i,\mathcal{T}) &\propto \int \mathcal{N}\left(\mathbf{a}_i|\widetilde{\mathbf{G}}\mathbf{x}_N,\Sigma_i\right)\mathcal{N}\left(\mathbf{x}_N|\mathbf{F}_{\mathcal{T}-t_n}\mathbf{x}_n,\mathbf{Q}_{\mathcal{T}-t_n}\right)\mathcal{N}(\mathbf{x}_n|\mathbf{F}_h\mathbf{x}_{n-1},\mathbf{Q}_h)d\mathbf{x}_N, \\ &\propto \mathcal{N}\left(\mathbf{a}_i|\widetilde{\mathbf{G}}\mathbf{F}_{\mathcal{T}-t_n}\mathbf{x}_n,\widetilde{\mathbf{G}}\mathbf{Q}_{\mathcal{T}-t_n}\widetilde{\mathbf{G}}' + \Sigma_i\right)\mathcal{N}(\mathbf{x}_n|\mathbf{F}_h\mathbf{x}_{n-1},\mathbf{Q}_h) \\ &= \mathcal{N}\left(\mathbf{x}_n|\mathbf{F}_{i,h}\mathbf{x}_{n-1}+\mathbf{M}_{i,h},\mathbf{Q}_{i,h}\right), \end{aligned} \tag{14}$$

where

$$\mathbf{F}_{i,h} = \mathbf{Q}_{i,h}\mathbf{Q}_h^{-1}\mathbf{F}_h,\ \mathbf{M}_{i,h} = \mathbf{Q}_{i,h}\mathbf{L}\mathbf{a}_i,\ \mathbf{Q}_{i,h} = \left(\mathbf{Q}_h^{-1} + \mathbf{L}\widetilde{\mathbf{G}}\mathbf{F}_{\mathcal{T}-t_n}\right)^{-1},\ \mathbf{L} = \left(\widetilde{\mathbf{G}}\mathbf{F}_{\mathcal{T}-t_n}\right)'\left(\widetilde{\mathbf{G}}\mathbf{Q}_{\mathcal{T}-t_n}\widetilde{\mathbf{G}}' + \Sigma_i\right)^{-1}.$$

Here (14) serves as the transition density for the pseudo-observation process, that is satisfies $p(\mathbf{x}_n|\mathcal{D}_i,\mathcal{T})$, based on which the state will evolve under the guidance of destination information. Figure 3 gives an example of the marginal distributions obtained according to the above pseudo-observation based process where the influence of the endpoint on the state distribution over time is evident. It can also be shown that the limiting distribution, $\lim_{t_N=\mathcal{T}\to\infty} p(\mathbf{x}_N|\mathcal{D}_i,\mathcal{T})$, of a state process having (14) as its transition density equates to $\mathcal{N}(\mathbf{a}_i,\Sigma_i)$ when $\widetilde{\mathbf{G}} = \mathbf{I}$. Moreover, setting $\widetilde{\mathbf{G}} = \mathbf{I}$ and $\Sigma_i = \mathbf{0}$ produces the same state transition density as (9), namely a canonical Gaussian bridge (Gasbarra et al., 2007) terminated at a certain state, with the fact that the endpoint $\mathbf{x}_N$ is certain. It should be stressed that the form of mapping matrix $\widetilde{\mathbf{G}}$ depends on what destination-related information is available at hand and thus it is not necessarily equal to an identity matrix; any such matrix is included in (14).

The state transition distributions in Equations (9) and (11) build the destination knowledge into the state dynamics and thus form the basis of BD-based destination-driven (or destination-constrained) motion models. For all nominal destinations $\mathcal{D}_i \in \mathbb{D}$, $N_{\mathcal{D}}$ such bridges are constructed, one per endpoint. In scenarios where we want the terminal state $\mathbf{x}_N$ at t_N as well as $\mathbf{x}_n$ at the current time step t_n to be jointly estimated, the transition model prescribed by (9) may be utilized. However, if the main objective is to predict the intended destination as in this paper with available information on the nominal endpoints (e.g., a certain region/area represented by an ellipsoidal shape), (11) can be used to construct a computationally efficient predictor since the hidden state dimension in this case is less than that of the joint estimation scheme (i.e., includes $\mathbf{x}_N$). In Section 3.1.2, we present a new intent predictor based on the destination-constrained prior as with (11). In comparison to Ahmad et al. (2016b), the new predictor requires less computations as it does not estimate the terminal state at t_N. It is constructed using pseudo-observation and therefore the underlying state process is still a Markov process. It also differs from the pseudo-observation

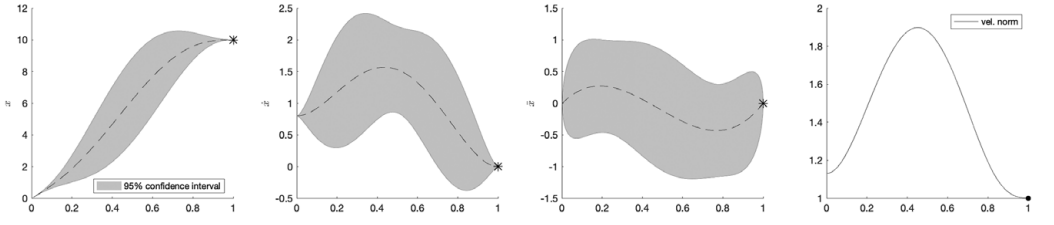


Figure 3. *1D (along x-axis in a Cartesian coordinate) distributions of a pseudo-observation based bridging distributions-constant acceleration process (with $\widetilde{G} = \mathbf{I}$, $\Sigma_i = \mathbf{0}$); in this case, the distribution at the endpoint (asterisk) reduces to $p(\mathbf{x}_N|D_i,\mathcal{T}) = \delta_{\mathbf{a}_i}(\mathbf{x}_N)$ with $\delta_{(\cdot)}$ being the Dirac delta function. From left to right: (a). $p(x_n|\mathcal{D}_i,\mathcal{T})$; (b). $p(\dot{x}_n|\mathcal{D}_i,\mathcal{T})$; (c). $p(\ddot{x}_n|\mathcal{D}_i,\mathcal{T})$; and (d). velocity norm. Horizontal axes are time (in percentage) and dashed lines are distribution means.*

based intent predictors presented in Liang et al. (2019) in that it utilizes a destination-constrained state transition density throughout the filtering procedure (although this implies a slightly higher computational burden). Finally, a pseudo-measurement technique for jointly estimating the object state and its destination is presented in Zhou et al. (2020) based on a linear equality constraint. It dictates that the object follows some straight line to its intended endpoint. Although this simplifies the inference procedure as the condition of arrival time is avoided, it does not capture realistic motion behavior of several objects of interest (e.g., constraint-free pointing motion in 3D). On the contrary, the presented stochastic modeling is general and does not impose such restrictive constraints on the target trajectory.

2.1.2. Nonlinear motion models: conditionally linear Gaussian settings

The computationally efficient Gaussian model assumes that the change in the object motion (i.e., pointing movements) in any time interval always follows a Gaussian distribution. However, for some irregular movements which cause rapid spatial changes (e.g., jolts in the pointing motion due to perturbations or any external nonintent-driven force), such an assumption is unsuitable and can lead to large inference errors. In order to model such erratic perturbations-induced maneuvers, we introduce a pure jump process to the original (destination-aware) Gaussian processes. Such formulations are known as jump diffusion models or Markov/semi-Markov jump models.

The adopted jump diffusion models retain the Brownian motion as one of the driven noise, and thus they can be considered as a conditionally linear Gaussian system. In particular, when the non-Gaussian pure jump process is given as a condition, the dynamics can be constructed in a standard Gaussian form to ensure computational efficiency.

Such approaches have been extensively adopted in financial modeling to describe the discrete movements (Kou 2002), and in object tracking field to capture sudden maneuvers undertaken by the target or induced by external forces (Godsill, 2007). Owing to the clear physical representation and computation tractability, such jump diffusion dynamical models have also been employed in Ahmad et al. (2014) and Gan et al. (2019) within a predictive touch system under high levels of perturbations due to road-driving conditions. The approach presented in Ahmad et al. (2014) embedded a self-decay jump process within a Gaussian process to pre-process the highly-perturbed pointing data, with the aim to obtain a smoothed trajectory for the later intent inference task, whereas Gan et al. (2019) introduces a jump diffusion model for a unified scheme for destination and state estimation. In this paper, we mainly discuss the latter recent work given its improved performance.

Since the target motion (e.g., pointing gesture movements) impacted by severe external perturbations or fast maneuvering is still destination reverting, we consider the jump diffusion model based on the following linear mean reverting SDE (3)

$$d\mathbf{x}_{t,i} = \mathbf{A}(\boldsymbol{\mu}_i - \mathbf{x}_{t,i})dt + \boldsymbol{\sigma} d\boldsymbol{\beta}_t + \mathbf{B} d\mathbf{J}_t, \tag{15}$$

where most parameters have the same definition as described in the previous sections. If we assume that the jumps only occur at key driving elements of the state (e.g., position for MRD, or acceleration in ERA), the parameter $\mathbf{B} = \text{diag}\{\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3\}$ (for 3D movements) such that $\mathbf{B}_s = [0,0,1]'$ $(s = 1,2,3)$ for ERA and $[0,1]'$ for ERV. The multivariate jump process $\mathbf{J}_t$ here is a compound Poisson process with Gaussian distributed jump size. Specifically, we have $\mathbf{J}_t = \sum_{\tau_k < t} \mathbf{S}_k$, with the jump size $\mathbf{S}_k \in \mathbb{R}^3$ and $\mathbf{S}_k \sim (\mathbf{S}_k | \boldsymbol{\mu}_J, \Sigma_J)$. Note that if isotropic distributed jump (i.e., the jump on each direction of the space are identically distributed) is considered, the parameters can be simplified as $\boldsymbol{\mu}_J = \mathbf{0}$ and $\Sigma_J = \sigma_J^2 \mathbf{I}$, where σ_J is defined as the standard deviation of the jump size in any dimension. The jump time τ_k which follows the Poisson process has the property that $\tau_k - \tau_{k-1} \sim \exp_{\lambda_J}(\cdot)$, where λ_J^{-1} is the mean value of the jump interarrival time.

Solving SDE (15) yields the transition density as follows,

$$p(\mathbf{x}_{n+1,i} | \mathbf{x}_{n,i}, \tau_{n:n+1}) = \mathcal{N}\left(\mathbf{x}_{n+1,i} | \boldsymbol{\mu}^*_{n+1}, \Sigma^*_{n+1}\right), \tag{16}$$

with

$$\boldsymbol{\mu}^*_{n+1} = \mathbf{F}_{i,t_{n+1}-t_n}\mathbf{x}_t + \mathbf{M}_{i,t_{n+1}-t_n} + \sum_{t_n<\tau_k\leq t_{n+1}} \mathbf{F}_{i,t_{n+1}-\tau_k}\mathbf{B}\boldsymbol{\mu}_J, \tag{17}$$

$$\Sigma^*_{n+1} = \mathbf{Q}_{i,t_{n+1}-t_n} + \sum_{t_n<\tau_k\leq t_{n+1}} \mathbf{F}_{i,t_{n+1}-\tau_k}\mathbf{B}\Sigma_J\mathbf{B}'\mathbf{F}'_{i,t_{n+1}-\tau_k}{}', \tag{18}$$

where $\mathbf{F}$,$\mathbf{M}$ and $\mathbf{Q}$ have been defined in (8), and jump time sequence $\tau_{n:n+1}$ consists of all jump times that occurred in the interval $(t_n, t_{n+1}]$, that is $\tau_{n:n+1} = \bigcup_{t_n<\tau_k\leq t_{n+1}} \tau_k$.

2.1.3. Observation model

The available sensory measurement $\mathbf{y}_n$ (e.g., gesture-tracker output) is a noisy observation of the true hidden state $\mathbf{x}_n$ (e.g., pointing finger actual location). In a state space form, it is described at time t_n by

$$\mathbf{y}_n = \mathbf{h}_n(\mathbf{x}_{n,i}) + \mathbf{w}_n, \tag{19}$$

where $\mathbf{h}_n(.)$ is the mapping from the hidden state to the observed measurement(s) and $\mathbf{w}_n$ is the measurement noise. Here and for simplicity, a linear and Gaussian measurement model can be assumed such that $\mathbf{y}_n = \mathbf{H}\mathbf{x}_{n,i} + \mathbf{w}_n$, with zero mean i.i.d Gaussian noise where $\mathbf{w}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{V}_n)$. For instance, if gesture tracker provides locations of the pointing finger in 3D and latent state $\mathbf{x}_{n,i}\in\mathbb{R}^3$ consists of the object location, the mapping measurement matrix in (19) is a 3×3 identity matrix, $\mathbf{H} = \mathbf{I}_3$. The noise covariance matrix $\mathbf{V}_n$ is specified by the tracker accuracy, that is in terms of determining the pointing finger position.

The overall system is described by the motion and observation models in (2) and (19), respectively. Next, we introduce various destination inference algorithms to estimate the sought probabilities $p(\mathcal{D} = \mathcal{D}_i|\mathbf{y}_{1:n}), \mathcal{D}_i\in\mathbb{D}$. As shown below, the intent inference routine complexity is dependent on the employed motion model. For instance, a Gaussian LTI set-up leads to a simple and computationally efficient Kalman-filer-based predictor for the destination inference task.

3. Destination Prediction

Recall from (1) that the key to sequentially infer the probability of the destination $\mathcal{D}_i$ being the intended one is to estimate the likelihood $p(\mathbf{y}_{0:n}|\mathcal{D} = \mathcal{D}_i)$. Furthermore, this likelihood can be recursively expanded according to *prediction error decomposition* (PED; Harvey, 1990) given by

$$p(\mathbf{y}_{0:n}|\mathcal{D}_i) = p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathcal{D}_i)p(\mathbf{y}_{0:n-1}|\mathcal{D}_i), \tag{20}$$

where we have abbreviated the condition $\mathcal{D} = \mathcal{D}_i$ as $\mathcal{D}_i$ henceforth to simplify notation. This sequential likelihood estimation serves as the basis of online Bayesian intent predictor as it only requires the evaluation of predictive likelihood $p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathcal{D}_i)$ at each time instant. In this section, we discuss the strategy to compute this predictive likelihood for the various models introduced in Section 2.

3.1. LTI Gaussian systems

The destination reverting models in Section 2.1.1 are devised in a Gaussian LTI form, which leads to linear Gaussian transition densities. Meanwhile, a linear Gaussian observation model (e.g., for an off-the-shelf gesture tracker) is assumed for (19). The standard Kalman filter is then sufficient to carry out the recursive filtering for intent inference, namely to produce the (optimal in the mean least squares error sense) PED (Haug, 2012), rather than the conventional state estimation task as shown next.

3.1.1. OU-based intent predictors

Recall that the destination conditioned transition function for OU-based model, for example in (7), and the adopted observation function are both linear Gaussian, the estimated target state can thus be explicitly described by a normal distribution. Specifically,

$$p(\mathbf{x}_n|\mathbf{y}_{1:n},\mathcal{D}_i)=\mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_{n|n},\mathbf{C}_{n|n}\right), \tag{21}$$

$$p(\mathbf{x}_n|\mathbf{y}_{1:n-1},\mathcal{D}_i)=\mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}_{n|n-1},\mathbf{C}_{n|n-1}\right). \tag{22}$$

The predictive likelihood can be computed as follows,

$$p(\mathbf{y}_n|\mathbf{y}_{1:n-1},\mathcal{D}_i)=\int p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{1:n-1},\mathcal{D}_i)d\mathbf{x}_n, \tag{23}$$

and this leads to a Gaussian density description $p(\mathbf{y}_n|\mathbf{y}_{1:n-1},\mathcal{D}_i)=\mathcal{N}\left(\mathbf{y}_n|\boldsymbol{\mu}_{y_n},\mathbf{C}_{y_n}\right)$, where

$$\boldsymbol{\mu}_{y_n}=\mathbf{H}\boldsymbol{\mu}_{n|n-1}, \tag{24}$$

$$\mathbf{C}_{y_n}=\mathbf{H}\mathbf{C}_{n|n-1}\mathbf{H}'+\mathbf{V}_n. \tag{25}$$

To compute $\boldsymbol{\mu}_{n|n-1}$ and $\mathbf{C}_{n|n-1}$ at each time step, the standard Kalman filter is required to estimate the state recursively, summarized as follows,

$$p(\mathbf{x}_n|\mathbf{y}_{1:n-1},\mathcal{D}_i)=\int p(\mathbf{x}_{n-1}|\mathbf{y}_{1:n-1},\mathcal{D}_i)p(\mathbf{x}_n|\mathbf{x}_{n-1},\mathcal{D}_i)d\mathbf{x}_{n-1}, \tag{26}$$

$$p(\mathbf{x}_n|\mathbf{y}_{1:n},\mathcal{D}_i)\propto p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{1:n-1},\mathcal{D}_i). \tag{27}$$

The corresponding matrix description is

$$\boldsymbol{\mu}_{n|n-1}=\mathbf{F}_{i,h}\boldsymbol{\mu}_{n-1|n-1}+\mathbf{M}_{i,h}, \tag{28}$$

$$\mathbf{C}_{n|n-1}=\mathbf{F}_{i,h}\mathbf{C}_{n-1|n-1}\mathbf{F}'_{i,h}+\mathbf{Q}_{i,h}, \tag{29}$$

$$\mathbf{K}=\mathbf{C}_{n|n-1}\mathbf{H}'\left(\mathbf{C}_{\mathbf{y}_n}\right)^{-1}, \tag{30}$$

$$\boldsymbol{\mu}_{n|n}=\boldsymbol{\mu}_{n|n-1}+\mathbf{K}\left(\mathbf{y}_n-\boldsymbol{\mu}_{y_n}\right), \tag{31}$$

$$\mathbf{C}_{n|n}=(\mathbf{I}-\mathbf{K}\mathbf{H})\mathbf{C}_{n|n-1}. \tag{32}$$

The above equations specify the the computation of predictive likelihood for a single time step, the likelihood for each destination being the intended one can then be evaluated with (1) and (20).

3.1.2. BD-based intent predictor using pseudo-observation formulation

In principle, BD-based intent predictors, including those in Ahmad et al. (2018) and Liang et al. (2019) and the new approach introduced here, all utilize (1) and (20) for inferring the target destination from the available noisy sensory observations. However, for the BD approach proposed here, we have $p(\mathbf{y}_{0:n}|\mathcal{D}_i,\mathcal{T})=p(\mathbf{y}_{0:n}|\Theta_i,\mathcal{T})$ with Θ_i containing destination-specific parameters (here, $\mathbf{a}_i$ and Σ_i) and $\mathcal{T}$ being the arrival time at the destination. As the likelihood term is further conditioning on an unknown arrival time $\mathcal{T}$, Equation (20) needs to be revised as follows:

$$p(\mathbf{y}_{0:n}|\mathcal{D}_i,\mathcal{T})=p(\mathbf{y}_n|\mathbf{y}_{0:n-1},\mathcal{D}_i,\mathcal{T})p(\mathbf{y}_{0:n-1}|\mathcal{D}_i,\mathcal{T}), \tag{33}$$

based on which $p(\mathbf{y}_{0:n}|\mathcal{D}_i)$ can be obtained via

$$p(\mathbf{y}_{0:n}|\mathcal{D}_i)=\int p(\mathbf{y}_{0:n}|\mathcal{D}_i,\mathcal{T})p(\mathcal{T}|\mathcal{D}_i)d\mathcal{T}, \tag{34}$$

where $p(\mathcal{T}|\mathcal{D}_i)$ is the prior distribution on the unknown arrival time. In general, the above integration is not analytically tractable and numerical approximation can be implemented. This is especially viable

since the arrival time is a one-dimensional quantity (and thereby the integral). In this paper, we will adopt the same quadrature approximation scheme as in Ahmad et al. (2018) for obtaining (34).

Henceforth, the aim is to compute the arrival-time-conditioned PED and likelihood (i.e., the *unknown* arrival time is treated as if it is available). We illustrate how to develop an intent predictor based on the destination-constrained state process defined in Section 2.1.1. Given observations up to t_n, the $\mathcal{T}$-conditioned likelihood term of interest can be expressed by

$$p(\mathbf{y}_n|\mathbf{y}_{0:n-1},\mathcal{D}_i,\mathcal{T}) = \int p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1},\mathcal{D}_i,\mathcal{T})p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1},\mathcal{D}_i,\mathcal{T})d\mathbf{x}_{n-1}d\mathbf{x}_n, \tag{35}$$

where the first component in the integral is the observation density, the second component is the destination-constrained state transition density as defined in (11) and the last component is a filtering distribution obtained at time t_{n-1}. Next, we outline how to calculate $p(\mathbf{y}_n|\mathbf{y}_{0:n-1},\mathcal{D}_i,\mathcal{T})$ at each time step for a linear and Gaussian dynamic system. For simplicity and without loss of generality, we use the same state model as with (13) with destination information incorporated via (10). This implies the availability of the destination-conditioned state transition density in Equation (14). With a linear Gaussian observation model, we have

$$p(\mathbf{y}_n|\mathbf{x}_n) = \mathcal{N}(\mathbf{y}_n|\mathbf{H}\mathbf{x}_n,\mathbf{V}_n), \tag{36}$$

where $\mathbf{H}$ is the observation matrix and $\mathbf{V}_n$ is the measurement noise covariance matrix. As a result, the filtering distribution $p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1},\mathcal{D}_i,\mathcal{T})$ at the previous time step t_{n-1} can be obtained using a standard Kalman filter in which (14) is used as the state transition density. Assuming at t_n we have obtained the filtering distribution given by the Kalman filter associated with $\mathcal{D}_i$ from the last time step t_{n-1} as

$$p(\mathbf{x}_{n-1}|\mathbf{y}_{0:n-1},\mathcal{D}_i,\mathcal{T}) = \mathcal{N}\left(\mathbf{x}_{n-1}|\boldsymbol{\mu}^i_{n-1|n-1},\mathbf{C}^i_{n-1|n-1}\right), \tag{37}$$

with $\boldsymbol{\mu}^i_{n-1|n-1}$ and $\mathbf{C}^i_{n-1|n-1}$ being the mean and covariance respectively, and substituting (36), (14) and (37) into (35), the sought likelihood can be shown to be

$$\begin{aligned} p(\mathbf{y}_n|\mathbf{y}_{0:n-1},\mathcal{D}_i,\mathcal{T}) &= \int \mathcal{N}(\mathbf{y}_n|\mathbf{H}\mathbf{x}_n,\mathbf{V}_n)\mathcal{N}\left(\mathbf{x}_n|\mathbf{F}_{i,h}\mathbf{x}_{n-1}+\mathbf{M}_{i,h},\mathbf{Q}_{i,h}\right) \\ &\quad \times\mathcal{N}\left(\mathbf{x}_{n-1}|\boldsymbol{\mu}^i_{n-1|n-1},\mathbf{C}^i_{n-1|n-1}\right)d\mathbf{x}_{n-1}d\mathbf{x}_n \\ &= \mathcal{N}\left(\mathbf{y}_n|\mathbf{H}\left(\mathbf{F}_{i,h}\boldsymbol{\mu}^i_{n-1|n-1}+\mathbf{M}_{i,h}\right),\ \mathbf{H}\left(\mathbf{F}_{i,h}\mathbf{C}^i_{n-1|n-1}\mathbf{F}'_{i,h}+\mathbf{Q}_{i,h}\right)\mathbf{H}'+\mathbf{V}_n\right). \end{aligned} \tag{38}$$

The above calculation can be further simplified by noticing that

$$\begin{aligned} \boldsymbol{\mu}^i_{n|n-1} &= \mathbf{F}_{i,h}\boldsymbol{\mu}^i_{n-1|n-1}+\mathbf{M}_{i,h} \\ \mathbf{C}^i_{n|n-1} &= \mathbf{F}_{i,h}\mathbf{C}^i_{n-1|n-1}\mathbf{F}'_{i,h}+\mathbf{Q}_{i,h} \end{aligned} \tag{39}$$

are actually the mean and covariance of the intermediate distribution $p(\mathbf{x}_n|\mathbf{y}_{0:n-1},\mathcal{D}_i,\mathcal{T}) = \mathcal{N}\left(\mathbf{x}_{n-1}|\boldsymbol{\mu}^i_{n|n-1},\mathbf{C}^i_{n|n-1}\right)$ obtained at the Kalman prediction step. As a result, there is no need to re-calculate these two quantities twice.

Combining (33), (38), and (34), $p(\mathbf{y}_{0:n}|\mathcal{D}_i)$ can be evaluated sequentially when new measurements become available. To complete the intent prediction algorithm, the above calculation needs to be performed for each destination $\mathcal{D}_i \in \mathbb{D}$. Furthermore, when a quadrature approximation scheme is used, (38) needs to be evaluated at each quadrature point of $\mathbb{T} = \{\mathcal{T}_q, q = 1,2,\ldots,N_{\mathcal{T}}\}$. A detailed implementation note is summarized in Algorithm I. It is noted that a guidance on the choice number of quadrature points for BD methods can be found in Ahmad et al. (2018).

Algorithm I BD-based Intent Predictor

Input: Observations: $\{\mathbf{y}_{0:N}\}$, Pseudo-observations: $\{\mathbf{a}_i,\Sigma_i\}_{1\le i\le N_\mathcal{D}}$, $\mathbb{T}=\{\mathcal{T}_q\}_{1\le q\le N_\mathcal{T}}$;

Initialization: $N_\mathcal{D}\times N_\mathcal{T}$ Kalman filters, each initialized with mean $\boldsymbol{\mu}^{i,q}_{-1|-1}$ and covariance $\mathbf{C}^{i,q}_{-1|-1}$.

for $n=0:N$ **do** ▷ For each time instant

for $\mathcal{D}_i\in\mathbb{D}$ **do** ▷ For each destination

for $\mathcal{T}_q\in\mathbb{T}$ **do** ▷ For each quadrature point

Construct the intent-driven transition density $p(\mathbf{x}_n|\mathbf{x}_{n-1},\mathcal{D}_i,\mathcal{T}_q)$ via (14);

Standard Kalman prediction to obtain $\boldsymbol{\mu}^{i,q}_{n|n-1}$ and $\mathbf{C}^{i,q}_{n|n-1}$ via (39);

Standard Kalman update to obtain $\boldsymbol{\mu}^{i,q}_{n|n}$ and $\mathbf{C}^{i,q}_{n|n}$;

Compute: $l^{i,q}_n=p(\mathbf{y}_n|\mathbf{y}_{0:n-1},\mathcal{D}_i,\mathcal{T}_q)$ via (38);

Update $\mathcal{T}_q$-conditioned likelihood via (33): $p(\mathbf{y}_{0:n}|\mathcal{D}_i,\mathcal{T}_q)=L^{i,q}_n=L^{i,q}_{n-1}\times l^{i,q}_n$

end for

Approximate $p(\mathbf{y}_{0:n}|\mathcal{D}_i)$ numerically using $\{L^{i,q}_n,q=1,2,\ldots,N_\mathcal{T}\}$;

end for

Obtain destination posterior at t_n: $p(\mathcal{D}_i|\mathbf{y}_{0:n})\approx\frac{p(\mathbf{y}_{0:n}|\mathcal{D}_i)\times p(\mathcal{D}_i)}{\sum_{\mathcal{D}_j\in\mathbb{D}}p(\mathbf{y}_{0:n}|\mathcal{D}_j)\times p(\mathcal{D}_j)}$;

end for

3.2. Intent predictors for jump diffusion models

The jump diffusion model introduced in Section 2.1.2 is constructed as a conditionally Gaussian form (16), that is, the transition density from time t to $t+h$ is a Gaussian density if the nonlinear component jump time sequence $\tau_{t:t+h}$ is given as a condition. Thus an efficient strategy would be estimating $\tau_{t:t+h}$ in a Monte Carlo sense, then for each sample of $\tau_{t:t+h}$, $p(\mathbf{x}_{t+h,i}|\mathbf{x}_{t,i},\tau_{t:t+h})$ is retained as Gaussian form so that the standard Kalman filter can be employed to carry out the estimation. Such strategy, known as Rao-Blackwellized variable rate particle filter (Godsill, 2007; Christensen et al., 2012), aims to strengthen the estimation accuracy by employing analytical computations as much as possible.

When $N_\mathcal{D}$ possible destinations are considered, the same number of particle filters are required, each with $N_\mathcal{P}$ particles for a particular destination $\mathcal{D}_i$. Here, we allow the $N_\mathcal{D}$ different particle filters to share the same sample set of jump times. This not only reduces the inference computational complexity, but can also circumvent spurious large differences between the likelihoods of the various destinations, induced by individual sample outlier(s). Nonetheless, this particular consideration is not expected to noticeably impact the intent prediction performance since the aim in this paper is *not* to accurately estimate the object state or the individual destination likelihood $p(\mathbf{y}_{0:n}|\mathcal{D}_i)$. Instead, the focus is on comparing the likelihoods for all nominal destinations, calculated under the same conditions, in order to determine the intended endpoint from the observed motion. At time t_n, each variable rate particle filter stores the samples $\tau^{(p)}_{0:n}$ $(p=1,2,\ldots,N_\mathcal{P})$, the *normalized* weight $\omega^{(p,i)}_n$, the mean $\boldsymbol{\mu}^{(p,i)}_{n|n}$ and covariance $\mathbf{C}^{(p,i)}_{n|n}$ for Gaussian density $p\left(\mathbf{x}_n|\mathbf{y}_{0:n},\tau^{(p)}_{0:n},\mathcal{D}_i\right)$. Subsequently, the empirical estimations for jump time and state can be described as follows,

$$p(\tau_{0:n}|\mathbf{y}_{0:n},\mathcal{D}_i)\approx\sum_{p=1}^{N_\mathcal{P}}\omega^{(p,i)}_n\delta_{\tau^{(p)}_{0:n}}(\tau_{0:n}), \tag{40}$$

$$p(\mathbf{x}_n|\mathbf{y}_{0:n},\mathcal{D}_i)\approx\sum_{p=1}^{N_\mathcal{P}}\omega^{(p,i)}_n\mathcal{N}\left(\mathbf{x}_n|\boldsymbol{\mu}^{(p,i)}_{n|n},\mathbf{C}^{(p,i)}_{n|n}\right). \tag{41}$$

Accordingly, the predictive likelihood can be approximated as

$$\begin{aligned} p(\mathbf{y}_{n+1}|\mathbf{y}_{0:n},\mathcal{D}_i) &= \int p(\mathbf{y}_{n+1}|\mathbf{y}_{0:n},\tau_{0:n+1},\mathcal{D}_i)p(\tau_{n:n+1}|\tau_{0:n})p(\tau_{0:n}|\mathbf{y}_{0:n},\mathcal{D}_i)d\tau_{0:n+1} \\ &\approx \sum_{p=1}^{N_\mathcal{P}} \widetilde{\omega}_{n+1}^{(p,i)}, \end{aligned} \tag{42}$$

where the *updated* weight $\widetilde{\omega}_{n+1}^{(p,i)}$, in the bootstrap particle filter setting, is defined as

$$\widetilde{\omega}_{n+1}^{(p,i)} = \omega_n^{(p,i)} p\left(\mathbf{y}_{n+1}|\mathbf{y}_{0:n},\tau_{0:n+1}^{(p)},\mathcal{D}_i\right), \tag{43}$$

and the new jump time samples $\tau_{n:n+1}^{(p)}$, in the corresponding (bootstrap) setup, are propagated according to the Poisson transition described in Section 2.1.2,

$$\tau_{n:n+1}^{(p)} \sim p\left(\tau_{n:n+1}|\tau_{0:n}^{(p)}\right). \tag{44}$$

It can be shown that the updated weight $\widetilde{\omega}_{n+1}^{(p,i)}$ is also required to compute the *normalized* weight $\omega_{n+1}^{(p,i)}$:

$$\omega_{n+1}^{(p,i)} = \frac{\widetilde{\omega}_{n+1}^{(p,i)}}{\sum_{p=1}^{N_\mathcal{P}} \widetilde{\omega}_{n+1}^{(p,i)}}. \tag{45}$$

Similar to (23), the $p\left(\mathbf{y}_{n+1}|\mathbf{y}_{0:n},\tau_{0:n+1}^{(p)},\mathcal{D}_i\right)$ in (43) can be computed in a closed form with the stored mean $\boldsymbol{\mu}_{n|n}^{(p,i)}$ and covariance $\mathbf{C}_{n|n}^{(p,i)}$, that is

$$p\left(\mathbf{y}_{n+1}|\mathbf{y}_{0:n},\tau_{0:n+1}^{(p)},\mathcal{D}_i\right) = \mathcal{N}\left(\mathbf{y}_{n+1}|\mathbf{H}\boldsymbol{\mu}_{n+1|n}^{(p,i)},\mathbf{H}\mathbf{C}_{n+1|n}^{(p,i)}\mathbf{H}' + \mathbf{V}_n\right), \tag{46}$$

where

$$\begin{aligned} \boldsymbol{\mu}_{n+1|n}^{(p,i)} &= \mathbf{F}_{i,t_{n+1}-t_n}\boldsymbol{\mu}_{n|n}^{(p,i)} + \mathbf{M}_{i,t_{n+1}-t_n} + \sum_{t_n<\tau_k^{(p)}\le t_{n+1}} \mathbf{F}_{i,t_{n+1}-\tau_k^{(p)}}\mathbf{B}\mu_J, \\ \mathbf{C}_{n+1|n}^{(p,i)} &= \mathbf{F}_{i,t_{n+1}-t_n}\mathbf{C}_{n|n}^{(p,i)}\mathbf{F}'_{i,t_{n+1}-t_n} + \sum_{t_n<\tau_k^{(p)}\le t_{n+1}} \mathbf{F}_{i,t_{n+1}-\tau_k^{(p)}}\mathbf{B}\boldsymbol{\Sigma}_J\mathbf{B}'\mathbf{F}'_{i,t_{n+1}-\tau_k^{(p)}} + \mathbf{Q}_{i,t_{n+1}-t_n}. \end{aligned} \tag{47}$$

In order to updated the stored density mean $\boldsymbol{\mu}_{n+1|n+1}^{(p,i)}$ and covariance $\mathbf{C}_{n+1|n+1}^{(p,i)}$, the following standard Kalman filter updated steps are required:

$$\begin{aligned} \boldsymbol{\mu}_{n+1|n+1}^{(p,i)} &= \boldsymbol{\mu}_{n+1|n}^{(p,i)} + \mathbf{K}\left(\mathbf{y}_n - \mathbf{H}\boldsymbol{\mu}_{n+1|n}^{(p,i)}\right), \\ \mathbf{C}_{n+1|n+1}^{(p,i)} &= (\mathbf{I}-\mathbf{K}\mathbf{H})\mathbf{C}_{n+1|n}^{(p,i)}, \\ \mathbf{K} &= \mathbf{C}_{n+1|n}^{(p,i)}\mathbf{H}'\left(\mathbf{H}\mathbf{C}_{n+1|n}^{(p,i)}\mathbf{H}' + \mathbf{V}_n\right)^{-1}. \end{aligned} \tag{48}$$

This procedure completes the variable rate particle filtering for a single time step and the overall intent prediction algorithm is summarized as Algorithm II.

Algorithm II Intent Inference with the jump model

Initialization: Create $N_\mathcal{D}$ variable rate particle filters, each with $N_\mathcal{P}$ particles;
for each observations $n = 1:N$ captured at t_n **do**
for particles $p = 1:N_\mathcal{P}$ **do**
Sample the jump time sequence from prior $\tau_{n:n+1}^{(p)}$ from (44);

end for
for destinations $i = 1 : N_{\mathcal{D}}$ **do**
if Resample **then**
Resample particles and set weights $\omega_{n-1}^{(p,i)} = 1/N_{\mathcal{P}}$;
end if
for particles $p = 1 : N_{\mathcal{P}}$ **do**
Predict the mean $\boldsymbol{\mu}_{n+1|n}^{(p,i)}$ and covariance $\mathbf{C}_{n+1|n}^{(p,i)}$ via (47);
Calculate the *updated* weight $\widetilde{\omega}_{n+1}^{(p,i)}$ according to (43)(46);
Update the mean $\boldsymbol{\mu}_{n+1|n+1}^{(p,i)}$ and covariance $\mathbf{C}_{n+1|n+1}^{(p,i)}$ via (48)
end for
Produce the predictive likelihood $p\left(\mathbf{y}_{n+1}|\mathbf{y}_{0:n}, \mathcal{D}_i\right)$ from (42);
Calculate the *normalized* weight $\omega_{n+1}^{(p,i)}$ according to (45);
Calculate likelihood $p\left(\mathbf{y}_{0:n+1}|\mathcal{D}_i\right)$ in (20);
end for
Determine endpoint probability: $p(\mathcal{D}_i|\mathbf{y}_{0:n})$ in (1);
end for

4. Results

0Figure 1. This system used the off-the-shelf sensor, Leap Motion, which can reliably track hand and finger positions in 3D during the pointing-selection tasks, at a rate exceeding 30 Hz. The utilized dataset contains 95 trajectories pertaining to four participants while undertaking pointing-selection tasks under various road and driving conditions. Here, we divide these data into two sets:

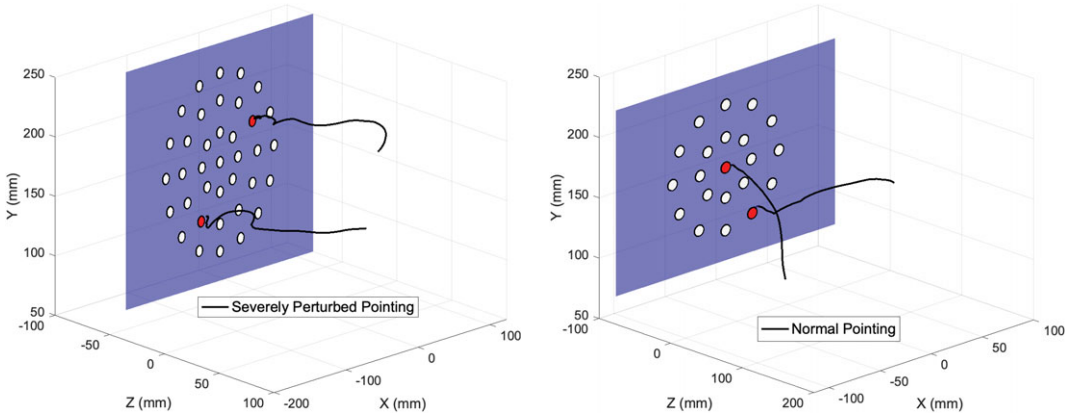
1. Dataset A with all 95 pointing tracks; this allows us to perform a comprehensive comparison between different algorithms for various levels of present perturbations (e.g., static, motorway driving and off-road driving).
2. Dataset B with 10 trajectories when the user input was subjected to severe level of noise due to driving on a badly maintained road or off-road driving. This dataset is a subsect of Dataset A and was collected in a Land Rover. It is particularly relevant to examine the outcome of the algorithms that incorporate a jump process, that is employ jump diffusion models.

During the above interaction tasks, an experimental user interface with multiple selectable circular icons was displayed on a touchscreen mounted to the car dashboard. The number of selectable icons is $|\mathbb{D}| = 21$ for Dataset A, and $|\mathbb{D}| = 37$ for Dataset B. Two typical pointing trajectories of each dataset are presented in Figure 4. Similar to the common ISO 9241 pointing task, often referred to as Fitt's law task, one randomly chosen GUI item is highlighted at a time and the user is expected to select it. Identical uniform prior is placed on all of the interface items, that is, $p(\mathcal{D} = \mathcal{D}_i) = 1/N_{\mathcal{D}}$ for all $\mathcal{D}_i \in \mathbb{D}$ in order for the results to be comparable to those in previous work.

Below, we use the aggregate inference success and the timely successful prediction over pointing duration to evaluate the predictors performance; both apply a maximum *a posteriori* criterion (i.e., pick the most probable icon) as per:

$$\widehat{\mathcal{D}}(t_n) = \arg\max_{\mathcal{D}_i \in \mathbb{D}} p(\mathcal{D} = \mathcal{D}_i|\mathbf{y}_{0:n}).$$

More specifically, the first is defined as the proportion of the total pointing gesture (in time), from its start at t_0 until touching the display surface at time $\mathcal{T}$, for which the predictor correctly inferred the true endpoint $\mathcal{D}_{\text{True}} \in \mathbb{D}$. The second captures the percentage of the correct prediction over all tested dataset as a function of the percentage of pointing task duration, thus indicating how early the predictor assigns the highest probability for the correct destination.



(a) Dataset A (21 icons) (b) Dataset B (37 icons)

Figure 4. *Example trajectories of collected real pointing trajectories.*

4.1. Prediction performance with linear Gaussian intent-driven models

For the 95 pointing tracks covering different levels of perturbations (i.e., Dataset A), the computationally efficient LTI Gaussian models are sufficient to predict the intended icon with a high accuracy. In this section, we evaluate all LTI Gaussian models introduced in Section 2.1.1 for this dataset. The parameters for all tested predictors are listed in Table 1. They are chosen in a manual way and from examining a few possible values (i.e., no training or fine tuning across all test trajectories was undertaken).

It is noted that this model parameterization is aimed at demonstrating the low training requirement of the adopted state-space-modeling-based inference approach, since the models are physically meaningful. Take the linear Gaussian mean reverting model as an example. Although a higher noise parameter σ would lead to higher uncertainty on the final endpoint, it permit more flexibility in the target dynamics manifested in elaborate maneuvers (e.g., swings) of the target (i.e., pointing finger) en-route to its destination, instead of simply following a straight line. A higher reversion parameter η would cause the stronger force towards the endpoint, such that a higher damping factor ρ (and/or γ) ensures that the finger speed upon touch is reasonable. A set of fined-tuned parameters can trade-off generalizability of the model to new data for a high (validation) prediction accuracy. Alternatively, the parameters of OU models may be set based on maximization of the likelihood $\prod_{k=1}^{K} p\left(\mathbf{y}_{0:n}^{[k]} | \mathcal{D} = \mathcal{D}_i, \Omega\right)$ for a sample of K typical full pointing finger trajectories; Ω is the set of the parameters for an intent-driven dynamic model. As the driver/passenger uses the touch system, the system can refine the applied model parameters from the larger available dataset(s). On the other hand, the automatic parameter tuning for BD models is more complicated due to the condition on unknown arrival time. Nonetheless, from our extensive experiments and Table 1 we can confirm that these empirically selected parameters of the BD methods work sufficiently well.

The timely successful prediction over pointing duration is shown in Figure 5. As expected, all methods generally exhibit an upward trend, that is their performance improves as the the pointing finger-hand approaches the intended endpoint. Specifically, the ERA model can perform poorly at the beginning period of the pointing motion (e.g., in the first 30%); however, it delivers comparable results thereafter. Combined with the overall success rate shown in Table 1, it can be seen that all examined models achieve comparable prediction successes. Hence, the predictive touch system could infer the intended on-display item remarkably early in the pointing-selection tasks. Nonetheless, it can be noticed from Table 1 and Figure 5 that the BD models achieve better results compared with the Gaussian mean reverting models. Furthermore, performance of models whose acceleration is driven by a Wiener process (BD–CA) are also superior to those constructed merely on target position and velocity (BD–CV). This may be due to the fact

Table 1. Linear time invariant (LTI) Gaussian models parameters and overall prediction performance for 95 tracks.

Models	Parameter values	Success rates
ERV (Ahmad et al., 2016b)	$\eta = 55, \rho = 15, \sigma = 3000$	62.9%
ERA (Gan et al., 2019)	$\eta = 1150, \rho = 320, \gamma = 29, \sigma = 1.7 \times 10^4$	63.4%
BD–CV (Ahmad et al., 2018)[a]	$\sigma_{\text{CV}} = 650, \sigma_{\text{pos}}^{\mathcal{D}_i} = 1.5, \sigma_{\text{vel}}^{\mathcal{D}_i} = 100$	64.4%
BD–CV (Liang et al., 2019)[a]	$\sigma_{\text{CV}} = 650, \sigma_{\text{pos}}^{\mathcal{D}_i} = 1.5, \sigma_{\text{vel}}^{\mathcal{D}_i} = 100$	65.4%
BD–CA (Liang et al., 2019)[a]	$\sigma_{\text{CA}} = 9400, \sigma_{\text{pos}}^{\mathcal{D}_i} = 1.5, \sigma_{\text{vel}}^{\mathcal{D}_i} = 50, \sigma_{\text{acc}}^{\mathcal{D}_i} = 1500$	68.3%
BD–CV (this paper)[a]	$\sigma_{\text{CV}} = 650, \sigma_{\text{pos}}^{\mathcal{D}_i} = 1.5, \sigma_{\text{vel}}^{\mathcal{D}_i} = 100$	65.2%
BD–CA (this paper)[a]	$\sigma_{\text{CA}} = 9500, \sigma_{\text{pos}}^{\mathcal{D}_i} = 1.5, \sigma_{\text{vel}}^{\mathcal{D}_i} = 25, \sigma_{\text{acc}}^{\mathcal{D}_i} = 1500$	68.3%

Abbreviations: BD–CA, bridging distributions-constant acceleration; BD–CV, bridging distributions-constant velocity; ERA, equilibrium reverting acceleration; ERV, equilibrium reverting velocity.
[a]For all BD models, $p(\mathcal{T}|\mathcal{D}_i) = \text{Unif}(0.1\,\text{s}, 1.9\,\text{s})$, the number of quadrature points $N_{\mathcal{T}} = 30$, $\widetilde{G} = \mathbf{I}$ and $\sigma^{\mathcal{D}_i}$ form the corresponding Σ_i.

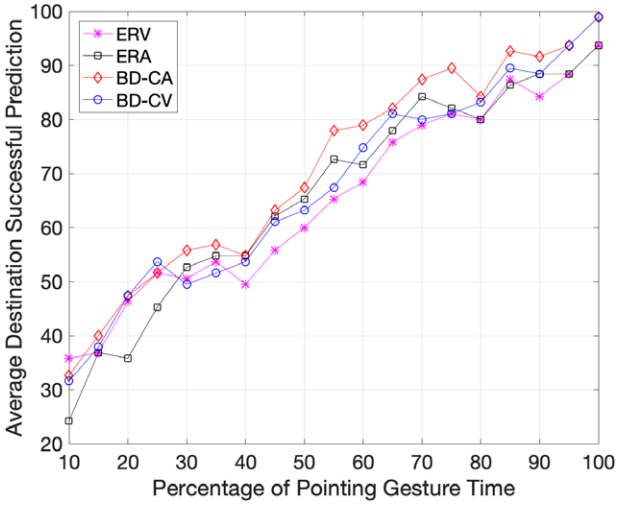


Figure 5. *Average successful prediction over time (Dataset A).*

that present accelerations can reflect the movement trend with more details. Additionally, the advantage of BD methods may be gained from more accurate end state construction such that a successful prediction can always be achieved at the end of pointing period. It is worth mentioning that, in our case, the intent predictors implemented according to Algorithm 1 of Liang et al. (2019) have the lowest complexity compared with other BD counterparts while the OU-based predictors have the least computational cost among all evaluated methods. Note that for better visualization we have chosen to only display the success rate against gesture time for the BD method proposed in this paper because the lines from previous BD formulations are visually very similar to that introduced here.

4.2. Highly perturbed scenarios and particle filtering

The intent inference performance for highly perturbed trajectories in Dataset B has been tested with jump models and Gaussian mean reverting models in Gan et al. (2019); Ahmad et al. (2016a). Results from the BD models introduced in this paper are also included for comparison. The aggregate inference success for

all algorithms and the timely successful prediction from four selected algorithms (i.e., omitting non-BD models for the clarity of presentation) are depicted in Figures 6 and 7, respectively. The applied jump models below are described in Algorithm II and each use 2000 particles, but it has been observed that a comparable performance can be achieved with a small number (e.g., 500) of particles; their parameters are listed in Table 2 (the jumps are assumed to be isotropic) and those for all of the LTI Gaussian models remain the same as in Table 1.

From Figure 7, one can see that the BD–CA model always achieves the highest successful prediction after the first 20 percents duration, and the BD models can always achieve the accurate prediction at the end stage of pointing due to its Markov bridge nature. Similar to the LTI ERA model, the jump-ERA model ascends from a relatively low successful prediction, to a comparable successful rate on the second half of the pointing duration. This insensitivity may be caused by a longer reflection on the observation from the acceleration constructed intention. The average success rate in Figure 6 indicates that the BD–CA outperforms other models for this dataset, while the jump-ERV model achieves the second best success rate. This may lead to the conclusion that the BD–CA is the best among other models on characterizing the intention of the hand pointing. However, it is worthwhile to note that the exploration for the parameters of jump models are more restrictive due to their larger number of parameters and

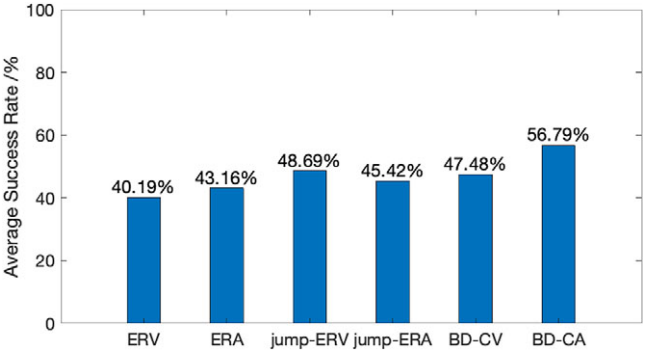


Figure 6. *Average success rate for Dataset B.*

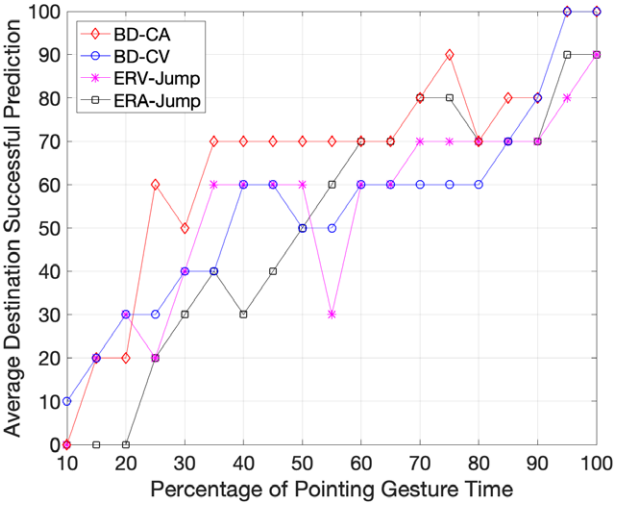


Figure 7. *Average successful prediction over time (Dataset B).*

Table 2. Jump models parameter sets.

Models	Mean-reverting dynamics	Jumps
Jump-ERV	$\eta = 60, \rho = 15, \sigma = 450$	$\mu_J = 0, \sigma_J = 866, \lambda_J^{-1} = 1$
Jump-ERA	$\eta = 1150, \rho = 320, \gamma = 30, \sigma = 8000$	$\mu_J = 0, \sigma_J = 1.6 \times 10^4, \lambda_J^{-1} = 0.2$

Abbreviations: ERA, equilibrium reverting acceleration; ERV, equilibrium reverting velocity.

time-consuming evaluation process. Thus it is possible that a better results can be achieved with other parameters for jump models, especially for the jump-ERA model. Additionally, the present jumps/jolts in those 10 tracks might not be of the severity (magnitude and/or transience) that a BD–CA model cannot successfully smooth out or follow. Under such high-levels of perturbations, the numerical marginalization of arrival time with BD can be challenging as the pointing-task duration can be subject to large delays, with the risk of it being very distinctive from the prior of $\mathcal{T}$. Nevertheless, the use of the particle filtering with a jump process offers additional advantages, not necessarily relevant to the predictive touch usecase, such as detecting the location-time of the perturbations-induced fast maneuvers (jumps) and potentially better destination-aware tracking results, see Gan et al. (2019).

5. Conclusion

In this paper, we presented an overview of the existing stochastic dynamic modeling methods for destination inference, with the in-vehicle predictive touch system as the case study. It covers linear Gaussian and nonlinear setups, both proposed within a Bayesian framework. The adopted continuous time intent-driven state space models naturally facilitate treating asynchronous data, including from multiple sensors. In addition, a new bridging distribution approach was proposed here, which has a moderate computational requirement and a clear stochastic interpretation compared with previous formulations. Results from real data of a predictive touch system demonstrated the efficacy of the various considered prediction algorithms, namely their ability to infer the user intent remarkably early in the pointing-selection task. Thereby, this can facilitate effective touchless interactions via the intuitive free hand pointing gestures. It is emphasized that the presented prediction techniques are also applicable to other fields, for example surveillance, smart navigation, robotics, etc. Nevertheless, there are several extensions to this work, for example bridging distributions for nonlinear and/or non-Gaussian systems (e.g., a stable Lévy system in Gan and Godsill, 2020), considering intrinsically nonlinear intent-driven motion models for highly maneuverable objects and various measurement models (one such example can be found in Liang et al., 2020). This paper serves as an impetus to further research on meta-level tracking models and inference algorithms.

Funding Statement. This research was supported by grants from Jaguar Land Rover under the Centre for Advanced Photonics and Electronics CAPE agreement.

Competing Interests. The authors declare no competing interests exist.

Data Availability Statement. The data used in this work is proprietary and confidential; it cannot be made publicly available. Readers are nonetheless encouraged to contact authors where data and code could be shared subject to the recipient abiding by certain terms and conditions.

Ethical Standards. The conducted user studies for predictive touch met all ethical guidelines of the University of Cambridge and Jaguar Land Rover, including adherence to the legal requirements of the study country.

Authorship Contributions. Conceptualization: all; Data curation: B. A.; Formal analysis: R. G., J. L., and B. A.; Funding acquisition: S. G., and B. A.; Investigation: R. G. and J. L.; Methodology: all; Software: R. G., and J. L.; Supervision: B. A., and S. G.; Validation: R. G., and J. L.; Writing-original draft: R.G, J. L., and B. A.; Writing-review editing: all; All authors approved the final submitted draft.

Notation

$\mathbb{D}$	discrete set of possible destinations, $\mathbb{D} = \{\mathcal{D}_i : i = 1,2,\ldots,N_{\mathcal{D}}\}$
$N_{\mathcal{D}}$	number of nominal endpoints
$\mathcal{D}_i$	the ith endpoint
$\mathcal{D}$	considered intended destination
$\widehat{\mathcal{D}}$	maximum *a posteriori* estimate for the intended destination
$\mathcal{T}$	destination arrival time, $\mathcal{T} = t_N$
$\widetilde{\mathbf{y}}_N^i$	pseudo-observation vector for destination $\mathcal{D}_i$
Σ_i	covariance of the Gaussian pseudo-observation model for destination $\mathcal{D}_i$
$\mathbf{x}_n$	target dynamic state at time t_n
$\mathbf{x}_{n,i}$	dynamic state at time t_n for an object travelling to $\mathcal{D}_i$
$\mathbf{y}_n$	observation vector captured at time t_n
$\boldsymbol{\beta}_t$	multivariate standard Wiener process
$\mathbf{J}_t$	compound Poisson process with Gaussian distributed jump size $\mathbf{S}_k$, that is $\mathbf{J}_t = \sum_{\tau_k < t} \mathbf{S}_k$
τ_k	arrival time of the k th jump
$\mathcal{N}(\mathbf{x}\|\mathbf{m},\mathbf{C})$	multivariate normal distribution for random variable $\mathbf{x}$ with mean $\mathbf{m}$ and covariance $\mathbf{C}$
$N_{\mathcal{P}}$	number of particles used in the particle filtering
$\omega_n^{(p,i)}$	*normalized* weight at time t_n for the p th particle for destination $\mathcal{D}_i$
$\widetilde{\omega}_n^{(p,i)}$	*updated* weight for the p th particle for endpoint $\mathcal{D}_i$
$\mathbf{I}$	identity matrix with the suitable size
$.'$	transpose operation
$p(\cdot)$	probability density function

References

Ahmad BI, **Hare C**, **Singh H**, **Shabani A**, **Lindsay B**, **Skrypchuk L**, **Langdon P and Godsill S** (2019a) Touchless selection schemes for intelligent automotive user interfaces with predictive mid-air touch. *International Journal of Mobile Human Computer Interaction (IJMHCI) 11*(3), 18–39.

Ahmad BI, **Langdon PM and Godsill SJ** (2019b) A Bayesian framework for intent prediction in object tracking. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 8439–8443.

Ahmad BI, **LangdonPM**, **Godsill SJ**, **Donkor R**, **Wilde R and Skrypchuk L** (2016a) You do not have to touch to select: a study on predictive in-car touchscreen with mid-air selection. In *Proceedings of the 8th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. ACM, pp. 113–120.

Ahmad BI, **Langdon PM**, **Godsill SJ**, **Hardy R**, **Skrypchuk L and Donkor R** (2015) Touchscreen usability and input performance in vehicles under different road conditions: an evaluative study. In *Proceedings of the 7th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. ACM, pp. 47–54.

Ahmad BI, **Murphy J**, **Langdon PM and Godsill SJ** (2014) 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)*. IEEE, pp. 1–6.

Ahmad BI, **Murphy JK**, **Godsill S**, **Langdon P and Hardy R** (2017) Intelligent interactive displays in vehicles with intent prediction: a Bayesian framework. *IEEE Signal Processing Magazine 34*(2):82–94.

Ahmad BI, **Murphy JK**, **Langdon P**, **Godsill S**, **Hardy R and Skrypchuk L** (2016b) Intent inference for pointing gesture based interactions in vehicles. *IEEE Transactions on Cybernetics 46*, 878–889.

Ahmad BI, **Murphy JK**, **Langdon PM and Godsill SJ** (2018) Bayesian intent prediction in object tracking using bridging distributions. *IEEE Transactions on Cybernetics*, *48*(1), 215–227.

Baccarelli E and Cusani R (1998) Recursive filtering and smoothing for reciprocal Gaussian processes with Dirichlet boundary conditions. *IEEE Transactions on Signal Processing 46*(3), 790–795.

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

Bark K, **Tran C**, **Fujimura K and Ng-Thow-Hing V** (2014) Personal navi: Benefits of an augmented reality navigational aid using a see-thru 3d volumetric hud. In *Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. ACM, pp. 1–8.

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

Broy N, **Guo M**, **Schneegass S**, **Pfleging B and Alt F** (2015) Introducing novel technologies in the car: conducting a real-world study to test 3d dashboards. In *Proceedings of the 7th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. ACM, pp. 179–186.

Castanon DA, **Levy BC and Willsky AS** (1985) Algorithms for the incorporation of predictive information in surveillance theory. *International Journal of Systems Science 16*(3), 367–382.

Christensen HL, **Murphy J and Godsill SJ** (2012) Forecasting high-frequency futures returns using online Langevin dynamics. *IEEE Journal of Selected Topics in Signal Processing 6*(4), 366–380.

Fanaswala M and Krishnamurthy V (2015) Spatiotemporal trajectory models for metalevel target tracking. *IEEE Aerospace and Electronic Systems Magazine 30*(1), 16–31.

Gan R and Godsill S (2020) α-stable Lévy state-space models for manoeuvring object tracking. In *23rd International Conference on Information Fusion (FUSION)*. IEEE, pp. 1–7.

Gan R, **Liang J**, **Ahmad B and Godsill S** (2019) Bayesian intent prediction for fast maneuvering objects using variable rate particle filters. In *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, pp. 1–6.

Gasbarra D, **Sottinen T and Valkeila E** (2007) Gaussian bridges. In *Stochastic Analysis and Applications*. Springer, pp. 361–382.

Gaurav S and Ziebart B. (2019) Discriminatively learning inverse optimal control models for predicting human intentions. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, pp. 1368–1376.

Godsill S (2007) Particle filters for continuous-time jump models in tracking applications. *ESAIM: Proceedings 19*, 39–52.

Goode N, **Lenné MG and Salmon P** (2012) The impact of on-road motion on bms touch screen device operation. *Ergonomics 55* (9), 986–996.

Harris CM and Wolpert DM (1998) Signal-dependent noise determines motor planning. *Nature 394*(6695), 780.

Harvey AC (1990) *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press.

Haug AJ (2012) *Bayesian Estimation and Tracking: A Practical Guide*. John Wiley & Sons.

Ba h KM,Jæger M. G, **Skov MB and Thomassen NG.** (2008) You can touch, but you can't look: interacting with in-vehicle systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, pp. 1139–1148.

Kou SG (2002) A jump-diffusion model for option pricing. *Management Science 48*(8), 1086–1101.

Li XR and Jilkov VP (2003) Survey of maneuvering target tracking. Part I. Dynamic models. *IEEE Transactions on Aerospace and Electronic Systems 39*(4), 1333–1364.

Liang J, **Ahmad BI**, **Gan R**, **Langdon P**, **Hardy R and Godsill S** (2019) On destination prediction based on markov bridging distributions. *IEEE Signal Processing Letters 26*(11), 1663–1667.

Liang J, **Ahmad BI and Godsill S** (2020) Simultaneous intent prediction and state estimation using an intent-driven intrinsic coordinate model. In *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*.

May KR, **Gable TM and Walker BN** (2017) Designing an in-vehicle air gesture set using elicitation methods. In *Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. ACM, pp. 74–83.

Millefiori LM, **Braca P**, **Bryan K and Willett P** (2016) Modeling vessel kinematics using a stochastic mean-reverting process for long-term prediction. *IEEE Transactions on Aerospace and Electronic Systems 52*(5), 2313–2330.

Plaumann K, **Weing M**, **Winkler C**, **Müller M and Rukzio E** (2018) Towards accurate cursorless pointing: the effects of ocular dominance and handedness. *Personal and Ubiquitous Computing 22*(4), 633–646.

Quinn P, **Lee SC**, **Barnhart M and Zhai S** (2019) Active edge: Designing squeeze gestures for the google pixel 2. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, p. 274.

Rezaie R and Li XR (2019a) Gaussian Conditionally Markov Sequences: Dynamic Models and Representations of Reciprocal and Other Classes. *IEEE Transactions on Signal Processing 68*, 155–169.

Rezaie R and Li XR (2019b) Gaussian conditionally Markov sequences: singular/nonsingular. *IEEE Transactions on Automatic Control 65*(5), 2286–2293.

Roider F and Gross T (2018) I see your point: integrating gaze to enhance pointing gesture accuracy while driving. In *Proceedings of the 10th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. ACM, pp. 351–358.

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

Zhou G, **Li K and Kirubarajan T** (2020) Constrained state estimation using noisy destination information. *Signal Processing 166*, 107226.

Cite this article: Gan, R. Liang, J. Ahmad, B. I. and Godsill, S. 2020. Modeling intent and destination prediction within a Bayesian framework: Predictive touch as a usecase. *Data-Centric Engineering*, 1: e12. doi:https://doi.org/10.1017/dce.2020.11