Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

A Hybrid Systems Model for Supervisory Cognitive State Identification and Estimation in Neural Prosthetics

N. Hudson and J.W. Burdick
Mechanical Engineering, California Institute of Technology, Pasadena, CA, 91125, USA
hudson@caltech.edu, jwb@robotics.caltech.edu

Abstract— **This paper presents a method to identify a class of hybrid system models that arise in cognitive neural prosthetic medical devices that aim to help the severely handicapped. In such systems a "supervisory decoder" is required to classify the activity of multi-unit extracellular neural recordings into a discrete set of modes that model the evolution of the brain's planning process. We introduce a Gibbs sampling method to identify the key parameters of a GLHMM, a hybrid dynamical system that combines a set of Generalized Linear Models (GLM) for dynamics of neuronal signals with a Hidden Markov Model (HMM) that describes the discrete transitions between the brain's cognitive or planning states. Multiple neural signals of mixed type, including local field potentials and spike arrival times, are integrated into the model using the GLM framework. The identified model can then be used as the basis for the supervisory decoding (or estimation) of the current cognitive or planning state. The identification algorithm is applied to extracellular neural recordings obtained from set of electrodes acutely implanted in the Posterior Parietal cortex of a rhesus monkey. The results demonstrate the ability to accurately decode changes in behavioral or cognitive state during reaching tasks, even when the model parameters are identified from small data sets. The GLHMM models and the associated identification methods are generally applicable beyond the neural application domain.**

I. Introduction

A "neural prosthetic" is a brain-machine interface that enables a human, via the use of surgically implanted electrode arrays and associated computer decoding algorithms, to control external electromechanical devices by pure thought alone. In this manner, some useful functions can be partially restored to patients with severe motor disorders (e.g. Lou Gehrig's disease) or with high level spinal cord injuries. *Cognitive neural prostheses* work by "decoding," or estimating, motor plans from the recorded electrical activity of multiple neurons in brain areas (such as the posterior parietal or dorsal premotor cortices) associated with motor planning. These decoded plans can be used to drive devices such as prosthetic arms or computer interfaces [1], [2], [3]. Future practical clinical neuroprostheses that seek to provide a facile interface for the paralyzed patient will require a *supervisory decoder* whose job is to classify, in real time, the *discrete* cognitive, behavioral, or planning *state* of the brain region from which the neural signals are recorded. I.e., the supervisory decoder must determine if: (1) the patient is asleep or disinterested in using the prosthetic; (2) the patient wishes to use the prosthetic; (3) the patient is planning an action that must be decoded; (4) the patient wants to execute the planned action; (5) the patient wants to scrub or change the current action. While the actual planning process in the brain is quite complex, for the purposes of supervisory decoding there are a finite number of cognitive states that model and govern the relevant activities of a brain-machine interface. The knowledge of the current state in the evolution of the planning process can be used in a variety of ways. For example, depending upon the current state, different algorithms, or different parameters in the algorithm, can be applied to the decoding of movement plans. Moreover, accurate knowledge of the current cognitive state will improve the action of the prosthetic system.

In this paper we model the neural processes related to the brain-machine interface as a hybrid dynamical system, where the discrete states are associated to the cognitive or planning brain states, and the continuous states model the observed neural activity, such as firing rate. Thus, the design of a supervisory decoder is a two part process: (1) the identification (or learning) of the hybrid model that represents neural activity in each discrete cognitive state as well as the transitions rules between cognitive states; (2) the design of an estimator which uses the identified hybrid model to classify the current neural activity into discrete cognitive states. In this paper we introduce a Gibbs sampling procedure to identify a *General Linear Hidden Markov Model* (GLHMM) class of hybrid dynamical that combine a discrete dynamical system that is governed by a Markov chain with continuous neural dynamics that can be modeled by a generalized linear model. The methods we use to decode, or estimate, the current GLHMM state using the identified model follow from existing procedures. While this class of dynamical systems is well suited to our motivating neural prosthetic application, the GLHMM models and methods presented in this paper can potentially be applied to other physical systems.

Section II reviews basic neural signal models. Section III describes the GLHMM class of hybrid dynamical models, while Section IV presents a procedure to identify GLHMMs using a two-stage Gibbs sampler. For purposes of demonstration and validation, Section V describes results obtained by applying the method to neural recordings in the parietal cortex of a macaque monkey while the animal carries out tasks that simulate the operation of neural prosthetic. Previous work [3], [4], [5] has demonstrated that the parietal reach region (PRR) in the posterior parietal cortex contains

978-1-4244-3124-3/08/$25.00 ©2008 IEEE

both motor planning activity as well neural correlates of the discrete cognitive and planning states needed for a supervisory controller.

II. Neurological Signal Models

The front end of a cortical neural prosthetic typically consists of a multi-electrode array implanted in cortical tissues [1], [2], [3], [5]. The signal recorded from each electrode contains multiple signals components that arise from different physiological origins and whose characteristics require different signal models, which are now briefly reviewed.

A. Local field potentials

Cortical local field potentials (LFP) arise from the aggregate dendritic electric potentials originating from neurons in a "listening sphere" that surrounds the electrically active tip of the recording electrode [4]. Such signals average the dendritic activity of a few thousand nearby neurons. In practice, the LFP signal component is derived by amplification and band-pass filtering (usually in the range 2-300 Hz) of the electrode signal. Historically, the LFP is modeled from the knowledge of its spectrogram [4], [6] which is optimally obtained from multitaper methods [7] which apply the Fourier transform to tapered time series obtained from the digitization of the LFP signal. Spectrograms of the LFP signal in the Parietal Cortex show that temporal variations of the power in certain frequency bands is correlated with intended arm reach direction, as well changes in planning state [4], [6]. The average power in each frequency band can be modeled as a random variable with a log normal distribution.

Autoregressive (AR) or vector autoregressive (VAR) equations can be used for parametric spectral estimation [7], and will be used here to model the LFP signal in the time domain. A pth order AR model, denoted AR(p), takes the form:

$$y_k = \sum_{i=1}^{p} \beta(i) y_{k-i} + \eta_k \ , \tag{1}$$

where $y_k \in \mathbb{R}$ is the LFP signal sampled at time t_k, and $\eta_k \sim \mathcal{N}(0, \sigma^2)$ is zero mean noise with covariance σ^2, and model parameters $\boldsymbol{\theta} = \left\{\beta(1), ..., \beta(p), \sigma^2\right\}$ Note that the spectral density of a stationary AR(p) process (1) is given by [7]:

$$S(f) = \frac{\sigma^2 \Delta t}{\left|1 - \sum_{j=1}^{p} \beta(j) \exp^{-i2\pi f j \Delta t}\right|^2} \ , \tag{2}$$

where f is the frequency, and Δt is the sampling period. While the spectrogram has been the primary LFP modeling tool in prior work, there are two main advantages of using time domain based AR models instead of frequency domain spectrogram methods. First, the real-time computation of the spectrogram is an excessive practical burden, and it additionally introduces a time lag in the response of the neural prosthetic system since a large window size (typically 512 or 1024 msec) is needed to obtain good precision. This lag may cause undesirable psychophysical delays for the prosthetic-using patient. The AR approach effectively uses considerably smaller window sizes: the experiments of Section V use a 55th order model on data LFP data sampled at 1kHz, resulting in an effective window width of 55msec.

B. Single Unit Activity

Neurons generate characteristic electrical pulses called action potentials, or spikes, whose arrival times, and not waveform shape, are believed to encode information. Mathematical models used to decode neural stimuli typically focus on the firing (spiking) rate of individual neurons [8]. Numerous studies have shown that *single unit activity*[1] can be correlated to intended reach direction, as well as temporal or cognitive state in the posterior Parietal cortex [4], [6]. Following standard practice, the spike arrival times are discretized into sufficiently small time bins (1 msec in our experiments) so that only one spike at most is assigned to each bin. Let the beginnings of each discretized sampling interval be denoted by the sequence of times $\{t_1, t_2, \ldots, t_k, \ldots, t_T\}$. Thus, each bin corresponds to the time interval $(t_k, t_{k+1}]$. The signal y_k is the number of spikes arriving in the interval $(t_k, t_{k+1}]$. When the bin size is sufficiently small, the spike arrival times can be modeled as a point process with a stationary Poisson distribution:

$$f(y_k, \lambda) = \frac{\lambda^{y_k} e^{-\lambda}}{y_k!} \ , \tag{3}$$

where λ is the firing rate of the neuron, and is the only parameter of the model ($\boldsymbol{\theta} = \lambda$). In previous studies [9], [10], we have used a more versatile non-stationary point process model [11] to represent single spiking unit activity. In this similar model, the non-stationary firing rate is a log linear function of the neuron's spiking history:

$$\lambda_k = \exp\left[\beta(0) + \sum_{i=1}^{p} \beta(i) y_{k-i}\right] \ . \tag{4}$$

While the methods presented in this paper handle this more general model, is not considered here due to the sparsity of the neural action potentials in the example of Section V.

III. Supervisory Decoder Model

This section describes a generalized linear hidden Markov model (GLHMM) hybrid dynamical system model that represents the discrete cognitive states and their transitions by a Hidden Markov Model (HMM), and uses Generalized Linear Models (GLM) to represent neural activity in each cognitive state. Note that both the AR-LFP models (1) and point process single unit models (3)-(4) are examples of generalized linear models. See [12] and references within for additional information on GLMs.

[1]The action potentials, or spikes, of more than one neuron may be recorded on a single electrode. A two step process isolates the activity of a single neuron, or unit. First, spike waveforms are detected (in the midst of substantial background noise) in the electrical signal. The detection process also provides an estimate of the spike waveform's *arrival time*, the time at which the spike amplitude peaks. A *spike sorting* process [5] then analyzes the waveform shapes and clusters the waveforms according to different putative neural signal sources.

A. Generalized Linear Hidden Markov Model

A GLHMM is formed around a set of d unobservable discrete states, $\mathcal{S} = \{S_1, S_2, \ldots, S_d\}$, whose evolution is governed by a first order Markov process. In our application, these discrete states model the different cognitive or planning states of the cortical area involved in the brain-machine interface. At each t_k, let m_k denote the mode index, i.e., at t_k the system is in state S_{m_k}. The probability of switching between modes of the system is governed by a first order Markov chain with transition matrix $A = [a_{i,j}]$:

$$P\left(m_k = j \middle| \, m_{k-1} = i\right) = a_{i,j} \ . \tag{5}$$

Let the vector $\boldsymbol{y}_k = \left[y_k^1, \ldots, y_k^n\right]^T$ denote the n neural signals measured at t_k. Each neural signal y_k^i is modeled as a generalized linear model, with parameters that depend on the discrete mode S_{m_k}:

$$y_k^i \sim f_i\left({g_i}^{-1}\left(\boldsymbol{\theta}_{m_k}^i \boldsymbol{x}_k^i\right)\right) \ . \tag{6}$$

Here f_i is a *probability distribution* from the exponential family, and g_i is a *link function*. The *linear predictor*, $\boldsymbol{\theta}_{m_k}^i \boldsymbol{x}_k^i$ is composed of a regressor $\boldsymbol{x}_k^i$ of n_y previous subset outputs and other system covariates or inputs $\boldsymbol{u}_k$:

$$\boldsymbol{x}_k^i = \left[y_{k-1}^i, \ldots, y_{k-n_y}^i, {\boldsymbol{u}_k}^T\right]^T \ , \tag{7}$$

and a corresponding parameter vector $\boldsymbol{\theta}_{m_k}^i$.

For example, the AR model for LFP activity can be written in the GLM form (6) by setting f as the normal distribution $f = \mathcal{N}(\cdot, \sigma^2)$, using identity for the link function g, and simply using the output history in the regressor $\boldsymbol{x}_k$:

$$y_k^i \sim \mathcal{N}\left(\beta_{m_k}^i \boldsymbol{x}_k^i, \sigma_{m_k}^{2\ i}\right) \tag{8}$$

Likewise both the stationary and non stationary Poisson point process models for single unit activity can be expressed in GLM form.

IV. Identification of GLHMM models

This section describes a two-stage Gibbs sampler to identify GLHMM model parameters from an observed data set, $\boldsymbol{y}_{1:T}$, where

$$\boldsymbol{y}_{1:T} = [\boldsymbol{y}_1, ..., \boldsymbol{y}_T] \ . \tag{9}$$

Using a probabilistic formulation of the problem, we seek to estimate the posterior density of the GLHMM parameters:

$$p\left(\Theta \middle| \, \boldsymbol{y}_{1:T}\right) \ , \tag{10}$$

where Θ is the set of all model parameters. Model identification is obtained from the mode of this distribution. Assuming that there are n total signal sources, with n_{LFP} LFP signals and n_{spikes} single unit signals, and d discrete modes, then without loss of generality:

$$\Theta = \left\{\boldsymbol{\theta}_m^i, A\right\}_{i=1}^n, \text{for } m = 1, ..., d \ , \tag{11}$$

where:

$$\boldsymbol{\theta}_m^i = \begin{cases} \lambda_m^i & \text{if } i <= n_{spikes} \\ [\beta_m^i(1), ..., \beta_m^i(p), \sigma_m^i] & \text{if } i > n_{spikes} \end{cases} \ , \tag{12}$$

and the variables β_m^i were defined in Equation (1).

Instead of estimating (10) directly, we instead consider the joint density function of the parameters Θ and the latent discrete modes $m_{1:T}$:

$$p\left(\Theta, m_{1:T} \middle| \, \boldsymbol{y}_{1:T}\right) \ , \tag{13}$$

as (10) is the marginal distribution of (13):

$$p\left(\Theta \middle| \, \boldsymbol{y}_{1:T}\right) = \sum_{m_{1:T} \in \mathcal{S}^T} p\left(\Theta, m_{1:T} \middle| \, \boldsymbol{y}_{1:T}\right) \ . \tag{14}$$

A two stage Gibbs sampler is used to draw samples from the distribution (13), from which the identification process is realized. Alternative methods for analyzing or maximizing this distribution include the Expectation Maximization algorithm [13], [14], as well as gradient and variational methods.

A. Gibbs Sampling

Gibbs sampling is an MCMC method for sampling from a potentially complicated joint pdf, $p\left(\phi_1, ..., \phi_n\right)$, where $\phi_1, \ldots, \phi_n$ are system states or parameters. Gibbs sampling can be usefully applied when the joint pdf $p\left(\phi_1, ..., \phi_n\right)$, has associated conditional pdfs,

$$p\left(\phi_1 \middle| \, \phi_2, ..., \phi_n\right), \ldots, p\left(\phi_i \middle| \, \phi_1, ..., \phi_{i-1}, \phi_{i+1}, ..., \phi_n\right), \\ \ldots, p\left(\phi_n \middle| \, \phi_1, ..., \phi_{n-1}\right) \ ,$$

which can be efficiently sampled (e.g., there is a closed form solution for the pdf). A single step in the Gibbs sampling cycle requires one sample to be drawn sequentially from each of the conditional pdfs, using the most recent sampled value in subsequent conditional arguments. At the end of each step, a new sample $\hat{\phi} = \left[\hat{\phi}_1, \ldots, \hat{\phi}_n\right]$ has been drawn. As the Gibbs sampler iterates through many steps, the samples $\left\{\hat{\phi}\right\}$ tend to the joint distribution [15]. In theory this property implies that the maximum of $p\left(\phi_1, ..., \phi_n\right)$, can always be found using a sufficiently lengthy Gibbs sampling process, as opposed to Expectation Maximization methods where only a local maximum is guaranteed. In practice, only a finite number of samples are drawn, and multiple runs of the Gibbs sampling algorithm from different starting points are usually conducted to test for convergence. The Gibbs sampling process allows the joint pdf (13) to be naturally decomposed into the component parts of identification, $p\left(\Theta \middle| \, M, \boldsymbol{y}_{1:T}\right)$, and classification, $p\left(M \middle| \, \Theta, \boldsymbol{y}_{1:T}\right)$. This method is well suited to high dimensional problems, and has asymptotic convergence properties [16].

B. Gibbs Sampling Algorithm for GLHMM

Algorithm 1 defines a two stage Gibbs sampling method for GLHMMs. This algorithm draws i_{max} samples from the joint distribution $p\left(\Theta, m_{1:T} \middle| \, \boldsymbol{y}_{1:T}\right)$. Let the ith sample of a variable, Θ, be denoted: $\hat{\Theta}^{(i)}$.

Algorithm 1 Gibbs Sampling for GLHMM

1: Initial mode estimate: $\hat{m}_{1:T}^{(0)}$
2: **for** $i = 0$ to i_{max} **do**
3: $\quad \hat{\Theta}^{(i+1)} \sim p\left(\Theta|\ \hat{m}_{1:T}^{(i)}, \boldsymbol{y}_{1:T}\right)$
4: $\quad \hat{m}_{1:T}^{(i+1)} \sim p\left(m_{1:T}|\ \hat{\Theta}^{(i+1)}, \boldsymbol{y}_{1:T}\right)$
5: **end for**

The core of algorithm involves sequentially drawing samples from the two conditional distributions shown in lines 3 and 4 of Algorithm 1, and can be considered identification and classification steps respectively. The following describes how to efficiently draw samples from the conditional distributions in Algorithm 1:

1) Parameter Conditional Distribution:

$$p\left(\Theta|\ m_{1:T}, \boldsymbol{y}_{1:T}\right) \tag{15}$$

The model parameters Θ, conditioned on discrete cognitive modes $m_{1:T}$ are independent. This can be shown by splitting the data $\boldsymbol{y}_{1:T}$ into discrete sets, creating d independent identification problems. $\boldsymbol{y}_{1:T}$ is split as follows:

$$\mathcal{Y}_i = \{\boldsymbol{y}_k\ :\ m_k = i,\ k = 1, \ldots, T\}\ . \tag{16}$$

The parameter distribution (15) then becomes:

$$\prod_{m=1}^{d} p\left(\boldsymbol{\theta}_m^1, ..., \boldsymbol{\theta}_m^n|\ \mathcal{Y}_m\right) p\left(A|\ m_{1:T}\right)\ . \tag{17}$$

Furthermore, assuming independence between the n signals,

$$p\left(\boldsymbol{\theta}_m^1, ..., \boldsymbol{\theta}_m^n|\ \mathcal{Y}_m\right) = \prod_{i=1}^{n} p\left(\boldsymbol{\theta}_m^i|\ \mathcal{Y}_m\right)\ , \tag{18}$$

where the individual distributions $p\left(\boldsymbol{\theta}_m^i|\ \mathcal{Y}_m\right)$ can be sampled from efficiently as they are in the GLM family, using either standard methods or using adaptive rejection sampling [17].

Each row, $a_{(i,1:d)}$, of A, the discrete Markovian kernel, can be sampled from independently [18], [14]:

$$p\left(A|\ m_{1:T}\right) = \prod_{i=1}^{d} p\left(a_{(i,1:d)}|\ m_{1:T}\right)\ . \tag{19}$$

To sample from each row, the number of transitions in $m_{1:T}$ are counted:

$$p\left(a_{i,1:d}|m_{1:T}\right) = \mathcal{D}\left(\alpha_{i,1} + \sum_{k=2}^{T} \delta_{(m_{k-1}=i)}\delta_{(m_k=1)}, \ldots, \alpha_{i,d} + \sum_{k=2}^{T} \delta_{(m_{k-1}=i)}\delta_{(m_k=d)}\right)\ , \tag{20}$$

where $\mathcal{D}$ is the Dirichlet distribution, δ is the dirac delta function, and the $\alpha_{i,j}$ are derived from priori information, and can be used to bias and constrain allowable transitions (see Sec. IV-C).

2) Discrete Cognitive State Probabilities:

$$p\left(m_{1:T}|\ \boldsymbol{y}_{1:T}, \Theta\right) \tag{21}$$

The discrete cognitive states $m_{1:T}$ can be sampled from jointly by using dynamic programming, specifically the forward filter (Algorithm 2) and then Markovian backwards sampling [14] (Algorithm 3)

Forward filtering is a dynamic programming process that yields the forward distributions:

$$p\left(m_k = i|\ \boldsymbol{y}_{1:k}, \Theta\right) \tag{22}$$

This algorithm can be derived directly using the Bayes theorem, and is used in the hidden Markov model literature [13].

Algorithm 2 Forward Filtering

1: $p\left(m_1 = i|\ \boldsymbol{y}_{1:1}, \Theta\right) = \pi(i)$
2: **for** $k = 2$ to T **do**
3: $\quad c_k = \sum_{i=1}^{d} p\left(m_k = i|\ \boldsymbol{y}_{1:k-1}, \Theta\right) p\left(\boldsymbol{y}_k|\ m_k = i, \Theta\right)$
4: $\quad p\left(m_k = j|\ \boldsymbol{y}_{1:k}, \Theta\right) = p\left(m_k = j|\ \boldsymbol{y}_{1:k-1}, \Theta\right) p\left(\boldsymbol{y}_k|\ m_k = i, \Theta\right)/c_k$
5: $\quad p\left(m_{k+1} = j|\ \boldsymbol{y}_{1:k}, \Theta\right) = \sum_{i=1}^{d} p\left(m_k = i|\ \boldsymbol{y}_{1:k}, \Theta\right) p\left(m_{k+1} = j|\ m_k = i, \Theta\right)$
6: **end for**

The normalizing constant c_k in line 3 is shorthand for $c_k = p\left(\boldsymbol{y}_k|\ \boldsymbol{y}_{1:k-1}, \Theta\right)$, and is the usual normalizing constant used in this algorithm.

Markovian backwards sampling, Algorithm 3, can then be used to simulate values of the discrete state $m_{1:T}$.

Algorithm 3 Markovian Backward Sampling

1: $m_T \sim p\left(m_T|\ \boldsymbol{y}_{1:T}, \Theta\right)$
2: **for** $k = T - 1$ to 1 **do**
3: $\quad m_k \sim p\left(m_k|\ \boldsymbol{y}_{1:k}, m_{k+1}, \Theta\right)$
4: **end for**

Note that the distributions in Algorithm 3, can be broken down into the forward distributions (22), and the Markov transition probabilities, by applying Bayes' theorem:

$$p\left(m_k = i|\ \boldsymbol{y}_{1:k}, m_{k+1} = j, \Theta\right) = \frac{p\left(m_k = i|\ \boldsymbol{y}_{1:k}, \Theta\right) p\left(m_{k+1} = j|\ m_k = i\right)}{\sum_{i=1}^{d} p\left(m_k =_i|\ \boldsymbol{y}_{1:k}, \Theta\right) p\left(m_{k+1} = j|\ m_k = i\right)}\ , \tag{23}$$

C. Prior Information

Constraints can be incorporated into the identification problem in a natural way through the use of prior distributions. While priors are used on all distributions, the most significant benefits are derived from instructive priors on the Markov transition matrix A and firing rate λ priors. For efficiency, conjugate prior distributions are used [16].

To avoid degeneracy in estimation, firing rates are constrained to be non-zero. This is a significant issue, as neural data streams with low firing rate may contain intervals in

which no action potentials are observed, leading to maximum likelihood estimates of zero firing rate, $\lambda = 0$. To overcome this problem, we use a gamma prior that represents physiologically plausible firing rates.

$$p\left(\lambda_k^i\right) = f(\lambda_k^i; \alpha, \beta) = (\lambda_k^i)^{\alpha-1} \frac{\beta^\alpha e^{-\beta} \lambda_k^i}{\Gamma(\alpha)} \quad , \tag{24}$$

where $\alpha = 2$, and $\beta = 100$.

The prior distribution of the Markov transition matrix, A, constrains how the model can transition between discrete states. A Dirichlet distribution is used for each row of A:

$$p\left(a_{i,1:d} | m_{1:T}\right) = \mathcal{D}\left(\alpha_{i,1}, \alpha_{i,2}, ..., \alpha_{i,d}\right) \quad , \tag{25}$$

where the parameters $\alpha_{i,j} \in \mathbb{N}$, the non-negative integers (including 0). For the neural prosthetic application in Sec. V, we have four discrete modes, which occur in a cyclic order. This transition structure is imposed on the GLHMM by introducing the following prior parameters:

$$[\alpha_{i,j}] = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \quad . \tag{26}$$

For example, this prior choice constrains the first discrete state S_1 to transition into S_2. Indeed this constraint leads to the desired behavior of $S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_4 \rightarrow S_1, ...,$. In general if $\alpha_{i,j} > 0$, then the transition $S_i \rightarrow S_j$ is allowed.

D. Model Order Selection

The AR order of the LFP signal models are automatically selected using the Bayesian information criterion (BIC) [19]. BIC is defined as:

$$BIC = 2 \ln L - k \ln(N) \tag{27}$$

where N are the number of observations, k are the number of free parameters, and L is the likelihood value for the estimated model. For GLHMMs, $k = (n_{LPF})pd + n_{spikes}d$, where p is the AR order, d is the number of discrete states, and n_{LFP} is the number of LFP signals. The number of observations is the number of elements in $\boldsymbol{y}_{1:T}$.

Instead of calculating the AR order p for every neural data set used in Sec. V, we chose a single data set, and calculated the BIC as follows: First we identified a GLHMM with a nominal AR order p_0. Then we used the corresponding discrete states $m_{1:T}$, to break the data set up into d sets, each set corresponding to a discrete mode. For each of these sets, we found the maximum likelihood estimate of the parameters, for AR order $p = 1, ..., 100$. The BIC value was computed for each p, and the maximum BIC estimate was chosen.

It is assumed that the number of discrete cognitive states d is a fixed and known number. This is appropriate for our neural prosthetics application where the discrete states have physical meaning and are related to experimental events, however for other applications, several models orders d should be selected and compared.

E. Estimation with GLHMMs: The Viterbi Algorithm

The Viterbi Algorithm [13] is typically used to find the single best hidden discrete state sequence $M = \{m_1, ..., m_T\}$ for hidden Markov models. It should be noted that it is not a causal filter, as it uses all observations $\{y_1, ..., y_k, ..., y_T\}$ when estimating the kth discrete mode m_k. Due to the Markov transition properties of the GLHMM model, it is simple to extend and apply here.

V. Application to Neural Data Set

We applied the GLHMM identification procedure described above to a neural data obtained from experiments with rhesus monkeys. This data set consists of recordings from two male rhesus (*Macaca mulatta*) monkeys, *Animal C* and *Animal D*, from electrodes placed in various positions within the Parietal Reach Region (PRR) of the posterior parietal cortex. The neural data set contains both LFP and neural spike arrival time signals [4]. While we have analyzed data from both animals, for brevity the results presented below focus on the 96 electrode recordings from animal *D*.

The data recordings occurred while the monkeys repetitively executed a *delayed center-out reaching task*, which is commonly used to simulate the actions of a neural prosthetic. Such simulations are a necessary step in the development of this technology for eventual human use. This task is illustrated in Fig 1. A task-board is placed within arm reaching distance in front of the monkey's visual field of view. Each trial proceeds as follows. A light located in the center of the task board is illuminated, and the monkey must place its reaching arm on the light to indicate that it is attending to the trial. A *target* light is flashed at one of 8 *target locations* around the task board perimeter for a short *cue period*, and then the target light is extinguished. After a random time delay (the *memory period*, during which the monkey must remember the target location and also plan its upcoming reach to the target), the center light is extinguished, cueing the monkey to reach to the remembered target location. If the monkey successfully touches the correct target location, while also respecting the temporal structure of the sequence, it is given a juice reward.

To simulate the action of a neural prosthetic, the neural signals from the PRR are "decoded" during the memory period (when the monkey can only be planning a reach, and not executing a physical reach), to predict the monkey's subsequent physical reach, even before the reach occurs. Successful prediction of the subsequent reach from the memory period signals validates the ability to decode a reaching plan from PRR neural activity. Practically, these experiments demonstrate, for example, the ability of the brain machine interface to control cursor movements on a computer screen using neural signals. Such cursor control is a basic function that would allow paralyzed patients to use a computer. In more advanced experiments that more accurately simulate a neural prosthetic, the monkey is taught to purely think about the reach to the target, and the desired cursor command is decode from this thought [5].

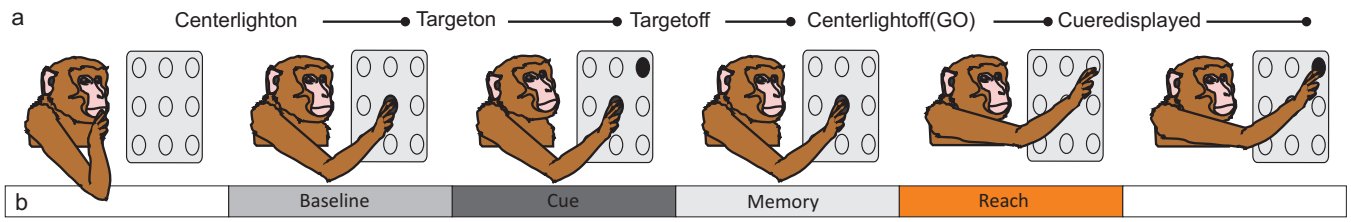

Fig. 1. Center-out reach experiment: a) Displayed experimental cues, b) corresponding cognitive or behaviorial states.

We can also use these trials to simulate and validate a supervisory decoding system. The trial structure has an associated discrete number of different cognitive and planning states: (1) a *baseline state* where the monkey is idle, or starting to attend to the upcoming trail; (2) a *cue period* during which the target location is lit; (3) a *memory period* during which the location of the now extinguished target must be remembered by the monkey, and during which the monkey plans its upcoming arm movement; (4) a short"go" period (which is really a transition between memory and execution states) during which the planned movement is initiated; and (5) a *reach* or *execute* period during which the arm moves to the target location. To successfully simulate a supervisory decoder, we seek to demonstrate that the onset and duration of these different planning/cognitive periods can be correctly estimated solely from the neural signals recorded during the trial. The actual behavior of the monkey is actively recorded during the task execution, providing us with a reasonably good ground truth model against which the predictions can be compared. We are particularly interested in estimating the onset of the reach state (the "go" signal). In an neural prosthetic, this signal will trigger the execution of an action associated to the decoded planning activity.

Training (identification) and testing (estimation) data sets were created by randomly choosing an n electrode subset, $\mathcal{E}_n$, from the set of available electrode signals, $\mathcal{E}$. We limited our selection to the subsets of the recorded data which included at least 7 successful reaches in each of 8 possible reach directions, and whose signal to noise quality exceeded a threshold [4]. From the data set $\mathcal{E}_n$, two reach trials in each of the 8 directions were randomly chosen (16 trials total) to form the training set. A testing data set was formed from the remaining 5 trials in each of the 8 directions (40 trials total). This is done 7 times for the given value of n, where n was varied from 1 to 6, resulting in 42 data sets, each containing 16 training trials and 40 testing trials.

For each of the 42 training data sets, a GLHMM was identified using the Gibbs sampling method described in Sec. IV. Estimation of the discrete cognitive state in the corresponding testing data sets is then done using the Viterbi algorithm. For each trial, a discrete state estimation, or decode, was considered *correct* when the reach state was decoded within a 300 ms window of the 'go' signal, as shown in Fig 2. The exact timing of the go signal is obtained experimentally by watching for the onset of the monkey's arm motion. The average percentage of correct decoding trials versus the number of electrodes n is shown in Fig 3. The error bars represent standard deviations of percent correct over the seven repetitions described above.

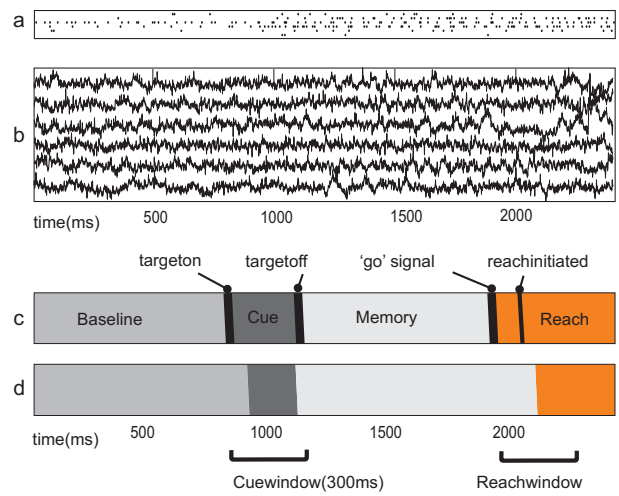

Fig. 2. Example decode: a) Recorded neural spike arrival times for 6 electrodes. b) Local field potentials for 6 electrodes. c) Cognitive states as defined by experimental cues. d) Decoded cognitive state.

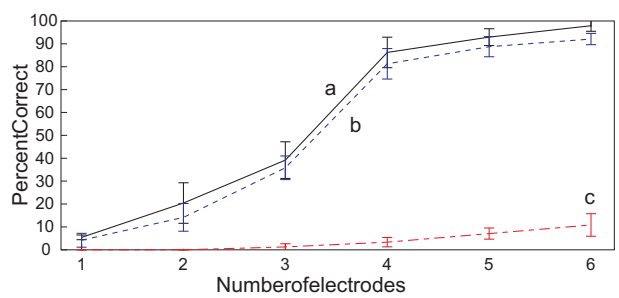

Fig. 3. Decoding results: Percentage of correctly decoded trials, where the reach state was correctly estimated within a finite time window of the actual reach occurring: a) Decoding using both LFP and single unit activity. b) Decoding using only LFP signals. c) Decoding using only single unit activity.

Figure 3 shows that a high level of decoding performance can be achieved using a relatively small number of electrodes. This is a promising result, as the surgical complexity and risk associated with the implantation of the electrodes is proportional to the number of electrodes.

In addition to the high percent correct of decodes, the lag between the estimated onset of the reach state and the actual reach is small, 0.027s on average. From the psychophysical point of view, this is a negligible lag.

To relate the identified models back to the science con-

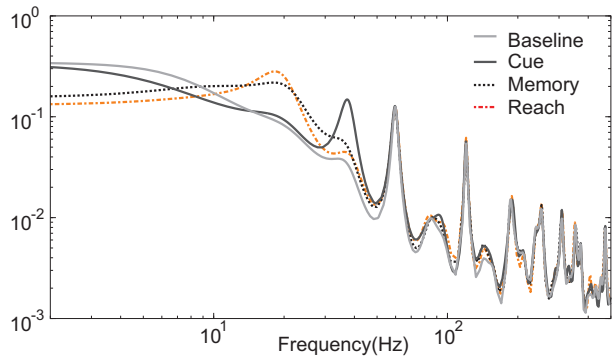

Fig. 4. Power spectrum of the identified AR models for each discrete cognitive state for a single electrode

ducted in [4], we can calculate the power spectral density (PSD) of the identified models, and show that we recover similar phenomenon, of changing power in different frequency bands through time in LFP signals. The PSD of the 55th AR models for a single electrode in each discrete state is shown in Fig 4. The AR models in each discrete state show the characteristic noise peak at 60Hz and 120Hz, however the large discrepancy at lower frequencies is consistent with the ranges considered in other studies using spectrographic methods.

VI. Conclusion

This paper presented a new method, based on a 2-stage Gibbs sampling process, to identify the parameters of a General Linear Hidden Markov Model class of hybrid dynamical systems. This procedure naturally decomposes the hybrid system identification problem into parameter identification and discrete state clustering components. While the GLHMM model is very well suited to our motivating application, it should be useful for modeling other physical systems. We demonstrated the use of these techniques on actual neural recordings–signals which are characteristically noisy and only weakly stationary.

As presented, our method is limited to stationary Hidden Markov models with an apriori known number of discrete states. Future work will focus on extending this methodology to include non-stationary Hidden Markov Models, and to allow automatic selection of the most probable number of discrete states.

Acknowledgment

The authors would like to thank Hans Scherberger and Richard Andersen for providing the neural data used in the demonstration of Section V. This work has been supported by the National Institutes of Health and the Defense Advanced Research Projects Agency.

References

[1] B. Pesaran, S. Musallam, and R. Andersen, "Cognitive neural prosthetics," *Current Biology*, vol. 16, pp. 77–80, 2006.

[2] M. A. L. Nicolelis, "Brainmachine interfaces to restore motor function and probe neural circuit," *Nature Reviews Neuroscience*, vol. 4, pp. 417–422, 2003.

[3] K. Shenoy, D. Meeker, S. Cao, S. Kureshi, B. Pesaran, C. Buneo, A. Batista, P. Mitra, J. Burdick, and R. Andersen, "Neural prosthetic control signals from plan activity," *Neuroreport*, vol. 14, no. 4, pp. 591–596, 2003.

[4] H. Scherberger, M. Jarvis, and R. Andersen, "Cortical local field potential encodes movement intentions in the posterior parietal cortex," *Neuron*, vol. 46, p. 347354, 2005.

[5] S. Musallam, B. D. Corneil, B. Greger, H. Scherberger, and R. A. Andersen, "Cognitive control signals for neural prosthetics," *Science*, vol. 305, pp. 258–262, 2004.

[6] B. Pesaran, J. S. Pezaris, M. Sahani, P. P. Mitra, and R. A. Andersen, "Temporal structure in neuronal activity during working memory in macaque parietal cortex," *Nature Neuroscience*, vol. 5, pp. 805 – 811, 2002.

[7] D. B. Percival and A. T. Walden, *Spectral analysis for physical applications: multitaper and conventional univariate techniques*. Cambridge University Press, 1993.

[8] P. Dayan and L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. MIT Press, 2001.

[9] N. Hudson and J. Burdick, "A stochastic framework for hybrid system identification with application to neurophysiological systems," *Lecture Notes in Computer Science : Hybrid Systems: Computation and Control*, pp. 568–582, 2007.

[10] ——, "Learning hybrid system models for supervisory decoding of discrete state, with applications to the parietal reach region," *Neural Engineering, 2007. CNE '07. 3rd International IEEE/EMBS Conference on*, pp. 587–592, 2-5 May 2007.

[11] W. Truccolo, U. T. Eden, M. R. Fellows, J. P. Donoghue, and E. N. Brown, "A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects," *J Neurophysiol*, vol. 93, no. 2, pp. 1074–1089, 2005.

[12] P. Dellaportas and A. F. M. Smith, "Bayesian inference for generalized linear and proportional hazards models via gibbs sampling," *Applied Statistics*, vol. 42, no. 3, pp. 443–459, 1993.

[13] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257–286, 1989.

[14] O. Cappe, E. Moulines, and T. Ryden, *Inference in Hidden Markov Models*. Springer, 2005.

[15] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, Eds., *Markov Chain Monte Carlo in Practice*. Chapman & Hall, 1996.

[16] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. Springer, 2004.

[17] W. R. Gilks, N. G. Best, and K. K. C. Tan, "Adaptive rejection metropolis sampling within gibbs sampling," *Applied Statistics*, vol. 44, no. 4, pp. 455–472, 1995.

[18] C. Robert, G. Celeux, and J. Diebolt, "Bayesian estimation of hidden Markov chains: A stochastic implementation," *Statistics & Probability Letters*, vol. 16, pp. 77–83, January 1993.

[19] G. McLachlan and D. Peel, *Finte Mixture Models*, ser. Wiley series in probability and statistics. Wiley Interscience Publication, 2000.