LETTER **Communicated by Ron Meir**

The Discriminative Kalman Filter for Bayesian Filtering with Nonlinear and Nongaussian Observation Models

Michael C. Burkhart
michael_burkhart@alumni.brown.edu
Division of Applied Mathematics, Brown University, Providence, RI 02912, U.S.A.

David M. Brandman
david_brandman@alumni.brown.edu
Department of Neuroscience, Brown University, Providence, RI 02912, U.S.A., and Department of Surgery (Neurosurgery), Dalhousie University, Halifax, NS, B3H 4R2, Canada

Brian Franco
brfranco34@gmail.com
Center for Neurotechnology and Neurorecovery, Neurology, Massachusetts General Hospital, Boston, MA 02114, U.S.A.

Leigh R. Hochberg
leigh_hochberg@brown.edu
Center for Neurotechnology and Neurorecovery, Neurology, Massachusetts General Hospital, Boston, MA 02114, U.S.A.; School of Engineering and Carney Institute for Brain Science, Brown University, Providence, RI 02912, U.S.A.; Neurology, Harvard Medical School, Boston, MA 02115, U.S.A.; and VA RR&D Center for Neurorestoration and Neurotechnology, Providence Veterans Affairs Medical Center, Providence, RI 02908, U.S.A.

Matthew T. Harrison
matthew_harrison@brown.edu
Division of Applied Mathematics, Brown University, Providence, RI 02912, U.S.A.

The Kalman filter provides a simple and efficient algorithm to compute the posterior distribution for state-space models where both the latent state and measurement models are linear and gaussian. Extensions to the Kalman filter, including the extended and unscented Kalman filters, incorporate linearizations for models where the observation model $p(\textbf{observation}|\textbf{state})$ is nonlinear. We argue that in many cases, a model for $p(\textbf{state}|\textbf{observation})$ proves both easier to learn and more accurate for latent state estimation.

B. Franco is now at NeuroPace, Mountain View, CA, U.S.A.

Neural Computation **32, 969–1017** (2020)
https://doi.org/10.1162/neco_a_01275

Approximating p(state|observation) as gaussian leads to a new filtering algorithm, the discriminative Kalman filter (DKF), which can perform well even when p(observation|state) is highly nonlinear and/or nongaussian. The approximation, motivated by the Bernstein–von Mises theorem, improves as the dimensionality of the observations increases. The DKF has computational complexity similar to the Kalman filter, allowing it in some cases to perform much faster than particle filters with similar precision, while better accounting for nonlinear and nongaussian observation models than Kalman-based extensions.

When the observation model must be learned from training data prior to filtering, off-the-shelf nonlinear and nonparametric regression techniques can provide a gaussian model for p(observation|state) that cleanly integrates with the DKF. As part of the BrainGate2 clinical trial, we successfully implemented gaussian process regression with the DKF framework in a brain-computer interface to provide real-time, closed-loop cursor control to a person with a complete spinal cord injury. In this letter, we explore the theory underlying the DKF, exhibit some illustrative examples, and outline potential extensions.

1 Introduction

Consider a state-space model for $Z_{1:T} := Z_1, \ldots, Z_T$ (latent states) and $X_{1:T} := X_1, \ldots, X_T$ (observations) represented as a Bayesian network:

$$\begin{array}{ccccccccc} Z_1 & \longrightarrow & \cdots & \longrightarrow & Z_{t-1} & \longrightarrow & Z_t & \longrightarrow & \cdots \longrightarrow & Z_T \\ \downarrow & & & & \downarrow & & \downarrow & & & \downarrow \\ X_1 & & & & X_{t-1} & & X_t & & & X_T \end{array}. \tag{1.1}$$

The conditional density of Z_t given $X_{1:t}$ can be expressed recursively using the Chapman-Kolmogorov equation and Bayes' rule (see Chen, 2003, for further details):

$$p(z_t|x_{1:t-1}) = \int p(z_t|z_{t-1})p(z_{t-1}|x_{1:t-1})\,dz_{t-1}, \tag{1.2a}$$

$$p(z_t|x_{1:t}) = \frac{p(x_t|z_t)p(z_t|x_{1:t-1})}{\int p(x_t|z_t)p(z_t|x_{1:t-1})\,dz_t} = \frac{p(x_t|z_t)p(z_t|x_{1:t-1})}{p(x_t|x_{1:t-1})}, \tag{1.2b}$$

where $p(z_0|x_{1:0}) = p(z_0)$ and the conditional densities $p(z_t|z_{t-1})$ and $p(x_t|z_t)$ are either specified a priori or learned from training data prior to filtering. Computing or approximating equation 1.2 is often called *Bayesian filtering*. Bayesian filtering arises in a large number of applications, including global positioning systems, target tracking, aircraft and spacecraft guidance, weather forecasting, computer vision, digital communications, and

brain-computer interfaces (Chen, 2003; Hall, 1966; Battin & Levine, 1970; Grewal & Andrews, 2010; Buehner, McTaggart-Cowan, & Heilliette, 2017; Brown & Hwang, 2012; Schmidt, Weinberg, & Lukesh, 1970; Brandman, Cash, & Hochberg, 2017).

Exact solutions to equation 1.2 are available only in special cases, such as the Kalman filter (Kalman, 1960; Kalman & Bucy, 1961). The Kalman filter models the conditional densities $p(z_t|z_{t-1})$ and $p(x_t|z_t)$ as linear and gaussian so that the posterior distribution $p(z_t|x_{1:t})$ is also gaussian and quickly computable. Beneš (1981) and Daum (1984, 1986) broadened the class of models for which the integrals in equation 1.2 are analytically tractable, but many model specifications still fall outside this class. When the latent state space is finite, the integrals in equation 1.2 become sums that can be calculated exactly using a grid-based filter (Elliott, 1994; Arulampalam, Maskell, Gordon, & Clapp, 2002). For more general models, there are many techniques for approximate Bayesian filtering see (Chen, 2003 for a review).

In some applications, parts of the underlying model are first learned from supervised training data consisting of (Z_t, X_t) pairs, and then the learned model is used for filtering on new (X_t) data. For instance, (Z_t, X_t) pairs might be used to learn $p(x_t|z_t)$ with nonparametric conditional density estimation, and then the learned $p(x_t|z_t)$, say, $\hat{p}(x_t|z_t)$, is substituted into whatever algorithm is used to approximate Bayes' rule in equation 1.2b. This motivates the search for combinations of approximation algorithms and learning methods that work well together. It also opens the door to novel approximation algorithms that would not traditionally be considered for a known model but become practical when the model can be learned. For instance, from (Z_t, X_t) pairs, we can choose to learn $p(x_t|z_t)$ or $p(z_t|x_t)$ and incorporate either into the approximation algorithm, whereas traditional approximation algorithms assume that only $p(x_t|z_t)$ is available.

In this letter, we explore the idea of using a novel approximation algorithm that pairs well with learning and demonstrate its use in an intracortical brain-computer interface (iBCI) for a human volunteer with tetraplegia as part of the ongoing BrainGate2 clinical trial. Our approach focuses on the approximation of Bayes' rule in equation 1.2b, making use of the fact that $p(x_t|z_t)$ can be replaced with $p(z_t|x_t)/p(z_t)$ throughout. (The $p(x_t)$ term cancels.) This strategy combines well with various gaussian assumptions that are often employed in approximate Bayesian filtering, resulting in what we call the *discriminative Kalman filter* (DKF). The DKF retains much of the computational simplicity of the classical Kalman filter but allows for arbitrary observation models. Some of our clinical research using the DKF has already been published (Brandman, Burkhart et al., 2018; Brandman, Hosman et al., 2018), and theoretical aspects of the DKF are further explored in the first author's dissertation (Burkhart, 2019).

2 The Discriminative Kalman Filter

In section 2.1, we derive the DKF approximation for a class of models that generalizes the Kalman filter by allowing for arbitrary observation models. We discuss approximation accuracy in section 2.2 and introduce a modified algorithm that can be more robust to model misspecification in section 2.3. In section 2.4, we compare the DKF formalism to a variety of existing approaches that generalize the Kalman filter, and in section 2.5, we discuss using the DKF approximation in models with nonlinear or nongaussian state dynamics.

We now introduce some notation and conventions. We let the latent states Z_t take values in $\mathbb{R}^{d\times 1}$ and the observations X_t take values in an abstract space $\mathcal{X}$. In all of our examples, $\mathcal{X} \subseteq \mathbb{R}^{n\times 1}$, but this is not necessary. We use $\eta_d(z; \mu, \Sigma)$ to denote the d-dimensional multivariate gaussian distribution with mean vector $\mu \in \mathbb{R}^{d\times 1}$ and covariance matrix $\Sigma \in \mathbb{S}_d$ evaluated at $z \in \mathbb{R}^{d\times 1}$, where $\mathbb{S}_d$ denotes the set of $d\times d$ positive-definite (symmetric) matrices. We let $A^\intercal$ refer to the transpose of a matrix A and use $\mathbb{E}$ and $\mathbb{V}$ for expected value and variance/covariance, respectively.

2.1 Filter Derivation. For the basic derivation, we assume that the latent states form a stationary, mean zero, gaussian, vector autoregressive model of order 1. Namely, for $A \in \mathbb{R}^{d\times d}$ and $S, \Gamma \in \mathbb{S}_d$,

$$p(z_0) = \eta_d(z_0; 0, S), \tag{2.1a}$$

$$p(z_t|z_{t-1}) = \eta_d(z_t; Az_{t-1}, \Gamma), \tag{2.1b}$$

for $t = 1, 2, \ldots$, where $S = ASA^\intercal + \Gamma$ so that the process is stationary. Note that equation 2.1 matches the latent state model for the stationary Kalman filter. (The assumption of zero mean is easily generalized, but it is usually more convenient to center the Z_t process by subtracting the common mean.)

The observation model $p(x_t|z_t)$ is assumed to not vary with t, so that the joint (Z_t, X_t) process is stationary but otherwise arbitrary. The observation model can be nongaussian, multimodal, or discrete, for example. For instance, in neural decoding for BCI applications, the observations are often vectors of counts of neural spiking events (binned action potentials), which might be restricted to small integers or even be binary-valued.

The DKF is based on a gaussian approximation for $p(z_t|x_t)$, namely,

$$p(z_t|x_t) \approx \eta_d(z_t; f(x_t), Q(x_t)), \tag{2.2}$$

where $f : \mathcal{X} \to \mathbb{R}^d$ and $Q : \mathcal{X} \to \mathbb{S}_d$. Note that equation 2.2 is not an approximation of the observation model, but rather of the conditional density of the latent state given the observation at a single time step. In section 2.4,

we compare this to other approaches that use gaussian approximations for Bayesian filtering. When the dimensionality of the observation space ($\mathcal{X}$) is large relative to the dimensionality of the state space ($\mathbb{R}^d$), the Bernstein–von Mises theorem states that f and Q exist such that this approximation will be accurate, requiring only mild regularity conditions on the observation model $p(x_t|z_t)$ (see section 2.2 in van der Vaart, 1998). Furthermore, we can take f and Q to be the conditional mean and covariance of Z_t given X_t, namely,

$$f(x) = \mathbb{E}(Z_t|X_t = x), \quad Q(x) = \mathbb{V}(Z_t|X_t = x), \tag{2.3}$$

which is the approach taken in this letter, although other choices are certainly possible, such as $f(x_t) = \arg\max_{z_t} p(z_t|x_t)$ or $f(x_t) = \arg\max_{z_t} p(x_t|z_t)$, the latter of which is most commonly used in statements of the Bernstein–von mises Theorem.

To make use of equation 2.2 for approximating equation 1.2, we first rewrite equation 1.2b in terms of $p(z_t|x_t)$ as

$$\begin{aligned} p(z_t|x_{1:t}) &= \frac{p(x_t)}{p(x_t|x_{1:t-1})} \frac{p(z_t|x_t)}{p(z_t)} p(z_t|x_{1:t-1}), \\ &= \frac{p(x_t)}{p(x_t|x_{1:t-1})} \frac{p(z_t|x_t)}{p(z_t)} \int p(z_t|z_{t-1}) p(z_{t-1}|x_{1:t-1})\, dz_{t-1}, \end{aligned} \tag{2.4}$$

where the second line follows from the Chapman–Kolmogorov equation (see equation 1.2a). We then substitute the latent state model, equation 2.1, and the DKF approximation, equation 2.2, into equation 2.4. We absorb terms not depending on z_t into a normalizing constant κ to obtain

$$\begin{aligned} &p(z_t|x_{1:t}) \\ &\approx \kappa(x_{1:t}) \frac{\eta_d(z_t; f(x_t), Q(x_t))}{\eta_d(z_t; 0, S)} \int \eta_d(z_t; Az_{t-1}, \Gamma) p(z_{t-1}|x_{1:t-1})\, dz_{t-1}. \end{aligned} \tag{2.5}$$

If $p(z_{t-1}|x_{1:t-1})$ is approximately gaussian, which it is for the base case of $t = 1$ from equation 2.1a (defining $p(z_0|x_{1:0}) = p(z_0)$), then all of the terms on the right side of equation 2.5 are approximately gaussian. If these approximations are exact and the analytic expression for covariance is valid (specifically if Σ_t in equation 2.7 is positive definite), we find that the right side of equation 2.5 is again gaussian, giving a gaussian approximation for $p(z_t|x_{1:t})$. We rely on the fact that dividing two gaussian pdfs yields an exponentiated quadratic form that will itself be gaussian if the associated covariance matrix is positive definite (and that the product of two gaussian pdfs is gaussian, without any additional assumptions). See the proof of lemma 1 in appendix B for a full derivation and further details.

Let

$$p(z_t|x_{1:t}) \approx \eta_d(z_t; \mu_t(x_{1:t}), \Sigma_t(x_{1:t})) \tag{2.6}$$

be the gaussian approximation of $p(z_t|x_{1:t})$ obtained from successively applying the approximation in equation 2.5. Defining $\mu_0 = 0$ and $\Sigma_0 = S$, we can sequentially compute $\mu_t = \mu_t(x_{1:t}) \in \mathbb{R}^{d\times 1}$ and $\Sigma_t = \Sigma_t(x_{1:t}) \in \mathbb{S}_d$ via

$$\begin{aligned} \nu_t &= A\mu_{t-1}, \\ M_t &= A\Sigma_{t-1}A^\intercal + \Gamma, \\ \Sigma_t &= \left(M_t^{-1} + Q(x_t)^{-1} - S^{-1}\right)^{-1}, \\ \mu_t &= \Sigma_t\left(M_t^{-1}\nu_t + Q(x_t)^{-1}f(x_t)\right). \end{aligned} \tag{2.7}$$

The first two steps incorporate the exact state dynamics in equation 2.1b and the final two steps incorporate the observation information using the DKF approximation in equation 2.2. The function Q needs to be defined so that Σ_t exists and is a proper covariance matrix. A sufficient condition that is easy to enforce in practice is $Q(\cdot)^{-1} - S^{-1} \in \mathbb{S}_d$ (see section A.3 in appendix A).

Equation 2.7 encapsulates the DKF. (For pseudocode, see algorithm 1.) Once $f(x_t)$ and $Q(x_t)$ have been evaluated, there is no remaining dependence on n and a single iteration of the algorithm takes $O(d^3)$ operations, which is at least as fast as the Kalman filter (when $d < n$). The power of the DKF, along with potential computational difficulties, comes from evaluating f and Q. If f is linear and Q is constant, the DKF and the Kalman filter are equivalent (see section 4.1). More general f and Q allow the filter to depend nonlinearly on the observations, improving performance in many cases. If f and Q can be quickly evaluated and the dimension d of Z_t is not too large, then the DKF is fast enough for use in real-time applications, such as the BCI decoding example below.

2.2 Approximation Accuracy. Let the observation space be $\mathcal{X} = B^n$ for some set B. As n grows, the Bernstein–von Mises (BvM) theorem guarantees under mild assumptions that the conditional distribution of $Z_t|X_t$ is asymptotically normal in total variation distance and concentrates at Z_t (van der Vaart, 1998). This asymptotic normality result provides the main rationale for our key approximation expressed in equation 2.2. The BvM theorem is usually stated in the context of Bayesian estimation. To apply it in our context, we equate Z_t with the parameter and X_t with the data, so that $p(z_t|x_t)$ becomes the posterior distribution of the parameter at a fixed time t. We then let the dimension n of x_t grow, meaning that we are observing growing amounts of data at a fixed time t associated with the parameter Z_t. Very loosely speaking, the BvM theorem tends to be applicable in

Algorithm 1: The DKF.

Data: observations $x_1, x_2, \ldots$; matrices $A \in \mathbb{R}^{d\times d}$ and $S, \Gamma \in \mathbb{S}_d$ such that

$z_0, z_1, \ldots$ are drawn from stationary process satisfying equation 2.1;

functions $f : \mathcal{X} \to \mathbb{R}^d$ and $Q : \mathcal{X} \to \mathbb{S}_d$ such that

$p(z_t|x_t) \approx \eta_d(z_t; f(x_t), Q(x_t))$ and $Q(\cdot)^{-1} - S^{-1} \in \mathbb{S}_d$, either derived analytically or approximated from data

Result: $\mu_t = \mu_t(x_{1:t}) \in \mathbb{R}^{d\times 1}$ and $\Sigma_t = \Sigma_t(x_{1:t}) \in \mathbb{S}_d$ to approximate the posterior distribution as $p(z_t|x_{1:t}) \approx \eta_d(z_t; \mu_t, \Sigma_t)$ for $t = 1, 2, \ldots$

Initialize $\mu_0 = 0$ and $\Sigma_0 = S$;

for $t \geq 1$ **do**

set $\nu_t = A\mu_{t-1}$ and $M_t = A\Sigma_{t-1}A^\intercal + \Gamma$;

set $\Sigma_t = \left(M_t^{-1} + Q(x_t)^{-1} - S^{-1}\right)^{-1}$ and $\mu_t = \Sigma_t\left(M_t^{-1}\nu_t + Q(x_t)^{-1}f(x_t)\right)$;

end

situations where X_t uniquely determines Z_t in the limit as $n \to \infty$ but does not uniquely determine Z_t for any finite n.

One concern is that equation 2.4 will amplify approximation errors. Along these lines, we prove the following result that holds whenever the BvM theorem is applicable for equation 2.2:

Theorem 1. *Under mild assumptions, the total variation distance between our approximation $\eta_d(z_t; \mu_t(x_{1:t}), \Sigma_t(x_{1:t}))$ and the exact filtering distribution $p(z_t|x_{1:t})$ converges in probability to zero for each t as $n \to \infty$.*

This result is stated formally and proven in appendix B. We interpret the theorem to mean that under most conditions, as the dimensionality of the observations increases, the approximation error of the DKF tends to zero.

The proof is elementary but involves several subtleties that arise because of the $p(z_t)$ term in the denominator of equation 2.4 corresponding to $\eta_d(z_t; 0, S)$. This term can amplify approximation errors in the tails of $p(z_t|x_t)$, which are not uniformly controlled by the asymptotic normality results in the BvM theorem. To remedy this, our proof also uses the concentration results in the BvM theorem to control pathological behaviors in the

tails. As an intermediate step, we prove that theorem 1 still holds when the $p(z_t)$ term is omitted from the denominator of equation 2.4 (see remark 3 in appendix B).

2.3 Robust DKF. Omitting the $p(z_t)$ from the denominator of equation 2.4 is also helpful for making the DKF robust to violations of the modeling assumptions and errors introduced when f and Q are learned from training data. Repeating the original derivation, but without $\eta_d(z_t; 0, S)$ in the denominator, gives the following filtering algorithm that we call the *robust DKF*. One can think of the robust DKF as a special case of the standard DKF where all eigenvalues of S^{-1} are so small that the effect of subtracting S^{-1} is negligible. This has the effect of placing an improper prior on Z_0. Defining $\mu_1(x_1) = f(x_1)$ and $\Sigma_1(x_1) = Q(x_1)$, we sequentially compute μ_t and Σ_t for $t \geq 2$ via

$$\begin{aligned} \nu_t &= A\mu_{t-1}, \\ M_t &= A\Sigma_{t-1}A^\intercal + \Gamma, \\ \Sigma_t &= \left(M_t^{-1} + Q(x_t)^{-1}\right)^{-1}, \\ \mu_t &= \Sigma_t\left(M_t^{-1}\nu_t + Q(x_t)^{-1} f(x_t)\right). \end{aligned} \tag{2.8}$$

(Note that we initialize at $t = 1$ and not $t = 0$ in the robust DKF.) Justification for the robust DKF comes from remark 3 in appendix B showing that the robust DKF accurately approximates the true $p(z_t|x_{1:t})$ in total variation distance for each t as n increases. We sometimes find that the robust DKF outperforms the DKF on real-data examples, but not on simulated examples that closely match the DKF assumptions. (For pseudocode, see algorithm 2.)

2.4 Other Gaussian Approximations. The DKF enforces a gaussian form for the filtering distribution $p(z_t|x_{1:t})$, a common strategy for approximate Bayesian filtering owing to the analytic and representational tractability of gaussians. In this section, we describe several other methods that use gaussian approximations, focusing on the case of linear, gaussian state dynamics. For this type of state dynamics, the transition from time $t-1$ to time t is usually separated into two distinct steps when using gaussian approximations. Beginning with

$$p(z_{t-1}|x_{1:t-1}) \approx \eta_d(z_{t-1}; \mu_{t-1}, \Sigma_{t-1}),$$

the first step uses the exact state dynamics, equation 2.1b, to create a gaussian approximation for $p(z_t|x_{1:t-1})$, namely,

$$p(z_t|x_{1:t-1}) \approx \eta_d(z_t; \nu_t, M_t), \tag{2.9}$$

Algorithm 2: The Robust DKF.

Data: observations $x_1, x_2, \ldots$; matrices $A \in \mathbb{R}^{d\times d}$ and $S, \Gamma \in \mathbb{S}_d$ such that

$z_0, z_1, \ldots$ are drawn from stationary process satisfying equation 2.1;

functions $f : \mathcal{X} \to \mathbb{R}^d$ and $Q : \mathcal{X} \to \mathbb{S}_d$ such that

$p(z_t|x_t) \approx \eta_d(z_t; f(x_t), Q(x_t))$

Result: $\mu_t = \mu_t(x_{1:t}) \in \mathbb{R}^{d\times 1}$ and $\Sigma_t = \Sigma_t(x_{1:t}) \in \mathbb{S}_d$ to approximate the

posterior distribution as $p(z_t|x_{1:t}) \approx \eta_d(z_t; \mu_t, \Sigma_t)$ for $t = 1, 2, \ldots$

Initialize $\mu_1(x_1) = f(x_1)$ and $\Sigma_1(x_1) = Q(x_1)$;

for $t \geq 2$ **do**

set $\nu_t = A\mu_{t-1}$ and $M_t = A\Sigma_{t-1}A^\intercal + \Gamma$;

set $\Sigma_t = \left(M_t^{-1} + Q(x_t)^{-1}\right)^{-1}$ and $\mu_t = \Sigma_t\left(M_t^{-1}\nu_t + Q(x_t)^{-1}f(x_t)\right)$;

end

where $\nu_t = A\mu_{t-1}$ and $M_t = A\Sigma_{t-1}A^\intercal + \Gamma$, as in equations 2.7 and 2.8. Most gaussian methods would proceed similarly for the first step under these state dynamics. Differences between methods appear for nonlinear or non-gaussian state dynamics (see section 2.5).

The second step attempts to incorporate the observation information x_t via Bayes' rule:

$$p(z_t|x_{1:t}) = \frac{p(x_t|z_t)p(z_t|x_{1:t-1})}{\int p(x_t|z_t)p(z_t|x_{1:t-1})\,dz_t}.$$

Beginning with the gaussian approximation from step 1 (equation 2.9) and enforcing the final approximation,

$$p(z_t|x_{1:t}) \approx \eta_d(z_t; \mu_t, \Sigma_t),$$

the problem reduces to finding μ_t and Σ_t so that

$$\eta_d(z_t; \mu_t, \Sigma_t) \approx \frac{p(x_t|z_t)\eta_d(z_t; \nu_t, M_t)}{\int p(x_t|z_t)\eta_d(z_t; \nu_t, M_t)\,dz_t} = q_t(z_t), \tag{2.10}$$

where q_t is defined by this equation.

There are many strategies in the literature for choosing μ_t and Σ_t in equation 2.10. The terminology is not standardized, but we will attempt to describe some prominent classes of strategies.

2.4.1 Gaussian Assumed Density Filter. The gaussian assumed density filter (G-ADF) usually refers to choosing μ_t and Σ_t to be the mean vector and covariance matrix of the density q_t in equation 2.10 (Kushner, 1967; Ito, 2000; Ito & Xiong, 2000; Minka, 2001a). Moment matching, in this case, minimizes the relative entropy $D(q_t \| \eta_d(\cdot; \mu_t, \Sigma_t))$. The G-ADF directly seeks a gaussian approximation to the full posterior $p(z_t|x_{1:t})$, whereas the DKF derives a gaussian approximation to the full posterior from a gaussian approximation of $p(z_t|x_t)$. While the G-ADF approach tends to prove quite accurate, it is practical only if the mean and covariance of q_t are available. In particular, we must be able to efficiently compute or easily approximate the integrals,

$$\begin{aligned} a &= \int p(x_t|z_t)\eta_d(z_t; v_t, M_t)\, dz_t, \\ b &= \int z_t\, p(x_t|z_t)\eta_d(z_t; v_t, M_t)\, dz_t, \\ c &= \int z_t z_t^\intercal\, p(x_t|z_t)\eta_d(z_t; v_t, M_t)\, dz_t, \end{aligned} \tag{2.11}$$

to obtain $\mu_t = b/a$ and $\Sigma_t = c/a - \mu_t \mu_t^\intercal$. There also exist extensions of the G-ADF. For instance, expectation propagation uses iterative refinement of estimates to improve on assumed density filtering (Minka, 2001a, 2001b). It may be possible to similarly improve the DKF, but iterating over the history of observations is typically not practical in an online setting, and we do not explore that approach here.

In cases where the DKF is derived from a known model, as opposed to being learned from training data, computing $f(x_t)$ and $Q(x_t)$ requires the computation of very similar integrals to those needed for the G-ADF, the difference being that v_t and M_t are replaced by 0 and S, respectively, throughout equation 2.11 (and then $f(x_t) = b/a$ and $Q(x_t) = c/a - f(x_t)f(x_t)^\intercal$). For this reason, in models where the G-ADF can be easily used, there would seem to be no reason to use the DKF. The main difference is that the DKF can be easily learned from training data, whereas the G-ADF cannot, since the latter is based on the conditional mean and variance of $Z_t|X_t$ derived under a different marginal distribution for Z_t at each time step, namely, $\eta_d(z_t; v_t, M_t)$. The example in section 4.2 illustrates a model where both the DKF and G-ADF can be analytically computed; there is little difference in performance. The example in section 4.3 illustrates a somewhat contrived model where the DKF can be easily computed, but it seems the G-ADF cannot.

2.4.2 Laplace Approximation. The Laplace approximation uses a Taylor approximation at the maximum to coerce the numerator in equation 2.10 into a gaussian form as a function of z_t (Butler, 2007; Koyama, Pérez-Bolde, Shalizi, & Kass, 2010; Quang, Musso, & Le Gland, 2015). Defining

$$g_t(z_t) = \log\big(p(x_t|z_t)\eta_d(z_t; \nu_t, M_t)\big) \quad \text{and} \quad z_t^* = \arg\max_{z_t} g_t(z_t),$$

a second-order Taylor approximation of g_t at z_t^* is

$$g_t(z_t) \approx g_t(z_t^*) + \dot{g}_t(z_t^*)(z_t - z_t^*) + (z_t - z_t^*)^\intercal \ddot{g}_t(z_t^*)(z_t - z_t^*)/2,$$

where $\dot{g}_t(z)$ and $\ddot{g}_t(z)$ denote, respectively, the $d\times 1$ gradient vector and the $d\times d$ Hessian matrix of g_t evaluated at z. The second term vanishes since $\dot{g}_t$ is zero at the maximum, giving

$$\begin{aligned} q_t(z_t) &\propto \exp(g_t(z_t)) \\ &\approx \exp\big(g_t(z_t^*) + (z_t - z_t^*)^\intercal \ddot{g}_t(z_t^*)(z_t - z_t^*)/2\big) \\ &\propto \eta_d(z_t; z_t^*, -\ddot{g}_t(z_t^*)^{-1}). \end{aligned}$$

This motivates the choice of $\mu_t = z_t^*$ and $\Sigma_t = -\ddot{g}_t(z_t^*)$. Similar to the DKF, the Laplace approximation can be justified in the limit of increasing observation dimensionality using the BvM theorem. If z_t^* or the derivatives of g_t are not available in closed form, then the Laplace approximation can be slow owing to the need to solve an optimization problem at each time step. Laplace approximations are also criticized for being too local, in that the local curvature in the density at z_t^* dictates the variance chosen for a global approximation to the density.

2.4.3 Linearization Methods. Several methods, often called linearization methods, can be motivated by attempting to approximate the numerator of equation 2.10 as jointly gaussian in (z_t, x_t), namely,

$$p(x_t|z_t)\,\eta_d(z_t; \nu_t, M_t) \approx \eta_{d+n}\left(\begin{pmatrix} z_t \\ x_t \end{pmatrix}\middle|\begin{pmatrix} \nu_t \\ h_t \end{pmatrix}, \begin{pmatrix} M_t & C_t \\ C_t^\intercal & N_t \end{pmatrix}\right), \tag{2.12}$$

where the history of observations $x_{1:t}$ is allowed to influence the choice of $h_t \in \mathbb{R}^{n\times 1}$, $N_t \in \mathbb{S}_n$, and $C_t \in \mathbb{R}^{d\times n}$. Using this approximation allows equation 2.10 to be exactly integrated to obtain

$$\mu_t = \nu_t + C_t N_t^{-1}(x_t - h_t) \quad \text{and} \quad \Sigma_t = M_t - C_t N_t^{-1} C_t^\intercal. \tag{2.13}$$

Methods differ in how they choose h_t, N_t, and C_t.

Using $\eta_d(z_t; \nu_t, M_t)$ as the marginal density for Z_t, equation 2.12 can be rewritten as

$$p(x_t|z_t) \approx \eta_n(x_t; b_t + H_t z_t, \Lambda_t). \tag{2.14}$$

The implicit linearization in equation 2.12 is now explicit: $\mathbb{E}(X_t|Z_t = z_t)$ is approximated as the linear function $b_t + H_t z_t$. The relationship between the different parameters in equations 2.12 and 2.14 is $b_t = h_t - C_t^\intercal M_t^{-1} \nu_t$, $H_t = C_t^\intercal M_t^{-1}$, and $\Lambda_t = N_t - C_t^\intercal M^{-1} C_t$. Upon reparameterization,[1] equation 2.13 can be used for filtering with

$$\begin{aligned}\Sigma_t &= (M_t^{-1} + H_t^\intercal \Lambda_t^{-1} H_t)^{-1},\\ \mu_t &= \Sigma_t\big(M_t^{-1}\nu_t + H_t^\intercal \Lambda_t^{-1}(x_t - b_t)\big),\end{aligned}$$

which has a similar appearance to the corresponding DKF updates in equation 2.7.

Equation 2.14 underlies several gaussian approximations to Bayes' rule, including the approximations used in the extended Kalman filter (EKF), the unscented Kalman filter (UKF: Julier & Uhlmann, 1997; Wan & van der Merwe, 2000; van der Merwe, 2004), and the statistically linearized filter (SLF: Gelb, 1974; Särkkä, 2013). The EKF, for instance, begins with the functions

$$h(z) = \mathbb{E}(X_t|Z_t = z) \quad \text{and} \quad \Lambda(z) = \mathbb{V}(X_t|Z_t = z),$$

which are assumed known, and takes $H_t = \dot{h}(\nu_t)$, $b_t = h(\nu_t) - H_t \nu_t$, and $\Lambda_t = \Lambda(\nu_t)$, where $\dot{h}(z)$ is the $n \times d$ matrix of partial derivatives of h evaluated at z. These choices of b_t and H_t correspond to a first-order Taylor approximation of h at the point ν_t. Like the Laplace approximation, the EKF is often criticized for being too local because the gradient of h at a single point drives the approximation.

The unscented Kalman filter (UKF) employs the eponymous transform to propagate weighted, deterministically chosen points through a nonlinear transformation and recover estimates for h_t, N_t, and C_t from equation 2.13. The estimates for all three parameters prove exact for linear transformations of gaussians but inexact for general higher-order polynomials (Särkkä, 2013), so we consider this a linearization method. Variations on this approach, collectively called sigma-point filters (van der Merwe, 2004), include the central difference Kalman filter (CDKF: Ito & Xiong, 2000; Nørgaard, Poulsen, & Ravn, 2000), the Gauss-Hermite Kalman filter,

[1] With $h_t = b_t + H_t \nu_t$, $C_t = M_t H_t^\intercal$, and $N_t = \Lambda_t + H_t M_t H_t^\intercal$.

the quadrature Kalman filter (Ito, 2000; Ito & Xiong, 2000), and the cubature Kalman filter (Arasaratnam, Haykin, & Elliott, 2007; Arasaratnam & Haykin, 2009).

The SLF is a related but more global approximation for the same observation model. It selects b_t and H_t to minimize the difference between the true observation model $X_t = h(Z_t) + \epsilon_t$ and the linear approximation $X_t \approx a_t + B_t Z_t + \epsilon_t$, where Z_t is chosen from the current, approximate, predicted distribution. For instance, a_t and B_t can be chosen to minimize

$$\int \|h(z_t) - (a_t + B_t z_t)\|^2 \eta_d(z_t; \nu_t, M_t)\, dz_t,$$

where $\|\cdot\|$ is the usual Euclidean norm in $\mathbb{R}^n$. Defining $\bar{h}_t = \int h(z_t)\eta_d(z_t; \nu_t, M_t)\, dz_t$ and $\bar{H}_t = \int (h(z_t) - \bar{h}_t)(z_t - \nu_t)^\intercal \eta_d(z_t; \nu_t, M_t)\, dz_t$, the solution is $B_t = \bar{H}_t M_t^{-1}$ and $a_t = \bar{h}_t - B_t \nu_t$, again with $\Lambda_t = \Lambda$. Like the EKF, this version of the SLF is best suited for additive, gaussian noise models, but it further requires that the integrals defining $\bar{h}_t$ and $\bar{H}_t$ can be efficiently computed or easily approximated.

The UKF, the SLF, and many related techniques improve on some of the deficiencies of the EKF. Nevertheless, these methods tend to perform poorly when the conditional distribution of X_t given Z_t cannot be well approximated as gaussian. The examples in sections 4.2 and 4.3 illustrate models where linearization proves completely ineffectual, as $h(z) = \mathbb{E}(X_t | Z_t = z) = 0$ for all z in these examples, even though the G-ADF and the DKF work well.

2.5 Nonlinear State Dynamics. Filtering can be conceptually separated into two steps. The first step uses the state dynamics to transition from $Z_{t-1}|X_{1:t-1}$ to $Z_t|X_{1:t-1}$ via equation 1.2a, and the second step uses Bayes' rule to update $Z_t|X_{1:t-1}$ into $Z_t|X_{1:t}$ via equation 1.2b. In this letter, difficulties with the first step are removed by assuming linear, gaussian, state dynamics (see equation 2.1). There are, however, a variety of approximation methods for more complicated state dynamics, including methods that approximate $p(z_t|x_{1:t-1})$ as a gaussian. Any such gaussian method could be easily combined with the DKF approximation, which relates to Bayes' rule in the second step. In particular, given the approximation

$$p(z_t|x_{1:t-1}) \approx \eta_d(z_t; \nu_t, M_t),$$

we simply use these values of ν_t and M_t in the DKF algorithm (see equation 2.7) or the robust DKF algorithm see equation 2.8, instead of computing them in the first two lines of these algorithms. In this letter, we do not explore in depth this generalization to nonlinear state dynamics, although we do provide a proof of concept example in section 4.4.

There is a vast literature on more general approximation algorithms for Bayesian filtering (Särkkä, 2013; Chen, 2003). Monte Carlo integration (Metropolis & Ulam, 1949) can almost always be used. Such approaches are called sequential Monte Carlo or particle filtering and include sequential importance sampling and sequential importance resampling (Handschin & Mayne, 1969; Handschin, 1970; Gordon, Salmond, & Smith, 1993; Kitagawa, 1996; del Moral, 1996; Doucet, Godsill, & Andrieu, 2000; Cappé, Moulines, & Ryden, 2005; Cappé, Godsill, & Moulines, 2007). These methods apply to all classes of models but tend to be the most expensive to compute online and suffer from the curse of dimensionality (Daum & Huang, 2003). Alternate sampling strategies (see, e.g., Chen, 2003; Liu, 2008) can be used to improve filter performance, including acceptance-rejection sampling (Handschin & Mayne, 1969), stratified sampling (Douc & Cappé, 2005), hybrid MC (Choo & Fleet, 2001), and quasi-MC (Gerber & Chopin, 2015). There are also ensemble versions of the Kalman filter that are used to propagate the covariance matrix in high dimensions, including the ensemble Kalman filter (enKF: Evensen, 1994) and the ensemble transform Kalman filter (ETKF: Bishop, Etherton, & Majumdar, 2001; Majumdar, Bishop, Etherton, & Toth, 2002), along with versions that produce local, parallelizable approximations for covariance (Ott et al., 2004; Hunt, Kostelich, & Szunyogh, 2007).

It may be possible to usefully combine the DKF approximation with some of these more advanced filtering techniques. The key approximation in the DKF is

$$p(x_t|z_t) = p(x_t)\frac{p(z_t|x_t)}{p(z_t)} \approx \kappa(x_t)\frac{\eta_d(z_t;\, f(x_t), Q(x_t))}{\eta_d(z_t;\, 0, S)}. \tag{2.15}$$

This approximation could, in principle, be substituted for the likelihood $p(x_t|z_t)$ in any filtering algorithm, including particle filters, which incorporate the likelihood into the particle weights. The normalizing term $\kappa(x_t)$ from equation 2.15 will generally cancel, since the final posterior distribution $p(z_t|x_{1:t})$ is invariant to terms depending only on $x_{1:t}$. The advantage of equation 2.15 is that $f(\cdot)$, $Q(\cdot)$, and S might be easier to learn from data than the full conditional density $p(x_t|z_t)$. For complex state dynamics, it is worth noting that the denominator $\eta_d(z_t; 0, S)$ will no longer precisely correspond to $p(z_t)$ but will also be an approximation. If the gaussian approximations for $p(z_t|x_t)$ and $p(z_t)$ are learned separately, some care may need to be taken to ensure the resulting approximation to $p(x_t|z_t)$ remains a good one. One strategy might be to learn a gaussian-shaped approximation to the density ratio $p(z_t|x_t)/p(z_t)$ as a function of z_t (Sugiyama, Suzuki, & Kanamori, 2012). Another strategy might be to use the robust DKF approximation as in section 2.3, which simply drops the denominator in equation 2.15. In future work, we plan to explore these and other approaches that might allow

a DKF-style approximation to be incorporated into more general filtering models.

3 Learning the DKF

The parameters in the DKF are A, Γ, $f(\cdot)$, and $Q(\cdot)$. (S is specified from A and Γ using the stationarity assumption.) In many problems, some or all of these parameters might be unknown or not easily computable. In this section, we discuss some strategies for learning or approximating the parameters in the situation where fully supervised training data are available, meaning that we have a sequence of (Z_t, X_t) pairs assumed to be sampled from the underlying Bayesian network in equation 1.1 and denoted $(z'_1, x'_1), \ldots, (z'_m, x'_m)$. This training data might be real data or might be simulated from a known generative model for which the parameters, particularly f and Q, are not easily computable.

We use $\hat{A}$, $\hat{\Gamma}$, $\hat{f}$, and $\hat{Q}$ to denote the respective learned parameters. We consider only the situation where the parameters are learned from training data and then fixed for subsequent filtering on a different sequence of observations. In particular, for filtering, we simply replace each parameter with its corresponding estimate in the DKF algorithm in equation 2.7. We do not consider a more fully Bayesian approach, where parameter uncertainty is propagated through the filtering equations.

A and Γ are the parameters of a well-specified statistical model given by equations 2.1a and 2.1b. In the learning experiments below, we learn them from (z'_{t-1}, z'_t) pairs using only equation 2.1b, which reduces to multiple linear regression and is a common approach when learning the parameters of a Kalman filter from fully observed training data (see, e.g., Wu et al., 2002).

The parameters f and Q are more unusual, since they are not uniquely defined by the model, but are introduced via a gaussian approximation in equation 2.2. One possibility, and the one we focus on here, is to define f and Q via equation 2.3 and then learn them directly from training data as

$$\hat{f}(x) \approx f(x) = \mathbb{E}(Z_t | X_t = x) \quad \text{and} \quad \hat{Q}(x) \approx Q(x) = \mathbb{V}(Z_t | X_t = x). \tag{3.1}$$

Using equation 3.1, we learn f and Q from (z'_t, x'_t) pairs ignoring the overall temporal structure of the data, which reduces to a standard nonlinear regression problem with heteroskedastic variance. The conditional mean f can be learned using any number of off-the-shelf regression tools, and then Q can be learned from the residuals, ideally using a held-out portion of the training data. We think that the ability to easily incorporate off-the-shelf discriminative learning tools into a closed-form filtering equation is one of the most exciting and useful aspects of this approach.

In the experiments that follow, we compare three standard nonlinear regression methods for learning f: Nadaraya-Watson (NW) kernel regression, neural network (NN) regression, and gaussian process (GP) regression. (Details are in sections A.4 to A.6.) While we have found that these methods work well with the DKF framework, one could readily use any arbitrary regression model.

For learning Q, we first define $R_t = Z_t - f(X_t)$ and $\hat{R}_t = Z_t - \hat{f}(X_t)$, so that

$$Q(x) = \mathbb{V}(Z_t|X_t = x) = \mathbb{E}(R_t R_t^\intercal|X_t = x) \approx \mathbb{E}(\hat{R}_t \hat{R}_t^\intercal|X_t = x). \tag{3.2}$$

The final expression in equation 3.2 is a conditional expectation and can in principle be learned with regression on $(\hat{R}_t \hat{R}_t^\intercal, X_t)$ pairs. Learning Q in this way using off-the-shelf regression tools is more challenging because of the additional requirement that $Q(x)$ be a valid covariance matrix. Since $\hat{R}_t \hat{R}_t^\intercal$ is positive semidefinite, any regression estimator that is a weighted average of the training data with only nonnegative weights will also be positive semidefinite and, in most cases, positive definite. NW kernel regression constitutes one such method, and we use it for learning Q in all of our examples. Given a subset of the training set $\{(z_i'', x_i'')\}_{i=1}^k$, distinct from the subset used to learn the function f, we define the residuals $\hat{r}_i = z_i'' - \hat{f}(x_i'')$, and then learn Q using NW kernel regression via

$$\hat{Q}(x) = \frac{\sum_{i=1}^k \hat{r}_i \hat{r}_i^\intercal \kappa(x, x_i'')}{\sum_{i=1}^k \kappa(x, x_i'')}, \tag{3.3}$$

for a kernel $\kappa : \mathcal{X} \times \mathcal{X} \to [0, \infty)$. (Complete details are in section A.4.)

4 Examples

In this section, we compare filter performance on both artificial models and real neural data. Corresponding Matlab code, and Python code for the long short-term memory (LSTM) comparison, is freely available online at https://github.com/burkh4rt/Discriminative-Kalman-Filter under the GNU General Public License v3.0 to encourage code use and adaptation. For timing comparisons, the code was run on a Mid-2018 MacBook Pro laptop with a 2.6 GHz Intel Core i7 processor using Matlab v. 2019a and Python v. 3.6.8.

4.1 Kalman Observation Model. The stationary Kalman filter observation model is

$$p(x_t|z_t) = \eta_n(x_t; b + Hz_t, \Lambda)$$

for observations in $\mathcal{X} = \mathbb{R}^{n\times 1}$ and for fixed $b \in \mathbb{R}^{n\times 1}$, $H \in \mathbb{R}^{n\times d}$, and $\Lambda \in \mathbb{S}_n$. Defining f and Q via equation 2.3 gives

$$Q(x) \equiv Q = (S^{-1} + H^\top \Lambda^{-1} H)^{-1} \quad \text{and} \quad f(x) = QH^\top \Lambda^{-1}(x - b).$$

It is straightforward to verify that the DKF in equation 2.7 is exactly the well-known Kalman filter recursion. Hence, the DKF computes the exact posterior $p(z_t|x_{1:t})$ in this special case.

4.2 Kalman Observation Mixtures. This example and the next are designed to illustrate how the gaussian approximation underlying the DKF is more similar in spirit to the G-ADF than to linearization approximations such as the Kalman filter, the EKF, and the UKF (see section 2.4). In particular, the specific observation model used in the simulation is engineered so that the state Z_t and the observation X_t are uncorrelated (but not independent). Linearization methods are useless in this case, whereas the DKF is able to take advantage of the higher-order dependence, much like the G-ADF.

The observation model is a probabilistic mixture of Kalman observation models (see section 4.1), namely,

$$p(x_t|z_t) = \sum_{\ell=1}^{L} \pi_\ell \eta_n(x_t; b_\ell + H_\ell z_t, \Lambda_\ell),$$

for a probability vector $\pi = \pi_{1:L}$, where each $b_\ell \in \mathbb{R}^{n\times 1}$, $H_\ell \in \mathbb{R}^{n\times d}$, and $\Lambda_\ell \in \mathbb{S}_n$. At each time step, one of L possible Kalman observation models is randomly and independently selected according to π and then used to generate the observation. This model can be viewed as a special case of a switching state-space model with independent switching (see Shumway & Stoffer, 1991; Ghahramani & Hinton, 2000). The integrals in equation 2.11 can be efficiently computed for any choice of ν_t and M_t, including $\nu_t = 0$ and $M_t = S$, so the G-ADF and the DKF can be computed exactly for this model (see section A.1 for details), although the DKF is much faster for large n, because it allows for more precomputation. Figure 1 illustrates that the DKF is comparable to the G-ADF in terms of root mean squared error (RMSE) for a particular instance of this model, and it also shows that the computational savings of the DKF over a particle filter with similar accuracy can be dramatic, especially as n gets large.

Define $\bar{b} = \sum_\ell \pi_\ell b_\ell$ and $\bar{H} = \sum_\ell \pi_\ell H_\ell$ so that

$$\mathbb{E}(X_t|Z_t) = \bar{b} + \bar{H} Z_t. \tag{4.1}$$

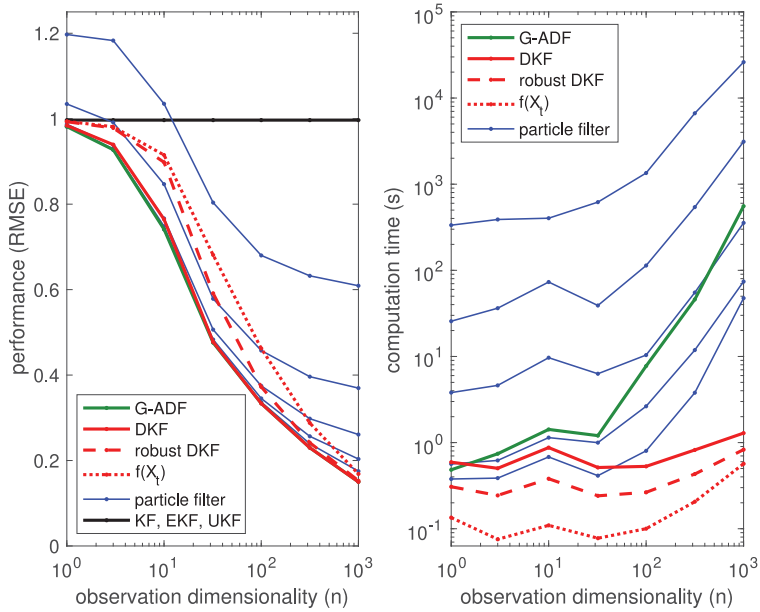

Figure 1: Kalman observation mixtures. This figure shows filtering performance on an instance of the model in section 4.2 for various approximation algorithms as the observation dimension n increases. The hidden state dimension is $d = 10$, and the state model parameters are $S = I_d$, $A = 0.95I_d - 0.05$, and $\Gamma = S - ASA^\intercal$. The number of categories is $L = 2$, the category probabilities are $\pi = (0.5, 0.5)$, and the Kalman parameters are $b_1 = b_2 = \bar{b} = 0$, $\Lambda_1 = I_n$, $\Lambda_2 = 5I_n$, and $H_2 = -H_1$, so that $\bar{H} = 0$ (see equation 4.1). The entries of H_1 were generated as independent $N(0, d^{-1})$ using the Matlab 9.6 code `rng(42,'twister'); H = randn(1000,10)/sqrt(10)`. The data were generated for an observation dimension of 1000, and the plot shows filter performance using only the first n dimensions of X_t for selected n between 1 and 1000. Filter performance was measured using RMSE (left panel) and computation time (s, right panel) on a single test sequence of length $T = 10^4$. Because X_t and Z_t are uncorrelated, linearization methods (e.g., KF, EKF, and UKF) ignore X_t and always predict $Z_t \approx \mathbb{E}(Z_t) = 0$, giving an RMSE of approximately 1 (black line) in this case. The accuracy of particle filtering increases with the number of particles at the expense of increased computation, and we show performance for different numbers of particles: $10^1, 10^2, 10^3, 10^4, 10^5$ (blue lines, ordered as expected). We also show RMSE for the optimal prediction using only X_t (as opposed to the entire history $X_{1:t}$), namely, $Z_t \approx \mathbb{E}(Z_t|X_t) = f(X_t)$ (dotted red line). (This serves to demonstrate the performance gain that filtering provides.) Finally, we caution that the model parameters have much more influence on the relative performance of the different gaussian approximation methods when n is small than when n is large. The parameters in this model were chosen so that the DKF also performs well for small n, even though we only have guarantees about its performance in the large n setting.

An interesting special case of this model is when $\bar{H} = 0$, so that the mean of X_t given Z_t does not depend on Z_t, and, consequently, X_t and Z_t are uncorrelated. Information about the states is found only in higher-order moments of the observations. Algorithms that are designed around $\mathbb{E}(X_t|Z_t)$, such as the Kalman filter, EKF, and UKF, are not useful when $\bar{H} = 0$, illustrating the important difference between a gaussian approximation for the observation model and the DKF approximation in equation 2.2. The simulation in Figure 1 used $\bar{H} = 0$, and the ineffectiveness of linearization techniques is easily seen.

4.3 Independent Bernoulli Mixtures. Here we describe a model where observations take values in $\{0, 1\}^n$ to further emphasize that our gaussian approximation is in the state space, not in the observation space. Like the example in section 4.2, this example is also engineered so that the states and observations are uncorrelated, rendering linearization-based methods ineffective (see section 2.4). Finally, the specific parameters of this example are chosen to have the peculiar property that the DKF is efficiently computable, whereas the G-ADF is not (insofar as we can tell).

The observation model is a probabilistic mixture of conditionally independent Bernoulli random variables, namely,

$$p(x_t|z_t) = \sum_{\ell=1}^{L} \pi_\ell \prod_{i=1}^{n} g_{\ell i}(z_t)^{x_{ti}} (1 - g_{\ell i}(z_t))^{1-x_{ti}},$$

for a probability vector $\pi = \pi_{1:L}$. For each $\ell = 1, \ldots, L$ and $i = 1, \ldots, n$, the functions $g_{\ell i} : \mathbb{R}^{d\times 1} \to (0, 1)$ are defined by

$$g_{\ell i}(z_t) = \alpha_{\ell i} \mathbb{1}\{z_{td_i} < \gamma_i\} + \beta_{\ell i} \mathbb{1}\{z_{td_i} \geq \gamma_i\},$$

where each $\gamma_i \in \mathbb{R}$, $\alpha_{\ell i}, \beta_{\ell i} \in (0, 1)$, $d_i \in \{1, \ldots, d\}$, and where z_{tk} indicates the kth coordinate of z_t. The ith coordinate of X_t depends on Z_t only through the d_ith coordinate of Z_t, and the probability distribution of X_{ti} is different depending on whether $Z_{td_i} < \gamma_i$. Each of the L components of the mixture changes the probability distribution of X_{ti}, via $\alpha_{\ell i}$ and $\beta_{\ell i}$, but it does not change the corresponding coordinate d_i or the change point γ_i.

For the state dynamics, we use $S = I_d$, which makes it possible to compute $f(z_t)$ and $Q(z_t)$ exactly (see section A.2). In general, however, the integrals in equation 2.11 are not easily evaluated, so the G-ADF is not a practical approximation technique in this example. Figure 2 suggests that the DKF approximation performs well for a particular instance of this model, in the sense that the DKF's RMSE is near or better than that of a particle filter with a large number of particles. The figure also shows that the computational savings over a particle filter with similar accuracy can be dramatic, especially as n gets large.

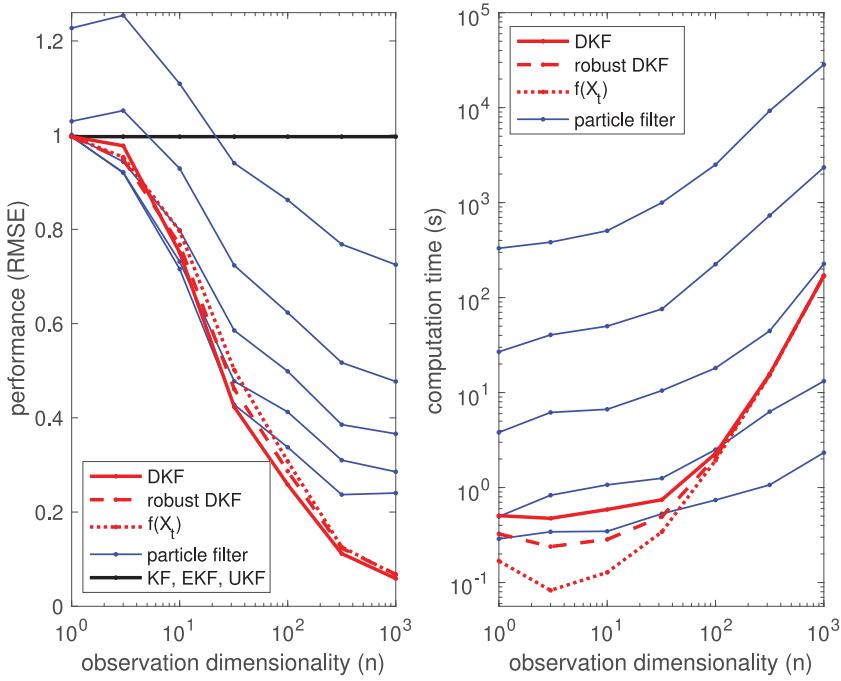

Figure 2: Independent Bernoulli mixtures. This figure shows filtering performance on an instance of the model in section 4.3 for various approximation algorithms as the observation dimension n increases. The state model (Z_t) and the figure conventions (and cautions) are the same as those described in the Figure 1 caption. (Using this many particles with higher n was too time-consuming.) The number of categories is $L = 2$, the category probabilities are $\pi = (0.5, 0.5)$, and for each i, $\alpha_{1i} = \beta_{2i} = 0.01$ and $\alpha_{2i} = \beta_{1i} = 0.99$, so that each $\bar{g}_i \equiv 0.5$ (see equation 4.2). The $d_{1:n}$ were chosen as independent uniform$\{1, \ldots, d\}$, and the $\gamma_{1:n}$ were chosen as independent $N(0, 1)$.

Define $\bar{g}_i = \sum_\ell \pi_\ell g_{\ell i}$, so that

$$\mathbb{E}(X_{ti}|Z_t) = \mathbb{P}(X_{ti} = 1|Z_t) = \bar{g}_i(Z_t). \tag{4.2}$$

An interesting special case of this model is when $\bar{g}_i$ is constant for each i, so that the mean of X_t given Z_t does not depend on Z_t, and, consequently, X_t and Z_t are uncorrelated. As in the previous section, linearization approximations like the Kalman filter, EKF, and UKF are not useful when $\bar{g}_i$ is constant. Furthermore, when $\bar{g}_i$ is constant, X_{ti} and Z_t are independent, that is, individual coordinates of the observations carry no information about the states. Only the vector of observations X_t can be used for meaningful

predictions of Z_t. The simulation in Figure 2 used $\bar{g}_i \equiv 0.5$ for all i, so that each coordinate of the observations is independent of the state.

4.4 Kalman Observation Mixtures with Nonlinear State Dynamics. This example illustrates how the DKF approximation can be combined with other filtering approximations for use with nonlinear state dynamics (see section 2.5). We include it here as a proof of concept and leave for future work a more thorough exploration of when the DKF approximation is useful for filtering with nonlinear state dynamics. We use the same mixture of Kalman observation models from section 4.2 but modify the state dynamics in equation 2.1 as follows. Define the 2×2 rotation matrix $R(\theta) = \left(\begin{smallmatrix} \sin\theta & \cos\theta \\ -\cos\theta & \sin\theta \end{smallmatrix}\right)$, and for even d, define the $d\times d$ rotation matrix $R_d(\theta)$ to be the block-diagonal matrix with $R(\theta)$ repeated along the diagonal:

$$R_d(\theta) = \begin{pmatrix} R(\theta) & 0 & \cdots & 0 \\ 0 & R(\theta) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R(\theta) \end{pmatrix}.$$

Define the function $a : \mathbb{R}^{d\times1} \to \mathbb{R}^{d\times1}$ via $a(z) = AR_d(|z|)z$, where $|\cdot|$ denotes the Euclidean norm. The new state dynamics are

$$\begin{aligned} p(z_0) &= \eta_d(z_0; 0, S), \\ p(z_t|z_{t-1}) &= \eta_d(z_t; a(z_{t-1}), \Gamma), \end{aligned}$$

for $t = 1, 2, \ldots$, where $S = ASA^\intercal + \Gamma$. These are the same dynamics as before except that the conditional mean of Z_t given Z_{t-1} has changed from the linear function AZ_{t-1} to the nonlinear function $a(Z_{t-1})$. In particular, before being multiplied by A, the state vector is rotated by an amount that depends on its length. This type of nonlinearity was chosen because when $S = I_d$ (as in our examples), Z_t remains marginally gaussian, an important part of the DKF approximation.

We use an unscented Kalman filter (UKF) approximation for the state dynamics; that is, we replaced ν_t and M_t in equations 2.7 and 2.8 with the mean and covariance obtained from performing the unscented transform (Julier & Uhlmann, 1997). We used Matlab's `unscentedKalmanFilter` implementation with `alpha=1,beta=kappa=0`. The UKF approximations of ν_t and M_t can also be substituted directly into the G-ADF used in section 4.2.

Figure 3 shows filtering performance for a specific instance of this model and illustrates that at least in this case, a DKF approximation for nonlinear, nongaussian observation models can be usefully combined with other approximations for nonlinear state dynamics and that there is little loss of performance compared to the G-ADF.

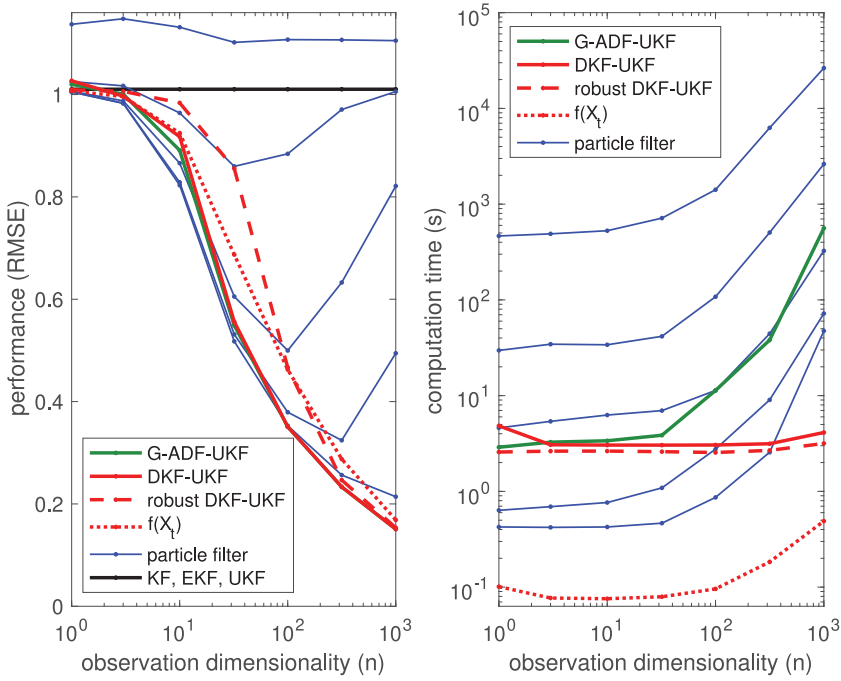

Figure 3: Nonlinear state dynamics. This figure shows filtering performance on an instance of the model in section 4.4 for various approximation algorithms as the observation dimension n increases. The observation model $(X_t|Z_t)$ and the figure conventions (and cautions) are the same as those described in the Figure 1 caption. The state model is now nonlinear, and μ_t and M_t in the DKF, robust DKF, and G-ADF are approximated using a UKF.

4.5 Unknown Observation Model: Macaque Reaching-Task Data. This example illustrates Bayesian filtering in a case where the observation model is unknown and must be learned from data. Flint, Lindberg, Jordan, Miller, and Slutzky (2012) implanted a rhesus monkey with a 96-channel microelectrode array (Blackrock Microsystems LLC) over the arm area of its primary motor cortex (M1). The monkey was trained to move a manipulandum to acquire illuminated targets for a juice reward. While performing this task, the monkey's neural spikes were recorded with a 128-channel acquisition system (Cerebus, Blackrock Microsystems LLC). The signal was sampled at 30 kHz, high-pass-filtered at 300 Hz, and then thresholded and manually sorted into spikes offline. Walker and Kording (2013) continue to make these data publicly available as part of the Database for Reaching Experiments and Models (DREAM). We used data from Flint et al. (2012) and aggregated spike counts over 100 ms bins. The first $n = 10$ principal

Table 1: Normalized RMSE (nRMSE) for Various Filtering Methods on the Flint Data Set.

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	Average
Kalman	0.765	0.942	0.788	0.793	0.780	0.765	0.805
DKF-NW	−21%	−18%	−17%	−23%	−20%	−23%	−20%
DKF-GP	−21%	−19%	−15%	−20%	−18%	−20%	−19%
DKF-NN	−19%	−15%	−13%	−13%	−13%	−17%	−15%
LSTM	−15%	−19%	−16%	−13%	−16%	−11%	−15%
EKF	2%	24%	12%	18%	12%	3%	12%
UKF	2%	31%	18%	18%	15%	6%	15%

Notes: The nRMSE is computed by dividing the RMSE by the root mean square of the observation vector, so that predicting identically zero would yield an nRMSE of 1. The top row shows the nRMSE of the Kalman filter. Each remaining row shows the percentage change in nRMSE relative to the Kalman filter, with methods ordered from best (top) to worst (bottom) average performance. Columns 1 to 6 refer to completely separate trials using new training and testing data. The final column gives the average performance across the six trials.

Table 2: Mean Absolute Angular Error (Radians) for Various Filtering Methods on the Flint Data Set.

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	Average
Kalman	0.889	0.955	1.025	0.933	0.964	0.926	0.949
DKF-NW	−15%	−1%	−20%	−17%	−25%	−28%	−18%
DKF-GP	−11%	7%	−22%	−16%	−24%	−25%	−15%
DKF-NN	−7%	−2%	−17%	−16%	−21%	−23%	−14%
LSTM	−2%	−2%	−12%	−6%	−10%	−8%	−7%
UKF	0%	3%	−3%	−3%	−8%	−6%	−3%
EKF	4%	3%	−2%	−4%	−8%	−7%	−2%

Notes: Because cursor speed is often adjustable in BCIs (Willett et al., 2019), this may provide a more informative measure of performance. See the caption for Table 1 for more details about the table arrangement. Note that $45^\circ = \pi/4 \approx 0.79$ radians, so all of these methods have fairly substantial angular error over 100 ms prediction intervals. Chance performance would be $\pi/2 \approx 1.57$ radians.

component analysis (PCA) components of neural data became the observed variable X_t, and we used the $d = 2$-dimensional (horizontal and vertical) cursor velocity (lagged 50 ms after the end of the spike count bin) as the latent variable Z_t.

Tables 1 and 2 compare filtering performance using various learning algorithms and filtering methods. For learning the function $f : \mathbb{R}^{10} \to \mathbb{R}^2$ for the DKF, we experimented with Nadaraya-Watson (NW) kernel regression, neural network (NN) regression, and gaussian process (GP) regression. In each case, we learned the function $Q : \mathbb{R}^{10} \to \mathbb{S}_2$ using NW kernel

regression from the approximate residuals as in equation 3.3. For the Kalman filter, parameters are learned in the usual manner via multivariate (linear) regression. For the EKF and UKF (see section 2.4), we learned the conditional mean $h : \mathbb{R}^2 \to \mathbb{R}^{10}$ defined by

$$h(z) = \mathbb{E}(X_t | Z_t = z) \tag{4.3}$$

via neural network regression and took the conditional covariance to be constant, namely, $\Lambda(z) = \mathbb{V}(X_t | Z_t = z) \equiv \Lambda \in \mathbb{S}_{10}$, which we learned from the approximate residuals. Finally, we also experimented with an LSTM recurrent neural network for predicting Z_t given $X_{1:t}$. In all cases, we used 5000 training points and a different 1000 testing points. (More details about all of these methods are in sections A.4 to A.7.)

The DKF using NW kernel regression was the best method among the ones that we tried, and all versions of the DKF were near the top in performance. Under the mean absolute angular error (MAAE) metric (Simeral, Kim, Black, Donoghue, & Hochberg, 2011), each version of the DKF outperformed prediction using the corresponding $\hat{f}$, illustrating the benefit of filtering to combine information from both past and present observations. The EKF and UKF performed poorly. We do not know the degree to which poor performance is a result of errors introduced by the EKF and UKF approximations or a result of errors introduced from learning the function h in equation 4.3. All versions of the DKF outperformed the LSTM that we used. The LSTM and its variants require manually selecting a neural network architecture and several tuning parameters. This is often done by experts through trial and error. While we suspect that there is some combination of architecture and tuning parameters that would allow the LSTM to meet or exceed the DKF performance, automating this process of searching through network architecture remains an area of active research requiring extensive computational resources (Zoph & Le, 2017; Real et al., 2017).

4.6 Closed-Loop Decoding in a Person with Paralysis. Neural decoding for closed-loop brain-computer interfaces (BCIs) provided the motivating application for the development of the DKF. BCIs use neural measurements from the brain to enable voluntary control of external devices (Wolpaw, Birbaumer, McFarland, Pfurtscheller, & Vaughan, 2002; Hochberg & Donoghue, 2006; Brandman, Cash, & Hochberg, 2017). Intracortical BCI systems (iBCIs) have been shown to provide users with paralysis the ability to control computer cursors (Pandarinath et al., 2015; Jarosiewicz et al., 2015; Nuyujukian et al., 2018), robotic arms (Hochberg et al., 2012; Collinger et al., 2013), and functional electrical stimulation systems (Bouton et al., 2016; Ajiboye et al., 2017) with the real-time decoded neural activity generated during attempted movement. State-of-the-art decoding approaches have been based on the Kalman filter (Pandarinath et al.,

2017; Jarosiewicz et al., 2015; Gilja et al., 2015), with observed neural features and latent motor intention used to move external devices. To construct a supervised training set, motor intentions are inferred as vectors from the instantaneous cursor position to the target position Z_t (Brandman, Hosman et al., 2018).

The DKF is a natural choice for closed-loop neural decoding using iBCIs for a few reasons. First, evidence suggests that neurons have very complex behavior. Neurons in the motor cortex have been shown to encode direction of movement (Georgopoulos, Kettner, & Schwartz, 1988), velocity (Schwartz, 1994), acceleration (Paninski, Fellows, Hatsopoulos, & Donoghue, 2004), muscle activation (Lemon, 2008; Pohlmeyer, Solla, Perreault, & Miller, 2007), proprioception (Bensmaia & Miller, 2014), visual information related to the task (Rao & Donoghue, 2014), and preparatory activity (Churchland et al., 2012). Hence, iBCI-related recordings are highly complex and nonlinear (Vargas-Irwin, Brandman, Zimmermann, Donoghue, & Black, 2015). Moving away from the linear constraints of the Kalman filter could potentially capture more of the inherent complexity of the signals, resulting in higher end-effector control for the user.

Second, evidence suggests that the quality of control directly relates to the rate at which the decoding systems perform real-time decoding. Modern iBCI sytems update velocity estimates on the order of 20 ms (Jarosiewicz et al., 2015) or even 1 ms (Pandarinath et al., 2015). Thus, any potential filtering technique must be computationally feasible to implement for real-time use.

Third, new technologies have allowed neuroscientists to simultaneously record from increasingly large numbers of neurons. In fact, the number of observed brain signals has been growing exponentially (Stevenson & Kording, 2011). By contrast, the dimensionality of the underlying device being controlled remains small, generally not exceeding 10 dimensions (Wodlinger et al., 2015; Vargas-Irwin et al., 2010).

We previously reported how three people with spinal cord injuries used the DKF with GP regression to rapidly gain closed-loop neural control (Brandman, Burkhart et al., 2018; Brandman, Hosman et al., 2018). Here, as an additional proof of concept, we present data from a person with amyotrophic lateral sclerosis (participant T9) using the DKF. In these research sessions, the observations constitute neural data collected from an electrode array surgically implanted in the participant's brain, and the hidden states represent the intended cursor velocity. The DKF prediction of intended cursor velocity is used at each time step to move the cursor. For learning the DKF parameters, training data are collected during an initial calibration phase in which the participant is instructed to attempt to move the cursor to various target locations, and the intended velocity at each time step is assumed to be pointing from the current cursor position to the instructed target. GP regression was used to learn f, and for computational efficiency, Q was assumed to be constant and set as the covariance of the residuals.

The participant's performance using an out-of-the-box DKF was comparable to state-of-the-art decoders based on modifications of the Kalman filter designed specifically for the BrainGate2 clinical trials.

4.6.1 Participant. The participant in this study was T9, a 52-year-old right-handed male with paralysis from late-stage amyotrophic lateral sclerosis (ALSFRS-R score = 7; see Cedarbaum et al., 1999, for a detailed explanation of this metric). T9 underwent surgical placement of two 96-channel intracortical silicon microelectrode arrays (Maynard et al., 1997) (1.5 mm electrode length, Blackrock Microsystems, Salt Lake City, UT) in the motor cortex as previously described (Kim, Simeral, Hochberg, Donoghue, & Black, 2008; Simeral et al., 2011). Data were used from trial (postimplant) days 292 and 293.

4.6.2 Signal Acquisition. Raw neural signals for each channel (electrode) were sampled at 30 kHz using the NeuroPort System (Blackrock Microsystems, Salt Lake City, UT). Further signal processing and neural decoding were performed using the xPC target real-time operating system (Mathworks, Natick, MA). Raw signals were downsampled to 15 kHz for decoding and denoised by subtracting an instantaneous common average reference (Gilja et al., 2015; Jarosiewicz et al., 2015) using 40 of the 96 channels on each array with the lowest root mean square value (selected based on their baseline activity during a 1 minute reference block run at the start of each session). The denoised signal was bandpass-filtered between 250 Hz and 5000 Hz using an 8th-order noncausal Butterworth filter (Masse et al., 2015). Spike events were triggered by crossing a threshold set at 3.5 times the root mean square amplitude of each channel, as determined by data from the reference block. The neural feature used was total power in the bandpass-filtered signal (Jarosiewicz et al., 2015; Brandman, Hosman et al., 2018). Neural features were binned in 20 ms nonoverlapping increments for decoding. We used the top 40 features ranked by signal-to-noise-ratio (Malik et al., 2015).

4.6.3 Decoder Calibration. Decoder calibration was performed using the standard radial-8 task (Simeral et al., 2011; Gilja et al., 2015) using custom-built software running Matlab. An LCD monitor was placed 55 to 60 cm at a comfortable angle and orientation to T9. Targets (size = 2.4 cm, visual angle = 2.5°) were presented sequentially in a pseudorandom order, alternating between one of eight radially distributed targets and a center target (radial target distance from center = 12.1 cm, visual angle = 12.6°). Successful target acquisition required the user to place the cursor (size = 1.5 cm, visual angle = 1.6°) within the target's diameter for 300 ms, before a predetermined timeout of 15 seconds. Target timeouts resulted in the cursor moving directly to the intended target, with immediate presentation of the next target.

Calibration began with 2 minutes of open-loop presentation of a cursor; that is, the cursor moved automatically to pseudorandomly presented targets in a straight path. During this time, T9 was instructed to "imagine" or "attempt" to move the computer cursor as if he had control of it. After 2 minutes, initial hyperparameters for the GP were learned. Next, T9 acquired targets for 3 minutes with 80% of the component of the decoded vector perpendicular to the vector between the cursor and the target (Jarosiewicz et al., 2013; Velliste, Perel, Spalding, Whitford, & Schwartz, 2008), in order to assist with target acquisition. GP hyperparameters were then recomputed with all of the available data. The radial-8 task was repeated two more times with the attenuated components at 50% and 20%, for a total of 11 minutes of calibration data collected. We collected 3000 data points randomly subsampled from the 11 minutes of collected data, using all 192 neural features (96 features per array, two arrays).

4.6.4 Performance Measurement. We quantified the performance of the DKF decoder with the mFitts1 task (Gilja et al., 2015; Simeral et al., 2011). Under the Fitts model (Fitts, 1954), movement time (MT) varies linearly with the index of difficulty (ID) as

$$\mathrm{MT} = a \cdot \mathrm{ID} + b, \tag{4.4}$$

where the parameters a and b depend on the input device. Parameters are estimated using linear regression on observed (ID, MT) pairs for each input method. These estimates are then used to evaluate filter performance.

A single target was presented on the screen in a pseudorandom location, with one of three pseudorandomly fixed diameters (size = 1.6 cm, 3.5 cm, and 5.6 cm; visual angles 1.7°, 3.7°, and 5.8°). Targets were acquired by having the cursor contact the target for 500 ms, within a timeout of 10 seconds. For the mFitts1 task, the index of difficulty for each trial was calculated as follows,

$$\mathrm{ID} = \log_2\left[\frac{D}{W} + 1\right],$$

where D is the distance from the cursor's start position to the goal and W is the sum of the target's diameter and cursor's radius. Hence, $\frac{D}{W}$ reflects a measure of difficulty for acquiring targets.

4.6.5 Results. T9 acquired 98% of targets presented over two research sessions ($N = 299$) with the mFitts1 task. The Fitts regression parameters were comparable to the previously described performance by different participants (T6 and T7) using the ReFIT decoder (Gilja et al., 2015, Fig. 4.6.5,

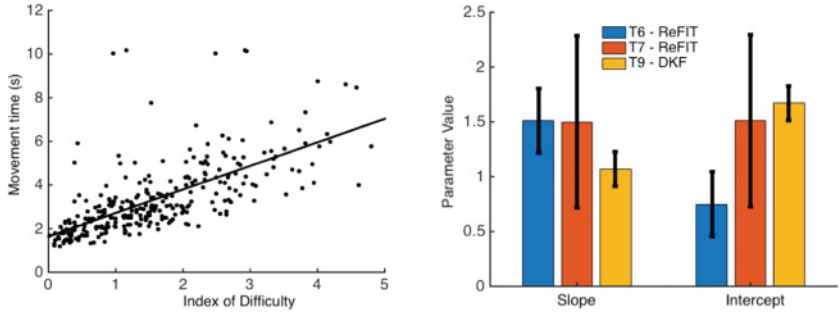

Figure 4: On the left, we plot movement time versus index of difficulty for T9 during the radial-8 task. On the right, we compare Fitts metrics for the DKF to those for Kalman ReFit. In particular, the slope and intercept from the line of best fit on the left correspond to the yellow bars for slope and intercept on the right. Error bars correspond to a 95% confidence interval for each estimated parameter. Following the discussion in section 4.6.4, lower values for the slope parameter (a in equation 4.4) correspond to less of an increase in movement time for more difficult targets. Estimates for the intercept parameter correspond to b in equation 4.4.

slope $= 1.08 \pm 0.06$, $p < 1.2 \times 10^{-30}$, intercept $= 1.6 \pm 1.3$, $p < 2.2 \times 10^{-41}$). (See Figure 4 for details.)

5 Discussion

The DKF is a novel filtering method that should prove to be a helpful addition to the filtering toolbox. It provides a fast, analytic approximation for models with linear, gaussian dynamics but nonlinear, nongaussian observations. The approximations underlying the DKF tend to improve as the dimensionality of the observation space increases relative to the dimensionality of the state space. For known models, the DKF is quite similar in nature to the G-ADF; however, when models must be learned from training data, as is the case for many practical applications, the G-ADF entails integrals that require approximation and does not provide a closed-form update. In comparison to Laplace or saddle-point approximations, the DKF provides a more global approximation to the true filtering distribution. As we demonstrate in our examples, there are many families of state-space models that render the EKF and UKF ineffective but for which the DKF performs well.

In applications where the model must be learned from supervised training data prior to filtering, off-the-shelf nonlinear or nonparametric regression tools can be used to learn the conditional mean and variance for the DKF directly, avoiding the more complicated task of learning the complete observation model $p(x_t|z_t)$. Using the DKF in this way appears to be novel

within the large literature on state-space models. Most approaches either learn a fully generative model and invert it for filtering (this includes the of use discriminative methods for training filters derived from generative models—Abbeel, Coates, Montemerlo, Ng, & Thrun, 2005; Hess & Fern, 2009) or learn a fully discriminative model that directly predicts states from the sequence of observations. The DKF allows a generative model for the state dynamics to be combined in a principled way with a discriminative model for predicting the states from the observations at individual time steps. We think that the ability to easily incorporate off-the-shelf discriminative learning tools into a closed-form filtering equation is one of the most exciting and useful aspects of this methodology.

Many promising opportunities exist to apply and extend the DKF. For example, using a gaussian approximation for $p(z_t|x_t)$ can permit a more principled approach to mitigating nonstationarities that occur in the measurement model. In neural decoding, a large change in the behavior of a single neuron that occurs between model training and filter use can result in significant performance degradation for the decoder. In the DKF framework with a GP regression model for $p(z_t|x_t)$, one can select a kernel function that ignores large differences along any single dimension. Clinical results demonstrate that this modification allows the filter to be more robust to erratic firing patterns in an arbitrary single neuron. (See Brandman, Burkhart et al., 2018, for further details.) It seems that this approach could be readily applied more generally to increase filter resilience to nonstationarities.

While the DKF assumes an approximately gaussian posterior, for general filtering models there may also be ways to incorporate the underlying gaussian approximation for $p(z_t|x_t)$ to improve performance. Methods that preserve the full form of the filtering distribution, such as particle filters, could be combined with alternatively specified measurement models, as in equation 2.15, to create general-purpose filters that are both more convenient to learn from data and use in filtering applications. The DKF marks a first step in this direction.

Appendix A: Technical Details

This appendix provides the derivations used in sections 4.2 and 4.3, along with some information on numerical stability and details for the discriminative learning methods employed in section 4.5.

A.1 Kalman Observation Mixtures. For the model in section 4.2 we provide analytic expressions for the integrals in equation 2.11, which are needed for the G-ADF and the DKF (using $\nu_t = 0$ and $M_t = S$ for the DKF). Define

$$U_{t\ell} = (M_t^{-1} + H_\ell^\intercal \Lambda_\ell^{-1} H_\ell)^{-1},$$

$$y_{t\ell} = U_{t\ell}(M_t^{-1}\nu_t + H_\ell^\intercal \Lambda_\ell^{-1}(x_t - b_\ell)),$$
$$\kappa_{t\ell} = \eta_d(\nu_t; 0, M_t)\eta_n(x_t; b_\ell, \Lambda_\ell)/\eta_d(y_{t\ell}; 0, U_{t\ell}).$$

Then

$$\begin{aligned}
a &= \int p(x_t|z_t)\eta_d(z_t; \nu_t, M_t)\, dz_t \\
&= \sum_{\ell=1}^{L} \pi_\ell \int \eta_n(x_t; b_\ell + H_\ell z_t, \Lambda_\ell)\eta_d(z_t; \nu_t, M_t)\, dz_t \\
&= \sum_{\ell=1}^{L} \pi_\ell \eta_n(x_t; b_\ell + H_\ell \nu_t, \Lambda_\ell + H_\ell M_t H_\ell^\intercal), \\
b &= \int z_t p(x_t|z_t)\eta_d(z_t; \nu_t, M_t)\, dz_t \\
&= \sum_{\ell=1}^{L} \pi_\ell \int z_t \eta_n(x_t; b_\ell + H_\ell z_t, \Lambda_\ell)\eta_d(z_t; \nu_t, M_t)\, dz_t \\
&= \sum_{\ell=1}^{L} \pi_\ell \kappa_{t\ell} \int z_t \eta_d(z_t; y_{t\ell}, U_{t\ell})\, dz_t = \sum_{\ell=1}^{L} \pi_\ell \kappa_{t\ell} y_{t\ell}, \\
c &= \int z_t z_t^\intercal p(x_t|z_t)\eta_d(z_t; \nu_t, M_t)\, dz_t = \sum_{\ell=1}^{L} \pi_\ell \kappa_{t\ell} \int z_t z_t^\intercal \eta_d(z_t; y_{t\ell}, U_{t\ell})\, dz_t \\
&= \sum_{\ell=1}^{L} \pi_\ell \kappa_{t\ell}(U_{t\ell} + y_{t\ell} y_{t\ell}^\intercal).
\end{aligned}$$

A.2 Independent Bernoulli Mixtures. For the model in section 4.3, we provide analytic expressions for the integrals in equation 2.11 for the special case of $\nu_t = 0$ and $M_t = S = I_d$, which are needed for the DKF. For each $k = 1, \ldots, d$, define $N_k = \{i : d_i = k\}$, $\Gamma_k = \{\gamma_i : i \in N_k\}$, $n_k = |\Gamma_k|$, and let $\gamma_{k,1} < \cdots < \gamma_{k,n_k}$ denote the sorted (distinct) values in Γ_k, using $\gamma_{k,0} = -\infty$ and $\gamma_{k,n_k+1} = +\infty$. Using $\eta(u) = \eta_1(u; 0, 1)$ to denote the standard normal pdf and $\phi(v) = \int_{-\infty}^{v} \eta(u)du$ to denote the corresponding distribution function, define

$$\Phi_{kj} = \int_{\gamma_{k,j-1}}^{\gamma_{k,j}} \eta(u)du = \phi(\gamma_{k,j}) - \phi(\gamma_{k,j-1}),$$
$$\Phi'_{kj} = \int_{\gamma_{k,j-1}}^{\gamma_{k,j}} u\eta(u)du - \Phi_{kj} = \eta(\gamma_{k,j-1}) - \eta(\gamma_{k,j}) - \Phi_{kj},$$

$$\Phi''_{kj} = \int_{\gamma_{k,j-1}}^{\gamma_{k,j}} u^2 \eta(u) du - \Phi_{kj} - 2\Phi'_{kj} = \gamma_{k,j-1}\eta(\gamma_{k,j-1}) - \gamma_{k,j}\eta(\gamma_{k,j}) - 2\Phi'_{kj},$$

$$\rho_{\ell ij} = \alpha_{\ell i}\mathbb{1}\{\gamma_{k,j} \le \gamma_i\} + \beta_{\ell i}\mathbb{1}\{\gamma_i < \gamma_{k,j}\}, \quad (i \in N_k),$$

for $k = 1, \ldots, d$ and $j = 1, \ldots, n_k + 1$ and $\ell = 1, \ldots, L$.

Let $x_{tN_k} = (x_{ti} : i \in N_k)$ and define

$$D_{\ell kj}(x_{tN_k}) = \prod_{i \in N_k} \rho_{\ell ij}^{x_{ti}} (1 - \rho_{\ell ij})^{1 - x_{ti}},$$

$$\begin{aligned} p_\ell(x_{tN_k}|z_{tk}) &= \prod_{i \in N_k} \left(\alpha_{\ell i}^{x_{ti}} (1 - \alpha_{\ell i})^{1 - x_{ti}} \mathbb{1}\{z_{tk} < \gamma_i\} + \beta_{\ell i}^{x_{ti}} (1 - \beta_{\ell i})^{1 - x_{ti}} \mathbb{1}\{z_{tk} \ge \gamma_i\} \right) \\ &= \sum_{j=1}^{n_k+1} \mathbb{1}\{\gamma_{k,j-1} \le z_{tk} < \gamma_{k,j}\} D_{\ell kj}(x_{tN_k}), \end{aligned}$$

so that $p(x_t|z_t) = \sum_{\ell=1}^{L} \pi_\ell \prod_{k=1}^{d} p_\ell(x_{tN_k}|z_{tk})$ and (with $S = I_d$),

$$p(x_t|z_t)\eta_d(z_t; 0, S) = p(x_t|z_t) \prod_{k=1}^{d} \eta(z_{tk}) = \sum_{\ell=1}^{L} \pi_\ell \prod_{k=1}^{d} p_\ell(x_{tN_k}|z_{tk})\eta(z_{tk}).$$

Hence, using $\delta_{kr} = \mathbb{1}\{k = r\}$,

$$\begin{aligned} a &= \int p(x_t|z_t)\eta_d(z_t; 0, S)\, dz_t = \sum_{\ell=1}^{L} \pi_\ell \prod_{k=1}^{d} \int p_\ell(x_{tN_k}|z_{tk})\eta(z_{tk})\, dz_{tk} \\ &= \sum_{\ell=1}^{L} \pi_\ell \prod_{k=1}^{d} \sum_{j=1}^{n_k+1} D_{\ell kj}(x_{tN_k}) \int_{\gamma_{k,j-1}}^{\gamma_{k,j}} \eta(z_{tk})\, dz_{tk} \\ &= \sum_{\ell=1}^{L} \pi_\ell \prod_{k=1}^{d} \sum_{j=1}^{n_k+1} D_{\ell kj}(x_{tN_k})\Phi_{kj}, \\ b_r &= \int z_{tr} p(x_t|z_t)\eta_d(z_t; 0, S)\, dz_t \\ &= \sum_{\ell=1}^{L} \pi_\ell \prod_{k=1}^{d} \sum_{j=1}^{n_k+1} D_{\ell kj}(x_{tN_k})(\Phi_{kj} + \Phi'_{kj}\delta_{kr}), \\ c_{rs} &= \int z_{tr} z_{ts} p(x_t|z_t)\eta_d(z_t; 0, S)\, dz_t \\ &= \sum_{\ell=1}^{L} \pi_\ell \prod_{k=1}^{d} \sum_{j=1}^{n_k+1} D_{\ell kj}(x_{tN_k})(\Phi_{kj} + \Phi'_{kj}\delta_{kr} + \Phi'_{kj}\delta_{ks} + \Phi''_{kj}\delta_{kr}\delta_{ks}), \end{aligned}$$

where in equation 2.11, the vector $b = (b_r : r = 1, \ldots, d)$ and the matrix $c = (c_{rs} : r, s = 1, \ldots, d)$. We have $f(x) = b/a$ and $Q(x) = c/a - f(x)f(x)^\intercal$.

A.3 Measures to Prevent Numerical Instabilities. The covariance matrix Σ_t must be positive definite for the DKF algorithm to make sense. As n gets large, using $Q(x_t) = \mathbb{V}(Z_t|X_t = x_t)$, the probability that Σ_t is positive definite goes to 1 (see lemma 1 below). However, when n is small or when Q is learned, Σ_t will often not be positive definite. An easy remedy is to force $Q^{-1}(x) - S^{-1}$ to be positive semidefinite for every x by shrinking the (generalized) eigenvalues of $Q(x)$ for any x where this constraint is not satisfied. In particular, beginning with a target $Q = Q(x)$ for a given fixed x, consider the generalized eigenvalue decomposition $QV = SVD$, where $V \in \mathbb{R}^{d\times d}$ is invertible and $D \in \mathbb{R}^{d\times d}$ is diagonal. (This decomposition can be computed in Matlab using `[V,D]=eig(Q,S)`.) Let $D \wedge 1$ denote the element-wise minimum of D and 1, and define $Q' = SV(D \wedge 1)V^{-1}$. By redefining $Q(x)$ as Q', we will ensure that $Q^{-1}(x) - S^{-1}$ is positive semidefinite, as required. Moreover, Q' will be the same as the original Q if this condition was already satisfied by the original Q, showing that this modification to the DKF algorithm does not affect our asymptotic analysis. We used this modification for all of the experiments with the DKF. The robust DKF does not require this modification. Here is a proof of the claims about this method: $Q^{-1} - S^{-1}$ is positive semidefinite if and only if $S - Q$ is positive semidefinite if and only if $S^{-1/2}(S - Q)S^{-1/2}$ is positive semidefinite. We have $S^{-1/2}(S - Q)S^{-1/2} = S^{-1/2}(S - SVDV^{-1})S^{-1/2} = I - S^{1/2}VD(S^{1/2}V)^{-1} = (S^{1/2}V)(I - D)(S^{1/2}V)^{-1}$, which is positive semidefinite if and only if all entries of D (which is diagonal) are ≤ 1. Replacing D with $D \wedge 1$ exactly enforces this constraint.

For our DKF experiments with nonlinear state dynamics using an extended Kalman filter (EKF) approximation (not described here), we found that the DKF-EKF became unstable for small n because the EKF approximation to the nonlinearity was quite poor. To remedy this, we modified the DKF algorithm to prevent μ_t from diverging too far from ν_t and $f(x_t)$ (the posterior means of Z_t given $X_{1:t-1}$ and given X_t, respectively). In particular, we forced $|\mu_t|^2 \leq |\nu_t|^2 + |f(x_t)|^2$ (by scaling μ_t whenever its norm exceeded our bound). For larger n, once the DKF approximation becomes more accurate, this constraint was always satisfied in our experiments without intervention, but for smaller n, enforcing it was important for preventing numerical instabilities. The robust DKF did not require this modification. Although not used in this letter, we report this modification in case others find it useful in their application.

A.4 Nadaraya-Watson Kernel Regression. We can learn $f : \mathbb{R}^n \to \mathbb{R}^d$ with a variety of regression methods. The well-known Nadaraya-Watson kernel regression estimator (Nadaraya, 1964; Watson, 1964) is

$$\hat{f}(x) = \frac{\sum_{i=1}^{m} z_i' \kappa_X(x, x_i')}{\sum_{i=1}^{m} \kappa_X(x, x_i')},$$

where the $\kappa_X(x, x')$ is a nonnegative kernel and m is the size of the training set. Bandwidth can be chosen using rule-of-thumb or with leave-one-out cross-validation, the latter scaling as $\mathcal{O}(m^2)$. Evaluation of $\hat{f}$ scales like $\mathcal{O}(m)$. In the examples, we use a gaussian kernel with a bandwidth chosen by minimizing leave-one-out mean squared error (MSE) on the training set.

A.5 Neural Network Regression. We can also learn f as a neural network (NN). NNs are attractive for online filtering, because evaluation of $\hat{f}$ scales $\mathcal{O}(1)$ with the size of the training set. With MSE as an objective function, we optimize parameters over the training set. Typically optimization continues until performance stops improving on a validation subset (to prevent overfitting), but instead we use Bayesian regularization to ensure network generalizability (MacKay, 1992; Foresee & Hagan, 1997). Training costs depend on the training algorithm chosen. Traditional optimizers include stochastic gradient descent, scaling with $\mathcal{O}(m)$; scaled conjugate gradient, with $\mathcal{O}(m^2)$; and Levenberg-Marquardt, with $\mathcal{O}(m^3)$ (Castillo, Guijarro-Berdiñas, Fontenla-Romero, & Alonso-Betanzos, 2010), where m is the size of the training set. More recently, Hessian-free approaches have been developed to train NNs on larger data sets (Schmidhuber, 2015). Training costs also grow with d, depending on the choice of architecture.

We implemented all feedforward neural networks with Matlab's Neural Network Toolbox R2019a. Our implementation consisted of a single hidden layer of tansig neurons trained via Levenberg-Marquardt optimization (Levenberg, 1944; Marquardt, 1963; Hagan & Menhaj, 1994) with Bayesian regularization.

A.6 Gaussian Process Regression. Gaussian process (GP) regression is another popular method for nonlinear regression (Rasmussen & Williams, 2006). The idea is to put a prior distribution on the function f and approximate f with its posterior mean given training data. We first briefly describe the case $d = 1$. We form an $m \times n$-dimensional matrix X' by concatenating the $1 \times n$-dimensional vectors X_i' and an $m \times d$-dimensional matrix Z' by concatenating the vectors Z_i'. We assume that $p(z_i'|x_i', f) = \eta(z_i'; f(x_i'), \sigma^2)$, where f is sampled from a mean-zero GP with covariance kernel $K(\cdot, \cdot)$. Under this model,

$$\hat{f}(x) = \mathbb{E}(f(x)|Z', X') = K(x, X')(K(X', X') + \sigma^2 I_m)^{-1} Z',$$

where $K(x, X')$ denotes the $1 \times m$ vector with ith entry $K(x, X_i')$, $K(X', X')$ denotes the $m \times m$ matrix with ijth entry $K(X_i', X_j')$, Z' is a column vector, and

I_m is the $m \times m$ identity matrix. The noise variance σ^2 and any parameters controlling the kernel shape are hyperparameters. For our examples, we used the radial basis function kernel with two parameters: length scale and maximum covariance. These hyperparameters were selected via maximum likelihood. For $d > 1$, we repeated this process for each dimension to separately learn the coordinates of f. Training costs for a single dimension scale as $\mathcal{O}(m^3)$. Sparse approximations to GPs can reduce training requirements to $\mathcal{O}(m \cdot N_S^2)$ where N_S is the size of the sparse GP (Quiñonero Candela & Rasmussen, 2005). Evaluation of $\hat{f}$ scales $\mathcal{O}(m)$ for each dimension, or $\mathcal{O}(N_S)$ for sparse approximations.

All GP training was performed using the publicly available GPML package (Rasmussen & Nickisch, 2010).

A.7 Comparison with a Long Short-Term Memory Neural Network. An LSTM is a stateful recurrent neural network designed to overcome error backflow problems (Hochreiter & Schmidhuber, 1997). Such recurrent neural networks have previously been shown to outperform state-of-the-art Kalman-based filters on this primate neural decoding task and so provide a good point of comparison (Sussillo, Stavisky, Kao, Ryu, & Shenoy, 2016; Sussillo et al., 2012; Pandarinath et al., 2018; Hosman et al., 2019). While there are many variants on the LSTM architecture, none seem to universally improve on the basic design (Jozefowicz, Zaremba, & Sutskever, 2015; Greff, Srivastava, Koutník, Steunebrink, & Schmidhuber, 2016). LSTM optimization uses many of the same methods that work for feedforward NN's (Schmidhuber, 2015). Training and evaluation requirements are similar.

All LSTM trials were conducted with TensorFlow r1.4.0 in a Python 3.6.8 environment. The LSTM cell used in these trials was built from scratch in TensorFlow following Gers, Schmidhuber, and Cummins (2000). Dropout was used to prevent overfitting (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014), but it was only applied to feedforward connections, not recurrent connections (Pham, Bluche, Kermorvant, & Louradour, 2014; Zaremba, Sutskever, & Vinyals, 2014). The recurrent states and outputs at each intermediate time step were batch-normalized to accommodate internal covariate shift (Ioffe & Szegedy, 2015). Model parameters were initialized via a Xavier-type method (Glorot & Bengio, 2010) designed to stabilize variance from layer to layer. Optimization was then performed with Adadelta (Zeiler, 2012), an algorithm designed to improve upon Adagrad (Duchi, Hazan, & Singer, 2011) with the explicit goals of decreasing sensitivity to hyperparameters and permitting the learning rate to sometimes increase.

Appendix B: Mathematical Results

Our main technical result is theorem 2. After stating the theorem, we translate it into the setting of the letter. Probability density functions (pdfs) are

with respect to Lebesgue measure over $\mathbb{R}^d$. $\|\cdot\|_1$ and $\|\cdot\|_\infty$ denote the L_1 and L_∞ norms, respectively, $\xrightarrow{w}$ denotes weak convergence of probability measures (equivalent, for instance, to convergence of the expected values of bounded continuous functions), and δ_c denotes the unit point mass at $c \in \mathbb{R}^d$. Define the Markov transition density $\tau(y, z) = \eta_d(z; Ay, \Gamma)$, and let τh denote the function

$$(\tau h)(z) = \int \tau(y, z) h(y) dy$$

for an arbitrary, integrable h. Define $p(z) = \eta_d(z; 0, S)$, where S satisfies $S = ASA^\intercal + \Gamma$.

Theorem 2. *Fix pdfs s_n and u_n $(n \geq 1)$ so that the pdfs*

$$p_n = \frac{u_n \tau s_n / p}{\|u_n \tau s_n / p\|_1} \tag{B.1}$$

are well defined for each n. Suppose that for some $b \in \mathbb{R}^d$ and some probability measure P over $\mathbb{R}^d$:

- *A1. $s_n \xrightarrow{w} P$ as $n \to \infty$;*
- *A2. There exists a sequence of gaussian pdfs (s'_n) such that $\|s_n - s'_n\|_1 \to 0$ as $n \to \infty$;*
- *A3. $u_n \xrightarrow{w} \delta_b$ as $n \to \infty$;*
- *A4. There exists a sequence of gaussian pdfs (u'_n) such that $\|u_n - u'_n\|_1 \to 0$ as $n \to \infty$;*
- *A5. $p_n \xrightarrow{w} \delta_b$ as $n \to \infty$;*

Then:

- *C1. $s'_n \xrightarrow{w} P$ as $n \to \infty$;*
- *C2. $u'_n \xrightarrow{w} \delta_b$ as $n \to \infty$;*
- *C3. The pdf*

$$p'_n = \frac{u'_n \tau s'_n / p}{\|u'_n \tau s'_n / p\|_1}$$

  is well defined and gaussian for n sufficiently large;

- *C4. $p'_n \xrightarrow{w} \delta_b$ as $n \to \infty$;*
- *C5. $\|p_n - p'_n\|_1 \to 0$ as $n \to \infty$.*

Remark 1. The L_1 distance between pdfs is equivalent to the total variation distance between the respective probability measures.

Remark 2. We are not content to show the existence of a sequence of gaussian pdfs (p'_n) that satisfy C4 and C5. Rather, we are trying to show that the specific p'_n defined in C3 satisfies C4 and C5 regardless of the choice of u'_n and s'_n.

Remark 3. An inspection of the proof shows that the pdf

$$r'_n = p'_n p / \|p'_n p\|_1 = u'_n \tau s'_n / \|u'_n \tau s'_n\|_1$$

is well defined and gaussian, with $r'_n \xrightarrow{w} \delta_b$ and

$$\|p_n - r'_n\|_1 \leq A_n + B_n + C_n,$$

where the terms A_n, B_n, C_n are those defined in equation B.2, each of which tends to zero in the limit. Thus $\|p_n - r'_n\|_1 \to 0$. These r'_n are precisely the estimates formed using the robust DKF.

Remark 4. Suppose the pdfs s_n, s'_n, u_n, u'_n ($n \geq 1$), the constant b, and the probability measure P are themselves random, defined on a common probability space, so that p_n is well defined with probability one, and suppose that the limits in A1 to A5 hold in probability. Then the probability that p'_n is a well-defined, gaussian pdf converges to one, and the limits in C1 to C5 hold in probability.

For the setting of this letter, first fix $t \geq 1$, and note that p is the common pdf of each Z_t and τ is the common conditional pdf of Z_t given Z_{t-1}. The limit of interest is for increasing dimension (n) of a single observation. To formalize this, we let each X_t be infinite dimensional and consider observing only the first n dimensions, denoted $X_t^{1:n} \in \mathbb{R}^n$. Similarly, $X_{1:t}^{1:n} = (X_1^{1:n}, \ldots, X_t^{1:n})$. We will abuse notation and use $\mathbb{P}(Z_t = \cdot|W)$ to denote the conditional pdf of Z_t given another random variable W. These conditional pdfs (formally defined via disintegrations) exist under very mild regularity assumptions (Chang & Pollard, 1997). Note that we are in the setting of remark 4, where the randomness comes from $X_{1:t}, Z_{1:t}$. With this in mind, define

$$\begin{aligned}
u_n(\cdot) &= u_n(\cdot; X_t^{1:n}) = \mathbb{P}(Z_t = \cdot|X_t^{1:n}) \\
u'_n(\cdot) &= u'_n(\cdot; X_t^{1:n}) = \eta_d(\cdot; f_n(X_t^{1:n}), Q_n(X_t^{1:n})) \\
s_n(\cdot) &= s_n(\cdot; X_{1:t-1}^{1:n}) = \mathbb{P}(Z_{t-1} = \cdot|X_{1:t-1}^{1:n}) \quad (t > 1) \\
s'_n(\cdot) &= s'_n(\cdot; X_{1:t-1}^{1:n}) = \eta_d(\cdot; \mu_{t-1,n}(X_{1:t-1}^{1:n}), \Sigma_{t-1,n}(X_{1:t-1}^{1:n})) \quad (t > 1) \\
p_n(\cdot) &= p_n(\cdot; X_{1:t}^{1:n}) = \mathbb{P}(Z_t = \cdot|X_{1:t}^{1:n}) \\
p'_n(\cdot) &= p'_n(\cdot; X_{1:t}^{1:n}) = \eta_d(\cdot; \mu_{t,n}(X_{1:t}^{1:n}), \Sigma_{t,n}(X_{1:t}^{1:n})) \\
b &= Z_t \\
P(\cdot) &= P(\cdot; Z_{t-1}) = \delta_{Z_{t-1}} \quad (t > 1),
\end{aligned}$$

and define $s_n \equiv s'_n \equiv P \equiv p$ when $t = 0$. The pdf u'_n is our gaussian approximation of the conditional pdf of Z_t for a given $X_t^{1:n}$. We have added the

subscript n to f and Q from the main text to emphasize the dependence on the dimensionality of the observations. The pdfs s'_n and p'_n are our gaussian approximations of Z_{t-1} and Z_t given $X_{1:t-1}^{1:n}$ and $X_{1:t}^{1:n}$, respectively. Again, we added the subscript n to μ_t and Σ_t from the text. Note that equation B.1 is simply a condensed version of equation 2.4 in the main text, and, for the same reason, the p'_n defined in C3 is the same p'_n defined above.

The Bernstein–von Mises (BvM) theorem gives conditions for the existence of functions f_n and Q_n so that A3 to A4 hold in probability. We refer readers to van der Vaart (1998) for details. Very loosely speaking, the BvM theorem requires Z_t to be completely determined in the limit of increasing amounts of data, but not completely determined after observing only a finite amount of data. The simplest case is when $X_t^{1:n}$ are conditionally independent and identically distributed given Z_t, and distinct values of Z_t give rise to distinct conditional distributions for $X_t^{1:n}$, but the result holds in much more general settings. A separate application of the BvM theorem gives A5 (in probability). In applying the BvM theorem to obtain A5, we also obtain the existence of a sequence of (random) gaussian pdfs (p''_n) such that $\|p_n - p''_n\|_1 \to 0$ (in probability), but we do not make use of this result, and, as explained in remark 2, we care about the specific sequence (p'_n) defined in C3.

As long as the BvM theorem is applicable, the only remaining thing to show is A1 and A2 (in probability). For the case $t = 1$, we have $s_n \equiv s'_n \equiv P \equiv p$, so A1 and A2 are trivially true, and the theorem holds. For any case $t > 1$, we note that s_n and s'_n are simply p_n and p'_n, respectively, for the case $t - 1$. So the conclusions C4 and C5 in the case $t - 1$ become the assumptions A1 and A2 for the subsequent case t. The theorem then holds for all $t \geq 1$ by induction. The key conclusion is C5, which says that our gaussian filter approximation p'_n will be close in total variation distance (see remark 1) to the true Bayesian filter distribution p_n with high probability when n is large.

Proof. C1 follows immediately from A1 and A2. C2 follows immediately from A3 and A4. C3 and C4 are proved in lemma 1 below. To show C5, we first bound

$$\begin{aligned}\|p_n - p'_n\|_1 \leq & \underbrace{\left\| p_n - \frac{p_n p}{p(b)} \right\|_1}_{A_n} + \underbrace{\left\| \frac{p_n p}{p(b)} - \frac{p_n p}{\|p_n p\|_1} \right\|_1}_{B_n} + \underbrace{\left\| \frac{p_n p}{\|p_n p\|_1} - \frac{p'_n p}{\|p'_n p\|_1} \right\|_1}_{C_n} \\ & + \underbrace{\left\| \frac{p'_n p}{\|p'_n p\|_1} - \frac{p'_n p}{p(b)} \right\|_1}_{B'_n} + \underbrace{\left\| \frac{p'_n p}{p(b)} - p'_n \right\|_1}_{A'_n}. \end{aligned} \tag{B.2}$$

Since $p_n \xrightarrow{w} \delta_b$ and $p(z)$ is bounded and continuous,

$$A_n = \int p_n \left| 1 - \frac{p}{p(b)} \right| = \mathbb{E}_{Z_n \sim p_n} \left| 1 - \frac{p(Z_n)}{p(b)} \right| \to \left| 1 - \frac{p(b)}{p(b)} \right| = 0$$

and

$$\begin{aligned} B_n &= \int \frac{p_n p}{\|p_n p\|_1} \left| \frac{\|p_n p\|_1}{p(b)} - 1 \right| = \left| \frac{\|p_n p\|_1}{p(b)} - 1 \right| \\ &= \left| \frac{\mathbb{E}_{Z_n \sim p_n} |p(Z_n)|}{p(b)} - 1 \right| \to \left| \frac{p(b)}{p(b)} - 1 \right| = 0. \end{aligned}$$

Similarly, since $p'_n \xrightarrow{w} \delta_b$,

$$A'_n = \int p'_n \left| 1 - \frac{p}{p(b)} \right| = \mathbb{E}_{Z_n \sim p'_n} \left| 1 - \frac{p(Z_n)}{p(b)} \right| \to \left| 1 - \frac{p(b)}{p(b)} \right| = 0$$

and

$$\begin{aligned} B'_n &= \int \frac{p'_n p}{\|p'_n p\|_1} \left| \frac{\|p'_n p\|_1}{p(b)} - 1 \right| = \left| \frac{\|p'_n p\|_1}{p(b)} - 1 \right| \\ &= \left| \frac{\mathbb{E}_{Z_n \sim p'_n} |p(Z_n)|}{p(b)} - 1 \right| \to \left| \frac{p(b)}{p(b)} - 1 \right| = 0. \end{aligned}$$

All that remains is to show that $C_n \to 0$.

We first observe that

$$\frac{p_n p}{\|p_n p\|_1} = \frac{u_n \tau s_n}{\|u_n \tau s_n\|_1} \quad \text{and} \quad \frac{p'_n p}{\|p'_n p\|_1} = \frac{u'_n \tau s'_n}{\|u'_n \tau s'_n\|_1}.$$

Define

$$\alpha = \mathbb{E}_{(Y,Z) \sim P \times \delta_b} \eta_d(Z; AY, \Gamma) = \mathbb{E}_{Y \sim P} \eta_d(b; AY, \Gamma) \in (0, \infty).$$

Since $s_n \xrightarrow{w} P$, $u_n \xrightarrow{w} \delta_b$, and $(z, y) \mapsto \tau(y, z) = \eta_d(z; Ay, \Gamma)$ is bounded and continuous, we have

$$\begin{aligned} \|u_n \tau s_n\|_1 &= \iint \eta_d(z; Ay, \Gamma) s_n(y) u_n(z) dy\, dz \\ &= \mathbb{E}_{(Y_n, Z_n) \sim s_n \times u_n} \eta_d(Z_n; AY_n, \Gamma) \to \alpha. \end{aligned}$$

Similarly, since $s'_n \xrightarrow{w} P$ and $u'_n \xrightarrow{w} \delta_b$,

$$\begin{aligned} \|u'_n \tau s'_n\|_1 &= \iint \eta_d(z; Ay, \Gamma) s'_n(y) u'_n(z) dy\, dz \\ &= \mathbb{E}_{(Y_n, Z_n) \sim s'_n \times u'_n} \eta_d(Z_n; AY_n, \Gamma) \to \alpha. \end{aligned}$$

Defining $\beta = \eta_d(0; 0, \Gamma) \in (0, \infty)$, gives

$$\begin{aligned}\|\tau h\|_\infty &\leq \sup_z |(\tau h)(z)| \\ &\leq \sup_{z,y} \eta_d(z; Ay, \Gamma) \int |h(t)|dt \leq \eta_d(0; 0, \Gamma)\|h\|_1 = \beta\|h\|_1\end{aligned}$$

for any integrable h. With these facts in mind, we obtain

$$\begin{aligned}C_n &= \left\| \frac{u_n \tau s_n}{\|u_n \tau s_n\|_1} - \frac{u'_n \tau s'_n}{\|u'_n \tau s'_n\|_1} \right\|_1 \\ &\leq \left\| \frac{u_n \tau s_n}{\|u_n \tau s_n\|_1} - \frac{u'_n \tau s_n}{\|u_n \tau s_n\|_1} \right\|_1 + \left\| \frac{u'_n \tau s_n}{\|u_n \tau s_n\|_1} - \frac{u'_n \tau s'_n}{\|u'_n \tau s'_n\|_1} \right\|_1 \\ &\leq \frac{\|\tau s_n\|_\infty}{\|u_n \tau s_n\|_1} \|u_n - u'_n\|_1 + \left\| \frac{\tau s_n}{\|u_n \tau s_n\|_1} - \frac{\tau s'_n}{\|u'_n \tau s'_n\|_1} \right\|_\infty \|u'_n\|_1 \\ &\leq \frac{\beta}{\|u_n \tau s_n\|_1} \|u_n - u'_n\|_1 + \left\| \frac{\tau s_n}{\|u_n \tau s_n\|_1} - \frac{\tau s_n}{\|u'_n \tau s'_n\|_1} \right\|_\infty \\ &\quad + \left\| \frac{\tau s_n}{\|u'_n \tau s'_n\|_1} - \frac{\tau s'_n}{\|u'_n \tau s'_n\|_1} \right\|_\infty \\ &\leq \frac{\beta}{\|u_n \tau s_n\|_1} \|u_n - u'_n\|_1 + \frac{\|\tau s_n\|_\infty}{\|u_n \tau s_n\|_1} \left| 1 - \frac{\|u_n \tau s_n\|_1}{\|u'_n \tau s'_n\|_1} \right| + \frac{\|\tau s_n - \tau s'_n\|_\infty}{\|u'_n \tau s'_n\|_1} \\ &\leq \underbrace{\frac{\beta}{\|u_n \tau s_n\|_1}}_{\to \beta/\alpha} \underbrace{\|u_n - u'_n\|_1}_{\to 0} + \underbrace{\frac{\beta}{\|u_n \tau s_n\|_1}}_{\to \beta/\alpha} \underbrace{\left| 1 - \frac{\|u_n \tau s_n\|_1}{\|u'_n \tau s'_n\|_1} \right|}_{\to |1-\alpha/\alpha|=0} \\ &\quad + \underbrace{\frac{\beta}{\|u'_n \tau s'_n\|_1}}_{\to \beta/\alpha} \underbrace{\|s_n - s'_n\|_1}_{\to 0}.\end{aligned}$$

Since $\alpha > 0$, we see that $C_n \to 0$, and the proof of the theorem is complete. □

Remark 4 follows from standard arguments by making use of the equivalence between convergence in probability and the existence of a strongly convergent subsequence within each subsequence. The theorem can be applied to each strongly convergent subsequence.

Lemma 1 *(DKF equation). If* $s'_n(z) = \eta_d(z; a_n, V_n)$ *and* $u'_n(z) = \eta_d(z; b_n, U_n)$, *then defining*

$$p'_n = \frac{u'_n \tau s'_n / p}{\|u'_n \tau s'_n / p\|_1}$$

gives

$$p'_n(z) = \eta_d(z; c_n, T_n),$$

where $G_n = AV_nA^\top + \Gamma$, $T_n = (U_n^{-1} + G_n^{-1} - S^{-1})^{-1}$, *and* $c_n = T_n(U_n^{-1}b_n + G_n^{-1}Aa_n)$, *as long as* T_n *is well defined and positive definite. Furthermore, if* $s'_n \xrightarrow{w} P$ *and* $u'_n \xrightarrow{w} \delta_b$, *then* p'_n *is eventually well defined, and* $p'_n \xrightarrow{w} \delta_b$.

Proof. See above for the definition of τ, p, A, Γ, S. Assuming $u'_n\tau s'_n/p$ is integrable, we have

$$p'_n(z) \propto \frac{\eta_d(z; b_n, U_n)}{\eta_d(z; 0, S)} \int \eta_d(z; Ay, \Gamma)\, \eta_d(y; a_n, V_n)\, dy.$$

Since

$$\int \eta_d(z; Ay, \Gamma)\, \eta_d(y; a_n, V_n)\, dy = \eta_d(z; Aa_n, AV_nA^\top + \Gamma) = \eta_d(z; Aa_n, G_n)$$

and

$$\begin{aligned}
\frac{\eta_d(z; b_n, U_n)}{\eta_d(z; 0, S)} &\propto \frac{\exp(-\frac{1}{2}(z - b_n)^\top U_n^{-1}(z - b_n))}{\exp(-\frac{1}{2}z^\top S^{-1}z)} \\
&\propto \exp\Big(-\frac{1}{2}(z^\top(U_n^{-1} - S^{-1})z - 2z^\top U_n^{-1}b_n)\Big) \\
&\propto \exp\Big(-\frac{1}{2}(z - b'_n)^\top (U'_n)^{-1}(z - b'_n)\Big) \\
&\propto \eta_d(z, b'_n, U'_n)
\end{aligned}$$

for $U'_n = (U_n^{-1} - S^{-1})^{-1}$ and $b'_n = U'_nU_n^{-1}b_n$, we have

$$\begin{aligned}
p'_n(z) &\propto \eta_d(z; b'_n, U'_n)\, \eta_d(z; Aa_n, G_n) \\
&\propto \eta_d(z; T_n((U'_n)^{-1}b'_n + G_n^{-1}Aa_n), T_n) \\
&= \eta_d(z; c_n, T_n).
\end{aligned}$$

As the normal density integrates to 1, the proportionality constant drops out.

Now suppose in addition that $s'_n \xrightarrow{w} P$ and $u'_n \xrightarrow{w} \delta_b$. Consider the characteristic functions

$$\phi_{s'_n}(t) = \mathbb{E}_{X\sim s'_n}[e^{itX}] = e^{it^\top a_n - \frac{1}{2}t^\top V_n t} \quad \text{and} \quad \phi_{u'_n}(t) = e^{it^\top b_n - \frac{1}{2}t^\top U_n t}$$

for these random variables. Lévy's continuity theorem (theorem 2.13 in van der Vaart, 1998) implies that $\phi_{s'_n}(t) \to \phi_P(t)$ and $\phi_{u'_n}(t) \to \phi_{\delta_b}(t)$ for all $t \in \mathbb{R}^d$ where

$$\phi_P(t) = e^{it^\top a - \frac{1}{2}t^\top V t} \quad \text{and} \quad \phi_{\delta_b}(t) = e^{it^\top b}$$

denote the characteristic functions for P and δ_b, respectively. Here, a and V are the mean vector and covariance matrix, respectively, of the distribution P, which must itself be gaussian, although possibly degenerate. It follows that

$$(it^\top a_n - \frac{1}{2}t^\top V_n t) \to (it^\top a - \frac{1}{2}t^\top V t)$$

and as $\phi_{s'_n}(-t) \to \phi_P(-t)$,

$$(-it^\top a_n - \frac{1}{2}t^\top V_n t) \to (-it^\top a - \frac{1}{2}t^\top V t),$$

so $t^\top a_n \to t^\top a$ and $t^\top V_n t \to t^\top V t$ for all $t \in \mathbb{R}^d$. Choosing t to be coordinate vectors, we see that this implies $a_n \to a$ and $V_n \to V$ coordinate-wise. An analogous argument allows us to conclude that $b_n \to b$ and $U_n \to 0_{d\times d}$. Thus, $G_n \to G = AVA^\top + \Gamma$, which is invertible, since Γ is positive definite, and so $G_n^{-1} \to G^{-1}$.

The Woodbury matrix identity gives

$$T_n = (U_n^{-1} + G_n^{-1} - S^{-1})^{-1} = U_n - U_n((G_n^{-1} - S^{-1})^{-1} + U_n)^{-1}U_n. \quad \text{(B.3)}$$

Since $U_n \to 0_{d\times d}$ and $((G_n^{-1} - S^{-1})^{-1} + U_n)^{-1} \to G^{-1} - S^{-1}$, we see that $T_n \to 0_{d\times d}$.

To show T_n is eventually well defined and strictly positive definite, it suffices to show the same for

$$T_n^{-1} = U_n^{-1} + D_n,$$

where we set $D_n = G_n^{-1} - S^{-1}$. For a symmetric matrix $M \in \mathbb{R}^{d\times d}$, let $\lambda_1(M) \geq \cdots \geq \lambda_d(M)$ denote its ordered eigenvalues. As a corollary to Hoffman and Wielandt's result (see corollary 6.3.8 in Horn & Johnson, 2013), it follows that

$$\max_j |\lambda_j(T_n^{-1}) - \lambda_j(U_n^{-1})| \leq \|D_n\|_2,$$

where $\|\cdot\|_2$ denotes the Frobenius norm. Since $\|D_n\|_2 \to \|G^{-1} - S^{-1}\|_2 < \infty$, the difference between the jth ordered eigenvalues for T_n^{-1} and U_n^{-1} is

upper-bounded independent of n for $1 \leq j \leq d$. Since U_n is positive definite and since $U_n \to 0_{d\times d}$, it follows that $\lambda_j(U_n^{-1}) \geq \lambda_d(U_n^{-1}) = 1/\lambda_1(U_n) \to \infty$. Hence, all eigenvalues of T_n^{-1} must eventually become positive, so that T_n^{-1} becomes positive definite, hence also T_n.

For the means, we have

$$c_n = T_n U_n^{-1} b_n + T_n G_n^{-1} A a_n.$$

Because $T_n \to 0_{d\times d}$, $G_n^{-1} \to G^{-1}$, and $a_n \to a$, we have $T_n G_n^{-1} A a_n \to \vec{0}$. Using equation B.3 for T_n gives

$$T_n U_n^{-1} b_n = b_n - U_n((G_n^{-1} - S^{-1})^{-1} + U_n)^{-1} b_n,$$

where the eventual boundedness of $(G_n^{-1} - S^{-1})^{-1} + U_n)^{-1}$ implies

$$U_n((G_n^{-1} - S^{-1})^{-1} + U_n)^{-1} b_n \to \vec{0}.$$

As $b_n \to b$, we conclude $c_n \to b$. Hence, $p'_n \xrightarrow{w} \delta_b$. □

Acknowledgments

We thank participant T9 and T9's family, the anonymous reviewers, and E. Crites for their thoughtful feedback on the manuscript; B. Travers and D. Rosler for administrative support; and C. Grant for clinical assistance. This work was supported by the National Institutes of Health: National Institute on Deafness and Other Communication Disorders, NIDCD (R01DC009899), Rehabilitation Research and Development Service, Department of Veterans Affairs (B6453R and N9228C); National Science Foundation (DMS1309004); National Institute of Health (IDeA P20GM103645, R01MH102840); Massachusetts General Hospital (MGH)–Deane Institute for Integrated Research on Atrial Fibrillation and Stroke; Joseph Martin Prize for Basic Research; the Executive Committee on Research of Massachusetts General Hospital; Canadian Institute of Health Research (336092); Killam Trust Award Foundation; and the Brown Institute of Brain Science. The content of this letter is solely our responsibility and does not necessarily represent the official views of the National Institutes of Health, the Department of Veterans Affairs, or the U.S. government.

References

Abbeel, P., Coates, A., Montemerlo, M., Ng, A. Y., & Thrun, S. (2005). Discriminative training of Kalman filters. In *Proceedings of Robotics: Science and Systems*. Cambridge, MA: MIT Press.

Ajiboye, A. B., Willett, F. R., Young, D. R., Memberg, W. D., Murphy, B. A., Miller, J. P., . . . Kirsch, R. F. (2017). Restoration of reaching and grasping movements through brain-controlled muscle stimulation in a person with tetraplegia: A proof-of-concept demonstration. *Lancet*, *389*, 1821–1830.

Arasaratnam, I., & Haykin, S. (2009). Cubature Kalman filters. *IEEE Trans. Autom. Control*, *54*(6), 1254–1269.

Arasaratnam, I., Haykin, S., & Elliott, R. J. (2007). Discrete-time nonlinear filtering algorithms using Gauss–Hermite quadrature. *Proc. IEEE*, *95*(5), 953–977.

Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking. *IEEE Trans. Signal Process.*, *50*(2), 174–188.

Battin, R. H., & Levine, G. M. (1970). Application of Kalman filtering techniques to the Apollo program. In C. T. Leondes (Ed.), *Theory and applications of Kalman filtering*. Neuilly sur Seine: NATO, Advisory Group for Aerospace Research and Development.

Beneš, V. E. (1981). Exact finite-dimensional filters for certain diffusions with nonlinear drift. *Stochastics*, *5*(1–2), 65–92.

Bensmaia, S. J., & Miller, L. E. (2014). Restoring sensorimotor function through intracortical interfaces: Progress and looming challenges. *Nat. Rev. Neurosci.*, *15*(5), 313–325.

Bishop, C. H., Etherton, B. J., & Majumdar, S. J. (2001). Adaptive sampling with the ensemble transform Kalman filter. Part I: Theoretical aspects. *Mon. Weather Rev.*, *129*(3), 420–436.

Bouton, C. E., Shaikhouni, A., Annetta, N. V., Bockbrader, M. A., Friedenberg, D. A., Nielson, D. M., . . . Rezai, A. R. (2016). Restoring cortical control of functional movement in a human with quadriplegia. *Nature*, *533*, 247–250.

Brandman, D. M., Burkhart, M. C., Kelemen, J., Franco, B., Harrison, M. T., & Hochberg, L. R. (2018). Robust closed-loop control of a cursor in a person with tetraplegia using gaussian process regression. *Neural Comput.*, *30*(11), 2986–3008.

Brandman, D. M., Cash, S. S., & Hochberg, L. R. (2017). Review: Human intracortical recording and neural decoding for brain-computer interfaces. *IEEE Trans. Neural Syst. Rehabil. Eng.*, *25*, 1687–1696.

Brandman, D. M., Hosman, T., Saab, J., Burkhart, M. C., Shanahan, B. E., Ciancibello, J. G., . . . Hochberg, L. R. (2018). Rapid calibration of an intracortical brain–computer interface for people with tetraplegia. *J. Neural Eng.*, *15*(2), 1–14.

Brown, R. G., & Hwang, P. Y. C. (2012). *Introduction to random signals and applied Kalman filtering*, 4th ed. Hoboken, NJ: Wiley.

Buehner, M., McTaggart-Cowan, R., & Heilliette, S. (2017). An ensemble Kalman filter for numerical weather prediction based on variational data assimilation: VarEnKF. *Mon. Weather Rev.*, *145*(2), 617–635.

Burkhart, M. C. (2019). *A discriminative approach to Bayesian filtering with applications to human neural decoding*. PhD diss., Brown University.

Butler, R. W. (2007). *Saddlepoint approximations with applications*. Cambridge: Cambridge University Press.

Cappé, O., Godsill, S. J., & Moulines, E. (2007). An overview of existing methods and recent advances in sequential Monte Carlo. *Proc. IEEE*, *95*(5), 899–924.

Cappé, O., Moulines, E., & Ryden, T. (2005). *Inference in hidden Markov models*. Berlin: Springer-Verlag.

Castillo, E., Guijarro-Berdiñas, B., Fontenla-Romero, O., & Alonso-Betanzos, A. (2010). A very fast learning method for neural networks based on sensitivity analysis. *J. Mach. Learn. Res.*, *7*, 1159–1182.

Cedarbaum, J. M., Stambler, N., Malta, E., Fuller, C., Hilt, D., Thurmond, B., & Nakanishi, A. (1999). The ALSFRS-R: A revised ALS functional rating scale that incorporates assessments of respiratory function. *J. Neurol. Sci.*, *169*(1), 13–21.

Chang, J. T., & Pollard, D. (1997). Conditioning as disintegration. *Stat. Neerl.*, *51*(3), 287–317.

Chen, Z. (2003). Bayesian filtering: From Kalman filters to particle filters, and beyond. *Statistics*, *182*(1), 1–69.

Choo, K., & Fleet, D. J. (2001). People tracking using hybrid Monte Carlo filtering. In *Proc. Int. Conf. Comput. Vis.* (vol. 2, pp. 321–328). Piscataway, NJ: IEEE.

Churchland, M. M., Cunningham, J. P., Kaufman, M. T., Foster, J. D., Nuyujukian, P., Ryu, S. I., & Shenoy, K. V. (2012). Neural population dynamics during reaching. *Nature*, *487*(7405), 1–20.

Collinger, J. L., Wodlinger, B., Downey, J. E., Wang, W., Tyler-Kabara, E. C., Weber, D. J., . . . Schwartz, A. B. (2013). High-performance neuroprosthetic control by an individual with tetraplegia. *Lancet*, *381*(9866), 557–564.

Daum, F. E. (1984). Exact finite dimensional nonlinear filters for continuous time processes with discrete time measurements. In *Proceedings of the IEEE Conf. Decis. Control* (pp. 16–22). Piscataway, NJ: IEEE.

Daum, F. E. (1986). Exact finite-dimensional nonlinear filters. *IEEE Trans. Autom. Control*, *31*(7), 616–622.

Daum, F. E., & Huang, J. (2003). Curse of dimensionality and particle filters. In *Proceedings of the 2003 IEEE Aerosp. Conf. Proc.* (vol. 4). Piscataway, NJ: IEEE.

del Moral, P. (1996). Nonlinear filtering using random particles. *Theory Probab. Appl.*, *40*(4), 690–701.

Douc, R., & Cappé, O. (2005). Comparison of resampling schemes for particle filtering. In *Proc. Int. Symp. Image and Signal Process. Anal.* (pp. 64–69). Piscataway, NJ: IEEE.

Doucet, A., Godsill, S., & Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Stat. Comput.*, *10*(3), 197–208.

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, *12*, 2121–2159.

Elliott, R. (1994). Exact adaptive filters for Markov chains observed in gaussian noise. *Automatica*, *30*(9), 1399–1408.

Evensen, G. (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *J. Geophys. Res: Oceans*, *99*, 10143–10162.

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *J. Exp. Pyschol.*, *47*(6), 381–391.

Flint, R. D., Lindberg, E. W., Jordan, L. R., Miller, L. E., & Slutzky, M. W. (2012). Accurate decoding of reaching movements from field potentials in the absence of spikes. *J. Neural Eng.*, *9*(4), 1–13.

Foresee, F. D., & Hagan, M. T. (1997). Gauss-Newton approximation to Bayesian learning. In *Proceedings of the Int. Conf. Neural Netw.* (3:1930–1935). Piscataway, NJ: IEEE.

Gelb, A. (1974). *Applied optimal estimation*. Cambridge, MA: MIT Press.

Georgopoulos, A. P., Kettner, R. E., & Schwartz, A. B. (1988). Primate motor cortex and free arm movements to visual targets in three-dimensional space. II. Coding of the direction of movement by a neuronal population. *J. Neurosci.*, *8*(8), 2928–2937.

Gerber, M., & Chopin, N. (2015). Sequential quasi Monte Carlo. *J. Roy. Stat. Soc. Ser. B (Stat. Methodol.)*, *77*(3), 509–579.

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural Comput.*, *12*(10), 2451–2471.

Ghahramani, Z., & Hinton, G. E. (2000). Variational learning for switching state-space models. *Neural Comput.*, *12*(4), 831–864.

Gilja, V., Pandarinath, C., Blabe, C. H., Nuyujukian, P., Simeral, J. D., Sarma, A. A., . . . Henderson, J. M. (2015). Clinical translation of a high-performance neural prosthesis. *Nat. Med.*, *21*(10), 1142–1145.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Int. Conf. Artif. Intell. Stats.* (9:249–256). PMLR.

Gordon, N. J., Salmond, D. J., & Smith, A. F. M. (1993). Novel approach to nonlinear/non-gaussian Bayesian state estimation. *IEE Proc. F—Radar and Signal Process.*, *140*(2), 107–113.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2016). LSTM: A search space odyssey. *IEEE Trans. Neural Netw. Learn. Syst.*, *28*(10), 1–11.

Grewal, M. S., & Andrews, A. P. (2010). Applications of Kalman filtering in aerospace 1960 to the present. *IEEE Control Syst. Mag.*, *30*(3), 69–78.

Hagan, M. T., & Menhaj, M. B. (1994). Training feedforward networks with the Marquardt algorithm. *IEEE Trans. Neural Netw.*, *5*(6), 989–993.

Hall, E. C. (1966). *Case history of the Apollo guidance computer*. Cambridge, MA: MIT Press.

Handschin, J. (1970). Monte Carlo techniques for prediction and filtering of non-linear stochastic processes. *Automatica*, *6*(4), 555–563.

Handschin, J. E., & Mayne, D. Q. (1969). Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering. *Int. J. Control*, *9*(5), 547–559.

Hess, R., & Fern, A. (2009). Discriminatively trained particle filters for complex multi-object tracking. In *Proceedings of Comput. Vis. Pattern Recognit.* (pp. 240–247). Piscataways, NJ: IEEE.

Hochberg, L. R., Bacher, D., Jarosiewicz, B., Masse, N. Y., Simeral, J. D., Vogel, J., . . . Donoghue, J. P. (2012). Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature*, *485*(7398), 372–375.

Hochberg, L. R., & Donoghue, J. P. (2006). Sensors for brain-computer interfaces. *IEEE Eng. Med. Biol. Mag.*, *25*(5), 32–38.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, *9*(8), 1735–1780.

Horn, R. A., & Johnson, C. R. (2013). *Matrix analysis*, 2nd ed. Cambridge: Cambridge University Press.

Hosman, T., Vilela, M., Milstein, D., Kelemen, J. N., Brandman, D. M., Hochberg, L. R., & Simeral, J. D. (2019). BCI decoder performance comparison of an LSTM recurrent neural network and a Kalman filter in retrospective simulation. In *Proceedings of the Int. IEEE EMBS Conf. Neural Eng.* Piscataway, NJ: IEEE.

Hunt, B. R., Kostelich, E. J., & Szunyogh, I. (2007). Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter. *Physica D: Nonlinear Phenom.*, *230*(1), 112–126.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In F. Bach, & D. Blei (Eds.), *Proceedings of the Int. Conf. Mach. Learn.*, vol. 37 (pp. 448–456). PMLR.

Ito, K. (2000). Gaussian filter for nonlinear filtering problems. In *Proceedings of the IEEE Conf. Decis. Control*, vol. 2. Piscataway, NJ: IEEE.

Ito, K., & Xiong, K. (2000). Gaussian filters for nonlinear filtering problems. *IEEE Trans. Autom. Control*, *45*, 910–927.

Jarosiewicz, B., Masse, N. Y., Bacher, D., Cash, S. S., Eskandar, E., Friehs, G., . . . Hochberg, L. R. (2013). Advantages of closed-loop calibration in intracortical brain-computer interfaces for people with tetraplegia. *J. Neural Eng.*, *10*(4), 1–17.

Jarosiewicz, B., Sarma, A. A., Bacher, D., Masse, N. Y., Simeral, J. D., Sorice, B., . . . Hochberg, L. R. (2015). Virtual typing by people with tetraplegia using a self-calibrating intracortical brain-computer interface. *Sci. Transl. Med.*, *7*(313), 1–11.

Jozefowicz, R., Zaremba, W., & Sutskever, I. (2015). An empirical exploration of recurrent network architectures. In F. Bach & D. Blei (Eds.), *Proceedings of the Int. Conf. Mach. Learn.*, vol. 37 (pp. 2342–2350).

Julier, S. J., & Uhlmann, J. K. (1997). New extension of the Kalman filter to nonlinear systems. *Proc. SPIE*, *3068*, 182–193.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *J. Basic Eng.*, *82*(1), 35–45.

Kalman, R. E., & Bucy, R. S. (1961). New results in linear filtering and prediction theory. *J. Basic Eng.*, *83*(1), 95–108.

Kim, S.-P., Simeral, J. D., Hochberg, L. R., Donoghue, J. P., & Black, M. J. (2008). Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia. *J. Neural Eng.*, *5*(4), 455–476.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *J. Comput. Graph. Stat.*, *5*(1).

Koyama, S., Pérez-Bolde, L. C., Shalizi, C. R., & Kass, R. E. (2010). Approximate methods for state-space models. *J. Am. Stat. Assoc.*, *105*(489), 170–180.

Kushner, H. (1967). Approximations to optimal nonlinear filters. *IEEE Trans. Autom. Control*, *12*(5), 546–556.

Lemon, R. N. (2008). Descending pathways in motor control. *Annu. Rev. Neurosci.*, *31*, 195–218.

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quart. Appl. Math.*, *2*, 164–168.

Liu, J. S. (2008). *Monte Carlo strategies in scientific computing*. Berlin: Springer.

MacKay, D. J. C. (1992). Bayesian interpolation. *Neural Comput.*, *4*(3), 415–447.

Majumdar, S. J., Bishop, C. H., Etherton, B. J., & Toth, Z. (2002). Adaptive sampling with the ensemble transform Kalman filter. Part II: Field program implementation. *Mon. Weather Rev.*, *130*(5), 1356–1369.

Malik, W. Q., Hochberg, L. R., Donoghue, J. P., Hochberg, L. R., Donoghue, J. P., & Brown, E. N. (2015). Modulation depth estimation and variable selection in state-space models for neural interfaces. *IEEE Trans. Biomed. Eng.*, *62*(2), 570–581.

Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *J. Soc. Indust. Appl. Math.*, *11*, 431–441.

Masse, N. Y., Jarosiewicz, B., Simeral, J. D., Bacher, D., Stavisky, S. D., Cash, S. S., . . . Donoghue, J. P. (2015). Non-causal spike filtering improves decoding of movement intention for intracortical BCIs. *J. Neurosci. Methods*, *244*, 94–103.

Maynard, E. M., Nordhausen, C. T., & Normann, R. A. (1997). The Utah intracortical electrode array: A recording structure for potential brain-computer interfaces. *Electroencephalogr. Clin. Neurophysiol.*, *102*(3), 228–239.

Metropolis, N., & Ulam, S. (1949). The Monte Carlo method. *J. Am. Stat. Assoc.*, *44*(247), 335–341.

Minka, T. P. (2001a). Expectation propagation for approximate Bayesian inference. *Proceedings of the Conf. Uncertain. Artif. Intell.* San Mateo, CA: Morgan Kaufmann.

Minka, T. P. (2001b). *A family of algorithms for approximate Bayesian inference*. PhD diss., MIT.

Nadaraya, E. A. (1964). On a regression estimate. *Teor. Verojatnost. i Primenen.*, *9*, 157–159.

Nørgaard, M., Poulsen, N. K., & Ravn, O. (2000). New developments in state estimation for nonlinear systems. *Automatica*, *36*(11), 1627–1638.

Nuyujukian, P., Albites Sanabria, J., Saab, J., Pandarinath, C., Jarosiewicz, B., Blabe, C. H., . . . Henderson, J. M. (2018). Cortical control of a tablet computer by people with paralysis. *PLOS One*, *13*(11).

Ott, E., Hunt, B. R., Szunyogh, I., Zimin, A. V., Kostelich, E. J., Corazza, M., . . . Yorke, J. A. (2004). A local ensemble Kalman filter for atmospheric data assimilation. *Tellus A*, *56*(5), 415–428.

Pandarinath, C., Gilja, V., Blabe, C. H., Nuyujukian, P., Sarma, A. A., Sorice, B. L., . . . Shenoy, K. V. (2015). Neural population dynamics in human motor cortex during movements in people with ALS. *eLife*, *4*.

Pandarinath, C., Nuyujukian, P., Blabe, C. H., Sorice, B. L., Saab, J., Willett, F., . . . Henderson, J. M. (2017). High performance communication by people with paralysis using an intracortical brain-computer interface. *eLife*, pp. 1–27.

Pandarinath, C., O'Shea, D. J., Collins, J., Jozefowicz, R., Stavisky, S. D., Kao, J. C., . . . Sussillo, D. (2018). Inferring single-trial neural population dynamics using sequential auto-encoders. *Nat. Methods*, *15*(10), 805–815.

Paninski, L., Fellows, M. R., Hatsopoulos, N. G., & Donoghue, J. P. (2004). Spatiotemporal tuning of motor cortical neurons for hand position and velocity spatiotemporal tuning of motor cortical neurons for hand position and velocity. *J. Clin. Neurophysiol.*, *91*, 515–532.

Pham, V., Bluche, T., Kermorvant, C., & Louradour, J. (2014). Dropout improves recurrent neural networks for handwriting recognition. In *Proceedings of the Int. Conf. Front. Handwriting Recognit.* (pp. 285–290). Piscataway, NJ: IEEE.

Pohlmeyer, E., Solla, S., Perreault, E. J., & Miller, L. E. (2007). Prediction of upper limb muscle activity from motor cortical discharge during reaching. *J. Neural Eng.*, *4*, 369–379.

Quang, P. B., Musso, C., & Le Gland, F. (2015). The Kalman Laplace filter: A new deterministic algorithm for nonlinear Bayesian filtering. In *Proceedings of the Intern. Conf. Inf. Fusion* (pp. 1566–1573). Piscataway, NJ: IEEE.

Quiñonero Candela, J., & Rasmussen, C. E. (2005). A unifying view of sparse approximate gaussian process regression. *J. Mach. Learn. Res.*, *6*, 1939–1959.

Rao, N. G., & Donoghue, J. P. (2014). Cue to action processing in motor cortex populations. *J. Neurophysiol.*, *111*(2), 441–453.

Rasmussen, C. E., & Nickisch, H. (2010). Gaussian processes for machine learning (GPML) toolbox. *J. Mach. Learn. Res.*, *11*, 3011–3015.

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. Cambridge, MA: MIT Press.

Real, E., Moore, S., Selle, A., Saxena, S., Suematsu, Y. L., Le, Q., & Kurakin, A. (2017). Large-scale evolution of image classifiers. In *Proceedings of the Int. Conf. Mach. Learn.* PMLR.

Särkkä, S. (2013). *Bayesian filtering and smoothing*. Cambridge: Cambridge University Press.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Netw.*, *61*, 85–117.

Schmidt, S. F., Weinberg, J. D., & Lukesh, J. S. (1970). Application of Kalman filtering to the C-5 guidance and control system. In C. T. Leondes (Ed.), *Theory and applications of Kalman filtering*. Neuilly sur Seine, NATO, Advisory Group for Aerospace Research and Development.

Schwartz, A. B. (1994). Direct cortical representation of drawing. *Science*, *265*(5171), 540–542.

Shumway, R. H., & Stoffer, D. S. (1991). Dynamic linear models with switching. *J. Am. Stat. Assoc.*, *86*(415), 763–769.

Simeral, J. D., Kim, S.-P., Black, M. J., Donoghue, J. P., & Hochberg, L. R. (2011). Neural control of cursor trajectory and click by a human with tetraplegia 1000 days after implant of an intracortical microelectrode array. *J. Neural Eng.*, *8*(2), 1–21.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, *15*, 1929–1958.

Stevenson, I. H., & Kording, K. P. (2011). How advances in neural recording affect data analysis. *Nat. Neurosci.*, *14*(2), 139–142.

Sugiyama, M., Suzuki, T., & Kanamori, T. (2012). *Density ratio estimation in machine learning*. Cambridge: Cambridge University Press.

Sussillo, D., Nuyujukian, P., Fan, J. M., Kao, J. C., Stavisky, S. D., Ryu, S., & Shenoy, K. (2012). A recurrent neural network for closed-loop intracortical brain–machine interface decoders. *J. Neural Eng.*, *9*(2), 1–21.

Sussillo, D., Stavisky, S. D., Kao, J. C., Ryu, S. I., & Shenoy, K. V. (2016). Making brain–machine interfaces robust to future neural variability. *Nat. Commun.*, *7*, 1–12.

van der Merwe, R. (2004). *Sigma-point Kalman filters for probabilistic inference in dynamic state-space models*. PhD diss., Oregon Health and Science University.

van der Vaart, A. W. (1998). *Asymptotic statistics*. Cambridge: Cambridge University Press.

Vargas-Irwin, C. E., Brandman, D. M., Zimmermann, J. B., Donoghue, J. P., & Black, M. J. (2015). Spike train SIMilarity space (SSIMS): A framework for single neuron and ensemble data analysis. *Neural Comput.*, *27*(1), 1–31.

Vargas-Irwin, C. E., Shakhnarovich, G., Yadollahpour, P., Mislow, J. M. K., Black, M. J., & Donoghue, J. P. (2010). Decoding complete reach and grasp actions from local primary motor cortex populations. *J. Neurosci.*, *30*(29), 9659–9669.

Velliste, M., Perel, S., Spalding, M. C., Whitford, A. S., & Schwartz, A. B. (2008). Cortical control of a prosthetic arm for self-feeding. *Nature*, *453*(7198), 1098–101.

Walker, B., & Kording, K. (2013). The database for reaching experiments and models. *PLOS One*, *8*(11).

Wan, E. A., & van der Merwe, R. (2000). The unscented Kalman filter for nonlinear estimation. In *Proceedings of the Adaptive Syst. for Signal Process., Commun., and Control Symp.* (pp. 153–158). Washington, DC: Society for Neuroscience.

Watson, G. S. (1964). Smooth regression analysis. *Sankhyā Ser. A*, *26*, 359–372.

Willett, F. R., Young, D. R., Murphy, B. A., Memberg, W. D., Blabe, C. H., Pandarinath, C., . . . Bolu Ajiboye, A. (2019). Principled BCI decoder design and parameter selection using a feedback control model. *Sci. Rep.*, *9*(8881).

Wodlinger, B., Downey, J. E., Tyler-Kabara, E. C., Schwartz, A. B., Boninger, M. L., & Collinger, J. L. (2015). Ten-dimensional anthropomorphic arm control in a human brain machine interface: Difficulties, solutions, and limitations. *J. Neural Eng.*, *12*(1), 1–17.

Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., & Vaughan, T. M. (2002). Brain-computer interfaces for communication and control. *Clin. Neurophysiol.*, *113*(6), 767–791.

Wu, W., Black, M. J., Gao, Y., Bienenstock, E., Serruya, M., & Donoghue, J. P. (2002). Inferring hand motion from multi-cell recordings in motor cortex using a Kalman filter. In *SAB'02-Workshop on Motor Control in Humans and Robots: On the Interplay of Real Brains and Artificial Devices* (pp. 66–73). Washington, DC: Society for Neuroscience.

Zaremba, W., Sutskever, I., & Vinyals, O. (2014). *Recurrent neural network regularization*. arXiv:1409.2329.

Zeiler, M. D. (2012). Adadelta: An adaptive learning rate method. arXiv:1212.5701.

Zoph, B., & Le, Q. V. (2017). Neural architecture search with reinforcement learning. In *Proceedings of the Int. Conf. Learn. Represent.* ICLR.

Received August 21, 2019; accepted December 24, 2019.