LETTER **Communicated by Emanuel Todorov**

Dynamic Analysis of Neural Encoding by Point Process Adaptive Filtering

Uri T. Eden
tzvi@neurostat.mgh.harvard.edu
Neuroscience Statistics Research Laboratory, Department of Anesthesia and Critical Care, Massachusetts General Hospital, Boston, MA 02114, and Division of Health Sciences and Technology, Harvard Medical School/Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.

Loren M. Frank
loren@neurostat.mgh.harvard.edu
Riccardo Barbieri
barbieri@neurostat.mgh.harvard.edu
Neuroscience Statistics Research Laboratory, Department of Anesthesia and Critical Care, Massachusetts General Hospital, Boston, MA 02114, U.S.A.

Victor Solo
vsolo@nmr.mgh.harvard.edu
School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, Australia, and Martinos Center for Biomedical Imaging, Massachusetts General Hospital, Harvard Medical School, Boston, MA 02114, U.S.A.

Emery N. Brown
brown@neurostat.mgh.harvard.edu
Neuroscience Statistics Research Laboratory, Department of Anesthesia and Critical Care, Massachusetts General Hospital, Boston, MA 02114, and Division of Health Sciences and Technology, Harvard Medical School/Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.

Neural receptive fields are dynamic in that with experience, neurons change their spiking responses to relevant stimuli. To understand how neural systems adapt their representations of biological information, analyses of receptive field plasticity from experimental measurements are crucial. Adaptive signal processing, the well-established engineering discipline for characterizing the temporal evolution of system parameters, suggests a framework for studying the plasticity of receptive fields. We use the Bayes' rule Chapman-Kolmogorov paradigm with a linear state equation and point process observation models to derive adaptive filters appropriate for estimation from neural spike trains. We derive point

process filter analogues of the Kalman filter, recursive least squares, and steepest-descent algorithms and describe the properties of these new filters. We illustrate our algorithms in two simulated data examples. The first is a study of slow and rapid evolution of spatial receptive fields in hippocampal neurons. The second is an adaptive decoding study in which a signal is decoded from ensemble neural spiking activity as the receptive fields of the neurons in the ensemble evolve. Our results provide a paradigm for adaptive estimation for point process observations and suggest a practical approach for constructing filtering algorithms to track neural receptive field dynamics on a millisecond timescale.

1 Introduction

Experience-dependent plasticity has been well documented in a number of brain regions in both animals and humans (Kaas, Merzenich, & Killackey, 1983; Merzenich, Nelson, Stryker, Cyader, Schoppmann, & Zook, 1984; Donoghue, 1995; Weinberger, 1993; Pettet, 1992; Jog, Kubota, Connolly, Hillegaart, & Graybiel, 1999). In rats, for example, place cells recorded in the CA1 region of the hippocampus change their receptive field location, scale, and temporal firing characteristics as the animal explores familiar environments (Mehta, Barnes, & McNaughton, 1997; Mehta, Quirk, & Wilson, 2000; Frank, Eden, Solo, Wilson, & Brown, 2002). Similarly, the firing properties of neurons in the primary motor cortex of primates change with the load against which the animal must exert force (Gandolfo, Li, Benda, Schioppa, & Bizzi, 2000; Li, Padoa-Schioppa, & Bizzi, 2001). Characterizing this receptive field plasticity is important for understanding how neural activity represents biological information.

To study the plasticity of receptive fields, we turn to adaptive signal processing, a well-established engineering discipline for characterizing the temporal evolution of system parameters (Solo & Kong, 1995; Haykin, 1996). Several well-known filtering algorithms that serve this purpose, such as the Kalman filter, recursive least-squares, and steepest-descent algorithms, fall into a class of filters that can be derived from Bayesí rule and the Chapman-Kolmogorov equations. But because these algorithms require continuous-valued measurements, they are of limited utility in the analysis of neural plasticity, where the spike train observations are point processes. We previously introduced a steepest-descent point process adaptive filter that used spike train measurements to analyze neural plasticity (Brown, Nguyen, Frank, Wilson, & Solo, 2001). This adaptive filter was derived from an instantaneous log-likelihood criterion function and was used in a detailed study of receptive field plasticity in hippocampal CA1 and entorhinal cortical neurons (Frank et al., 2002) to track the evolution of these receptive fields on a millisecond timescale. Those analyses suggested a number of aspects of point process filtering for further development. In particular, those filters used a static learning rate and did not allow calculation of confidence

bounds for the parameter estimates. Furthermore, the relation of the point process filters and well-established adaptive filters for continuous-valued data was not discussed.

Here we extend our point process adaptive filtering framework by deriving a new set of filters using the Bayesí rule Chapman-Kolmogorov paradigm with a linear state equation and point process observation models. This extends the derivation in Brown et al. (2001) and allows us to derive point process analogues of the Kalman filter, recursive least squares, and steepest-descent algorithms. We discuss the properties of the new filters and illustrate their performance in two simulated data examples. The first is a study of both linear and jump evolution of place receptive fields in hippocampal neurons. The second is an adaptive decoding study in which a signal is decoded from ensemble neural spiking activity as the receptive field properties of the neurons in the ensemble evolve and this evolution is simultaneously estimated.

2 Theory

2.1 The Conditional Intensity Function. Given an observation interval $(0, T]$, let $N(t)$ be the counting process giving the total number of spikes fired by a given neuron in the interval $(0, t]$, for $t \in (0, T]$. A stochastic point process representing neural spike data can be characterized by its conditional intensity function, $\lambda(t \mid \mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{H}(t))$, where $\mathbf{x}(t)$ is a vector representing the biological signal to which the neural system is tuned, $\boldsymbol{\theta}(t)$ is a vector representing the tuning function parameters for this neuron, and $\mathbf{H}(t)$ denotes the history of the spiking process and the biological signal up to time t (Snyder & Miller, 1991). This conditional intensity function defines the neuron's instantaneous firing rate in terms of the instantaneous probability of spiking as

$$\lambda(t \mid \mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{H}(t)) = \lim_{\Delta t \to 0} \frac{\Pr(N(t + \Delta t) - N(t) = 1 \mid \mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{H}(t))}{\Delta t}. \tag{2.1}$$

Thus, the probability of the neuron's firing a single spike in a small interval $[t, t + \Delta t)$ can be approximated as $\lambda(t \mid \mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{H}(t))\Delta t$. This conditional intensity function is a history-dependent generalization of the inhomogeneous Poisson rate function (Daley & Vere-Jones, 1988).

Here, we switch from a continuous time to a discrete time framework and present our methods as recursive algorithms rather than as differential equations in order to simplify ensuing calculations. The observation interval can be partitioned into $\{t_k\}_{k=0}^{K}$ and individual time steps represented by $\Delta t_k = t_k - t_{k-1}$. The continuous-valued state, parameter, and firing history functions defined above are the same in discrete time, but are now indexed

by an integer k representing the associated time step in the partition. For example, we use x_k for $x(t_k)$, N_k for $N(t_k)$, and so forth. $\Delta N_k = N_k - N_{k-1}$ is then the new spiking information observed during the time interval $(t_{k-1}, t_k]$. If Δt_k is sufficiently small, then the probability of more than one spike occurring in this interval is negligible. In that case, ΔN_k takes on the value 0 if there is no spike in $(t_{k-1}, t_k]$ or 1 if there is a spike. For this reason, we refer to ΔN_k as the spike indicator function for the neuron at time t_k.

We let $\Delta \mathbf{N}_{1:k} = [\Delta N_1, \ldots, \Delta N_k]$ denote the ensemble spike observations up to time t_k. For a fine partition, this is the sequence of 0s and 1s that is typically used to represent a spike train in discrete time. To simplify notation, we aggregate all of the history that is pertinent to the firing probability at t_k into a common term $\mathbf{H}_k = [\boldsymbol{\theta}_{1:k-1}, \mathbf{x}_{1:k}, \mathbf{N}_{1:k-1}]$.

2.2 Derivation of the Point Process Adaptive Filter. We assume that $\boldsymbol{\theta}_k$ is a time-varying parameter vector to be estimated at each discretized point in time. To develop an adaptive filter, a recursive expression for $\boldsymbol{\theta}_k$ is derived in terms of its previous value and the state and spiking histories. Therefore, estimates of $\boldsymbol{\theta}_k$ will be based on its posterior density, conditioned on the set of past observations, $p(\boldsymbol{\theta}_k \mid \Delta \mathbf{N}_k, \mathbf{H}_k)$. This posterior density evolves with time and with each observation. Tracking this evolution allows for real-time estimation of the parameters of the conditional intensity function based on all of the spiking data observed up to the current time.

2.2.1 The System and Observation Equations. We define the system equation as a set of parameter vectors with linear evolution processes and gaussian errors,

$$\boldsymbol{\theta}_{k+1} = \mathbf{F}_k \boldsymbol{\theta}_k + \eta_k, \tag{2.2}$$

where $\mathbf{F}_k$ is a system evolution matrix and η_k is a zero-mean white-noise process with covariance matrix $\mathbf{Q}_k$. If $\boldsymbol{\theta}_0$ and η_k have gaussian distributions, then each future $\boldsymbol{\theta}_k$ will also have a gaussian distribution with its own mean and variance.

The second essential component for the construction of a recursive estimation equation is the likelihood or observation model, $p(\Delta N_k \mid \boldsymbol{\theta}_k, \mathbf{H}_k)$. This model defines the probability of observing spikes in the interval $(t_{k-1}, t_k]$, based on the current state of the system and the past spiking information. From the conditional intensity function in equation 2.1, we obtain an expression for the probability of observing a spike in the interval $(t_{k-1}, t_k]$, $\Pr(\Delta N_k) = [\lambda(t_k \mid \boldsymbol{\theta}_k, \mathbf{H}_k)\Delta t_k]^{\Delta N_k}[1 - \lambda(t_k \mid \boldsymbol{\theta}_k, \mathbf{H}_k)\Delta t_k]^{1-\Delta N_k} + o(\Delta t_k)$. For small values of Δt_k, this Bernoulli probability expression is well approximated as

$$\begin{aligned}&\Pr(\Delta N_k \text{ spikes in } (t_{k-1}, t_k] \mid \boldsymbol{\theta}_k, \mathbf{H}_k)\\ &\quad = \exp(\Delta N_k \log(\lambda(t_k \mid \boldsymbol{\theta}_k, \mathbf{H}_k)\Delta t_k) - \lambda(t_k \mid \boldsymbol{\theta}_k, \mathbf{H}_k)\Delta t_k)\end{aligned} \tag{2.3}$$

(Brown, Barbieri, Eden, & Frank, 2003). Bayes' rule can be used to obtain the posterior density for the parameter vector at time t_k as a function of a prediction density of the state at time t_k given the observations only up to time t_{k-1}, and the probability of firing a spike in the interval $(t_{k-1}, t_k]$,

$$p(\boldsymbol{\theta}_k \mid \Delta N_k, \mathbf{H}_k) = \frac{p(\Delta N_k \mid \boldsymbol{\theta}_k, \mathbf{H}_k) p(\boldsymbol{\theta}_k \mid \mathbf{H}_k)}{p(\Delta N_k \mid \mathbf{H}_k)}. \tag{2.4}$$

The first term in the numerator of equation 2.4 is the likelihood in equation 2.3, and the second term is the one-step prediction density defined by the Chapman-Kolmogorov equation as

$$p(\boldsymbol{\theta}_k \mid \mathbf{H}_k) = \int p(\boldsymbol{\theta}_k \mid \boldsymbol{\theta}_{k-1}, \mathbf{H}_k) p(\boldsymbol{\theta}_{k-1} \mid \Delta N_{k-1}, \mathbf{H}_{k-1})\, d\boldsymbol{\theta}_{k-1}. \tag{2.5}$$

Equation 2.5 has two components, $p(\boldsymbol{\theta}_k \mid \boldsymbol{\theta}_{k-1}, \mathbf{H}_k)$, given by the state evolution model in equation 2.2, and $p(\boldsymbol{\theta}_{k-1} \mid \Delta N_{k-1}, \mathbf{H}_{k-1})$, the posterior density from the last iteration step. The denominator in equation 2.4 defines a normalizing factor that ensures that the posterior distribution will integrate to 1.

Substituting equation 2.5 into equation 2.4 yields the recursive expression for the evolution of the posterior density of the time-varying parameter vector,

$$p(\boldsymbol{\theta}_k \mid \Delta N_k, \mathbf{H}_k) = \lambda(t_k \mid \boldsymbol{\theta}_k, \mathbf{H}_k) \Delta t_k^{\Delta N_k} \exp(-\lambda(t_k \mid \boldsymbol{\theta}_k, \mathbf{H}_k) \Delta t_k) \times \frac{\int p(\boldsymbol{\theta}_k \mid \boldsymbol{\theta}_{k-1}, \mathbf{H}_k) p(\boldsymbol{\theta}_{k-1} \mid \Delta N_{k-1}, \mathbf{H}_{k-1})\, d\boldsymbol{\theta}_{k-1}}{p(\Delta N_k \mid \mathbf{H}_k)}. \tag{2.6}$$

Equation 2.6 is the exact posterior density equation. It allows us to compute the evolution of the density of the state vector, conditioned on all the spike observations made up to that time. There are a number of ways to proceed in evaluating equation 2.6, depending on the desired balance between accuracy and computational efficiency. In this letter, a gaussian approximation is made to this posterior distribution at each point in time. This leads to a simple algorithm that is computationally tractable as the dimension of the parameter vector increases.

By assumption, the state evolution in equation 2.2 has a gaussian random component, so that $p(\boldsymbol{\theta}_k \mid \boldsymbol{\theta}_{k-1})$ is a gaussian function in the variable $\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1}$. If the posterior density at the last time step is approximated by a gaussian, then equation 2.5 is the convolution of two gaussian curves, and hence the one-step prediction distribution, $p(\boldsymbol{\theta}_{k-1} \mid \mathbf{H}_k)$, must be gaussian itself.

Let $\boldsymbol{\theta}_{k|k-1} = \mathrm{E}[\boldsymbol{\theta}_k \mid \mathbf{H}_k]$ and $\mathbf{W}_{k|k-1} = \mathrm{var}[\boldsymbol{\theta}_k \mid \mathbf{H}_k]$ be the mean and variance for the one-step prediction density and $\boldsymbol{\theta}_{k|k} = \mathrm{E}[\boldsymbol{\theta}_k \mid \Delta N_k, \mathbf{H}_k]$ and $\mathbf{W}_{k|k} = \mathrm{var}[\boldsymbol{\theta}_k \mid \Delta N_k, \mathbf{H}_k]$ be the mean and variance for the posterior density. The goal in this derivation is to obtain a recursive expression for the

posterior mean, $\boldsymbol{\theta}_{k|k}$, and the posterior variance, $\mathbf{W}_{k|k}$, in terms of observed and previously estimated quantities.

As detailed in the appendix, by writing a quadratic expansion of the log of the posterior density in equation 2.6 about $\boldsymbol{\theta}_{k|k-1}$, the following recursive algorithm for updating the posterior mean and variance can be obtained,

$$\boldsymbol{\theta}_{k|k-1} = \mathbf{F}_k \boldsymbol{\theta}_{k-1|k-1}, \tag{2.7}$$

$$\mathbf{W}_{k|k-1} = \mathbf{F}_k \mathbf{W}_{k-1|k-1} \mathbf{F}_k' + \mathbf{Q}_k, \tag{2.8}$$

$$(\mathbf{W}_{k|k})^{-1} = (\mathbf{W}_{k|k-1})^{-1} + \left[\left(\frac{\partial \log \lambda}{\partial \boldsymbol{\theta}_k} \right)' [\lambda \Delta t_k] \left(\frac{\partial \log \lambda}{\partial \boldsymbol{\theta}_k} \right) - (\Delta N_k - \lambda \Delta t_k) \frac{\partial^2 \log \lambda}{\partial \boldsymbol{\theta}_k \partial \boldsymbol{\theta}_k'} \right]_{\boldsymbol{\theta}_{k|k-1}}, \tag{2.9}$$

$$\boldsymbol{\theta}_{k|k} = \boldsymbol{\theta}_{k|k-1} + \mathbf{W}_{k|k} \left[\left(\frac{\partial \log \lambda}{\partial \boldsymbol{\theta}_k} \right)' (\Delta N_k - \lambda \Delta t_k) \right]_{\boldsymbol{\theta}_{k|k-1}}, \tag{2.10}$$

where, for simplicity, λ denotes $\lambda(t_k \mid \mathbf{x}_k, \boldsymbol{\theta}_k, \mathbf{H}_k)$ and λ and its derivatives are evaluated at $\boldsymbol{\theta}_{k|k-1}$. Equations 2.7 through 2.10 comprise a recursive point process adaptive filter. Equations 2.7 and 2.8 are the one-step prediction equations for the parameter vector mean and covariance, respectively, corresponding to the distribution on the left-hand side of equation 2.5, whereas equations 2.10 and 2.9 are the posterior mean and variance equations corresponding to the distribution in equation 2.4. Because this adaptive estimation algorithm tracks the evolution of a stochastic state or parameter vector using point process observations, we will refer to it as a stochastic state point process filter (SSPPF).

2.3 Properties of the Algorithm

2.3.1 Innovations. The term $(\Delta N_k - \lambda(t_k \mid \boldsymbol{\theta}_{k|k-1}, \mathbf{H}_k)\Delta t_k)$ in equation 2.10 is the point process version of the innovation vector from standard adaptive filter theory (Daley & Vere-Jones, 1988; Haykin 1996; Brown, Frank, Tang, Quirk, & Wilson, 1998). For short intervals, ΔN_k is the spike indicator function, and $\lambda(t_k \mid \boldsymbol{\theta}_{k|k-1}, \mathbf{H}_k)\Delta t_k$ is approximately the probability of a spike in the interval $(t_{k-1}, t_k]$. Therefore, at each step of the recursive update algorithm, the innovation can be thought of as the difference between the probability of observing a spike given the current model estimate and whether one is actually observed. At all observation times when a spike does occur, the parameter estimates are adjusted to increase the instantaneous firing rate estimate, and at all nonspike observation times, they are adjusted to decrease it. Thus, this adaptive estimation procedure uses information from both spiking and nonspiking intervals to compute its receptive field estimate.

2.3.2 Learning Rate. Each algorithm uses the innovation vector as the basic error signal between the instantaneous rate estimate of the receptive field model and the current spiking activity. The term that modulates the innovation vector controls the learning rate of the algorithm. This is analogous to the gain on a standard adaptive filter and is determined by the choice of a criterion function or the method of derivation (Haykin, 1996).

In the case of the SSPPF, the learning rate is adaptive and is proportional to the one-step prediction variance. In this manner, it is similar to the Kalman gain for a standard Kalman filter. As with the Kalman gain, this learning rate uses the statistical properties of the underlying process to characterize the uncertainty of its estimates. One important feature of the standard Kalman filter is that the recursive update equation for the Kalman gain is independent of the observations and can therefore be computed prior to filtering. The posterior variance in equation 2.9, on the other hand, contains a term relating to the current firing rate estimate and another term relating to the innovation, neither of which is known in advance, and so it must be computed during filtering.

2.3.3 Confidence Intervals. Since we estimate the covariance of the parameter vector at each point in time, we can construct confidence intervals about the parameter estimates. If the observation and state models are correct, the $1-\alpha$ confidence interval provides bounds that contain the true parameter value with probability $1-\alpha$. For our gaussian posterior densities, the 99% confidence interval for the ith component of the parameter vector is given by $\theta^i_{k|k} \pm 2.475(W^{i,i}_{k|k})^{1/2}$, where $W^{i,i}_{k|k}$ is the ith diagonal element of the posterior covariance estimate.

2.4 Related Algorithms. As is true for the Kalman filter, we can derive two additional filters as special cases of the SSPPF algorithm.

2.4.1 A Point Process Analogue of the RLS Filter. If the state evolution noise η_k in equation 2.2 is zero, that is, if the state evolution is deterministic rather than stochastic, then the SSPPF reduces to an analogue of the well-known recursive least squares (RLS) filter (Haykin, 1996). The resulting point process algorithm is

$$\boldsymbol{\theta}_{k|k-1} = \mathbf{F}_k \boldsymbol{\theta}_{k-1|k-1}, \tag{2.11}$$

$$(\mathbf{W}_{k|k})^{-1} = (\mathbf{F}_k \mathbf{W}_{k-1|k-1} \mathbf{F}_k')^{-1} + \left[\left(\frac{\partial \log \lambda}{\partial \boldsymbol{\theta}_k} \right)' [\lambda \Delta t_k] \left(\frac{\partial \log \lambda}{\partial \boldsymbol{\theta}_k} \right) - (\Delta N_k - \lambda \Delta t_k) \frac{\partial^2 \log \lambda}{\partial \boldsymbol{\theta}_k \partial \boldsymbol{\theta}_k'} \right]_{\boldsymbol{\theta}_{k|k-1}}, \tag{2.12}$$

$$\boldsymbol{\theta}_{k|k} = \boldsymbol{\theta}_{k|k-1} + \mathbf{W}_{k|k} \left[\left(\frac{\partial \log \lambda}{\partial \boldsymbol{\theta}_k} \right)' (\Delta N_k - \lambda \Delta t_k) \right]_{\boldsymbol{\theta}_{k|k-1}}. \tag{2.13}$$

This filter is applicable in situations in which the receptive field parameters undergo deterministic linear evolution.

2.4.2 A Steepest Descent Point Process Filter (SDPPF). If we replace the adaptive learning rate term in the posterior state equation by a constant matrix ε and set $\mathbf{F}_k = \mathbf{I}$, we can derive a point process analogue of the method of steepest descent (Haykin, 1996; Brown et al., 2001). This algorithm eliminates the variance update equations 2.8 and 2.9 and thus becomes

$$\boldsymbol{\theta}_{k|k} = \boldsymbol{\theta}_{k-1|k-1} + \left[\varepsilon \left(\frac{\partial \log \lambda}{\partial \boldsymbol{\theta}_k}\right)' (\Delta N_k - \lambda \Delta t_k)\right]_{\boldsymbol{\theta}_{k-1|k-1}}. \tag{2.14}$$

We have kept the

$$\left.\frac{\partial \log \lambda}{\partial \boldsymbol{\theta}_k}\right|_{\boldsymbol{\theta}_{k-1|k-1}}$$

term in the update equation. This term serves to scale the innovation so that updates in $\boldsymbol{\theta}$ cause appropriately sized changes in the predicted firing rate.

In order to apply a steepest-descent filter, it is first necessary to choose an appropriate learning rate matrix. A diagonal matrix is often chosen with individual elements representing the learning rates of the individual parameters. In some cases, a good conjecture for these values can be suggested by the nature of the coding problem. When a training data set is available and the values of the parameters are already well estimated, an appropriate matrix can be estimated either by minimizing some error function over this data set or running the full SSPPF on the training data set and then averaging the resulting adaptive gain matrix, $\bar{\mathbf{W}} = K^{-1} \sum_{k=1}^{K} \mathbf{W}_{k|k}$, and using either this full matrix or just its diagonal terms.

This algorithm is computationally simpler than the SSPPF because it requires no matrix inversions. What is lost in this simplification is the flexibility associated with an adaptive learning rate, as well as the measure of uncertainty that the posterior variance estimate provides.

2.5 Goodness-of-Fit Measures. Point process observation models based on the conditional intensity function offer a framework for measuring the goodness-of-fit between the model and the observed spike trains. The Kolmogorov-Smirnov (KS) statistic is one such measure. As described in Brown, Barbieri, Ventura, Kass, and Frank (2002), if $\lambda(t_k \mid \mathbf{x}_k, \boldsymbol{\theta}_k, \mathbf{H}_k)$ correctly describes the rate of the process generating the spike train, then the set of rescaled interspike intervals (ISIs) $z_i = 1 - \exp(\int_{l_{i-1}}^{l_i} \lambda(u \mid \theta_u, H_u) du)$, where l_i is the time of the ith spike, should be independently distributed according to a uniform density. The KS statistic is then the maximum absolute deviation between the order statistics of these rescaled ISIs and the cumulative density of a uniform. It ranges from 0 to 1, with lower values representing

better fit to the data. To provide meaning to this number, we generally compute the 95% confidence intervals to the KS statistic by $1.36 \cdot N^{-1/2}$, where N is the number of observed ISIs. If the estimated model is sufficiently close to the one that produced the spike train, then the KS statistic should be less than this confidence bound with probability 0.95. In addition, in our simulation studies, we compute the mean squared error (MSE) between the true and estimated parameter trajectories.

3 Applications

3.1 Dynamic Analysis of Spatial Receptive Fields in the CA1 Hippocampal Pyramidal Neurons. In this example, we examine the simulated response of pyramidal cells in the CA1 region of the rat hippocampus to the movement of the animal on a linear track. These neurons have place receptive fields in that a neuron fires only when the animal is in certain regions of space (O'Keefe & Dostrovsky, 1971), and these place fields evolve during the course of an experiment, even when the animal is in a familiar environment (Mehta et al., 1997, 2000; Frank, Brown, & Wilson, 2000). In previous work (Brown et al., 2001), we demonstrated the feasibility of tracking place field evolution using an SDPPF. In the current analysis of two simulated place field evolution scenarios, we compare the performance of the SSPPF, the SDPPF, a pass-by-pass method, and an extended Kalman filter algorithm using an empirical firing rate estimate.

The receptive field model is identical to the one of our previous analyses (Brown et al., 2001),

$$\lambda(t_k \mid \boldsymbol{\theta}_k) = \exp\left\{\alpha_k - \frac{(x_k - \mu_k)^2}{2\sigma_k^2}\right\}, \tag{3.1}$$

where x_k is the position of the animal on the linear track at time t_k, μ_k is the center of the place field, σ_k is the scale, $\exp(\alpha_k)$ is the maximum firing rate, and $\boldsymbol{\theta}_k = [\alpha_k \quad \mu_k \quad \sigma_k]^T$. In each of the two evolution scenarios we studied, the parameter vector evolved from an initial value of $[\alpha \quad \mu \quad \sigma]_0 = [\log(10) \quad 250 \quad 12^{1/2}]$ to a final value $[\alpha \quad \mu \quad \sigma]_{800} = [\log(30) \quad 150 \quad 20^{1/2}]$ over 800 seconds. The values of the initial and final parameter vectors are based on observed changes in actual hippocampal place field properties (Mehta et al., 1997, 2001; Frank et al., 2002). In the first evolution scenario, each parameter evolved linearly, as in Brown et al. (2001). The resulting receptive field was a slowly evolving gaussian curve that steadily grew in maximum firing rate and increased in scale, as its center migrated at a constant velocity. This scenario is consistent with the types of changes observed when a rat executes a behavioral task in a familiar linear environment (Mehta et al., 1997, 2001). In the second scenario, the parameters remained constant at their initial values for 400 seconds, after which they jumped instantaneously to their final values, yielding a sudden change in

the location, scale, and height of the gaussian place field. This evolution scenario is motivated by recent studies of learning in the rat while recording from the hippocampus that have suggested that place fields can appear or transform very rapidly (Frank, Stanley, & Brown, 2003).

We simulated the movement of a rat running back and forth along a 300 cm linear track at a speed of 125 cm per second. These position values, along with the parameter vectors, $\boldsymbol{\theta}_k$, determined the firing rate at each time point using equation 3.1 under an inhomogeneous Poisson process model. The spike trains for the linear and jump evolution place fields were generated using the time-rescaling algorithm in Brown et al. (2002). Figure 1 shows the simulated spike trains for each of the place field evolution scenarios.

Our goal was to estimate the temporal evolution of the place fields from the observed spike trains. In order to apply the SSPPF to this problem, we first define a state model for the evolution of parameter vector. For simplicity, we used a random walk for the state evolution model of the parameter vector in equation 2.2 by setting $\mathbf{F}_k = \mathbf{I}$, letting $\mathbf{Q}_k$ be the diagonal matrix, $\mathbf{Q}_k = Diag(10^{-5}, \quad 10^{-3}, \quad 10^{-4})$, and using a Δt_k of 20 msec. The covariance parameters were selected based on the maximum rate of change observed in actual hippocampal place receptive fields (Mehta et al., 1997; Brown et al., 2001).

We applied the SSPPF in equations 2.7 through 2.10 with the first and second derivative terms evaluated as

$$\left.\frac{\partial \log \lambda}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}_{k|k-1}} = [1 \quad \sigma_{k|k-1}^{-2}(x(t_k) - \mu_{k|k-1}) \quad \sigma_{k|k-1}^{-3}(x(t_k) - \mu_{k|k-1})^2]', \quad (3.2)$$

$$\left.\frac{\partial^2 \log \lambda}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'}\right|_{\boldsymbol{\theta}_{k|k-1}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\sigma_{k|k-1}^{-2} & -2\sigma_{k|k-1}^{-3}(x(t_k) - \mu_{k|k-1}) \\ 0 & -2\sigma_{k|k-1}^{-3}(x(t_k) - \mu_{k|k-1}) & -3\sigma_{k|k-1}^{-4}(x(t_k) - \mu_{k|k-1})^2 \end{bmatrix}. \quad (3.3)$$

Next, we applied the SDPPF in equation 2.14, letting ε be a time-invariant diagonal 3×3 matrix with diagonal elements $\varepsilon_\alpha = 0.02$, $\varepsilon_\mu = 10$, and $\varepsilon_\sigma = 1$, representing the learning rates for the α, μ, and σ parameters, respectively. These parameters were selected by minimizing the mean-squared error between the true and estimated receptive field parameters over a simulated training data set.

We also used a pass-by-pass approach, in which a histogram of the spiking activity of the neuron over one back-and-forth pass through the environment was used to compute a static estimate of the receptive field. To implement the pass-by-pass method, on each pass we collected the spikes in 1 cm bins and let x_b and N_b give the location and the number of spikes fired in the bth bin, respectively. The maximum firing rate was computed as $e^{\hat{\alpha}} =$

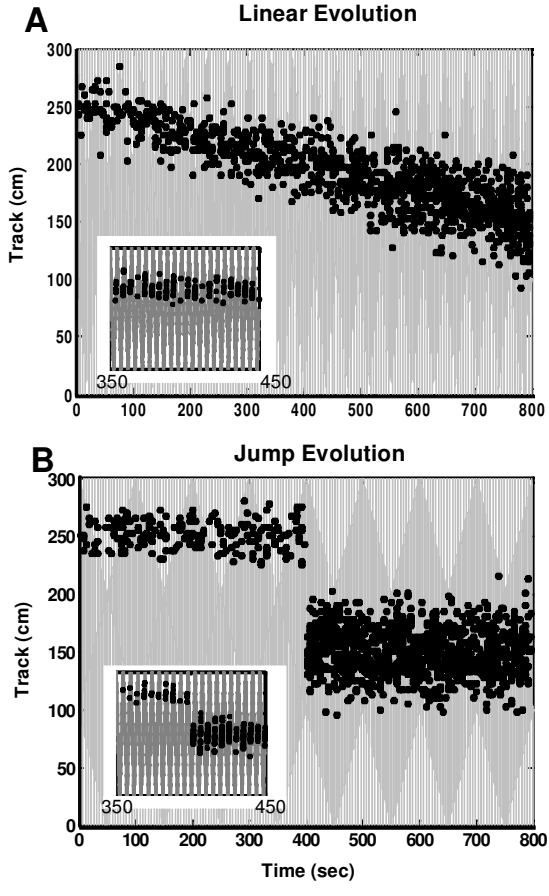


Figure 1: Simulated spiking activity of a rat hippocampal place cell with (A) linear and (B) jump evolution in place field parameters. The simulated animal ran back and forth on a 300 cm linear track. The place cells are directional in that they fire as the animal passes through its place field in one direction but not in the other (see the insets). Due to the stochastic nature of the spiking activity, the number of spikes fired can change appreciably from one pass to the next, even if there is little or no change in the place field.

$T_{pass}^{-1} \sum_b N_b$, the place field center was estimated as $\hat{\mu} = (\sum_b N_b)^{-1} \cdot \sum_b x_b N_b$, and the place field scale factor as $\hat{\sigma}^2 = (\sum_b N_b)^{-1} \cdot \sum_b (x_b - \hat{\mu})^2 N_b$, where T_{pass} is the time required to complete one full back and forth pass. For this algorithm, there was no continuous time representation of the place field and therefore no continual updates of the place field estimates. Instead, the place field parameters were estimated only once, after the rat completed a

back-and-forth run along the track. Previous analyses of place field plasticity have been conducted with a pass-by-pass method (Mehta et al., 1997, 2000).

To construct the extended Kalman filter, we computed a continuous firing rate estimate by convolving the observed spike train with a normalized half-gaussian curve with a fixed width of 0.25 seconds. This convolution kernel was chosen to preserve the causality of the relation between incoming spiking activity and the firing rate estimate. The extended Kalman filter was defined as

$$\mathbf{W}_{k|k}^{-1} = (\mathbf{W}_{k-1|k-1} + \mathbf{Q}_k)^{-1} + \left[\left(\frac{\partial \log \lambda}{\partial \boldsymbol{\theta}}\right)' (\lambda \Delta t_k) \left(\frac{\partial \log \lambda}{\partial \boldsymbol{\theta}}\right)\right]_{\boldsymbol{\theta}_{k-1|k-1}}, \tag{3.4}$$

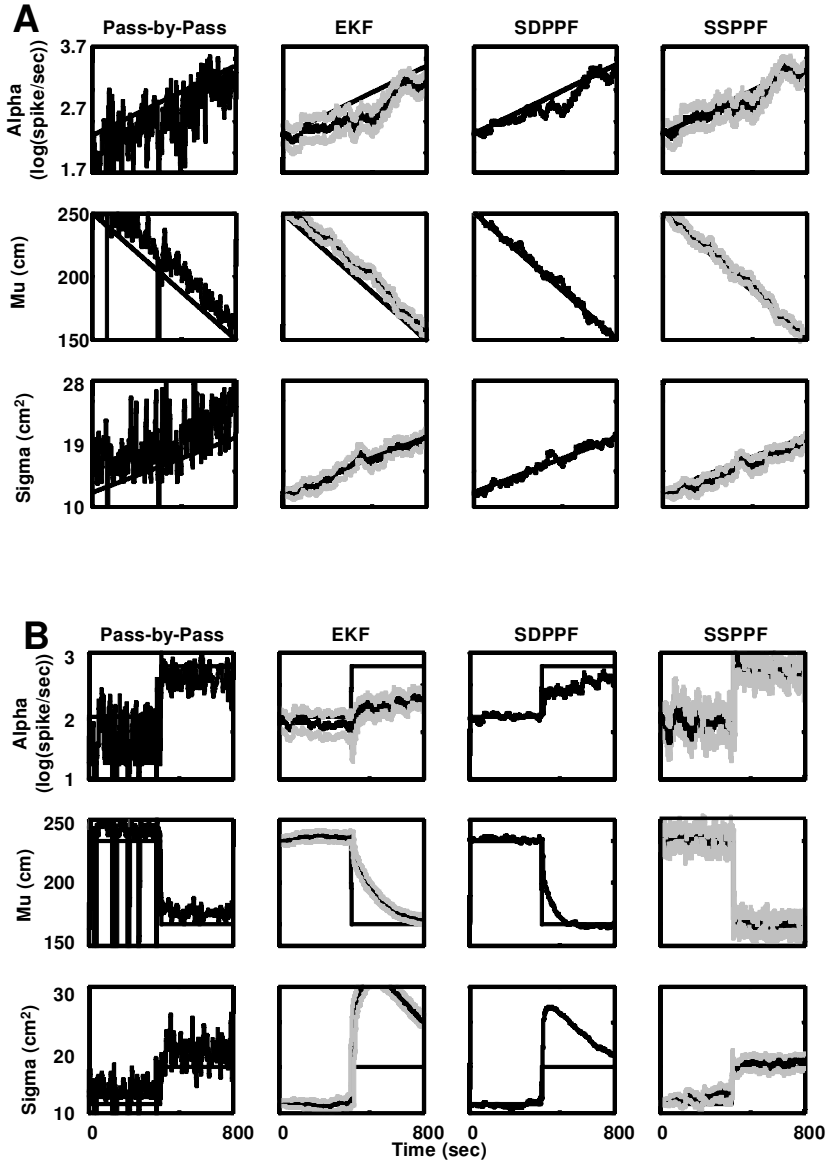
$$\boldsymbol{\theta}_{k|k} = \boldsymbol{\theta}_{k-1|k-1} + \mathbf{W}_{k|k}\left[\left(\frac{\partial \log \lambda}{\partial \boldsymbol{\theta}}\right)' (r(t) - \lambda)\Delta t_k\right]_{\boldsymbol{\theta}_{k-1|k-1}}, \tag{3.5}$$

where $r(t_k)$ was the firing rate estimate (Haykin, 1996). Recent analyses of neural decoding in motor cortex have used similar rate-based extended Kalman filters (Wu et al., 2002).

We compared the ability of these four algorithms to track the evolving receptive fields as estimated from the simulated spike trains for both the linear and rapid jump evolution scenarios, using three different measures: the mean-squared error between the true parameters and their estimated values, the percentage of the time that the true parameter values were covered by the estimated 99% confidence intervals, and the Kolmogorov-Smirnov (KS) statistics.

Figure 2 shows examples of the parameter estimates predicted by each filter for each place field evolution scenario. In the linear evolution scenario,

Figure 2: *Facing page.* Example of tracking place field parameter evolution with each of the adaptive estimation and other statistical analysis methods for the (A) linear or (B) jump parameter evolution scenarios in Figure 1. The estimated parameter trajectories from each procedure are plotted along with the true parameter trajectories in the indicated column: pass-by-pass, column 1; extended Kalman filter (EKF), column 2; steepest descent point process filter (SDPPF), column 3; and stochastic state point process filter (SSPPF), column 4. The pass-by-pass method provides only one estimate per pass, while the other three algorithms provide an estimate at each 20 msec time step. Additionally, the EKF and SSPPF are plotted with their instantaneous 95% confidence interval (gray lines above and below the trajectory estimates). The pass-by-pass method tracks least well the evolution of the place field parameters. Both the EKF and SDPPF track the parameter trajectories well in the linear evolution scenario but underestimate the change in α and μ and overestimate the change in σ in the jump evolution scenario. The SSPPF tracks all three parameters well in both scenarios.


A
Pass-by-Pass
EKF
SDPPF
SSPPF
Alpha (log(spike/sec))
3.7
2.7
1.7
Mu (cm)
250
200
150
Sigma (cm²)
28
19
10
B
Pass-by-Pass
EKF
SDPPF
SSPPF
Alpha (log(spike/sec))
3
2
1
Mu (cm)
250
200
150
Sigma (cm²)
30
20
10
0
800
Time (sec)

each algorithm tracked each parameter with the correct direction of movement and approximately the correct magnitude of change (see Figure 2A). In addition, the EKF and SSPPF provided instantaneous variance estimates, which we used to compute the confidence intervals shown in gray.

The pass-by-pass method gave the most variable parameter trajectory estimates. This was because on several passes, the actual number of spikes observed was small, and thus there was little information on which to base the estimate. Several passes had no spikes, and hence no place field estimate could be computed. For the pass-by-pass method, there was also a lag in tracking the true parameter evolution because it estimated the parameters only after completing each full pass along the track. For the EKF, SDPPF, and SSPPF, we obtained comparable tracking for each of the three parameters, with the exception of the EKF tracking of the place field center, μ, which showed a bias toward overestimation. The confidence intervals were comparable between the EKF and SSPPF algorithms for each parameter.

In the jump evolution scenario, the pass-by-pass method again performed poorly (see Figure 2B). Its estimates were noisy in both the baseline condition and after the jump. The EKF showed the same bias as in the linear evolution scenario, as can be seen in the initial segment of tracking μ. Both the EKF and the SDPPF had problems tracking the place field after the jump. Both algorithms overestimated the change in the place field scale and underestimated both the changes in the place field center and in its peak firing rate. Immediately after the jump in the place field parameters, the first indication that the place field estimate was no longer reasonable came when the observed firing activity increased drastically in an area where the place field estimate predicted a near-zero firing rate. There are multiple ways to update the parameters based on these observations. The rate parameter, α, could increase, causing an increase in the firing everywhere, the place field width, σ, could increase to include the current area of firing, or the center of the place field, μ, could shift. In this example, all three are occurring simultaneously, but the most prominent effect is due to the field center shifting. The EKF and SDPPF were unable to determine rapidly the relative effects of each of these and tended to overweight the learning rate for σ and underweight the rates for α and μ.

The SSPPF tracked the place field dynamics in the rapid evolution scenario well. This was because it was the only algorithm that adapted its learning rate in the proper way to scale the information obtained about each parameter from the spiking activity. The SSPPF scaled up each of its relatively low learning rates right after the jump, and then scaled them down rapidly so as to continue to track correctly the again constant parameter trajectories.

As shown in Table 1, the mean-squared error (MSE) between the true and estimated place field parameter trajectories was highest in both evolution scenarios for the pass-by-pass method. Because this algorithm estimated

Table 1: Summary Analysis of the Adaptive Parameter Estimation of the Place Receptive Field Dynamics for the Linear and Jump Parameter Evolution Scenarios.

Linear Evolution		Pass-by-Pass	EKF	SDPPF	SSPPF
MSE	α	0.5	0.1	0.03	0.01
	μ	2200	200	12	60
	σ	35	2.5	1.1	0.5
Coverage of 99% confidence interval	α	NA	68	NA	98
	μ	NA	7	NA	74
	σ	NA	99	NA	99
KS statistic		0.18	0.09	0.057	0.058
Jump Evolution					
MSE	α	0.4	0.15	0.1	0.04
	μ	3000	400	200	50
	σ	25	10	40	2
Coverage of 99% confidence interval	α	NA	55	NA	99
	μ	NA	15	NA	99
	σ	NA	64	NA	92
KS statistic		0.09	0.22	0.11	0.06

Notes: Each entry is derived from an average of the indicated method being applied to 10 simulated spike trains. MSE (mean squared error); KS (Kolmogorov-Smirnov) statistic. The coverage probability is the fraction of time the 99% confidence intervals for the parameters contained the true parameter value. If the intervals are correct, they should cover the true parameter value 99% of the time. The 95% confidence interval on the KS statistic is 0.044 for the linear evolution scenario and 0.040 for the jump evolution scenario.

only one place field over a full pass, it required multiple passes before a reasonable statement about the place field could be made. Hence, it was difficult to track the place field evolution accurately.

For the linear evolution, the SSPPF gave the lowest MSE for the place field scale parameter and maximum firing rate parameter trajectories, and the SDPPF gave the lowest MSE for the place field center trajectory. For each of these, the lowest MSE was approximately one order of magnitude smaller than the EKF MSE and two orders of magnitude smaller than that of the pass-by-pass algorithm. For the jump evolution scenario, the SSPPF gave the smallest MSE of all the algorithms for each parameter.

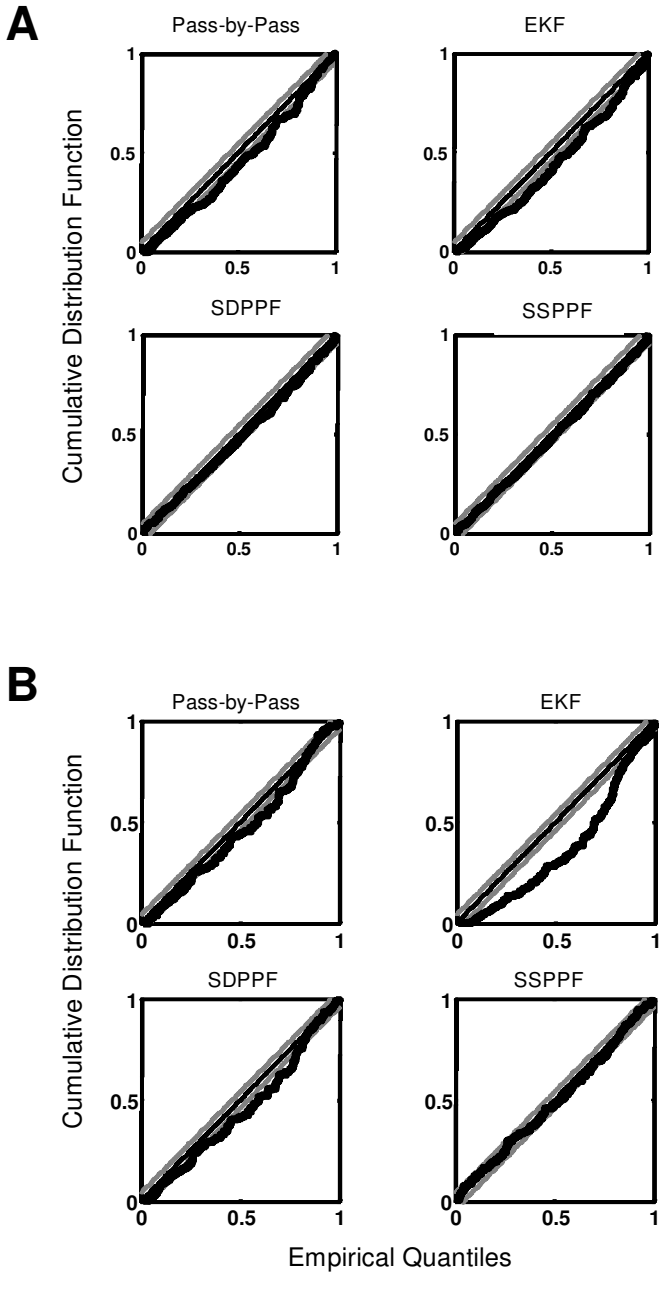
Neither the pass-by-pass algorithm nor the SDPPF provided an instantaneous variance estimate. Therefore, no measure of confidence could be ascribed to their trajectory estimates. For the SSPPF and EKF, the degree to

which their parameter trajectory estimates remained within the 99% confidence intervals measured how well these algorithms characterized their uncertainty. In both scenarios, the SSPPF parameter trajectories stayed within its confidence intervals more than the EKF. This was especially true for the place field center, which was in the 99% confidence interval 74% and 99% of the time for the SSPPF, but only 7% and 15% of the time for the EKF for the linear and jump evolution scenarios, respectively.

To assess goodness-of-fit, the observed ISIs were rescaled using the time-rescaling theorem (Brown et al., 2002) in order to compute the KS statistic for each algorithm. Figure 3 shows examples of KS plots comparing the cumulative distribution of the transformed ISIs with the cumulative distribution function of a uniform on the interval $[0, 1)$ for each trajectory scenario for each of the four estimation strategies. The 95% confidence intervals for the KS statistic are plotted as gray bounds in each panel in Figure 3. The KS plots show that both the pass-by-pass and rate-based estimators tended to predict fewer short ISIs than were observed. The expected ISI for an animal running through its place field at a constant speed decreased as the α parameter increased. Therefore, if the estimation algorithm were slow to track these changes, it would underestimate the probability of shorter ISIs.

For the linear evolution scenario, the SDPPF and SSPPF were equivalent in terms of their KS statistics and significantly better than the pass-by-pass and EKF algorithms. In the rapid jump scenario, the SSPPF performed much better than the SDPPF. In this case, the pass-by-pass estimates actually described the data better than the SDPPF. This is because even though the pass-by-pass estimates were very imprecise, they contained no history dependence and were adjusted to approximate the new place field in a single pass after the jump. Although this estimate was inaccurate, it

Figure 3: *Facing page*. Examples of Kolmogorov-Smirnov (KS) plots assessing goodness-of-fit by comparing the fits of the pass-by-pass, EKF, SDPPF and the SSPPF to the spike trains from the (A) linear and (B) jump evolution scenarios. Under the time-rescaling transformation, the cumulative distribution function is that of the uniform on the interval [0,1). In each panel, the center 45 degree gray line represents perfect agreement between the indicated model fit and the spike train data. The parallel 45 degree black lines on either side are the 95% confidence limits. The wiggly black line is the model estimate of the Kolmogorov-Smirnov plot. For the linear evolution scenario, the KS plots for the pass-by-pass method and the EKF lie outside the 95% confidence intervals, while the SDPPF and SSPPF plots lie within the bounds, suggesting that the latter two point process filters give reasonably good estimates of the observed spiking data for the linear place field evolution scenario. For the jump evolution scenario, none of the KS plots was consistently between the 95% confidence intervals. However, the KS statistic for the SSPPF was much smaller than those from the other three algorithms. This suggests that although no model describes the observed spiking data completely, the SSPPF offers a more accurate description than the other three algorithms.


A
Pass-by-Pass
EKF
SDPPF
SSPPF
Cumulative Distribution Function
0
0.5
1
B
Pass-by-Pass
EKF
SDPPF
SSPPF
Cumulative Distribution Function
Empirical Quantiles

did not lead to the in-between waiting times predicted by the more slowly adapting SDPPF and EKF. Nevertheless, the pass-by-pass method had a larger KS statistic than the SSPPF, which used a more appropriate learning rate.

In order to reevaluate the choice for the constant learning rate matrix in the steepest-descent point process algorithm, the average adaptive gain matrix from the SSPPF over the entire trial, $\bar{\mathbf{W}}_{k|k} = K^{-1}\sum_{k=1}^{K}\mathbf{W}_{k|k}$, was computed, and its diagonal terms were compared to the steepest-descent learning rate. The resulting averaged learning rate parameters were consistently of the same order of magnitude as those obtained by minimizing the mean squared error of the estimates over a simulated training data set. Thus, the SSPPF was able to track the parameter trajectories in the jump evolution scenario better than the SDPPF even though the average magnitudes of their learning rates were the same. We conclude that the SSPPF algorithm's ability to adapt its learning rate estimate was essential to its tracking success. Overall, the SSPPF provided the most accurate tracking of both the linear and jump evolution scenarios.

3.2 A Simulation Study of Simultaneous Adaptive Estimation of Receptive Fields and Neural Spike Train Decoding. In this example, we show how to combine an adaptive encoding algorithm with a neural spike train decoding algorithm in order to reconstruct a signal in the presence of receptive fields that are evolving.

A standard decoding paradigm involves a two-step process (Brown et al., 1998). First, an encoding step, performed on a subset of the spike data, is used to characterize the receptive field properties of each neuron in the ensemble. Then in a separate step, a decoding algorithm uses the receptive field estimates from the encoding step along with the remaining spiking activity to reconstruct the signal. This process assumes that the receptive fields are static during both the encoding and decoding periods.

If the receptive field evolves during the decoding interval, no such separation between encoding and decoding periods can be made. Instead, the decoding algorithm must simultaneously update each neuron's receptive field estimate and use the updated estimates to reconstruct the signal. This creates an interdependence between encoding and decoding in that the accuracy of the receptive field estimate depends on the accuracy of the signal estimate and vice versa.

To study this problem, we examined an ensemble of four simulated neurons responding to a one-dimensional stimulus. We used a notation similar to that used above to describe the receptive field dynamics of hippocampal place cells. We added superscripts to denote each neuron in the ensemble. For illustrative purposes, we regard these simulated cells as motor neurons and the simulated stimulus as a one-dimensional hand velocity. Our receptive field model consists of a background firing rate modulated by the hand

velocity, $\lambda^j(t_k) = \exp(\mu + \beta_k^j v_k)$, where $\exp(\mu)$ represents the background firing rate for each neuron in the absence of any movement and β_k^j represents for neuron j the modulation in the firing rate due to the arm velocity v_k, where $j = 1, \ldots, 4$. This model is conceptually similar to the velocity and direction tuned motor cortical neural model developed by Moran and Schwartz (1999), though simplified for a one-dimensional movement.

We allowed both the hand velocity and the modulation parameters to be time varying. The hand movement velocity was simulated as a random walk with a noise variance of 2.5×10^{-5} at each 1 ms time step. The background rate for each cell was fixed at $\exp(\mu) = 1$. The time-dependent tuning functions for the cells in this simulated ensemble are shown as the straight gray lines in Figure 4. Neurons 1 and 2 had fixed modulation parameters throughout the 800 second simulation, $\beta_k^1 = 3$ and $\beta_k^2 = -3$, respectively. Neurons 3 and 4 had time-varying modulation parameters given by

$$\beta_k^3 = \begin{cases} 0 & 0 \le t_k \le 200 \\ 0.0125(t_k - 200) & 200 \le t_k \le 400 \\ 2.5 & 400 \le t_k \le 800 \end{cases}, \tag{3.6}$$

$$\beta_k^4 = \begin{cases} 0 & 0 \le t_k \le 400 \\ -0.0125(t_k - 400) & 400 \le t_k \le 600 \\ -2.5 & 600 \le t_k \le 800 \end{cases}. \tag{3.7}$$

Thus, initially, only two of the four neurons responded to the hand velocity; however, by the end of the simulation, all four neurons responded (see Figure 4C). The goal of this analysis was to decode the hand velocity while simultaneously tracking the evolution of the receptive field modulation parameters.

To conduct this analysis, we augmented the state-space of the SSPPF to accommodate all the receptive field parameter estimates as well as the hand velocity signal (Ljung & Soderstrom, 1987; Haykin, 1996). That is, the five-dimensional augmented state vector is defined as $\boldsymbol{\theta}_k = [v_k \quad \beta_k^1 \quad \beta_k^2 \quad \beta_k^3 \quad \beta_k^4]'$. The state equation for this system is given by equation 2.2 with $\mathbf{F}_k = \mathrm{Diag}(0.99, 1, 1, 1, 1)$ and $\mathbf{Q}_k = 10^{-5} \cdot \mathrm{Diag}(2.5, 1, 1, 1, 1)$. The derivation of the SSPPF in the appendix is sufficiently general to allow the analysis of this ensemble adaptive decoding problem.

From equations A.7 and A.8, the derivatives of the intensity function with respect to $\boldsymbol{\theta}$ required to define the SSPPF are

$$\left.\frac{\partial \log \lambda}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}_{k|k-1}} = \begin{bmatrix} \beta_{k|k-1}^1 & \beta_{k|k-1}^2 & \beta_{k|k-1}^3 & \beta_{k|k-1}^4 \\ v_{k|k-1} & 0 & 0 & 0 \\ 0 & v_{k|k-1} & 0 & 0 \\ 0 & 0 & v_{k|k-1} & 0 \\ 0 & 0 & 0 & v_{k|k-1} \end{bmatrix}, \tag{3.8}$$

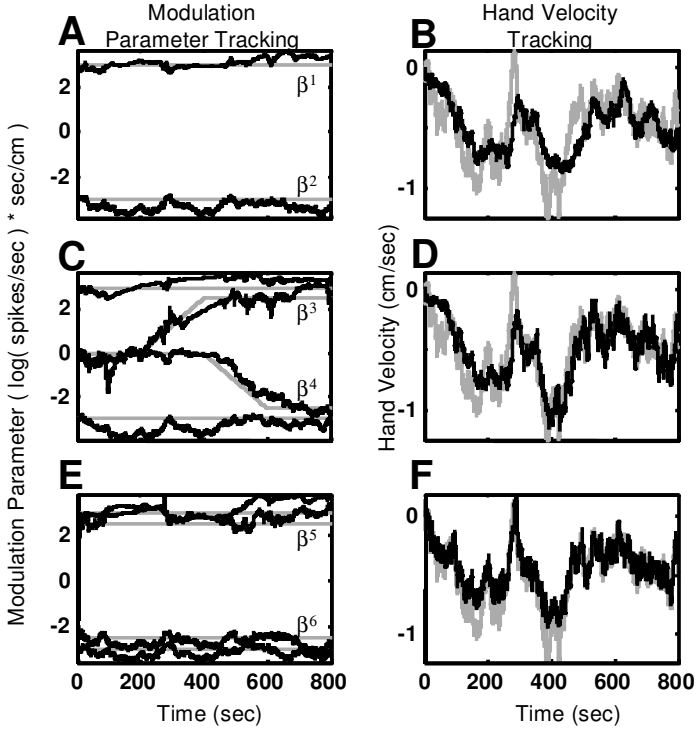


Figure 4: Adaptive-decoding analysis of simulated ensemble neural spike train activity using the SSPPF algorithm. The true modulation parameter trajectory is gray in *A*, *C*, and *E*, and the SSPPF estimate is in black. In *B*, *D*, and *F* the true hand velocity signal is gray and the SSPPF decoding estimate of the hand velocity signal is black. (A) Adaptive estimation of the modulation parameters in the two-neuron ensemble consisting of neurons 1 and 2 with true modulation parameter values of $\beta_k^1 = 3$ and $\beta_k^2 = -3$. (B) True hand velocity trajectory and decoded hand velocity trajectory estimated from the spiking activity of the two neurons in *A*. (C) Adaptive estimation of the modulation parameters in a four-neuron ensemble consisting of neurons 3 and 4 with true modulation parameter values of β_k^3 and β_k^4, defined in equations 3.6 and 3.7, respectively, included with neurons 1 and 2 from *A*. (D) True hand velocity trajectory and the decoded hand velocity trajectory estimated from the spiking activity of the four neurons in *C*. (E) Adaptive estimation of the modulation parameters of a four-neuron ensemble consisting of neurons 5 and 6 with modulation parameters $\beta_k^5 = 2.5$ and $\beta_k^6 = -2.5$, respectively, included with neurons 1 and 2 from *A*. (F) True hand velocity trajectory and the decoded hand velocity trajectory estimated from the spiking activity of the four neurons in *E*.

where $\frac{\partial \log \lambda^j}{\partial \boldsymbol{\theta}}|_{\boldsymbol{\theta}_{k|k-1}}$ is the jth column of equation 3.8, and $\frac{\partial^2 \log \lambda^j}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'}|_{\boldsymbol{\theta}_{k|k-1}}$ is the 5×5 matrix with a 1 in the first row and $(j+1)$th column, a 1 in the first column and $(i+1)$th row, and zeros elsewhere.

Figure 4 shows the SSPPF tracking simultaneously the evolution of the modulation parameter of each of the four neurons (see Figure 4C) while reconstructing the hand velocity (see Figure 4D). Initially, there were two neurons with static nonzero modulation parameters, but after 200 seconds, an additional two neurons began spiking, and their output was used as well in the hand velocity reconstruction. The algorithm tracks well the trajectories of the neurons with static modulation parameters as well as the trajectories of the modulation parameters of the two new neurons.

To analyze the performance of the simultaneous adaptive estimation (see Figure 4C) and the decoding (see Figure 4D) with the SSPPF algorithm, we compared these findings to its performance in adaptive decoding in the case where there are two neurons with static modulation parameters (see Figure 4A) and four neurons with static modulations (see Figure 4E). The two neurons in Figure 4A are neurons 1 and 2 in Figure 4C, whereas the four neurons in Figure 4E are neuron 1 and neuron 2 in addition to neurons 5 and 6, whose modulation parameters are $\beta_k^5 = 2.5$ and $\beta_k^6 = -2.5$, respectively. The hand velocity reconstruction is better for the ensemble with four neurons with static nonzero modulation parameters (see Figure 4F) than for the two neurons (see Figure 4B). The MSE between the true and estimated velocity trajectories is 0.01 for the four neurons and 0.038 for the two neurons. By comparison, for the first 200 seconds, the quality of the decoding and the MSE of 0.041 for the ensemble of adapting neurons in Figure 4C is comparable to decoding with the two neurons with static modulation parameters. However, after 200 seconds as the second two neurons begin to spike and the SSPPF estimates their modulation parameters (see Figure 4E) and uses their output in the decoding, the quality of the decoding improves (see Figure 4F) and the MSE decreases. The MSE for reconstruction for the time between 600 and 800 seconds for the dynamic ensemble is 0.01, similar to the reconstruction error from four neurons with static modulation parameters. This example illustrates the feasibility of reconstructing a neural signal from an ensemble whose receptive fields are evolving.

4 Discussion

We have presented a paradigm for constructing adaptive filters to track the temporal evolution of neural receptive fields from spike train observations. Our formulation involves first specifying a probabilistic model of neural spiking and then combining it with a state evolution model using the Chapman-Kolmogorov equation in a Bayesian framework to obtain a recursive expression for the posterior distribution of a parameter vector to be tracked. We constructed a set of adaptive filtering algorithms using a gaus-

sian approximation to this posterior distribution. These filters extended the SDPPF that we have already applied in actual neural data analyses (Brown et al., 2001; Frank et al., 2002; Wirth et al., 2003).

We developed a stochastic state point process filter by taking a quadratic approximation to the logarithm of the posterior density in equation 2.6. We did this by expanding the log posterior at t_k in a Taylor series about an arbitrary point, $\boldsymbol{\theta}^*$ and retaining only terms of order 2 or smaller. By choosing $\boldsymbol{\theta}^*$ to be the one-step prediction estimate, $\boldsymbol{\theta}_{k|k-1}$, we obtained our SSPPF, which is a point process analogue of the Kalman filter. Alternatively, if we had chosen $\boldsymbol{\theta}^* = \boldsymbol{\theta}_{k|k}$, we would have obtained a nonlinear point process adaptive filter that gives the maximum a posteriori estimate as used in Brown et al. (1998). Our filter construction provided two additional adaptive estimation algorithms as special cases: point process analogues of the RLS and steepest-descent algorithms. Additional algorithms could be similarly constructed based on alternative probability distributions. One option is the smoothing distribution, $p(\boldsymbol{\theta}_k \mid \mathbf{x}_{1:K}, N_{1:K})$, which can be used to estimate the parameter $\boldsymbol{\theta}$ at time k if all the data from time steps 1 to $K \geq k$ have been observed.

Both our SSPPF and EKF use gaussian approximations to the posterior density at each time point. The EKF computes its gaussian approximation by linearizing the state evolution and observation equations to yield a gaussian estimate of the state and observation noise models (Haykin, 1996). In the SSPPF, we used point process observation models to construct an exact posterior density update equation, whose solution is then approximated by a gaussian. The SSPPF maintains the point process description of the neural spiking and obviates using a gaussian approximation to a spike train model, an approximation that may be highly inaccurate in neural systems with low spiking rates.

In order to apply the SSPPF, it is first necessary to specify an observation model that describes the spiking probability distribution in terms of a biological signal, a set of receptive field model parameters, and the nature of the history dependence. The history dependence allows us to construct non-Poisson neural spike train models. In Frank et al. (2002), we showed how a history-dependent process could be used to obtain improved predictions of theta and burst-dependent firing in rodent hippocampus. In Barbieri, Quirk, Frank, Wilson, and Brown (2001), history dependence was incorporated into the model by using a change of variables to construct inhomogeneous gamma and inverse gaussian temporal firing models. These models can be incorporated into our current framework, as well.

Another essential component for these adaptive filters is the state model for the evolution of the place field parameters. In our examples, the exact state model used to simulate the data was not used in the estimation algorithm. In fact, knowledge of the exact state model for neural receptive fields is not possible at present. It is important, however, to select a model that encompasses the anticipated quality and degree of change to be tracked. In

our hippocampal example, we used a random walk model, which is useful in situations where the expected direction of change of each of the parameters is unknown, but the magnitude of change can be estimated. In some cases, previous research can suggest a more informative system model. For the example of the rat hippocampus, we could have applied our knowledge that the place field's center tends to move backward while its height tends to increase, in order to construct a set of $\mathbf{F}_k$ matrices that capture these trends. Additionally, if the form of the model can be specified but the model parameters are unknown, then the parameters can be estimated using an expectation-maximization algorithm (Smith & Brown, 2003; Roweis & Ghahramani, 2000).

The advantages of using point process filters on neural data are highlighted in the first example, in which we compared several traditional and point process adaptive estimation algorithms tracking the evolution of simulated hippocampal place fields. The pass-by-pass method of estimating the receptive field by histogramming the spike train over a full pass computes only a single place field estimate for each pass through the environment. In contrast, our filters, based on a stochastic parameter evolution model, allow us to update our receptive field estimates instantaneously, whether or not a spike was observed. The model-based algorithms impose an implicit continuity constraint on the model place field dynamics, preventing the large fluctuations in the parameter estimates observed with the pass-by-pass method from one pass to the next.

The SSPPF maintains adaptive estimates of both the parameter mean and variance, permitting the construction of confidence intervals about the receptive field estimates. The one-step variance estimate provides an implicit learning rate, which allows the filter to track trajectories that vary over multiple time courses. Finally, as mentioned above, our filters are based on point process observation models of the spike train, so that data enter the algorithm as a point process. This eliminates the need to construct a separate rate estimation algorithm, which could lead to biased estimates and erroneous learning rates, as in the case of the EKF algorithm. The SSPPF was the only one of these estimation algorithms that had all of these advantages and was the only one to track place field parameters accurately in both the linear and jump evolution scenarios.

The second example illustrates a novel application of our adaptive algorithm: the reconstruction of a biological signal from ensemble spiking activity while the receptive fields of the neurons in the ensemble are evolving. To do this, we couple adaptive estimation and decoding. Such adaptive decoding may be necessary for reconstructing signals in neural systems where the receptive fields are dynamic. For example, during a center-out reaching task, the preferred direction for MI neurons can change depending on the load against which the animal is reaching (Li et al., 2001). If the objective were to reconstruct hand trajectory under these conditions, we would need to be able simultaneously to track the receptive field evolution and

to use these estimates to track the intended movement. Another situation where this application becomes important is when the firing properties of a population of neurons are changing. This may occur with long-term electrode implants, where some cells stop responding or are lost to the recording device while the activity of other neurons becomes observable. Both of these problems are likely to arise with the eventual development of neural prostheses (Donoghue, 2002).

In this example, the key to adaptive decoding is the augmented state space model, which allows us to incorporate into one equation the temporal evolution of the hand trajectory and of each neuron's receptive field properties. If the number of neurons in the ensemble is large, it may be computationally more tractable to implement separate adaptive filters in lock step—one estimating the current state variable from previously estimated receptive field model parameters and the other updating the model parameters using the current state estimate. This approach treats the encoding and decoding components as separate problems, in much the same way as do two-step static decoders (Brown et al., 1998).

While our goodness-of-fit tests suggest that the SSPPF algorithm is performing well, a detailed study of the accuracy of the gaussian approximation we use should be undertaken. This can be studied by computing the posterior density (see equation 2.6) and the one-step prediction density (see equation 2.7) using numerical integration or sequential Monte Carlo methods (Kitagawa & Gersh, 1996; Doucet, de Freitas, & Gordon, 2001). The SSPPF can be used to construct the proposal densities in the implementation of the sequential Monte Carlo algorithms. The difference between the Monte Carlo–derived posterior density and the gaussian estimate would provide a straightforward measure of the error in the approximation.

In summary, the point process filtering paradigm we have developed should be useful to track the dynamics of neural receptive fields across a wide variety of conditions.

Appendix: Derivation of the Stochastic State Point Process Filter

For the purpose of this derivation, we examine the general situation of an ensemble of neurons firing simultaneously. For the most part, in the encoding problems we described, information from individual neurons was studied independently. However, in the second example and in most decoding or reconstruction problems, it will be necessary to combine information from an ensemble of spiking neurons.

We use superscript notation to distinguish the properties of individual cells. For example, for an ensemble containing C cells, $\boldsymbol{\theta}_k^i$ represents the parameter vector, $\lambda^i(t_k \mid \mathbf{x}_k, \boldsymbol{\theta}_k^i, \mathbf{N}_{1:k-1})$ represents the conditional intensity function, and ΔN_k^i is the number of spikes fired for the ith cell in the ensemble.

The definitions for the means and variances of the one-step prediction and posterior densities, along with the evolution model, are sufficient to compute the gaussian one-step prediction probability density, whose moments are given by,

$$\boldsymbol{\theta}^i_{k|k-1} = \mathbf{F}^i_k \boldsymbol{\theta}^i_{k-1|k-1}, \tag{A.1}$$

$$\mathbf{W}^i_{k|k-1} = \mathbf{F}^i_k \mathbf{W}^i_{k-1|k-1} \mathbf{F}^{i'}_k + \mathbf{Q}^i_k. \tag{A.2}$$

We call these two equations the one-step prediction equation and one-step variance equation, respectively. We write the posterior probability density of the neural activity of the ensemble in the interval $(t_{k-1}, t_k]$, by multiplying over all neurons the independent probabilities in equation 2.6 and we make a gaussian approximation:

$$\begin{aligned} p(\boldsymbol{\theta}_k \mid \Delta N_k, \mathbf{H}_k) &\propto \prod_{j=1}^{C} (\lambda^j(t_k \mid \boldsymbol{\theta}^j_k, \mathbf{H}_k)\Delta t_k)^{\Delta N^j_k} \exp(-\lambda^j(t_k \mid \boldsymbol{\theta}^j_k, \mathbf{H}_k)\Delta t_k \\ &\quad - \frac{1}{2}(\boldsymbol{\theta}^j_k - \boldsymbol{\theta}^j_{k|k-1})'(\mathbf{W}^j_{k|k-1})^{-1}(\boldsymbol{\theta}^j_k - \boldsymbol{\theta}^j_{k|k-1})) \\ &\propto \prod_{i=1}^{C} \exp\left(-\frac{1}{2}(\boldsymbol{\theta}^i_k - \boldsymbol{\theta}^i_{k|k})'(\mathbf{W}^i_{k|k})^{-1}(\boldsymbol{\theta}^i_k - \boldsymbol{\theta}^i_{k|k})\right). \end{aligned} \tag{A.3}$$

The first line in equation A.3 represents the posterior density as a function of the variable $\boldsymbol{\theta}^i_k$, and the last line is a gaussian approximation to this function with the parameters $\boldsymbol{\theta}^i_{k|k}$ and $\mathbf{W}^i_{k|k}$ still to be determined. We could expand the first line of equation A.3 in a Taylor series about some point, drop all terms of third order or higher, and complete a square to obtain an expression of the form of the second line of equation A.3. Here instead we present an alternate derivation.

Taking the log of both sides gives

$$\begin{aligned} -\frac{1}{2}\sum_{i=1}^{C}(\boldsymbol{\theta}^i_k - \boldsymbol{\theta}^i_{k|k})'(\mathbf{W}^i_{k|k})^{-1}(\boldsymbol{\theta}^i_k - \boldsymbol{\theta}^i_{k|k}) &= \sum_{j=1}^{C} \Delta \mathbf{N}^j_k \log(\lambda^j) - \lambda^j \Delta t_k \\ - \frac{1}{2}(\boldsymbol{\theta}^j_k - \boldsymbol{\theta}^j_{k|k-1})'(\mathbf{W}^j_{k|k-1})^{-1}(\boldsymbol{\theta}^j_k - \boldsymbol{\theta}^j_{k|k-1}) + R. \end{aligned} \tag{A.4}$$

Here, R is a catch-all constant that contains terms relating to the state evolution statistics and the normalizing constants. We can now differentiate this expression with respect to $\boldsymbol{\theta}^i_k$ in order to get an equation with only linear terms,

$$\begin{aligned} (\mathbf{W}^i_{k|k})^{-1}(\boldsymbol{\theta}^i_k - \boldsymbol{\theta}^i_{k|k}) &= (\mathbf{W}^i_{k|k-1})^{-1}(\boldsymbol{\theta}^i_k - \boldsymbol{\theta}^i_{k|k-1}) \\ &\quad - \sum_{j=1}^{C}\left(\frac{\partial \log \lambda^j}{\partial \boldsymbol{\theta}^i_k}\right)'(\Delta \mathbf{N}^j_k - \lambda^j \Delta t_k). \end{aligned} \tag{A.5}$$

If the gaussian approximation is valid, then this relation should be approximately true for all values of $\boldsymbol{\theta}_k^i$. We can therefore choose any specific point to evaluate this expression. Evaluating at $\boldsymbol{\theta}_k^i = \boldsymbol{\theta}_{k|k-1}^i$ gives

$$-(\mathbf{W}_{k|k}^i)^{-1}(\boldsymbol{\theta}_{k|k-1}^i - \boldsymbol{\theta}_{k|k}^i) = \sum_{j=1}^{C}\left[\left(\frac{\partial \log \lambda^j}{\partial \boldsymbol{\theta}_k^i}\right)'(\Delta \mathbf{N}_k^j - \lambda^j \Delta t_k)\right]_{\boldsymbol{\theta}_{k|k-1}^i}. \quad \text{(A.6)}$$

Solving for $\boldsymbol{\theta}_{k|k}^i$ then gives the posterior state equation,

$$\boldsymbol{\theta}_{k|k}^i = \boldsymbol{\theta}_{k|k-1}^i + \mathbf{W}_{k|k}^i \sum_{j=1}^{C}\left[\left(\frac{\partial \log \lambda^j}{\partial \boldsymbol{\theta}_k^i}\right)'(\Delta \mathbf{N}_k^j - \lambda^j \Delta t_k)\right]_{\boldsymbol{\theta}_{k|k-1}^i}. \quad \text{(A.7)}$$

Differentiating equation A.6 again and evaluating at $\boldsymbol{\theta}_k^i = \boldsymbol{\theta}_{k|k-1}^i$ gives the posterior variance equation,

$$(\mathbf{W}_{k|k}^i)^{-1} = (\mathbf{W}_{k|k-1}^i)^{-1} + \sum_{j=1}^{C}\left[\left(\frac{\partial \log \lambda^j}{\partial \boldsymbol{\theta}_k^i}\right)'[\lambda^j \Delta t_k]\left(\frac{\partial \log \lambda^j}{\partial \boldsymbol{\theta}_k^i}\right) - (\Delta N_k^j - \lambda^j \Delta t_k)\frac{\partial^2 \log \lambda^j}{\partial \boldsymbol{\theta}_k^i \partial \boldsymbol{\theta}_k^{i\prime}}\right]_{\boldsymbol{\theta}_{k|k-1}^i}. \quad \text{(A.8)}$$

In the case where we observe the firing of only one cell at a time, the sums in equations A.7 and A.8 simplify to a single term, and the superscripts are no longer needed.

Acknowledgments

This research was supported in part by NIH grants MH59733, MH61637, and DA015644, and NSF grant IBN 0081548 to E.N.B. and NIH grant MH65108 to L.M.F. Part of this research was performed while E.N.B. was on sabbatical at the Laboratory for Information and Decision Systems in the Department of Electrical Engineering and Computer Science at MIT.

References

Barbieri, R., Quirk, M. C., Frank, L. M., Wilson, M. A., & Brown, E. N. (2001). Construction and analysis of non-Poisson stimulus response models of neural spike train activity. *J. Neurosci. Meth., 105*, 25–37.

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

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., 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. *J. Neurosci., 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. *PNAS, 98,* 12261–12266.

Daley, D., & Vere-Jones, D. (1988). *An introduction to the theory of point process.* New York: Springer-Verlag.

Donoghue, J. P. (1995). Plasticity of sensorimotor representations. *Curr. Opin. Neurobiol., 5,* 749–754.

Donoghue, J. P. (2002). Connecting cortex to machines: Recent advances in brain interfaces. *Nat. Neurosci., 5,* 1085–1088.

Doucet, A., de Freitas, N., & Gordon, N. (2001). *Sequential Monte Carlo methods in practice.* New York: Springer-Verlag.

Frank, L. M., Brown, E. N., & Wilson, M. A. (2000). Trajectory encoding in the hippocampus and entorhinal cortex. *Neuron, 27,* 169–178.

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 entorhinal cortex: An adaptive filtering approach. *J. Neurosci., 22,* 3817–3830.

Frank, L. M., Stanley, G. B., & Brown, E. N. (2003). *New places and new representations: Place field plasticity in the hippocampus and cortex* (Program No. 91.18). Washington, DC: Society for Neuroscience.

Gandalfo, F., Li, C. S. R., Benda, B. J., Schioppa, C. P., & Bizzi, E. (2000). Cortical correlates of learning in monkeys adapting to a new dynamical environment. *Proc. Natl. Acad. Sci. USA, 97,* 2259–2263.

Haykin, S. (1996). *Adaptive filter theory.* Englewood Cliffs, NJ: Prentice Hall.

Jog, M. S., Kubota, Y., Connolly, C. I., Hillegaart, V., & Graybiel, A. M. (1999). Building neural representations of habits. *Science, 286,* 1745–1749.

Kaas, J. H., Merzenich, M. M., & Killackey, H. P. (1983). The reorganization of somatosensory cortex following peripheral nerve damage in adult and developing mammals. *Annu. Rev. Neurosci., 6,* 325–356.

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

Li, C. S. R., 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.

Ljung, L., & Soderstrom, T. (1987). *Theory and practice of recursive identification.* Cambridge, MA: MIT Press.

Mehta, M. R., Barnes, C. A., & McNaughton, B. L. (1997). Experience-dependent, asymmetric expansion of hippocampal place fields. *Proc. Natl. Acad. Sci. USA, 94,* 8918–8921.

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., Nelson, R. J., Stryker, M. P., Cynader, M. S., Schoppmann, A., & Zook, J. M. (1984). Somatosensory cortical map changes following digit amputation in adult monkeys. *J. Comp. Neurol., 224,* 591–605.

Moran, D. W., & Schwartz, A. B. (1999). Motor cortical representation of speed and direction during reaching. *J. Neurophysiol., 82,* 2676–2692.

O'Keefe, J., & Dostrovsky, J. (1971). The hippocampus as a spatial map: Preliminary evidence from unit activity in the freely-moving rat. *Brain Res., 34,* 171–175.

Pettet, N. M. (1992). Dynamic changes in receptive-field size in cat primary visual cortex. *Proc. Natl. Acad. Sci. USA, 89,* 8366–8370.

Roweis, S., & Ghahramani, Z. (2000). An EM algorithm for identification of nonlinear dynamical systems. Available on-line: http://www.gatsby.ucl.ac.uk/publications/papers/index.html.

Smith, A. C., & Brown, E. N. (2003). Estimating a state-space model from point process observations. *Neural Computation, 15,* 965–991.

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

Solo, V., & Kong, X. (1995). *Adaptive signal processing algorithms: Stability and performance*. Upper Saddle River, NJ: Prentice Hall.

Weinberger, N. M. (1993). Learning-induced changes of auditory receptive fields. *Curr. Opin. Neurobiol., 3,* 570–577.

Wirth, S., Yanike, M., Frank, L. M., Smith, A. C., Brown, E. N., & Suzuki W. A. (2003). Single neurons in the monkey hippocampus and learning of new associations. *Science, 300*(5625), 1578–1581.

Wu, W., Black, M. J., Gao, Y., Bienenstock, E., Serruya, M., Shaikhouni, A., & Donoghue, J. P. (2002). *Neural decoding of cursor motion using a Kalman filter*. Paper presented at the *Neural Information Processing Systems Conference*, Vancouver, BC.

Received April 7, 2003; accepted October 6, 2003.