Intent Inference for Hand Pointing Gesture Based Interactions in Vehicles

Bashar I. Ahmad, James K. Murphy, Patrick M. Langdon, Simon J. Godsill, Robert Hardy and Lee Skrypchuk

***Abstract*—Using interactive displays, such as a touchscreen, in vehicles typically requires dedicating a considerable amount of visual as well as cognitive capacity and undertaking a hand pointing gesture to select the intended item on the interface. This can act as a distractor from the primary task of driving and consequently can have serious safety implications. Due to road and driving conditions, the user input can also be highly perturbed resulting in erroneous selections compromising the system usability. In this paper, we propose intent-aware displays that utilise a pointing gesture tracker in conjunction with suitable Bayesian destination inference algorithms to determine the item the user intends to select, which can be achieved with high confidence remarkably early in the pointing gesture. This can drastically reduce the time and effort required to successfully complete an in-vehicle selection task. In the proposed probabilistic inference framework, the likelihood of all the nominal destinations are sequentially calculated by modelling the hand pointing gesture movements as a destination-reverting process. This leads to a Kalman filter-type implementation of the prediction routine that requires minimal parameter training and has low computational burden; it is also amenable to parallelisation. The substantial gains obtained using an intent-aware display are demonstrated using data collected in an instrumented vehicle driven under various road conditions.**

***Index Terms*—Interactive Displays, Finger Tracking, Human Computer Interactions, Bayesian Inference, Kalman filtering.**

I. Introduction

INTERACTIVE displays are becoming an integrated part of the modern vehicle environment and are increasingly replacing traditional static controls such as buttons, knobs, switches and gauges [1]–[6]. This has been facilitated by the proliferation of the touchscreen technology and the ability of such displays to effectively handle a multitude of functions. They can accommodate the large quantities of information associated with control of and feedback from In-Vehicle Infotainment Systems (IVIS). The functionality and complexity of IVIS have steadily increased to incorporate, amongst other services, route guidance, communications, climate control and music players. Touchscreens also facilitate intuitive interactions via pointing gestures, particularly for novice users, and offer design flexibility through a combined display-input–feedback module [1]–[4]. The display can easily be adapted to the context of use and thereby can minimise clutter in the vehicle interior introduced by mechanical controls; see, for example, the Volvo concept car [5].

However, using a touchscreen entails undertaking a hand pointing gesture that inevitably requires substantial visual, cognitive and manual capacity [1], [7]–[9]. The user input can also be highly perturbed, with the pointing finger exhibiting erratic movements due to driving and/or road conditions [10]–[13]. This leads to erroneous selections. Rectifying these errors or adapting to the noisy environment ties up further attention that would otherwise be available for driving [10]–[12]. Such distractions can have serious safety implications by hampering the driver's situational awareness, steering capability, and lane maintenance [14]–[16]. Figure 1 shows the frequency of successfully selecting the intended item on a Graphical User Interface (GUI) displayed on an in-vehicle touchscreen. The results shown were obtained from four passengers, undertaking a large number of pointing tasks on a variety of road conditions. It is clear from the figure that the difficulty of the selection task increases as the road conditions deteriorate. The erroneous selection rate exceeds 75% when driving over harsh terrain. The selection success rate is expected to be even lower for the driver as their attention is divided between pointing and driving [17], [18].

In this paper, we propose intent-aware interactive displays that can determine, early in the pointing gesture, the item a user intends to select, e.g. a GUI icon displayed on a touchscreen. This enables significant reduction in the pointing time and therefore effort (visual, cognitive and manual) required to accomplish the selection task, helping maintain the driver's focus on the road. The system introduced here, depicted in Figure 2, employs a gesture tracking sensor to capture, in real-time, the pointing hand/finger location(s) that are used

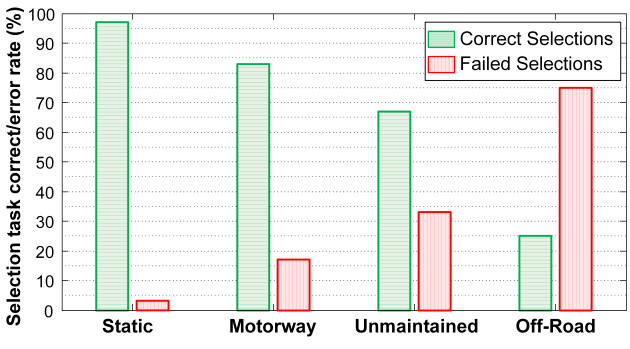


Figure 1: Target selection on an in-vehicle touchscreen.

Manuscript received XXXX XX, XXXX; revised XXXX XX, XXXX; ; accepted XXXX XX, XXXX.

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.

R. Hardy and L. Skrypchuck are with Jaguar Land Rover, Whitley, Coventry, UK. Email:{rhardy, lskrypch}@jaguarlandrover.com.

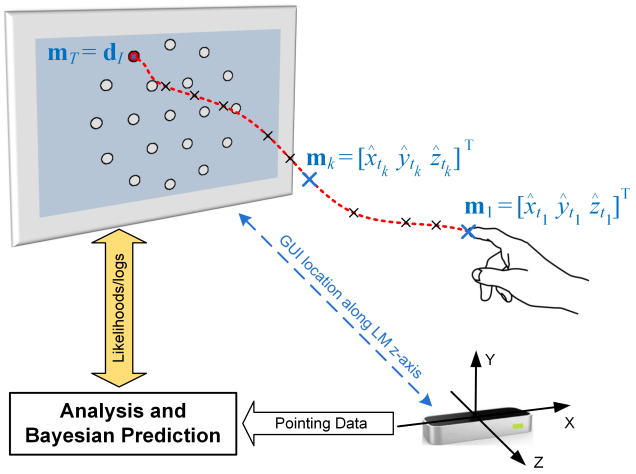


Figure 2: System block diagram with a complete trajectory of the pointing gesture to select the I^{th} highlighted GUI icon.

by the destination inference module. Several such sensors, including the Microsoft Kinect [19], Leap Motion controller [20] and Nimble UX, have emerged recently, facilitating accurate gestures tracking and recognition. The development of these devices has been motivated by a desire to extend Human Computer Interaction (HCI) beyond traditional keyboard input and mouse pointing [21]. In this study, a Leap Motion (LM) sensor is used to capture pointing finger trajectories.

The intent prediction algorithms developed here are based on linear motion models that incorporate the nominal destinations of the pointing gesture. These allow the likelihood of each of the possible endpoints to be obtained dynamically, as the user points towards the display. A destination is assigned high probability if the available partial pointing finger trajectory is consistent with the motion model that postulates this endpoint as the intended destination. Pointing data collected in a vehicle is used to illustrate the system performance.

The remainder of this paper is organised as follows. In Section II, existing work is addressed, highlighting certain features of the pointing gesture. The problem is formulated in Section III and the destination-reverting prediction models are detailed in Section IV where a pseudo-code of the implementation of proposed algorithms is provided. In Section V, the performance of the proposed inference framework is evaluated and results of the destination-reverting prediction models are compared against other benchmark techniques using real pointing data. Finally, conclusions are drawn in Section VI.

II. Related Work

The benefits of predicting the destination of a pointing task are widely recognised in the HCI field, e.g. [22]–[29]. Such studies focus on pointing via a mouse or mechanical device in a 2D set-up to acquire on-screen targets, a common mode of human computer interaction over the past few decades. In this paper, however, we consider 3D free hand pointing gestures to interact with touchscreens, which is an increasingly popular mode of interaction, including recently in automotive contexts. Fitt's law [30] states that the time t_T required to select an item of width W at distant ℓ from the starting position is given by

$$t_T = a + b\log_2(1 + \ell/W) \tag{1}$$

with $\mathrm{ID} = \log_2(1 + \ell/W)$ being an index of difficulty; a and b are empirically estimated [22], [31]. As would be intuitively expected, the selection process can be simplified and expedited by using a pointing facilitation method such as increasing the target size or moving targets closer to the cursor location. The high complexity of typical GUIs, containing many simultaneously selectable icons of different sizes and shapes, renders such assistance ineffective without knowledge of the intended destination [22]. Therefore, destination prediction should precede any pointing facilitation action.

Existing prediction algorithms include the *nearest neighbour*, which chooses the target that is closest to the cursor's current position [24], and *bearing angle*, which assumes that the cursor moves in a nearly constant direction towards the intended endpoint [23], [25]. These methods treat destination inference as a classification problem and rely on known priors such as selection pattern(s). Here, probabilistic prediction is pursued and the likelihood of each nominal destination can be determined independently from the underlying priors. If priors become available, they can be easily included.

Regression-based predictors for 2D environments in [22], [26], [27] leverage learnt mouse cursor movement kinematics. If $\mathbf{m}_k = [\hat{x}_{t_k}\ \ \hat{y}_{t_k}]'$ is the observed cursor position at the time instant t_k ($\mathbf{x}'$ denotes the transpose of $\mathbf{x}$), the cursor location at the end of the pointing task at time t_T is estimated as

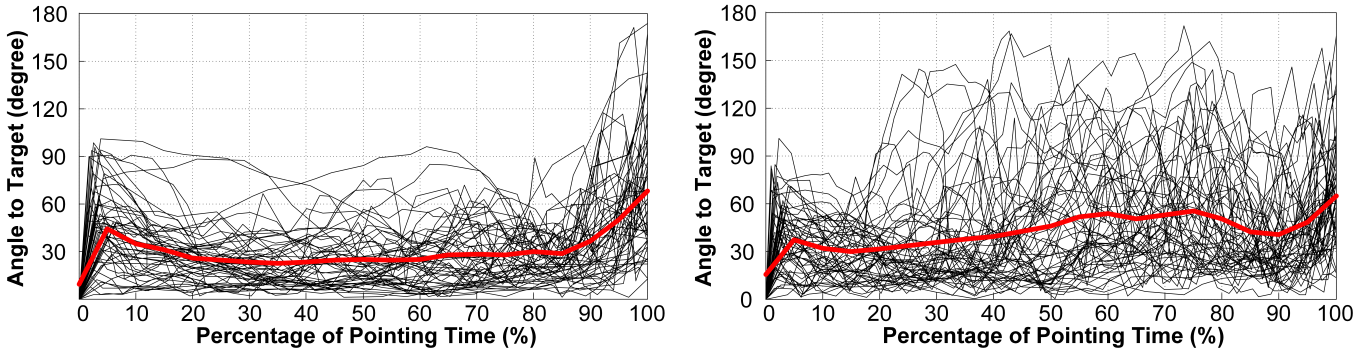
$$\hat{\mathbf{m}}_T = \mathbf{m}_1 + \ell_T \frac{\mathbf{m}_k - \mathbf{m}_1}{\|\mathbf{m}_k - \mathbf{m}_1\|_2} \tag{2}$$

where ℓ_T is the total distance travelled by the cursor since the task start time t_1, i.e. from $\mathbf{m}_1$, and $\|\mathbf{x}\|_2$ is the L_2 norm of vector $\mathbf{x}$. In [26], this distance is calculated using $\ell_T = a\nu_{max} + b$, where $\nu_{max} = \max_t \|\mathbf{m}_t - \mathbf{m}_{t-1}\|_2$ is the peak observed cursor velocity up to the current time. Regression parameters a and b are learnt *a priori*. To allow for the typically slow start of a pointing movement, ℓ_T is not predicted until ν_{max} is above some predefined threshold. In [27], the velocity ν_k at t_k is assumed to be related to the distance travelled via $\nu_k = a\ell_k^2 + b\ell_k + c$. After estimating the coefficients a, b and c from the cursor trajectory up to time t_k, the total distance is calculated using $\ell_T = \alpha\hat{\ell}_T$ such that $\hat{\ell}_T$ is determined based on the premise that $\nu_T = 0$ at destination (i.e. solving $a\hat{\ell}_T^2 + b\hat{\ell}_T + c = 0$) and α is a correction parameter learned from training data.

A machine learning predictor based on inverse-optimal control was introduced in [28]. It models the pointing movements as

$$\dot{\mathbf{m}}_t = \mathbf{F}\dot{\mathbf{m}}_{t-1} + \mathbf{C}\mathbf{f}_t + \boldsymbol{\varepsilon}_t \tag{3}$$

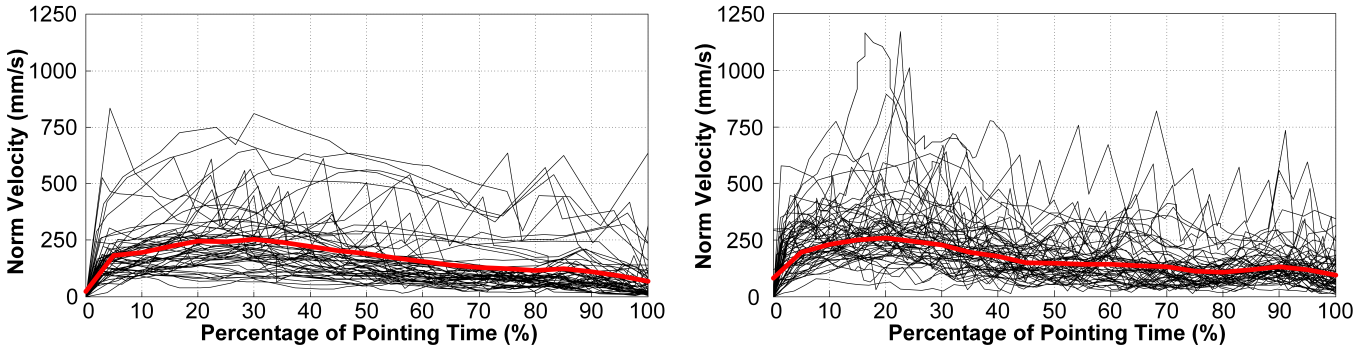
where $\dot{\mathbf{m}}_t$ is a latent state that includes pointing finger position $\mathbf{m}_t$, velocity, acceleration and jerk; $\mathbf{f}_t$ is a control parameters vector and $\boldsymbol{\varepsilon}_t$ is noise. A maximum entropy approach is used to obtain the probabilities of all possible targets. This optimal-control predictor has a high computational cost, requiring substantial parameter training, making real-time implementation difficult. The approach proposed in this paper requires



(a) Stationary vehicle.

(b) Mobile vehicle; varying speeds and road conditions.

Figure 3: Angle to target θ_k for 60 in-vehicle pointing tasks; thick red line is the mean value from all tasks.



(a) Stationary vehicle.

(b) Mobile vehicle; varying speeds and road conditions.

Figure 4: Pointing finger velocity, $\|\mathbf{m}_k - \mathbf{m}_{k-1}\|_2$, for 60 in-vehicle pointing tasks; thick red line is the mean value.

minimal training, has low computational cost, is amenable to parallelisation and delivers competitive performance.

In (2) and (3), the cursor is assumed to head at a nearly constant angle towards its destination. Possible destinations are collinear along this bearing; inferring the track length therefore predicts the intended destination. Whilst this premise makes intuitive sense for 2D GUIs, it does not hold for 3D pointing trajectories. Figures 3 and 4 depict the pointing finger heading angle to target θ_k and velocity for 60, 3D pointing tracks collected in both stationary and mobile vehicles. If $\mathbf{d}_I = [d_{x,I} \quad d_{y,I} \quad d_{z,I}]'$ is the 3D coordinates of the intended GUI icon, the angle between the pointing finger heading and this destination is $\theta_k \triangleq \angle(\mathbf{m}_k - \mathbf{m}_{k-1}, \mathbf{d}_I)$; $\mathbf{m}_{k-1}$ and $\mathbf{m}_k$ are two successive finger-tip positions. Each of $\mathbf{m}_n$ and $\mathbf{d}_i$ are relative to the gesture-tracking sensor centre and orientation. It is clear from Figure 3 that θ_k varies drastically over time, especially when the vehicle is in motion, due to in-vehicle perturbations. Furthermore, Figure 4 shows that the pointing velocity is not zero upon selecting the icon on the touchscreen unlike pointing in 2D via a mouse cursor. Neither does the velocity exhibit a consistent pronounced peak during the initial *ballistic* pointing phase, unlike 2D trajectories [26], [27], [32]. Thus, assuming a constant heading angle to the intended target and a predefined velocity profile as in (2) and (3) lead to poor quality predictions for 3D free hand pointing gestures.

Recently, there has been interest in countermeasures against perturbed user input on touchscreens due to situational impairments (such as walking vibrations) and/or divided attention for mobile computing platforms [33]–[36]. Solutions typically rely on built-in inertial measurement units or camera(s) to dynamically adapt the GUI layout and/or compensate for the present noise. In automotive settings, the pointing time and distance are noticeably longer than those for hand-held devices. Most importantly, it is demonstrated in [37] that the correlation between the accelerations-vibrations of the pointing hand and those experienced in (or by) the vehicle are often weak and ambiguous, due to the human response to noise, seat position, cushioning, etc. Thus, compensating for measured in-vehicle perturbations has limited effect.

In application areas, such as surveillance and defence, determining the destination of a tracked object can be valuable since it dictates the trajectory followed by the object and offers information on possible threats [38]–[40]. Conventional methods use tracking algorithms to infer the object state (including position and velocity) followed by an additional mechanism to infer its destination. In [39] and [40], the monitored spatial area is discretised into a grid. Tracked objects can then pass through a finite number of predefined zones. For 3D free hand pointing gestures, however, there are infinite possible paths and discretisation is a burdensome task. Instead, here we introduce a simple approach that does not impose tracks the user's hand would be expected to follow. In this framework, tracking and

intent-inference are a single operation.
Finally, relative ray-cast pointing, i.e. pointing at a display from a distance, is becoming popular due to the availability of devices such as the Nintendo Wii Remote, PlayStation Move controller, etc. A recent overview of ray-cast pointing facilitation schemes is given in [41]. These are similar to those used for mouse pointing and the problem is often transformed into 2D with a minimum device-display distance imposed.

III. Problem formulation

A. Probabilistic Prediction Approach

Let $\mathbb{D} = \{\mathcal{D}_i : i = 1, 2, ..N\}$ be the set of N nominal destinations, for example GUI icons displayed on an in-vehicle touchscreen. The known 3D coordinates of the i^{th} destination are denoted by $\mathbf{d}_i$, but no further assumptions are made about the GUI layout. The objective is to determine the probability of each possible destination being the intended endpoint of the pointing gesture given the k available observations up to time t_k, i.e. to calculate the probability $P(\mathcal{D}_I = \mathcal{D}_i \mid \mathbf{m}_{1:k})$ for $i = 1, 2, ..., N$, where $\mathcal{D}_I \in \mathbb{D}$ is a random variable representing the (unknown) intended destination. Measurements $\mathbf{m}_{1:k} \triangleq \{\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_k\}$ are captured by the gesture-tracker at times $t_1, t_2, ..., t_k$, with $\mathbf{m}_n = [\hat{x}_{t_n} \;\; \hat{y}_{t_n} \;\; \hat{z}_{t_n}]'$ being the Cartesian coordinates of the pointing finger at t_n.
After inferring the probabilities $P(\mathcal{D}_I = \mathcal{D}_i \mid \mathbf{m}_{1:k})$ at time t_k, a point estimate of the intended destination $\hat{I}(t_k) \in \mathbb{D}$ can be made by minimising a cost function via

$$\hat{I}(t_k) = \underset{\mathcal{D}^* \in \mathbb{D}}{\arg\min} \;\; \mathbb{E}_{\mathcal{D}_I}\left[\mathcal{C}(\mathcal{D}^*, \mathcal{D}_I) \mid \mathbf{m}_{1:k}\right] \tag{4}$$

where $\mathbb{E}_{\mathcal{D}_I}[\cdot]$ is the expected value over possible intended destinations $\mathcal{D}_I$, and $\mathcal{C}(\mathcal{D}^*, \mathcal{D}_I)$ is the cost of deciding $\mathcal{D}^*$ as the endpoint given $\mathcal{D}_I$ is the true intended destination. An intuitive classification strategy is to select the most probable target using

$$\hat{I}(t_k) = \underset{\mathcal{D}^* \in \mathbb{D}}{\arg\max} \;\; P(\mathcal{D}_I = \mathcal{D}^* \mid \mathbf{m}_{1:k}), \tag{5}$$

which is the Maximum *a Posteriori* (MAP) estimate. It can be seen that (5) is a special case of (4) if the binary decision criterion $\mathcal{C}(\mathcal{D}^*, \mathcal{D}_I) = 1$ if $\mathcal{D}^* \neq \mathcal{D}_I$ and $\mathcal{C}(\mathcal{D}^*, \mathcal{D}_I) = 0$ otherwise, is applied, since

$$\mathbb{E}_{\mathcal{D}_I}[\mathcal{C}(\mathcal{D}^*, \mathcal{D}_I) \mid \mathbf{m}_{1:k}] = \sum_{i=1}^{N} \mathcal{C}(\mathcal{D}^*, \mathcal{D}_i) P\left(\mathcal{D}_I = \mathcal{D}_i | \, \mathbf{m}_{1:k}\right). \tag{6}$$

For simplicity, the MAP estimate in (5) is adopted henceforth although more elaborate cost functions can be devised. These can also be applied to groups $\mathbb{D}_j \subset \mathbb{D}$ rather than individual icons; such costing strategies are not explored in this paper. For a pointing task of duration t_T, correct intent inference at t_k can reduce the pointing time by $t_T - t_k$.

B. Inference Requirements

Given the constraints of a typical vehicle environment, a suitable predictor should possess the following features [28]:

- **Computational Efficiency:** this is crucial for a real-time implementation in a vehicle environment where the available computing resources are limited and a pointing task is often completed within a second.
- **Belief-based:** IVIS applications can have different accuracy requirements. It is important that the predictor convey a level of certainty along with any inference. Dynamically estimating the probability of each destination allows flexibility in applying pointing facilitation schemes.
- **Case independent:** reliable predictors are expected to be applicable to a wide range of possible scenarios, IVIS functionalities and GUI designs, and should be independent of the selections sequence, GUI layout, etc.
- **Adaptable:** the characteristics of the pointing gesture can be affected by many factors, including the user's physical ability, prior experience and driving/road conditions. The intent predictor should be able to make use of any available priors on the user's behaviour or road/driving conditions to refine its results.

As illustrated below, the probabilistic inference system proposed in this paper meets these requirements.

IV. Bayesian Intent Inference

Using Bayes' theorem, the probabilities of the nominal destinations can be expressed as

$$P\left(\mathcal{D}_I = \mathcal{D}_i | \, \mathbf{m}_{1:k}\right) \propto P(\mathcal{D}_I = \mathcal{D}_i) P(\mathbf{m}_{1:k} | \mathcal{D}_I = \mathcal{D}_i), \tag{7}$$

for $i = 1, 2, ..., N$. The priors $P\left(\mathcal{D}_I = \mathcal{D}_i\right)$, $\mathcal{D}_i \in \mathbb{D}$, summarise existing knowledge about the probability of various endpoints in $\mathbb{D}$ being the intended one, before any pointing data is observed. Uninformative priors can be constructed by assuming that all possible destinations are equally probable, i.e. $P(\mathcal{D}_I = \mathcal{D}_i) = 1/N$ for $i = 1, 2, ..., N$; this is used in the experiments in Section V. However, if priors are available based on relevant contextual information, such as target selection history, interface design or user profile, they can easily be incorporated as per (7).
Linear destination-reverting models are proposed here, allowing the likelihood $\mathcal{L}_i(t_k) = P(\mathbf{m}_{1:k} \mid \mathcal{D}_I = \mathcal{D}_i)$ to be estimated for all possible destinations, $i = 1, 2, ..., N$, using Kalman filtering. These models require an intended target as an input parameter and model the pointing finger motion accordingly. Since the true intended destination is unknown, they must be evaluated for each $\mathcal{D}_I \in \mathbb{D}$ in order to determine the corresponding likelihoods. We assume that the observed $\mathbf{m}_k$ at t_k is derived as a noisy measurement of a true, but unknown, underlying pointing finger position $\mathbf{c}_n = [x_{t_n} \;\; y_{t_n} \;\; z_{t_n}]'$, and that $\dot{\mathbf{c}}_n = [\dot{x}_{t_n} \;\; \dot{y}_{t_n} \;\; \dot{z}_{t_n}]'$ is the true finger velocity vector.
The (latent) state $\mathbf{s}_{I,k}$ of a destination-reverting model at time t_k includes an estimate of the true pointing finger position and may also encompass further properties such as true finger velocity. It follows a linear Gaussian motion model

$$\mathbf{s}_{I,k} = \mathbf{F}_{I,k}\mathbf{s}_{I,k-1} + \boldsymbol{\kappa}_{I,k} + \mathbf{w}_k \tag{8}$$

where $\mathbf{s}_{I,k-1}$ and $\mathbf{s}_{I,k}$ are the latent state vectors at two consecutive observation times t_{k-1} and t_k respectively. The deterministic transition matrix $\mathbf{F}_{I,k}$ moves the state from time t_{k-1} to t_k, whilst $\boldsymbol{\kappa}_{I,k}$ is a control parameter; both

of these can be dependent on the destination $\mathcal{D}_I$. The term $\mathrm{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ represents noise in the motion model and is modelled as a zero mean Gaussian random vector of covariance $\mathbf{Q}_k$.

The linear observation model that maps the state space into the observation space is given by

$$\mathbf{m}_k = \mathbf{H}_k \mathbf{s}_{I,k} + \mathbf{n}_k. \tag{9}$$

where $\mathbf{H}_k$ is an observation matrix mapping from the hidden state to the observed measurement, and $\mathbf{n}_k \sim \mathcal{N}(0, \mathbf{R}_k)$ is zero mean Gaussian noise with covariance $\mathbf{R}_k$. The pointing movements along the x, y and z axes are assumed to be independent, as is common in many tracking models [42].

A. Destination-Reverting Dynamic Models

Equations (8) and (9) can represent any system with a linear Gaussian dynamic and observation model. Inference of the hidden state $\mathbf{s}_{I,k}$ in such systems can be performed efficiently using the Kalman filter. This makes any model that can be cast in such a form of particular interest in real-time applications. In this section, two particular destination-reverting models, which fit within the proposed framework and are suitable for intent inference, are introduced. They are dubbed the Mean Reverting Diffusion (MRD) and Equilibrium Reverting Velocity (ERV) models.

1) MRD Model: In continuous-time, the movements of the pointing finger are modelled as a multivariate Ornstein-Uhlenbeck process with a mean-reverting term [43]. The evolution of the system state is governed by the following stochastic differential equation

$$d\mathbf{s}_{I,t} = \mathbf{\Lambda}\left(\mathbf{d}_I - \mathbf{s}_{I,t}\right) dt + \boldsymbol{\sigma} d\mathbf{w}_t, \tag{10}$$

where $\mathbf{s}_{I,t} = \mathbf{c}_t$, i.e. the system state consists of the finger position only. This model captures the premise that the motion of the pointing finger 'reverts' towards the intended destination. The expected motion of the finger will be in the direction of the target, and more strongly so if the finger is further away from the destination. This reflects the latter part of the typical velocity profile of pointing tasks as illustrated in Figure 4. Users tend to move relatively fast towards $\mathcal{D}_I$ located at $\mathbf{d}_I$ during the initial pointing stage, with frequent diversions from the shortest path, slowing down as they approach the intended destination.

The diagonal matrix $\mathbf{\Lambda} = \mathrm{diag}\{\lambda_x, \lambda_y, \lambda_z\}$ dictates the mean reversion rates in each dimension. The process $\mathbf{w}_t$ is a standard Wiener process of unit variance, with the matrix $\boldsymbol{\sigma} = \mathrm{diag}\{\sigma_x, \sigma_y, \sigma_z\}$ specifying the standard deviation of the noise component of the motion process in each dimension. It can be useful to choose different parameter values along the axis perpendicular to the interactive display, as this axis can exhibit a different velocity profile to the others.

By integrating (10) over the time interval $\mathcal{T} = [t, t+\tau]$, we obtain expressions for the terms in the general motion model in (8), with

$$\mathbf{F}_{I,k} = e^{-\mathbf{\Lambda}\tau_k}, \quad \boldsymbol{\kappa}_{I,k} = \left[\mathbf{I}_3 - e^{-\mathbf{\Lambda}\tau_k}\right] \mathbf{d}_I \tag{11}$$

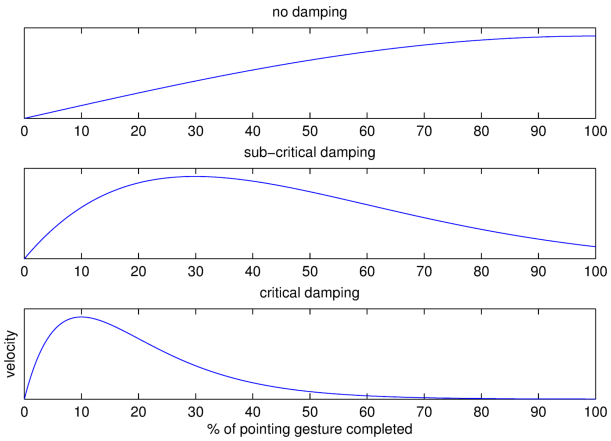


Figure 5: Expected velocity profile versus the proportion of pointing gesture completed (in time) for different levels of damping (none, sub-critical and critical) in the ERV model.

where $\tau_k = t_k - t_{k-1}$ is the time step (see Appendix A for further details). It follows that

$$P\left(\mathbf{s}_{I,k} \mid \mathbf{s}_{I,k-1}\right) = \mathcal{N}\left(\mathbf{s}_{I,k}; \mathbf{F}_{I,k}\mathbf{s}_{I,k-1} + \boldsymbol{\kappa}_{I,k}, \mathbf{Q}_k\right) \tag{12}$$

where

$$\mathbf{Q}_k = \left[\frac{\mathbf{I}_3 - e^{-2\mathbf{\Lambda}\tau_k}}{2\mathbf{\Lambda}}\right] \boldsymbol{\sigma}^2. \tag{13}$$

In this case, the state vector $\mathbf{s}_{I,t} = [x_t \;\; y_t \;\; z_t]' \in \mathbb{R}^3$ is an estimate of the pointing finger position at time t and the observation matrix in (9) is the identity matrix, $\mathbf{H}_k = \mathbf{I}_3$.

2) ERV Model: In this model, the destination is assumed to exert an attractive force with strength proportional to distance away from its centre $\mathbf{d}_I$; a similar nonlinear model was proposed in [44]. Its physical interpretation is that the pointing finger is drawn towards the destination as if by a spring of zero natural length attached to the intended target. In reality, the 'force' directing the finger towards the destination is provided by the action of the user, thus this model is a crude approximation of the complex control system employed by a user when moving their finger towards a target. A consequence of the ERV model is that the attraction force (and therefore modelled acceleration of the finger) is greatest when the finger is far from the intended destination. By incorporating a linear damping term, this model can produce a velocity profile similar to that observed in Figure 4. An initial acceleration will cause velocity to reach a maximum, followed by a gradual decline towards the destination. Figure 5 illustrates expected velocity profiles from the ERV model for several levels of damping coefficients.

The state vector of this model includes the velocity of the pointing finger and is given by $\mathbf{s}_{I,t} = [x_t \;\; \dot{x}_t \;\; y_t \;\; \dot{y}_t \;\; z_t \;\; \dot{z}_t]'$. The evolution of the system for an intended destination $\mathcal{D}_I$ can be described by the stochastic differential equation

$$d\mathbf{s}_{I,t} = \mathbf{A}\left(\boldsymbol{\mu}_I - \mathbf{s}_{I,t}\right) dt + \boldsymbol{\sigma} d\mathbf{w}_t, \tag{14}$$

where the mean $\boldsymbol{\mu}_I = [d_{x,I} \;\; 0 \;\; d_{y,I} \;\; 0 \;\; d_{z,I} \;\; 0]'$ specifies the coordinates of the destination and $\mathbf{w}_t$ is a standard Wiener process. The matrix $\mathbf{A}$ is block-diagonal, given by

$\mathbf{A} = \text{diag}\{\mathbf{A}_x, \mathbf{A}_y, \mathbf{A}_z\}$, where

$$\mathbf{A}_x = \begin{bmatrix} 0 & -1 \\ \eta_x & \rho_x \end{bmatrix} \quad \mathbf{A}_y = \begin{bmatrix} 0 & -1 \\ \eta_y & \rho_y \end{bmatrix} \quad \mathbf{A}_z = \begin{bmatrix} 0 & -1 \\ \eta_z & \rho_z \end{bmatrix}$$

such that η_x, η_y and η_z set the strength of the restoration force along the corresponding axis (physically this can be interpreted as spring strength). The coefficients ρ_x, ρ_y and ρ_z represent the strength of damping in each direction and are an essential component in modelling the velocity profile. Whilst critical damping might seem like a natural choice, it implies zero pointing finger velocity at the destination, which contradicts the velocity profiles displayed in Figure 4. Sub-critical damping (overdamping) is therefore expected to best model the pointing movement in most tasks, see Figure 5. The level of additive Gaussian noise present in the dynamics is set by $\boldsymbol{\sigma} = \text{diag}\{\boldsymbol{\sigma}_x, \boldsymbol{\sigma}_y, \boldsymbol{\sigma}_z\}$ where $\boldsymbol{\sigma}_x = \text{diag}\{\sigma_{x,1}, \sigma_{x,2}\}$, $\boldsymbol{\sigma}_y = \text{diag}\{\sigma_{y,1}, \sigma_{y,2}\}$ and $\boldsymbol{\sigma}_z = \text{diag}\{\sigma_{z,1}, \sigma_{z,2}\}$; $\sigma_{x,1}$ and $\sigma_{x,2}$ specify the standard deviation of noise along the $x-axis$ in the position and velocity components, respectively (elements of $\boldsymbol{\sigma}_y$ and $\boldsymbol{\sigma}_z$ are similarly defined).
By integrating (14) over the time interval $\mathcal{T} = [t, t+\tau]$, expressions for the terms in the general motion model in (8) can be derived as

$$\begin{aligned} \mathbf{F}_{I,k} &= \text{diag}\left\{e^{-\mathbf{A}_x\tau_k}, e^{-\mathbf{A}_y\tau_k}, e^{-\mathbf{A}_z\tau_k}\right\}, \\ \boldsymbol{\kappa}_{I,k} &= \begin{bmatrix} \left(\mathbf{I}_2 - e^{-\mathbf{A}_x\tau_k}\right)[b_{x,I} \;\; 0]' \\ \left(\mathbf{I}_2 - e^{-\mathbf{A}_y\tau_k}\right)[b_{y,I} \;\; 0]' \\ \left(\mathbf{I}_2 - e^{-\mathbf{A}_z\tau_k}\right)[b_{z,I} \;\; 0]' \end{bmatrix}; \end{aligned} \tag{15}$$

see Appendix A for more details. It follows that

$$P\left(\mathbf{s}_{I,k} \mid \mathbf{s}_{I,k-1}\right. = \mathcal{N}\left(\mathbf{s}_{I,k}; \mathbf{F}_k\mathbf{s}_{I,k-1} + \boldsymbol{\kappa}_{I,k}, \mathbf{Q}_k\right) \tag{16}$$

where

$$\mathbf{Q}_k = \text{diag}\left\{\boldsymbol{\chi}_{k,x}, \boldsymbol{\chi}_{k,y}, \boldsymbol{\chi}_{k,z}\right\}, \tag{17}$$

with $\boldsymbol{\chi}_{k,x} = \boldsymbol{\chi}_{k,x,2}\left(\boldsymbol{\chi}_{k,x,4}\right)^{-1}$, $\boldsymbol{\chi}_{k,y} = \boldsymbol{\chi}_{k,y,2}\left(\boldsymbol{\chi}_{k,y,4}\right)^{-1}$, $\boldsymbol{\chi}_{k,z} = \boldsymbol{\chi}_{k,z,2}\left(\boldsymbol{\chi}_{k,z,4}\right)^{-1}$, such that

$$\begin{bmatrix} \boldsymbol{\chi}_{k,x,1} & \boldsymbol{\chi}_{k,x,2} \\ \boldsymbol{\chi}_{k,x,3} & \boldsymbol{\chi}_{k,x,4} \end{bmatrix} = exp\left(\begin{bmatrix} -\mathbf{A}_x & \boldsymbol{\sigma}_x\boldsymbol{\sigma}_x' \\ \mathbf{0} & \mathbf{A}_x' \end{bmatrix}\tau_k\right),$$

$$\begin{bmatrix} \boldsymbol{\chi}_{k,y,1} & \boldsymbol{\chi}_{k,y,2} \\ \boldsymbol{\chi}_{k,y,3} & \boldsymbol{\chi}_{k,y,4} \end{bmatrix} = exp\left(\begin{bmatrix} -\mathbf{A}_y & \boldsymbol{\sigma}_y\boldsymbol{\sigma}_y' \\ \mathbf{0} & \mathbf{A}_y' \end{bmatrix}\tau_k\right),$$

and

$$\begin{bmatrix} \boldsymbol{\chi}_{k,z,1} & \boldsymbol{\chi}_{k,z,2} \\ \boldsymbol{\chi}_{k,z,3} & \boldsymbol{\chi}_{k,z,4} \end{bmatrix} = exp\left(\begin{bmatrix} -\mathbf{A}_z & \boldsymbol{\sigma}_z\boldsymbol{\sigma}_z' \\ \mathbf{0} & \mathbf{A}_z' \end{bmatrix}\tau_k\right).$$

Since the model state includes velocity elements, the observation matrix is given by

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Linear destination-reverting models with states that embody higher order kinematics, e.g. acceleration and/or jerks, can be also considered within the framework given here.

B. Sequential Likelihood Evaluation

In the systems described by (10) and (14), the model state is explicitly dependent on the destination, which is not known *a priori*. However, by considering N such models, one for each possible endpoint, $\mathcal{D}_i$ that leads to a model best explaining the observed partial pointing trajectory $\mathbf{m}_{1:k}$ is assigned the highest probability of being the intended destination $\mathcal{D}_I$.
The likelihood of the partially observed trajectory up to time t_k can be written as

$$\begin{aligned} P(\mathbf{m}_{1:k} \mid \mathcal{D}_I = \mathcal{D}_i) = P(\mathbf{m}_k \mid \mathbf{m}_{1:k-1}, \mathcal{D}_I = \mathcal{D}_i)\times \\ P(\mathbf{m}_{k-1} \mid \mathbf{m}_{1:k-2}, \mathcal{D}_I = \mathcal{D}_i), ... \times P(\mathbf{m}_1 | \mathcal{D}_I = \mathcal{D}_i). \end{aligned} \tag{18}$$

This implies that calculating the Prediction Error Decomposition (PED) $P\left(\mathbf{m}_k \mid \mathbf{m}_{1:k-1}, \mathcal{D}_I = \mathcal{D}_i\right)$ after each measurement is sufficient to sequentially obtain the likelihood $P(\mathbf{m}_{1:k} \mid \mathcal{D}_I = \mathcal{D}_i)$ for each $\mathcal{D}_i \in \mathbb{D}$, using (18) as outlined below. This enables inference of the intended destination $\mathcal{D}_I$ at t_k via the MAP estimator in (5). For intent inference, the primary objective is the estimation of the observation likelihoods $\mathcal{L}_i(t_k)$ given the N nominal destinations, rather than estimating the posterior distribution of the hidden state, as in traditional tracking applications [42], [45]. Nonetheless, the hidden state can be inferred as described in Section IV-C. To simplify the notation, let

$$\begin{aligned} \hat{\mathbf{e}}_{i,k|k-1} &= \mathbb{E}\left[\mathbf{e}_k \mid \mathbf{m}_{1:k-1}, \mathcal{D}_I = \mathcal{D}_i\right] \\ &= \int_{\mathbb{R}^n} \mathbf{e}_k P\left(\mathbf{e}_k \mid \mathbf{m}_{1:k-1}, \mathcal{D}_I = \mathcal{D}_i\right) d\mathbf{e}_{i,k} \end{aligned} \tag{19}$$

be the *predicted* mean (point estimate) of the arbitrary vector $\mathbf{e}_k \in \mathbb{R}^n$ and let its predicted covariance be

$$\begin{aligned} \mathbf{P}^{\mathbf{ee}}_{i,k|k-1} = \mathbb{E}\left[\left(\mathbf{e}_k - \hat{\mathbf{e}}_{k|k-1}\right)\left(\mathbf{e}_k - \hat{\mathbf{e}}_{k|k-1}\right)'\right] \triangleq \int_{\mathbb{R}^n} (\mathbf{e}_k \\ -\hat{\mathbf{e}}_{i,k|k-1})\left(\mathbf{e}_k - \hat{\mathbf{e}}_{i,k|k-1}\right)' P\left(\mathbf{e}_k \mid \mathbf{m}_{1:k-1}, \mathcal{D}_I = \mathcal{D}_i\right) d\mathbf{e}_{i,k}, \end{aligned} \tag{20}$$

where $\mathcal{D}_i \in \mathbb{D}$ is the destination. Also, the *corrected* point estimate given by $\hat{\mathbf{e}}_{i,k|k} = \mathbb{E}\left[\mathbf{e}_k \mid \mathbf{m}_{1:k}, \mathcal{D}_I = \mathcal{D}_i\right]$ and $\mathbf{P}^{\mathbf{ee}}_{i,k|k} = \mathbb{E}\left[\left(\mathbf{e}_k - \hat{\mathbf{e}}_{k|k}\right)\left(\mathbf{e}_{i,k} - \hat{\mathbf{e}}_{k|k}\right)'\right]$ is similarly defined.
The Chapman-Kolmogorov identity states that the predicted distribution of the system state is given by $P\left(\mathbf{s}_{i,k} \left|\mathbf{m}_{1:k-1}\right.\right) = \int_{\mathbb{R}^n} P\left(\mathbf{s}_{i,k} \left|\mathbf{s}_{i,k-1}\right.\right) P\left(\mathbf{s}_{i,k-1} \left|\mathbf{m}_{1:k-1}\right.\right) d\mathbf{s}_{i,k-1}$ and thereby the *predictive* distribution of the next observation is

$$\begin{aligned} P\left(\mathbf{m}_k \left|\mathbf{m}_{1:k-1}, \mathcal{D}_I = \mathcal{D}_i\right.\right) = \int_{\mathbb{R}^n} P\left(\mathbf{m}_k \left|\mathbf{s}_{i,k}\right.\right) \\ \times P\left(\mathbf{s}_{i,k} \left|\mathbf{m}_{1:k-1}\right.\right) d\mathbf{s}_{i,k}. \end{aligned}$$

This leads to a predictive mean given by

$$\begin{aligned} \hat{\mathbf{m}}_{i,k|k-1} &= \int_{\mathbb{R}^3}\int_{\mathbb{R}^n} \mathbf{m}_k P\left(\mathbf{m}_k \mid \mathbf{s}_{i,k}\right) P\left(\mathbf{s}_{i,k} \mid \mathbf{m}_{1:k-1}\right) d\mathbf{s}_{i,k} d\mathbf{m}_k \\ &= \mathbf{H}_k\hat{\mathbf{s}}_{i,k|k-1} \end{aligned} \tag{21}$$

since the observation noise is zero-mean, $\mathbb{E}\left[\mathbf{m}_k \mid \mathbf{s}_{i,k}\right] = \mathbf{H}_k\mathbf{s}_{i,k}$, and the mean of the predictive state-vector is given by $\hat{\mathbf{s}}_{i,k|k-1} = \int_{\mathbb{R}^n} \mathbf{s}_{i,k} P\left(\mathbf{s}_k \mid \mathbf{m}_{1:k-1}\right) d\mathbf{s}_{i,k}$, as per (19).

Correspondingly, the predictive covariance reduces to

$$\begin{aligned}\mathbf{P}^{\mathbf{mm}}_{i,k|k-1} &= \int_{\mathbb{R}^n} \left[\mathbf{H}_k\mathbf{s}_{i,k} + \mathbf{n}_k - \hat{\mathbf{m}}_{i,k|k-1}\right]\left[\mathbf{H}_k\mathbf{s}_{i,k} + \mathbf{n}_k \right. \\ &\quad \left. - \hat{\mathbf{m}}_{i,k|k-1}\right]' P\left(\mathbf{s}_{i,k}\,|\mathbf{m}_{1:k-1}\right) d\mathbf{s}_{i,k} \\ &= \mathbf{H}_k\mathbf{P}^{\mathbf{ss}}_{i,k|k-1}\mathbf{H}'_k + \mathbf{R}_k. \end{aligned} \tag{22}$$

such that $\mathbf{P}^{\mathbf{ss}}_{i,k|k-1} = \int_{\mathbb{R}^n}\left[\mathbf{s}_{i,k} - \hat{\mathbf{s}}_{i,k|k-1}\right]\left[\mathbf{s}_{i,k} - \hat{\mathbf{s}}_{i,k|k-1}\right]' \times P\left(\mathbf{s}_{i,k}\,|\mathbf{m}_{1:k-1}\right) d\mathbf{s}_{i,k}$. This is based on the assumption that the mapped state $\mathbf{H}_k\mathbf{s}_{i,k}$ is independent of the observation noise $\mathbf{n}_k$. The predictive state mean $\hat{\mathbf{s}}_{i,k|k-1}$ and covariance $\mathbf{P}^{\mathbf{ss}}_{i,k|k-1}$ are conditioned on all but the current observation $\mathbf{m}_k$ at time t_k. Thus, the PED for $\mathcal{D}_I = \mathcal{D}_i$ is

$$P\left(\mathbf{m}_k\,|\mathbf{m}_{1:k-1}, \mathcal{D}_I = \mathcal{D}_i\right) = \mathcal{N}\left(\mathbf{m}_k; \hat{\mathbf{m}}_{i,k|k-1}, \mathbf{P}^{\mathbf{mm}}_{i,k|k-1}\right). \tag{23}$$

The predictive state distribution is given by

$$P\left(\mathbf{s}_{i,k}\,|\mathbf{m}_{1:k-1}\right) = \mathcal{N}\left(\mathbf{s}_{i,k}; \hat{\mathbf{s}}_{i,k|k-1}, \mathbf{P}^{\mathbf{ss}}_{i,k|k-1}\right). \tag{24}$$

Its mean and covariance are required to calculate $\hat{\mathbf{m}}_{i,k|k-1}$ and $\mathbf{P}^{\mathbf{mm}}_{i,k|k-1}$ in (23). They can be deduced in a similar way to (21)-(23) and are defined by

$$\hat{\mathbf{s}}_{i,k|k-1} = \mathbf{F}_{i,k}\mathbf{s}_{i,k-1|k-1} + \boldsymbol{\kappa}_{i,k} \tag{25}$$

and

$$\mathbf{P}^{\mathbf{ss}}_{i,k|k-1} = \mathbf{F}_{i,k}\mathbf{P}^{\mathbf{ss}}_{i,k-1|k-1}\mathbf{F}^T_{i,k} + \mathbf{Q}_k, \tag{26}$$

noting that the dynamic noise $\mathbf{w}_k$ is uncorrelated with $(\mathbf{F}_{i,k}\mathbf{s}_k + \boldsymbol{\kappa}_{i,k})$ and has a zero mean. The previously estimated model state $\hat{\mathbf{s}}_{i,k-1|k-1}$ and the estimation covariance matrix $\mathbf{P}^{\mathbf{ss}}_{i,k-1|k-1}$ utilise all the available observations $\mathbf{m}_{1:k-1}$ at the previous time instant t_{k-1}. Thus, at t_k they are available when computing (25) and (26).

To determine $\hat{\mathbf{s}}_{i,k|k}$ and $\mathbf{P}^{\mathbf{ss}}_{i,k|k}$ necessary for calculating the likelihoods at the next time step t_{k+1}, i.e. $\mathcal{L}_i(t_{k+1})$, via (18), (23) and (24), the Kalman filtering update equation is applied. It produces

$$\hat{\mathbf{s}}_{i,k|k} = \hat{\mathbf{s}}_{i,k|k-1} + \mathbf{G}_{i,k}\left(\mathbf{m}_k - \hat{\mathbf{m}}_{i,k|k-1}\right) \tag{27}$$

and

$$\mathbf{P}^{\mathbf{ss}}_{i,k|k} = \mathbf{P}^{\mathbf{ss}}_{i,k|k-1} - \mathbf{G}_{i,k}\mathbf{P}^{\mathbf{mm}}_{i,k|k-1}\mathbf{G}'_{i,k} \tag{28}$$

such that $\mathbf{G}_{i,k} = \mathbf{P}^{\mathbf{sm}}_{i,k|k-1}\left(\mathbf{P}^{\mathbf{mm}}_{i,k|k-1}\right)^{-1}$ is the Kalman gain and $\mathbf{P}^{\mathbf{sm}}_{i,k|k-1} = \mathbf{P}^{\mathbf{ss}}_{i,k|k-1}\mathbf{H}'_k$ [45].

Algorithm 1 details a sequential implementation of the proposed probabilistic intent inference approach whose block diagram, including a bank of N Kalman filters, is depicted in Figure 6. In the pseudo-code, each of $P(\mathbf{m}_{1:k} \mid \mathcal{D}_I = \mathcal{D}_i)$ and $P\left(\mathcal{D}_I = \mathcal{D}_i \mid \mathbf{m}_{1:k}\right)$ for $\mathcal{D}_i \in \mathbb{D}$ are calculated with the arrival of a new observation from the gesture-tracker. Normalization is performed to ensure that $\sum_{i=1}^{N} P\left(\mathcal{D}_I = \mathcal{D}_i \mid \mathbf{m}_{1:k}\right) = 1$. At the initial time t_1, initialisation of the inputs $\hat{\mathbf{s}}_{i,1}$, τ_1 and $\mathbf{P}^{\mathbf{ss}}_{i,1|1}$ is based on prior knowledge of their possible values.

Applying Algorithm 1 also requires specifying the model parameters, such as the reversion matrices ($\boldsymbol{\Lambda}$ or $\mathbf{A}$) and dynamic noise matrix $\boldsymbol{\sigma}$. The state transition and observation functions are independent of time for the MRD and ERV models. With a constant time step between observations (i.e. for fixed-rate data measurements), the state transition matrix $\mathbf{F}_{i,k}$ and covariance matrix $\mathbf{Q}_k$ are the same for all destinations in both models, and thus need only be calculated once at the start of the algorithm. Similarly, the observation noise $\mathbf{R}_n$ covariance is often independent of time.

The Kalman filter can be calculated efficiently. It has a computational cost of order $\mathcal{O}(n^3)$, where n is the dimension of model state in (8), e.g. $n = 3$ for MRD and $n = 6$ for ERV. The proposed approach entails running N simultaneous such filters. Since these are independent, the method is straightforward to parallelise. Given the minimal parameter training required for the destination-reverting models and the low dimensionality of their states, the proposed prediction framework is computationally efficient and lends itself to real-time implementation. For example, an unoptimised MATLAB implementation of the destination-reverting predictor for $N = 21$ destinations can be computed in under 7ms on a standard desktop PC (with Intel i7 CPU running at 3.4 GHz) at each step (i.e. per observation).

Algorithm 1 Proposed Sequential Intent Inference

Input: $P(\mathcal{D}_I = \mathcal{D}_i)$ (priors), $\mathbf{d}_i$ (location of the nominal destinations) $i = 1, 2, ...N$, $\mathbf{R}_n$ (observation noise covariance), $\boldsymbol{\sigma}$ (state transition noise standard deviation), model parameters (e.g. $\boldsymbol{\Lambda}$ for MRD and $\mathbf{A}$ for ERV).
Initialise $\hat{\mathbf{s}}_{i,1}$, τ_1 , $\mathbf{P}^{\mathbf{ss}}_{i,1|1}$.
for each observation ($\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_T$ captured at time instants $t_k = t_1, t_2, ..., t_T$) **do**
– Calculate τ_k (time step), $\mathbf{H}_k$ (observation matrix) and $\mathbf{Q}_k$ (dynamic noise covariance).
for (each possible destination) $i = 1...N$ **do**
– Calculate $\boldsymbol{\kappa}_{i,k}$ (state control parameter) and $\mathbf{F}_{i,k}$ (state transition function).
– Calculate $\hat{\mathbf{s}}_{i,k|k-1}$ via (25) and $\mathbf{P}^{\mathbf{ss}}_{i,k|k-1}$ via (26).
– Calculate $\hat{\mathbf{m}}_{i,k|k-1}$ and $\mathbf{P}^{\mathbf{mm}}_{i,k|k-1}$ via (21) and (22).
– Calculate $P\left(\mathbf{m}_k\,|\mathbf{m}_{1:k-1}, \mathcal{D}_I = \mathcal{D}_i\right)$ in (23).
– Calculate $P(\mathbf{m}_{1:k}|\mathcal{D}_I = \mathcal{D}_i)$ via (18).
– Compute $\hat{\mathbf{s}}_{i,k|k}$ via (27) and $\mathbf{P}^{\mathbf{ss}}_{i,k|k}$ via (28) to be utilised in the calculations for the next observation.
– Compute the destination *unnormalised* probability $\hat{P}\left(\mathcal{D}_I = \mathcal{D}_i|\,\mathbf{m}_{1:k}\right) = P(\mathcal{D}_I = \mathcal{D}_i)P(\mathbf{m}_{1:k}|\mathcal{D}_I = \mathcal{D}_i)$
end for
– Determine the probability of each destination via

$$P\left(\mathcal{D}_I = \mathcal{D}_i|\,\mathbf{m}_{1:k}\right) \approx \frac{\hat{P}\left(\mathcal{D}_I = \mathcal{D}_i|\,\mathbf{m}_{1:k}\right)}{\sum_{\mathcal{D}_i \in \mathbb{D}} \hat{P}\left(\mathcal{D}_I = \mathcal{D}_i|\,\mathbf{m}_{1:k}\right)}.$$

– Infer the MAP destination $\hat{I}(t_k)$ via (5).
end for

C. Hidden State Estimation

The filters described above can be utilised to estimate the pointing trajectory $\mathbf{c}_{1:k}$ that is free of unintentional perturbation-generated movements. This can be achieved by calculating the posterior distribution of the portions of the state vector corresponding to the pointing finger position. The

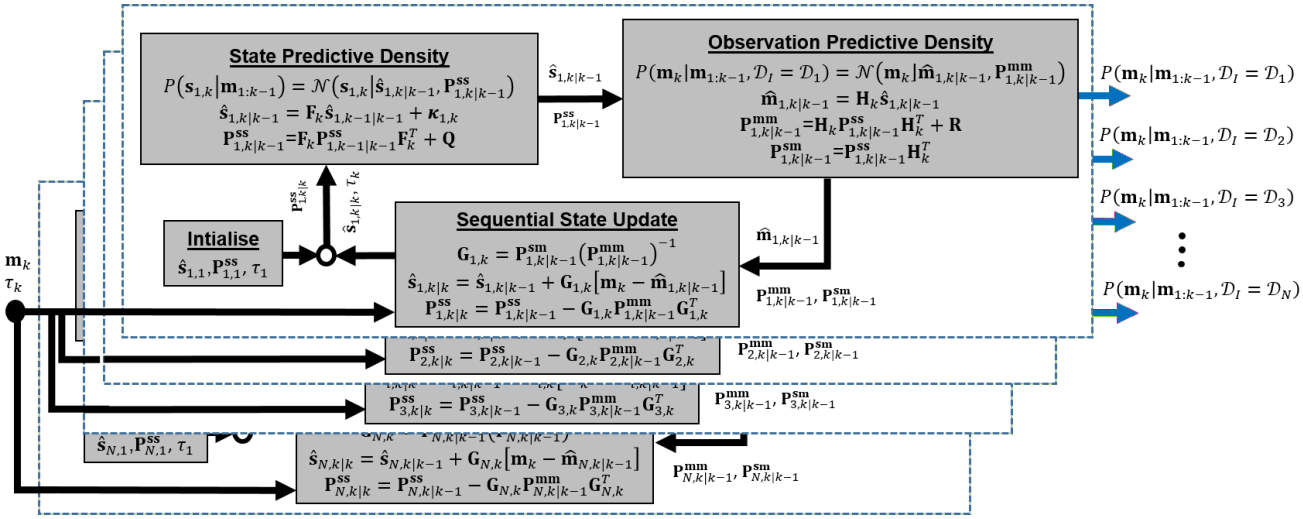


Figure 6: Block diagram for sequentially determining the prediction error decomposition for N nominal destinations.

posterior distribution of the state $\mathbf{s}_k$ at time t_k is a Gaussian mixture distribution. It is given by a weighted mixture of the conditional posterior estimates corresponding to each of the N destinations, $\mathcal{D}_i \in \mathbb{D}$, and can be expressed by

$$P(\mathbf{s}_k|\mathbf{m}_{1:k}) = \sum_{i=1}^{N} P\left(\mathbf{s}_{i,k}\,|\mathbf{m}_{1:k}\right) P\left(\mathcal{D}_I = \mathcal{D}_i \;\;|\mathbf{m}_{1:k}\right), \quad (29)$$

where the component weights are given by

$$P\left(\mathcal{D}_I = \mathcal{D}_i \mid \mathbf{m}_{1:k}\right) = \frac{P\left(\mathbf{m}_{1:k} \mid \mathcal{D}_I = \mathcal{D}_i\right) P\left(\mathcal{D}_I = \mathcal{D}_i\right)}{\sum_{n=1}^{N} P\left(\mathbf{m}_{1:k} \mid \mathcal{D}_I = \mathcal{D}_n\right) P\left(\mathcal{D}_I = \mathcal{D}_n\right)}. \quad (30)$$

The state distribution $P\left(\mathbf{s}_{i,k} \mid \mathbf{m}_{1:k}\right) = \mathcal{N}\left(\mathbf{s}_{i,k}; \hat{\mathbf{s}}_{i,k|k}, \mathbf{P}^{\mathbf{ss}}_{i,k|k}\right)$ is calculated by the sequential state update step in Figure 6 using (27) and (28), and the likelihoods $P\left(\mathbf{m}_{1:k} \mid \mathcal{D}_I = \mathcal{D}_i\right)$ are calculated in (18).

Hence, the proposed approach not only allows destination prediction, but also permits the estimation of the true underlying pointing gesture trajectory. It combines intent inference and state estimation in a single framework. For very severe perturbations arising from rough terrains and manifesting themselves as significant jolts in the pointing finger movements, more advanced filtering techniques such as Sequential Monte Carlo (SMC) methods can be used to remove nonlinear unintentional components of the pointing gesture during a pre-processing stage as in [13]. In the latter, the perturbations-originated movements are modelled as having a jump-diffusion driven mean-reverting velocity process and a variable rate particle filter implementation is employed to filter the trajectory.

V. Evaluation and Results

This section assesses the performance of the proposed intent inference algorithms using data collected in an instrumented vehicle driven over several types of test tracks; other common benchmark prediction techniques are also examined. The system in Figure 2 is mounted to the car dashboard as depicted in Figure 7. Prediction and analysis is performed by a module running on the touchscreen and the observation collection interval is $\tau_k \approx 20$ms. The evaluation is based on 85 typical pointing tasks conducted by four users in a mobile vehicle for various road conditions. The layout of the experimental GUI is identical to that shown in Figure 2 with $N = 21$ nominal destinations (less than 2cm apart). Each pointing task requires a participant to point at a specified (highlighted) icon on the interface. Tasks that resulted in the user successfully selecting the highlighted icon are considered since the ground truth intention is known; this is necessary to objectively evaluate and compare the prediction models being tested. The task and predictions commence once the user's pointing finger starts moving in the direction of the touchscreen. Figure 8 shows measurements obtained from three complete pointing trajectories, i.e. $\mathbf{m}_{1:T}$. They are recorded under three distinct conditions, which noticeably influence the pointing hand movements. For example, perturbations due to the harsh terrain are clearly visible in the corresponding pointing trajectory, compared to the smooth track recorded whilst the vehicle is stationary.

A. Performance Metrics

Prediction performance is evaluated in terms of the algorithm ability to successfully establish the true intended icon $\mathcal{D}^+$ of a pointing task via the MAP estimator in (5), i.e. the prediction success is defined by $\mathcal{S}(t_k) = 1$ if $\hat{I}(t_k) = \mathcal{D}^+$ and $\mathcal{S}(t_k) = 0$ otherwise, for observations arriving at time $t_k \in \{t_1, t_2, ..., t_T\}$. Here, the following metrics are examined:

1) *Temporal intent inference* (Figure 9): the classification success is displayed against the percentage of the pointing time $t_p = 100 \times t_k/t_T$. This illustrates how early in the pointing gesture the predictor can correctly infer the intended destination.
2) *Aggregate prediction success* (Figure 10): this shows the proportion of the total pointing gesture (in time) for which the algorithm correctly predicted the true intended item on the interface, i.e. $\frac{1}{T}\sum_{k=1}^{T} \mathcal{S}(t_k)$.

3) *Prediction uncertainty* (Figure 11): this captures the level of confidence the inference mechanism has in its predictions. The log uncertainty is given by: $\vartheta(t_k) = -\log_{10}\hat{P}\left(\mathcal{D}_I = \mathcal{D}^+ \mid \mathbf{m}_{1:k}\right)$ where $\hat{P}\left(\mathcal{D}_I = \mathcal{D}^+ \mid \mathbf{m}_{1:k}\right)$ is the estimated posterior probability of the true destination being the intended target at time t_k. It should be noted that high prediction success does not necessarily imply high prediction certainty and vice versa. Nevertheless, it is expected that $\vartheta(t_k) \to 0$ as $t_k \to t_T$ for a reliable predictor.

In each of Figures 9, 10 and 11, the outcomes from 85 pointing tasks are averaged. In all experiments, the predictor is unaware of the trajectory end time and destination when making decisions. Figures 9 and 11 show results after completing 10% of the pointing task, since prior to this no meaningful predictions are observed.

B. Other Prediction Models

In addition to the destination reverting models, the following two benchmark methods are also tested:

- **Nearest Neighbour (NN)**: this is an intuitive model that assigns the highest probability to $\mathcal{D}_i \in \mathbb{D}$ closest to the observed pointing finger position. Thus, we can write $P\left(\mathbf{m}_k \mid \mathcal{D}_I = \mathcal{D}_i\right) = \mathcal{N}\left(\mathbf{m}_k; \mathbf{d}_i, \boldsymbol{\sigma}^2_{\text{NN}}\right)$ where $\mathbf{d}_i$ is the location of the i^{th} destination and $\boldsymbol{\sigma}^2_{\text{NN}}$ is the covariance of the multivariate normal distribution.
- **Bearing Angle (BA)**: this is based on the premise that the user points directly towards the destination, i.e. the cumulative angle to the intended destination should be minimal. The heading angle is assumed to be a zero mean random variable with fixed variance, i.e. $P\left(\mathbf{m}_k \mid \mathbf{m}_{k-1}, \mathcal{D}_I = \mathcal{D}_i\right) = \mathcal{N}\left(\theta_k; 0, \sigma^2_{\text{BA}}\right)$ where $\theta_k = \angle\left(\mathbf{m}_k - \mathbf{m}_{k-1}, \mathbf{d}_i\right)$. Thus, we can express the measurements likelihood as $P(\mathbf{m}_{1:k} \mid \mathcal{D}_I = \mathcal{D}_i) = P\left(\mathbf{m}_1 \mid \mathcal{D}_I = \mathcal{D}_i\right)\prod_{n=1}^{k} P\left(\mathbf{m}_k \mid \mathbf{m}_{k-1}, \mathcal{D}_I = \mathcal{D}_i\right)$ for $\mathcal{D}_i \in \mathbb{D}$ with σ^2_{BA} a design parameter.

C. Design Parameters and Models Training

The trajectories considered are divided into two groups based on the level of perturbation present: relatively 'smooth' tracks, and trajectories including sudden sharp movements (see Figure 8, for example). For each group, a number of sample trajectories (under 20% of the group) are used to train the NN, BA, MRD and ERV models by choosing appropriate parameter values for $\boldsymbol{\sigma}_{\text{NN}}$, σ_{BA}, $\boldsymbol{\sigma}$, $\boldsymbol{\Lambda}$, $\mathbf{A}$ and $\mathbf{R}_n$. These parameters are then employed when applying the methods to the remaining out of sample trajectories in each group. The parameter training criterion is the maximisation of the model likelihood $P(\mathbf{m}_{1:T} \mid \mathcal{D}_I = \mathcal{D}^+, \Omega)$ for the true destination $\mathcal{D}^+$, which is known for the in-sample training trajectories. The parameter set Ω encompasses all the model parameters, e.g. $\Omega = \{\mathbf{A}, \boldsymbol{\sigma}, \mathbf{R}_n\}$ for ERV. Thus,

$$\hat{\Omega} = \arg\max_{\Omega} \prod_{j=1}^{J} P(\mathbf{m}^j_{1:T} \mid \mathcal{D}_i = \mathcal{D}_I, \Omega), \qquad (31)$$

where $\mathbf{m}^j_{1:T}$ is j^{th} complete pointing trajectory in the training set and J is the number of training tracks. Unlike applying the chosen parameters only on the training set as in the preliminarily study in [37], [46], this parameter estimation procedure is more suitable for an operational on-line system and better reflects the system performance in practice. Parameter training is an off-line process, but need only be completed once.

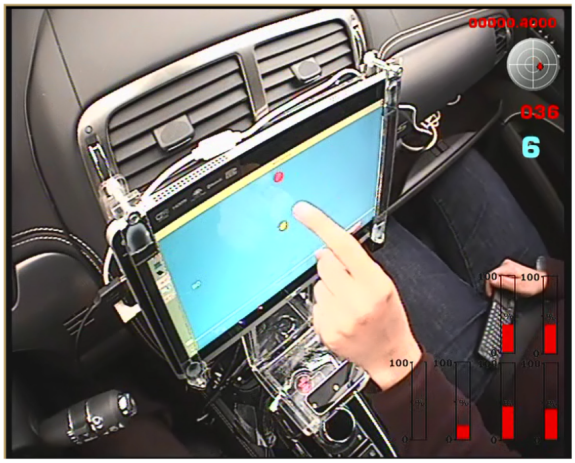


Figure 7: Data collection system in an instrumented vehicle.

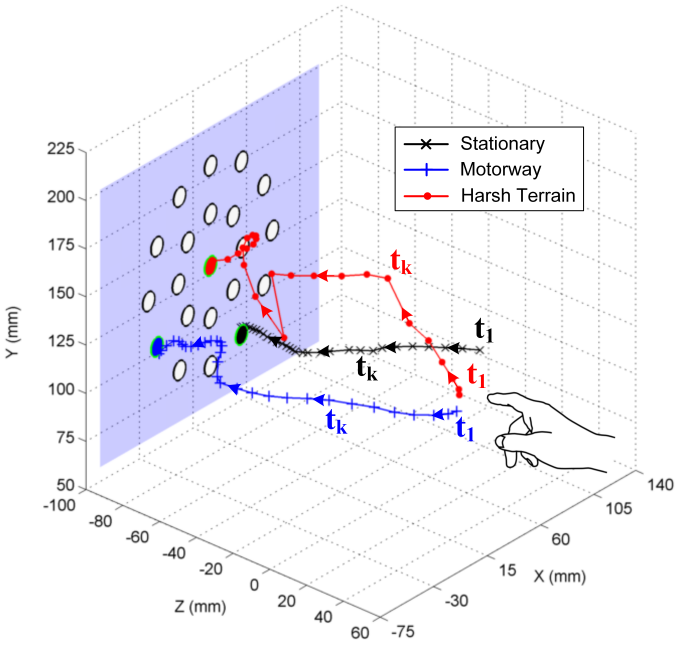


Figure 8: Three typical complete 3D pointing finger-tip trajectories to select the highlighted items on the in-vehicle touchscreen (blue plane); arrows indicate the finger direction of travel over time, with $t_1 < t_k$.

D. Results[1]

Figure 9 shows that the proposed destination inference methods allow prediction of the intended destination significantly earlier in the pointing gesture than benchmark methods.

[1]Please refer to the attached video, *http://link.eng.cam.ac.uk/Main/BIA23*, demonstrating the proposed intent inference algorithms operating in real-time on a sample of typical in-vehicle pointing gestures.

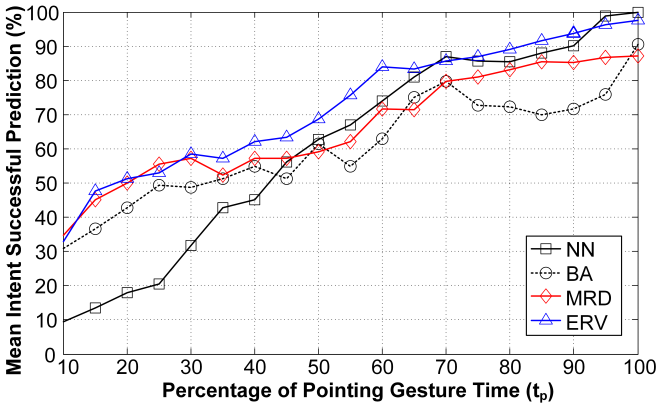


Figure 9: Mean percentage of destination successful prediction as a function of pointing time t_p.

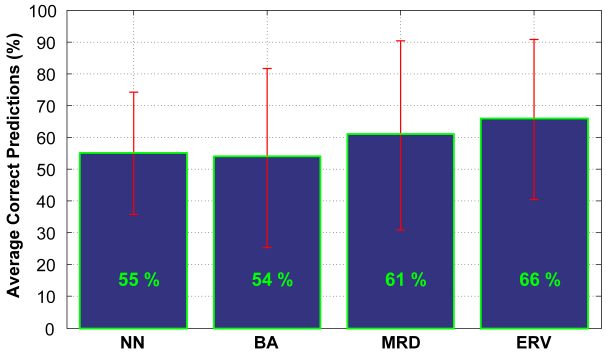


Figure 10: Gesture portion (in time) during which the correct destination is inferred.

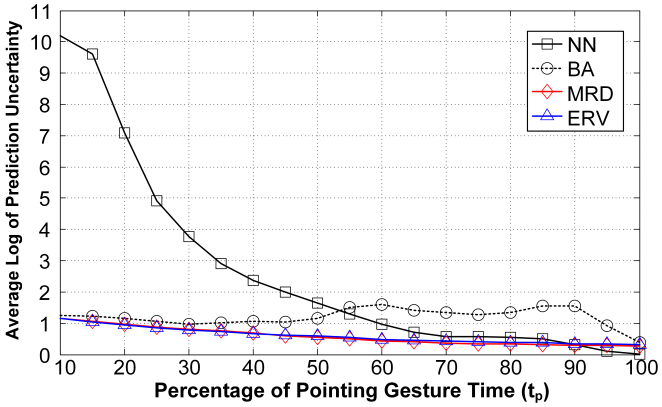


Figure 11: Average log prediction uncertainty.

They significantly outperform the NN, especially in the first 10%-70% of the pointing task (in time). This is the crucial time period for which enabling pointing facilitation regimes can be most effective. 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. For example, the ERV model has a successful prediction rate two to three times that of NN for t_p between 10% and 30% of the total track time. As expected, the performance gap between the ERV and NN diminishes towards the end of the pointing task as the pointing finger becomes inherently close to the destination.

The ERV consistently outperforms the NN and BA (except in the last 5% of the pointing time) in Figure 9, unlike the optimal-control-based predictor as reported in [28]. However, the MRD model falls behind the NN for $t_p \geq 50\%$, after which changes in position and thereby the reverting term effect become limited. Additionally, the bearing angle performance drastically deteriorates with the increase of t_p since the reliability of the heading angle as a measure of intent declines as the the pointing finger gets closer to the destination. For instance, as $t_p \rightarrow 100\%$, θ_k can take almost arbitrarily values, especially for perturbed trajectories, see Figure 3, resulting in the high level of classification errors observed. In the early portion of the pointing gesture, however, BA notably outperforms NN.

In terms of overall prediction success, Figure 10 reveals that the destination-reverting linear models deliver the highest overall correct predictions across the pointing trajectories; ERV has the highest aggregate correct predictions, exceeding 65% of the pointing time. Both NN and BA exhibit similar performance for the relatively large data set examined. The NN model has the lowest variance in correct predictions, as shown by the error bars in Figure 10. This stems from the model simplicity as it has only one design parameter $\boldsymbol{\sigma}_{NN}$, although this robustness is undermined by the NN's distinctly poor performance during the critical time region $t_p \leq 45\%$. The results in Figures 9 and 10 clearly illustrate the superior performance of the MRD and ERV models, especially with regard to early destination prediction. For example, in 60% of cases, the ERV model can make successful destination inference after only 30% of the gesture, potentially reducing pointing time and effort by 70%. Since interactions with displays are very prevalent in modern vehicle environments small improvements in pointing task efficiency, even reducing pointing times by few milliseconds, will have substantial aggregate benefits on safety and overall user experience, especially for a driving system user.

Figure 11 shows that the destination-reverting models introduced here can make correct predictions with substantially higher confidence levels compared to benchmark techniques for the majority of the pointing trajectory. As intuitively expected, the NN method, which only uses current position information, is highly uncertain early in the pointing task and its uncertainty $\vartheta(t_k)$ decreases as $t_k \rightarrow t_T$. NN prediction certainty inevitably becomes higher than that of the MRD and ERV methods towards the end of the selection task as the pointing finger becomes very close to the destination (for $t_p \geq 90\%$ in Figure 11). Notably, prediction certainty for BA declines as t_p increases, reflecting the unreliability of the heading angle measure, particularly later in the gesture.

Whilst the above simulations do not constitute a complete experimental evaluative study, which is not the purpose of this paper, Figures 9, 10 and 11 clearly demonstrate the tangible performance gains provided by the proposed predictors. Testing was conducted on a relatively large number of typical

pointing trajectories. The improvements are particularly visible in the critical early portion of the pointing task. The models introduced here require relatively minimal training, with less than a fifth of the available trajectories used to learn model parameters via a maximum likelihood procedure. The results also indicate that the simple nearest neighbour model delivers competitive performance towards the end of the pointing gestures, i.e. as the pointing finger approaches the screen. It performs even better than more complex models such as MRD during this late period. Hence, an effective strategy might be to use MRD or ERV until the pointing finger is close to the interactive display surface, e.g. based on a predefined distance to the touchscreen (along the z-axis in Figures 2 and 8), after which NN predictions could be used. This would have the further benefit of very intuitive prediction in the region near the screen, potentially reducing user frustration at any incorrect predictions in 'easy' cases.

VI. Conclusion

This paper sets out a framework for probabilistic belief-based intent inference for pointing gesture based interactions in a 3D environment. By using a gesture-tracker and suitable destination-reverting linear models, in-vehicle intent-aware displays can predict the item a user intends to select remarkably early in the pointing gesture. This can significantly reduce the pointing time and the effort associated with interacting with the GUI. Thus, usability of interactive displays in the vehicle environment can be significantly improved by minimising the visual, cognitive and manual workload necessary to operate them, especially for the driver [1]–[3], [8]–[10], [22]. The proposed system could also be used to facilitate pointing at 3D or virtual displays where depth information is crucial, and where the interactive area is projected rather than displayed on a physical surface.

The two prediction models introduced can provide substantial performance enhancements compared to existing methods. Moreover, they are: 1) computationally efficient, with a Kalman filter-type implementation, 2) easy to train, requiring minimal training data, 3) probabilistic belief-based algorithms, and 4) adaptable to the application requirements and/or interface design via easily configured priors on the probability of selecting interface elements; such priors can be acquired from additional sensory data such as eye gaze [47]. Whilst mean reverting diffusion and equilibrium reverting velocity models are proposed in this paper, other linear destination-reverting models could be applied within the formulated framework.

This study serves to motivate further research into intent-aware interactive displays, especially with the increased interest in gesture based interactions in vehicles, e.g. [48] and, more generally, [21]. It calls for a complete experimental evaluative study that considers a large number of users, human factors, road categories and driving conditions. This will best quantify the gains of the proposed intent predictors and will serve to inform the choice of pointing facilitation technique(s) that make best use of the intent prediction results to improve the overall user experience.

Appendix A
Integration and Moments of a Linear Model

Both MRD and ERV models can be represented in a continuous-time setting by the linear, time-invariant stochastic differential equation

$$d\mathbf{s}_{I,t} = \mathbf{A}\left(\mathbf{a}_I - \mathbf{s}_{I,t}\right)dt + \boldsymbol{\sigma} d\mathbf{w}_t, \tag{32}$$

where $\mathbf{s}_{I,t} \in \mathbb{R}^{n\times 1}$ is the latent system state, $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{a}_I \in \mathbb{R}^{n\times 1}$ and $\boldsymbol{\sigma} \in \mathbb{R}^{n\times n}$ are (constant) parameters of the system, and $\mathbf{w}_t \in \mathbb{R}^{n\times 1}$ is a standard Weiner process. Let $f(\mathbf{s}_{I,t}, t) = e^{\mathbf{A}t}\mathbf{s}_{I,t}$, then, using Itô's lemma, we have

$$df(\mathbf{s}_{I,t}, t) = \mathbf{A}\mathbf{a}_I e^{\mathbf{A}t}dt + \boldsymbol{\sigma} e^{\mathbf{A}t} d\mathbf{w}_t \tag{33}$$

Integrating (33) over $\mathcal{T} = [t_1, t_2]$ including the initial value $e^{\mathbf{A}t_1}\mathbf{s}_{I,t_1}$ leads to

$$e^{\mathbf{A}t_2}\mathbf{s}_{I,t_2} = e^{\mathbf{A}t_1}\mathbf{s}_{I,t_1} + \left[e^{\mathbf{A}t_2} - e^{\mathbf{A}t_1}\right]\mathbf{a}_I + \int_{t_1}^{t_2} e^{\mathbf{A}v}\boldsymbol{\sigma} d\mathbf{w}_v. \tag{34}$$

Hence,

$$\mathbf{s}_{I,t_2} = e^{-\mathbf{A}\tau}\mathbf{s}_{I,t_1} + \left[\mathbf{I}_n - e^{-\mathbf{A}\tau}\right]\mathbf{a}_I + \int_{t_1}^{t_2} e^{\mathbf{A}(v-t_2)}\boldsymbol{\sigma} d\mathbf{w}_v \tag{35}$$

such that $\tau = t_2 - t_1$ and $\mathbf{I}_n$ is a $n \times n$ identity matrix. Noting that $\mathbb{E}\left[d\mathbf{w}_v\right] = 0$, it can be easily seen that

$$\mathbb{E}\left[\mathbf{s}_{I,t_2}|\mathbf{s}_{I,t_1}\right] = e^{-\mathbf{A}\tau}\mathbf{s}_{I,t_1} + \left[\mathbf{I}_n - e^{-\mathbf{A}\tau}\right]\mathbf{a}_I. \tag{36}$$

The conditional covariance can be calculated using Itô's isometry such that

$$\begin{aligned}\mathrm{Cov}\left[\mathbf{s}_{I,t_2}|\mathbf{s}_{I,t_1}\right] &= \mathbb{E}\left[\left(\int_{t_1}^{t_2} e^{\mathbf{A}(v-t_2)}\boldsymbol{\sigma} d\mathbf{w}_v\right)^2 \mid \mathbf{s}_{I,t_1}\right] \\ &= \int_{t_1}^{t_2} e^{\mathbf{A}(v-t_2)}\boldsymbol{\sigma}\boldsymbol{\sigma}' e^{\mathbf{A}'(v-t_2)}\boldsymbol{d}v \end{aligned} \tag{37}$$

which can be simplified based on the structure of $\mathbf{A}$ and $\boldsymbol{\sigma}$ for the MRD and ERV models. Two distinct simplified derivations of (37) are given in [43] and [49].

Acknowledgment

The authors would like to thank Jaguar Land Rover for funding this research under the CAPE agreement and facilitating the data collection.

References

[1] 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.

[2] C. Harvey and N. A. Stanton, *Usability evaluation for in-vehicle systems*. CRC Press, 2013.

[3] M. J. Pitts, G. Burnett, L. Skrypchuk, T. Wellings, A. Attridge, and M. A. Williams, "Visual–haptic feedback interaction in automotive touchscreens," *Displays*, vol. 33, no. 1, pp. 7–16, 2012.

[4] F.-G. Wu, H. Lin, and M. You, "Direct-touch vs. mouse input for navigation modes of the web map," *Displays*, vol. 32, no. 5, pp. 261–267, 2011.

[5] Volvo Cars, "Volvo car group unveils concept estate at geneva motor show (27th February 2014)." accessed on: 14 October 2014 from https://www.media.volvocars.com/global/en-gb/media/pressreleases/139220/volvo-car-group-to-unveil-concept-estate-at-geneva-motor-show.

[6] A. Sears, C. Plaisant, and B. Shneiderman, "A new era for touchscreen applications: High precision, dragging icons, and refined feedback," *Advances in Human-Computer Interaction*, vol. 3, 1991.

[7] M. G. Jæger, M. B. Skov, N. G. Thomassen *et al.*, "You can touch, but you can't look: interacting with in-vehicle systems," in *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, 2008, pp. 1139–1148.

[8] R. Swette, K. R. May, T. M. Gable, and B. N. Walker, "Comparing three novel multimodal touch interfaces for infotainment menus," in *Proceedings of the 5th AutomotiveUI*. ACM, 2013, pp. 100–107.

[9] "Visual-manual NHTSA driver distraction guidelines for in-vehicle electronic devices," Federal Register, Vo.77, No.37, Department of Transportation, NHTSA, 2012.

[10] P. Salmon, M. Lenné, T. Triggs, N. Goode, M. Cornelissen, and V. Demczuk, "The effects of motion on in-vehicle touch screen system operation: A battle management system case study," *Transportation Research: Traffic Psychology and Behaviour*, vol. 14, no. 6, pp. 494–503, 2011.

[11] N. Goode, M. G. Lenné, and P. Salmon, "The impact of on-road motion on BMS touch screen device operation," *Ergonomics*, vol. 55, no. 9, pp. 986–996, 2012.

[12] H. Kim and H. Song, "Evaluation of the safety and usability of touch gestures in operating in-vehicle information systems with visual occlusion," *Applied ergonomics*, vol. 45, no. 3, pp. 789–798, 2014.

[13] B. I. Ahmad, J. K. Murphy, P. M. Langdon, and S. J. Godsill, "Filtering perturbed in-vehicle pointing gesture trajectories: Improving the reliability of intent inference," in *Proc. of IEEE International Workshop on Machine Learning for Signal Processing (MLSP '14)*, 2014.

[14] V. L. Neale, T. A. Dingus, S. G. Klauer, J. Sudweeks, and M. Goodman, "An overview of the 100-car naturalistic study and findings," *National Highway Traffic Safety Administration, Paper*, no. 05-0400, 2005.

[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] Y. Liang and J. D. Lee, "Combining cognitive and visual distraction: Less than the sum of its parts," *Accident Analysis & Prevention*, vol. 42, no. 3, pp. 881–890, 2010.

[17] M. Falkenstein, J. Hohnsbein, J. Hoormann, and L. Blanke, "Effects of crossmodal divided attention on late ERP components. II. error processing in choice reaction tasks," *Electroencephalography and Clinical Neurophysiology*, vol. 78, no. 6, pp. 447–455, 1991.

[18] F. I. Craik, M. Naveh-Benjamin, G. Ishaik, and N. D. Anderson, "Divided attention during encoding and retrieval: Differential control effects?" *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 26, no. 6, p. 1744, 2000.

[19] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with Microsoft Kinect sensor: A review," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1318–1334, 2013.

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

[21] L. Garber, "Gestural technology: Moving interfaces in a new direction [technology news]," *Computer, IEEE*, vol. 46, no. 10, pp. 22–25, 2013.

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

[23] A. Murata, "Improvement of pointing time by predicting targets in pointing with a PC mouse," *IJHCI*, vol. 10, no. 1, pp. 23–32, 1998.

[24] D. Lane, S. Peres, A. Sándor, and H. Napier, "A process for anticipating and executing icon selection in graphical user interfaces," *IJHCI*, vol. 19, no. 2, pp. 241–252, 2005.

[25] J. O. Wobbrock, J. Fogarty, S. Liu, S. Kimuro, and S. Harada, "The angle mouse: target-agnostic dynamic gain adjustment based on angular deviation," in *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, 2009, pp. 1401–1410.

[26] 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*. ACM, 2005, pp. 133–141.

[27] 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.

[28] 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*, 2012, pp. 1–10.

[29] B. I. Ahmad, P. M. Langdon, P. Bunch, and S. J. Godsill, "Probabilistic intentionality prediction for target selection based on partial cursor tracks," *in Proc. of the 8th International Conference on Universal Access in Human-Computer Interaction (UAHCI 2014). Lecture Notes in Computer Science*, vol. 8515, pp. 427–438, 2014.

[30] P. M. Fitts and J. R. Peterson, "Information capacity of discrete motor responses." *Journal of Experimental Psychology*, vol. 67, no. 2, p. 103, 1964.

[31] I. S. MacKenzie and P. Isokoski, "Fitts' throughput and the speed-accuracy tradeoff," in *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, 2008, pp. 1633–1636.

[32] P. A. Hancock and K. M. Newell, "The movement speed-accuracy relationship in space-time," in *Motor Behavior*. Springer, 1985, pp. 153–188.

[33] T. Yamabe and K. Takahashi, "Experiments in mobile user interface adaptation for walking users," in *International Conference on Intelligent Pervasive Computing*. IEEE, 2007, pp. 280–284.

[34] A. Bragdon, E. Nelson, Y. Li, and K. Hinckley, "Experimental analysis of touch-screen gesture designs in mobile environments," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2011, pp. 403–412.

[35] M. Goel, L. Findlater, and J. Wobbrock, "Walktype: using accelerometer data to accommodate situational impairments in mobile touch screen text entry," in *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2012, pp. 2687–2696.

[36] A. Oulasvirta, A. Reichel, W. Li, Y. Zhang, M. Bachynskyi, K. Vertanen, and P. O. Kristensson, "Improving two-thumb text entry on touchscreen devices," in *Proc.of the SIGCHI Conference on Human Factors in Computing Systems*, 2013, pp. 2765–2774.

[37] B. I. Ahmad, P. M. Langdon, S. J. Godsill, R. Hardy, E. Dias, and L. Skrypchuk, "Interactive displays in vehicles: Improving usability with a pointing gesture tracker and Bayesian intent predictors," in *Proc. of International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 14)*, 2014, pp. 1–8.

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

[39] 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.

[40] 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 '13)*, 2013, pp. 661–668.

[41] S. Bateman, R. L. Mandryk, C. Gutwin, and R. Xiao, "Analysis and comparison of target assistance techniques for relative ray-cast pointing," *International Journal of Human-Computer Studies*, 2013.

[42] 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.

[43] A. Meucci, "Review of statistical arbitrage, cointegration, and multivariate Ornstein-Uhlenbeck," *SSRN preprint 1404905*, 2009.

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

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

[46] B. I. Ahmad, J. K. Murphy, P. M. Langdon, and S. J. Godsill, "Bayesian target prediction from partial finger tracks: Aiding interactive displays in vehicles," in *Proc. of the 17th International Conference on Information Fusion (FUSION '14)*, 2014, pp. 1–7.

[47] C. Topal, S. Gunal, O. Koçdeviren, A. Dogan, and O. N. Gerek, "A low-computational approach on gaze estimation with eye touch system," *IEEE Transactions on Cybernetics*, vol. 44, no. 2, pp. 228–239, 2014.

[48] E. Ohn-Bar and M. M. Trivedi, "Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1 – 10, 2014.

[49] J. Murphy, "Hidden states, hidden structures: Bayesian learning in time series models," Ph.D. dissertation, University of Cambridge, 2014.