2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

A Miniature Robot for Isolating and Tracking Neurons in Extracellular Cortical Recordings

Michael T. Wolf, Jorge G. Cham, Edward A. Branchaud, and Joel W. Burdick
Mechanical Engineering, California Institute of Technology, Pasadena, CA, 91125
wolf@caltech.edu, jwb@robotics.caltech.edu

Abstract— **This paper presents a miniature robot device and control algorithm that can autonomously position electrodes in cortical tissue for isolation and tracking of extracellular signals of individual neurons. Autonomous electrode positioning can significantly enhance the efficiency and quality of acute electrophysiolgical experiments aimed at basic understanding of the nervous system. Future miniaturized systems of this sort could also overcome some of the inherent difficulties in estabilishing long-lasting neural interfaces that are needed for practical realization of neural prostheses. The paper describes the robot's design and summarizes the overall structure of the control system that governs the electrode positioning process. We present a new sequential clustering algorithm that is key to improving our system's performance, and which may have other applications in robotics. Experimental results in macaque cortex demonstrate the validity of our approach.**

I. Introduction

This paper describes a miniaturized robotic mechanism and an associated set of algorithms for extracellular neural recording. Much of what we know about brain function has been provided by extracellular neural recordings, which are obtained by positioning the tip of an electrode near an individual neuron. Extracellular recordings are also the key inputs to emerging neuroprosthetic devices that promise to aid the severely handicapped [1], [2], [3], [4]. However, a fundamental problem lies in creating a neural recording interface that can reliably and stably record neural signals for long periods of time (e.g., for a few years). The robotic system described in this paper can autonomously position recording electrodes in cortical tissue in order to initially optimize the recorded neuronal signal and subsequently track the cell over shifts in position and signal characteristics. As we describe below, our autonomous electrode paradigm can increase the efficiency and quality of basic neurobiological research, and also offers the potential to overcome some of the difficulties in making the long-lasting stable neural interfaces that are needed for neural prostheses.

The goal of extracellular recordings is to detect and localize in time the occurrence of a neuron's electrical impulses, termed *action potentials* or *spikes*, which are the basis for neural communication and information processing. Neural recordings can generally be categorized as *acute* (typically lasting several hours) or *chronic* (lasting weeks, months, or years). In acute experiments, which are primarily used for scientific research, an electrophysiologist commands a *microdrive* (an electromechanical device which can position an electrode along a linear track in micron-scale steps) to advance electrodes into neural tissue until the action potentials of a neuron are clearly discernible, or "isolated." This process of neural isolation is something of an art, and is typically guided by both visual (oscilliscope) and audio (relaying the amplified recording to a loudspeaker) means. The isolation process requires significant training to master, and also consumes a significant amount of the experimenter's time. Moreover, after the initial isolation, the microdrive must be repeatedly repositioned by the experimenter to maintain a high quality signal in the face of neural drift that arises from tissue relaxation.

Chronic recordings use implanted arrays of stationary electrodes. The signal yield of the array depends, however, on the luck of the initial surgical placement. High impedance electrodes have a limited "listening sphere," and their tips must be positioned within 30 microns (and preferably within 10 microns) from a neuron's soma in order to obtain a high quality signal. It is generally impossible for all of the implanted electrode tips to fall within an active recording region. Moreover, blood pressure variations and small mechanical shocks will cause tissue migrations, and thus subsequent degradation of the signal. Finally, reactive gliosis can encapsulate the electrode, diminishing signal quality.

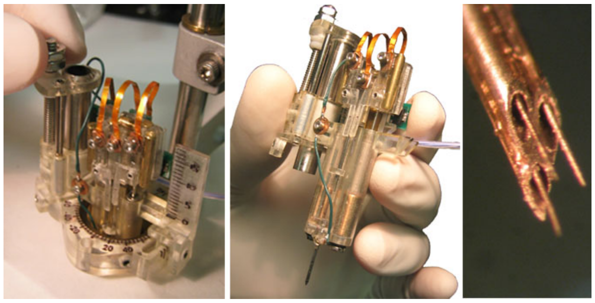

Fig. 1. Photographs of the Robotic Electrode Microdrive (left and center), and a closeup of electrodes extending from their the guide tubes (right)

We describe below a control system that can autonomously position electrodes so as to isolate and then maintain high quality extracellular recordings. We have used this algorithm for thousands of hours on several existing commercial microdrives [5]. We also describe a novel miniature robotic microdrive that can finely position three electrodes independently. A first generation microdrive prototype has been previously reported in [6], [7], and an early version of our control system was described in [8], [9]. The robot described in this paper is a substantial improvement over that earlier design in terms of manufacturability, ease of use,

978-1-4244-1647-9/08/$25.00 ©2008 IEEE.

and signal quality. Additionally, we have developed a new signal clustering, classification, and tracking algorithm that is at the core of our approach. Not only does this algorithm significantly improve the operation of our control system, it should also have applications to other areas of robotics.

Our autonomous electrode positioning paradigm has applications to various types of neural recordings. For acute scientific recordings our approach can increase the efficiency and quality of neural signal recording by freeing the experimenter from the tedious task of positioning electrodes. As compared to traditional manual positioning of the electrodes, our automated approach also significantly increases the number of electrodes that can be simultaneously positioned during an acute recording, thereby increasing the quality of the scientific results. Our microdrive's small, light design can enable it to remain on a subject (non-human primate) for days at a time, allowing neuroscientists to study issues of neural plasticity. The microdrive presented in this paper also serves as a testbed to develop to parameters necessary for our ongoing efforts to build large arrays of individually actuated micro-electrodes using MEMS technology [10]. These miniaturized implantable arrays would serve as the front end for neuroprosthetic systems, and their autonomous positioning capability would help to overcome the low signal yield of chronic fixed electrode arrays. Muthuswamy et al. [11] also are working toward implantable movable electrodes, but actuators with longer range and lower power are needed, as well as an accompanying control algorithm such as ours.

Section II describes the robotic microdrive mechanism, including some pertinent manufacturing details, in the context of the requirements of chronic and semi-chronic neural recording devices. Section III describes the structure of the control algorithm, including recent developments in the finite state machine designed to address the challenges of realistic recording environments. Section IV highlights a new method of neuron tracking via a new Bayesian classification algorithm. This algorithm potentially has applications to other signal processing problems in robotics.

II. Mechanism Design and Fabrication

Microdrives suitable for chronic use must be implantable, safe, minimally obtrusive, and require minimal maintenance. Preferably, the device should be small in size and mass in order to allow the experimental subject free movement and comfort. Conversely, the compactness and proximity of all the electrical pathways in a small device can increase noise and interference in the recorded signal. Size limitations also restrict the number of actuators, and hence recording electrodes, that can be packaged in the device. The device must be secure against leaks and impacts and easily sterilized. Commercially available microdrives (for example: Thomas Recording GmbH, Germany; FHC Inc., USA; Narishige Inc., Japan) are too large to be practical for chronic use. Previous chronic microdrives [12], [13], [14], [15] all require manual repositioning of the electrodes (e.g., via lead screws or temporary interfaces to actuators).

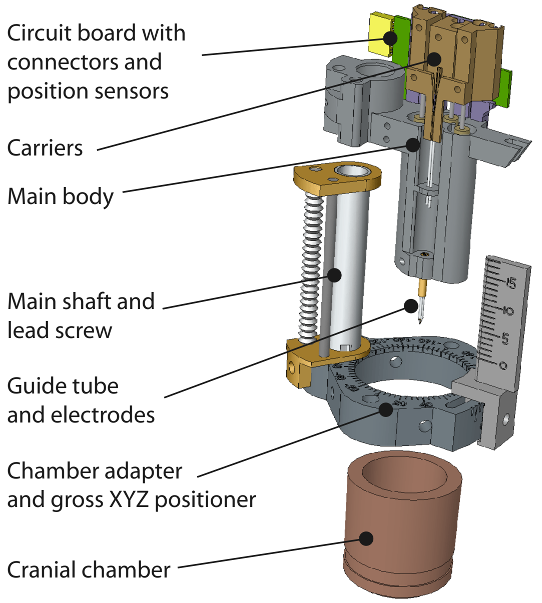
Circuit board with connectors and position sensors
Carriers
Main body
Main shaft and lead screw
Guide tube and electrodes
Chamber adapter and gross XYZ positioner
Cranial chamber

Fig. 2. Exploded view of electrode microdrive structure.

Figure 2 shows a schematic diagram of our microdrive mechanism, while Figure 1 shows photographs of the device. The mechanism's main body encases three piezo-electric linear actuators and contains electrode guide tubes and a circuit board. The actuators (Klocke Nanotechnik, Germany) provide both high precision (sub-micron steps) and long range of motion (about 5.6 millimeters) and do not suffer from gear backlash, which introduces significant imprecision in other drives. Hall-effect sensors built into a small mounted circuit board measure electrode depth to 1-micron precision.

Each linear actuator moves a *carrier*, to which the electrodes are attached both electrically and mechanically. The electrodes consist of platinum-iridium wires insulated with glass along their length, except at the recording tip and the back end (Alpha Omega Co., USA). The electrode signals are routed to the circuit board via flexible, polyimide-shielded copper strips, and then routed to a standard multi-pin connector that connects to a headstage amplifier. The electrodes are loaded tail-first through the guide tube and their corresponding carrier tubes, and fixing them under screw heads on the carriers

The main body assembly is held to a *chamber adapter* via a main shaft (see Figure 2). This design allows the electrodes to be positioned over any point within a 12mm diameter circular area inside the *recording chamber*[1]. To initiate a recording session, the microdrive is lowered manually by the vertical lead screw until the *guide tube* pierces the dura, which is a tough layer of tissue protecting the brain. The guide tube protects the fragile electrodes during this process. The use of the depth gauge on the side of the device aids the user in establishing the rough depth of the electrodes. The autonomous algorithm then positions the electrodes via the linear actuators. For semi-chronic use, structural elements are

[1] The recording chamber is surgically implanted into the experimental subject over the brain regions of interest. These chambers are standard in the neuroscience recording community and not part of our design.

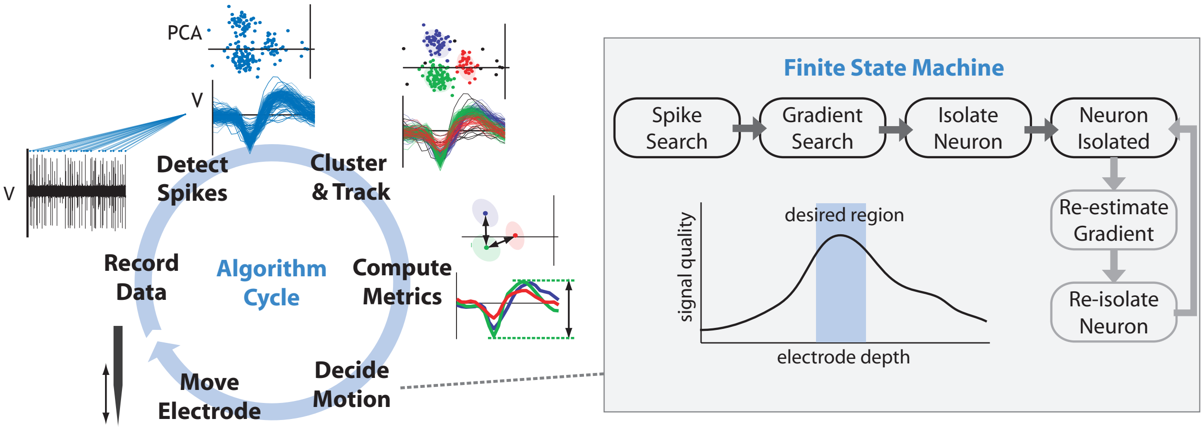

Fig. 3. Left: Control algorithm cycle. Right: Prototypical path of Finite State Machine used in electrode movement decisions. See text for details.

locked into place with set screws, and a cover can be placed over the entire assembly to protect against impact. Since there are no parts to disassemble between uses, cleaning is accomplished by a simple bath in a disinfectant solution.

Manufacture of the Drive. The majority of the microdrive's components were manufactured through a layered stereolithography (SLA) process. This approach minimizes the number of fasteners, which often work themselves loose or give way to leaks in the wet conditions of living tissue. Because the material used in the SLA process is not biocompatible, its parts were coated with a 20-micron thick film of Parylene, a conformal plastic known for its excellent biocompatibility properties. To account for the limited precision (0.1 mm) of the SLA process, Teflon bearing inserts are used in the sliding joints, and their position can be adjusted via small set-screws to achieve the joint precision and smoothness needed when advancing the guide tube. The final assembled device weighs 26.1g.

III. Structure of the Control Algorithm

This section provides a brief overview of the control algorithm's overall structure (details of specific algorithm components can be found in the references). The control system's main goal is to determine, based on the recorded signal and the electrode's position history, the best position — or "depth" — for each electrode. Because each electrode is moved independently, we focus on the processing steps for a single electrode. The control algorithm operates in a cycle (see Figure 3): 1) the electrode signal is acquired over an interval while the electrode is stationary; 2) this signal is analyzed (in several steps) to determine if and how the electrode should be repositioned; 3) the electrode is moved to a new position. The main challenge is to automate the elements of this cycle, which are typically carried out by an electrophysiologist, via unsupervised signal processing methods, while also respecting constraints on the recording process (such as not moving too fast to cause tissue damage, or skewering a neuron, hastening its death) that are normally handled by the electrophysiologist's experience and intuition.

The first unsupervised signal processing step, known as spike detection, identifies the action potential "events" in the electrode's voltage trace. We employ a wavelet-based method developed by Nenadic and Burdick [16] specifically designed for this application. The approximately 1.2-msec long spike waveforms are extracted from the voltage trace and aligned by their minima in preparation for the next steps.

The recorded signal from a single electrode may contain actional potential waveforms from several neurons near the electrode tip. Next, a *spike sorting* procedure is used to classify the recorded spikes into different clusters, each of which is attributed to a different neuron. A main contribution of this paper is a new algorithm for this task, along with the related data association and tracking algorithms that match current clusters with specific neurons that were identified during previous algorithm processing cycles (Section IV).

After these preprocessing steps, electrode movement decisions are based on two metrics. A *signal quality metric* (SQM), such as the signal to noise ratio (SNR) of the spike waveforms, is the algorithm's primary optimization. We only require that the SQM increases with proximity to a neuron. The *dominant neuron* has the highest SQM over a recent time period. Additionally, because a neuron's signal is only valuable if it can be distinguished from those of surrounding neurons, we also employ an *isolation quality metric* (IQM) that measures the separation of the dominant neuron's waveforms from those of other neurons.

In the idealized scenario where one stable neural signal source is ever present, the algorithm then determines the electrode motion to increase the SQM of the dominant neuron. A stochastic optimization procedure for this goal of maximizing an unknown function based on noisy observations is detailed in [9]. Since there may be confounding neurons as well, a minimum IQM threshold must also be met to for a given SQM optimization to be acceptable. A thorough discussion of choice and use of metrics is found in Branchaud [5].

To account for many additional challenges in practical recording, we use a finite state machine architecture, with individual states and state transition crafted to address these practical problems (see [5] for details). For example, Figure 3 shows a prototypical path of state transitions. In the Spike Search state, the electrode is moved relatively quickly through cortex to seek a region where neural activity is above a threshold (indicating that there may be isolatable neurons nearby). Once neural activity is detected, the Gradient Search state moves the electrode so as to find a region where a statistically significant positive-sloped SQM curve may be constructed. The Isolate Neuron state then invokes the stochastic optimization method [9] to both build the SQM curve and determine the electrode's motion towards the maximum of that curve. Within this state, another finite state machine monitors the possiblity that the electrode will move along a path that will puncture, and thus kill, a neuron. In this case, the electrode may be commanded to "back away" if deemed too close to the neuron. As mentioned earlier, electrode–tissue drift may degrade the signal over time. The states Re-estimate Gradient and Re-isolate Neuron adjust the electrode to maintain a strong isolation. Finally, state transitions account for other typical recording events, such as the temporary or permanent appearance or disappearance of neurons on the recorded signal.

IV. Bayesian Spike Clustering and Neuron Tracking

As noted earlier, the electrode tip may happen to be within the "listening sphere" of multiple neurons, causing the spike activity of several neurons to be recorded. Sorting these spikes according to their generating neuron is a challenging process, typically completed manually via analysis of the voltage waveforms. The waveforms from an example recording are shown in the top left of Figure 3, with their 2-dimensional principal component (PCA) representation above them, where each waveform is represented as a point. Although all waveforms are similar, enough separation in PCA space exists to distinguish three neurons.

Within each recording interval that initates one cycle of the algorithm (one time *step*), it is essential that the spikes that arise from different neuronal sources are classified correctly. Knowing which spikes belong to which neurons is critical for computing an accurate SQM and IQM, as well as for the eventual scientific or clinical use of the neural data. Moreover, to build a faithful SQM curve which is at the basis of our optimization based approach, we must track neurons over sequential recording intervals by matching current signal clusters with previous clusters.

The non-stationarity of spike waveforms due to electrode drift is a commonly cited culprit for difficulties in tracking neurons over time [17], [18], [19]. However, when the recording application involves repeated sampling and clustering over time, our experience has shown that the inconsistency of conventional clustering methods' output is a crucial issue. Due to the difficulties of unsupervised clustering, data sampled during consecutive time intervals will often be clustered much differently, changing each cluster's statistics enough to significantly reduce the reliability of matching clusters across consecutive time windows.

In our novel classification and clustering approach, we mitigate this effect by incorporating prior information into a Bayesian clustering algorithm. Our strategy's foundation is the optimization of the model parameters of a Gaussian mixture via expectation-maximization (EM) to best represent the probability distributions of the signal-generating neurons [20], [21]. Under the assumption that the preceding recording interval has a reasonable clustering result, we first use the preceding model's statistics to guess the initial values (or seed) for the EM algorithm, which greatly increases our chances of avoiding poor local optima while still mindful of phenomena commonly encountered in clustering neural data over time. Then we incorporate the preceding cluster locations as a prior during the EM algorithm, thus seeking maximum-*a-posteriori* (MAP) rather than maximum likelihood (ML) results for the purpose of improved cluster tracking throughout the recording session. Importantly, we implement the prior in a manner such that the method will likely succeed even if the preceding clustering was incorrect or if different neurons' signals are recorded during the two intervals. Not only does this procedure provide more consistent clustering results, but we will show that it provides neuron tracking (or data association across time) "for free," as it generates measures that quantify the probability that a given cluster in the current recording interval is associated with a signal cluster from a previous recording interval. A Bayesian technique for choosing the best mixture model order is embedded in our approach as well. We assume the clusters are nearly stationary over small time intervals.

Our clustering and tracking procedures are executed in a d-dimensional feature space of the extracted, aligned spike waveforms. Typically, we use a 2-dimensional PCA basis, which is also commonly used in manual spike classification.

A. Mixture Model Optimization via EM

While many traditional clustering procedures have been used to classify neural waveforms, the optimization of a (typically Gaussian) mixture model [20] has been shown to be an effective approach [22], [17], [23]. The underlying assumption is that the spike waveform samples in different clusters can be modeled as samples from different multivariate statistical distributions, where each distribution represents the signal features of a specific neuron.

Let us first review the classical mixture model and corresponding ML optimization. We consider the *mixture likelihood*, $\mathcal{L}_M$, of the model parameters given the data:

$$\begin{aligned}\mathcal{L}_M\big(\Theta_m \mid Y, \mathcal{M}_m\big) &= p\big(Y \mid \Theta_m, \mathcal{M}_m\big)\\ &= \prod_{n=1}^{N} \sum_{g=1}^{G_m} \pi_g f_g\big(y_n \mid \theta_g\big), \quad (1)\end{aligned}$$

where:

- Y is a set of N (spike feature) observations $y_n \in \mathbb{R}^d$.
- $\mathcal{M}_m$ is the m^{th} model class under consideration in the current recording interval, which dictates the model order G_m, the form of the g^{th} probability density f_g (typically Gaussian), and the model parameters $\Theta_m = \{\theta_g, \pi_g\}_{g=1}^{G}$.
- θ_g are the parameters of g^{th} mixture component (mean and covariance matrix for a Gaussian mixture, $\theta_g = \{\mu_g, \Sigma_g\}$).

- π_g is the mixture weight of component g, i.e., the probability that an observed spike belongs to component g (i.e. generated by source neuron g), with $\pi_g \geq 0$ and $\sum_{g=1}^{G_m} \pi_g = 1$.

The powerful EM algorithm [24] is typically applied to estimate the mixture parameters by log-likelihood maximization. At the same time, the EM procedure assigns data points to the appropriate mixture component, thereby effecting the separation of spikes. To apply this technique, we view our data Y as "incomplete" and augment it by Z, the set of membership variables $z_n = (z_{n1}, ..., z_{nG})$,

$$z_{ng} = \begin{cases} 1 & \text{if spike waveform } y_n \text{ belongs to cluster } g \\ 0 & \text{otherwise.} \end{cases}$$

Incorporating Z we can derive the corresponding *complete-data log-likelihood*

$$l_{CD}(\Theta_m \mid Y, Z, \mathcal{M}_m) = \sum_{n=1}^{N} \sum_{g=1}^{G_m} z_{ng} \log \left[\pi_g f_g\big(y_n \mid \theta_g\big) \right] . \tag{2}$$

The EM algorithm iterates between an E-step to calculate the conditional expectation $\hat{z}_{ng} = E[z_{ng} \mid y_n, \Theta] \in [0, 1]$ using the current parameter estimates, and an M-step to find the parameter estimates $\hat{\Theta}$ that maximize (2) given $\hat{z}_{ng}$, until some convergence threshold is reached. The algorithm requires an initial guess or "seed clusters" for initialization, and a new seeding method is discussed in more detail below.

B. Bayesian Framework for Model Selection and Parameter Estimation

To develop an appropriate spike waveform clustering method, let us now incorporate the sequential nature of our data sampling, include a model selection framework (to determine the most appropriate number of clusters, G_m), and employ MAP parameter estimation (determining $\hat{\Theta}$ and thus cluster membership Z). Denote Y^k the data at the k^{th} time step, and $Y^{1:k}$ all data from the first through k^{th} step of recording. We choose a set of model classes $\mathcal{M}_m, m = 1, ..., M$ for consideration, typically encoding a range of model orders (number of clusters to seek), but perhaps also different parsimonious models of covariance matrices Σ_g or different probability densities f_g. Then the problem consists of three related parts:

1) Find the MAP parameters and cluster membership for each model class: MAP parameters are naturally estimated using Bayes' Rule:

$$\begin{aligned} &p\big(\Theta_m \mid Y^{1:k}, \mathcal{M}_m\big) \\ &\qquad \propto p\big(Y^k \mid \Theta_m, \mathcal{M}_m\big) p\big(\Theta_m \mid Y^{1:k-1}, \mathcal{M}_m\big), \end{aligned} \tag{3}$$

where we have removed the unnecessary conditioning on $Y^{1:k-1}$ in the likelihood term and have suppressed the k-dependence of parameters — i.e., Θ_m for Θ_m^k — when it can be inferred to be at time k. Our EM solution (see Section IV-C) provides the cluster membership Z in the same way as discussed above.

2) Determine the model class $\mathcal{M}_m$: Again we will use a Bayesian approach to determine the probability of the model given the data:

$$\begin{aligned} P\big(\mathcal{M}_m \mid Y^{1:k}\big) &= P\big(\mathcal{M}_m \mid Y^k, Y^{1:k-1}\big) \\ &= \frac{p\big(Y^k \mid Y^{1:k-1}, \mathcal{M}_m\big) P\big(\mathcal{M}_m \mid Y^{1:k-1}\big)}{p\big(Y^k \mid Y^{1:k-1}\big)}. \end{aligned} \tag{4}$$

This probability is difficult to compute because the evidence term $p\big(Y^k \mid Y^{1:k-1}, \mathcal{M}_m\big)$ theoretically requires an integration over all possible parameters. However, by employing Laplace's asymptotic approximation [25], we can estimate a value while evaluating only at the MAP parameters $\hat{\Theta}_m$:

$$\begin{aligned} &p\big(Y^k \mid Y^{1:k-1}, \mathcal{M}_m\big) \approx p\big(Y^k \mid \hat{\Theta}_m, \mathcal{M}_m\big) \\ &\quad \cdot p\big(\hat{\Theta}_m \mid Y^{1:k-1}, \mathcal{M}_m\big) (2\pi)^{N_m/2} |\mathbf{H}_m(\hat{\Theta}_m)|^{-1/2} \end{aligned} \tag{5}$$

The first factor is the well-known likelihood of the Gaussian mixture, while the other terms are collectively known as the Ockham factor, as they penalize the complexity of the model parameterization. Other popular approaches to model selection, such as the Bayes Information Criterion (BIC), eliminate terms and are essentially approximations to Laplace's approximation. For our application, the Laplace approach naturally incorporates the prior on Θ_m, and provides greater accuracy that is especially useful for distinguishing models in noisy neural data. The main disadvantage is the need to calculate the Hessian matrix $\mathbf{H}_m(\hat{\Theta}_m)$.

We can see that the prior, $P\big(\mathcal{M}_m \mid Y^{1:k-1}\big)$, is simply the output of the previous clustering step, under the assumption that the model class is constant. To include some probability that the model can change (e.g. new neural signal sources appear, or existing signal sources disappear), we use a weighted mixture of the previous result with a uniform prior. The denominator in (4) is simply the sum of the numerator over all M models, normalizing the probability.

3) Match clusters from the current recording interval to clusters identified in the previous interval: To track neurons, the currently identified clusters are matched to those in the previous step(s), while also allowing for possible changes in the number of clusters. We will show below how the probability that each current cluster matches to either a cluster from the previous step or to a "new neuron" arises from our EM routine. The match with the highest probability is taken for each current cluster. If a neuron from the previous step is not matched, it is assumed to have disappeared from the signal. In the infrequent case that multiple current clusters match a common prior cluster, this prior cluster is taken to have split into two now-distinguishable neurons.

C. Extending EM for Cluster Location Priors

For use in (3), we will construct a prior based on our clustering results from the previous step, so that $p\big(\Theta_m \mid Y^{1:k-1}, \mathcal{M}_m\big) = p\big(\Theta_m \mid \hat{\Theta}_m^{k-1}, \mathcal{M}_m\big)$. Most important to the issue of cluster consistency and tracking is the location of each cluster center, μ_g. We look for the g^{th} mean μ_g to be near to *any* of the preceding step's cluster locations, without regard to which one, and use an equitable mixture of Gaussians representing all of the previous step's cluster means. We give uniform priors to all Σ_g and π_g, as they are not as important for tracking, and they drop out of most calculations as constants. Thus,

$$p\big(\Theta_m \mid \hat{\Theta}_m^{k-1}, \mathcal{M}_m\big) \propto \prod_{g=1}^{G_m} \sum_{j=1}^{\hat{G}^{k-1}} \frac{1}{\hat{G}^{k-1}} f_{\mathcal{N}}\big(\mu_g \mid \hat{\mu}_j^{k-1}, \Sigma_{\hat{\mu}_j}^{k-1}\big) \tag{6}$$

where $f_{\mathcal{N}}$ is the d-dimensional multivariate normal density, $\hat{G}^{k-1}$ is the number of clusters identified in the previous step, $\hat{\mu}_j^{k-1}$ is the estimated (MAP value) of the j^{th} cluster mean in the previous step, and $\Sigma_{\hat{\mu}_j}^{k-1}$ is the covariance associated with our estimation of the prior mean $\hat{\mu}_j^{k-1}$. Note that, if using data-dependent features such as PCA, we clearly must ensure the prior statistics and current data are in the same coordinate frame — we convert the prior step's full-dimension spike data to the current PCA space, then calculate the prior clusters' statistics in this space.

Note that the prior (6) bears distinct resemblence to the mixture likelihood (1) and, when incorporated into (3), will in fact share the same difficulty of maximization. We can apply the same solution: add hidden variables and optimize via EM. Let the new "membership" variables $\mathcal{Z} = \{\zeta_{gj}\}$ indicate whether the prior of previous cluster j should influence the current cluster g, or, ideally,

$$\zeta_{gj} = \begin{cases} 1 & \text{if } \mu_g \text{ and } \hat{\mu}_j^{k-1} \text{ represent the same neuron} \\ 0 & \text{otherwise.} \end{cases}$$

Using this approach, we can derive the complete-data prior log density on the means:

$$\begin{aligned} &\log p\big(\mu, \mathcal{Z} \mid \Theta^{k-1}\big) \\ &= \sum_{g=1}^{G_m} \sum_{j=1}^{\hat{G}^{k-1}} \zeta_{kj} \log \left[\frac{1}{\hat{G}^{k-1}} f_{\mathcal{N}}\big(\mu_g \mid \hat{\mu}_j^{k-1}, \Sigma_{\hat{\mu}_j}^{k-1}\big) \right]. \end{aligned} \tag{7}$$

Returning to (3) and expanding to include the hidden variables (with $\Theta^{k-1} = \{\hat{\mu}_j^{k-1}, \Sigma_{\hat{\mu}_j}^{k-1}\}_{j=1}^{\hat{G}^{k-1}}$), we have

$$\begin{aligned} p\big(\Theta_m, \mathcal{Z} \mid Y^{1:k}, Z, \mathcal{M}_m\big) &\propto p\big(Y^k, Z \mid \Theta_m, \mathcal{Z}, \mathcal{M}_m\big) \\ &\quad \cdot p\big(\Theta_m, \mathcal{Z} \mid Y^{1:k-1}, \mathcal{M}_m\big). \end{aligned} \tag{8}$$

To maximize the log posterior, take the log of (8) and substitute in (2) and (7),

$$\begin{aligned} &\log p\big(\Theta_m, \mathcal{Z} \mid Y^{1:k}, Z, \mathcal{M}_m\big) \\ &= \sum_{n=1}^{N} \sum_{g=1}^{G_m} z_{ng} \log \left[\pi_g f_{\mathcal{N}}\big(y_n \mid \mu_g, \Sigma_g\big)\right] \\ &\quad + \sum_{g=1}^{G_m} \sum_{j=1}^{\hat{G}^{k-1}} \zeta_{kj} \log \left[\frac{1}{\hat{G}^{k-1}} f_{\mathcal{N}}\big(\mu_g \mid \hat{\mu}_j^{k-1}, \Sigma_{\hat{\mu}_j}^{k-1}\big) \right] + \mathcal{C}. \end{aligned} \tag{9}$$

This complete-data posterior (9) is the object equation of the EM iterations, which are modified from the ML (see [20]) iterations as follows.

1) E-Step: As before, we find spike-to-cluster memberships $\hat{z}_{ng}$ given parameter estimates from the M-step. Now, additionally calculate the expectation of the other hidden data, current-to-prior cluster membership $\hat{\zeta}_{gj} = E\left[\zeta_{gj} \mid Y^{1:k}, \Theta_m\right]$

$$\hat{\zeta}_{gj} = \frac{f_{\mathcal{N}}\big(\mu_g \mid \hat{\mu}_j^{k-1}, \Sigma_{\hat{\mu}_j}^{k-1}\big)}{\sum_{l=1}^{\hat{G}^{k-1}} f_{\mathcal{N}}\big(\mu_g \mid \hat{\mu}_l^{k-1}, \Sigma_{\hat{\mu}_l}^{k-1}\big)}. \tag{10}$$

2) M-Step: As our prior term in (9) is independent of the parameters π_g and Σ_g, these estimates remain the same as the ML version[2]. Maximizing (9) with respect to μ_g, however, gives

$$\begin{aligned} \hat{\mu}_g = &\left[\sum_{n=1}^{N} \hat{z}_{ng} \hat{\Sigma}_g^{-1} + \sum_{j=1}^{\hat{G}^{k-1}} \hat{\zeta}_{gj} \left(\Sigma_{\hat{\mu}_j}^{k-1}\right)^{-1} \right]^{-1} \cdot \\ &\left[\sum_{n=1}^{N} \hat{z}_{ng} \hat{\Sigma}_g^{-1} y_n + \sum_{j=1}^{\hat{G}^{k-1}} \hat{\zeta}_{gj} \left(\Sigma_{\hat{\mu}_j}^{k-1}\right)^{-1} \hat{\mu}_j^{k-1} \right], \end{aligned} \tag{11}$$

in contrast to the ML estimation,

$$\hat{\mu}_g = \frac{\sum_{n=1}^{N} \hat{z}_{ng} y_n}{\sum_{n=1}^{N} \hat{z}_{ng}}.$$

Note that (11) has the form of a weighted average of the data points y_n with (fuzzy) membership to cluster g and the prior means $\hat{\mu}_j^{k-1}$ (fuzzily) affiliated to cluster g, with the weights governed by the respective covariance matrices.

D. Tracking Clusters Across Time Steps

Ultimately our goal is to "track neurons" — that is, to associate specific neurons with specific signals over time by matching the current clusters with clusters of the preceding step (or possibly identify them as newly appearing or disappearing neurons). If we consider this a data association task on the locations (i.e., means) of the clusters over consecutive time steps, we can match each current cluster g to its best prior cluster via $j^* = \arg\max_j \hat{\zeta}_{gj}$.

However, one biophysical possibility must be accounted for. The quantity $\hat{\zeta}_{gj}$ encodes the probability that current cluster g was generated from the same distribution ("neuron") as prior cluster j *relative* to all $\hat{G}^{k-1}$ prior clusters, implicitly assuming the same neurons are in approximately the same locations. A high value $\hat{\zeta}_{gj}$ only indicates the closest of the prior clusters (in the Mahalanobis sense), but cluster g could be a newly appearing neuron. We add, then, a uniform "background" distribution as a mixture component in our prior (6). This effectively places a minimum threshold on the Mahalanobis distance allowed to match g to j, otherwise g is considered a new neuron. Note that disappearing neurons will simply not have a match among current clusters and thus marked by that evidence.

Additionally, a mathematical possibility exists for two current clusters to be matched to the same j^*. A single-match nearest neighbor approach could be implemented to

[2] Parsimonious estimations of Σ_g are also possible [26]; an equal-volume assumption among clusters may be preferred.

avoid this. However, the two current neurons' spikes may have been indistinguishable on the prior step, and thus we choose to track this as a split of the prior "neuron."

E. Generating Seed Clusters

The EM algorithm is highly susceptible to local optima near its initial values, or seed clustering. To start with the best possible seeds, we again leverage our preceding step's result. The basic approach to using this "prior" information in generating the seed is to assign the current data points to whichever prior cluster is closest, using the Mahalanobis distance. Recall, however, that we need a seed clustering for each of a *range* of model orders (numbers of clusters), say $G_m = 1, ..., G_{max}$.

The primary complications arise in cases G_m is different from the number of prior clusters $\hat{G}^{k-1}$. For the case $G_m < \hat{G}^{k-1}$, we choose prior cluster(s) to "eliminate" and assign points to the remaining ones. The case $G_m > \hat{G}^{k-1}$ is trickier, as we must implement a "pre-clustering" routine simply for the seeds. We have obtained good results from using k-means with carefully chosen starting centroids. We refer the reader to Wolf and Burdick [27] for more detail.

The plots in Figure 4 demonstrate the algorithm's performance in simultaneously positioning three electrodes for acceptable isolations. This data set was recorded from macaque parietal cortex in an acute recording session with platinum-iridium, 1.5 MΩ-impedence electrodes. For each electrode channel, the isolation quality metric (IQM), electrode depth, and algorithm mode are shown. Channel A represents a nearly ideal case wherein a neuron is said to be isolated throughout the recording. Channels B and C show more typical cases: the algorithm first seeks neuronal signals, builds the SQM curve, isolates a neuron, and then itermittently readjusts electrode position to re-isolate a neural signal.

Figure 5 highlights the algorithm's clustering and tracking performance, while also showing spike waveforms collected from the robotic microdrive. This figure displays electrode position over a 9 minute interval with associated samples of recorded spike waveforms at different sampling steps. This sequence emphasizes the importance of consistent spike clustering to effective neuron tracking, contrasting the maximum likelihood (ML) clustering results below our results. Importantly, all thirteen steps during this time period have the "same" three clusters identified and tracked by our proposed algorithm. Each step contains 20 seconds of data with separating intervals of approximately 20 seconds.

V. Experimental Results

The ML "baseline" algorithm used for this figure follows the proposal of [28], using EM with ML parameters, seeded with clusters from a standard hierarchical agglomerative technique. Model order is selected according to the Bayesian information criterion (BIC), an approximation to the Bayes factor that includes a penalty for model complexity: $BIC \equiv 2l_M(\hat{\Theta} \mid Y, \mathcal{M}) - p_{\mathcal{M}} \log N$ for model $\mathcal{M}$, maximized mixture log-likelihood l_M, and number of independent model parameters $p_{\mathcal{M}}$. We have used this method extensively for the past two years in hundreds of recording sessions, originally chosen based on its high rate of success compared to other spike sorting options. In both algorithms, we implemented a "background" mixture component of uniform distribution to capture outliers, which appear as black waveforms/points. Overall, notice our algorithm consistently identifies three clusters in approximately the same PCA position, whereas the ML method gives statistically sound but somewhat incongruous results, seemingly more volatile to noise variations. Moreover, the consistency of the cluster colors in each step signifies our algorithm's tracking of the neurons through the sequence of samples. No such attempt is made for the baseline case and thus the colors are essentially random.

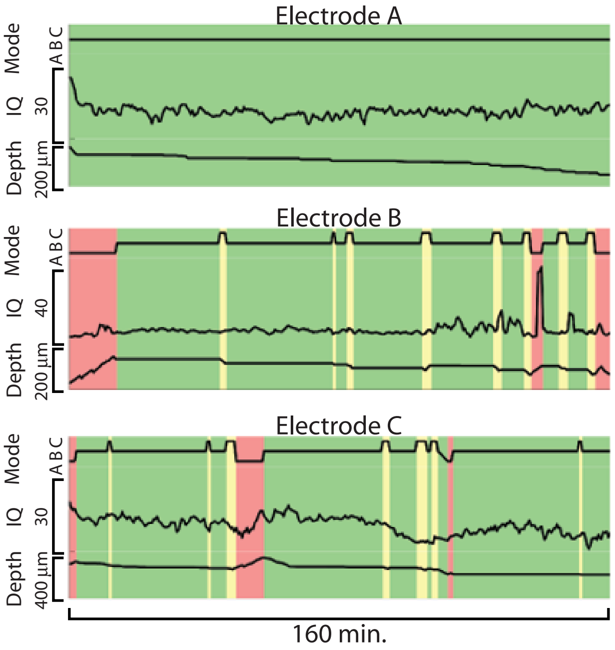

Fig. 4. Algorithm mode, isolation quality, and electrode depth over time on three simultaneously-operated electrodes during an acute recording session. Both Mode label and background color indicate the algorithm state: (A) [red] Isolate, includes search states and gradient climbing; (B) [green] Neuron Isolated; (C) [yellow] Re-isolate Neuron, includes gradient re-estimation.

VI. Conclusion

We have presented control algorithm and a miniature robot capable of automating the process of isolating and maintaining high quality extracellular neural recordings for hours at a time. The algorithm's finite state machine architecture enables it to cope with many unstructured challenges existent in a realistic recording environment. The new sequential clustering and classification algorithm presented in this paper gives significantly more consistent classification results, and enables us to track individual neurons over consecutive recording intervals. We hope that these advances provide further steps along the path to an implantable device with many independently movable electrodes, each capable of tracking individual neurons indefinitely.

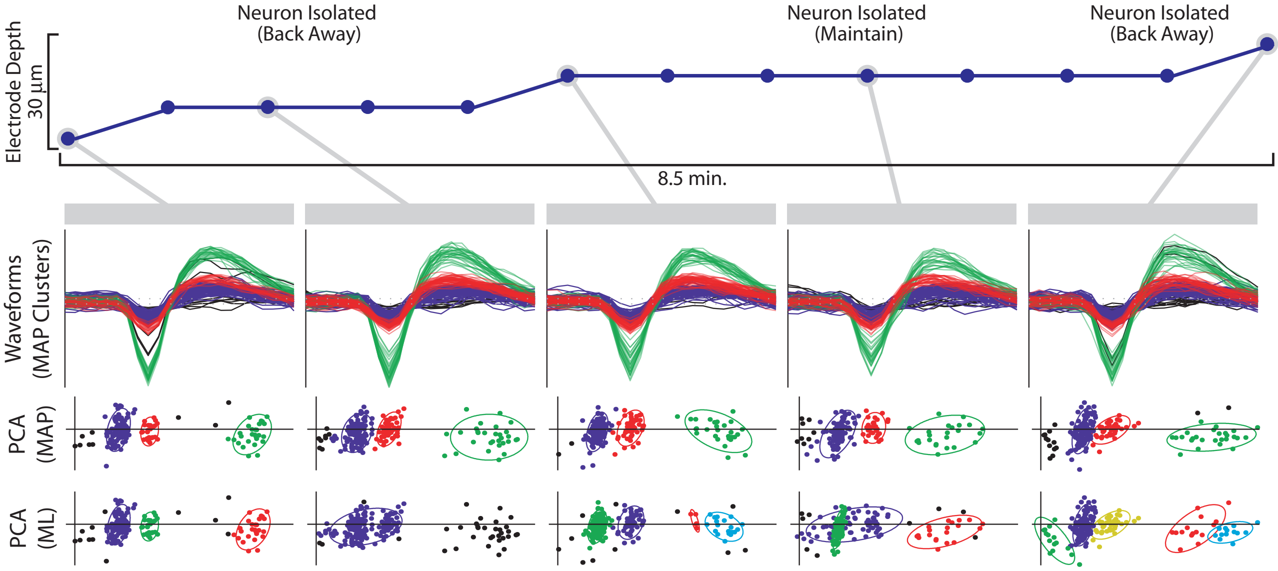

Fig. 5. Electrode depth and sample waveforms with cluster results over 8.5 minutes of an isolation three times. The electrode backs away to avoid damaging the neuron. Colors of waveforms and top PCA row indicate our (MAP) clusters, showing three consistently tracked neurons — all were tracked over the entire time period above, including steps not shown. For comparison, ML cluster results show sometimes erratic results.

REFERENCES

[1] R. A. Andersen, J. W. Burdick, S. Musallam, B. Pesaran, and J. G. Cham, "Cognitive neural prosthetics," *Trends in Cognitive Sciences*, vol. 8, no. 11, pp. 486–493, November 2004.

[2] D. M. Taylor, S. I. H. Tillery, and A. B. Schwartz, "Direct cortical control of 3d neuroprosthetic devices," *Science*, vol. 296, no. 5574, pp. 1829–1832, 2002.

[3] M. D. Serruya, N. G. Hatsopoulos, L. Paninski, M. R. Fellows, and J. P. Donoghue, "Brain-machine interface: Instant neural control of a movement signal," *Nature*, vol. 416, no. 6877, pp. 141–142, 2002.

[4] J. Wessberg, C. R. Stambaugh, J. D. Kralik, P. D. Beck, M. Laubach, J. K. Chapin, J. Kim, S. J. Biggs, M. A. Srinivasan, and M. A. L. Nicolelis, "Real-time prediction of hand trajectory by ensembles of cortical neurons in primates," *Nature*, vol. 408, no. 6810, pp. 361–365, 2000.

[5] E. A. Branchaud, "An algorithm for the autonomous isolation of neurons in extracellular recordings," Ph.D. dissertation, California Institute of Technology, 2006.

[6] E. A. Branchaud, J. G. Cham, Z. Nenadic, and J. W. Burdick, "A miniature robot for autonomous single neuron recordings," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2005.

[7] J. G. Cham, E. A. Branchaud, Z. Nenadic, B. Greger, R. A. Andersen, and J. W. Burdick, "Semi-chronic motorized microdrive and control algorithm for autonomously isolating and maintaining optimal extracellular action potentials," *Journal of Neurophysiology*, vol. 93, pp. 570–579, January 2005.

[8] E. A. Branchaud, R. A. Andersen, and J. W. Burdick, "An algorithm for autonomous isolation of neurons in extracellular recordings," in *IEEE/RAS-EMBS Intl. Conf. on Biomedical Robotics and Biomechatronics (BioRob)*, 2006.

[9] Z. Nenadic and J. W. Burdick, "A control algorithm for autonomous optimization of extracellular recordings," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 5, pp. 941–955, May 2006.

[10] C. Pang, Y. C. Tai, J. W. Burdick, and R. A. Andersen, "Electrolysis-based diaphragm actuators," *Nanotechnology*, vol. 17, no. 4, pp. S64–S68, 2006.

[11] J. Muthuswamy, M. Okandan, A. Gilletti, M. S. Baker, and T. Jain, "An array of microactuated microelectrodes for monitoring single-neuronal activity in rodents," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 8, pp. 1470–1477, Aug 2005.

[12] R. C. deCharms, D. T. Blake, and M. M. Merzenich, "A multielectrode implant device for the cerebral cortex," *Journal of Neuroscience Methods*, vol. 93, no. 1, pp. 27–35, Oct. 1999.

[13] S. N. Baker, N. Philbin, R. Spinks, E. M. Pinches, D. M. Wolpert, D. G. MacManus, Q. Pauluis, and R. N. Lemon, "Multiple single unit recording in the cortex of monkeys using independently moveable microelectrodes," *Journal of Neuroscience Methods*, vol. 94, no. 1, pp. 5–17, Dec. 1999.

[14] J. D. Kralik, D. F. Dimitrov, D. J. Krupa, D. B. Katz, D. Cohen, and M. A. L. Nicolelis, "Techniques for chronic, multisite neuronal ensemble recordings in behaving animals," *Methods*, vol. 25, no. 2, pp. 121–150, Oct. 2001.

[15] M. S. Fee and A. Leonardo, "Miniature motorized microdrive and commutator system for chronic neural recording in small animals," *J Neuroscience Methods*, vol. 112, no. 2, pp. 83–94, Dec. 2001.

[16] Z. Nenadic and J. W. Burdick, "Spike detection using the continuous wavelet transform," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 1, pp. 74–87, January 2005.

[17] A. Bar-Hillel, A. Spiro, and E. Stark, "Spike sorting: Bayesian clustering of non-stationary data," *J Neuroscience Methods*, vol. 157, no. 2, pp. 303–316, Oct. 2006.

[18] R. K. Snider and A. B. Bonds, "Classification of non-stationary neural signals," *J Neuroscience Methods*, vol. 84, no. 1-2, pp. 155–166, Oct. 1998.

[19] A. Emondi, S. Rebrik, A. Kurgansky, and K. Miller, "Tracking neurons recorded from tetrodes across time," *J Neuroscience Methods*, vol. 135, pp. 95–105, 2004.

[20] G. McLachlan and D. Peel, *Finite Mixture Models*. Wiley Interscience, 2000.

[21] P. Cheeseman and J. Stutz, "Bayesian classification (AutoClass): Theory and results," in *Advances in Knowledge Discovery and Data Mining*, U. Fayyard, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds. Cambridge, MA: AAAI/MIT Press, 1996, ch. 6.

[22] M. S. Lewicki, "A review of methods for spike sorting: the detection and classification of neural action potentials," *Network: Computation in Neural Systems*, vol. 9, pp. R53–R78, 1998.

[23] S. Shoham, M. R. Fellows, and R. A. Normann, "Robust, automatic spike sorting using mixtures of multivariate t-distributions," *Journal of Neuroscience Methods*, vol. 127, no. 2, pp. 111–122, Aug. 2003.

[24] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[25] J. L. Beck and K.-V. Yuen, "Model selection using response measurements: Bayesian probabilistic approach," *J Engng Mech*, vol. 130, no. 2, pp. 192–203, 2004.

[26] G. Celeux and G. Govaert, "Gaussian pasimonious clustering models," *Pattern Recognition*, vol. 28, pp. 781–793, 1995.

[27] M. T. Wolf and J. W. Burdick, "Spike clustering and neuron tracking over successive time windows," in *IEEE EMBS Conf. on Neural Engineering (NER)*, 2007.

[28] C. Fraley and A. E. Raftery, "How many clusters? Which clustering method? Answers via model-based cluster analysis," *Computer Journal*, 1998.