A Monte Carlo Sequential Estimation for Point Process Optimum Filtering

Yiwen Wang, *Student Member*, António R. C. Paiva, *Student Member*, and José C. Príncipe, *Fellow*

***Abstract*— Adaptive filtering is normally utilized to estimate system states or outputs from continuous valued observations, and it is of limited use when the observations are discrete events. Recently a Bayesian approach to reconstruct the state from the discrete point observations has been proposed. However, it assumes the posterior density of the state given the observations is Gaussian distributed, which is in general restrictive. We propose a Monte Carlo sequential estimation methodology to estimate directly the posterior density. Sample observations are generated at each time to recursively evaluate the posterior density more accurately. The state estimation is obtained easily by collapse, i.e. by smoothing the posterior density with Gaussian kernels to estimate its mean. The algorithm is tested in a simulated neural spike train decoding experiment and reconstructs better the velocity when compared with point process adaptive filtering algorithm with the Gaussian assumption.**

I. Introduction

In sequential state estimation, the system state changes over time with a sequence of noisy measurements of continuous observations made on the system. The state vector that contains all the relevant information is used to describe the system through a time-series modeling. There are two models required to analyze and infer the state of a dynamical system: the system model, which describes the evolution of the state with time, and the continuous observation measurement model, which relates the noisy measurements to the state. The probabilistic state space formulation and the updating of information are rooted on the Bayesian approach of incorporating information from measurements. A recursive algorithm is used to construct the posterior probability density function of the state for each time based on all available information, which embodies all available statistical information and in principle the solution to the estimation problem. There are two stages to adapt the filter: prediction and updating. The first stage uses the system model to predict the posterior probability density of the state given the observation from one measurement to the next; the second stage revises the predicted posterior probability density based on the latest measurement of the observation. The Kalman filter is such an analytical solution that embodies the above conceptual filtering under the assumption that the time-series was generated by a linear system and the posterior density of the state given the observation at every step is Gaussian, hence only parameterized by mean and covariance.

Sequential state estimation normally is applied to continuous value observations, and cannot be directly applied to discrete point processes. A relevant example of a discrete point process is neural activity in motor cortex, with the goal of decoding, i.e. reconstructing the animal movement from the spiking activity. The observation becomes the discrete spike train, where only the time instance of the spike event matters but not the amplitude of the spike. A point process adaptive filtering algorithm was recently proposed by Brown et al. [1]. In their approach, the discrete observation of the neural firing spike was utilized to probabilistically reconstruct the position of a freely running rat in space, taken as the state. This approach also embodies the conceptual Bayesian filtering algorithm: predicting the posterior density by a linear state update equation and revising it with the next observation measurement. However, the method assumes that the posterior density of the state vector given the discrete observation is always Gaussian distributed, which may not be the case. We develop a probabilistic filtering algorithm to reconstruct the state from the discrete observation (spiking event) by generating a sequential set of samples to estimate the distribution of the state posterior density without the Gaussian assumption. The posterior density is recursively propagated and revised by the coming spike observation over time. The state at each time is determined by the maximum likelihood estimation or the expectation of the posterior density inferred by a collapsing of the mixture of Gaussian kernels when estimating the posterior density. The algorithm will be described in the next section, followed in Section III by the illustration of the algorithm performance in a simulated neuron decoding example and the comparison to the probabilistic velocity reconstruction with Gaussian assumption on posterior density.

II. Adaptive algorithms for Point Processes

In this section, we review the design of adaptive filters for point processes under the Gaussian assumption, and then introduce our method, a Monte Carlo sequential estimation, to probabilistically reconstruct the state from discrete (spiking) observation events.

A. Adaptive Filtering for Point Processes with Gaussian Assumption

One can model a point process using a Bayesian approach to estimate the system state by evaluating the posterior density of the state given the discrete observation [2]. This framework provides a nonlinear time-series probabilistic model between the state and the spiking event [3].

Given an observation interval $(0, T]$, the number $N(t)$ of events (e.g. spikes) can be modeled as a stochastic inho-

The authors are with the Computational NeuroEngineering Laboratory, Electrical and Computer Engineering Department, University of Florida, Gainesville, FL 32611 USA (email: {wangyw,arpaiva,principe}@cnel.ufl.edu).

mogeneous Poisson process characterized by its conditional intensity function $\lambda(t|\mathbf{x}(t), \theta(t), \mathbf{H}(t))$, i.e. the instantaneous rate of events, defined as

$$\lambda(t|\mathbf{x}(t), \theta(t), \mathbf{H}(t)) \\ = \lim_{\Delta t \to 0} \frac{\Pr(N(t+\Delta t) - N(t) = 1|\mathbf{x}(t), \theta(t), \mathbf{H}(t))}{\Delta t} \quad (1)$$

where $\mathbf{x}(t)$ is the system state, $\theta(t)$ is the parameter of the adaptive filter, and $\mathbf{H}(t)$ is the history of all the states, parameters and the discrete observations up to time t. The relationship between the single parameter Poisson process λ, the state $\mathbf{x}(t)$, and the parameter $\theta(t)$ is a nonlinear model represented by

$$\lambda(t|\mathbf{x}(t), \theta(t)) = f(\mathbf{x}(t), \theta(t)) \quad (2)$$

Using the nonlinear function $f(\cdot)$, assumed to be known or specified according to the application. Let us consider hereafter the parameter $\theta(t)$ as part of the state vector $\mathbf{x}(t)$. Given a binary observation event ΔN_k over the time interval $(t_{k-1}, t_k]$, the posterior density of the whole state vector $\mathbf{x}(t)$ at time t_k can be represented by Bayes' rule as

$$p(\mathbf{x}_k|\Delta N_k, \mathbf{H}_k) = \frac{p(\Delta N_k|\mathbf{x}_k, \mathbf{H}_k) p(\mathbf{x}_k|\mathbf{H}_k)}{p(\Delta N_k|\mathbf{H}_k)} \quad (3)$$

where $p(\Delta N_k|\mathbf{x}_k, \mathbf{H}_k)$ is the probability of observing spikes in the interval $(t_{k-1}, t_k]$, considering the Poisson process

$$\Pr(\Delta N_k|\mathbf{x}_k, \mathbf{H}_k) \\ = (\lambda(t_k|\mathbf{x}_k, \mathbf{H}_k)\Delta t)^{\Delta N_k} \exp(-\lambda(t_k|\mathbf{x}_k, \mathbf{H}_k)\Delta t) \quad (4)$$

and $p(\mathbf{x}_k|H_k)$ is the one-step prediction density given by the Chapman-Kolmogorov equation as

$$p(\mathbf{x}_k|\mathbf{H}_k) \\ = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{H}_k) p(\mathbf{x}_{k-1}|\Delta N_{k-1}, \mathbf{H}_{k-1}) d\mathbf{x}_{k-1} \quad (5)$$

where the state $\mathbf{x}_k$ evolves according to the linear relation

$$\mathbf{x}_k = F_k \mathbf{x}_{k-1} + \eta_k. \quad (6)$$

F_k establishes the dependence on the previous state and η_k is zero-mean white noise with covariance Q_k. Substituting equations (4) and (5) in (3) the posterior density of the state $p(\mathbf{x}_k|\Delta N_k, \mathbf{H}_k)$ can be recursively estimated from the previous one based on all the spike observations. Assuming the posterior density given by (3) and the noise term η_k in the state evolution equation (6) are Gaussian distributed, the Chapman-Kolmogorov equation (5) becomes a convolution of two Gaussian curves, from which the estimation of the state at each time has a close form expression below (see Brown et al. [1] for the derivation details).

$$\mathbf{x}_{k|k-1} = F_k \mathbf{x}_{k-1|k-1} \quad (7a)$$

$$W_{k|k-1} = F_k W_{k-1|k-1} F_k' + Q_k \quad (7b)$$

$$(W_{k|k})^{-1} = (W_{k|k-1})^{-1} + \Bigg[\left(\frac{\partial \log \lambda}{\partial \mathbf{x}_k}\right)' (\lambda \Delta t_k) \left(\frac{\partial \log \lambda}{\partial \mathbf{x}_k}\right) \\ - (\Delta N_k - \lambda \Delta t_k) \frac{\partial^2 \log \lambda}{\partial \mathbf{x}_k \partial x_k'} \Bigg]_{\mathbf{x}_{k|k-1}} \quad (7c)$$

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} \\ + W_{k|k} \left[\left(\frac{\partial \log \lambda}{\partial x_k}\right)' (\Delta N_k - \lambda \Delta t_k) \right]_{\mathbf{x}_{k|k-1}} \quad (7d)$$

The may reason the Gaussian assumption was used in the first place is because it allows to solve *analytically* (5) and therefore, for a closed form solution of (3) as (7).

Although the above set of equations may seem daunting they can be interpreted quite easily. First, (7a) establishes a prediction for the state based on the previous. Then, (7b) and (7c) are used in (7d) to correct or refine the previous estimate, after which the recurrent process is repeated.

B. Monte Carlo Sequential Estimation for Point Processes

The Gaussian assumption applied to the posterior distribution in the algorithm just described may not be true in general. Therefore, for the discrete observations case, a non-parametric approach is developed here which poses no constraints on the form of the posterior density.

Suppose at time instant k the previous system state is $\mathbf{x}_{k-1}$. Recall that because the parameter θ was embedded in the state, all we need is the estimation of the state from the conditional intensity function (1), since the nonlinear relation $f(\cdot)$ is assumed known. Random state samples are generated using Monte Carlo simulations [4] in the neighborhood of the previous state according to (6). Then, weighted Parzen windowing [5] was used with a Gaussian kernel to estimate the posterior density. Due to the linearity of the integral in the Chapman-Kolmogorov equation and the weighted sum of Gaussians centered at the samples we are still able to evaluate directly from samples the integral. The process is recursively repeated for each time instant propagating the estimate of the posterior density, and the state itself, based on the discrete events over time. Notice that due to the recursive approach the algorithm not only depend on the previous observation, but also depend on the whole path of the spike observation events.

Let $\{\mathbf{x}_{0:k}^i, w_k^i\}_{i=1}^{N_S}$ denote a Random Measure [6] in the posterior density $p(\mathbf{x}_{0:k}|N_{1:k})$, where $\mathbf{x}_{0:k}^i, i = 1, \cdots, N_S$ is the set of all state samples up to time k with associated normalized weights $\{w_k^i, i = 1, \cdots, N_S\}$, and N_S is the number of samples generated at each time index. Then, the posterior density at time k can be approximated by a weighted convolution of the samples with a Gaussian kernel as

$$p(\mathbf{x}_{0:k}|N_{1:k}) \approx \sum_{i=1}^{N_S} w_k^i \cdot k(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^i, \sigma) \quad (8)$$

where $N_{1:k}$ is the spike observation events up to time k modeled by an inhomogeneous Poisson Process in Section II-A, and $k(\mathbf{x}-\overline{\mathbf{x}},\sigma)$ is the Gaussian kernel in term of $\mathbf{x}$ with mean $\overline{\mathbf{x}}$ and covariance σ. By generating samples from a proposed density $q(\mathbf{x}_{0:k}|N_{1:k})$ according to the principle of Importance Sampling [7] [8], which usually assumes dependence on $\mathbf{x}_{k-1}$ and N_k only, the weights can be derived by Bayes' rule and Markov Chain property (see Arulampalam et al. [6] for details on a similar derivation, for the continuous case).

$$\begin{aligned} w_k^i &\propto \frac{p(\mathbf{x}_{0:k}^i|N_{1:k})}{q(\mathbf{x}_{0:k}^i|N_{1:k})} && \text{(9a)} \\ &\propto \frac{p(\Delta N_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)p(\mathbf{x}_{0:k-1}^i|N_{1:k})}{q(\mathbf{x}_k^i|\mathbf{x}_{0:k-1}^i,N_{1:k})q(\mathbf{x}_{0:k-1}^i|N_{1:k-1})} && \text{(9b)} \\ &= w_{k-1}^i\frac{p(\Delta N_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i,\Delta N_k)} && \text{(9c)} \end{aligned}$$

Usually the importance density $q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i,\Delta N_k)$ is chosen to be the prior density $p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)$, requiring the generation of new samples from $p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)$ by (6) as a prediction stage.

After the algorithm is applied for a few iterations, a phenomenon called degeneracy may rise, where all but one sample has negligible weight [8], which implies that a large computational effort is taken to update the samples that almost have no contribution to estimate the posterior density. When a significant degeneracy appears, resampling is applied to eliminate the samples with small weight and to concentrate on samples with large weights. In our Monte Carlo sequential estimation of the point process, Sequential Importance Resampling [9] is applied at every time index, so that the samples are i.i.d. and uniform distributed with weights $w_{k-1}^i=\frac{1}{N_S}$. The weights then change proportionally given by

$$w_k^i \propto p(\Delta N_k|\mathbf{x}_k^i) \tag{10}$$

where $p(\Delta N_k|\mathbf{x}_k^i)$ is defined by equation (4) in Section II. Using (6), (10) and the resampling step, the posterior density of the state $\mathbf{x}_k$ given the whole path of the observed events up to time t_k can be approximated as

$$p(\mathbf{x}_k|N_{1:k}) \approx \sum_{i=1}^{N_S} p(\Delta N_k|\mathbf{x}_k^i)\cdot k(\mathbf{x}_k-\mathbf{x}_k^i) \tag{11}$$

Equation (11) shows that the posterior density of the current state given the observation is modified by the latest probabilistic measurement of the observing spike event $p(\Delta N_k|\mathbf{x}_k^i)$, which is the updating stage in adaptive filtering.

Without a close form of the state estimation, we measure the posterior density of the state given the observed spike event $p(\mathbf{x}_k|N_{1:k})$ every time and apply two methods to get the state estimation $\tilde{\mathbf{x}}_k$. One is Maximum Likelihood Estimation (MLE), which picks out the sample $\mathbf{x}_k^{i*}$ with maximum posterior density. The other is to use the expectation of the posterior density as the state estimation. As we smooth the posterior density by convolving with a Gaussian kernel, we can easily obtain the expectation $\tilde{\mathbf{x}}_k$ and its error covariance V_k by collapse [10]:

$$\tilde{\mathbf{x}}_k = \sum_{i=1}^{N_S} p(\Delta N_k|\mathbf{x}_k^i)\cdot \mathbf{x}_k^i \tag{12}$$

$$V_k = \sum_{i=1}^{N_S} p(\Delta N_k|\mathbf{x}_k^i)\cdot(\sigma+(\mathbf{x}_k^i-\tilde{\mathbf{x}}_k)(\mathbf{x}_k^i-\tilde{\mathbf{x}}_k)^T) \tag{13}$$

From (12) and (13), we can see that without complex computation we can easily estimate the next state. Hence, the expectation by collapse is simple and elegant.

III. Simulation of Monte Carlo Sequential Estimation on Neural Spike Train Decoding

Neurons dynamically change their responses to specific input stimuli patterns through learning which has been modeled with the help of receptive fields. Neural decoding can be used to analyze receptive field plasticity and understand how the neurons learn and adapt by modeling the tuning function of neuronal responses. In the rat hippocampus, for example, information about the spatial movement can be extracted from neural decoding, i.e. from the activity of simultaneously recorded noisy place cells [11] [12] representing the spike-observed events.

In a conceptually simplified motor cortical neural model [13], the one-dimensional velocity can also be reconstructed from the neuron spiking events by Monte Carlo sequential estimation algorithm as described in Section II. This algorithm can provide a probabilistic approach to infer the most probable velocity as one of the components of the state. This decoding simulation updates the state estimation simultaneously and applies this estimation to reconstruct the signal, which assumes the interdependence between the encoding and decoding so that the accuracy of the receptive field estimation and the accuracy of the signal reconstruction are reliable on each other. Notice that dealing with a point process measure is a more restrictive problem than when spike counts are used. Actually, the latter situation can be considered directly under the general framework provided by particle filtering [14].

Here, one neuron is simulated as a motor cell and the stimulus as one-dimensional velocity. The tuning function of the receptive field model between the velocity and the firing rate is

$$\lambda(t_k)=\exp(\mu+\beta_k v_k) \tag{14}$$

where $\exp(\mu)$ is the background firing rate without any movement and β_k is the modulation in firing rate due to the velocity v_k. In practice, the function is unknown. Therefore, an educated guess needs to be made; typically based on evidence collected experimentally.

The velocity was generated as a random walk with a noise variance 2.5×10^{-5} at each 1ms time step. The background-firing rate $\exp(\mu)$ and the modulation parameter β_k are set to be 1 and 3 respectively for the whole simulation time, 200s. A neuron spike is drawn as a Bernoulli random variable with probability $\lambda(t_k)\Delta t$ within each 1ms time window [15].

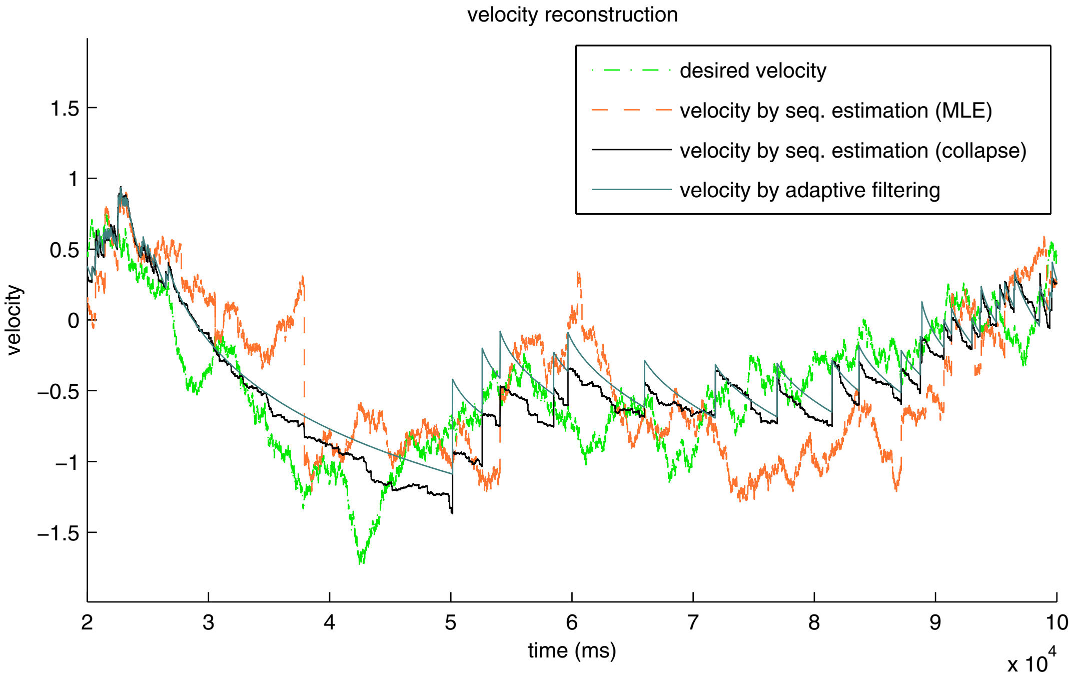

Fig. 1. Velocity reconstruction by the different algorithms.

TABLE I

Comparison Results of All Algorithms with Different Q_k.

diag(Q_k)	Reconstruction error (MSE)		
	Adaptive filtering of Point Process	Sequential Estimation	
		MLE	Collapse
$[2 \times 10^{-5} \;\; 1 \times 10^{-7}]$	**0.04801**	**0.1199**	**0.04522**
$[2 \times 10^{-5} \;\; 1 \times 10^{-6}]$	0.1081	0.1082	0.0489
$[1 \times 10^{-5} \;\; 1 \times 10^{-6}]$	0.1076	0.1013	0.0588

To implement the Monte Carlo Sequential estimation of the point process, we regard both modulation parameter β_k and velocity v_k as the state $\mathbf{x}_k = [v_k \quad \beta_k]^T$. The new samples are generated according to the linear state evolution (6), where $F_k = \mathrm{diag}[1 \quad 1]$ and Q_k were chosen according to the indications given before. The kernel size used in (8) to estimate MLE of the posterior density was the average interval space of the dynamic range of the samples generated at each time index. The MSE between the desired trajectory and the model output is shown in Table I, for different runs of the covariance matrices of the state generation, Q_k. Results for both algorithms described in Section II are compared. In general, if Q_k is too large, the continuity constraint of the whole sequential sample generate has little effect. If it is too small, this constraint may become too restrictive and the reconstructed velocity may get stuck in the same position while the real one moves away by a distance much larger than Q_k. The 100 samples of velocity v_0^i and modulation parameter β_k were initially drawn from uniform and Gaussian distributions, respectively .

With a Monte Carlo running, the state estimation for each time $\tilde{v}_k\tilde{\beta}_k$ is drawn by Maximum Likelihood Estimation and the expectation by collapse. The best velocity reconstruction, shown in the first row of Table I, by both methods is shown in Fig. 1 when $Q_k = \mathrm{diag}[2.5 \times 10^{-5} \quad 1 \times 10^{-7}]$. We have limited the time shown for display clarity.

From Fig. 1 and Table I, we can see that, compared with the desired velocity (dash-dotted line), the best velocity reconstruction was achieved by the sequential estimation with the collapse algorithm (solid black line), is more sensitive than the one by adaptive filtering involving the Gaussian assumption (solid gray line). This is because the method estimates the real posterior density more accurately since no assumptions are made. The reconstruction by sequential estimation with MLE (dashed line) is always very noisy, which is because the sample that the algorithm picks was generated with noisy information. The Monte Carlo sequential estimation on point processes shows a good capability to estimate the state from the discrete spiking events of observation.

IV. Conclusions

Point process adaptive filtering is a two- step Bayesian approach based on the Chapman-Kolmogorov equation to estimate parameters from discrete observed events. However, with the Gaussian assumption of posterior density of the state given the observation, it may not give the most accurate state reconstruction because of the less accurate evaluation of posterior density. We present in this paper a Monte Carlo sequential estimation to modify the amplitude of the observed discrete events by the probabilistic measurement - posterior density. A sequence of samples is generated to estimate the posterior density more accurately. Through sequential estimation and weighted Parzen windowing, we avoid the numerical computation of the integral in the C-K equation. Due to the smoothing of the posterior density with the Gaussian kernel from Parzen windowing, we use collapse to easily get the expectation of the posterior density, which leads to a better result of state estimate than noisy Maximum Likelihood Estimation. The Monte Carlo estimation shows better capability to probabilistically estimate the state because the better approximation of the posterior density than the point process adaptive filtering algorithm with Gaussian assumption.

Acknowledgments

This work was partially supported by NSF ECS 0422718. The work of A. R. C. Paiva was supported by Fundação para a Ciência e a Tecnologia under grant SFRH/BD/18217/2004.

References

[1] E. N. Brown, D. P. Nguyen, L. M. Frank, M. A. Wilson, and V. Solo, "An anlysis of neural receptive field plasticity by point process adaptive filtering," *PNAS*, vol. 98, pp. 12 261–12 266, 2001.

[2] U. R. Eden, L. M. Frank, R. Barbieri, V. Solo, and E. N. Brown, "Dynamic analysis of neural encoding by point process adaptive filtering," *Neural Computation*, vol. 16, no. 5, pp. 971–998, May 2004.

[3] E. N. Brown, L. Frank, and M. Wilson, "Statistical approaches to place field estimation and neuronal population decoding," *Soc. of Neurosci. Abstr.*, vol. 26, p. 910, 1996.

[4] J. Carpenter, P. Clifford, and P. Feamhead, "Improved particle filter for non-linear problems," in *IEE Proc. on Radar and Sonar Navigation*, 1999.

[5] E. Parzen, "On the estimation of a probability function and the mode," *Annals of Mathematical Statistics*, vol. 33, no. 14, pp. 1065–1076, 1962.

[6] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Trans. on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[7] N. Bergman, "Recursive bayesian estimation: Navigation and tracking applications," Ph.D. dissertation, Linkoping University, Sweden, 1999.

[8] A. Doucet, "On sequential monte carlo sampling methods for bayesian filtering," Department of Engineering, University of Cambridge, UK, Tech. Rep., 1998.

[9] N. Gordon, D. Salmond, and A. F. M. Smith, "Novel approach to nonlinear and non-gaussian bayesian state estimation," in *IEE proceedings-F*, vol. 140, 1993, pp. 107–113.

[10] W. Wu, M. J. Black, D. Mumford, Y. Gao, E. Bienenstock, and J. P. Donoghue, "Modeling and decoding motor cortical activity using a switching kalman filter," *IEEE Trans. on Biomedical Engineering*, vol. 51, no. 6, pp. 933–042, June 2004.

[11] M. R. Mehta, M. C. Quirk, and M. Wilson, "A experience-dependent asymmetric shape of hippocampal receptive fields," *Neuron*, vol. 25, pp. 707–715, 2000.

[12] J. O'Keefe and J. Dostrovsky, "The hippocampus as a spatial map: Preliminary evidence from unit activity in the freely moving rat," *Brain Res.*, vol. 34, pp. 171–175, 1971.

[13] D. W. Moran and A. B. Schwartz, "Motor cortical representation of speed and direction during reaching," *J. Neurophysiol.*, vol. 82, pp. 2676–2692, 1999.

[14] A. E. Brockwell, A. L. Rojas, and R. E. Kass, "Recursive Bayesian Decoding of Motor Cortical Signals by Particle Filtering," *J. Neurophysiol.*, vol. 91, no. 4, pp. 1899–1907, 2004.

[15] E. N. Brown, R. Barbieri, V. Ventura, R. E. Kass, and L. M. Frank, "The time-rescaling theorem and its application to neural spike train data analysis," *Neural Computation*, vol. 14, pp. 325–246, 2001.