Ann Inst Stat Math (2010) 62:37–59
DOI 10.1007/s10463-009-0249-x

Bayesian decoding of neural spike trains

**Shinsuke Koyama · Uri T. Eden ·
Emery N. Brown · Robert E. Kass**

Received: 27 June 2008 / Revised: 2 April 2009 / Published online: 30 July 2009

Abstract Perception, memory, learning, and decision making are processes carried out in the brain. The performance of such intelligent tasks is made possible by the communication of neurons through sequences of voltage pulses called spike trains. It is of great interest to have methods of extracting information from spike trains in order to learn about their relationship to behavior. In this article, we review a Bayesian approach to this problem based on state-space representations of point processes. We discuss some of the theory and we describe the way these methods are used in decoding

S. Koyama (✉) · R. E. Kass
Department of Statistics, Center for the Neural Basis of Cognition,
Carnegie Mellon University, Pittsburgh, PA 15213, USA
e-mail: koyama@stat.cmu.edu

U. T. Eden
Department of Mathematics and Statistics,
Boston University, Boston, MA 02215, USA
e-mail: tzvi@bu.edu

E. N. Brown
Neuroscience Statistics Research Laboratory,
Department of Anesthesia and Critical Care,
Massachusetts General Hospital, Boston, MA 02114, USA

E. N. Brown
Division of Health Sciences and Technology, Harvard Medical School/
Massachusetts Institute of Technology, Cambridge, MA 02139, USA
e-mail: brown@neurostat.mgh.harvard.edu

R. E. Kass
Department of Machine Learning, Center for the Neural Basis of Cognition,
Carnegie Mellon University, Pittsburgh, PA 15213, USA
e-mail: kass@stat.cmu.edu

motor cortical activity, in which the hand motion is reconstructed from neural spike trains.

Keywords Point process · State-space model · Recursive Bayesian filter · Sequential Gaussian approximation · Neural decoding

1 Introduction

The brain receives, processes, and transmits information about the outside world through stereotyped electrical discharges, called action potentials or spikes. External sensory signals are transformed into intricate sequences of these spike events at an early stage of processing within the central nervous system, and all subsequent brain area interactions and computations are carried out using these spike trains. Spike trains are the starting point for virtually all of the processing performed by the brain (Kandel 2000; Dayan and Abbot 2001). It is therefore very important to have good statistical models for spike trains, and methods for examining their relationship to particular stimuli or behavior.

A basic property of action potentials is that they are "all or nothing" events: each time a neuron fires, its voltage wave form is nearly the same, regardless of context. This allows us to reduce spike trains to sequences of event times. On the other hand, while all action potentials remain essentially the same across repeated firing events, the spike sequences generated in response to identical stimuli applied on separate occasions will not be identical; although they may share common statistical features, they exhibit substantial variation. These two fundamental facts suggest that spike train data might be effectively analyzed with the theory of point processes (Snyder 1972; Snyder and Miller 1991; Daley and Vere-Jones 2003). Point processes are used to model physical systems that produce a stochastic set of localized events in time or space. In this case, we use temporal point processes to represent the times of the spiking events and to express the probability distribution of specific sequences of spike times.

While neurons have as immediate inputs the spiking activity of other neurons, these inputs are generally not recorded. Instead, the relationship of spiking activity to a stimulus or behavior becomes the focus of neurophysiological investigation. The prototypical neuron we consider here has inputs from multiple sources and produces a spike train in response. Describing the way a neuron represents information about a stimulus or behavior is the *encoding problem*. The dual problem of reproducing the stimulus or behavior from neural spike trains is the *decoding problem*. The ability to reconstruct a signal from observed neural activity provides a way of verifying that explicit information about the external world is represented in particular neural spike trains (Rieke et al. 1997; Dayan and Abbot 2001). Its engineering importance comes from efforts to design brain-machine interface devices, especially neural motor prostheses such as computer cursors, robotic arms, etc (Schwartz 2004). A natural starting point is to take the behavior, such as a hand movement, to follow a dynamic state model and to infer the evolution of states using Bayes' theorem.

An additional consideration is that neural systems are dynamic, in that individual neurons constantly change their response properties to relevant stimuli (Brown et al. 2001; Frank et al. 2002, 2004). Furthermore, groups of neurons maintain dynamic representations of relevant stimuli in their ensemble firing patterns (Brown et al. 1998; Barbieri et al. 2004). For these reasons, the development of state-space algorithms to characterize the dynamic properties of neural systems from point process observations has been a productive research area in computational neuroscience.

In this article, we review the state-space approach for solving the neural decoding problem. We proceed to construct a state space estimation and inference framework by writing state models that describe the evolution of the stochastic signals to estimate, and intensity models for neurons that define the probability density of observing a particular sequence of spike times. Posterior densities can then be computed using a recursive Bayesian filter for discrete-time analyses. In Sect. 2, we discuss the construction of point process models to describe spiking activity of individual neurons within an ensemble. In Sect. 3, we develop our point process estimation framework for discrete time analyses. We begin by constructing state transition equations and probability densities for discrete time observations and using these to derive an exact recursive expression for the posterior density of the state given the spiking data. By using a Gaussian approximation to the posterior density, we construct a computationally efficient recursive estimation algorithm for neural spiking data. We then show that this algorithm does not lead to accumulation of approximation error. In Sect. 4, we illustrate applications of the point process estimation to neural decoding problems. Section 5 summarizes our results and discusses areas for future study.

2 Stochastic neural spiking model

2.1 Conditional intensity function

Given an observation interval $(0, T]$, let $N(t)$ be a counting process, which is the total number of spikes fired by a given neuron in the interval $(0, t]$, for $t \in (0, T]$. The conditional intensity function for a stochastic neural point process is defined by

$$\lambda(t|H(t)) = \lim_{\Delta t \to 0} \frac{P(N(t+\Delta t) - N(t) = 1|H(t))}{\Delta t},$$

where $H(t)$ includes the neuron's spiking history up to time t and other relevant covariates. According to this definition, we can write the probability of a neuron firing a single spike in the interval $[t, t+\Delta t]$ by $\Pr(N(t+\Delta t) - N(t)|H_t) = \lambda(t|H(t))\Delta t + o(\Delta t)$. When the spiking distribution is independent of its past history, this becomes a Poisson process, and $\lambda(t|H(t)) = \lambda(t)$ is simply a rate function of the Poisson process. Thus, the conditional intensity function is a generalization of the intensity function of the Poisson process including the dependency of the spike history.

Let $0 < u_1 < u_2 < \cdots < u_{n-1} < u_n \leq T$ be a set of spike times from a point process. Using the conditional intensity function $\lambda(t|H(t))$, the joint probability distribution of a spike train $\{u_i\}_{i=1}^{n}$ in an interval $(0, T]$ is expressed as

$$p(\{t_i\}_{i=1}^{n}) = \left[\prod_{i=1}^{n} \lambda(u_i|H(u_i))\right] \exp\left(-\int_0^T \lambda(u|H(u))\mathrm{d}u\right), \qquad (1)$$

Intuitively, we can think of the first term on the right-hand side of (1) as the probability density associated with observing action potentials at the actual spike times and the second term as the probability of not observing any other spikes anywhere in the interval. The conditional intensity function provides a succinct representation of the joint probability density of a sequence of spike times, and fully characterizes the stochastic structure of the modeled neuron's spiking activity. In the following, we provide several classes of point processes by specifying the form of the conditional intensity function for modeling neural spike trains.

2.2 Neural point process models

We can categorize many of the different types of signals that covary with spiking activity of a neuron into three distinct groups. First, neural activity is often associated with extrinsic biological and behavioral signals, such as sensory stimuli and behavioral or specific motor outputs. In neural decoding problems, these external signals are the ones about which we typically want to make inferences from the observed spiking activity. Second, the probability of firing a spike at a particular moment in time will generally depend on the neuron's past spiking history. Third, recent technological advancements that allow for the simultaneous recording of a large number of neurons have shown that the spiking activity of a particular neuron can be related to the concurrent and past activity of a population of other neurons (Pillow et al. 2008; Smith and Kohn 2008).

The class of conditional intensity models with which we work must be able to incorporate the simultaneous effects of extrinsic covariates, internal spiking history, and ensemble activity. In other words, the conditional intensity processes that we use can be written generally as

$$\lambda(t|H(t)) = f(t, x_{[0,t]}, N_{[0,t)}, \{N^c_{[0,t)}\}_{c=1}^{C}),$$

where $x_{[0,t]}$ represents the values of a set of external covariates up to and including the present time, $N_{[0,t)}$ represents the neuron's spiking history, and $\{N^c_{[0,t)}\}_{c=1}^{C}$ represents the firing histories of other C neurons.

2.2.1 Generalized additive models

A simple class of models is built so that the logarithm of the conditional intensity function has a linear combination of general functions of the covariates,

$$\log \lambda(t|H(t)) = \sum_{i=1}^{m} \theta_i g_i(\{x_j(t)\}), \qquad (2)$$

where g_i is a general function of covariates $\{x_i(t)\}$, and θ_i is a parameter representing the degree of modulation of the conditional intensity to the specified function of the covariates. Equation (2) has the same form as a generalized additive model (GAM) under a Poisson probability model and a log link function (Hastie and Tibshirani 1990). When g_i is linear, it corresponds to the generalized linear model (GLM) (McCullagh and Nelder 1989). The GAM framework is advantageous because it is integrated into several standard mathematical and statistical packages and a model is easy to construct. These properties make the GAM framework ideal for rapidly assessing the possible relevance of a large number of covariates on the spiking properties of neurons.

For example, the model that incorporates both the spike history and an extrinsic covariate $I(t)$ may have the form,

$$\log \lambda(t|H(t)) = \theta_0 + \theta_1 g(I(t)) + \theta_2 \Delta N_{t-1},$$

where $\exp(\theta_0)$ is the baseline firing rate, and ΔN_{t-1} is an indicator function that is 1 if the neuron has just fired a spike in the preceding millisecond. Therefore, this model is able to capture both the neuron's dependence on the input stimulus and the spike history dependence with a simple functional form. Truccolo et al. (2005) provides a complete discussion of the construction and evaluation of GAM models for neural systems.

2.2.2 Inhomogeneous Markov interval model

An inhomogeneous Markov interval (IMI) model simply incorporates into the conditional intensity the dependency of the last spike time preceding t:

$$\lambda(t|H(t)) = r(t, t - s_*(t)),$$

where $s_*(t)$ is the preceding spike time. Regarding the interaction between the spike history dependence and the effect of extrinsic covariates, two special classes of IMI models have been considered in the following literature: multiplicative IMI models (Kass and Ventura 2001, and references therein) and time-rescaled renewal-process models (Barbieri et al. 2001; Koyama and Shinomoto 2005; Reich et al. 1998).

The conditional intensity function of the multiplicative IMI model has the form

$$\lambda(t|H(t)) = \lambda_1(t) g_1(t - s_*(t)),$$

where $\lambda_1(t)$ modulates the firing rate only as a function of the experimental clock, while $g_1(t - s_*(t))$ represents the dependence on the last spike time preceding t.

The time-rescaled renewal-process model has the form

$$\lambda(t|H(t)) = \lambda_0(t) g_0(\Lambda_0(t) - \Lambda_0(s_*(t))),$$

where g_0 is the hazard function of a renewal process and $\Lambda_0(t)$ is defined as

$$\Lambda_0(t) = \int_0^t \lambda_0(u)\mathrm{d}u.$$

In both models, λ_0 and λ_1 are often called *excitability functions* to indicate that they modulate the amplitude of the firing rate, while g_0 and g_1 are called *recovery functions* to indicate that they affect the way the neuron recovers its ability to fire after generating a spike. The fundamental difference between the two models is the way the excitability interacts with the recovery function (Koyama and Kass 2008). In the multiplicative IMI model, the refractory period (in which a spike cannot be evoked after the last spike) represented in the recovery function is not affected by excitability or firing rate variations. In the time-rescaled renewal-process model, however, the refractory period is no longer fixed but is scaled by the firing rate.

2.3 Goodness-of-fit tools

We have introduced several classes of neural point process models above. Any such models are, however, only approximations of the true physiological mechanisms underlying spike generation. Goodness-of-fit measures are used to assess the extent to which the models can capture the statistical structure presented in data, and allow us to compare models.

2.3.1 AIC and BIC

The probability distribution a sequence of spikes (1) serves as the point process likelihood of a model with a specified set of parameters. The likelihood can therefore be used to infer static parameters for a given model class via maximum likelihood (Paninski 2004), or to compare among different model classes. In order to select among different model classes with potentially different number of parameters, Akaike (1974) developed an information criterion (AIC) that uses a penalized log likelihood function,

$$\mathrm{AIC(model)} = -2\log P(\{N(t)\}|\{x(t)\}) + 2m,$$

where $\{N(t)\}$ and $\{x(t)\}$ are an observed counting process and covariate, respectively, m is the number of parameters for the model class, and the likelihood function is computed at the maximum likelihood estimate of the model parameters. AIC was originally developed as an approximate estimate the Kullback–Leibler divergence between the true stochastic process generating observations and a proposed model for the process. Thus, under this criterion, the model class that has the smallest AIC most parsimoniously describes the statistical structure of the process.

The Bayesian information criterion (BIC) is an alternative to AIC for model selection (Schwartz 1978; Kass and Raftery 1995). The generic form of BIC for the point process model is

$$\mathrm{BIC(model)} = -2\log P(\{N(t)\}|\{x(t)\}) + \log n \cdot m,$$

where n is the number of spikes observed in the interval, and the likelihood function is computed at the maximum likelihood estimate of the parameters as with AIC. The difference from AIC is that the factor 2 of the penalty term in AIC is replaced by $\log n$. In spite of its similarity with AIC, BIC arises in quite a different way; BIC estimates the posterior probability of the model given data under the assumption that the prior over models is uniform. BIC tends to penalize complex models more strongly than does the AIC, giving preference to simpler models in selection.

2.3.2 Time-rescaling transformation and Kolmogorov–Smirnov test

While AIC and BIC are useful for comparing among competing models, the scale of them is, by itself, somewhat difficult to interpret. Thus, it is useful to develop diagnostic tools to assess the goodness-of-fit of proposed models to data. The Kolmogorov–Smirnov (K–S) test and corresponding K–S test can be used for this purpose (Berman 1983; Ogata 1988; Brown et al. 2002).

For constructing the K–S plot, consider the time-rescaling transformation obtained from the estimated conditional intensity,

$$\Lambda(t) = \int_0^t \lambda(u|H(u))\mathrm{d}u,$$

which is a monotonically increasing function because $\lambda(u|H(u))$ is nonnegative. If the conditional intensity were correct, then according to the time-rescaling theorem rescaling spike times, $\tau_i = \Lambda(t_i)$, have the distribution of a stationary Poisson process of unit rate (Papangelou 1972). Thus, the rescaled inter-spike intervals, $y_i = \Lambda(t_i) - \Lambda(t_{i-1})$, would be independent exponential random variables with mean 1. Using the further transformation, $z_i = 1 - \exp(-y_i)$, $\{z_i\}$ would then be independent uniform random variables on the interval $(0, 1)$. In the K–S plot we order the z_is from smallest to largest and, denoting the ordered values as $z_{(i)}$, plot the values of the cumulative distribution function of the uniform density, i.e., $b_i = \frac{i-\frac{1}{2}}{n}$ for $i = 1, \ldots, n$, against the $z_{(i)}$s. If the model were correct, then the points would lie close to a 45° line. For moderate to large sample sizes the 95% probability bands are well approximated as $b_i \pm 1.36/n^{1/2}$ (Johnson and Kotz 1970). The Kolmogorov–Smirnov test rejects the null-hypothetical model if any of the plotted points lie outside these bands.

If the conditional intensity were correct, the rescaled intervals should also be independent of each other. To test for independence of the transformed intervals, we can plot each transformed interval z_i against the next z_{i+1}; if there were no serial correlation, the resulting plot is distributed uniformly in a unit square.

2.3.3 Point process residual analysis

A stochastic process model divides each observation into a predictable component and a random component. One approach to examine if a model describes a statistical structure of the observation data is to search for structure in the residuals between the data and the predictive component of the model. For a univariate point process, the residual error process can be defined as

$$R(t) = N(t) - \int_0^t \lambda(u|H(u))\mathrm{d}u.$$

If the conditional intensity model is correct, then this residual process is a martingale and has zero expectation over any finite interval, independent of the state process. For a neural system, this residual process should be uncorrelated with any potential biological input signal. If there is a statistically significant correlation between the residual process and an input signal, then we should refine the neural model that captures the predictable residual component resulting from this correlation.

3 Neural decoding

The previous section focused on the construction of neural spiking models, which use relevant covariates, $x(t)$ related to sensory stimuli, physiological states or motor behavior, to describe the statistical structure of neural spike trains, $\{u_i\}$. Neural decoding is the reverse problem of inferring a set of dynamic extrinsic covariates, $x(t)$, from the observed spiking activity, $\{u_i\}$. To construct the decoding algorithm, we switch from a continuous-time to a discrete-time framework so that we can make use of recursive algorithms, to be introduced below. In a discrete time setting, it is first necessary to partition the observation interval into a discrete set of times, $\{t_i : t_0 < t_1 < \cdots < t_K < T\}$. All pertinent model components, such as the system state and observation values, are then only defined at these specified times. For convenience, we write x_k for $x(t_k)$. Likewise, we write λ_k for $\lambda(t_k|H(t_k))$, the history-dependent conditional intensity of a single neuron at these times, or write λ_k^i for that of the ith neuron in an ensemble. The difference $\Delta t_k = t_k - t_{k-1}$ is the duration of the interval of the partition. The difference in the counting process for a spiking neuron between two adjacent points in the partition, $\Delta N_k = N(t_k) - N(t_{k-1})$, defines a random variable giving the number of spikes fired during this interval. We write ΔN_k^i when we specify the spike counts of the ith neuron in an ensemble in the duration of Δt_k. If the values of Δt_k are small enough, then there will never be more than a single event in an interval, and the collection, $\{\Delta N_k\}_{k=1}^K$, is just the typical sequence of zeros and ones used to express spike train data in a discrete time series. We use $\Delta N_{1:k} \equiv \{\Delta N_i\}_{i=1}^k$ to represent the collection of all observed spikes up to time t_k.

3.1 Recursive Bayesian estimation

The estimate of the state signal, x_k, is based on its posterior density conditioned on the set of all past observations, $p(x_k|\Delta N_{1:k})$, which is recursively computed by combining the state and observation models.

Consider a vector random state signal, $\{x_k\}_{k=0}^K$, that correlates with the observed spiking activity of a population of neurons. Assume that this signal can be modeled as a dynamic system whose evolution is given by a linear stochastic difference equation:

$$x_{k+1} = F_k x_k + \epsilon_k, \tag{3}$$

where ϵ_k is zero mean white noise process with $\mathrm{Var}(\epsilon) = Q_k$, and the distribution of ϵ_k is independent of that of ϵ_j for all $j \neq k$. This state equation defines the probability of the state at each discrete time point given its previous value, $p(x_{k+1}|x_k) = \mathcal{N}(F_k x_k, Q_k)$.

The observation model, $p(\Delta N_k|x_k, H_k)$, is obtained by approximating the conditional intensity function, which has been defined in Sect. 2, in the discrete-time for the probability of observing a spike in the interval $(t_{k-1}, t_k]$ as

$$p(y_k|x_k, H_k) = \exp(y_k \log(\lambda_k \Delta t_k) - \lambda_k \Delta t_k) + o(\Delta t_k). \tag{4}$$

(Brown et al. 2003). Notice that up to order $o(\Delta t_k)$, this spiking distribution is equivalent to a Poisson distribution with parameter $\lambda_k \Delta t_k$ even though the spiking process is not a Poisson process.

By applying Bayes' rule, we obtain the posterior density of the state given the observations up to the current time,

$$\begin{aligned} p(x_k|\Delta N_{1:k}) &= \frac{p(\Delta N_{1:k}, x_k)}{\Pr(\Delta N_{1:k})} \\ &= \frac{p(\Delta N_k, \Delta N_{1:k-1}, x_k)}{\Pr(\Delta N_{1:k})} \\ &= \frac{\Pr(\Delta N_k|\Delta N_{1:k-1}, x_k) p(\Delta N_{1:k-1}, x_k)}{\Pr(\Delta N_{1:k})} \\ &= \frac{\Pr(\Delta N_k|\Delta N_{1:k-1}, x_k) p(x_k|\Delta N_{1:k-1})}{\Pr(\Delta N_k|\Delta N_{1:k-1})}. \end{aligned} \tag{5}$$

The first term in the numerator of (5) is the observation model. The second term is the one-step prediction density defined by the Chapman–Kolmogorov equation as

$$p(x_k|\Delta N_{1:k-1}) = \int p(x_k|x_{k-1}) p(x_{k-1}|\Delta N_{1:k-1}) \mathrm{d}x_{k-1}. \tag{6}$$

Equation (6) has two components: the state model, $p(x_k|x_{k-1})$, and the posterior density from the last iteration step, $p(x_{k-1}|\Delta N_{1:k-1})$. Thus, Eqs. (5) and (6) give a complete, recursive solution to the filtering problem for state-space models. When both the state and observation models are linear Gaussian, the solution of (5) to (6) reduces exactly the Kalman filter. (For historical remarks on the Kalman filter see Akaike (1994)). Equation (4), however, is in general nonlinear and non-Gaussian, ruling out any hope of a finite-dimensional exact filter. We develop the approximate filters based on Gaussian approximations in the following.

3.2 Approximate filters

Let $x_{k|k-1}$ and $V_{k|k-1}$ be the (approximate) mean and variance for the one-step prediction density, $p(x_k|\Delta N_{1:k-1})$, and $x_{k|k}$ and $V_{k|k}$ be the mean and variance for the posterior density, $p(x_k|\Delta N_{1:k})$, at time k. The goal in this derivation is to obtain a recursive expression for the posterior mean, $x_{k|k}$, and the posterior variance, $V_{k|k}$, in terms of observed and previously estimated quantities. There are a number of ways to approximating the posterior distribution depending on the desired balance between accuracy and computational efficiency (Tanner 1996; Julier and Uhlmann 1997; Doucet et al. 2001). Here, we introduce Gaussian approximations in the posterior distribution at each point in time, which provides a simple algorithm that is computationally tractable.

Under a Gaussian approximation in the posterior density, the one-step prediction density (6) at time k is also Gaussian with the mean and variance are, respectively, computed from those of posterior density at the previous time-step as

$$x_{k|k-1} = F_k x_{k-1|k-1}, \tag{7}$$

$$V_{k|k-1} = F_k V_{k-1|k-1} F_k^T + Q_k. \tag{8}$$

The posterior density (5) at time k is then approximated to a Gaussian with the parameters $x_{k|k}$ and $V_{k|k}$ which are to be determined. Depending on the choice of these parameters, we can construct different approximate filters as follows (Brown et al. 1998; Eden et al. 2004; Koyama et al. 2008).

3.2.1 Stochastic state point process filter

One way to make a Gaussian approximation is to expand the log posterior distribution in a Taylor series about some point $\bar{x}$ up to the second-order term, and complete a square to obtain an expression of the form of the Gaussian. Let $l(x_k) = \log \Pr(\Delta N_k|x_k, H_k)$ $p(x_k|\Delta N_{1:k-1})$ be the log posterior distribution (in which we omit the normalization constant) at time k. Expanding the log posterior distribution about a point $\bar{x}$ up to the second order yields

$$\begin{aligned} l(x_k) &\approx l(\bar{x}) + l'(\bar{x})(x_k - \bar{x}) + \frac{1}{2}(x_k - \bar{x})^T l''(\bar{x})(x_k - \bar{x}) \\ &= \frac{1}{2}\left[x_k - \{\bar{x} - l''(\bar{x})^{-1} l'(\bar{x})\}\right]^T l''(\bar{x}) \left[x_k - \{\bar{x} - l''(\bar{x})^{-1} l'(\bar{x})\}\right] \\ &\quad + \text{const.} \end{aligned}$$

Thus the posterior distribution is approximated to a Gaussian whose mean $x_{k|k}$ and the variance $V_{k|k}$ are, respectively, given by

$$x_{k|k} = \bar{x} - l''(\bar{x})^{-1} l'(\bar{x}), \tag{9}$$

$$V_{k|k} = [-l''(\bar{x})]^{-1}. \tag{10}$$

We may choose to make this approximation precise at any single point by evaluating this expression for $\bar{x}$ at that point. Evaluating at $\bar{x} = x_{k|k-1}$ gives a simple posterior state equation,

$$V_{k|k}^{-1} = V_{k|k-1}^{-1} + \sum_{j=1}^{C} \left[\left(\frac{\partial \log \lambda_k^j}{\partial x_k} \right)^T [\lambda_k^j \Delta t_k] \left(\frac{\partial \log \lambda_k^j}{\partial x_k} \right) - (\Delta N_k^j - \lambda_k^j \Delta t_k) \frac{\partial^2 \log \lambda_k^j}{\partial x_k \partial x_k^T} \right]_{x_{k|k-1}}, \tag{11}$$

and

$$x_{k|k} = x_{k|k-1} + V_{k|k} \sum_{j=1}^{C} \left[\left(\frac{\partial \log \lambda_k^j}{\partial x_k} \right)^T (\Delta N_k^j - \lambda_k^j \Delta t_k) \right]_{x_{k|k-1}}. \tag{12}$$

Equations (7), (8), (11) and (12) comprise a recursive estimation of the states. This approximate filter, introduced by Eden et al. (2004), is called *stochastic state point process filter* (SSPPF). An important feature of this filter is the fact that the posterior mean equation (12) is linear in $x_{k|k}$, eliminating the need for iterative solving procedures at each time step.

3.2.2 A maximum a posterior (MAP) point process filter

Alternatively, if we choose to set $\bar{x}$ in (9) and (10) to the maximum of the posterior distribution, $\hat{x} \equiv \arg\max_{x_k} l(x_k)$, so that $\bar{x} = x_{k|k}$, we obtain a modified approximate filter:

$$x_{k|k} = x_{k|k-1} + V_{k|k} \sum_{j=1}^{C} \left[\left(\frac{\partial \log \lambda_k^j}{\partial x_k} \right)^T (\Delta N_k^j - \lambda_k^j \Delta t_k) \right]_{x_{k|k}}, \tag{13}$$

and

$$V_{k|k}^{-1} = V_{k|k-1}^{-1} + \sum_{j=1}^{C} \left[\left(\frac{\partial \log \lambda_k^j}{\partial x_k} \right)^T [\lambda_k^j \Delta t_k] \left(\frac{\partial \log \lambda_k^j}{\partial x_k} \right) - (\Delta N_k^j - \lambda_k^j \Delta t_k) \frac{\partial^2 \log \lambda_k^j}{\partial x_k \partial x_k^T} \right]_{x_{k|k}}. \tag{14}$$

Equations (7), (8), (13) and (14) are identical to the MAP point process filter derived in Brown et al. (1998). Unlike the expression for the posterior mean estimator in the SSPPF, (13) is a nonlinear expression in $x_{k|k}$. In general, the posterior mean estimator can be solved at each time step using an iterative procedure such as Newton's method, with the one-step prediction estimate as the starting point.

Note that Eq. (9) corresponds to the Newton step to maximize the log posterior distribution. Thus the approximated posterior mean (12) in the SSPPF can be interpreted as one step modification of the prediction mean, $x_{k|k-1}$, toward the MAP estimate.

3.2.3 Laplace–Gaussian filter

It turns out in view of Laplace approximation [an asymptotic expansion of a Laplace-type integral Erdelyi (1956)] that the MAP estimate, $x_{k|k} = \hat{x}$, gives the first-order approximation for the posterior mean with respect to an expansion parameter, γ, with error of order $O(\gamma^{-1})$. Here γ measures the concentration of the posterior distribution, (that is, $l(x_k)/\gamma$ is a constant-order function of γ as $\gamma \to \infty$) depending on sample size, variability in the state model, and details of the observation and state models. This insight motivates us to utilize the second-order Laplace approximation for the posterior mean, $x_{k|k}$, in order to improve the accuracy of the approximation. In the ordinary static context, Tierney et al. (1989) analyzed a refined procedure, the "fully exponential" Laplace approximation, which gives a second-order approximation for posterior mean, having an error of order $O(\gamma^{-2})$.

The second-order (fully exponential) Laplace approximation for the posterior expectation in the one-dimensional case is calculated as follows. [The multi-dimensional extension is straightforward; see Koyama et al. (2008).] For a given function g of the state, let

$$q(x_k) = \log g(x_k) \mathrm{Pr}(\Delta N_k | x_k, H_k) p(x_k | \Delta N_{1:k-1})$$

and $\tilde{x}$ maximize m. The posterior expectation of g for the second-order approximation is then

$$\hat{E}[g(x_k)|\Delta N_k, H_k] = \frac{|-q''(\tilde{x})|^{-\frac{1}{2}} \exp[q(\tilde{x})]}{|-l''(\hat{x})|^{-\frac{1}{2}} \exp[l(\hat{x})]}. \tag{15}$$

When the g we care about is not necessarily positive, a simple and practical trick is to add a large constant c to g so that $g(x) + c > 0$, apply (15), and then subtract c. The posterior mean is thus calculated as $x_{k|k} = \hat{E}[x_k + c] - c$. See Tierney et al. (1989) for details of the method. The posterior variance is set to be $V_{k|k} = [-l''(x_{k|k})]^{-1}$, as this suffices for obtaining the second-order accuracy.

We should notice that the second-order Laplace's method is not a Taylor series expansion of the log posterior distribution around $x_{k|k}$; the approximated mean and variance are computed by Laplace's method, and the posterior distribution is replaced by the Gaussian with same mean and variance at each time-step. Combining with (7)–(8) we call this *Laplace-Gaussian filter* (LGF). Since the MAP point process filter provides the first-order approximation, we also call it the first-order LGF, distinguishing from the second-order LGF. Koyama et al. (2008) gives the complete description of these methods including theoretical analysis of them.

3.3 Smoothing

The smoothing problem refers to evaluating the distribution for the state vector at a given time, given past, and future observations. In particular, we are interested in the problem of fixed-interval smoothing, where spiking activity is recorded over an observation interval, $[0, t_k]$, and we wish to compute $p(x_t|\Delta N_{1:k})$ at all times t for $0 \leq t \leq t_k$. Using a Bayes' rule and the Chapman-Kolmogorov equation, we obtain the following expression for the smoothing distribution (Kitagawa and Gersh 1996):

$$\begin{aligned} p(x_k|\Delta N_{1:K}) &= \int p(x_k, x_{k+1}|\Delta N_{1:K})\mathrm{d}x_{k+1} \\ &= \int p(x_{k+1}|\Delta N_{1:K})p(x_k|x_{k+1}, \Delta N_{1:k})\mathrm{d}x_{k+1} \\ &= p(x_k|\Delta N_{1:k})\int \frac{p(x_{k+1}|\Delta N_{1:K})p(x_{k+1}|x_k)}{p(x_{k+1}|\Delta N_{1:k})}\mathrm{d}x_{k+1}. \end{aligned} \tag{16}$$

$p(x_k|\Delta N_{1:k})$ and $p(x_{k+1}|\Delta N_{1:k})$ in (16) corresponds to the posterior density (5) and the one-step prediction density expressed in (6), respectively. Thus in this framework, the first step to point process smoothing involves running a point process filter. In this case, the approximate Gaussian filters developed above have the additional advantage that the Gaussian approximations to the posterior filtering distributions lead to Gaussian densities for the smoothing distribution as well. Let $x_{k|K}$ and $V_{k|K}$ be the estimates of the mean and covariance of this Gaussian approximation to the smoothing distribution at time t_k. We then obtain the recursive smoothing equation corresponding to (16),

$$x_{k|K} = x_{k|k} + H_k(x_{k+1|K} - x_{k+1|k}), \tag{17}$$

$$V_{k|K} = V_{k|k} + H_k(V_{k+1|K} - V_{k+1|k})H_k^T, \tag{18}$$

where

$$H_t = V_{k|k}F_k V_{k+1|k}^{-1}. \tag{19}$$

The recursion for the smoothing mean and variance is backward, starting with $x_{K|K}$ and $V_{K|K}$, which are obtained as the last step of the point process filter, and stepping back one time step at each iteration until it reaches $x_{1|K}$ and $V_{1|K}$.

3.4 Stability of LGF

We have introduced the recursive Gaussian approximations for the posterior density above. By using the Laplace's method, we can achieve the second-order approximation for the posterior estimate with respect to the expansion parameter γ at each filtering step (the first-order approximation if we employ the MAP estimate). The accuracy of posterior approximation at one-step, however, does not guarantee the accuracy in whole simulation time-steps. Since the posterior density is approximated to the

Gaussian repeatedly, we must worry whether the approximation errors add up across time, leading to an explosive amplification of error; if the discrepancy between the actual filter and its Gaussian approximation after K time-steps is $O(K)$, for instance, this scheme will not be of much use. Here, we provide the theorems giving sufficient conditions to forbid such amplification of error over time, thereby allowing us to use the sequential Gaussian approximation with confidence. We state the simplified theorems for the state-space models with the Gaussian state evolution equation; those for more general cases and their complete proofs will be given elsewhere (Koyama et al. 2008). Let $h(x_k)$ be

$$h(x_k) = -\frac{1}{\gamma} \log p(\Delta N_k | x_k, H_k) p(x_k | \Delta N_{1:k-1}), \tag{20}$$

where γ is the expansion parameter which is taken so that $h(x_k)$ is a constant-order function of γ as $\gamma \to \infty$. In general, in state-space models γ would be interpreted in terms of sample size, the concentration of the observation density, and the inverse of the noise in the state dynamics. We assume suitable regularity conditions for the validity of posterior expansions based on Laplacefs method, and an inhomogeneity condition which prohibits ill-behaved, “explosive” trajectories in state space. The details of these conditions are given in Koyama et al. (2008).

Theorem 1 *Under suitable conditions, the accuracy of the approximated posterior mean computed by the* $\alpha(= 1, 2)$*-order LGF is*

$$x_{k|k} = \mathrm{E}(x_k | \Delta N_k, H_k) + O(\gamma^{-\alpha}), \tag{21}$$

for $k \in \mathbb{N}$*, and the error terms are bounded uniformly across time.*

This theorem is proved by calculating the asymptotic expansions of both the true and approximated posterior mean, and matching terms. The intuitive meaning of error non-amplification is that the Gaussian approximation introduces an $O(\gamma^{-\alpha})$ error into the posterior mean at each time step, while the errors propagated from the previous time-step shrink exponentially. The result is that the error in the posterior mean is dominated by the error due to the Gaussian approximation, but there is no accumulation of propagated errors.

Theorem 2 *Under suitable conditions, the accuracy of the approximated mean of the smoothing distribution computed by the* $\alpha(= 1, 2)$*-order LGF with (17)–(19) is*

$$x_{k|K} = \mathrm{E}(x_k | \Delta N_{1:K}) + O(\gamma^{-\alpha}) \tag{22}$$

for $k = 1, 2, \ldots, K, \;\; K \in \mathbb{N}$*, and the error terms are bounded uniformly across time.*

4 Illustration of applications

4.1 Motor cortical decoding

4.1.1 Model setup

To illustrate how the state-space framework we introduced so far is applied to solve the neural decoding problem, we use an example of motor cortical decoding. Many neurons in the motor cortex fire preferentially in response to the velocity $x_k \in \mathbb{R}^3$ of the hand during an arm movement task (Georgopoulos et al. 1986; Paninski et al. 2004). This observation suggests that the hand motion can be reconstructed from neural activity in the motor cortex, and many studies have applied this finding to the problem of building neuro-prosthetic devices (Chapin et.al 1999; Serruya et al. 2002; Taylor et al. 2002; Lebedev and Nicolelis 2006; Brockwell et al. 2007; Velliste et al. 2008). We consider N such neurons and take the conditional intensity function of the ith neuron to be

$$\lambda_i(x_k) = \exp(\alpha_i + \beta_i \cdot x_k), \tag{23}$$

where $\alpha_i \in \mathbb{R}^1$ sets the baseline firing rate, and $\beta_i \in \mathbb{R}^3$ is the preferred direction of the ith neuron. This model is made under the assumption that each neuron has firing properties that do not depend on its own spike history or that of any other neuron in the ensemble, and therefore this activity comprises an inhomogeneous Poisson process. The observation interval is partitioned into uniform bins with size of Δt. The state evolution equation is taken to be

$$x_k = x_{k-1} + \sigma \epsilon_k, \tag{24}$$

where ϵ_k a zero mean 3-D Gaussian random variable whose covariance matrix is given by the identity matrix.

The point process filters (the SSPPF, the first- and second-order LGF) are then constructed following Sect. 3.2. Note that for constructing the LGFs, we identify the expansion parameter γ for this model as

$$\gamma = \frac{1}{\sigma^2} + \Delta t \sum_{i=1}^{N} e^{\alpha_i} \|\beta_i\|^2.$$

We see that γ combines the number and the firing rate of neurons, the sharpness of neuronal tuning curves, and the noise in the state dynamics. The convergence criterion for the Newton–Raphson iterations for maximizing the log-posterior distribution at each time step is, then, set to be $\|x^{(l+1)} - x^{(l)}\| < \gamma^{-\alpha}$, $\alpha = 1$ for the first-order approximation and $\alpha = 2$ for the second-order approximation.

4.1.2 Simulation study

We first applied the filters to synthesized data. In the simulation of the model, we considered $K = 50$ time-steps, each of size $\Delta t = 0.05$. For each time point $k = 1, \ldots, K$,

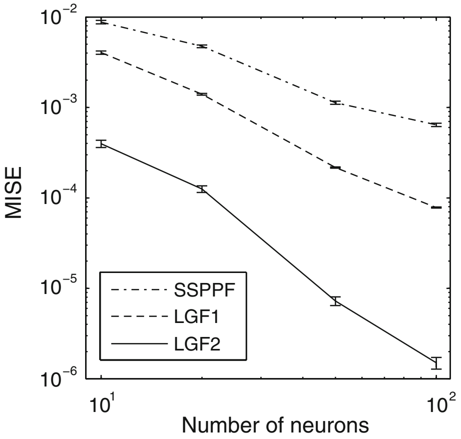

Fig. 1 MISE for approximating the actual posterior mean as a function of the number of neurons simulated. *LGF-1*, *-2* first- and second-order LGF, *SSPPF* stochastic state point process filter. The mean and standard error in these figures were computed with ten repetitions

the hand velocity was taken to be $x_k = (\sin \frac{2\pi}{K} k, \sin \frac{2\pi}{K} k, \cos \frac{2\pi}{K} k)$, K is the simulation time-interval. The spike trains were then drawn from the Poisson processes with the intensity function (23). We considered neurons with $\alpha_i = 2.5 + \mathcal{N}(0, 1)$, and β_i uniformly distributed on the unit sphere in $\mathbb{R}^3$. The value of σ^2 in (24) was determined by maximizing the likelihood function of the state model,

$$L(\sigma^2) = \prod_{k=1}^{K} (2\pi\sigma^2)^{-3/2} \exp\left[-\frac{1}{2\sigma^2} \|(x_k - x_{k-1})\|^2\right]. \tag{25}$$

In the simulation study, we used training data $\{x_k\}_{k=1}^{K}$ which consisted of $K = 100$ time-steps for estimation.

Figure 1 shows the filters' mean integrated squared error (MISE) in approximating the actual posterior mean as a function of the number of neurons. (Here, we obtained the actual posterior mean by using a particle filter (Kitagawa 1996; Doucet et al. 2001) with 10^6 particles and averaging over ten independent realizations.) The second-order LGF gives the best approximation, followed by the first-order LGF (the MAP point process filter) and SSPPF, as is expected from the construction of these filters. We note that the MISE between the true state and the actual posterior mean is 0.0957 ± 0.0044 for decoding 100 neurons, which is order of magnitude larger than the approximation errors. Most of the filtering error in estimating the true state is inherent statistical error of the posterior itself, and not due to the approximations. Thus, the SSPPF is sufficient for decoding the state process under this model setup.

4.1.3 Real data analysis

We applied the point process filters to data recorded from the motor cortex of a monkey executing a 3D reaching task. We first summarize the experimental design and data collection (Taylor et al. 2002; Velliste et al. 2008). A monkey was presented with a virtual 3-D space, containing eight possible targets located on the corners of a cube, and a cursor which was controlled by the subject's hand position. A multi-electrode

Table 1 The MISE in estimating the actual velocity.

	LGF2	LGF1	SSPPF	KF	PVA
MISE ($\times 10^{-2}$)	1.07 ± 0.06	1.08 ± 0.06	1.08 ± 0.06	1.11 ± 0.06	3.89 ± 0.16

LGF-1, *-2* first- and second-order LGF, *SSPPF* stochastic state point process filter. *KF* Kalman filter, *PVA* population vector algorithm. The mean and standard error were computed with 104 trials

array was implanted to record neural activity; raw voltage waveforms were thresholded and spikes were sorted to isolate the activity of individual cells. In all, 78 distinct neurons were recorded simultaneously. Our data set consisted of 104 trials. Each trial consist of time series of spike-counts from these neurons, along with the recorded hand positions, and hand velocities found by taking differences in hand position at successive $\Delta t = 0.03$ s intervals.

For decoding, we used the same observation model (23) and the AR(1) as the velocity model,

$$x_k = F x_{k-1} + \epsilon_k, \tag{26}$$

where $F \in \mathbb{R}^{3\times 3}$ and ϵ_k is a zero mean 3-D Gaussian random variable with $\mathrm{Var}(\epsilon_k) = Q$. The parameters in the intensity function of the neurons, α_i and β_i, were estimated by Poisson regression of spike counts on hand velocity. The AR coefficients in the state model were estimated by the Yule–Walker method. The time-lag between the hand movement and each neural activity was also estimated from the same training data. This was done by fitting a model over different values of time-lag ranging from 0 to $3\Delta t$. The estimated optimal time-lag was the value at which the model had the highest R^2. After the parameters were estimated, hand motions were reconstructed from spike trains. For comparison, we also reconstructed hand motions with a Kalman filter (KF) (Wu et al. 2005) and a population vector algorithm (PVA) (Dayan and Abbot 2001).

Table 1 gives the MISEs between filtered estimates and the actual cursor velocity; the point process filters (LGFs and SSPPF) and the KF are more accurate than the PVA. There is no substantial difference between the point process filters and the KF. Figure 2 illustrates a sample trajectory along with the decoded estimates. In Fig. 2b, we show the result obtained by applying the smoothing algorithm (17)–(19). As seen in this figure, the smoothed trajectory is less variable than that obtained by filtering.

Finally, we decoded the cursor velocity with the AR(2) process as the state model instead of (26) to examine if it improves the accuracy of estimation. The AR parameters were determined in the same way as before. The resulting MISE between the true velocity and filtered one by the LGF2 across 104 trials is $1.59 \pm 0.01 (\times 10^{-2})$, which is worse than that with the AR(1).

5 Discussion

We have described here a statistical framework for modeling and studying neural spike trains and their relation to behavior. At the center is the conditional intensity

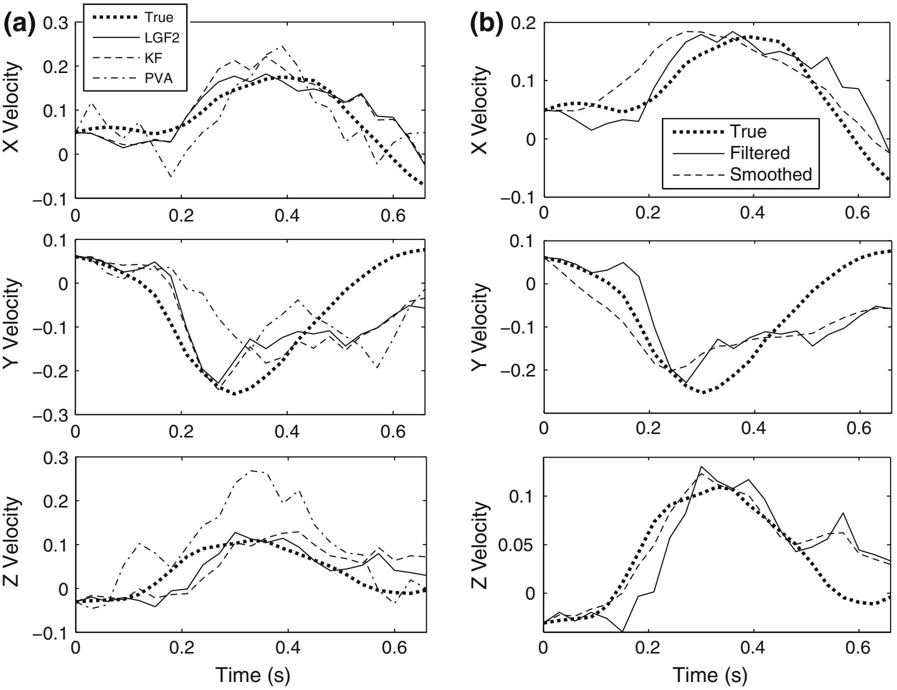

Fig. 2 An estimation on the velocity. **a** Filtering result. *True* true trajectory, *LGF2* second-order LGF, *KF* Kalman filter, *PVA* population vector algorithm. The decoded trajectories with the LGF1 and SSPPF are not shown in this figure (they are very similar to those decoded with LGF2). **b** Smoothing result. The LGF2 was used for filtering

function, which is the starting point for point process modeling. To reconstruct behavior from spike trains we have used state-space models. We have emphasized the use of approximate filters based on asymptotic approximations because we have found them to be effective. The stochastic state point process filter was obtained from the quadratic terms of the Taylor series expansion of the posterior distribution about the one-step prediction estimate of the state. The Laplace–Gaussian filters were obtained from first-order and second-order Laplace approximation to the posterior mean. The results in Sect. 4 demonstrated the accuracy of these methods. Other work by Eden et al. (2004) had also shown how decoding methods based on observed spike times can outperform rate-based approaches. We also note that although only the simple linear Gaussian state model was used in our data analysis, we can use more sophisticated models to improve the estimation in the same framework; such an attempt was made in Yu et al. (2007).

It is worth emphasizing the distinction between the point process filters and other approximate filters. Particle filtering approximates the integrals (5)–(6) stochastically. Just as the error of Laplace's method shrinks as $\gamma \rightarrow \infty$, so too does the error of particle filtering vanish as the number of particles grows. In fact, while γ for the LGF is set by the system, the number of particles can be made as large as desired. Schnatter (1992); Fruhwirth-Schnatter (1994) performed a sequential approximation of the posterior distribution by using numerical integration during the filtering process. At each time point the prior of the state was approximated by a Gaussian and carried out a Gauss-Hermite procedure. Durbin and Koopman (1997) used a Monte Carlo simulation taking a Gaussian centered by the posterior mode as the importance density. These methods can be regarded as an improvement of the approximate posterior mode estimator, and may provide a better estimation than the point process filters when the posterior density is far from Gaussian. The computational burden of these methods may, however, become a drawback in application to real-time neural decoding. For controlling neuroprosthetic devices such as a computer cursor or a robotic arm from neural activity on-line, computing time must be less than allowable control signal delay, which is roughly 30 ms (Velliste et al. 2008; Koyama et al. 2009). The computational cost of effective particle filtering or the numerical integration could quickly become prohibitive, and thus the Gaussian approximations could be preferable for such applications (Koyama et al. 2008).

An important application of point process filtering relates to the problem of tracking changes in the firing properties of neurons. This is of interest partly because electrophysiology experiments, and therefore the neural recordings that they generate, are inherently nonstationary. In addition, a neuron's response to a stimulus can change over time as a result of experience (Merzenich et al. 1983; Weinberger 1993; Edeline 1999; Kaas et al. 1999). Experience-dependent change (plasticity) within the adult brain has been well documented in a number of brain regions in both animals and humans. Place cells in the CA1 region of the hippocampus in the rat change their receptive field location, scale, and temporal firing characteristics as the animal explores its environment (Mehta et al. 2000). Neurons in the cercal system of the cricket may alter their firing in response to changes in wind velocity (Rieke et al. 1997). The firing properties of neurons in the primary motor cortex of primates changes with the load against which the animal must exert force (Li et al. 2001). Investigation of these receptive field dynamics is fundamental to our understanding of how these systems reflect information about the outside world. State-space models are well suited for describing these changes in neural firing properties. In order to apply these methods, conditional intensity models are constructed with parameters that are given by a dynamic state process characterized by an evolution equation as in (3). The output of the point process filters then provides a dynamic conditional intensity model descriptions that at each instant combines the estimates from the previous time step with the new spiking data. These methods can produce accurate instantaneous estimates even when only a few spikes are observed (Brown et al. 2001; Frank et al. 2002).

In this article, we reviewed a collection of discrete-time state space filters based on stochastic models for the evolution of the state variable and for the probability distribution of observing neural spiking outputs as a function of the value of the state variable. For the general state-space estimation problem, the models describing the state dynamics and observation processes can be cast in either a discrete-time or continuous-time setting (Snyder and Miller 1991; Solo 2000; Eden and Brown 2008a), leading to discrete or continuous time representations for state estimators, respectively. These two frameworks provide equivalent descriptions of the process being modeled, in that for any time point in the discrete time partition, the probability densities produced by the discrete and continuous time processes are identical. Both discrete and continuous time frameworks, respectively, have distinct advantages. For example, continuous valued expressions are generally more amenable to symbolic analyses, or common engineering methods such as frequency domain analysis. There are also direct physical or physiological interpretations for variables in continuous time setting. Nevertheless, continuous time descriptions for many dynamical systems require computations that can only be implemented in discrete time. Clearly, it is important to appreciate both points of view and to be able to exploit the particular advantages of each.

An important extension of the state-space methods is including multiple observations and state variables (Srinivasan et al. 2007; Eden and Brown 2008b). Although only the single point process observation was considered in this paper, we can potentially combine many sources of observations to improve the performance of neural decoding: spike trains, local field potentials (LFPs), electrocorticography (ECoG), electroencephalography (EEG) or electromyography (EMG). Whereas spiking activity at millisecond resolution is better described by point process observation, continuous field potentials are typically described by continuous valued (Gaussian) observation models. Likewise, the covariates which are related to neural activity would be not only a continuous variable such as hand kinematics, but also a discrete variable which may describe a subject's intension. These multiple states and observations could be combined in hybrid state-space models or in dynamic Bayesian networks. It is a challenging goal in neuroscience to read out higher cognitive states from many sources of brain activity (Haynes et al. 2007; Kay et al. 2008). We believe that the extension of the state-space framework presented in this article will provide a key methodology for these applications.

Acknowledgments This work was supported by grants RO1 MH064537, RO1 EB005847 and RO1 NS050256.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control, AC-19*, 716–723.

Akaike, H. (1994). Experiences on the development of time series models. In H. Bozdogan (Ed.), *Proceedings of the first US/Japan conference on the frontiers of statistical modeling: an informational approach* (pp. 33–42). Dordrecht: Kluwer. Reprinted in E. Parzen, K. Tanabe, G. Kitagawa (Eds.) (1998). *Selected papers of Hirotugu Akaike*. New York: Springer.

Barbieri, R., Quirk, M. C., Frank, L. M., Wilson, M. A., Brown, E. N. (2001). Construction and analysis on non-Poisson stimulus-response models of neural spiking activity. *Journal of Neuroscience Methods, 105*, 25–37.

Barbieri, R., Frank, L. M., Nguyen, D. P., Quirk, M. C., Solo, V., Wilson, M. A., Brown, E. N. (2004). Dynamic analyses of information encoding by neural ensembles. *Neural Computation, 16*, 277–307.

Berman, M. (1983). Comment on Likelihood analysis of point processes and its applications to seismological dat by Ogata. *Bulletin of the International Statistical institute, 50*, 412–418.

Brockwell, A. E., Kass, R. E., Schwartz, A. B. (2007). Statistical signal processing and the motor cortex. *Proceedings of the IEEE, 95*, 881–898.

Brown, E. N., Frank, L. M., Tang, D., Quirk, M. C., Wilson, M. A. (1998). A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells. *Journal of Neuroscience, 18*, 7411–7425.

Brown E. N., Nguyen D. P., Frank L. M., Wilson M. A., Solo V. (2001). An analysis of neural receptive field plasticity by point process adaptive filtering. *Proceedings of the National Academy of Sciences, 98*, 12261–12266.

Brown, E. N., Barbieri, R., Ventura, V., Kass, R. E., Frank, L. M. (2002). The time-rescaling theorem and its application to neural spike train data analysis. *Neural Computation, 14*, 325–346.

Brown, E. N., Barbieri, R., Eden, U. T., Frank, L. M. (2003). Likelihood methods for neural data analysis. In J. Feng (Ed.), *Computational Neuroscience: a comprehensive approach* (pp. 253–286). London: CRC.

Chapin, J. K., Moxon, K. A., Markowitz, R. S., Nicolelis, M. A. L. (1999). Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex. *Nature Neuroscience, 2*, 664–670.

Daley, D. J., Vere-Jones, D. (2003). *An introduction to the theory of point processes* (2nd ed.). New York: Springer.

Dayan, P., Abbot, L. F. (2001). *Theoretical neuroscience: computational and mathematical modeling of neural systems.* Cambridge: The MIT Press.

Doucet, A., de Nando, F., Gordon, N. (2001). *Sequential Monte Carlo methods in practice.* Berlin: Springer.

Durbin, J., Coopman, S. J. (1997). Monte Carlo maximum likelihood estimation for non-Gaussian state space models. *Biometrika, 84*, 669–684.

Edeline, J. M. (1999). Learning-induced physiological plasticity in the thalamo-cortical sensory sytems: a critical evaluation of receptive field plasticity, map changes and their potential mechanisms. *Progress in Neurobiology, 57*, 165–224.

Eden, U. T., Brown, E. N. (2008a). Continuous-time filters for state estimation from point process models of neural data. *Statistica Sinica, 18*, 1293–1310.

Eden, U. T., Brown, E. N. (2008b). Mixed observation filtering for neural data. In *33rd International conference on acoustics, speech, and signal processing, Las Vegas, NV*. March 30–April 4.

Eden, U. T., Frank, L. M., Barbieri, R., Solo, V., Brown, E. N. (2004). Dynamic analyses of neural encoding by point process adaptive filtering. *Neural Computation, 16*, 971–998.

Erdélyi, A. (1956). *Asymptotic expansions.* New York: Dover.

Frank, L. M., Eden, U. T., Solo, V., Wilson, M. A., Brown, E. N. (2002). Contrasting patterns of receptive field plasticity in the hippocampus and the entorhinalcortex: an adaptive filtering approach. *Journal of Neuroscience, 22*, 3817–3830.

Frank, L. M., Stanley, G. B., Brown E. N. (2004). Hippocampal plasticity across multiple days of exposure to novel environments. *Journal of Neuroscience, 24*, 7681–7689.

Fruhwirth-Schnatter, S. (1994). Applied State space modeling of non-Gaussian Time series using integration-based Kalman-filtering. *Statistics and Computing, 4*, 259–269.

Georgopoulos, A. B., Schwartz, A. B., Kettner, R. E. (1986). Neural population coding of movement direction. *Science, 233*, 1416–1419.

Hastie, T. J., Tibshirani, R. J. (1990). *Generalized additive models.* Florida: Chapman & Hall/CRC.

Haynes, J., Sakai, K., Rees, G., Gilbert, S., Firth, C., Passingham, R. E. (2007). Reading hidden intentions in the human brain. *Current Biology, 17*, 323–328.

Johnson, A., Kotz, S. (1970). *Distributions in statistics: continuous univariate distributions* (vol. 2). New York: Wiley.

Julier, S. J., Uhlmann, J. K. (1997). A new extension of the Kalman filter to nonlinear systems. In *The proceedings of aerosense: the 11th international symposium on aerospace/defense sensing, simulation and controls, multi sensor fusion, tracking and resource management II.*

Kaas, J. H., Florence, S. L., Jain, N. (1999). Subcortical contributions to massive cortical reorganizations. *Neuron, 22*, 657–660.

Kandel, E. R. (2000). *Principles of Neural Science* (4th ed.). New York: McGraw-Hill.

Kass, R. E., Raftery, A. E. (1995). Bayes factor. *Journal of the American Statistical Association, 90*, 773–795.

Kass, R. E., Ventura, V. (2001). A spike-train probability model. *Neural Computation, 13*, 1713–1720.

Kay, K. N., Naselaris, T., Prenger, R. J., Gallant, J. L. (2008). Identifying natural images from human brain activity. *Nature, 452*, 352–356.

Kitagawa, G. (1996). Monte Carlo flter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics, 5*, 1–25.

Kitagawa, G., Gersh, W. (1996). *Smoothness priors analysis of time series.* New York: Springer.

Koyama, S., Kass, R. E. (2008). Spike-train probability models for stimulus-driven leaky integrate-and-fire neurons. *Neural Computation, 20*, 1776–1795.

Koyama, S., Shinomoto, S. (2005). Empirical Bayes interpretations of random point events. *Journal of Physics A: Mathematical and General, 38*, L531–L537.

Koyama, S., Pérez-Bolde, L. C., Shalizi, C. R., Kass, R. E. (2008). Approximate methods for state-space models (submitted).

Koyama, S., Chase, S. M., Whitford, A. S., Velliste, M., Schwartz, A. B., Kass, R. E. (2009). Comparison of brain-computer interface decoding algorithms in open-loop and closed-loop control (submitted).

Lebedev, M. A., Nicolelis, A. L. (2006). Brain-machine interfaces: past, present and future. *Trends in Neuroscience, 29*, 536–546.

Li, C., Padoa-Schioppa, C., Bizzi, E. (2001). Neuronal correlates of motor performance and motor learning in the primary motor cortex of monkeys adapting to an external force field. *Neuron, 30*, 593–607.

McCullagh, P., Nelder, J. A. (1989). *Generalized linear models* (2nd ed.). New York: Chapman & Hall.

Mehta, M. R., Quirk, M. C., Wilson, M. A. (2000). Experience-dependent asymmetric shape of Hippocampal receptive fields. *Neuron, 25*, 707–715.

Merzenich, M. M., Kaas, J. H., Wall, J. T., Sur, M., Nelson, R. J., Felleman, D. J. (1984). Progression of change following median nerve section in the cortical representation of the hand in areas 3b and 1 in adult owl and squirrel monkeys. *Neuroscience, 10*, 639–665.

Ogata, Y. (1988). Statistical models for earthquake occurrences and residual analysis for point processes. *Journal of American Statistical Association, 83*, 9–27.

Paninski, L. (2004). Maximum likelihood estimation of cascade point-process neural encoding models. *Network: Computation in Neural Systems, 15*, 243–262.

Paninski, L., Fellows, M., Hatsopoulos, N., Donoghue, J. (2004). Spatiotemporal tuning properties for hand position and velocity in motor cortical neurons. *Journal of Neurophysiology, 91*, 515–532.

Papangelou, F. (1972). Integrability of expected increments of point processes and a related random change of scale. *Transactions of the American Mathematical Society, 165*, 483–506.

Pillow, J., Shlens, J., Paninski, L., Sher, A., Litke, A., Chichilnisky, E., Simoncelli, E. (2008). Spatiotemporal correlations and visual signaling in a complete neuronal population. *Nature, 454*, 995–999.

Reich, D. S., Victor, J. D., Knight, B. W. (1998). The power ratio and interval map: Spiking models and extracellular recordings. *Journal of Neuroscience, 18*, 10090–10104.

Rieke, F., Warland, D., de Ruyter van Steveninck, R. R., Bialek, W. (1997). *Spikes: Exploring the neural code*. Cambridge: MIT Press.

Schnatter, S. (1992). Integration-based Kalman-filtering for a dynamic generalized linear trend model. *Computational Statistics and Data Analysis, 13*, 447–459.

Schwartz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics, 6*, 461–464.

Schwartz, A. B. (2004). Cortical neural prosthetics. *Annual Review of Neuroscience, 27*, 487–507.

Serruya, M., Hatsopoulos, N. G., Paninski, L., Fellows, M. R., Donoghue, J. P. (2002). Brain-machine interface: instant neural control of a movement signal. *Nature, 416*, 141–142.

Smith, M. A., Kohn, A. (2008). Spatial and temporal scales of neuronal correlation in primary visual cortex. *Journal of Neuroscience, 28*, 12591–12603.

Snyder, D. L. (1972). *Random point processes.* New York: Wiley.

Snyder, D. L., Miller, M. I. (1991). *Random point processes in time and space.* New York: Springer.

Solo, V. (2000). Unobserved Monte Carlo method for identification of partially observed nonlinear state space systems, Part II: counting process observations. In *Proceedings of the 39th IEEE conference on decision and control* (pp. 3331–3336). Sydney, Australia.

Srinivasan, L., Eden, U. T., Mitter, S. K., Brown, E. N. (2007). General-purpose filter design for neural prosthetic devices. *Journal of Neurophysiology, 98*, 2456–2475.

Tanner, M. A. (1996). *Tools for statistical inference*. New York: Springer.

Taylor, D. M., Tillery, H., Stephen, I., Schwartz, A. B. (2002). Direct cortical control of 3D neuroprosthetic devices. *Science, 296*, 1829–1832.

Tierney, L., Kass R. E., Kadane, J. B. (1989). Fully exponential Laplace approximations to expectations and variances of nonpositive functions. *Journal of the American Statistical Association, 84*, 710–716.

Truccolo, W., Eden, U. T., Fellows, M. R., Donoghue, J. P., Brown, E. N. (2005). A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of Neurophysiology, 93*, 1074–1089.

Velliste, M., Perel, S., Spalding, M. C., Whitford, A. S., Schwartz, A. B. (2008). Cortical control of a prosthetic arm for self-feeding. *Nature*. doi:10.1038/nature06996.

Weinberger, N. M. (1993). Leaning-induced changes of auditory receptive fields. *Current Opinion in Neurobiology, 3*, 570–577.

Wu, W., Gao, Y., Biemenstock, E., Donoghue, J. P., Black, M. J. (2005). Bayesian population decoding of motor cortical activity using a Kalman filter. *Neural Computation, 18*, 80–118.

Yu, B. M., Kemere, C., Santhanam, G., Afshar, A., Ryu, S. I., Meng, T. H., Sahani, M., Shenoy, K. V. (2007) Mixture of trajectory models for neural decoding of goal-directed movements. *Journal of Neurophysiology, 97*, 3763–3780.