

A Control Algorithm for Autonomous Optimization of Extracellular Recordings

Zoran Nenadic*, *Member, IEEE*, and Joel W. Burdick

Abstract—This paper develops a control algorithm that can autonomously position an electrode so as to find and then maintain an optimal extracellular recording position. The algorithm was developed and tested in a two-neuron computational model representative of the cells found in cerebral cortex. The algorithm is based on a stochastic optimization of a suitably defined signal quality metric and is shown capable of finding the optimal recording position along representative sampling directions, as well as maintaining the optimal signal quality in the face of modeled tissue movements. The application of the algorithm to acute neurophysiological recording experiments and its potential implications to chronic recording electrode arrays are discussed.

Index Terms—Optimization of recordings, autonomous electrode control, unsupervised signal processing, stochastic optimization.

I. INTRODUCTION

RECORDING the spiking activity of a single neuron or a population of neurons has become a primary method for understanding the physiological function of the brain. This paper focuses on extracellular recordings which are predominantly used in experiments with behaving animals. We present an algorithm that can autonomously position an extracellular recording electrode, so as to first optimize, and then maintain the quality of its signal. The algorithm can be potentially useful for both acute and chronic extracellular recordings. This paper focuses mainly on the theoretical aspects and the development of the algorithm in a simulated environment. The successful implementation of our algorithm in a multitude of acute recording experiments has been reported in [1].

A main motivation for the development of this algorithm stems from the tediousness and complexity of *acute recording experiments*, where electrodes are introduced in neural tissue for a period of several hours. During these experiments, operators advance high-impedance electrodes into neural tissue and *manually* position them with micron accuracy with the ultimate goal of a *single unit isolation*. A satisfactory isolation is usually achieved by placing the electrode in a close proximity

of the target cell, which in turn makes its activity dominant with respect to the activity of other cells and background noise. In the simplest acute recording scenario the operator advances a single microelectrode to a target depth, where neural signals are sampled and intuitively monitored by oscilloscopes and/or speakers. These devices help the experimenter assess the quality of observed data. Typically these signals contain no distinct neural activity, in which case the electrode is lowered further, often in somewhat larger increments, and the above procedure is repeated. As the electrode approaches active cells, distinct action potentials (spikes) emerge from the background noise. Very often these emerging spikes originate from multiple cells, making the process of cell isolation more challenging. The amplitude of these spikes is very sensitive to the distance between the active cell and the tip of the recording electrode (both theoretical [2] and experimental [3] studies point to an exponential type of dependency), therefore, once the spikes are detected, the operator proceeds cautiously (with a smaller step size). From this point on, the electrode guidance depends largely on the operators' intuition and their (subjective) assessment of the quality of recorded signals. Once the isolation is attained, frequent readjustments of the electrode's position are often necessary to actively maintain a high quality signal in the face of inherent tissue drift and relaxation. As the number of recording electrodes increases, the traditional paradigm of manual electrode positioning and frequent readjustments becomes tedious at best, and perhaps unworkable for experiments that use large electrode arrays [4]. An automated procedure to isolate cells, and then maintain the recording quality on a large numbers of electrodes, would lead to improved experimental productivity and higher quality data. These observations serve as a main motivation for the development of the algorithm which will be briefly described next.

To achieve single unit isolation and maintenance, an autonomous control system must replicate the essential steps performed by a human operator. First, such a system would need to recognize whether the recorded signal contains any distinct spikes or not. Clearly, such a recognition has to be performed in a completely unsupervised fashion, as opposed to current practice. If spikes are not found, the algorithm must advance the electrode with appropriate sampling step until the spikes are found. Once spikes are detected, the quality of observed signals needs to be estimated. Unlike the human operator, who often relies on a subjective estimate of the signal quality, we define a suitable *signal quality metric* and proceed by maximizing such a metric. Apart from spike detection, estimating such a metric from noisy data includes several additional steps, of which spike classification is the most important

Manuscript received April 19, 2004; revised October 2, 2005. This work was supported in part by the National Science Foundation (NSF) under Grant 9402726 and in part by the Defense Advanced Research Projects Agency (DARPA) under Grant MDA972-00-1-0029. Asterisk indicates corresponding author.

*Z. Nenadic was with the Division of Engineering and Applied Science, California Institute of Technology, Pasadena, CA 91125 USA. He is now with the Department of Biomedical Engineering, University of California, Irvine, CA 92697 USA (e-mail: znenadic@uci.edu).

J. W. Burdick is with the Division of Engineering and Applied Science, California Institute of Technology, Pasadena, CA 91125 USA.

Digital Object Identifier 10.1109/TBME.2005.863930

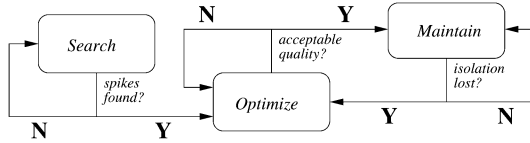


Fig. 1. FSM diagram of the algorithm.

one. Whereas humans simply look at an oscilloscope trace to establish the presence of multiple cells, the task is more challenging when performed by a computer in an unsupervised manner. Not only must the spikes be correctly classified, but also the number of distinct spike classes must be determined from the data. As the electrode moves through the tissue, spikes from new cells may appear and the signals from existing neurons may disappear, making the number of classes variable. Upon convergence, the electrode position should be (locally) optimal with respect to the chosen signal quality metric. Once the optimal position is found, the signal quality must be monitored, so that if it degrades significantly, the electrode position can be updated to re-optimize the signal.

Since the dynamics of the above system is driven by abrupt occurrence of physical events, the system is appropriately modeled as a discrete event system [5] with a finite state machine (FSM) (see Fig. 1). The states are represented by boxes and the state transitions are denoted by arrows. In each state, the algorithm samples observed signals for a short period of time and performs an analysis of the data (e.g., spike detection, classification, and signal quality calculation). Based on the outcome of the signal analysis, the algorithm commands the electrode's movement and a state transition is executed. In the *Search* state the electrode is moved with a constant step Δ_s , chosen by the user before the experiment initiates. In the *Optimize* state the electrode is moved according to a *recursive stochastic optimization*, that is developed in Section IV. In the *Maintain* state the electrode is not moved. The algorithm remains in the *Search* state (by means of self-transitions) until the spikes are detected, which triggers the transition to the *Optimize* state. The algorithm remains in the *Optimize* state until acceptable signal quality is attained (i.e., until the maximum of the signal quality metric is found), at which point the algorithm transitions to the *Maintain* state. At this state, the algorithm checks for the signal quality variations without the electrode movement. If these variations exceed some prespecified tolerance, the optimality is considered lost, and the algorithm transitions back to the *Optimize* state.

In the remainder of the paper, we present the details of the algorithm just described. Section II summarizes a computational model for the extracellular field around an active neuron, which serves as a front end for our control algorithm. In Section III, we introduce a set of unsupervised signal processing tools necessary for the autonomous operation of the algorithm. In Section IV, we formulate the problem of optimal electrode positioning as a maximization of a regression function. We quantify the algorithm's performance in a simulated environment in Section V, followed by a discussion in Section VI. Finally, the concluding remarks are given in Section VII and some mathematical details are given in the Appendix.

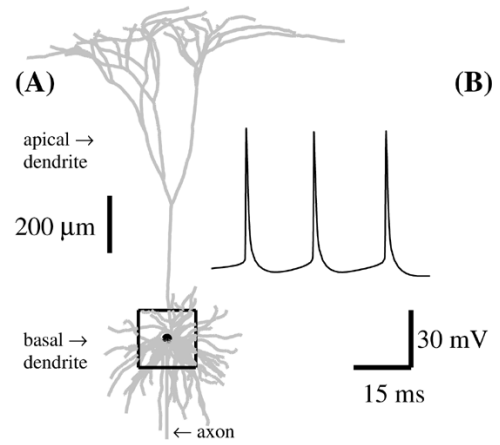


Fig. 2. (A) Two-dimensional projection of the pyramidal cell model. The rectangle in the lower left represents a $200 \times 200 \mu\text{m}$ boundary around the soma (visible as black dot in the center). (B) Intracellular voltage trace at the soma showing three action potentials.

II. MODELING EXTRACELLULAR POTENTIAL

The algorithm was initially developed and validated in a simulated environment. The purpose of the computational model is to simulate the recording electrode positioning processes in a repeatable and reliable way. Additionally, this model provides a biophysical basis for our control methodology. The simulator consists of two components. In the first part we solve for the time-varying membrane currents using a detailed compartmental model of a neuron. In the second part these currents are used as boundary conditions for a partial differential equation (PDE) that models the propagation of the extra-cellular potential.

A. Computational Model of a Single Neuron

A model of a neocortical pyramidal cell from layer 5 of the cat visual cortex was used for simulations [6]. This model was selected mainly because of its ability to emulate firing properties of real cortical cells. We emphasize that nothing about our control algorithm depends upon the specific properties of this model. A brief account of the model is given here for the convenience (see [6] for details).

The model consists of 3720 compartments, thus capturing the complex cell geometry (Fig. 2), and was developed with the NEURON simulator [7]. The model has low- (high-) density sodium Na^+ channels in the soma and dendrites (the axon hillock and initial segment), respectively. Fast potassium K^+ channels are present in the axon and soma, but are excluded from the dendrites. This type of channel distribution is responsible for spike initiation at the axon initial segment [8]. To ensure repetitive firing, slow K^+ channels were added to the soma and dendrites, along with one type of high-threshold calcium Ca^{2+} channel. Based on a modification by Holt [9], the neuron was activated by synapses uniformly distributed throughout the dendrites.

Generally, the propagation of electric potentials through conductive neural media gives rise to complex PDEs. Depending on the number of assumptions made in the modeling process, these equations can be substantially simplified. Only final results will

be presented here. For a detailed derivation of the results from the first principles and underlying assumptions, we refer to [10]. In summary, the electric potential in the space in and around the neuron is governed by a system of Laplace¹ equations

$$\nabla^2 \phi_i = 0 \quad \nabla^2 \phi_e = 0 \quad (1)$$

with the boundary conditions $\sigma_i \nabla \phi_i \mathbf{n}_i = J_m$ and $\sigma_e \nabla \phi_e \mathbf{n}_e = J_m$, where ϕ_i and ϕ_e are the intracellular and extracellular potential (respectively), σ_i and σ_e are the corresponding conductivities per unit length, \mathbf{n}_i and \mathbf{n}_e are normal vectors to the cell membrane ($\mathbf{n}_i = -\mathbf{n}_e$), and J_m is the transmembrane current per unit area. For a neuron with complex dendritic structure, the system (1) is virtually unsolvable. Most neural simulators assume one-dimensional (1-D) intracellular space and a weak coupling from extracellular to intracellular potential (effectively $\phi_e = 0$). Under these conditions, the ϕ_i -part of the system (1) can be viewed as a 1-D PDE driven by the *dynamics* of the active membrane. This PDE is then solved numerically by converting it to a system of ordinary differential equations through a compartmental modeling process. A train of intracellular action potentials at the soma compartment of the model cell is shown in Fig. 2. Once ϕ_i and J_m are known for each compartment, it is possible to solve for ϕ_e , although the solution is not straightforward as we shall see next.

B. Extracellular Potential Simulator

Despite existing numerical routines for solving PDEs, the complexity of the boundary condition renders the solution of the ϕ_e -part of (1) prohibitively expensive. An approximation has been developed that gives a fast and relatively accurate solution [9]. Assuming current is generated at a *point source* in an unbounded isotropic volume, we have

$$\phi_e(r, t) = I(t) / (4\pi\sigma_e r) \quad (2)$$

where $I(t)$ is the source current and r is the distance to the source. If we neglect the thickness of the compartmental model segments, each segment can be treated as a continuous line of point sources. For a single line segment, the potential at a point (r, z) due to this *line source approximation* is

$$\phi_e(r, z, t) = \int_{-s_0}^0 \frac{i_m(\zeta, t) d\zeta}{4\pi\sigma_e \sqrt{r^2 + (z - \zeta)^2}} \quad (3)$$

where i_m is the transmembrane current per unit length, r is the distance to the line, z measures distance in the direction of the line underlying the segment, and $-s_0$ and 0 are the endpoints of the line segment in a local coordinate system attached to the segment (see Fig. 3). Note that this approximation implies a radial symmetry of the extracellular potential ϕ_e . Moreover, for a fixed time t , the transmembrane current is constant along the segment ($i_m(\zeta, t) = i_m(t)$) and the solution to (3) can be found

¹The time constants for the extracellular and intracellular space are much smaller than those of an active membrane [10]. Hence, both the extracellular and intracellular space are almost purely resistive and no time derivatives appear in (1).

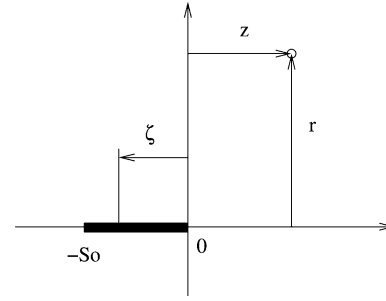


Fig. 3. Geometry of the line source approximation for a single segment with length s_0 . The evaluation point is located at (r, z) .

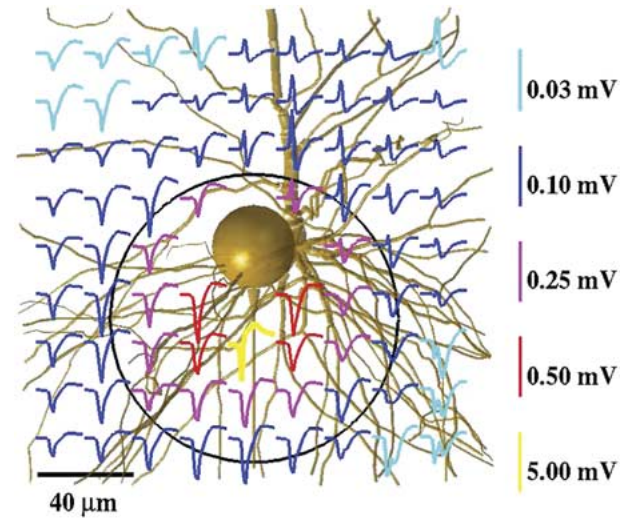


Fig. 4. Spatio-temporal variations of the extracellular potential in a horizontal plane confined to rectangular region shown in Fig. 2 and passing through the soma. Each trace represents a 2 ms interval of calculated potential at a point located in the center of the trace. The colored bars to the right are indicating the scales for different traces. The peak potential (yellow trace) occurs next to the axon hillock/initial segment. The circle corresponds to a 120- μ m-diameter sphere centered at the axon hillock. For typical recording noise levels, the spikes outside the sphere would not be detectable.

in a closed form. The exact form of this solution and the accuracy of the line source approximation for this particular model are discussed in [10]. At a fixed time, the potential at any point in the extracellular space is found by summing the contributions of all the line segments (3) and the soma (2).

The spatio-temporal variations of the simulated extracellular potential in a horizontal plane, passing through the soma of the model cell undergoing an action potential, are shown in Fig. 4. Note the variability in the shape of spikes at different spatial locations. Also, note that the amplitude of individual spikes drops exponentially with the distance from the point of the largest potential (axon hillock/initial segment for this model). We will see later that our electrode control methodology is facilitated by these large amplitude gradients. Consistent with both empirical [11] and theoretical [2] studies for small cortical neurons (soma diameter 10–30 μ m), the average value of the peak-to-peak signal amplitude 60 μ m away from the axon hillock is less than 100 μ V (see Fig. 4). For many recording systems this is close to the level of noise imposed by various factors (see below), which makes these signals indiscernible from the noise.

C. Modeling Noise Field

It is necessary to add noise to the model in order to mimic experimental conditions. The major sources of noise in recording experiments are: the thermal noise in the recording electrode, the ambient noise of the recording hardware, the quantization noise due to analog-to-digital (A/D) conversion, and the *biological noise* [11], i.e., the spikes of neurons relatively distant to the recording site.

In the simplest scenario, the observed signal z can be viewed as a useful *deterministic* signal, s , corrupted by an additive noise w , i.e., $z(t) = s(t) + w(t)$, where w subsumes various noise sources from above. To model noise w properly, we make the following observations. First, neural noise is nonstationary, i.e., its statistical properties are changing over time. The nonstationarity of neural noise has been reported on time scales as small as 20 ms [12]. Secondly, the distribution of neural noise is non-Gaussian. Several studies [12]–[14] report super-Gaussian density of neural noise. Furthermore, like many processes arising in physical applications, neural noise belongs to the category of $1/f$ processes [15], with a spectrum² $S(f) \propto |f|^{-p}$ ($p \in \mathbb{R}, p > 0$) over a wide range of frequencies. Since these processes are not realizable with linear models, other modeling attempts have been made in the past [16]–[18], although no general solution exists to date.

To avoid these problems and to take advantage of recorded data³ at our disposal, we sampled neural noise from the Rhesus parietal cortex recordings that did not yield any visible spikes [Fig. 5(A)]. This noise, which we refer to as *recorded noise*, contains all the noise sources listed above and has a characteristic $1/f$ spectrum [Fig. 5(B)]. The nonstationarity of the noise is shown as follows: some 100 s of recorded noise were broken into 50-ms nonoverlapping subsegments, and the subsegments were whitened using the eigenvectors of their common covariance matrix. Note that the stationarity is invariant under linear transformations, allowing us to test the whitened subsegments. Given two whitened subsegments, the null hypothesis that they obey the same distribution was tested using a 2-sample Kolmogorov–Smirnov (KS) test. Many of the subsegment pairs failed the test at significance level 0.05, some with a p-value as small as 10^{-11} . Likewise, since the Gaussian property is invariant under linear transformations, we can test the noise for Gaussian distribution using the whitened subsegments. In particular, many of the subsegments failed a KS test for normality at significance level 0.05, some with a p-value as small as 10^{-19} . The histogram of one such noise subsegment is shown in Fig. 5(C).

These properties, together with the abundant number of sample paths that can be drawn from recorded data (over 15 min of recording), make the recorded noise a suitable noise

²We ignore for the moment an apparent paradox of the spectra of nonstationary processes.

³A single platinum-iridium microelectrode (Frederic Haer Company, Bowdoinham, ME) with the nominal impedance of $2\text{ M}\Omega$ at 1 kHz was used for the recording. The electrode was advanced manually using a commercial micromanipulator (Narishige International USA, Inc., Long Island, NY). The data was acquired through a recording system (Plexon Inc., Dallas, TX) with a preamplifier and a bandpass filter (band 154 Hz–13 kHz). The signals were amplified and digitized (12-bit A/D converter, digitization rate 40 kHz) by a data acquisition card PCI-MIO 16E-4 with LabView (National Instruments, Austin, TX).

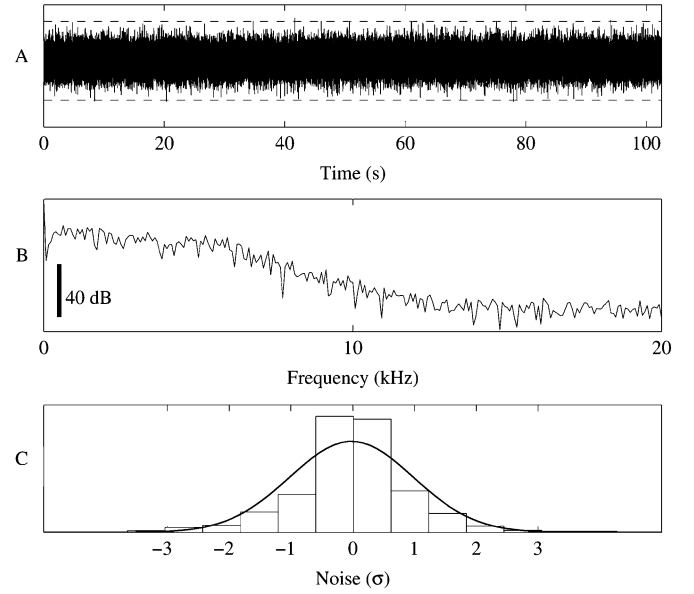


Fig. 5. (A) Recorded data containing no visible spikes—recorded noise. Dashed lines mark the 4σ bounds. (B) The power spectral density of the noise with the characteristic $1/f$ shape. (C) Normalized histogram of the whitened noise samples together with a Gaussian probability density function whose mean and variance are equal to the sample mean and sample variance. Note the discrepancy between the histogram and the Gaussian density, consistent with the KS test.

candidate for our computational model. Therefore, for each subsegment of simulated data, a subsegment of the recorded noise was selected at random, scaled to a standard deviation $\sigma_w = 20\text{ }\mu\text{V}$, and added to the simulated data. Note that this level of noise is consistent with typical noise levels in extracellular recordings [19], [20].

Although the neural noise was modeled as a nonstationary, non-Gaussian process, we will assume exactly the opposite for the purpose of algorithm development. The stationarity assumption allows us to by-pass the cumbersome treatment associated with nonstationary processes, while the Gaussian assumption often leads to closed form solutions. The downside is that the violation of these assumptions generally produces suboptimal solutions. We shall see in Section V-D that the algorithm is robust with respect to the violation of these assumptions. The success of our algorithm in experimental environment [1], where these assumptions are likely to be violated, points to a similar conclusion.

The level of noise in the system is measured by a signal-to-noise ratio (SNR), defined here as the energy of a spike signal divided by the expected noise energy. For a zero-mean wide sense stationary noise with a variance σ_w^2 , this definition reduces to

$$\text{SNR} \triangleq \frac{\text{RMS}(s)}{\sigma_w} \quad (4)$$

where $\text{RMS}(s)$ is the root-mean-square value of s , calculated over the supporting time interval of a single spike (1.6 ms in the present study).

III. UNSUPERVISED SIGNAL PROCESSING

As discussed in Section I, the core of our algorithm is based on optimization of the quality of recorded signals, which is measured by defining an appropriate signal quality metric. In order to obtain such a metric, the electrode signal (simulated signal in our study) must be processed accordingly. First, the signal needs to be tested for the presence of spikes. This is achieved through *spike detection*. Once the spikes are detected, we need to assess whether the detected spikes originate from a single neuron or multiple neurons. If spikes from multiple units are present, they need to be classified properly. This is achieved by means of *spike classification*, typically preceded by alignment and feature extraction, which serve as preprocessing steps (see Fig. 6). Although these processing steps have been traditionally used in neuroscience, we emphasize the fact that for a fully autonomous function of the control algorithm, all signal processing steps have to be *unsupervised*. Furthermore, a moving electrode will encounter a range of spike shapes and amplitudes (SNRs), even from a single neuron (see Fig. 4); thus, the signal processing must be adaptive and robust to these variations. Next, we describe a set of algorithms that carry out the key spike processing steps in a robust and unsupervised fashion that is necessary for our application. Some of these algorithms are novel, while some are variations of known techniques to make them suitable for our needs.

A. Spike Detection

Because of its practical importance to experimental neuroscience, the detection of spikes in noisy extracellular data is a classical problem. Unfortunately, most of the existing spike detection methods, such as window discriminator [22], principal components [23], or matched filtering [24], are supervised and, therefore, not suitable for autonomous probe operation. Other methods, such as amplitude detection [13] and power detection [24], can be implemented in an unsupervised manner, although with large variations in SNRs and firing rates, typically found in movable electrode operations, these methods are fraught with inconsistent performance [21].

Combining the theory of wavelets, statistics and detection theory, we have recently developed a robust unsupervised spike detection algorithm. The theory behind the algorithm is not the subject of this paper, and a detailed discussion on this topic can be found in [21]. This approach can be used for unsupervised spike detection over a wide range of SNRs and firing rates. Fig. 6 shows a simulated neural activity of two cells and the spikes detected by our wavelet method. Throughout the rest of the paper the spike detection will be assumed to be the wavelet-based technique of [21].

B. Spike Alignment

After detection, alignment of spikes is used to overcome the effects of jitter arising from background noise and finite sampling of the signal. Many classical approaches to spike alignment problem, such as alignment by the peak [25], or alignment by the “center of mass” [13], are not amenable to unsupervised applications, therefore, the correlation method of [23] was modified and used for unsupervised alignment. Briefly, the spikes were ordered by their amplitude, and the first spike was fixed.

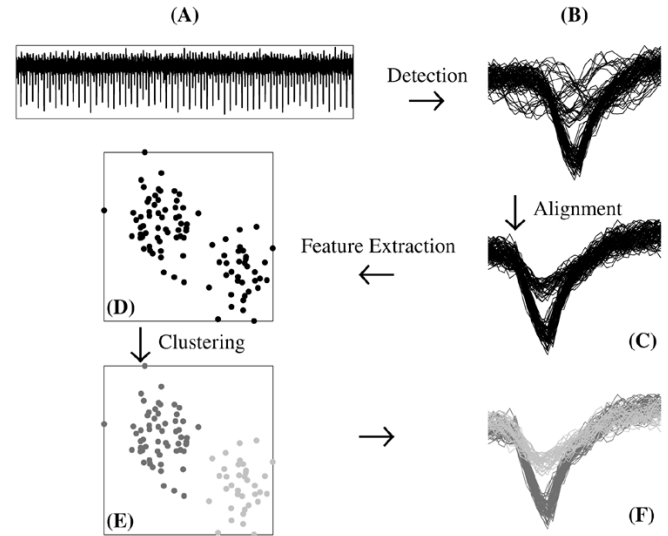


Fig. 6. (A) One second of simulated activity of two cells (sampling track V-6, see Section V-B). (B) Spikes detected by the wavelet detection method [21]. The 1.6 ms traces are centered at the estimated spike arrival times. Note the jitter in the estimated arrival times, especially for low-amplitude spikes. (C) Spikes aligned using our correlation method. (D) Spike features ($N_f = 2$) extracted as the two Haar wavelet coefficients with the largest magnitude. Each dot is a spike from the previous panel. (E) Two clusters (dark and light gray) identified using a Gaussian mixture model. There were no outliers in this case. (F) Class memberships in time domain.

A short (1.6-ms) segment in the neighborhood of the second arrival time was found that maximized the absolute value of the correlation with the first spike. This segment was taken as the second spike. A short segment in the neighborhood of the third arrival time was found that maximized the absolute value of the correlation with the first and the second spike. This segment was then declared the third spike, and the procedure was repeated in this fashion until all the spikes were aligned. Fig. 6(C) shows the spikes after alignment using our correlation method. Note that spike alignment effectively re-estimates the arrival times of detected spikes.

C. Spike Classification

To determine the sources of individual spikes in data containing multiunit activity, we use cluster analysis. Apart from their heuristic nature, the most widely used clustering methods, such as hierarchical [26] and k-means methods [27], do not provide the information regarding the number of clusters in the data, which makes them ill-suited for unsupervised applications. Alternatively, cluster analysis can be formulated within a probabilistic framework of finite mixture models [28], in which case the choice of the number of clusters reduces to a model selection problem. Additionally, finite mixture models efficiently handle outliers. The basic steps of our finite mixture model approach, which follows closely that of [29] and [30], are summarized below. A more detailed description is provided in the Appendix I.

To reduce the number of parameters necessary to describe the model, the first step extracts a low-dimensional feature set from the spike data. This process is referred to as *feature extraction*, and an example of 2-D spike features extracted using the Haar wavelet transform are shown in Fig. 6(D). In the next step it is

assumed that these features are sampled from some unknown probability density function (PDF), modeled as a *linear mixture* of a fixed number of component PDFs. The number of components equals the number of clusters, and neither the number of components, nor the parameters of the component PDFs are known. If we fix the number of components, the remaining unknown parameters can be found through the *expectation-maximization* (EM) algorithm [31], after which the component PDFs are fully specified. In a *model selection* step, we define a family of mixture models by varying the number of components, and for each member of the family we perform the model identification using the EM algorithm. Finally, a model is selected that maximizes the Bayesian information criterion (BIC) [32]. For the observations of Fig. 6(D), a 2-component model had the largest BIC, among a family of five different models. At each observation, the component PDFs are then evaluated and an observation is assigned to the component (cluster) with the highest PDF. The results of this clustering assignment are shown in Fig. 6(E).

D. Signal Quality Metric

The choice of a signal quality metric is not unique and may be application dependent. Perhaps the simplest spike feature that is representative of the signal quality is given by its peak-to-peak amplitude (PTPA), which was used throughout this paper. The pros and cons of this particular metric choice are discussed in Section VI.

In general, the presence of multiunit activity in the data has to be accounted for in the definition of signal quality metric, for otherwise the metric will be averaged over different cells, and will underestimate the signal quality of the target cell. This, in turn may confound the algorithm to spurious maxima, which could be far from optimum. Therefore, the PTPA is first evaluated over individual clusters, and the cluster which provides the largest average value of the PTPA is determined. We refer to this cluster as the *dominant cluster*. The PTPA of a spike from the dominant cluster defines a single observation of the signal quality metric. Generally, the dominant cluster will contain multiple spikes, thus providing multiple observations of the signal quality metric, which is important for stochastic optimization scheme discussed in Section IV.

IV. RECURSIVE STOCHASTIC OPTIMIZATION

We now formulate the electrode positioning problem within a framework of a recursive stochastic optimization, which is the key component of the *Optimize* state as discussed in Section I.

Let u and $y(u)$ denote the position of the electrode and the associated signal quality metric along a linear track with an arbitrary origin, respectively. Because of the noise in neural signals, the metric y is a random variable with an associated regression function $M(u) = E(y|u)$, where $E(\cdot|\cdot)$ denotes the conditional expectation operator. Generally, the function $M(u)$ is nonlinear. Moreover, it often exhibits unimodal character as suggested by experimental data in Fig. 7. Next, we will outline an algorithm that maximizes the regression function of the signal quality metric y given the position of the electrode u . The regression function itself is *unknown*; all we have are the noisy

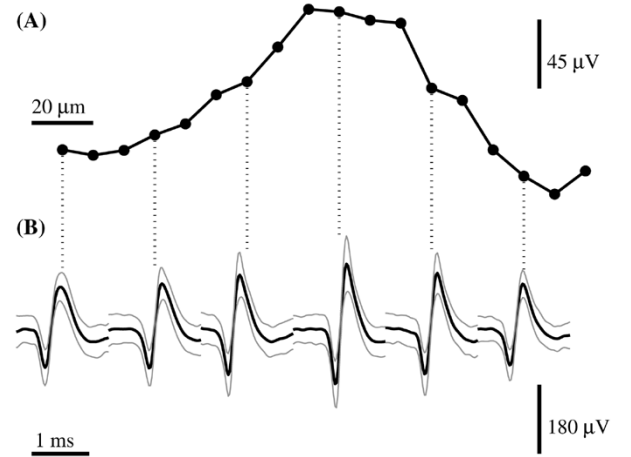


Fig. 7. (A) PTPA curve of a neuron from the parietal reach region of monkey posterior parietal cortex (see Section II-C for recording details). The data was obtained by sampling along a straight line in 10- μ m increments and recording for approximately 10 s at each location. The curve was constructed by averaging PTPA over the spikes detected at each location. The number of spikes per location varied from 213 to 478. In the large number of spikes limit, this function converges (pointwise) to the regression function of the PTPA given the position of the electrode. (B) Ensemble average \pm standard deviation of spikes at locations marked by dotted lines.

observations of the signal quality metric, sampled from the dominant cluster.

The problem of a regression function maximization from noisy observations has been traditionally solved through a recursive stochastic optimization. In their pioneering work [33], Kiefer and Wolfowitz devised a stochastic version of the gradient ascent method, also known as the Kiefer-Wolfowitz (KW) scheme. Under relatively mild regularity conditions on $M(u)$ and for appropriate choice of the recursion parameters, the KW scheme is provably convergent [33], [34]. Unfortunately, in the movable probe context, the KW scheme has very limited applicability. For example, since the method relies on a finite difference estimate of the regression function derivative, excursions of the electrode at each position are required to gather the necessary data. Moreover, as the iteration progresses, the variance of the estimated derivative becomes larger, which renders the convergence rather slow and leads to excessive electrode dithering. In practical applications, the unnecessary back-and-forth electrode movement inevitably leads to excessive tissue damage and possible inflammatory reactions. Next, we propose an alternative that adequately handles these problems.

At the core of the KW scheme is a calculation of the derivative through a finite difference method. In general, taking a difference of noisy data amplifies the effect of noise, which is the main reason for the nonsmoothness of difference based techniques. Therefore, it might be worthwhile to estimate the regression function itself, rather than its derivative. In the sequel, we propose a model-based approach, where the regression function estimation reduces to the problem of parameter estimation. Allied to a recursive scheme similar to the KW scheme, this provides an efficient way of estimating an optimal electrode position, u^* . The major benefit of this approach is that multiple

observations of the signal quality metric at each position, u_k , effectively smooth out the estimate of the regression function, so that excessive dithering of the sequence $\{u_k : k \in \mathbb{N}\}$ is eliminated. Also, if a differentiable model for the regression function is chosen, the derivatives of the regression function can be found *analytically*, thus eliminating the need for the electrode excursions, as required by the KW scheme.

We will explain briefly the idea behind our model-based approach here. A more detailed description and mathematical derivations can be found in Appendix II. The regression function model is assumed to be a linear combination of a certain number of basis functions chosen by the user. The model parameters and the number of basis functions (complexity) of the model are unknown and have to be estimated from the data. The model estimation procedure is *adaptive* in the sense that both the parameters and the complexity of the model will change throughout the algorithm's iterations as new data become available. Once the model (at iteration k) is known, the following recursive scheme is derived (see Appendix II):

$$u_{k+1} = u_k + C|H_k|^{-1}\xi_k \quad \forall k = k_0, k_0 + 1, \dots \quad (5)$$

where $C > 0$ is an appropriately chosen scale factor, and ξ_k and H_k are the estimates of the first and second derivative of the regression function at point u_k , respectively. Note that (5) represents a stochastic version of Newton's method with superior convergence properties than the stochastic gradient ascent method [35]. The convergence is considered attained at iteration k^* if $C|H_{k^*}|^{-1}\xi_{k^*} < \text{Tol}$, where Tol is a tolerance chosen by the user. The position u_{k^*} is then declared a solution, and the algorithm transitions to the *Maintain* state (see Section I).

V. RESULTS

The performance of the autonomously movable electrode algorithm was tested in a simulated environment. To properly test all the operations of our algorithm in a multiunit context, we simulated the extracellular potential of two neurons.

A. Simulations

Two identical cell models are placed in parallel [see Fig. 8(A)] to mimic the parallel organization of the cells in cortical columns. In a local coordinate system, the somata of cell 1 and cell 2 are centered at $(0, 0, 0) \mu\text{m}$ and $(50, 0, 0) \mu\text{m}$, respectively. Rotation of the cells around y-axis did not significantly affect the results. The $50 \mu\text{m}$ distance between the somata is consistent with the wide range of cell densities in the cerebral cortex [36]. To account for complex nonlinearities in the kinetics of voltage-gated channels, the simulations were carried out with variable step size in NEURON, ranging between 0.02 ms and 1.0 ms [10]. The firing rate of each cell was 58 Hz.

While our extracellular simulator can work at an arbitrary sampling rate, we found a rate of 20 kHz a good compromise between computational speed and accuracy of spike representation. Thus, the transmembrane currents were re-sampled at 20 kHz in MATLAB[®] using linear interpolation. The choice of re-sampling method did not cause any significant differences in

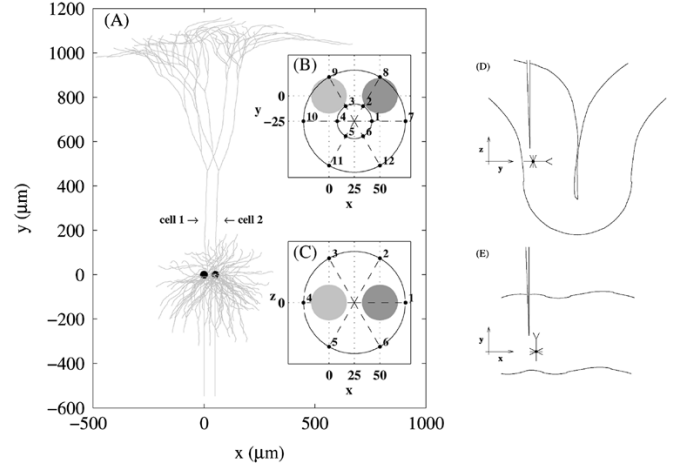


Fig. 8. (A) Two dimensional projection of two cells in the space. (B) A close-up of the two somata (cell 1 light gray and cell 2 dark gray) showing 12 vertical sampling tracks. Dendrites are left out for clarity. (C) Equivalent plot for horizontal sampling tracks. (D) Recording from the bank of the sulcus where the recording track is perpendicular to the apical dendrite. Local coordinate system is shown to the left. (E) Recording from a regular cortical layer where the recording track is parallel to the apical dendrite.

the results. The membrane currents of the second cell were delayed by 8 ms to prevent the cells from firing simultaneously.⁴ For each sampling position, the cells' locations were evaluated based on the model in Appendix III, and the extracellular potential due to the two cells was calculated as a *superposition* of the potentials of individual cells, which were obtained as described in Section II-B. Due to relatively high firing rates, 1 s of data, simulated at each electrode position, provided enough spikes for successful postprocessing (e.g., clustering and objective function evaluation). A 1-s-long subsegment of the recorded noise, seen in Fig. 5, was selected at random, re-sampled at 20 kHz, scaled to $\sigma_w = 20 \mu\text{V}$, and added to the simulated data at each electrode position.

B. Definition of Sampling Tracks

The performance of our algorithm was tested along many transects, including the two principal directions seen in Fig. 8(B) and (C). These two sampling directions mimic the process of recording from a sulcus and regular cortical layers, as shown in Fig. 8(D) and (E). We simulated several individual sampling tracks along each of these two principal directions. These tracks are numbered, with the prefix V or H used to distinguish between the vertical and horizontal tracks. The tracks of the vertical and horizontal directions start in a plane located at $z = 100$ and $y = 80$, respectively.

To obtain statistically significant results, 100 trials were performed for each sampling track, with the admissible range of electrode movement of $z \in [100, -100]$ for V-tracks and $y \in [80, -120]$ for H-tracks. These values are in agreement with a relatively small field of the detectable potentials of the model cell (see Fig. 4). They also reduce the time the algorithm spends in the *Search* state, thereby lowering the computational overhead of individual trials. If the algorithm did not converge to a solution within these ranges, the trials were aborted and declared

⁴For demonstration purposes we ignore the problem of spikes which are overlapping in time.

TABLE I
CHOICE OF PARAMETERS AND METHODS USED FOR PRESENT ANALYSIS

Variable/Operation	Value/Type	Reference
$[W_{\min}, W_{\max}]$ (spike width)	$[0.5, 1.0]$ ms	[21]
L (detection sensitivity)	0	[21]
N_f	2	Section I-A
feature extraction	Haar wavelet	Section I-A
N_G	$\lfloor \log_2 N_s - 1 \rfloor, N_s \geq 4$ $\{\emptyset\}$ no clustering, $N_s < 4$	Appendix I-D
N	5	Appendix II
k_0	6	Appendix II-A
C	1	Eq. (5)
Tol	$0.5 \mu\text{m}$	Section IV
Δ_s	$25 \mu\text{m}$	Section I
Δ	$5 \mu\text{m}$	Eq. (13)
Δ_{\max}	$10 \mu\text{m}$	Appendix II-C

unsuccessful. Depending on their position relative to the cells, the SNRs of individual V-tracks were in the range of $[0.4, 17]$, and in the range $[0.4, 5]$ for H-tracks. The SNR was calculated as

$$\text{SNR} \triangleq \frac{\text{RMS}(s_1) + \text{RMS}(s_2)}{2\sigma_w} \quad (6)$$

which is a modification of the previous definition to account for the presence of two neurons. Because both cells have the same firing rates, the two RMS values are weighted equally [21].

C. Choice of Parameters

The choice of algorithm parameters and spike processing methods used in the analysis that follows, is given by Table I. Generally, the performance of the algorithm depends on chosen parameters and methods, and analyzing the performance over a range of parameters and methods is not the subject of this paper. However, we have found that as long as model selection parameters k_0 , N , and N_G are chosen reasonably, the algorithm offers consistent performance.

D. Performance of the Algorithm

Since the electrode and the cells are movable, an efficient way to test the performance of the algorithm is to measure how close the electrode comes to the dominant (closest) cell upon convergence (at iteration k^*) and compare this distance, denoted by d_{k^*} , to an *exact optimum*, d^* , across trials.

The exact optimum d^* was calculated off-line through a brute force numerical search (described below) over the sampling tracks, using very fine ($1 \mu\text{m}$) sampling steps. To eliminate the dependence of the exact optimum on the electrode movement, the cell coordinates were kept fixed in the search process. At each sampling location, a 1-s data segment was simulated and the spikes corresponding to the dominant cell were used to obtain the samples of the signal quality metric. Since the precise occurrence times of the spikes from the two cells were exactly known, no detection, alignment or clustering was performed. This eliminated the error due to false detection, misclassification, etc., and ensured that the exact optimum evaluation was error-free. The regression function $M(u)$ was estimated as the sample average of the signal quality metric and its maximizer

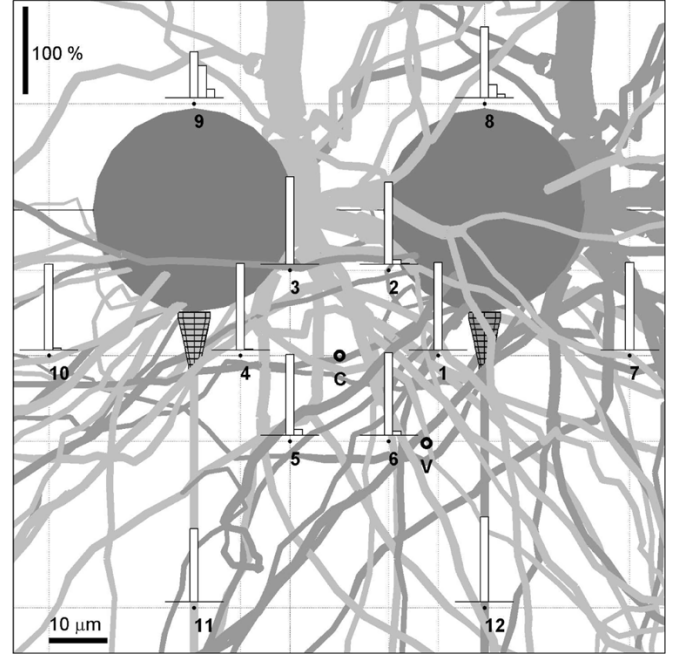


Fig. 9. Histogram (bin size $8 \mu\text{m}$) of error $\varepsilon \triangleq d_{k^*} - d^*$ over different vertical tracks. The data is only shown for successful trials, e.g., the histogram of V11 normalizes to 84%. The bin center $\varepsilon = 0$ is located next to the dot representing the sampling track. The dendrites of cell 1 and cell 2 are in light and dark gray, respectively. The two axon hillocks are shown as meshed cones adjacent to the somata.

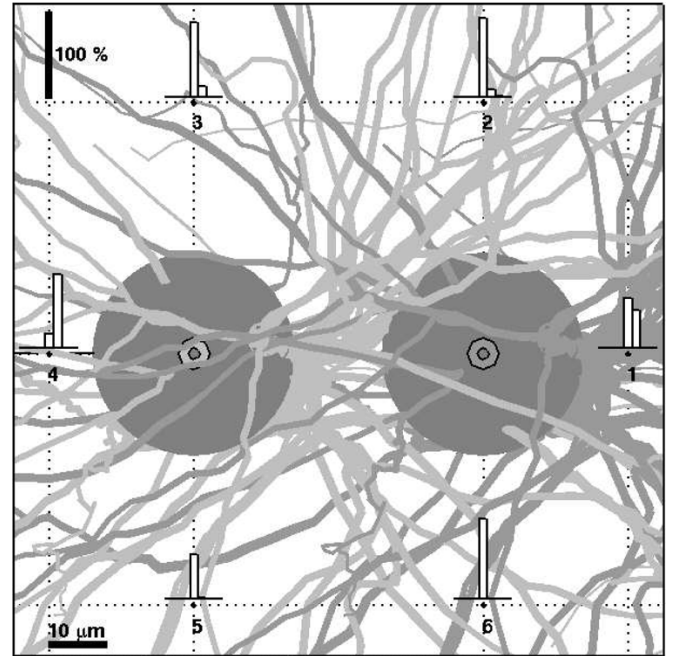


Fig. 10. Equivalent plot for horizontal tracks. The axon hillocks are shown as two annular regions in the center of the somata.

u^* was found. At the point u^* , the exact optimum d^* was found as the distance between the electrode tip and the soma center of the dominant cell.

1) *Convergence*: The convergence properties of our algorithm are quantified by the distribution of error $\varepsilon \triangleq d_{k^*} - d^*$, calculated over 100 trials (see Figs. 9 and 10). For V-tracks

(H-tracks), the estimated solutions were within $10\ \mu\text{m}$ of the exact optimum in 96% (90%) of the trials, and the algorithm failed to converge in less than 2% (9%) of the trials, respectively. The *success rate* of a track, defined as a percentage of convergent trials, was dependent on the maximal SNR attained over the track. The tracks with the smallest SNR were the ones that did not yield 100% success rates, namely: V11 (SNR = 1.92, success rate 84%), V12 (SNR = 2.01, success rate 97%), H5 (SNR = 1.84, success rate 54%), and H6 (SNR = 1.91, success rate 92%). Likewise, the lowest SNRs yielding 100% success rates were: 2.01 (V9), 2.04 (V8), 2.02 (H3), and 2.04 (H2). These results suggests that there is a *critical value* of SNR (~ 2) that separates fully successful from partially successful sampling tracks. Additional analysis of partially successful tracks revealed regression functions that were hardly peaked. Coupled with a low SNR, this occasionally confounded the estimated model to a flat line and prevented convergence. Furthermore, the errors in spike detection, alignment and clustering were more frequent under low SNRs, thereby increasing the likelihood of failure in these trials. Finally, the average *convergence rates* (defined here as the number of iterations in the *Optimize* state), of these sampling tracks were lower than those of the fully successful sampling tracks (16 versus 13 for V-tracks and 20 versus 15.5 for H-tracks). Similar convergence rates (10–20 iterations) have been reported in experimental applications of our algorithm [1].

2) *Cell Isolation*: The analysis of convergent trials revealed that the algorithm always isolated the correct target neuron: cell 1 for tracks V3–V5, V9–V11, and H3–H5, and cell 2 otherwise. Among other factors, the isolation of a correct target neuron depends critically on correctly performed signal processing steps, especially spike alignment and classification. Another very important factor for correct isolation is the position of the sampling track with respect to the nearby cells. For example, probing the algorithm along a track C that lies exactly in between the two neurons (see Fig. 9), yielded a solution that confused the two cells. Additional analysis of the spike waveforms revealed that they were rather similar, and that the classifier consistently failed to estimate the number of clusters correctly. Clearly, this “confusion region” is not confined to a single track, rather a whole area around the track C is affected. Determining the boundaries of this area, as well as the probability of confusion, is beyond the scope of this paper. It suffices to know that all the sampling tracks that we tested are outside of this area.

3) *Cell Tracking*: This section demonstrates that the algorithm is capable of re-optimizing signal quality under modeled tissue movements (see Appendix III). Fig. 11 shows a typical course of a cell tracking simulation for the sampling track V in Fig. 9. Similar results were obtained for other sampling tracks. After several iterations in the *Search* state, spikes were found and the algorithm transitioned to the *Optimize* state. Upon convergence, the algorithm transitioned to the *Maintain* state, where it remained until the deviation of the signal quality exceeded a tolerance, set in advance as 3 standard deviations of the signal quality acquired initially. The degradation of the signal quality was due to slow (relaxation) tissue movements. The optimality was then maintained by toggling between the *Maintain* and *Optimize* state, where the frequency of the switching depends on the time constant of tissue drift, and on the value of the tolerances. As the number of iterations increased, the cells drifted

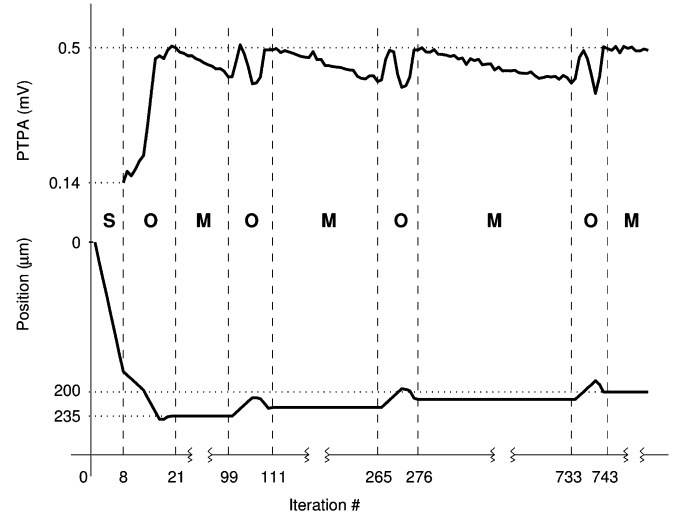


Fig. 11. (Top) The average value of signal quality metric (PTPA) through iterations. Note the drift in the signal quality metric due to tissue movement. The dashed vertical lines denote the state transitions: S (*Search*), O (*Optimize*), and M (*Maintain*). (Bottom) The position (depth) of the “electrode” as a function of iteration number. Despite the cells’ drift, the objective remains fairly constant ($\sim 0.5\ \text{mV}$) at the convergence iterations $k^* = 21, 111, 276, 743$.

closer to their initial position (see Appendix III) and the algorithm remained in the *Maintain* state for longer periods of time. Although the newly found optima moved with the tissue, the average signal quality remained unchanged, as seen in Fig. 11. The simulation was discontinued after 900 iterations, which at $\sim 6\ \text{s}$ /per iteration (see Appendix III), roughly corresponds to $1(1/2)$ hours of recording time. The time constant of the signal quality drift and the frequency of the switching are consistent with the experimental applications of our algorithm [1], where SNR was used as a signal quality metric. Likewise, the number of iterations to re-isolate the cell ranged between 10 and 20, similar to our experimental findings.

VI. DISCUSSION

Stability of Cell Isolation and Tracking: Under dynamically moving tissue conditions, a critical factor for the stability of cell isolation is the *iteration time*, i.e., the average time necessary to acquire and process data per iteration. While the iteration time of our simulations was relatively short (see Appendix III), as much as 30 s iteration time was observed in experimental applications [1], primarily due to low firing rates. If the iteration time is long relative to the time constant of the tissue movement, by the time the iteration is executed, the target cell may have moved too far. Determining the exact conditions for this to happen, involves not only the iteration time and the time constant of tissue movement, but also the variable factors such as the step size of electrode movement, the direction of tissue movements, etc. For the representative sampling tracks of our model, and under the assumption that the tissue moved parallel to the tracks, the transient tissue movements (see Appendix III) as fast as $0.5\ \mu\text{m/s}$ were handled by the algorithm successfully. Said differently, the displacement of the cells, as large as $3\ \mu\text{m}$, between two iterations, did not pose any threat for the algorithm to converge. A similar analysis applies to the stability of cell tracking, with the

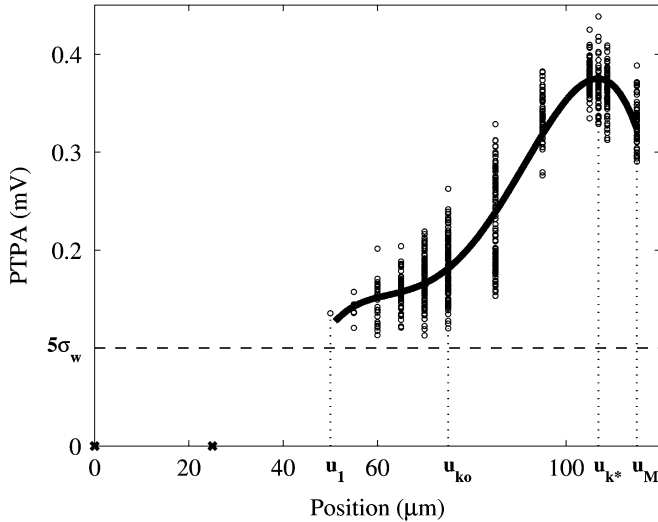


Fig. 12. Observations of signal quality metric as a function of electrode position. The first two iterations, marked by (x), did not yield any observations because the spikes were well below the noise level (marked by the dashed line). The gray line is the regression function model estimated at the convergence iteration k^* . The overshoot, defined as: $O \triangleq u_M - u_{k^*}$, was 8 μm in this case (a single trial along the track V5).

addition of the signal quality tolerance as a parameter. Under the same assumption, the algorithm easily compensated for slow tissue movements ($\tau = 30$ min, see Appendix III) with the mean velocities up to 2.5 $\mu\text{m}/\text{min}$. Similarly, from the experimental applications of our algorithm [1], we estimated mean tissue velocities of ~ 1.5 $\mu\text{m}/\text{min}$.

Limitations of the Algorithm: One of the limitations of the autonomously movable electrode algorithm stems from the use of PTPA as a signal quality metric. As the electrode is moved closer to the target cell and the corresponding PTPA is increased, the spikes from the neighboring cells may also have increasing amplitude. Because the PTPA metric relies solely on the information from the dominant cluster, it could lead to a solution with a strongly *confounding activity*. In a feature space, the maximization of the PTPA metric can be viewed as a maximization of the distance between the dominant cluster and some noise cluster, obtained by projecting the noise samples to the feature space. A more sophisticated definition of a signal quality metric would also incorporate the distance between the dominant cluster and other clusters, as suggested by [37]. However, such a definition is very sensitive to misclassification, so the choice of a metric should be viewed as a tradeoff between the sensitivity to misclassification and the probability of converging to a confounding solution.

A potential source of confusion among cells is the *overshoot* of the electrode's position beyond the optimal location that occurs frequently during adaptive regression model estimation (see Fig. 12). Although, the average value of overshoot across trials and sampling tracks was rather small (< 3 μm), values as high as 11 μm were observed occasionally ($\sim 1\%$ of the trials). This raises the possibility that the algorithm switches its focus to a new cell that lies along the sampling track, right behind the target neuron. Clearly, the geometric arrangement of the cells and sampling tracks, the cell size, the channel distribution,

SNR and many other factors determine the outcome of this event. While the 10-micron overshoot was not a concern for our simulations (soma diameter 35 μm , cell distance 50 μm), it may be so for smaller, densely packed neurons (e.g., soma diameter < 20 μm , cell distance < 30 μm). The practical consequences of this event, however, are unclear, given that the cells are so close together, and that the signal quality they provide is quite similar.

Extensions to Experimental Applications: The remarkable success rates (Figs. 9 and 10) are somewhat expected given the regularity of firing of the simulated neurons. Highly regular firing patterns may not always be observed in actual recording experiments. Instead, the activity of neurons could be highly nonstationary, with a wide range of firing rates. For example, in experimental applications of our algorithm [1], the firing rate was typically very low (a few hertz) and up to 20 s of data at a single electrode position was necessary to correctly perform unsupervised clustering and in turn correctly estimate the signal quality metric. Clearly, a cell with an arbitrarily low firing rate can be tracked, but an increased iteration time is likely. This tradeoff can be practically handled by setting a lower bound for the firing rate of the cell to be tracked, for example, a bound of 2 Hz was used in [1].

Another practical issue arises when an electrode comes too close to a cell, which may damage the cell's membrane, and cause its subsequent death. This phenomenon is often concurrent with the observation of very large action potentials. This situation can be practically handled by constraining optimization process to stop when the signal quality metric exceeds some suitably chosen upper bound. For example, an upper bound of $\text{SNR} = 12$ was used for the experiments in [1].

Finally, unpredictable events that sometimes arise in recordings from behaving animals, cannot be handled by a simple FSM shown in Fig. 1. For example, the cell being tracked may fire intermittently or may stop firing in the middle of the optimization process. To prevent suboptimal or meaningless solutions, the FSM can be augmented with additional states and transitions that account for these effects [1].

The successful use of our algorithm in acute recording experiments would significantly improve the productivity of recording neuroscientists by freeing them from the tasks such as manual positioning and frequent readjustments of the electrodes. The algorithm is implemented in software, therefore, it is portable and can be easily adapted to many configurations that incorporate a computer controlled motorized microdrive. Nowadays, many commercial motorized microdrives (e.g., Thomas recording GmbH) are controllable through a universal serial bus, or serial port, which makes them suitable for integration with our autonomously movable electrode algorithm. Since the role of human operator is minimized, the implementation of the algorithm essentially reduces to the off-line selection of parameters, similar to the ones given by Table I.

Extensions to Chronic Applications: If successfully implemented in chronic recording systems, our algorithm could potentially increase the longevity of the recording implant and improve the signal yield and signal quality while reducing the number of necessary implantation surgeries. These results would be important for successful practical implementation

of emerging brain-machine interfaces such as neural prosthesis. However, the implementation of autonomously movable recording electrodes in chronic applications is contingent upon the design of chronic motorized microdrives, and in the case of neuro-prosthetic systems, upon the development of appropriate microelectromechanical systems technology.

VII. CONCLUSION

Based on a detailed biophysical model of extracellular potentials, we have developed an algorithm that autonomously finds and maintains the optimal electrode recording position. For a two-neuron model with the cell sizes, densities and orientations representative of a cerebral cortex, we have tested the algorithm along a representative set of sampling tracks. The basic results are as follows: 1) for the sampling tracks exceeding a critical SNR ($\text{SNR} \sim 2$), the algorithm converges to a close vicinity ($< 10 \mu\text{m}$) of the exact optimum; 2) for the same set of sampling tracks, the algorithm is capable of maintaining the optimality of recorded signals in the face of modeled tissue movements, with the tissue velocities comparable to those found in actual recording experiments. The algorithm is facilitated by a stochastic optimization of a suitably defined signal quality metric. Although we have tested the algorithm against the signal quality metric based on the spike amplitude, other signal quality metric choices are acceptable, as discussed in [1]. The evaluation of the signal quality metric requires a series of neural signal processing steps such as spike detection, alignment and classification. For fully autonomous operation, the processing steps must be implemented in an unsupervised fashion. Our current research efforts are directed toward finding the “best” spike features for unsupervised classification and the design of a robust alternative to the amplitude-based signal quality metric. Additional research efforts are aimed at the implementation of our algorithm in novel acute multielectrode recording devices.

APPENDIX I

BASIC CLUSTERING STEPS

A. Feature Extraction

Let $S \in \mathbb{R}^{N_s \times N_t}$ be a matrix of N_s aligned spikes, where N_t is the number of samples per spike. For simplicity, we consider features that are linear function of data, i.e., $F = SE$, where $F \in \mathbb{R}^{N_s \times N_f}$ is the feature matrix, $E \in \mathbb{R}^{N_t \times N_f}$ is the transformation matrix, and N_f is the dimension of the feature space chosen by the user. The merit of this method is that $N_f \ll N_t$. We project the spike data onto the Haar wavelet basis, which results in simple wavelet coefficient features. The N_f coefficients with the largest average magnitude are then selected as features (formally the columns of E are chosen as the basis vectors corresponding to these coefficients). Clustering with wavelet features is somewhat superior to clustering with traditional principal component features, perhaps due to the fact that the bi-phasic shape of the spikes (see Fig. 4) is better captured by the Haar wavelets, resulting in a more discriminative low-dimensional representation.

B. Finite Mixture Models

Let f_i be a sample of the feature vector (i th row of matrix F) corresponding to the i th spike. In the finite mixture model framework [38], it is assumed that this observation is sampled from a PDF that can be modeled as a linear combination of $G+1$ component PDFs, p_j , i.e.,

$$p(f_i | \Pi; \Theta) = \sum_{j=0}^G \pi_j p_j(f_i | \theta_j) \quad \forall i = 1, 2, \dots, N_s$$

where π_j are the *mixing parameters* satisfying $\sum_j \pi_j = 1$, θ_j is a parameterization of the PDF p_j , and $\{\Pi; \Theta\} \triangleq \{\pi_j; \theta_j : j = 0, \dots, G\}$. Viewed as a function of the parameters Π and Θ rather than a function of the feature data, the overall mixture density is called the *mixture likelihood* and is given by

$$\mathcal{L}_{\text{MIX}}(\Pi; \Theta | F) = \prod_{i=1}^{N_s} \sum_{j=0}^G \pi_j p_j(f_i | \theta_j) \quad (7)$$

where we have assumed that the samples f_i are statistically independent. Conditioned upon a spike from cell j , the distribution of features f_i is Gaussian.⁵ In other words, $\{p_j(\cdot | \theta_j) : j = 1, 2, \dots, G\}$ are chosen as Gaussian components. The outliers are handled through a uniform component $p_0(\cdot | \theta_0) = 1/V$, where V is defined as the hyper-volume of the data [29], [30]. For the sake of notational compactness the symbol θ_0 is retained, although $\theta_0 = \{\emptyset\}$. If an observation f_i belongs to a well-defined cluster, it is generated by one of the Gaussian components. Otherwise, the observation is generated by the uniform component p_0 , and declared an outlier. The number of Gaussian components G defines the number of clusters.

C. Expectation-Maximization Algorithm

In general, the maximization of the mixture likelihood (7) with respect to Π and Θ must be performed numerically. For this task we employ the EM algorithm, for which there is an extensive literature in the case of Gaussian mixtures [39]. Among known limitations of the EM algorithm, sensitivity to the choice of initial condition is the most serious one. As suggested by [29], a reasonable initial guess can be obtained by using the heuristic clustering techniques. We used the hierarchical clustering with Ward distance [26], which provided reasonably good initial partition of the data.

D. Model Selection

Given a family of candidate mixture models $\{M_G : G = 1, 2, \dots, N_G\}$, the goal is to find the order G of a mixture that optimally fits the data, subject to some objective function. Note that the mixture likelihood (7) is a poor choice of objective, since the most complex mixture M_{N_G} is bound to have the highest

⁵Under the Gaussian noise assumption (Section II-C) the entries of the spike matrix S are Gaussian random variables. Linear feature extraction further implies that the features are also Gaussian.

likelihood, which leads to over-clustering. From Bayes' theorem, the probability of any candidate model, given the data F and some prior knowledge I , can be written as

$$P(M_G | F, I) \propto p(F | M_G, I) P(M_G | I) \quad \forall G = 1, 2, \dots, N_G \quad (8)$$

where P denotes a probability and p represents a PDF. Ideally, the model that maximizes the posterior (8) should be chosen. Assuming that the candidate models have uniform priors $P(M_G | I) = 1/N_G$, the maximization of the posterior reduces to the maximization of the integrated likelihood term $p(F | M_G, I)$. As the name suggests, the exact evaluation of this quantity involves numerical integration in multidimensional parameter space, e.g., a mixture of uniform and three Gaussian components in a two-dimensional (2-D) feature space requires 19 parameters. We use an approximation based on the BIC, although other successful approximations have been reported [40]. The BIC is defined as

$$\text{BIC}_G \triangleq 2 \log p(F | M_G, I) \approx 2 \log \mathcal{L}_{\text{MIX}}(\Pi^*; \Theta^* | F) - \nu_G \log N_s, \quad \forall G = 1, 2, \dots, N_G$$

where ν_G is the number of parameters of the mixture M_G . The optimal mixture is the one with the highest BIC. Once the order, G^* , of the optimal mixture is known the clustering rule is enforced via

$$f_i \in C_{j^*} \Leftrightarrow j^* = \arg \max_{0 \leq j \leq G^*} p_j(f_i | \theta_j^*)$$

where C_j is the label of the j th cluster, and θ_j^* are the optimal parameters of the mixture, found through the EM algorithm.

We finish this section by noting that for a successful clustering step, the number of observations N_s must be sufficiently high with respect to the maximal order N_G of the candidate model. If not, the covariance matrices of individual components are typically ill-conditioned or singular, and may cause the EM algorithm to fail. Since we do not know how many spikes will be detected at each sampling position, we will choose the maximal order adaptively $N_G = N_G(N_s)$ (see Table I).

APPENDIX II

ADAPTIVE ESTIMATION OF REGRESSION FUNCTION MODEL

Let $\{u_1, u_2, \dots, u_k\}$ be a sequence of (electrode) positions after k iterations with the corresponding signal quality metric $\mathcal{Y}_{1:k} = \{y(u_1, \omega), y(u_2, \omega), \dots, y(u_k, \omega)\}$, where the variable ω signifies the random sampling of y . At each position u_j ($j = 1, 2, \dots, k$), multiple observations of the signal quality metric are taken, i.e., $y(u_j, \omega) = \{y(u_j, 1), y(u_j, 2), \dots, y(u_j, m_j)\}$, where m_j is the total number of observations at u_j . In general, $m_1 \neq m_2 \neq \dots \neq m_k$. The regression function $\hat{M}(u)$ after k iterations is modeled as follows:

$$\hat{M}(u, n_k, B_k) = \sum_{i=1}^{n_k} b_{i,k} \psi_i(u) \quad (9)$$

where $\psi_i(u) = u^{(i-1)}$ are polynomial basis functions, $B_k = [b_{1,k}, b_{2,k}, \dots, b_{n_k,k}]^T$ are the corresponding expansion coefficients, and n_k is the number of basis functions. The justification of this model choice will become clear shortly. Note that the model \hat{M} depends linearly on parameters B_k , and that neither B_k , nor n_k are known and, therefore, must be estimated from the data. Similar to Appendix I-D, the estimation of the "right" n_k amounts to a *model selection* problem. Given a family of models $\{\hat{M}(u, n_k, B_k) : n_k = 1, 2, \dots, N\}$, where N is the maximal order of the candidate model (a number chosen before experiment begins), the goal is to select the model that optimally fits the data. The definition of optimality affects the estimate \hat{M} significantly, therefore, it is crucial to choose the metric appropriately. In particular, maximizing the likelihood of the model leads to a selection of the most complex model ($n_k = N$) and over-fitting occurs. An especially elegant solution to the problem of over-fitting is offered by Bayesian probability theory [41], which in the case of polynomial basis functions yields solution analytically and is, therefore, amenable to fast calculations. This is also the main argument against the use of more general basis functions, such as radial basis functions, in the model (9). Generally, more complex basis functions render the Bayesian model selection computationally expensive, as discussed briefly in Appendix I-D. In this case the model selection is typically performed by either heuristic [42], or approximate [43], [32] criteria, both of which are inferior to Bayesian model selection. On the other hand, signal quality curves are often simple (see Fig. 7), and they can be accurately modeled using polynomials of low order. To increase the flexibility of the model and to account for modeling of more complex regression functions, we can always increase the maximum order N of the candidate model.

A. Bayesian Model Selection

Given a family of candidate models $\{\hat{M}(u, n_k, B_k) : n_k = 1, 2, \dots, N\}$, the goal is to select the order of the model that is most probable in view of the data $\mathcal{Y}_{1:k}$ and some prior information, I . The probability of the model \hat{M}_{n_k} given $\mathcal{Y}_{1:k}$ and I follows from Bayes' theorem

$$P(\hat{M}_{n_k} | \mathcal{Y}_{1:k}, I) = \frac{p(\mathcal{Y}_{1:k} | \hat{M}_{n_k}, I) P(\hat{M}_{n_k} | I)}{p(\mathcal{Y}_{1:k} | I)} \quad \forall n_k = 1, 2, \dots, N \quad (10)$$

where \hat{M}_{n_k} is short for $\hat{M}(u, n_k, B_k)$ with fixed n_k , P is a probability mass function and p is a PDF. For successful Bayesian model selection, the number of iterations k has to be sufficiently high with respect to the maximal order N . We denote the smallest admissible number of iterations by k_0 . For iterations up to k_0 , the sampling of data is simply not adequate to reliably model the regression function and the heuristic of the recursive stochastic optimization for $k < k_0$ will be discussed later. The order of the model is selected that maximizes the posterior probability $P(\hat{M}_{n_k} | \mathcal{Y}_{1:k}, I)$, i.e.,

$$n_k^* = \arg \max_{1 \leq n_k \leq N} P(\hat{M}_{n_k} | \mathcal{Y}_{1:k}, I) \quad \forall k = k_0, k_0 + 1, \dots$$

In order to perform the maximization above, the posterior $P(\hat{M}_{n_k} | \mathcal{Y}_{1:k}, I)$ of each candidate model \hat{M}_{n_k} must be evaluated. To carry out this calculation, the unknown parameters B_k must be integrated out through *marginalization* [44]. Because of the Gaussian noise assumption and linear dependence of the model on the parameters, the marginalization of B_k , and in turn the calculation of the posterior $P(\hat{M}_{n_k} | \mathcal{Y}_{1:k}, I)$, can be performed analytically. The details of this calculation are omitted, but a similar model selection example can be found in [44]. In the spirit of Bayesian probability theory, the posterior $P(\hat{M}_{n_k} | \mathcal{Y}_{1:k}, I)$ calculated at iteration k is used as a guess for the prior $P(\hat{M}_{n_{k+1}} | I)$ at iteration $k+1$ in (10). The recursion is initialized as $P(\hat{M}_{n_k} | I) = 1/N$ at iteration $k = k_0$, which reflects our complete initial ignorance about the model. Once the order n_k^* of optimal model at iteration k is known, the parameters of the model $\hat{M}(u, n_k^*, B_k)$ need to be estimated.

B. Parameter Estimation

While Bayesian probability theory can also be used to infer the posterior of parameters B_k given the observations $\mathcal{Y}_{1:k}$ and prior information I , it is easier and computationally more efficient to use a maximum likelihood (ML) method, which under the linear model (9) and Gaussian noise assumption, reduces to the linear least squares method. As the number of observations increases, the influence of the prior information I on the estimate decreases, and the ML solution approaches the maximum posterior solution obtained through Bayesian probability theory. Therefore, the parameter estimation problem can be formulated as

$$B_k^* = \arg \min_{B_k} \left\{ \sum_{j=1}^k \|\Psi_{j,k} B_k - Y_j\|^2 \right\} \quad \forall k = k_0, k_0 + 1, \dots \quad (11)$$

where $Y_j = [y(u_j, 1), y(u_j, 2), \dots, y(u_j, m_j)]^T$ and the matrix $\Psi_{j,k} \in \mathbb{R}^{m_j \times n_k^*}$ consists of m_j identical rows given by: $[1, u_j, \dots, u_j^{(n_k^*-1)}]$. The solution to (11) reduces to finding the pseudoinverse of a matrix. Once the optimal parameters B_k^* are estimated, the optimal model $\hat{M}^{(k)}(u) \triangleq \hat{M}(u, n_k^*, B_k^*)$ at iteration k is fully specified.

C. Recursive Stochastic Optimization With Basis Functions

From (9) it follows that the derivative $M'(u)$ of the regression function at iteration k can be estimated as

$$\xi_k = \sum_{i=1}^{n_k^*-1} i b_{i+1,k}^* u_k^{(i-1)} \quad \forall k = k_0, k_0 + 1, \dots \quad n_k^* \geq 2.$$

Similarly, we can estimate $M''(u)$ at iteration k , denoted by H_k . Based on this we write the following variant of the Newton's method

$$u_{k+1} = u_k + C |H_k|^{-1} \xi_k \quad \forall k = k_0, k_0 + 1, \dots \quad (12)$$

where $C > 0$ is an appropriately chosen scale factor. In principle, the scale factor C can be used to calibrate the step size in (12), still large steps may occur occasionally, especially in the early iterations when the estimated model $\hat{M}^{(k)}(u)$ may be

far from the true function $M(u)$. From an experimental perspective, large steps of the electrode are generally unacceptable, therefore, we limit the maximum step size by a constant Δ_{\max} , chosen before the experiment. This is especially useful for iterations where the optimal model is found to be a straight line ($n_k^* = 2$), which results in $H_k = 0$ and infinitely large step size in (12). Likewise, if for some $k > k_0$ we obtain a flat line model, i.e., $n_k^* = 1$, then $\xi_k = 0$ and the recursion (12) breaks. In this case we use a simple control strategy

$$u_{k+1} = u_k + \Delta \quad (13)$$

where Δ is a constant step size chosen before the experiment initiates. Finally, for iterations $k < k_0$ the control strategy (13) is employed again.

APPENDIX III MODELING TISSUE MOVEMENT

The nature of neural tissue movement caused by the movement of a recording electrode is not well understood. The understanding of the overall process likely involves a complex mechanical analysis. Here we develop a simple model of the tissue movement. The purpose of this model is not to capture the complexity of the tissue dynamics, but rather to perturb the cell's position, so that the algorithm can be tested against these types of disturbances, which are commonly found in acute extracellular recordings.

It is believed that the movement of neural tissue in response to electrode movement has two significant time scales: fast and slow. Due to friction forces, even the sharpest electrode will cause tissue compression as the electrode is advanced through the neural tissue. The time scale of these movements is short (seconds and minutes). Presumably, the energy stored through the tissue compression is released through subsequent relaxation, which is a much slower process (time constant 1/2–1 h). While the fast (transient) tissue movements are more relevant for the convergence of the algorithm (*Optimize* state), the slow (relaxation) movements are more relevant for the stability of the optimal solution (*Maintain* state).

Let $ep_k, cp_k \in \mathbb{R}^3$ denote the position of the electrode tip and the position of the cell (its soma center) at iteration $k = \{0, 1, \dots\}$, respectively. We simply assume that all the segments of the cell undergo the same displacement (rigid motion), which is proportional to the electrode displacement, i.e.,

$$cp_k = cp_0 + \alpha g(ep_k - ep_0) \exp(-k/\tau) \quad \forall k = \{1, 2, \dots\} \quad (14)$$

where α is a proportionality constant, g is a gain factor and τ is a time constant of the tissue relaxation. Let k^* be the iteration at which the convergence is attained. Since $k^* \ll \tau$, for transient tissue movements ($k \leq k^*$), the last term in (14) can be ignored. In this case we have $cd = \alpha g ed$, (see Fig. 13), where $cd \triangleq cp_k - cp_0$ and $ed \triangleq ep_k - ep_0$ are the cell displacement and the electrode displacement, respectively. Under the assumption that

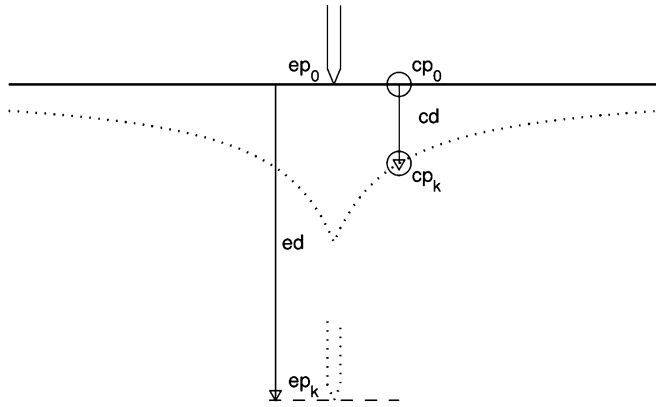


Fig. 13. Initial position (solid line) and the position at iteration k (dotted line) of the electrode and the cell. The cells farther from the electrode path undergo smaller displacement.

the cells closer to the electrode path are affected by the electrode displacement more than the cells that are further away, we write

$$\alpha = \frac{d}{d+r}$$

where r is the distance between the center of the soma and the electrode track, and d is a suitably chosen constant. For example, $\alpha = 1$ if the center of the soma lies directly in the electrode path, and α approaches 0 for the cells that are far away from the electrode track (see Fig. 13). The rate of the decay of α is regulated by the constant d ($d = 10 \mu\text{m}$ for this study). The gain factor $g \in (0, 1)$ determines what fraction of the electrode displacement translates into the cell displacement ($g = 0.9$ for the present study). For example, the displacement of the electrode by $100 \mu\text{m}$, causes the displacement of $30 \mu\text{m}$ of the cell whose soma center is located $20 \mu\text{m}$ from the electrode path.

For relaxation movements ($k > k^*$), we have $cp_k = cp_0 + cd^* \exp(-k/\tau)$, where cd^* is a constant vector defined as $cd^* = \alpha g(ep_{k^*} - ep_0)$. For very large k , the tissue relaxes back to its original position, i.e., $cp_k \rightarrow cp_0$. For the present study we chose $\tau = 300$, which given the fact that a single iteration takes ~ 6 s (2 s-simulation, 4 s-processing), translates into a time constant of 30 min.

ACKNOWLEDGMENT

The authors would like to thank D. Meeker and E. Branchaud for their generous help in providing experimental data. The authors would also like to acknowledge the anonymous reviewers and the editorial staff of this transactions.

REFERENCES

- [1] 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," *J. Neurophysiol.*, vol. 93, no. 1, pp. 570–579, 2005.
- [2] W. Rall, "Electrophysiology of a dendritic neuron model," *Biophys. J.*, vol. 2, pp. 145–167, 1962.
- [3] C. M. Gray, P. E. Maldonado, M. Wilson, and B. McNaughton, "Tetrodes markedly improve the reliability and yield of multiple single-unit isolation from multi-unit recordings in cat striate cortex," *J. Neurosci. Meth.*, vol. 63, no. 1–2, pp. 43–54, 1995.

- [4] S. Baker, N. Philbin, R. Spinks, E. Pinches, D. Wolpert, D. MacManus, Q. Pauluis, and R. Lemon, "Multiple single unit recording in the cortex of monkeys using independently moveable microelectrodes," *J. Neurosci. Meth.*, vol. 94, no. 1, pp. 5–17, 1999.
- [5] P. J. G. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proc. IEEE*, vol. 77, no. 1, pp. 81–98, Jan. 1989.
- [6] Z. Mainen and T. Sejnowski, "Influence of dendritic structure on firing pattern in model neocortical neurons," *Nature*, vol. 382, pp. 363–366, 1996.
- [7] M. Hines and N. Carnevale, "The neuron simulation environment," *Neural Comput.*, vol. 9, pp. 1179–1209, 1997.
- [8] Z. F. Mainen, J. Joerges, J. R. Huguenard, and T. J. Sejnowski, "A model of spike initiation in neocortical pyramidal neurons," *Neuron*, vol. 15, no. 8, pp. 1427–1439, 1995.
- [9] G. R. Holt and C. Koch, "Electrical interactions via the extracellular potential near cell bodies," *J. Comp. Neurosci.*, vol. 6, pp. 169–184, 1999.
- [10] G. Holt, "A critical reexamination of some assumptions and implications of cable theory in neurobiology," Ph.D. dissertation, California Inst. Technol., Pasadena, 1998.
- [11] R. Lemon, *Methods for Neuronal Recording in Conscious Animal*. New York: Wiley, 1984, ch. 2, pp. 17–38.
- [12] M. S. Fee, P. P. Mitra, and D. Kleinfeld, "Variability of extracellular spike waveforms of cortical neurons," *J. Neurophysiol.*, vol. 69, pp. 175–188, 1996.
- [13] M. Sahani, "Latent variable models for neural data analysis," Ph.D. dissertation, California Inst. Technol., Pasadena, 1999.
- [14] M. Lewicki, "Bayesian modeling and classification of neural signals," *Neural Comput.*, vol. 6, pp. 1005–1030, 1994.
- [15] M. S. Keshner, "1/f noise," *Proc. IEEE*, vol. 70, pp. 212–218, 1982.
- [16] B. B. Mandelbrot and J. W. van Ness, "Fractional brownian motions, fractional noises and applications," *SIAM Rev.*, vol. 10, no. 4, pp. 422–437, 1968.
- [17] P. Flandrin, "Wavelet analysis and synthesis of fractional brownian motion," *IEEE Trans. Inf. Theory*, vol. 38, no. 2, pp. 910–917, Feb. 1992.
- [18] G. Wornell, "Wavelet-based representation for the 1/f family of fractal processes," *Proc. IEEE*, vol. 81, no. 10, pp. 1428–1450, Oct. 1993.
- [19] K. Frank and M. C. Becker, "Microelectrodes for recording and stimulation," in *Physical Techniques in Biological Research*, W. L. Natus, Ed. New York: Academic Press, 1964, vol. V, ch. 2, pp. 22–87.
- [20] D. Humphrey, *Electrophysiological Techniques*. Atlanta: Society for Neuroscience, 1979.
- [21] Z. Nenadic and J. W. Burdick, "Spike detection using the continuous wavelet transform," *IEEE Trans. Biomed. Eng.*, vol. 52, no. 1, pp. 74–87, Jan. 2005.
- [22] J. Welsh and C. Schwarz, "Multielectrode recording from the cerebellum," in *Methods for Neural Ensemble Recordings*, M. Nicolelis, Ed. Boca Raton, FL: CRC, 1999, ch. 5, pp. 79–100.
- [23] M. Abeles and M. Goldstein, "Multispike train analysis," *Proc. IEEE*, vol. 65, no. 5, pp. 762–773, May 1977.
- [24] I. Bankman, K. Johnson, and W. Schneider, "Optimal detection, classification, and superposition resolution in neural waveform recordings," *IEEE Trans. Biomed. Eng.*, vol. 40, no. 8, pp. 836–841, Aug. 1993.
- [25] K. Harris, D. A. Henze, J. Csicsvari, H. Hirase, and G. Buzsaki, "Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements," *J. Neurophysiol.*, vol. 84, pp. 401–414, 2000.
- [26] J. Ward, "Hierarchical grouping to optimize an objective function," *J. Am. Statist. Assoc.*, vol. 58, pp. 236–244, 1963.
- [27] J. Hartigan, *Clustering Algorithms*. New York: Wiley, 1975.
- [28] G. McLachlan and K. Basford, *Mixture Models: Inference and Applications to Clustering*. New York: Marcel Dekker, 1988.
- [29] C. Fraley and A. Raftery, "How many clusters? which clustering method? answers via model-based cluster analysis," *Comput. J.*, vol. 41, pp. 578–588, 1998.
- [30] C. Fraley and A. Raftery, *Model-Based Clustering, Discriminant Analysis, and Density Estimation* Univ. Washington, Dept. Statist., Oct. 2000, Tech. Rep. 380 [Online]. Available: citeseer.nj.nec.com/fraley00modelbased.html
- [31] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. Roy. Statist. Soc., B: Meth.*, vol. 39, no. 1, pp. 1–38, 1977.
- [32] G. Schwarz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, no. 2, pp. 461–464, 1978.
- [33] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression curve," *Ann. Math. Statist.*, vol. 23, pp. 462–466, 1952.
- [34] Y. Ermoliev and R. J.-B. Wets, "Stochastic programming, an introduction," in *Numerical Techniques for Stochastic Optimization*, Y. Ermoliev and R. J.-B. Wets, Eds. New York: Springer-Verlag, 1988, ch. 1, pp. 1–32.

- [35] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*. Cambridge, MA: The MIT Press, 1983, ch. 2, pp. 12–66.
- [36] D. A. Robinson, “The electrical properties of metal microelectrodes,” *Proc. IEEE*, vol. 56, no. 6, pp. 1065–1071, 1968.
- [37] K. D. Harris, H. Hirase, X. Leinekugel, D. A. Henze, and G. Buzsaki, “Temporal interaction between single spikes and complex spike bursts in hippocampal pyramidal cells,” *Neuron*, vol. 32, pp. 141–149, 2001.
- [38] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*. New York: John Wiley & Sons, 1973.
- [39] J. Bilmes, A Gentle Tutorial on the EM Algorithm and Its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models Univ. California Berkeley, Dept. Elect. Eng. and Comput. Sci., May 1997, Tech. Rep..
- [40] P. Cheeseman and J. Stutz, “Bayesian classification (autoclass): Theory and results,” in *Advances in Knowledge Discovery and Data Mining*, U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds. Menlo Park, CA: AAAI, 1996, pp. 153–180.
- [41] E. Jaynes, *Probability Theory: The Logic of Science*, G. L. Bretthorst, Ed. New York: Cambridge Univ. Press, 2003.
- [42] M. Stone, “Cross-validatory choice and assesment of statistical predictions,” *J. Roy. Statist. Soc. B*, vol. 36, pp. 111–147, 1974.
- [43] H. Akaike, “A new look at the statistical model identification,” *IEEE Trans. Automat. Contr.*, vol. AC-19, pp. 716–723, 1974.
- [44] G. L. Bretthorst, “Bayesian analysis. II. Signal detection and model selection,” *J. Magn. Reson.*, vol. 88, pp. 552–570, 1990.



Zoran Nenadic (M'06) received the Diploma in control engineering from the University of Belgrade, Belgrade, Serbia and Montenegro, in 1995 and the M.S. and D.Sc. degrees in systems science and mathematics from Washington University, St. Louis, MO, in 1998 and 2001, respectively.

From 2001 to 2005, he was a Postdoctoral Fellow with the Division of Engineering and Applied Science at the California Institute of Technology, Pasadena. Since 2005, he has been with the Department of Biomedical Engineering, University of

California, Irvine, where he is currently an Assistant Professor. His research interests are in the area of adaptive biomedical signal processing, control algorithms for biomedical devices, brain-machine interfaces, and modeling and analysis of biological neural networks.



Joel W. Burdick received the B.S. degrees in mechanical engineering and chemistry from Duke University, Chapel Hill, NC, in 1981, and the M.S. and Ph.D. degrees in mechanical engineering from Stanford University, Stanford, CA, in 1982 and 1988, respectively.

Since 1988, he has been with the Division of Engineering and Applied Science at the California Institute of Technology, Pasadena, where he is currently a Professor of Mechanical Engineering and Bioengineering. His research interests include

robotic locomotion, sensor-based robot motion planing, multifingered robotic grasping, neural prosthetics, and applied nonlinear control theory.

Prof. Burdick has been the recipient of the National Science Foundation Presidential Young Investigator award, the Office of Naval Research Young Investigator award, and the Feynman fellowship. He has also received the ASCIT award for excellence in undergraduate teaching and the GSA award for excellence in graduate student education.