Title

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

Journal

Neural Computation, 32(5)

Authors

Burkhart, Michael

Brandman, David

Franco, Brian

et al.

Publication Date

2020-05-01

DOI

10.1162/neco_a_01275

Peer reviewed

HHS Public Access

Author manuscript

Neural Comput. Author manuscript; available in PMC 2021 July 06.

Published in final edited form as:

Neural Comput. 2020 May ; 32(5): 969–1017. doi:10.1162/neco_a_01275.

The Discriminative Kalman Filter for Bayesian Filtering with Nonlinear and Non-Gaussian Observation Models

Michael C. Burkhart[1], **David M. Brandman**[2,3], **Brian Franco**[4,†], **Leigh R. Hochberg**[4,5,6,7], **Matthew T. Harrison**[1]

[1]Division of Applied Mathematics, Brown University, Providence, RI, U.S.A.

[2]Department of Neuroscience, Brown University, Providence, RI, U.S.A.

[3]Department of Surgery (Neurosurgery), Dalhousie University, Halifax, NS, Canada.

[4]Center for Neurotechnology and Neurorecovery, Neurology, Massachusetts General Hospital, Boston, MA, U.S.A.

[5]School of Engineering and Carney Institute for Brain Science, Brown University, Providence, RI, U.S.A.

[6]Neurology, Harvard Medical School, Boston, MA, U.S.A.

[7]VA RR&D Center for Neurorestoration and Neurotechnology, Providence Veterans Affairs Medical Center, Providence, RI, U.S.A.

Abstract

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(observation|state) is nonlinear. We argue that in many cases a model for p(state|observation) proves both easier to learn and more accurate for latent state estimation.

Approximating p(state|observation) as Gaussian leads to a new filtering algorithm, the discriminative Kalman filter (DKF), that can perform well even when p(observation|state) is highly nonlinear and/or non-Gaussian. 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 non-Gaussian observation models than Kalman-based extensions.

When the observation model must be learned from training data prior to filtering, off-the-shelf nonlinear and/or 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

---

[†]current affiliation: NeuroPace, Inc., Mountain View, CA, U.S.A.

with $h_t = b_t + H_t v_t$, $C_t = M_t H_t^\top$, and $N_t = \Lambda_t + H_t M_t H_t^\top$

computer interface to provide real-time closed-loop cursor control to a person with a complete spinal cord injury. In this paper, we explore the theory underlying the DKF, exhibit some illustrative examples, and outline potential extensions.

Keywords

Bayesian filtering; discriminative learning; dynamic state-space models; neural decoding; the Kalman filter

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:

$$\begin{array}{ccccccccc} Z_1 & \longrightarrow & \cdots & \longrightarrow & Z_{t-1} & \longrightarrow & Z_t & \longrightarrow & \cdots & \longrightarrow & Z_T \\ \downarrow & & & & \downarrow & & \downarrow & & & & \downarrow \\ X_1 & & & & X_{t-1} & & X_t & & & & X_T \end{array} \quad (1)$$

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):

$$p(z_t|x_{1:t-1}) = \int p(z_t|z_{t-1})p(z_{t-1}|x_{1:t-1})dz_{t-1}, \quad (2a)$$

$$p(z_t|x_{1:t}) = \frac{p(x_t|z_t)p(z_t|x_{1:t-1})}{\int p(x_t|z_t)p(z_t|x_{1:t-1})dz_t} = \frac{p(x_t|z_t)p(z_t|x_{1:t-1})}{p(x_t|x_{1:t-1})}, \quad (2b)$$

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 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 and Levine, 1970; Grewal and Andrews, 2010; Buehner et al., 2017; Brown and Hwang, 2012; Schmidt et al., 1970; Brandman et al., 2017).

Exact solutions to Equation 2 are only available in special cases, such as the Kalman filter (Kalman, 1960; Kalman and 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 2 are analytically tractable but many model specifications still fall outside this class. When the latent state space is finite, the integrals in Equation 2 become sums that can be calculated exactly using a grid-based filter (Elliott, 1994; Arulampalam et al., 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 $\hat{p}(x_t \mid z_t)$, is substituted into whatever algorithm is used to approximate Bayes' rule in Equation 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 paper, 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 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 et al., 2018b,a), 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 and/or non-Gaussian 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 $A\top$ 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 $S, \Gamma \in \mathbb{S}_d$,

$$p(z_0) = \eta_d(z_0; 0, S), \tag{3a}$$

$$p(z_t | z_{t-1}) = \eta_d(z_t; Az_{t-1}, \Gamma), \tag{3b}$$

for $t = 1, 2, \ldots$, where $S = ASA^\top + \Gamma$ so that the process is stationary. Note that Equation 3 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 non-Gaussian, multimodal, discrete, etc. 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,

$$p(z_t | x_t) \approx \eta_d(z_t; f(x_t), Q(x_t)), \tag{4}$$

where $f : \mathcal{X} \to \mathbb{R}^d$ and $Q : \mathcal{X} \to \mathbb{S}_d$. Note that Equation 4 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 $\left(\mathbb{R}^d\right)$, the Bernstein–von Mises Theorem states that there exists f and Q 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,

$$f(x) = \mathbb{E}(Z_t | X_t = x), \quad Q(x) = \mathbb{V}(Z_t | X_t = x), \tag{5}$$

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

To make use of Equation 4 for approximating Equation 2, we first rewrite Equation 2b in terms of $p(z_t|x_t)$ as

$$\begin{aligned} p(z_t | x_{1:t}) &= \frac{p(x_t)}{p(x_t | x_{1:t-1})} \frac{p(z_t | x_t)}{p(z_t)} p(z_t | x_{1:t-1}), \\ &= \frac{p(x_t)}{p(x_t | x_{1:t-1})} \frac{p(z_t | x_t)}{p(z_t)} \int p(z_t | z_{t-1}) p(z_{t-1} | x_{1:t-1}) dz_{t-1}, \end{aligned} \tag{6}$$

where the second line follows from the the Chapman–Kolmogorov equation (Equation 2a). We then substitute the latent state model (Equation 3) and the DKF approximation (Equation

4) into Equation 6. We absorb terms not depending on z_t into a normalizing constant κ to obtain

$$p(z_t|x_{1:t}) \approx \kappa(x_{1:t}) \frac{\eta_d(z_t; f(x_t), Q(x_t))}{\eta_d(z_t; 0, S)} \int \eta_d(z_t; Az_{t-1}, \Gamma) p(z_{t-1}|x_{1:t-1}) dz_{t-1} \quad (7)$$

.

If $p(z_{t-1}|x_{1:t-1})$ is approximately Gaussian, which it is for the base case of $t = 1$ from Equation 3a (defining $p(z_0|x_{1:0}) = p(z_0)$), then all of the terms on the right side of Equation 7 are approximately Gaussian. If these approximations are exact and the analytic expression for covariance is valid (specifically if Σ_t in Equation 9 is positive definite), we find that the right side of Equation 7 is again Gaussian, giving a Gaussian approximation for $p(z_t|x_{1:t})$. We rely on the fact that dividing two Gaussian pdf's 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 pdf's is Gaussian, without any additional assumptions). See the proof of Lemma 1 in Appendix 7 for a full derivation and further details.

Let

$$p(z_t|x_{1:t}) \approx \eta_d(z_t; \mu_t(x_{1:t}), \Sigma_t(x_{1:t})) \quad (8)$$

be the Gaussian approximation of $p(z_t|x_{1:t})$ obtained from successively applying the approximation in Equation 7. 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

$$\boxed{v_t = A\mu_{t-1}, M_t = A\Sigma_{t-1}A^\top + \Gamma, \Sigma_t = \left(M_t^{-1} + Q(x_t)^{-1} - S^{-1}\right)^{-1}, \mu_t = \Sigma_t\left(M_t^{-1}v_t + Q(x_t)^{-1}f(x_t)\right).} \quad (9)$$

The first two steps incorporate the exact state dynamics in Equation 3b and the final two steps incorporate the observation information using the DKF approximation in Equation 4. 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 Appendix 6.3.

Equation 9 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, then the DKF and the Kalman filter are equivalent (*cf.* 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 4. 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 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.

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 3; 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 $M_t = A\Sigma_{t-1}A^\top + \Sigma$;

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

One concern is that Equation 6 will amplify approximation errors. Along these lines, we prove the following result that holds whenever the BvM Theorem is applicable for Equation 4:

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 7. 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 6 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 the theorem above still holds when the $p(z_t)$ term is omitted from the denominator of Equation 6 (see Remark 3 in Appendix 7).

2.3 Robust DKF

Omitting the $p(z_t)$ from the denominator of Equation 6 is also helpful for making the DKF robust to violations of the modeling assumptions and to 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

$$\boxed{v_t = A\mu_{t-1}, M_t = A\,\Sigma_{t-1}\,A^{\top} + \Gamma\,,\ \Sigma_t = \left(M_t^{-1} + Q(x_t)^{-1}\right)^{-1}, \mu_t = \Sigma_t\left(M_t^{-1}v_t + Q(x_t)^{-1}f(x_t)\right).} \quad (10)$$

(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 7 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 $p(z_t|x_{1:t})$, which is 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 the first step uses the exact state dynamics (Equation 3b) to create a Gaussian approximation for $p(z_t|x_{1:t-1})$, namely,

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

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 3; 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 $M_t = A\ \Sigma_{t-1}\ A^\top + \Gamma$;

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

$$p(z_t | x_{1:t-1}) \approx \eta_d(z_t; \nu_t, M_t), \tag{11}$$

where $\nu_t = A\mu_{t-1}$ and $M_t = A\ \Sigma_{t-1}\ A^\top + \Gamma$, as in Equations 9 and 10. 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 11) 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

$$\eta_d(z_t; \mu_t,\ \Sigma_t\ ) \approx \frac{p(x_t | z_t) \eta_d(z_t; \nu_t, M_t)}{\int p(x_t | z_t) \eta_d(z_t; \nu_t, M_t) dz_t} = q_t(z_t), \tag{12}$$

where q_t is defined by this equation.

There are many strategies in the literature for choosing μ_t and Σ_t in Equation 12. 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 12 (Kushner, 1967; Ito, 2000; Ito and 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 only practical if the mean and covariance of q_t are available. In particular, we must be able to efficiently compute or easily approximate the integrals

$$\begin{aligned} a &= \int p(x_t \mid z_t)\eta_d(z_t; \nu_t, M_t)dz_t, \\ b &= \int z_t p(x_t \mid z_t)\eta_d(z_t; \nu_t, M_t)dz_t, \\ c &= \int z_t z_t^\top p(x_t \mid z_t)\eta_d(z_t; \nu_t, M_t)dz_t, \end{aligned} \tag{13}$$

to obtain $\mu_t = b/a$ and $\Sigma_t = c/a - \mu_t\mu_t^\top$. There also exist extensions of the G-ADF. For instance, expectation propagation uses iterative refinement of estimates to improve upon assumed density filtering (Minka, 2001b,a). 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 ν_t and M_t are replaced by 0 and S, respectively, throughout Equation 13 (and then $f(x_t) = b/a$ and $Q(x_t) = c/a - f(x_t)f(x_t)^\top$). 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, $\eta_d(z_t; \nu_t, M_t)$. The example in Section 4.2 below 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 12 into a Gaussian form as a function of z_t (Butler, 2007; Koyama et al., 2010; Quang et al., 2015). Defining

$$g_t(z_t) = \log(p(x_t \mid z_t)\eta_d(z_t; \nu_t, M_t)) \quad \text{and} \quad z_t^* = \operatorname{argmax}_{z_t} g_t(z_t),$$

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

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

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

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

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 12 as jointly Gaussian in (z_t, x_t), namely,

$$p(x_t \mid z_t)\eta_d(z_t; \nu_t, M_t) \approx \eta_{d+n}\left(\begin{pmatrix} z_t \\ x_t \end{pmatrix}; \begin{pmatrix} \nu_t \\ h_t \end{pmatrix}, \begin{pmatrix} M_t & C_t \\ C_t^\top & N_t \end{pmatrix}\right), \tag{14}$$

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 12 to be exactly integrated to obtain

$$\mu_t = \nu_t + C_t N_t^{-1}(x_t - h_t) \quad \text{and} \quad \Sigma_t = M_t - C_t N_t^{-1} C_t^\top. \tag{15}$$

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 14 can be rewritten as

$$p(x_t \mid z_t) \approx \eta_n(x_t; b_t + H_t z_t, \Lambda_t). \tag{16}$$

The implicit linearization in Equation 14 is now explicit: $\mathbb{E}(X_t \mid Z_t = z_t)$ is approximated as the linear function $b_t + H_t z_t$. The relationship between the different parameters in Equations 14 and 16 is $b_t = h_t - C_t^\top M_t^{-1} v_t$, $H_t = C_t^\top M_t^{-1}$, and $\Lambda_t = N_t - C_t^\top M^{-1} C_t$. Upon re-parameterization, Equation 15 can be used for filtering with

$$\Sigma_t = \left(M_t^{-1} + H_t^\top \Lambda_t^{-1} H_t\right)^{-1},$$

$$\mu_t = \Sigma_t \left(M_t^{-1} \nu_t + H_t^\top \Lambda_t^{-1} (x_t - b_t)\right),$$

which has a similar appearance to the corresponding DKF updates in Equation 9.

Equation 16 underlies several Gaussian approximations to Bayes' rule, including the approximations used in the extended Kalman filter (EKF), the unscented Kalman filter (UKF: Julier and Uhlmann, 1997; Wan and 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 15. 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 and Xiong, 2000; Nørgaard et al., 2000), the Gauss–Hermite Kalman Filter, the Quadrature Kalman filter (Ito, 2000; Ito and Xiong, 2000) and the Cubature Kalman filter (Arasaratnam et al., 2007; Arasaratnam and 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 $\overline{H}_t = \int (h(z_t) - \bar{h}_t)(z_t - \nu_t)^\top \eta_d(z_t; \nu_t, M_t) dz_t$, the solution is $B_t = \overline{H}_t M_t^{-1}$, 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 $\overline{H}_t$ can be efficiently computed or easily approximated.

The UKF, the SLF, and many related techniques improve upon 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 below 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

As described in the previous section, 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 2a and the second step uses Bayes' rule to update $Z_t|X_{1:t-1}$ into $Z_t|X_{1:t}$ via Equation 2b. In this paper, difficulties with the first step are removed by assuming linear, Gaussian, state dynamics (Equation 3). 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 (Equation 9) or the robust DKF algorithm (Equation 10), instead of computing them in the first two lines of these algorithms. In this paper, 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 below.

There is a vast literature on more general approximation algorithms for Bayesian filtering (Särkkä, 2013; Chen, 2003). Monte Carlo integration (Metropolis and 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 and Mayne, 1969; Handschin, 1970; Gordon et al., 1993; Kitagawa, 1996; del Moral, 1996; Doucet et al., 2000; Cappé et al., 2005, 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 and Huang, 2003). Alternate sampling strategies (see, e.g., Chen, 2003; Liu, 2008) can be used to improve filter performance, including: acceptance-rejection sampling (Handschin and Mayne, 1969), stratified sampling (Douc and Cappé, 2005), hybrid MC (Choo and Fleet, 2001), and quasi-MC (Gerber and 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 et al., 2001; Majumdar et al., 2002), along with versions that produce local, parallelizable approximations for covariance (Ott et al., 2004; Hunt et al., 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

$$p(x_t|z_t) = p(x_t)\frac{p(z_t|x_t)}{p(z_t)} \approx \kappa(x_t)\frac{\eta_d(z_t; f(x_t), Q(x_t))}{\eta_d(z_t; 0, S)}. \quad (17)$$

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 17 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 17 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 et al., 2012). Another strategy might be to use the robust DKF approximation as in Section 2.3, which simply drops the denominator in Equation 17. 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 is 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 and denoted $(z'_1, x'_1), \ldots, (z'_m, x'_m)$. This training data might be real data, or it 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 only consider 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 9. 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 3a–3b. In the learning experiments below we learn them from (z'_{t-1}, z'_t) pairs using only Equation 3b, 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, for example, 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 4. One possibility, and the one we focus on here, is to define f and Q via Equation 5 and then learn them directly from training data as

$$\hat{f}(x) \approx f(x) = \mathbb{E}(Z_t | X_t = x) \quad \text{and} \quad \hat{Q}(x) \approx Q(x) = \mathbb{V}(Z_t | X_t = x). \tag{18}$$

Using Equation 18, 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 heteroscedastic 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 below, 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 the Appendix. 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

$$Q(x) = \mathbb{V}(Z_t | X_t = x) = \mathbb{E}\left(R_t R_t^\top \middle| X_t = x\right) \approx \mathbb{E}\left(\hat{R}_t \hat{R}_t^\top \middle| X_t = x\right). \tag{19}$$

The final expression in Equation 19 is a conditional expectation and can in principle be learned with regression on $\left(\hat{R}_t \hat{R}_t^\top, X_t\right)$ 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 $\hat{R}_t \hat{R}_t^\top$ 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 below. 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

$$\hat{Q}(x) = \frac{\sum_{i=1}^k \hat{r}_i \hat{r}_i^\top \kappa(x, x_i'')}{\sum_{i=1}^k \kappa(x, x_i'')}, \tag{20}$$

for a kernel $\kappa: \mathcal{X} \times \mathcal{X} \to [0, \infty)$. Complete details are in the Appendix.

4 Examples

In this section, we compare filter performance on both artificial models and on real neural data. Corresponding MATLAB code (and Python code for the 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

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

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 5 gives

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

It is straightforward to verify that the DKF in Equation 9 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 (Section 2.4). In particular, the specific observation model used in the simulation below 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 (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 $\boldsymbol{\pi} = \boldsymbol{\pi}_{1:L}$, 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 $\boldsymbol{\pi}$ 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 and Stoffer, 1991; Ghahramani and Hinton, 2000). The integrals in Equation 13 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 Appendix 6.1 for details), although the DKF is much faster for large n, because it allows for more pre-computation. 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 $\overline{H} = \sum_\ell \pi_\ell H_\ell$ so that

$$\mathbb{E}(X_t|Z_t) = \bar{b} + \overline{H} Z_t. \tag{21}$$

An interesting special case of this model is when $\overline{H} = 0$, 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 only found 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 $\overline{H} = 0$,

illustrating the important difference between a Gaussian approximation for the observation model and the DKF approximation in Equation 4. The simulation in Figure 1 used $\overline{H} = 0$, 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 (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,

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

for a probability vector $\boldsymbol{\pi} = \boldsymbol{\pi}_{1:L}$. 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

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

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 $Z_{t d_i} < \gamma_i$ or not. 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 Appendix 6.2. In general, however, the integrals in Equation 13 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.

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

$$\mathbb{E}(X_{ti} \mid Z_t) = \mathbb{P}(X_{ti} = 1 \mid Z_t) = \bar{g}_i(Z_t). \tag{22}$$

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, then X_{ti} and Z_t are independent, i.e., 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 we modify the state dynamics in Equation 3 as follows. Define the 2×2 rotation matrix $R(\theta) = \begin{pmatrix} \sin\theta & \cos\theta \\ -\cos\theta & \sin\theta \end{pmatrix}$ 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, namely,

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

Define the function $a: \mathbb{R}^{d \times 1} \to \mathbb{R}^{d \times 1}$ via $a(z) = AR_d(|z|)z$, where $|\cdot|$ denotes the Euclidean norm. The new state dynamics are

$$p(z_0) = \eta_d(z_0; 0, S),$$

$$p(z_t | z_{t-1}) = \eta_d(z_t; a(z_{t-1}), \Gamma),$$

for $t = 1, 2, \ldots$, where $A = ASA^\top + \Gamma$. 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), then Z_t remains marginally Gaussian, which is an important part of the DKF approximation.

We use an unscented Kalman filter (UKF) approximation for the state dynamics; i.e., we replaced ν_t and M_t in Equations 9 and 10 with the mean and covariance obtained from performing the unscented transform (Julier and 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, non-Gaussian 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.

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 et al. (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 this 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 100ms bins. The first $n = 10$ principal 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 50ms 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 20. 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

$$h(z) = \mathbb{E}(X_t \mid Z_t = z) \tag{23}$$

via neural network regression, and took the conditional covariance to be constant, namely, $\Lambda(z) = \mathbb{V}(X_t \mid Z_t = z) \equiv \Lambda \in \mathbb{S}_{10}$, which we learned from the approximate residuals. Finally, we also experimented with a Long Short Term Memory (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 Appendices 6.4–6.7.

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 et al., 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 23. 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 and 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 et al., 2002; Hochberg and Donoghue, 2006; Brandman et al., 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 et al., 2018b).

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 et al., 1988), velocity (Schwartz, 1994), acceleration (Paninski et al., 2004), muscle activation (Lemon, 2008; Pohlmeyer et al., 2007), proprioception (Bensmaia and Miller, 2014), visual information related to the task (Rao and Donoghue, 2014) and preparatory activity (Churchland et al., 2012). Hence, iBCI-related recordings are highly complex and non-linear (Vargas-Irwin et al., 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 20ms (Jarosiewicz et al., 2015) or even 1ms (Pandarinath et al., 2015). Thus, any potential filtering technique must be computationally feasible to implement for real-time use.

Third, over the past decades, 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 and Kording, 2011). By contrast, the dimensionality of the underlying device being controlled remains small, generally not exceeding ten 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 et al., 2018b,a). 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 is 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 et al., 2008; Simeral et al., 2011). Data was used from trial (post-implant) days 292 and 293.

4.6.2 Signal acquisition—Raw neural signals for each channel (electrode) were sampled at 30kHz using the Neuro-Port 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 15kHz for decoding and de-noised 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 one minute reference block run at the start of each session). The de-noised signal was band-pass filtered between 250 Hz and 5000 Hz using an 8th order non-causal Butterworth filter (Masse et al., 2015). Spike events were triggered by crossing a threshold set at 3.5x the root mean square amplitude of each channel, as determined by data from the reference block. The neural feature used was the the total power in the band-pass filtered signal (Jarosiewicz et al., 2015; Brandman et al., 2018b). Neural features were binned in 20ms non-overlapping 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 55–60 cm at a comfortable angle and orientation to T9.

Targets (size = 2.4 cm, visual angle = 2.5°) were presented sequentially in a pseudo-random 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 = 1.5cm, visual angle = 1.6°) within the target's diameter for 300ms, before a pre-determined 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 two minute 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 two minutes, initial hyperparameters for the GP were learned. Next, T9 acquired targets for three 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 et al., 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 a total of 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

$$\mathrm{MT} = a \cdot \mathrm{ID} + b \tag{24}$$

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 = 1.6cm, 3.5cm, and 5.6cm, visual angles 1.7°, 3.7°, and 5.8°). Targets were acquired by having the cursor contact the target for 500ms milliseconds, 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, slope = 1.08 ± 0.06, $p < 1.2 \times 10^{-30}$, intercept = 1.6 ± 1.3, $p < 2.2 \times 10^{-41}$).

5 Discussion

The DKF is a novel filtering method that should prove a helpful addition to the filtering toolbox. It provides a fast, analytic approximation for models with linear, Gaussian dynamics, but nonlinear, non-Gaussian 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 which 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 and/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 et al., 2005; Hess and 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 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 et al. (2018a) 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 17, 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.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

The authors would like to 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 (ECOR) 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 paper is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health, the Department of Veterans Affairs, or the United States Government.

Appendix

Section 6 covers technical details and section 7 includes a proof of the main theorem.

6 Technical details

This section 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.

6.1 Kalman observation mixtures

For the model in Section 4.2 we provide analytic expressions for the integrals in Equation 13, which are needed for the G-ADF and the DKF (using $\nu_t = 0$ and $M_t = S$ for the DKF). Define

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

$$y_{t\ell} = U_{t\ell}\left(M_t^{-1}\nu_t + H_\ell^\top \Lambda_\ell^{-1}(x_t - b_\ell)\right),$$

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

Then

$$a = \int p(x_t|z_t)\eta_d(z_t;\nu_t,M_t) = \sum_{\ell=1}^{L}\pi_\ell \int \eta_n(x_t;b_\ell + H_\ell z_t,\ \Lambda_\ell)\eta_d(z_t;\eta_t,M_t)dz_t$$
$$= \sum_{\ell=1}^{L}\pi_\ell \eta_n\left(x_t;b_\ell + H_\ell \nu_t,\ \Lambda_\ell + H_\ell M_t H_\ell^\top\right),$$

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

$$c = \int z_t z_t^\top p(x_t|z_t)\eta_d(z_t;\nu_t,M_t)dz_t = \sum_{\ell=1}^{L}\pi_\ell\kappa_{t\ell}\int z_t z_t^\top \eta_d(z_t;y_{t\ell},U_{t\ell})dz_t$$
$$= \sum_{\ell=1}^{L}\pi_\ell\kappa_{t\ell}\left(U_{t\ell} + y_{t\ell}y_{t\ell}^\top\right).$$

6.2 Independent Bernoulli mixtures

For the model in Section 4.3 we provide analytic expressions for the integrals in Equation 13 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 $\phi(v) = \int_{-\infty}^{v}\eta(u)$ to denote the corresponding distribution function, define

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

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

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

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

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

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

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

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

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

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

$$\begin{aligned} a &= \int p(x_t | z_t) \eta_d(z_t; 0, S) dz_t = \sum_{\ell=1}^{L} \pi_\ell \prod_{k=1}^{d} \int p_\ell(x_t N_k | z_{tk}) \eta(z_{tk}) dz_{tk} \\ &= \sum_{\ell=1}^{L} \pi_\ell \prod_{k=1}^{d} \sum_{j=1}^{n_k + 1} D_{\ell kj}(x_t N_k) \int_{\gamma_{k,j-1}}^{\gamma_{k,j}} \eta(z_{tk}) dz_{tk} \\ &= \sum_{\ell=1}^{L} \pi_\ell \prod_{k=1}^{d} \sum_{j=1}^{n_k + 1} D_{\ell kj}(x_t N_k) \Phi_{kj}, \end{aligned}$$

$$\begin{aligned} b_r &= \int z_{tr} p(x_t | z_t) \eta_d(z_t; 0, S) dz_t \\ &= \sum_{\ell=1}^{L} \pi_\ell \prod_{k=1}^{d} \sum_{j=1}^{n_k + 1} D_{\ell kj}(x_t N_k) \left(\Phi_{kj} + \Phi'_{kj} \delta_{kr} \right), \end{aligned}$$

$$\begin{aligned} c_{rs} &= \int z_{tr} z_{ts} p(x_t | z_t) \eta_d(z_t; 0, S) dz_t \\ &= \sum_{\ell=1}^{L} \pi_\ell \prod_{k=1}^{d} \sum_{j=1}^{n_k + 1} D_{\ell kj}(x_t N_k) \left(\Phi_{kj} + \Phi'_{kj} \delta_{kr} + \Phi'_{kj} \delta_{ks} + \Phi''_{kj} \delta_{kr} \delta_{ks} \right), \end{aligned}$$

where in Equation 13 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 $Q(x) = c/a - f(x) f(x)^\top$.

6.3 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 $S^{-1/2}(S - Q)S^{-1/2} = S{-}1/2(S - SVDV{-}1)S{-}1/2 = I - S1/2\,V\,D(S1/2\,V)^{-1} = (S1/2\,V)(I - D)(S1/2\,V)^{-1}$, 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 paper, we report this modification in case others find it useful in their application.

6.4 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.

6.5 Neural network regression

We can also learn f as a neural network (NN). NN's are attractive for online filtering, because evaluation of $\hat{f}$ scales $\mathcal{O}(1)$ with the size of the training set. With mean squared error (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 and 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 $\mathcal{O}(m^2)$; Levenberg–Marquardt, with $\mathcal{O}(m^3)$ (Castillo et al., 2010), where m is the size of the training set. More recently, Hessian-free approaches have been developed to train NN's on larger data sets (Schmidhuber, 2015). Training costs also grow with d, depending on 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 and Menhaj, 1994) with Bayesian regularization.

6.6 Gaussian process regression

Gaussian process (GP) regression is another popular method for nonlinear regression (Rasmussen and 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 will 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 a $m \times d$-dimensional matrix Z' by concatening the vectors Z_i'. We assume that $p(z_i' \mid x_i', f) = \eta\left(z_i'; f(x_i'), \sigma^2\right)$, where f is sampled from a mean-zero GP with covariance kernel $K(\cdot, \cdot)$. Under this model,

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

where $K(x,X')$ denotes the $1 \times m$ vector with ith entry $K(x, X_i')$, where $K(X',X')$ denotes the $m \times m$ matrix with ijth entry $K\left(X_i', X_j'\right)$, where Z' is a column vector, and where 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 $\mathcal{O}\left(m^3\right)$. Sparse approximations to GP's can reduce training requirements to $\mathcal{O}\left(m \cdot N_S^2\right)$ where N_S is the size of the sparse GP (Quiñonero Candela and 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 and Nickisch, 2010).

6.7 Comparison with a long short term memory (LSTM) neural network

An LSTM is a stateful recurrent neural network designed to overcome error backflow problems (Hochreiter and 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 et al., 2012, 2016; 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 et al., 2015; Greff et al., 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 et al. (2000). Dropout was used to prevent overfitting (Srivastava et al., 2014), but it was only applied to feedforward connections, not recurrent connections (Pham et al., 2014; Zaremba et al., 2014). The recurrent states and outputs at each intermediate time step were batch-normalized to accommodate internal covariate shift (Ioffe and Szegedy, 2015). Model parameters were initialized via a Xavier-type method (Glorot and 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 et al., 2011) with the explicit goals of decreasing sensitivity to hyperparameters and permitting the learning rate to sometimes increase.

7 Mathematical results

Our main technical result is Theorem 2 below. After stating the theorem we translate it into the setting of the paper. 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 $S = ASA^\top + \Gamma$.

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

$$p_n = \frac{u_n \tau s_n / p}{\|u_n \tau s_n / p\|_1} \tag{25}$$

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–C5. Rather, we are trying to show that the specific* p'_n *defined in C3 satisfies C4–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 26, *each of which tend 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–A5 hold in probability. Then the*

probability that p'_n is a well-defined, Gaussian pdf converges to one, and the limits in C1–C5 hold in probability.

For the setting of the paper, 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, $X_{1:t}^{1:n} = \left(X_1^{1:n}, \ldots, X_t^{1:n}\right)$. We will abuse notation and use $\mathbb{P}(Z_t = \cdot \mid W)$ 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 and Pollard, 1997). Note that we are in the setting of Remark 4, where the randomness comes from $X_{1:b}Z_{1:t}$. With this in mind, define

$$u_n(\cdot) = u_n\left(\cdot; X_t^{1:n}\right) = \mathbb{P}\left(Z_t = \cdot \,\middle|\, X_t^{1:n}\right)$$

$$u'_n(\cdot) = u'_n\left(\cdot; X_t^{1:n}\right) = \eta_d\left(\cdot; f_n\left(X_t^{1:n}\right), Q_n\left(X_t^{1:n}\right)\right)$$

$$s_n(\cdot) = s_n\left(\cdot; X_{1:t-1}^{1:n}\right) = \mathbb{P}\left(Z_{t-1} = \cdot \,\middle|\, X_{1:t-1}^{1:n}\right) \quad (t > 1)$$

$$s'_n(\cdot) = s'_n\left(\cdot; X_{1:t-1}^{1:n}\right) = \eta_d\left(\cdot; \mu_{t-1,n}\left(X_{1:t-1}^{1:n}\right), \sum_{t-1,n}\left(X_{1:t-1}^{1:n}\right)\right) \quad (t > 1)$$

$$p_n(\cdot) = p_n\left(\cdot; X_{1:t}^{1:n}\right) = \mathbb{P}\left(Z_t = \cdot \,\middle|\, X_{1:t}^{1:n}\right)$$

$$p'_n(\cdot) = p'_n\left(\cdot; X_{1:t}^{1:n}\right) = \eta_d\left(\cdot; \mu_{t,n}\left(X_{1:t}^{1:n}\right), \sum_{t,n}\left(X_{1:t}^{1:n}\right)\right)$$

$$b = Z_t$$

$$P(\cdot) = P(\cdot; Z_{t-1}) = \delta_{Z_{t-1}} \quad (t > 1),$$

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 25 above is simply a condensed version of Equation 6 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–A4 hold in probability. We refer the reader 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 iid 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–A2 (in probability). For the case $t = 1$, we have $s_n \equiv s'_n \equiv P \equiv p$, so A1–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–C5 in the case $t - 1$ become the assumptions A1–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

$$\begin{aligned}\|p_n - p'_n\|_1 \leq \underbrace{\left\| p_n - \frac{p_n p}{p(b)} \right\|_1}_{A_n} + \underbrace{\left\| \frac{p_n p}{p(b)} - \frac{p_n p}{\|p_n p\|_1} \right\|_1}_{B_n} \\ + \underbrace{\left\| \frac{p_n p}{\|p_n p\|_1} - \frac{p'_n p}{\|p'_n p\|_1} \right\|_1}_{C_n} + \underbrace{\left\| \frac{p'_n p}{\|p'_n p\|_1} - \frac{p'_n p}{p(b)} \right\|_1}_{B'_n} + \underbrace{\left\| \frac{p'_n p}{p(b)} - p'_n \right\|_1}_{A'_n}.\end{aligned} \tag{26}$$

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

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

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

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

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

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

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

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

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

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

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

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

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 $G_n = AV_nA^\top + \Gamma$, $T_n = \left(U_n^{-1} + G_n^{-1} - S^{-1}\right)^{-1}$, *and* $c_n = T_n\left(U_n^{-1} b_n + G_n^{-1} A a_n\right)$, *as long as* T_n *is well-defined and positive definite. Furthermore, if* $s_n' \xrightarrow{w} P$, *then* $u_n' \xrightarrow{w} \delta_b$ *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

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

and

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

for $U_n' = \left(U_n^{-1} - S^{-1}\right)$ and $b_n' = U_n' U_n^{-1} b_n$, we have

$$\begin{aligned}p_n'(z) &\propto \eta_d(z; b_n', U_n')\eta_d(z; Aa_n, G_n) \\ &\propto \eta_d\left(z; T_n\left((U_n')^{-1}b_n' + G_n^{-1}Aa_n\right), T_n\right) \\ &= \eta_d(z; c_n, T_n)\,.\end{aligned}$$

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

Now, suppose additionally that $s_n' \xrightarrow{w} P$ and $u_n' \xrightarrow{w} \delta_b$. Consider the characteristic functions

$$\phi_{s_n'}(t) = \mathbb{E}_{X\sim s_n'}\left[e^{itX}\right] = e^{it^\top a_n - \frac{1}{2}t^\top U_n t}$$

for these random variables. Lévy's continuity theorem (Thm. 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

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

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

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

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, $G_n \to G = AV\,A^\top + \Gamma$, which is invertible, since Γ is positive definite, and so $G_n^{-1} \to G^{-1}$.

The Woodbury matrix identity gives

$$T_n = \left(U_n^{-1} + G_n^{-1} - S^{-1}\right)^{-1} = U_n - U_n\left(\left(G_n^{-1} - S^{-1}\right)^{-1} + U_n\right)^{-1} U_n. \tag{27}$$

Since $U_n \to 0_{d\times d}$ and $\left(\left(G_n^{-1} - S^{-1}\right)^{-1} + U_n\right)^{-1} \to G^{-1} - S^{-1}$, 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 Cor. 6.3.8 in Horn and Johnson, 2013), it follows that

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

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 independently 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 $\lambda_j\left(U_n^{-1}\right) \geq \lambda_d\left(U_n^{-1}\right) = 1/\lambda_1(U_n) \to \infty$. 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 27 for T_n gives

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

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

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

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

References

Abbeel P, Coates A, Montemerlo M, Ng AY, and Thrun S (2005). Discriminative training of Kalman filters. In Robotics: Sci. and Syst. (RSS)

Ajiboye AB, Willett FR, Young DR, Memberg WD, Murphy BA, Miller JP, Walter BL, Sweet JA, Hoyen HA, Keith MW, Peckham PH, Simeral JD, Donoghue JP, Hochberg LR, and Kirsch RF (2017). Restoration of reaching and grasping movements through brain-controlled muscle stimulation in a person with tetraplegia: a proof-of-concept demonstration. The Lancet, 389:1821–1830.

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

Arasaratnam I, Haykin S, and Elliott RJ (2007). Discrete-time nonlinear filtering algorithms using Gauss–Hermite quadrature. Proc. IEEE, 95(5):953–977.

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

Battin RH and Levine GM (1970). Application of Kalman filtering techniques to the Apollo program. In Leondes CT, editor, Theory and Applications of Kalman Filtering, chapter 14 NATO, Advisory Group for Aerospace Research and Development.

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

Bensmaia SJ and Miller LE (2014). Restoring sensorimotor function through intracortical interfaces: progress and looming challenges. Nat. Rev. Neurosci, 15(5):313–325. [PubMed: 24739786]

Bishop CH, Etherton BJ, and Majumdar SJ (2001). Adaptive sampling with the ensemble transform Kalman filter. part I: Theoretical aspects. Mon. Weather Rev, 129(3):420–436.

Bouton CE, Shaikhouni A, Annetta NV, Bockbrader MA, Friedenberg DA, Nielson DM, Sharma G, Sederberg PB, Glenn BC, Mysiw WJ, Morgan AG, Deogaonkar M, and Rezai AR (2016). Restoring cortical control of functional movement in a human with quadriplegia. Nature, 533:247–250. [PubMed: 27074513]

Brandman DM, Burkhart MC, Kelemen J, Franco B, Harrison MT, and Hochberg LR (2018a). Robust closed-loop control of a cursor in a person with tetraplegia using gaussian process regression. Neural Comput, 30(11):2986–3008. [PubMed: 30216140]

Brandman DM, Cash SS, and Hochberg LR (2017). Review: Human intracortical recording and neural decoding for brain-computer interfaces. IEEE Trans. Neural Syst. Rehabil. Eng, PP(99).

Brandman DM, Hosman T, Saab J, Burkhart MC, Shanahan BE, Ciancibello JG, Sarma AA, Milstein DJ, Vargas-Irwin CE, Franco B, Kelemen J, Blabe C, Murphy BA, Young DR, Willett FR, Pandarinath C, Stavisky SD, Kirsch RF, Walter BL, Bolu Ajiboye A, Cash SS, Eskandar EE, Miller JP, Sweet JA, Shenoy KV, Henderson JM, Jarosiewicz B, Harrison MT, Simeral JD, and Hochberg LR (2018b). Rapid calibration of an intracortical brain–computer interface for people with tetraplegia. J. Neural Eng, 15(2).

Brown RG and Hwang PYC (2012). Introduction to Random Signals and Applied Kalman Filtering John Wiley & Sons, Inc., fourth edition.

Buehner M, McTaggart-Cowan R, and 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 MC (2019). A Discriminative Approach to Bayesian Filtering with Applications to Human Neural Decoding PhD thesis, Brown University, Providence, Rhode Island, U.S.A.

Butler RW (2007). Saddlepoint approximations with applications, volume 22 of Cambridge Series in Statistical and Probabilistic Mathematics Cambridge University Press.

Cappé O, Godsill SJ, and 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, and Ryden T (2005). Inference in Hidden Markov Models Springer-Verlag.

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

Cedarbaum JM, Stambler N, Malta E, Fuller C, Hilt D, Thurmond B, and 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. [PubMed: 10540002]

Chang JT and 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 and Fleet DJ (2001). People tracking using hybrid Monte Carlo filtering. In Proc. Int. Conf. Comput. Vis, volume 2, pages 321–328.

Churchland MM, Cunningham JP, Kaufman MT, Foster JD, Nuyujukian P, Ryu SI, and Shenoy KV (2012). Neural population dynamics during reaching. Nature, 487(7405):1–20.

Collinger JL, Wodlinger B, Downey JE, Wang W, Tyler-Kabara EC, Weber DJ, McMorland AJC, Velliste M, Boninger ML, and Schwartz AB (2013). High-performance neuroprosthetic control by an individual with tetraplegia. Lancet, 381(9866):557–564. [PubMed: 23253623]

Daum FE (1984). Exact finite dimensional nonlinear filters for continuous time processes with discrete time measurements. In IEEE Conf. Decis. Control, pages 16–22.

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

Daum FE and Huang J (2003). Curse of dimensionality and particle filters. In 2003 IEEE Aerosp. Conf. Proc, volume 4.

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

Douc R and Cappé O (2005). Comparison of resampling schemes for particle filtering. In Proc. Int. Symp. Image and Signal Process. Anal, pages 64–69.

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

Duchi J, Hazan E, and 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 PM (1954). The information capacity of the human motor system in controlling the amplitude of movement. J. Exp. Pyschol, 47(6):381–391.

Flint RD, Lindberg EW, Jordan LR, Miller LE, and Slutzky MW (2012). Accurate decoding of reaching movements from field potentials in the absence of spikes. J. Neural Eng, 9(4).

Foresee FD and Hagan MT (1997). Gauss-Newton approximation to Bayesian learning. In Int. Conf. Neural Netw, volume 3, pages 1930–1935.

Gelb A (1974). Applied Optimal Estimation The MIT Press.

Georgopoulos AP, Kettner RE, and Schwartz AB (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. [PubMed: 3411362]

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

Gers FA, Schmidhuber J, and Cummins F (2000). Learning to forget: Continual prediction with LSTM. Neural Comput, 12(10):2451–2471. [PubMed: 11032042]

Ghahramani Z and Hinton GE (2000). Variational learning for switching state-space models. Neural Comput, 12(4):831–864. [PubMed: 10770834]

Gilja V, Pandarinath C, Blabe CH, Nuyujukian P, Simeral JD, Sarma AA, Sorice BL, Perge JA, Jarosiewicz B, Hochberg LR, Shenoy KV, and Henderson JM (2015). Clinical translation of a high-performance neural prosthesis. Nat. Med, 21(10):1142–1145. [PubMed: 26413781]

Glorot X and Bengio Y (2010). Understanding the difficulty of training deep feedforward neural networks. In Int. Conf. Artif. Intell. Stats, volume 9, pages 249–256.

Gordon NJ, Salmond DJ, and Smith AFM (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proc. F - Radar and Signal Process, 140(2):107–113.

Greff K, Srivastava RK, Koutník J, Steunebrink BR, and Schmidhuber J (2016). LSTM: A search space odyssey. IEEE Trans. Neural Netw. Learn. Syst, PP(99):1–11.

Grewal MS and Andrews AP (2010). Applications of Kalman filtering in aerospace 1960 to the present. IEEE Control Syst. Mag, 30(3):69–78.

Hagan MT and Menhaj MB (1994). Training feedforward networks with the Marquardt algorithm. IEEE Trans. Neural Netw, 5(6):989–993. [PubMed: 18267874]

Hall EC (1966). Case History of the Apollo Guidance Computer MIT.

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

Handschin JE and Mayne DQ (1969). Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering. Int. J. Control, 9(5):547–559.

Hess R and Fern A (2009). Discriminatively trained particle filters for complex multi-object tracking. In Comput. Vis. Pattern Recognit, pages 240–247.

Hochberg LR, Bacher D, Jarosiewicz B, Masse NY, Simeral JD, Vogel J, Haddadin S, Liu J, Cash SS, van der Smagt P, and Donoghue JP (2012). Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. Nature, 485(7398):372–375. [PubMed: 22596161]

Hochberg LR and Donoghue JP (2006). Sensors for brain-computer interfaces. IEEE Eng. Med. Biol. Mag, pages 32–38.

Hochreiter S and Schmidhuber J (1997). Long short-term memory. Neural Comput, 9(8):1735–1780. [PubMed: 9377276]

Horn RA and Johnson CR (2013). Matrix Analysis Cambridge University Press, second edition.

Hosman T, Vilela M, Milstein D, Kelemen JN, Brandman DM, Hochberg LR, and Simeral JD (2019). BCI decoder performance comparison of an LSTM recurrent neural network and a Kalman filter in retrospective simulation. In Int. IEEE EMBS Conf. Neural Eng.

Hunt BR, Kostelich EJ, and 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 and Szegedy C (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Bach F and Blei D, editors, Int. Conf. Mach. Learn, volume 37, pages 448–456. PMLR.

Ito K (2000). Gaussian filter for nonlinear filtering problems. In IEEE Conf. Decis. Control, volume 2.

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

Jarosiewicz B, Masse NY, Bacher D, Cash SS, Eskandar E, Friehs G, Donoghue JP, and Hochberg LR (2013). Advantages of closed-loop calibration in intracortical brain-computer interfaces for people with tetraplegia. J. Neural Eng, 10(4).

Jarosiewicz B, Sarma AA, Bacher D, Masse NY, Simeral JD, Sorice B, Oakley EM, Blabe C, Pandarinath Cand Gilja V, Cash SS, Eskandar EN, Friehs GM, Henderson JM, Shenoy KV, Donoghue JP, and Hochberg LR (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, and Sutskever I (2015). An empirical exploration of recurrent network architectures. In Bach F and Blei D, editors, Int. Conf. Mach. Learn, volume 37, pages 2342–2350. PMLR.

Julier SJ and Uhlmann JK (1997). New extension of the Kalman filter to nonlinear systems. Proc. SPIE, 3068:182–193.

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

Kalman RE and Bucy RS (1961). New results in linear filtering and prediction theory. J. Basic Eng, 83(1):95–108.

Kim S-P, Simeral JD, Hochberg LR, Donoghue JP, and Black MJ (2008). Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia. J. Neural Eng, 5(4).

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 LC, Shalizi CR, and Kass RE (2010). Approximate methods for state-space models. J. Am. Stat. Assoc, 105(489):170–180. [PubMed: 21753862]

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

Lemon RN (2008). Descending pathways in motor control. Annu. Rev. Neurosci, 31:195–218. [PubMed: 18558853]

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

Liu JS (2008). Monte Carlo Strategies in Scientific Computing Springer.

MacKay DJC (1992). Bayesian interpolation. Neural Comput, 4(3):415–447.

Majumdar SJ, Bishop CH, Etherton BJ, and Toth Z (2002). Adaptive sampling with the ensemble transform Kalman filter. part II: Field program implementation. Mon. Weather Rev, 130(5):1356–1369.

Malik WQ, Hochberg LR, Donoghue JP, Hochberg LR, Donoghue JP, and Brown EN (2015). Modulation depth estimation and variable selection in state-space models for neural interfaces. IEEE Trans. Biomed. Eng, 62(2):570–581. [PubMed: 25265627]

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

Masse NY, Jarosiewicz B, Simeral JD, Bacher D, Stavisky SD, Cash SS, Oakley EM, Berhanu E, Eskandar E, Friehs G, Hochberg LR, and Donoghue JP (2015). Non-causal spike filtering improves decoding of movement intention for intracortical bcis. J. Neurosci. Methods, 244:94–103. [PubMed: 25681017]

Maynard EM, Nordhausen CT, and Normann RA (1997). The utah intracortical electrode array: a recording structure for potential brain-computer interfaces. Electroencephalogr. Clin. Neurophysiol, 102(3):228–239. [PubMed: 9129578]

Metropolis N and Ulam S (1949). The Monte Carlo method. J. Am. Stat. Assoc, 44(247):335–341. [PubMed: 18139350]

Minka TP (2001a). Expectation propagation for approximate Bayesian inference. Uncertain. Artif. Intell

Minka TP (2001b). A family of algorithms for approximate Bayesian inference PhD thesis, MIT, Cambridge, Massachusetts, U.S.A.

Nadaraya EA (1964). On a regression estimate. Teor. Verojatnost. i Primenen, 9:157–159.

Nørgaard M, Poulsen NK, and 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 CH, Franco B, Mernoff ST, Eskandar EN, Simeral JD, Hochberg LR, Shenoy KV, and Henderson JM (2018). Cortical control of a tablet computer by people with paralysis. PLOS ONE, 13(11).

Ott E, Hunt BR, Szunyogh I, Zimin AV, Kostelich EJ, Corazza M, Kalnay E, Patil DJ, and Yorke JA (2004). A local ensemble Kalman filter for atmospheric data assimilation. Tellus A, 56(5):415–428.

Pandarinath C, Gilja V, Blabe CH, Nuyujukian P, Sarma AA, Sorice BL, Eskandar EN, Hochberg LR, Henderson JM, and Shenoy KV (2015). Neural population dynamics in human motor cortex during movements in people with ALS. eLife, 4.

Pandarinath C, Nuyujukian P, Blabe CH, Sorice BL, Saab J, Willett F, Hochberg LR, Shenoy KV, and Henderson JM (2017). High performance communication by people with paralysis using an intracortical brain-computer interface. eLife, pages 1–27.

Pandarinath C, O'Shea DJ, Collins J, Jozefowicz R, Stavisky SD, Kao JC, Trautmann EM, Kaufman MT, Ryu SI, Hochberg LR, Henderson JM, Shenoy KV, Abbott LF, and Sussillo D (2018). Inferring single-trial neural population dynamics using sequential auto-encoders. Nat. Methods, 15(10):805–815. [PubMed: 30224673]

Paninski L, Fellows MR, Hatsopoulos NG, and Donoghue JP (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, and Louradour J (2014). Dropout improves recurrent neural networks for handwriting recognition. In Int. Conf. Front. Handwriting Recognit, pages 285–290.

Pohlmeyer E, Solla S, Perreault EJ, and Miller LE (2007). Prediction of upper limb muscle activity from motor cortical discharge during reaching. J. Neural Eng, 4:369–379. [PubMed: 18057504]

Quang PB, Musso C, and Le Gland F (2015). The Kalman Laplace filter: A new deterministic algorithm for nonlinear Bayesian filtering. In Intern. Conf. Inf. Fusion, pages 1566–1573.

Quiñonero Candela J and Rasmussen CE (2005). A unifying view of sparse approximate Gaussian process regression. J. Mach. Learn. Res, 6:1939–1959.

Rao NG and Donoghue JP (2014). Cue to action processing in motor cortex populations. J. Neurophysiol, 111(2):441–453. [PubMed: 24174650]

Rasmussen CE and Nickisch H (2010). Gaussian processes for machine learning (GPML) toolbox. J. Mach. Learn. Res, 11:3011–3015.

Rasmussen CE and Williams CKI (2006). Gaussian processes for machine learning MIT Press, Cambridge, MA.

Real E, Moore S, Selle A, Saxena S, Suematsu YL, Le Q, and Kurakin A (2017). Large-scale evolution of image classifiers. Int. Conf. Mach. Learn

Särkkä S (2013). Bayesian Filtering and Smoothing Cambridge University Press.

Schmidhuber J (2015). Deep learning in neural networks: An overview. Neural Netw, 61:85–117. [PubMed: 25462637]

Schmidt SF, Weinberg JD, and Lukesh JS (1970). Application of Kalman filtering to the C-5 guidance and control system. In Leondes CT, editor, Theory and Applications of Kalman Filtering, chapter 13 NATO, Advisory Group for Aerospace Research and Development.

Schwartz AB (1994). Direct cortical representation of drawing. Science, 265(5171):540–542. [PubMed: 8036499]

Shumway RH and Stoffer DS (1991). Dynamic linear models with switching. J. Am. Stat. Assoc, 86(415):763–769.

Simeral JD, Kim S-P, Black MJ, Donoghue JP, and Hochberg LR (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).

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

Stevenson IH and Kording KP (2011). How advances in neural recording affect data analysis. Nat. Neurosci, 14(2):139–142. [PubMed: 21270781]

Sugiyama M, Suzuki T, and Kanamori T (2012). Density Ratio Estimation in Machine Learning Cambridge University Press.

Sussillo D, Nuyujukian P, Fan JM, Kao JC, Stavisky SD, Ryu S, and Shenoy K (2012). A recurrent neural network for closed-loop intracortical brain–machine interface decoders. J. Neural Eng, 9(2).

Sussillo D, Stavisky SD, Kao JC, Ryu SI, and Shenoy KV (2016). Making brain–machine interfaces robust to future neural variability. Nat. Commun, 7.

van der Merwe R (2004). Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models PhD thesis, Oregon Health & Science University, Portland, Oregon, U.S.A.

van der Vaart AW (1998). Asymptotic statistics Cambridge University Press, Cambridge.

Vargas-Irwin CE, Brandman DM, Zimmermann JB, Donoghue JP, and Black MJ (2015). Spike train SIMilarity space (SSIMS): a framework for single neuron and ensemble data analysis. Neural Comput, 27(1):1–31. [PubMed: 25380335]

Vargas-Irwin CE, Shakhnarovich G, Yadollahpour P, Mislow JMK, Black MJ, and Donoghue JP (2010). Decoding complete reach and grasp actions from local primary motor cortex populations. J. Neurosci, 30(29):9659–9669. [PubMed: 20660249]

Velliste M, Perel S, Spalding MC, Whitford AS, and Schwartz AB (2008). Cortical control of a prosthetic arm for self-feeding. Nature, 453(7198):1098–101. [PubMed: 18509337]

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

Wan EA and van der Merwe R (2000). The unscented Kalman filter for nonlinear estimation. In Adaptive Syst. for Signal Process., Commun., and Control Symp, pages 153–158.

Watson GS (1964). Smooth regression analysis. Sankhyā Ser. A, 26:359–372.

Willett FR, Young DR, Murphy BA, Memberg WD, Blabe CH, Pandarinath C, Stavisky SD, Rezaii P, Saab J, Walter BL, Sweet JA, Miller JP, Henderson JM, Shenoy KV, Simeral JD, Jarosiewicz B, Hochberg LR, Kirsch RF, and Bolu Ajiboye A (2019). Principled bci decoder design and parameter selection using a feedback control model. Sci. Rep, 9(8881).

Wodlinger B, Downey JE, Tyler-Kabara EC, Schwartz AB, Boninger ML, and Collinger JL (2015). Ten-dimensional anthropomorphic arm control in a human brain machine interface: difficulties, solutions, and limitations. J. Neural Eng, 12(1).

Wolpaw JR, Birbaumer N, McFarland DJ, Pfurtscheller G, and Vaughan TM (2002). Brain-computer interfaces for communication and control. Clin. Neurophysiol, 113(6):767–91. [PubMed: 12048038]

Wu W, Black MJ, Gao Y, Bienenstock E, Serruya M, and Donoghue JP (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, pages 66–73.

Zaremba W, Sutskever I, and Vinyals O (2014). Recurrent neural network regularization. ArXiv e-prints

Zeiler MD (2012). Adadelta: An adaptive learning rate method. ArXiv e-prints

Zoph B and Le QV (2017). Neural architecture search with reinforcement learning. Int. Conf. Learn. Represent

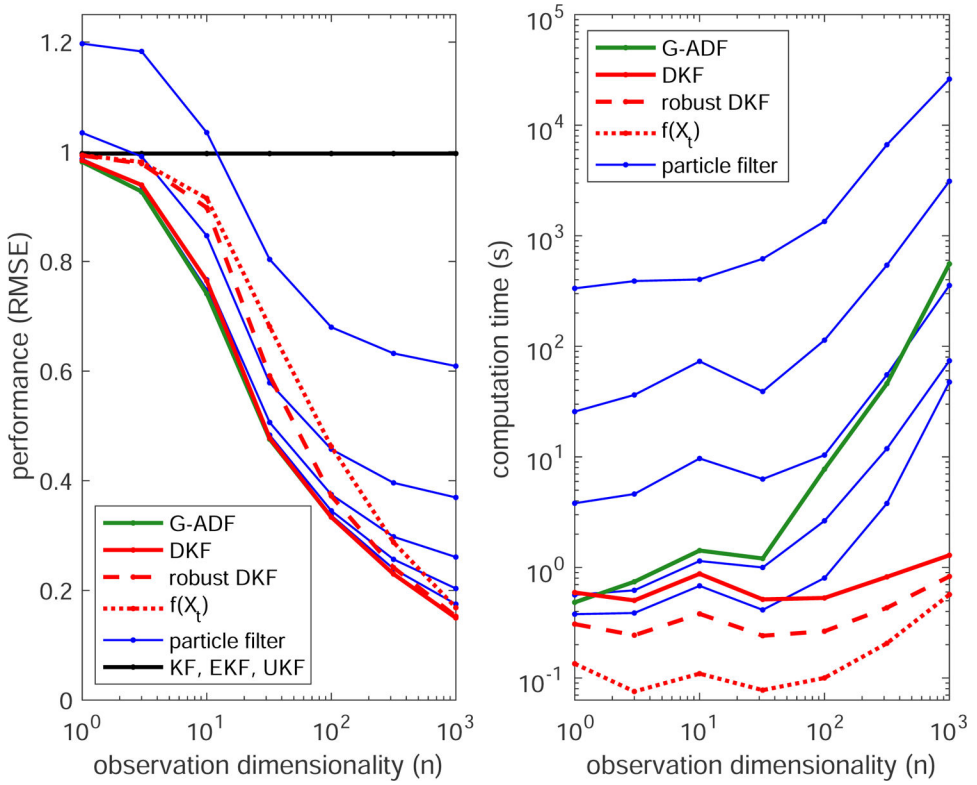

Figure 1: Kalman observation mixtures.

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 $\Gamma = S - ASA^\top$. The number of categories is $L = 2$, the category probabilities are $\boldsymbol{\pi} = (0.5, 0.5)$, 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 $\overline{H} = 0$; see Equation 21. 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 was 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 root mean squared error (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.

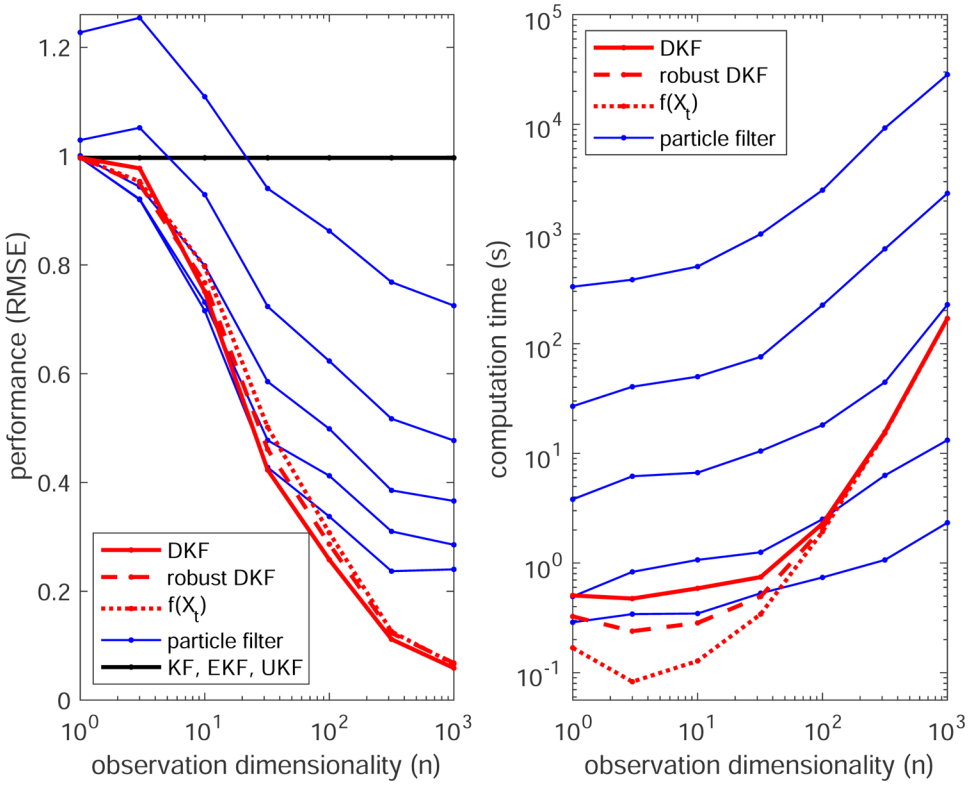

Figure 2: Independent Bernoulli mixtures.

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 $\boldsymbol{\pi} = (0.5, 0.5)$, 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 22. The $d_{1:n}$ were chosen as independent uniform$\{1,\ldots,d\}$, and the $\gamma_{1:n}$ were chosen as independent $\mathcal{N}(0, 1)$.

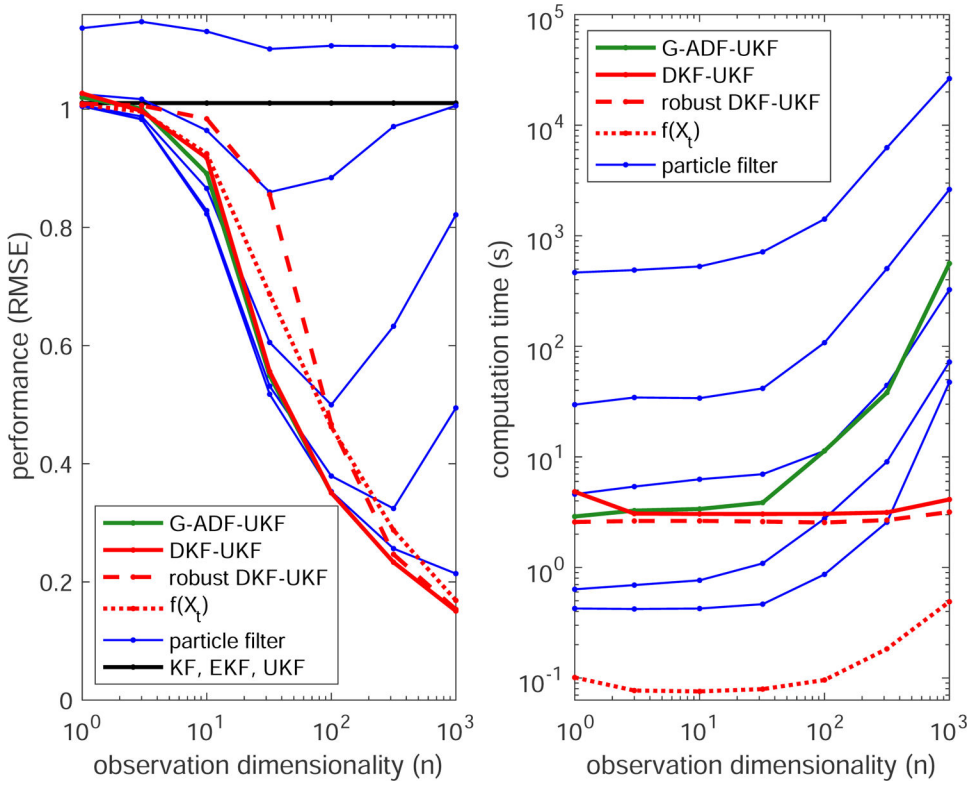

Figure 3: Nonlinear state dynamics.

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 an UKF.

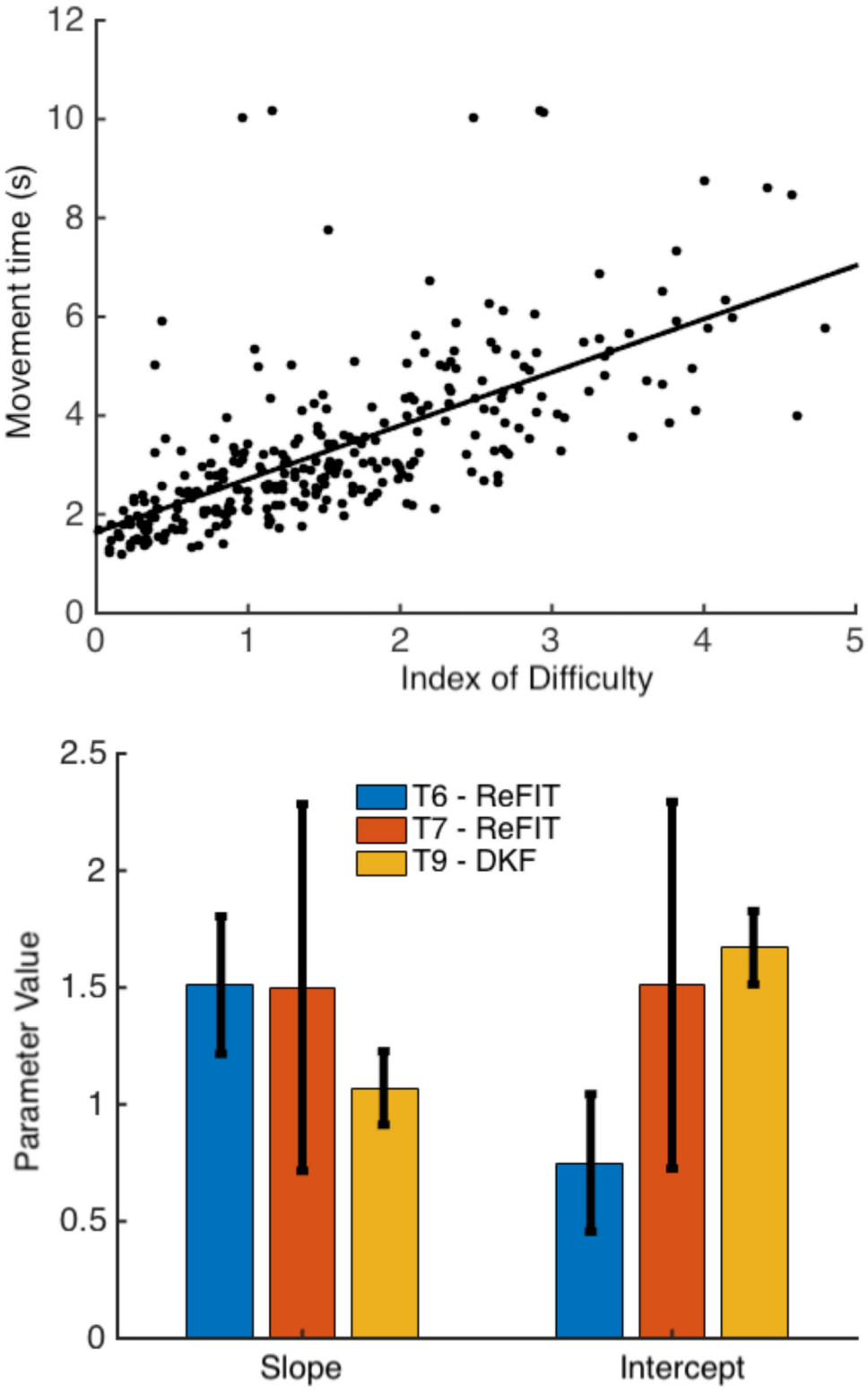

Figure 4:

On the left, we plot movement time vs. 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 24) correspond to less of an increase in movement time for more difficult targets. Estimates for the intercept parameter correspond to b in Equation 24.

Table 1:

This figure compares the normalized RMSE (nRMSE) for various filtering methods on the Flint dataset from Section 4.5. The nRMSE is computed by dividing the RMSE by the root mean square of the observation vector, so that predicting identically zero would yield a 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 1–6 refer to completely separate trials using new training and testing data. The final column gives average performance across the six trials.

	Trial 1	**Trial 2**	**Trial 3**	**Trial 4**	**Trial 5**	**Trial 6**	**Avg.**
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%

Table 2:

This figure compares the mean absolute angular error (radians) for various filtering methods on the Flint dataset from Section 4.5. 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° = $\pi/4 \approx 0.79$ 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.

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	Avg
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%