Target Tracking Using Clustered Measurements, with Applications to Autonomous Brain–Machine Interfaces

Thesis by

Michael T. Wolf

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

2008

(Defended May 16, 2008)

To Stephanie

Acknowledgments

It gives me great pleasure to contemplate the many people whose work, advice, and support have helped make this thesis possible.

Foremost, I would like to thank my advisor, Dr. Joel Burdick. Through his exceptional mentorship and collaboration, Joel has provided me an extraordinary role model of a scholar and teacher. I appreciate also his unwavering, warmhearted focus on his students' best interests.

I thank the other thesis committee members, Dr. Richard Andersen, Dr. Jim Beck, Dr. Richard Andersen, and Dr. Pietro Perona, for their time and their expert technical counsel. Deserving particular mention is Dr. Andersen, whose welcoming of roboticists into his neurobiology lab has been essential to this work; without his collaboration, I would simply not have been able to explore the topic of this thesis.

The contributions in this thesis owe much to previous Burdick team members Jorge Cham, Eddie Branchaud, and Zoran Nenadic. Jorge, in particular, provided noteworthy design leadership on the electrode microdrive.

Thank you to the members of the Andersen Lab, especially Grant Mulliken, Zoltan Nadasdy, EunJung Hwang, Michael Campos, and Kelsie Pejsa, for educating me on neurophysiology, providing valuable test data and spike sorting expertise, and promptly addressing my many pesky questions.

I have shared many times of laughter, academic struggles, and Ernie's lunches with my friends in the Caltech community, making so much my past five years enjoyable (and the rest of the time at least "bearable"). The collaboration and friendship of Nicolas Hudson is especially appreciated.

The constancy of my parents' love and support has helped shape the person I've become. Moving back to LA and seeing them more frequently (though they'll tell you I haven't) has been a pleasure. I thank the rest of my family, particularly my parents-in-law, for their love and encouragement as well.

Topping the "love and support" category is my wife and partner, Stephanie, who has sacrificed much for my doctoral pursuits. One might consider it her own fault, though, as my efforts stem from the fact that she inspires me, by example, towards a principled and thoughtful life.

Abstract

This thesis presents new methods for classifying and tracking the signals of targets that produce clusters of observations, measured in successive recording intervals or scans. This multitarget tracking problem arises, for instance, in extracellular neural recordings, in which an electrode is inserted into the brain to detect the spikes of individual neurons. Since multiple active neurons may lie near the electrode, each detected spike must be assigned to the neuron that produced it, a task known as *spike sorting.* In the scenario considered in this thesis, the electrode signal is sampled over many brief recording intervals. In each recording interval, all spikes must first be *clustered* according to their generating neurons, and then each cluster must be associated to clusters from previous recording intervals, thus *tracking* the signals of putative neuron "targets."

This thesis introduces a novel multitarget tracking solution for the above problem, called *multiple hypothesis tracking for clusters* (MHTC). The MHTC algorithm has two main parts: a Bayesian clustering algorithm for associating observations to clusters in each interval and a probabilistic supervisory system that manages association hypotheses across intervals. The clustering procedure provides significantly more consistent results than previously available methods, enabling more accurate tracking of targets over time. Such consistency is promoted by a maximum *a posteriori* (MAP) approach to optimizing a Gaussian mixture model via expectation-maximization (EM), in which information from the preceding intervals serves as a prior for the current interval while still allowing the number and locations of targets to change. MHTC's hypothesis management system, like that of traditional multiple hypothesis tracking (MHT) algorithms, propagates various possibilities for how to assign measurements to existing targets and uses a delayed decision-making logic to resolve data association ambiguities. It also, however, maintains several options, termed *model hypotheses*, for how to cluster the observations of each interval. This combination of clustering and tracking in a single solution enables MHTC to robustly maintain the identities of cluster-producing targets in challenging recording scenarios.

In addition to these classification and tracking techniques, this thesis presents advances in a miniature robotic electrode microdrive capable of extracellular recordings lasting for days at a time. As a whole, these contributions can play an important role in enabling an autonomous neural interface, which, by frequent automatic repositioning of its recording electrodes, can optimize the recording quality of extracellular signals associated with individual neurons and maintain high quality recordings for long periods of time. Such autonomous movable electrodes may eventually overcome key barriers to engineering a practical neuroprosthetic device and, in the near term, can significantly improve state-of-the-art neuroscience experimental procedures.

Contents

List of Figures

List of Tables

Notation

Clustering	
i	indexes spike observations from the current interval
g	indexes cluster (mixture component), current interval
j	indexes cluster (mixture component), previous interval
k	indexes interval (discrete time step)
m	indexes model class
T_k	recording interval
Δ	duration of recording interval
y_i	spike observation in feature space
d	dimension of feature space
d_{w}	dimension of full waveform space
N	number of spike observations in current interval
Y^k	set of spike observations on kth interval
$Y^{1:k}$	set of all spike observations from intervals 1 to k
$\mathcal{M}_m$	model class
$\bar{M}$	number of candidate model classes
η_m	number of parameters in $\mathcal{M}_m$
$\mathcal{C}_g$	gth cluster in current interval
n_g	number of spike observations in cluster $\mathcal{C}_g$
G_m	number of clusters (mixture model components) in $\mathcal{M}_m$
$G_{\max}$	maximal number of clusters considered by any model class
$\hat{G}^{k-1}$	number of clusters identified on previous time step
Θ_m^k	parameter set for mixture model class $\mathcal{M}_m$
π_g^k	mixture component weight
θ_g^k	mixture component density parameters
μ_g^k	mixture component mean
Σ_g^k	mixture component covariance matrix
λ^k	covariance volume
z_{ig}	spike–cluster membership indicators
Z	set of all spike–cluster membership indicators (current interval)
ζ_{gj}	cluster–neuron association indicators
$\mathcal{Z}$	set of all cluster–neuron association indicators (current interval)
ω_j^k	mixture component weight in prior
ψ_j^k	mixture component density parameters in prior
S_j^{k-1}	covariance matrix of prediction that prior mean is in same location
V	observation volume
α	forget factor
$\mathbf{H}_m$	Hessian matrix

Hypotheses	
k	indexes interval or scan (discrete time step)
l	indexes hypothesis
$\rho(l)$	index of parent global hypothesis of l
$m(l)$	index of parent model hypothesis of l
g	indexes measurements of a current interval
j	indexes targets in parent hypothesis
$\mathcal{M}_m$	model hypothesis
h_l	data association hypothesis
H_l^k	joint hypothesis (particular model and data association)
$H_l^{1:k}$	global hypothesis through first k intervals
Ω^k	all global hypotheses and observations at time k
τ_l	set of assignments of measurements to existing targets under h_l
ν_l	set of measurements identified as new targets under h_l
ϕ_l	set of measurements identified as false clusters under h_l
$\delta_{j,l}$	indicator of whether jth target is tracked under h_l
N_t	number of active targets (tracks) in parent hypothesis
N_τ	number of measurements from existing targets (under h_l)
N_ν	number of measurements from new targets (under h_l)
N_ϕ	number of measurements from false clusters (under h_l)
$\hat{G}_m$	number of measurements in mth model hypothesis
$\mathsf{P}_{\mathsf{d},j}$	probability of detection of jth target
λ_ν	Poisson rate of new targets
λ_ϕ	Poisson rate of false clusters
β	minimum threshold for model class to become model hypothesis
K_{miss}	number of missed detections before target is deleted
$\mathbf{A}$	data association matrix
a_{gj}	elements of data association matrix
$\mathbf{A}^*$	linear assignment cost matrix

Dynamical Model and State Estimation	
x_j^k	state vector
$\hat{\mu}_j^k$	measurement (cluster mean)
F^k	state transition matrix
H^k	measurement matrix
v_j^k	process noise
w_j^k	measurement noise
Q_j^k	process noise covariance matrix
R_j^k	measurement noise covariance matrix
$x_j^{k\mid k-1}$	state prediction
$\Lambda_j^{k\mid k-1}$	state prediction covariance
$\hat{\mu}_j^{k\mid k-1}$	measurement prediction
S_j^k	innovation covariance
K_j^k	Kalman gain
$x_j^{k\mid k}$	updated state estimate
$\Lambda_j^{k\mid k}$	updated state estimate covariance

Chapter 1

Introduction

This thesis presents new methods for classifying and tracking the signals of individual neurons in extracellular neural recordings, as well as advances in a novel miniature robotic mechanism capable of obtaining such recordings. These contributions will play a pivotal role in enabling an autonomous neural interface, which, by frequent automatic repositioning of its recording electrodes, optimizes the extracellular signals associated with individual neurons and maintains high-quality recordings for long periods of time. Such autonomous movable electrodes may eventually overcome key barriers to engineering a practical neuroprosthetic device and, in the near term, can significantly improve state-of-the-art neuroscience experimental procedures. The remainder of this chapter provides further motivation for this work, its problem statements and technical context, and an overview of the contributions of subsequent chapters.

1.1 Motivation

1.1.1 Autonomous Neural Interfaces for Neuroprostheses

Recent progress in neuroscience has provided hope that paralyzed people may someday use thoughts to control electromechanical devices such as robotic limbs to partially restore lost motor function [1–4]. Such a device, termed a *neuroprosthesis*, could benefit patients with little other opportunity for physically affecting their environment, such as those with severe spinal cord lesions or trauma, neurodegenerative diseases such as amyotrophic lateral sclerosis (ALS), or stroke to motor cortex, as well as those who have lost limbs. To accomplish this goal, a neuroprosthesis must perform three distinct functions (Figure 1.1): obtain useful neural signals from the brain, decode the user's intentions from these signals, and control a mechanism (e.g., an actuated prosthetic arm) that carries out these intentions. These same requirements apply to a brain–computer interface (BCI) and other

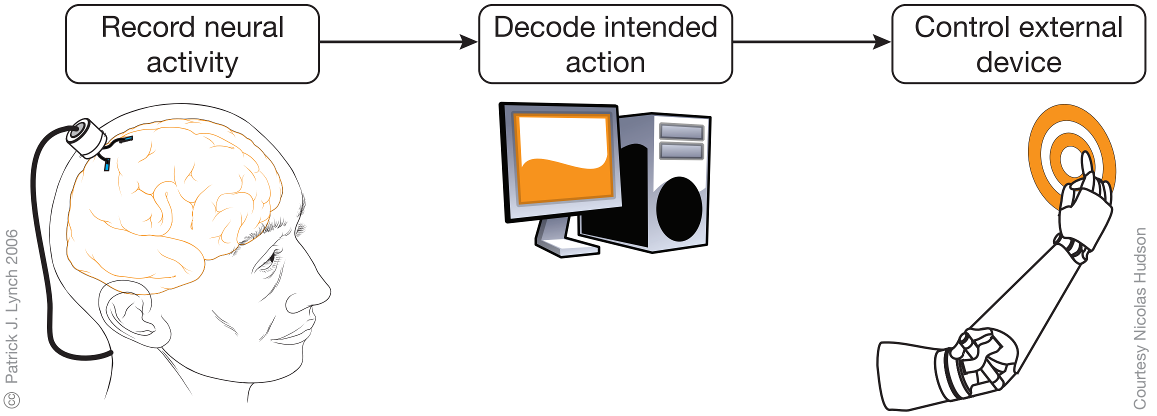

Figure 1.1: Principal technical functions of a neuroprosthesis.

brain–machine interfaces (BMI)[1].

While neuroprostheses have been demonstrated in several research labs, a critical barrier to future practical neuroprosthetic devices now lies in the signal acquisition [5]. Scientifically, the most desirable source of neural signals to control a neuroprosthesis, especially one with many degrees of freedom, is the activity of individual neurons [1–4], or *single units*, which is obtained via *extracellular recording*. This activity is measured by microelectrodes inserted into the brain; for prostheses patients, a robust *neural interface* with many electrodes must be surgically implanted in the brain region(s) of interest and must operate for years at a time[2]. The goal of single-unit extracellular recording, more thoroughly described in Chapter 2, 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. It is widely accepted that the information output of a neuron is encoded in the timing of its spiking activity, not in the shape of its spike waveforms, which are highly stereotyped (see Figure 1.2). A successful extracellular recording, then, is one in which the firing of spikes of individual neurons can be reliably detected and differentiated from other signal sources; the neurons are then considered *isolated*.

The timing of these spikes may then be analyzed for scientific studies or for control of a neuroprosthesis, decoding the intentions of a paralyzed user. In some brain regions, a strong correlation may be observed between the activity of a neuron—often quantified as a *firing rate* of the number of spikes over a short time interval (e.g., ~ 80 ms)—and the subject's concurrent or subsequent

[1] These terms are sometimes used interchangeably. In this thesis, a *brain–machine interface (BMI)* is broad term for a device that interacts with neural signals; a *neuroprosthesis* will refer to a type of BMI that uses a brain-controlled mechanism to replace lost motor function (although the term is often used for sensory prostheses such as cochlear implants and motor prostheses attached to the peripheral nervous system); and the phrase *brain–computer interface (BCI)* applies when the objective is to control a computer (e.g., a cursor on a screen) rather than a physically moving mechanism.

[2] Non-invasive recording techniques such as EEG, which records brain waves from electrodes placed on the scalp, can provide estimates of gross brain activity summed over large cortical regions. While these signals may be useful for simpler BMIs or BCIs, they are hindered by lacking the specificity that is likely required for high-fidelity control of a many degree-of-freedom manipulator [6].

physical motion. Often, this correlation relates to a certain direction of motion (or spatial location), in which case we say the neuron is *tuned* to that direction (or location). For operation of a BMI, a computer algorithm must first learn the tuning preferences of a collection of neurons during a "training phase." Subsequently, the user's intentions may be *decoded* from the observed firing rates of these same neurons, provided the number of tuned units recorded is sufficient for "coverage" of the physical region of interest. Thus, a successful neural interface must isolate many cells and maintain these isolations of these neurons so that their previously calibrated characteristics may be used for decoding commands.

Whether a cell is successfully isolated, and thus provides useful signals for the neuroprosthesis, relies almost entirely on the effective placement of the uninsulated electrode tip with respect to that cell body. In work related to this thesis, a robotic system has been proposed to autonomously position electrodes so as to initially optimize and then maintain the quality of the recorded signal over long periods of time [7–9]. This system consists of two parts: a mechanism, termed a *microdrive*, capable of independently positioning an electrode along a linear track with micron-scale precision, and a hierarchical control algorithm to determine appropriate electrode movement commands. In the algorithm's main loop, the electrode's signal is sampled for an interval of, say, 10 seconds, and then analyzed to calculate the optimal electrode position adjustment.

Ultimately, the goal of this project is to build an array of many (perhaps one hundred) independently actuated electrodes, each controlled by the autonomous positioning algorithm, all in a package small enough to allow surgical implantation. Such a device offers the potential to overcome many of the difficulties inherent in establishing the effective, long-lasting neural interfaces required for practical neuroprostheses. Additionally, a robotic electrode paradigm can increase the quality and efficiency of neuroscientific research techniques by eliminating the tedious manual process by which electrophysiologists have traditionally optimized electrode placement. (The current state of the autonomous electrode positioning system and a more detailed discussion of extracellular recording procedures and challenges are provided in Chapter 2.)

1.1.2 The Spike Clustering and Neuron Tracking Problems

During extracellular recording, a single electrode's signal may contain action potentials generated by multiple neurons lying near the electrode tip. Because the goal of extracellular recordings is to detect the activity of individual neurons, each detected spike must be associated to the neuron that produced it (its *generating neuron*)—a task known as *spike sorting.* Typically, spike sorting procedures classify the spikes according to waveform shape and amplitude; although the shape of action potentials are very similar across neurons, the techniques assume that the separation of waveforms generated by different neurons is sufficiently larger than the variability of recorded

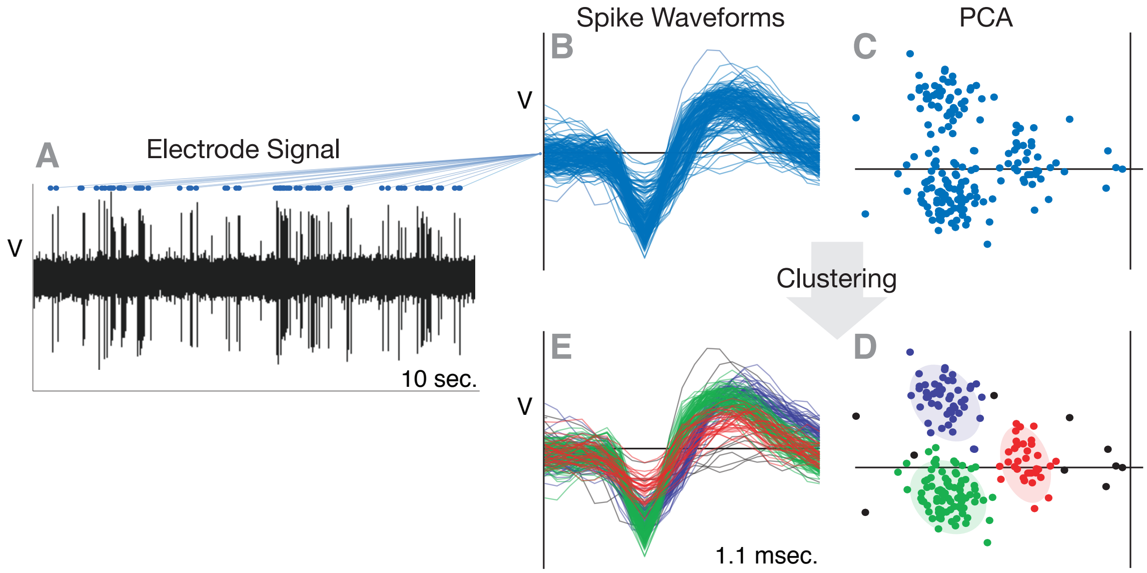

Figure 1.2: The spike clustering problem. (A) The electrode signal recorded over a particular interval T_k may contain the spike waveforms of multiple neurons. (B) The spikes are extracted from the recorded voltage trace and aligned by their minimum and then (C) projected onto a feature space (here, the first two principal components). (D) In this space, the spikes of disparate neurons are "clustered" into sets. As will be conventional throughout this thesis, clusters are indicated with color and filled 2-sigma ellipses; black points indicate classification as "outliers." Plot (E) shows the waveforms colored according to clustering results.

waveforms from the same neuron[3].

Accurate spike sorting is critical, as the metrics from the signals of each distinct neuron are vital both to electrode positioning, whose goal is to maximize signal quality, and to the scientific or prosthetic uses of the recorded data, which generally rely on estimated neuronal firing rates from the recordings. If spikes are incorrectly classified, these metrics may be severely corrupted. Because of its importance and difficulty, spike sorting is typically achieved through a largely manual process in neuroscience experiments, via visual examination of the spike waveforms. However, in the autonomous electrode positioning algorithm introduced above, and for practical neuroprostheses, spike sorting must be achieved in an unsupervised manner. Additionally, because the electrode signal is sampled over many brief, successive recording intervals, not only must spikes be associated to their generating neuron within a particular recording interval, but spikes from the same generating neuron must be associated with each other across recording intervals. Thus, the ability to track individual neurons over successive intervals is necessary for the algorithm to assess whether a change in electrode position has improved the signal quality of these neurons.

Specifically, consider the following problem, illustrated in Figure 1.2. An electrode signal $\mathcal{S}$ is sampled over an interval T_1 of length Δ. In a set of preprocessing steps, the spikes in $\mathcal{S}$ must be

[3] Note that the recorded signal is affected by the neurons' varying distances from the electrode tip and the inhomogeneities of surrounding tissue, so that even if action potentials of two neurons are identical, they may be distinguishable on the recorded signal.

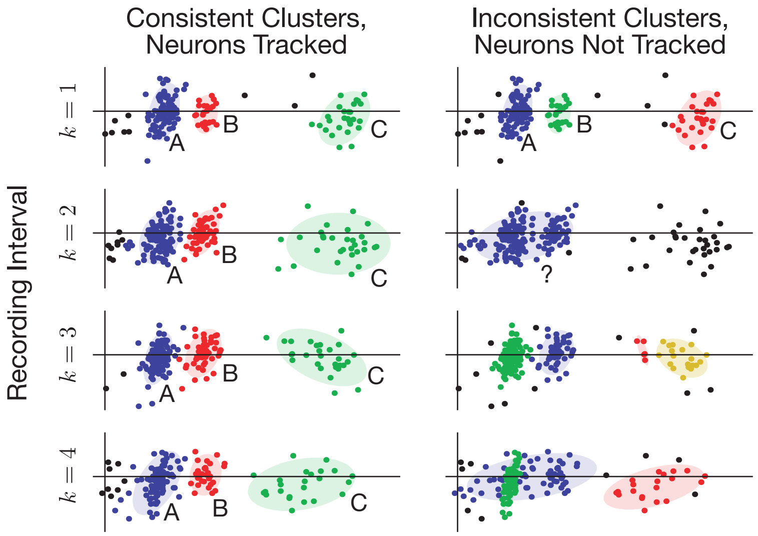

Figure 1.3: The neuron tracking problem. In each recording interval, the current clusters must be matched to those in the preceding interval, thus propagating the identities of and associating all the spikes of persisting neurons. Four consecutive recording intervals of clustered neuronal spikes in a common PCA basis are shown. In the left column, three clusters are consistently identified in each recording interval; because each cluster appears in approximately the same location as in the last interval, the clusters are associated across intervals and carry forward the neuron IDs A, B, and C. In the right column, different clusters have been identified; the inconsistency of these clusters from one interval to the next complicates the tracking task, and the associations to clusters in the preceding interval are unknown.

detected, extracted from the voltage trace, and then aligned so that their waveforms may be more readily compared. Then, the waveforms found in T_1 are projected onto an d-dimensional feature space (e.g., a 2-dimensional principal component (PCA) basis; see Section 2.2.2.2) so that each waveform is represented as a point. These points must then be grouped into sets via a *clustering* procedure, where each cluster is assumed to be associated with a unique neuron in the multi-unit signal.

Next, additional signal samples are taken across successive intervals T_2, T_3, and so on. After clustering results are computed for the interval T_k, each cluster in this interval must be associated to a neuron that was previously identified in $\{T_1, \ldots, T_{k-1}\}$, or identified as a newly appearing neuron not previously recorded. This *tracking* process, as illustrated by an example in Figure 1.3, must be robust to changes and variability in the numbers, alignments, shapes, and amplitudes of the neuronal signals over the recording. To reliably identify the signals of individual neurons across successive sampling intervals, the clustering procedure must not only provide high quality results within a sampling interval but also consistently identify similar clusters across sampling intervals so that tracking is feasible.

The majority of this thesis addresses the above spike clustering and neuron tracking problems, which are the critical challenges in the spike sorting process. (Previous work is leveraged for spike detection, waveform alignment, and feature space selection, and the techniques developed here remain compatible with many choices for these procedures.) To summarize, two data association challenges must be faced: *Classification* or *clustering* refers to the process of grouping spike observations from a single interval T_k into distinct sets, or clusters, effectively asserting that all the spikes in a single cluster have arisen from the same neuron. *Tracking* is the procedure of associating clusters to each other across recording intervals, identifying them as "belonging to" the same neuron—which, in turn, assigns all the spikes associated with these clusters to a common generating neuron.

1.1.3 Other Applications

Although this work is primarily motivated by the needs of autonomous electrode positioning systems, the spike sorting problem described above also arises in other electrophysiology applications. For example, during the training phase of some brain–machine interfaces, multi-unit signals from each electrode of an implanted static electrode array are sampled during repetitive execution of a task, which typically lasts a relatively short duration (e.g., $\Delta = \sim 5$ sec.). In order to properly estimate the tuning properties of the neurons sampled by the array, the signal sources must be sorted on each electrode and matched across each task execution. The neuronal properties learned during this training phase may then be used during task execution of the BMI, provided the same neurons may again be identified.

In multi-unit recordings gathered during basic electrophysiology experiments, automating the spike sorting task can relieve this time-consuming burden from experimenters and perhaps even improve the accuracy or the results, as manual sorting is known to be inconsistent [10]. For these applications, it can be useful to divide lengthy recordings into short time intervals for spike sorting and analysis, as the data are apt to be effectively stationary over these periods (see Section 2.3 and Section 3.1 for more on non-stationarity). Here again, the neuronal signals must be clustered in each analysis interval and then matched across intervals.

More generally, the fundamental clustering and tracking procedures addressed in this thesis are not specific to electrophysiological data. Thus, the solutions presented here may be applicable to other domains in which objects must be observed through probabilistically distributed groups of measurements and tracked over successive "scans" or measurement intervals. Such problem statements may occur in fields such as computer vision or other sensor processing disciplines.

1.2 Review of Existing Literature

This thesis builds upon work from several disparate domains: devices and algorithms for extracellular recording and electrode positioning; spike sorting and, more generally, classification methods; and target tracking. Existing literature from each domain is summarized below and reviewed more thoroughly in the chapters that follow.

While electrode microdrives have long been used for basic research in neurobiology, the system introduced in [7–9] represents the first efforts to fully automate the process of extracellular recording and in fact represents one of the first robots to operate autonomously within a primate brain for extended periods of time. Previous attempts at automating small portions of the neuron isolation process were reported by Fee [11], who demonstrated a method to stabilize intracellular recording electrodes for a period of a few minutes, and by Baker et. al [12], who demonstrated a control architecture for an acute microdrive that autonomously advances electrodes until target cells are detected, at which point a human operator must optimize the recorded signal.

The electrode positioning algorithm requires a microdrive mechanism compatible with the autonomous system paradigm and should be small enough for continuous use for days or even weeks at a time[4]. Commercially available motorized microdrives are much too large to be practical for such semi-chronic use and generally require a subject to be restrained for the experiment's duration. While chronic implantable microdrives have been developed [13–17], these devices require manual intervention to reposition the electrodes, such as turning lead screws. Muthuswamy et al. have developed micro-machined actuators for implantable movable electrodes and have demonstrated a prototype in an acute rat experiment [18]; however, it is unclear whether the high power consumption and limited actuator range of their device will be appropriate for chronic setups in primates. Also, an accompanying control algorithm would still be necessary, as it is not practical to require constant human supervision to adjust the electrodes to achieve optimal signals.

A rich body of literature has addressed unsupervised classification, and many traditional clustering procedures have been adapted to sort neural waveforms, including hierarchical [19], k-means [20,21], neural networks [22], superparamagnetic [23], template matching [24], and density grids [25]. The optimization of a (typically Gaussian) mixture model [26] has been shown to be a particularly effective approach in spike sorting [27–32]. However, most of these existing techniques are designed for offline batch processing of large data sets, and no existing technique specifically addresses the challenges of real-time processing of successive sampling intervals.

For such a recording scenario, the inconsistency of conventional clustering methods' output and the non-stationarity of the neuronal signals are the crucial issues, as each interval's spikes are clustered separately but must be matched to those in the preceding and subsequent interval(s) for neuron tracking. Bar-Hillel et al. [32] are, to the author's knowledge, the only others to explicitly address these matters, but present a non-causal, computationally intensive method designed for offline processing and hence not applicable to the real-time applications that motivate the work in this thesis. Other authors have also characterized and addressed signal non-stationarity for single intervals of long duration [33,34], but these methods are not designed for the short, separate intervals discussed

[4] The origin of these requirements and their relationship to the fully implantable system suggested in Section 1.1.1 is discussed in Chapter 2.

here.

Although existing spike sorting techniques have seldom addressed the tracking problem described in Section 1.1.2, an abundance of established *multitarget tracking* (MTT) literature exists, primarily intended for military and, more recently, computer vision applications (see [35] for a summary of techniques). Most of these methods assume measurements of targets of interest are obtained in successive "scans" of an observation volume, a scenario resembling the use of repeated sampling intervals to track neurons [36, 37]. The key difference for the neuron tracking problem is that measurements of the neuron's "position" are actually obtained through *groups* of observations (clusters of spikes) in every scan, whose associations are unknown[5]. In general, the tracking of objects observed by uncertain clusters of measurements is a novel problem addressed by this thesis.

Among the many target tracking techniques, a data association strategy called multiple hypothesis tracking (MHT), attributed to Reid [38], is generally agreed upon as the preferred solution [39] but is encumbered by computational infeasibility of the ideal implementation. From the perspectives of this thesis, much of the recent target tracking literature falls into two categories: efforts to formulate the MHT solution for a practical implementation even for large numbers of targets [40,41] and increasingly sophisticated methods for scenarios such as maneuvering targets, nonlinear target dynamics, or specific sensor types [37, 42]. The improvements in this latter category are generally not necessary for the neuron tracking application, as models for neuron "dynamics" are typically simpler and slower than those considered for modern tracking systems. Many techniques from the former category may be useful for neuron tracking, although the numbers of neurons, or agents to be tracked, are usually small compared to other target tracking applications.

1.3 Thesis Contributions and Organization

The remainder of this thesis is organized as follows. Chapter 2 provides further technical background to contextualize the thesis' contributions, specifically describing common techniques for extracellular recording and previous work on the autonomous electrode positioning system. Additionally, it further discusses the challenges inherent to extracellular recording to establish why the spike sorting task is difficult.

In Chapter 3, a novel clustering method is developed, capable of overcoming many of these challenges. Its strategy is based on the optimization of a Gaussian mixture model (GMM) via expectation-maximization (EM) [26,43]. Assuming that the analysis of the data in the interval T_{k-1} has yielded a reasonable clustering result, the model parameters estimated from interval T_{k-1} provide a Bayesian prior for the clustering of data in interval T_k. Thus, clustering is effected as a maximum *a*

[5] One may argue that, if only one spike for each neuron is observed at a time, the traditional tracking methods apply. However, determining a sampling interval length Δ that is likely to contain one spike from each surrounding neuron may be impossible.

posteriori (MAP) method rather than maximum likelihood (ML) method. Additionally, the model's statistics from the preceding interval provide initial values (or *seed clusters*) for the EM computation. Importantly, 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 it provides a simple neuron tracking solution "for free," as it quantifies the probability that a given cluster found in interval T_k is associated with a cluster found in interval T_{k-1}. A Bayesian technique for choosing the best mixture model class is embedded in the approach as well.

A more sophisticated and robust solution to the tracking problem is presented in Chapter 4, which incorporates the clustering procedure of Chapter 3 into a multiple hypothesis tracking (MHT) framework. This approach is novel in its combination of clustering and tracking into a single solution, tracking targets that may be observed only through collections of measurements from each recording interval. Because the associations of these measurements (i.e., the clustering) is uncertain, multiple "model hypotheses" for how an interval's data may be clustered are maintained as well as the standard data association hypotheses. The method, referred to as *multiple hypothesis tracking for clusters* (MHTC), fits naturally with the probabilistic theory and computations of the clustering method described in Chapter 3.

Chapter 5 addresses the hardware required for the autonomous electrode positioning system, reporting a novel electrode microdrive capable of semi-chronic use. This neural interface is the next generation prototype of the microdrive presented in [7], which was the first specifically designed for fully autonomous extracellular recordings. The advances in the robot described in this thesis provide substantial improvements in terms of signal quality, robustness to biological environments, experimental ease of use, and manufacturability. The microdrive is designed as a testbed for the autonomous electrode paradigm and as a means to develop the specifications for future miniaturized implantable devices. The current design is also immediately useful to the neuroscience research community for longer-term electrophyisology experiments that cannot be carried out with currently existing microdrives.

Finally, Chapter 6 summarizes the contributions of the thesis and suggests possible directions for future work.

Chapter 2

Background

This chapter provides context for the contributions of this thesis. First, since the thesis concerns the acquisition and analysis of extracellular neuronal signals, the principal techniques and issues of recording these signals are presented in Section 2.1, along with a description of the neurophysiological environment. Second, Section 2.2 presents previous work on the autonomous electrode positioning algorithm, in which the clustering and tracking methods are designed to operate and which controls the electrode microdrive. Finally, several challenges that make spike sorting a difficult task are discussed in Section 2.3.

2.1 Extracellular Recording: Environment and Techniques

The electrical impulses known as action potentials or spikes are the primary means of information processing and transmission in the nervous system. Each nerve cell, or neuron, accepts input principally via its *dendritic tree*, a set of branched projections that receive incoming signals from other cells across connections known as *synapses*. This electrochemical stimulation is transmitted to the neuron's *soma*, or cell body, and may trigger the neuron to fire an action potential of its own, which will propagate out along its *axon*, a slender projection that carries the action potential toward downstream neurons. It is believed that the connectivity of neurons determines how information is processed and stored, and that particular neurons have particular functions or associations to particular memories, sensory processes, motor commands, etc. [44].

The goal of *extracellular single-unit recording* is to detect the spikes of individual neurons[1]. Scientific experimenters may then examine the timing (e.g., firing rate) of each unit's action potentials to infer, for example, the role of the neurons in a particular brain region or the connectivity of larger brain structures. When used for brain–machine interfaces (BMIs), as described in Section 1.1.1, the

[1] Other useful signals may also be obtained from extracellular electrodes. The *local field potential (LFP)*, for example, comprises the activity of hundreds or thousands of neurons around an electrode and has shown value in controlling a neuroprosthesis [6]. Some investigators have also studied *multi-unit activity*, the unsorted action potentials from a small set of neurons at the electrode tip, due to the practical issues that make the signals of the single units difficult to discern.

neurons' *tuning* may first be learned and then used to guide the BMI.

These extracellular recordings are made by inserting electrodes, typically sharpened metal wires insulated along their length and exposed at the tip[2], into neural tissue to measure the electrochemical disturbance in the extracellular medium caused by a neuron's action potentials[3]. The tip of a recording electrode must generally lie within about 50 μm of the neuron's soma to be able to discriminate these disturbances, which are usually 100 μV or less, above the background of gross neural activity and measurement noise. This requirement defines a "listening sphere" around the neuron (see Figure 2.1C), and a closer proximity may be required to sufficiently distinguish the signals of different neurons [46, 47]. Each neuron cell body is approximately 10–50 μm wide and generally the signals of a maximum of about four units may be discernible on the electrode's signal at any given time (though the actual cell density within 50 μm may be significantly greater) [9, 48]. As summarized below, extracellular recordings can be carried out in an *acute* or in a *chronic* manner. The autonomous electrode positioning system introduced in Section 1.1.1, and thus the contributions of this thesis, can benefit both types of extracellular recordings.

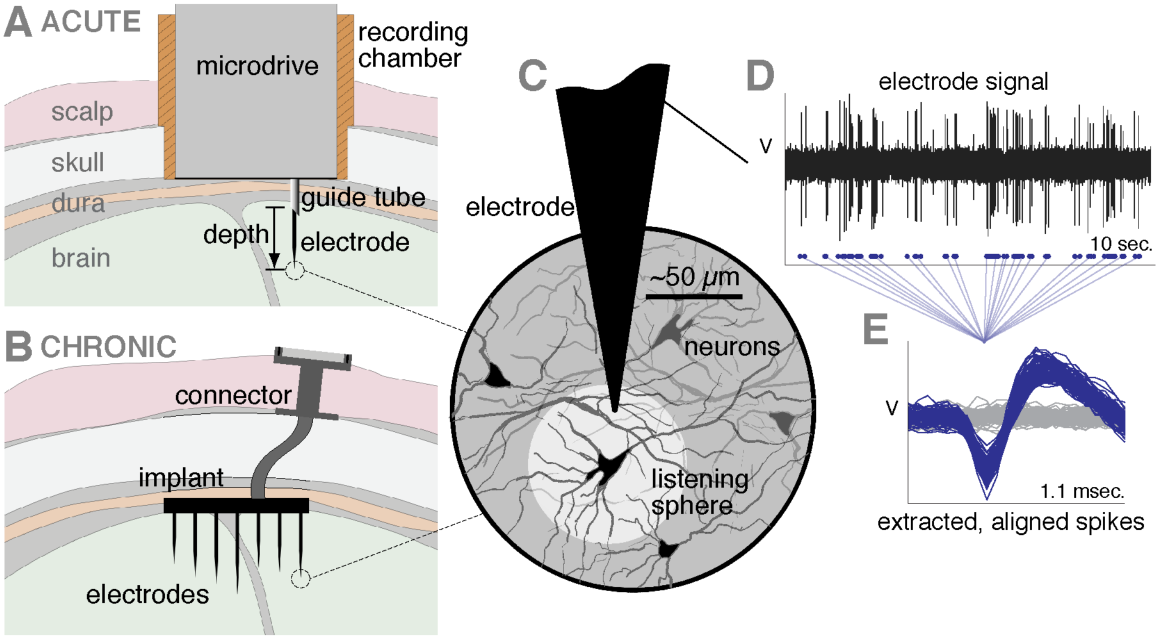

Figure 2.1: Extracellular recording environment and example signals: cross-sectional diagrams of (A) acute and (B) chronic recording setups; (C) detail drawing of recording site at electrode tip; (D) 10-second filtered signal sample from an electrode, with (E) the action potential (spike) waveforms extracted from the recording and aligned by their minimum.

[2] Silicon shafts with electrically active recording sites along their shanks may also be used [45].

[3] This technique may be contrasted to *intracellular recording*, in which the electrode is placed inside a neuron to measure the voltage across the cell membrane. Intracellular recordings are difficult to achieve *in vivo* and not applicable to brain–machine interface applications, as the nerve cells penetrated for intracellular recording typically die within a few hours.

2.1.1 Acute Recordings

In acute recordings, which are primarily used for scientific research, electrodes are inserted and removed from the neural tissue during each recording session (which typically lasts a few hours). To enable these recordings in cortex, a portion of the skull over the brain region of interest is typically removed and replaced with a sealable cranial *recording chamber* (see Figure 2.1A); for example, a 16-mm inner-diameter cylindrical recording chamber is a standard used in the neuroscience community. During an acute recording session, a microdrive[4], affixed to the opened chamber, is used to lower one or more electrodes into cortical tissue and then subsequently finely position the electrodes. Electrodes are advanced through neural tissue along a straight line, with the position of each electrode described by its *depth* along this linear track. Note that linear movement of the electrode through tissue substantially reduces the amount of tissue damage relative to possible curvilinear motion of the electrode tip; if one wishes to interrogate a 3-dimensional volume of neural tissue, multi-electrode devices are employed in an arrayed geometry.

While the electrode movement is typically motorized, the electrode's motion is at present manually determined by the experimenter. The process of determining the exact position of each electrode is commonly guided by the use of visual (oscilloscope) and auditory (loudspeaker) representations of the voltage signal, and the experimenter relies on experience and intuition to determine proper electrode placement. The electrode must be close enough to the neuron for a high quality recording, yet far enough away to avoid damaging it. During the course of a typical experiment, the experimenter must monitor the electrode and often reposition it to account for tissue decompression effects. Sorting the spikes of different neurons may be achieved manually in real time using commercial software aids or may be deferred for later offline processing of the entire recording session. The process of isolating and maintaining high quality neuronal signals thus consumes a significant amount of the experimenter's time and focus.

Simultaneous recordings with many electrodes are becoming an increasingly important technique for understanding how local networks of neurons process information, as well as how computations are coordinated across multiple brain areas. Commercial microdrives with sixteen or more electrodes are now available [12]. As the number of electrodes increases, the manual task of positioning each electrode to maintain a high quality neuronal signal becomes intractable for a single experimenter. Data collection in these experiments is essentially limited by how many channels an experimenter can effectively monitor—most experimenters agree that about three or four electrodes is the maximum that can be juggled effectively by an experienced electrophysiologist. Thus, by continually monitoring the signal and automating the process of placing and repositioning electrodes, an autonomous system

[4] Recall that a microdrive is an electromechanical device that can position an electrode along a linear track with micron-scale precision. The device itself may be quite large. Commercial microdrives are offered, for example, from Thomas Recording GmbH, Germany; FHC Inc., USA; Narishige Inc., Japan; NAN Instruments LTD, Israel; etc. A photo with several microdrives is shown in Figure 5.1.

can significantly improve the efficiency and quality of acute multi-electrode studies.

2.1.2 Chronic Recordings

In chronic recordings, multi-electrode assemblies with fixed geometry, which typically consist of bundles of thin wires or arrays of silicon probes, are surgically implanted in the region of interest [45, 49–51] (see Figure 2.1B) and remain in place for weeks, months, or possibly years at a time. Such chronic implants enable investigations of larger populations of neurons and can be used as the front end of a neuroprosthesis or for longer-term scientific studies.

Current chronic recording technology suffers from a number of limitations. The implant's signal yield (the percentage of the array's electrodes that can record a useful signal) depends largely upon the luck of the initial surgical placement. Because it is generally impossible for all of the implanted electrode tips to fall within the "listening sphere" of an active neuron, not all of the implanted electrodes will provide a useful signal. Moreover, blood pressure variations, breathing, and small mechanical shocks can cause migration of the electrodes in the tissue, leading to further degradation of the signal [11, 52]. Finally, reactive gliosis can encapsulate the electrode, diminishing signal quality over time [53]. All of these effects conspire to limit the usefulness and practical longevity of chronically implanted electrode arrays.

A chronic array whose electrodes can be continually repositioned after implantation may overcome many of these limitations. With such an implant, the overall signal yield can be improved by moving the electrodes to optimal neuronal recording sites. Further, neurons whose activity is well correlated with the objectives of the neuroprosthesis could be specifically sought, thus providing more information per electrode channel than static arrays. Moving electrodes may also enable recording in brain regions that are not easily accessible, such as those within a cortical sulcus (a fissure resulting from the folded nature of the cerebral cortex).

Commercially available motorized microdrives are much too large to be practical for chronic use and generally require a subject to be restrained for the experiment's duration. While chronic implantable microdrives have been developed [13–17], these devices require manual intervention to reposition the electrodes, such as turning lead screws. Muthuswamy et al. have developed implantable actuated electrodes and have demonstrated a prototype in an acute rat experiment [18]. However, it is unclear whether the high power consumption and limited actuator range of their device will be appropriate for chronic placement in primate brains. Also, an accompanying control algorithm with automated spike sorting and electrode positioning would still be necessary, as it is not practical to require constant human supervision to adjust the electrodes to achieve optimal signals.

The algorithms and experimental demonstrations described in the next section provide the foundation for future generations of chronic "smart" implantable multi-electrode systems. Although new actuator technologies will ultimately be required to sufficiently reduce its size for chronic use (see [54] for initial attempts at developing miniaturized, biocompatible, actuated electrodes that would enable a compact, implantable implementation of this approach), the small size of the microdrive described in Chapter 5 allows it to serve as a testbed for these future devices.

2.2 Autonomous Electrode Positioning Algorithm

This section summarizes prior relevant work aimed at creating a control algorithm for autonomous electrode positioning in extracellular recordings [7–9]. The control algorithm utilizes a hierarchical closed loop approach to determine, based on the recorded signal and the electrode's position history, the best depth for each electrode. The goal is to place each electrode so that the spikes from an *isolated* neuron can be unambiguously detected in the noisy voltage recording and discriminated from the signals of other nearby neurons. This section presents the control system structure and then describes its individual components in more detail. Because each electrode is moved independently, only the processing steps for a single electrode need to be considered; these steps are run in parallel for each electrode in a multi-electrode microdrive.

2.2.1 Control System Structure

The control algorithm operates in a cycle, illustrated in Figure 2.2. Let these cycles be indexed by the integer k, $k = 1, 2, \ldots$. The cycle begins with sampling the electrode signal over a short sampling interval (denoted by T_k, which is typically of duration 10–20 seconds) while the electrode is stationary, followed by analysis of this signal to determine if and how the electrode should be repositioned, and ending with the movement of the electrode to a new position (if necessary).

A hierarchical control algorithm determines the electrode movement commands. The inner-most loop of this algorithm (Section 2.2.3) attempts to isolate an individual neuron by optimizing the quality of the recorded signal via small local movements of the electrode tip, assuming that the tip is close enough to a neuron for the isolation process to be possible. The outer control structure consists of a finite state machine supervisory controller (Section 2.2.4) which has several purposes. First, it manages the neuron isolation process: It moves the electrode until a region of sufficiently strong neuronal signal sources is found and then further searches this region to acquire the information needed to initiate the isolation procedure. Additionally, the supervisory system handles several of the complicating realities of the extracellular recording process. Of course, to provide useful neuronal signal metrics from the electrode recording, these algorithms require several signal processing steps, which most critically include spike sorting, motivating Chapters 3–4 of this thesis. Most of these processing tasks have traditionally been performed manually in electrophysiology experiments; producing automated and unsupervised methods presents significant challenges in addition to those in determining proper electrode position based on those methods' output.

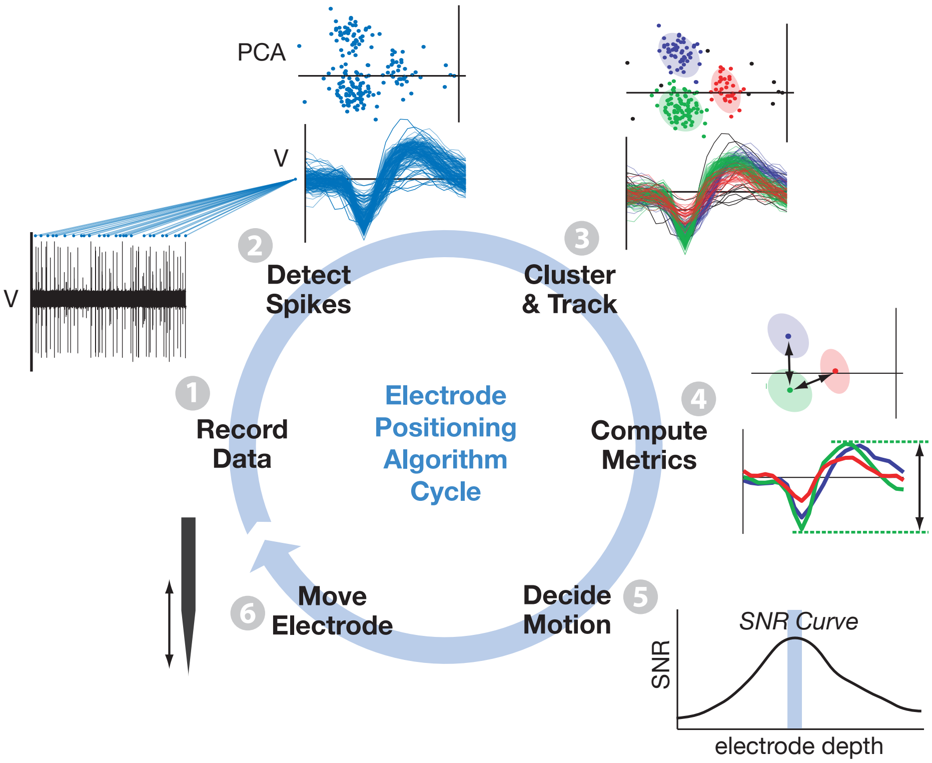

Figure 2.2: Autonomous electrode positioning algorithm cycle. Plots show key data involved at each step of the cycle. During the kth cycle: (1) A short data sample (voltage trace) is recorded during interval T_k, from which (2) spike waveforms are detected, extracted, and aligned. (3) Using their PCA representations, these spikes are clustered by their generating neurons and associated with the neurons recorded on the previous cycle. (4) SNR and IQM metrics are computed and then (5) used to determine the electrode motion commands to optimize the SNR curve. (6) Finally, the electrode is moved to its commanded position.

2.2.2 Signal Processing and Metrics

2.2.2.1 Spike Detection

The first step of unsupervised signal processing on the electrode's recorded voltage trace is *spike detection*, which identifies the action potential events in the raw electrode signal of interval T_k. We employ a wavelet-based method developed by Nenadic and Burdick [55] specifically designed for this application. By projecting the electrode signal onto a specially designed wavelet basis, spike-like waveforms can be detected in the raw signal, and short intervals (∼1.1 ms in length) of the signal centered on the putative spike occurrence are extracted (see Figure 2.1D and E). All of the spike-like waveforms found during T_k are temporally aligned by their waveform minima in preparation for the next steps.

2.2.2.2 PCA and Other Feature Spaces

If each extracted waveform contains d_{w} voltage samples (e.g., $d_{\mathsf{w}} = 23$ for a 1.1 ms interval sampled at 20 kHz), then each waveform x_i may be considered a vector in d_{w} dimensions (i.e., $x_i \in \mathbb{R}^{d_{\mathsf{w}}}$). Drastically reducing the dimension of the spike waveform representations to $d \ll d_{\mathsf{w}}$ dimensions is computationally preferred for most spike sorting procedures, and this step may often be accomplished without losing much of the discriminability information contained in the waveforms. Dimensionality reduction is accomplished by selecting highly informative *features* of the waveforms and using them as the basis in which spike sorting operates; an early feature space, for example, consisted of waveform amplitude and a measure of its width.

The use of a 2-dimensional principal component analysis (PCA) basis is common practice in spike sorting [27]. Let $\{w_j\}$ be the eigenvectors of the sample covariance matrix[5] of all waveforms $\{x_i\}$ detected in interval T_k, and let these eigenvectors be ordered from greatest to least eigenvalue λ_j. Then the first d eigenvectors (called the *principal components* or *PC*s) form the d-dimensional PCA basis; the feature space spike representation may be calculated by $y_i = W^T x_i$, where $W = [w_1\, w_2\, \ldots\, w_d]$ [56]. Geometrically, the first principal component is the direction of largest variance in $\{x_i\}$, the second PC is the direction orthogonal to the first PC with largest variance, and so on; the features $\{y_i\}$ are the projections of $\{x_i\}$ onto the space defined by the PCs. PCA representations typically capture 70% or more of the spike waveform variance. Several other choices of feature space are possible and are later discussed as an area of future investigation.

2.2.2.3 Spike Clustering and Neuron Tracking

The spike sorting task of clustering the spikes according to their generating neuron and tracking the neurons across successive recording intervals comes next. This is the primary topic of Chapters 3–4 of this thesis. Before the contributions presented in those chapters were developed, the control algorithm was tested with a clustering technique based on maximum likelihood (ML) optimization of a Gaussian mixture model [57]. No attempt to explicitly track the neurons across sampling intervals was implemented.

2.2.2.4 Signal and Isolation Quality Metrics

After the spikes have been processed as above, two signal metrics are calculated for the neurons identified in interval T_k:

- A *signal quality metric* (SQM) determines the general quality of the extracellular signals associated with a particular neuron.

[5] The sample covariance of the points $\{x_i\}_{i=1}^N$ is $\Sigma = \frac{1}{N-1}\sum_{i=1}^N (x_i - \bar{x})(x_i - \bar{x})^T$, where $\bar{x} = \frac{1}{N}\sum_{i=1}^N x_i$ is the sample mean.

- An *isolation quality metric* (IQM) measures the separation of one neuron's waveforms from those of other neuronal signals that appear in the same recording interval.

The SQM is the algorithm's main target, and the *dominant* neuron is chosen as the one whose signals have the highest average SQM. This is the neuron whose signal is to be ostensibly optimized by the electrode's movements. The signal-to-noise ratio (SNR) will here be assumed to be the signal quality metric, although other choices of SQM are possible (see [9] for examples). In this application, the SNR is defined as the mean peak-to-peak amplitude of the neuron's waveforms detected in T_k divided by the RMS amplitude of a spike-free noise sample taken during interval T_k.

Because a neuron's signal is only valuable if it can be distinguished from those of surrounding neurons, the IQM provides a measure of "isolation" of the waveforms of the dominant neuron from other detected spikes. The IQM is based on the *isolation distance* (ID) [58], which, for cluster C_g containing N_g spike samples, is defined as the Mahalanobis distance between its center μ_g and the ${N_g}^{\text{th}}$ closest spike not in cluster C_g (denoted by y_j):

$$\text{ID}_g = \sqrt{(y_j - \mu_g)^T \Sigma_g^{-1} (y_j - \mu_g)} \ .$$

That is, the ID is the radius of the smallest ellipse (with shape defined by Σ_g) containing all the spikes in cluster C_g and an equal number of spikes not in cluster C_g (in effect, a measure of the "moat" around cluster C_g). In practice, the noise sample observations are included here as well, handling the case when one neuron has generated more than half of the spikes during T_k. Note that the SQM is calculated from the spike waveforms, while the IQM is computed in feature space (PCA basis). A thorough discussion of different neuronal signal metrics and their uses is documented in [9].

2.2.3 The Isolation Control Loop

Assume for now that in the current interval T_k the signal from the dominant neuron is sufficiently strong. Based on the processed neural data and the quality metrics just defined, the isolation control loop determines if repositioning the electrode can improve the signal quality of the dominant neuron. In the idealized scenario where the dominant neuron's signals may be consistently tracked from one recording interval to the next, and are ever present, the algorithm commands the electrode motion solely to increase the SNR of that neuron as outlined below.

Detailed computational models [8] of the extracellular field generated around a typical cortical pyramidal neuron show that when the electrode tip is within the "listening sphere" of a neuron, the variation of the neuronal signal's SNR with respect to electrode position traces out a unimodal curve, dubbed the *SNR curve* (see diagram in Figure 2.2, step 5). Let u denote the position of the electrode tip along its linear track. Let $R(u)$ denote the SNR curve. The goal is to find the peak of this curve and then maintain the electrode position sufficiently close to this peak. Because neural signals are highly noisy, the metric R should be considered a random variable with an associated regression function $M(u) = E[R \,|\, u]$, where $E[\,\cdot\,|\,\cdot\,]$ denotes conditional expectation. This regression function is *a priori* unknown, except that it has a unimodal shape. Only noisy observations of the SNR, obtained via the preprocessing steps summarized above, are available. In order to optimize the SNR using only the available noisy samples, the isolation process adaptively estimates the regression function (the smoothed SNR curve), and the electrode's movements are chosen to seek the extremal point of the adaptively evolving SNR curve.

The regression function model $M(u)$ is assumed to be a linear combination of basis functions: $M(u, m_k, B_k) = \sum_{i=1}^{m_k} b_{i,k}\, \psi_i(u)$, where m_k is the number of basis functions employed during cycle k and $B_k = [b_{1,k}, b_{2,k}, \ldots, b_{m_k,k}]^T$ are the corresponding expansion coefficients. The model parameters B_k and model complexity m_k must be estimated from the SNR observations and adaptively updated as new data become available. For a given model estimate, the electrode's next position, u_{k+1}, is determined as:

$$u_{k+1} = u_k + C\, |H_k|^{-1}\, \xi_k \ , \tag{2.1}$$

where $C > 0$ is an appropriately chosen scale factor and ξ_k and H_k are respectively the estimates of the first and second derivatives of the regression function at the electrode's current position, u_k. Note that Eq. (2.1) represents a stochastic version of Newton's method. Convergence of the electrode position to the maximum of the SNR curve is considered attained at iteration k^* if $C|H_{k^*}|^{-1}\xi_{k^*} < \epsilon$, where ϵ is a tolerance chosen by the user. The position u_{k^*} is then declared the optimal electrode placement, whereupon the finite state machine supervisory controller transitions to a "maintain" mode (see Section 2.2.4). The regression function $M(u)$, from which ξ_k and H_k are determined, is estimated as follows.

While many basis function choices are possible, polynomial bases can sufficiently capture the geometry of unimodal SNR curves (see [8]) and greatly simplify the estimation process. For polynomial bases, the regression function after k iterations is

$$\hat{M}(u, m_k, B_k) = \sum_{i=1}^{m_k} b_{i,k}\, u^{(i-1)} \ .$$

Let $\{u_1,\, u_2, \ldots,\, u_k\}$ be a sequence of (electrode) positions with the corresponding SNR samples denoted $\mathcal{R}_{1:k} = \{\mathbf{r}(u_1),\, \mathbf{r}(u_2), \ldots,\, \mathbf{r}(u_k)\}$. At each electrode location u_j ($j = 1,\, 2, \ldots,\, k$), multiple observations of SNR have been taken (one for each isolated neuronal waveform), i.e., $\mathbf{r}(u_j) = \left[r_1(u_j),\, r_2(u_j), \ldots,\, r_{n_j}(u_j)\right]^T$, where n_j is the total number of observations at u_j (this number may vary across sampling intervals).

Determining the "correct" number of basis functions, m_k, amounts to *model selection* problem. Given a family of candidate models $\{\hat{M}(u, m_k, B_k) \,:\, m_k = 1, 2, \ldots, m_{\max}\}$, the goal is to select the order of the model that is most probable in view of the data $\mathcal{R}_{1:k}$ and any prior information, I. The probability of the model $\hat{M}_{m_k}$ given $\mathcal{R}_{1:k}$ and I follows from Bayes' theorem

$$P(\hat{M}_{m_k} \,|\, \mathcal{R}_{1:k}, I) = \frac{p(\mathcal{R}_{1:k} \,|\, \hat{M}_{m_k}, I) P(\hat{M}_{m_k} \,|\, I)}{p(\mathcal{R}_{1:k} \,|\, I)} \qquad m_k = 1,\, 2, \ldots,\, N_{\max}, \tag{2.2}$$

where $\hat{M}_{m_k}$ is short for $\hat{M}(u, m_k, B_k)$ with fixed m_k. Here, I represents the model selection result obtained in the previous interval T_{k-1}—the posterior $P(\hat{M}_{m_{k-1}} \,|\, \mathcal{R}_{1:k-1}, I)$ calculated at iteration $(k-1)$ can be used as the prior at iteration k in Eq. (2.2). The recursion is initialized with a uniform prior density $P(\hat{M}_{m_{k_0}} \,|\, I) = \frac{1}{m_{\max}}$, where k_0 denotes the smallest admissible number of iterations, below which there is an insufficient amount of data to reliably model the regression function. The model order is chosen to maximize the posterior probability (2.2), i.e.,

$$m_k^* = \arg \max_{1 \le m_k \le m_{\max}} P(\hat{M}_{m_k} \,|\, \mathcal{R}_{1:k}, I) \qquad k = k_0, k_0 + 1, \ldots\,.$$

To perform this maximization, the posterior $P(\hat{M}_{m_k} \,|\, \mathcal{R}_{1:k}, I)$ of each candidate model $\hat{M}_{m_k}$ must be evaluated by marginalizing the unknown parameters B_k. With a Gaussian noise assumption and polynomial bases, the marginalization of B_k can be performed analytically [8].

Once the optimal model order m_k^* at time k is known, the parameters of the model $\hat{M}(u, m_k^*, B_k)$ are estimated by a linear least-squares method:

$$B_k^* = \arg \min_{B_k} \left\{ \sum_{j=1}^{k} \|\Psi_{j,k}\, B_k - \mathbf{r}(u_j)\|^2 \right\} \qquad k = k_0,\, k_0 + 1, \ldots\,,$$

where the matrix $\Psi_{j,k} \in \mathbb{R}^{n_j \times m_k^*}$ consists of n_j identical rows given by $[1,\, u_j,\, \cdots,\, u_j^{(m_k^*-1)}]$. Once the optimal parameters B_k^* are estimated, the optimal model $\hat{M}_k^*(u) \equiv \hat{M}(u, m_k^*, B_k^*)$ at iteration k is fully specified. From this result the gradient and Hessian of the optimal model are then used in Eq. (2.1) to determine the electrode movement.

Because sudden large electrode movements are unacceptable, the maximum step size is limited by a constant $\Delta_{\max}$, chosen before the experiment. This is especially useful for iterations where the optimal model is found to be a straight line ($m_k^* = 2$), which results in $H_k = 0$ and infinitely large step size in Eq. (2.1). Likewise, if for some $k > k_0$ we obtain $\hat{M}^k(u) = b_{1,k}^*$, i.e., $m_k^* = 1$, then $\xi_k = 0$ and the recursion (2.1) breaks. In this case the algorithm uses a simple control strategy:

$$u_{k+1} = u_k + \Delta_{\text{sample}}\ ,$$

where Δ_{sample} is a constant.

2.2.4 Finite State Machine Supervisory Controller

To manage the basic neuron isolation process, while also accounting for many additional challenges of practical extracellular recording, a finite state machine architecture guides the overall electrode movement process. This system is termed the *supervisory finite state machine* (SFSM). During each algorithm cycle, the electrode movement decision (immediately following the signal acquisition and analysis steps) depends on the current state of the SFSM, with individual states and state transitions crafted to guide behavior appropriate to seeking and isolating neurons. A prototypical pathway of state transitions is described below to describe the most common issues and SFSM operation; for more details, see Branchaud's thesis [9].

When electrodes are first lowered into neural tissue, the electrode tip may not lie in electrically active tissue. The SFSM initiates in the *Spike Search* state (see numbered states in Figure 2.3), whose goal is to find an electrically active tissue region. In this state, the electrode moves in increments of Δ_{search} (~ 20 μm) until a sufficient number of spikes are detected in interval T_k (according to a minimum firing rate set before the experiment), at which point the SFSM transitions to the *Gradient Search* state. The *Gradient Search* state seeks to determine if a viable SNR curve can be constructed. Observations of the SNR are made at regular intervals of Δ_{sample} (~ 10 μm) until k_0 observations are completed (typically, $k_0 = 3$ is used), at which point the optimization procedure of Section 2.2.3 determines the most likely order m_k^* that fits the SNR observations. As described above, if $m_k^* = 1$ the electrode continues in steps of Δ_{sample} (the SFSM stays in *Gradient Search*). If $m_k^* > 1$, indicating that a potentially viable SNR curve has been found (i.e., there is a high likelihood that a nearby neuron can be isolated), the SFSM transitions to *Isolate Neuron*.

As long as the SFSM remains in *Isolate Neuron*, the algorithm described in Section 2.2.3 operates, updating the SNR curve with the new observations and moving the electrode toward the estimated maximum. When the maximum of the SNR curve is reached, the SFSM state transitions to *Neuron Isolated*, but only if certain IQM conditions are also met (see below).

In *Neuron Isolated*, the electrode generally remains stationary while the SNR is continually monitored over successive sampling intervals. Often, the dominant neuron will drift away from the electrode, causing a decrease in the SNR. When the SNR drops below a percentage (typically 85%) of its value at the original isolation, the SFSM transitions to *Re-Estimate Gradient* in an attempt to reposition the electrode to maintain the high quality isolation. In the *Re-Estimate Gradient* state, the electrode is moved in increments of Δ_{resample} (~ 5 μm) to find a new gradient now that the dominant neuron has likely drifted away. In this state, the electrode is retracted, as the most common neuron drift is due to tissue decompression and is directed up towards the electrode. Once a new gradient is found, a transition is made to *Re-Isolate Neuron*, where the optimization procedure is again used to isolate the neuron. If, at any time in the *Re-Estimate Gradient* or *Re-Isolate Neuron* states, the SNR reaches or exceeds the SNR value obtained during the original *Neuron Isolated*

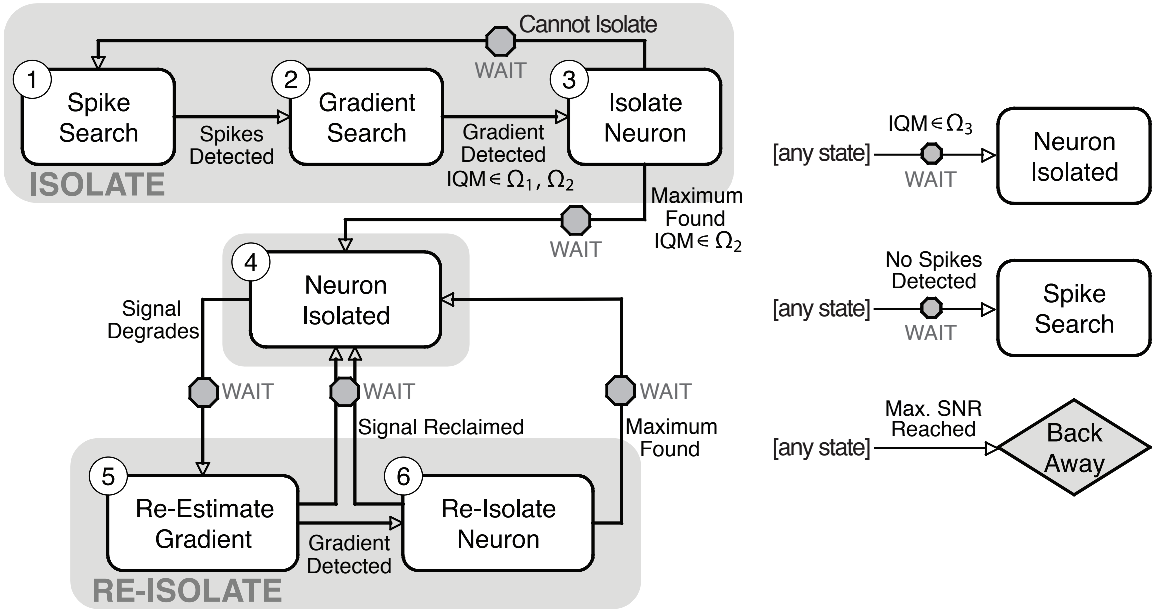

Figure 2.3: The supervisory finite state machine (SFSM). Transition criteria are noted between states. States are grouped into three *modes* (Isolate, Isolated, and Re-Isolate) for convenience. Transitions with Wait must meet transition criteria in R consecutive cycles, reducing sensitivity to transients. Transitions on the right may be made from any state.

declaration, the *Neuron Isolated* state is restored.

The isolation quality metric (IQM) plays a strong role in governing the SFSM state transitions, summarized in Table 2.1. SNR is a good metric to indicate the overall signal strength and reliability but is insufficient for judging whether the signals of two neurons with similar SNR can be discriminated. Thus, IQM thresholds ensure that the signals of a particular dominant neuron are reasonably separable from other signals during and after SNR curve optimization. Also, a transition to *Neuron Isolated* will occur from *any* SFSM state when the IQM is very high (above γ_3), even though the SNR curve peak has not been reached. In this case, the neuron likely lies close to or on the electrode's path, and continued advancement to possibly improve the SNR is not worth the risk of damaging the neuron (and thus losing the signal).

Table 2.1: Key Intervals Isolation Quality Metric

Interval	Definition	Description
Ω_3	$\gamma_3 \leq \text{IQM}$	Neuron is well isolated; immediately stop and declare isolation as further movement may damage neuron.
Ω_2	$\gamma_2 \leq \text{IQM} < \gamma_3$	Neuron is acceptably isolated, if maximum of SNR curve is reached.
Ω_1	$\gamma_1 \leq \text{IQM} < \gamma_2$	Isolation quality is high enough to follow gradient, but not high enough for acceptable isolation.
Ω_0	$\text{IQM} < \gamma_1$	Isolation quality is too low for reliable measurements; do not follow gradient for stochastic optimization.

In addition to the problems described above, several other practical difficulties commonly arise in extracellular recording experiments, well known to practicing electrophysiologists but often difficult to characterize. Many of these issues can complicate the spike sorting task and are discussed in Section 2.3, including signal non-stationarity due to tissue decompression, intermittent neural activity, and false positives from the spike detector. For a more thorough discussion of how the SFSM is designed to cope with these and other recording challenges, such as possible cell death and hysteresis from electrode-tissue mechanical interactions, the interested reader is directed to [9].

2.3 Spike Sorting

Often, during extracellular recordings the electrode tip lies within the listening radius of several neurons. As previously described in Section 1.1.2, spike sorting is the task of associating the signals of disparate units to their generating neurons. Usually this classification procedure relies on small differences in the spike waveforms generated by different neurons. The sections below discuss the critical nature of accurate spike sorting within the electrode positioning algorithm and describe the difficulties inherent to this task.

2.3.1 Importance of Spike Sorting in the Control Algorithm

Within the context of the above electrode positioning algorithm, it is easy to see the importance of accurate spike clustering and neuron tracking, which are part of the signal processing steps of Section 2.2.2. First, the SNR and IQM are calculated based on the statistics of each separate cluster in each sampling interval T_k. Thus, the spikes in T_k must be correctly attributed to their generating neurons for these metrics to be accurate. Otherwise, all downstream decisions made by the algorithm may be based on corrupted metrics.

Second, the identity of the dominant neuron must be tracked across sampling intervals for the algorithm to assess appropriately how a change in electrode position affects the quality of the signal recorded from that electrode. From the perspective of the isolation procedure in Section 2.2.3, an SNR curve necessarily consists of measurements taken over several different sampling intervals—each point on the curve is the average SNR of the spikes in the cluster representing the dominant neuron at a certain electrode depth, and multiple electrode depths are required. Ideally, this SNR curve should depict the SNR trends related to a single neuron only. Thus, the SNR curve cannot be constructed without correctly associating clusters to generating neurons across sampling intervals.

Finally, whether the neural interface is used for neuroscience experiments or as a front end for a neuroprosthesis, the goal is usually to maintain a neuron isolation for as long as possible. (The success of a recording is often judged by the length of time a particular neuron's signals were discriminable, or "held.") Therefore, even if a high SNR signal exists in each sampling interval over an entire recording session, the algorithm should be able to state whether the neuron(s) at the end of the session are the same as the one(s) at the beginning, or at least be capable of giving measures of these events' probabilities.

In the implementation of the control algorithm that existed before the work of this thesis [9], the control algorithm used an SNR curve based on whichever cluster's SNR was the highest on the current sampling interval, with no explicit tracking procedure. This implicitly assumes that the dominant neuron of interval T_k will be the same as the dominant neuron of interval T_{k+1}, T_{k+2}, and so on. Exacerbating this issue is the inconsistency of the current clustering algorithm, which results in some corruption of the SNR metric. As a result, the SNR curve used by the pre-existing control algorithm can be erratic (with sudden jumps of SNR and multiple peaks) at times, instead of a smooth, unimodal curve of the ideal scenario. Taken together, the above issues illustrate the pivotal role of the clustering and tracking procedures within the control algorithm, motivating the work presented in Chapters 3–4.

2.3.2 Spike Sorting Challenges

Several factors complicate the spike sorting task within an autonomous control algorithm—and for extracellular recordings in general. The clustering and tracking procedures must possess robustness to these challenges, which arise both from the nature of neural data collection and from errors in upstream data processing. Mostly because of the type of difficulties listed here, spike sorting remains a manual process in most electrophysiology experiments. However, for the applications that motivate this thesis, spike sorting must be accomplished in a manner that is *autonomous* (without human intervention) and *unsupervised* (without training on known data).

Low SNR. Neural data are notoriously noisy and difficult to measure, often resulting in very low signal-to-noise ratios (SNR). Extracellular recordings must often detect action potentials with peak-to-peak (PTP) amplitudes as small as 50 μV at the electrode site, and this signal is obscured by both surrounding neuroelectrical activity and noise inherent to the measurement apparatus. As a result, spikes from disparate neurons may not form well-separated groups and therefore may be difficult to cluster accurately, especially considering that all action potentials share a similar shape to begin with.

Jitter. Imprecision in the waveform alignment step can also complicating clustering. For example, suppose the waveforms are to be aligned by their minimum. The true time instant of the action potential's minimum, however, is likely to lie between two electrode voltage observations, which are made at a finite sampling rate (e.g., 20 kHz). Thus, aligning spikes by the observed minima will add small shifts, known as *jitter*, to the waveforms, increasing the variability of spike shapes from the same neuron[6].

[6] Other alignment options may be used [59–61], and techniques exist to help reduce the effects of jitter [62].

False spike outliers. Spurious noise artifacts that may appear quite similar to spikes can be caused by, for example, electrode vibration from abrupt movement of the subject or electromagnetic fields emitted from the electrode microdrive motors. Spike detection procedures, which are typically variations of template matching and/or amplitude thresholding techniques, frequently produce false positives. Thus, the set of spikes examined for clustering usually includes samples of non-spike events, and the clustering method must identify these errors as *outliers* rather than assign them to a neuron.

Signal non-stationarity. A major issue in tracking neuronal signals over time is their *non-stationarity*—the mean waveform shape does not remain the same over time. Most signal non-stationarity is likely caused by changes in electrode position relative to the recorded neurons, which may have several sources:

- Most obviously, in cases where the electrode is positioned by a microdrive, a desired change in electrode position may be commanded by the control algorithm or human operator.

- A phenomenon known as *electrode drift* is known to occur even when the electrode is not commanded to move. Some of these gradual changes may have root in physiological activity such as blood pressure variations and breathing [11, 52], but the more noteworthy drifts arise from tissue decompression, which has at least two sources:

  – At the beginning of a scientific experiment, before the fragile electrodes are advanced into the brain, protective “guide tubes” (see Chapter 5) must puncture through the dura (a tough protective layer of tissue between the brain and skull). This process causes compression of brain tissue (up to several millimeters), and further tissue compression may occur as the electrodes travel to their desired depth. Once the electrodes stop moving, the tissue decompresses over the course of several hours.

  – Even after the effects of initial bulk tissue compression have subsided, smaller scale tissue drift can occur following electrode movements. Although the electrodes are narrow and sharp, evidence of some such drift has been observed after electrode movements of less than 100 μm [9].

- When the subject or patient moves, mechanical shocks can cause small re-positionings of implanted electrodes, leading to sudden appreciable changes in the neuronal signal [63].

For the short recording interval durations Δ considered in this thesis, non-stationarity manifests primarily as an obstacle to tracking, as the neurons are “moving targets.” However, when Δ is large, such signal non-stationarity can result in a “smear” of data points in feature space, making it difficult to discriminate clusters in a single recording interval.

Intermittent neural activity. Finally, the firing rates of individual neurons are highly variable, and cells may lapse into brief periods of inactivity during which no spikes are emitted. Such temporary "silence" may result in very few or no spikes being recorded from a particular neuron on a particular recording interval, adding another consideration with which the tracking procedure must contend.

Chapter 3

Bayesian Clustering over Successive Recording Intervals

Accurately clustering the spikes detected in a recording interval T_k is a fundamental requirement of the signal classification and tracking task motivated in the previous chapters. This chapter presents a Bayesian clustering method used to improve the consistency and accuracy of signal clusters across successive recording intervals. Section 3.1 reviews the clustering problem statement and existing spike sorting literature. Section 3.2 describes a classical clustering method based on expectation-maximization (EM) applied to a Gaussian mixture model, so that the subsequent extensions to this method can be more clearly delineated. Section 3.3 details the proposed method for sequential clustering based on Bayesian parameter estimation and model selection, while Section 3.4 discusses how output of this clustering process provides a measure of data association of clusters across intervals useful for tracking neurons. Although a more robust neuron tracker is presented in Chapter 4, a simple nearest neighbor tracker can be implemented using these data association metrics directly, and thus the procedure presented in this chapter may stand alone as a complete spike sorting solution. Applications of this method to neural recordings in macaque parietal cortex are presented in Section 3.5 and discussed in Section 3.6.

3.1 Clustering Context and Contribution

Recall that, starting at time t_1, a signal $\mathcal{S}$ is sampled from an extracellular electrode over an interval T_1 of duration Δ, and that the electrode tip may happen to be within the "listening sphere" of multiple neurons, causing the spiking activity of several neurons to appear in $\mathcal{S}$. After some preprocessing steps (the spikes in $\mathcal{S}$ are detected and temporally aligned) the spike waveforms found in T_1 are projected onto an d-dimensional feature space (e.g., a 2-dimensional principal component (PCA) basis) so that each waveform is represented as a point. These points must then be "clustered" into sets, each assumed to be associated with a unique neuron in the multi-unit signal. This process is repeated for succeeding sampling intervals $T_2 \equiv (t_2, t_2 + \Delta]$, $T_3 \equiv (t_3, t_3 + \Delta]$, and so on[1]. The goal is to accurately associate the spikes in each T_k to their generating neurons and then track the clusters representing individual neurons across successive sampling intervals (as well as to discover the appearance or disappearance of a neuron). Because autonomous microdrives and neural interfaces motivate this work, clustering must be unsupervised and applicable to on-line recording—or at least to small, real-time batches as described above.

As previously noted, unsupervised spike sorting has long been an important and difficult problem in the neuroscience community (see [27] for a review). Many traditional clustering techniques have been adapted for sorting neuronal spike waveforms, including hierarchical [19], k-means [20, 21], neural networks [22], superparamagnetic [23], template matching [24], and density grids [25]. The optimization of a (typically Gaussian) mixture model [26] has been shown to be a particularly effective approach in spike sorting [27–32] and is the foundation of the technique presented in this chapter.

Most existing techniques are designed for offline batch processing of a single large data set. The short, successive intervals inherent to the autonomous electrode positioning system complicate the clustering operation because fewer data points (spikes) are available to process, exacerbating issues of volatility and variations due to noise. This volatility often increases the *inconsistency* of the clustering results across intervals—that is, spikes sampled in neighboring intervals from the same neurons may produce drastically different clusters in the different intervals. This issue is illustrated in Figure 1.3, where a traditional unsupervised clustering method (an implementation of the one in Section 3.2) gives grossly inconsistent clustering results when applied to recordings in macaque parietal cortex.

Such inconsistency in clustering across intervals limits the effectiveness of automated probes and neural interfaces, as well as the scientific value of the data. When attempting to track neurons, clustering inconsistency significantly reduces the reliability with which clusters can be associated across consecutive intervals. For example, in Figure 1.3 it is difficult to accurately associate clusters from neighboring intervals to the same neuron because the clusters vary dramatically. This, in turn, eliminates the electrode positioning algorithm's ability to assess whether a change in electrode position has improved the signal quality of any of these neurons. These clustering errors also corrupt the data for any intended scientific use, and may affect neural interfaces that serve as the front end for BMIs, as each neuron must be identified through time so that their previously calibrated characteristics may be used for decoding commands.

Another drawback of many prevailing techniques is the assumption of stationary distributions of each neuron's waveforms. Several authors have shown that signals evolve over time due to electrode

[1] The lengths of successive sampling intervals do not need to be the same, nor must the intervals be adjacent. It is only assumed that Δ is sufficiently long to capture a nontrivial number of spikes; see Section 3.3.

drift and other causes, even without active electrode movement [32–34], as discussed in Section 2.3. In long recordings (large Δ), such signal non-stationarity can result in a "smear" of data points in feature space, making it difficult to discriminate clusters. This effect can be mitigated by breaking up long recordings into short time windows, over which waveforms are likely stationary [32, 34]. This approach, then, also requires a method to cluster the spikes in each time window and associate clusters across these intervals to the same neuron, while allowing for possible changes in each neuron's average waveform shape between intervals.

To address the above issues, this chapter presents a model-based clustering technique that increases clustering consistency across short successive time windows, designed to succeed even in low firing rates (few samples per cluster), low signal-to-noise ratio, poor cluster separability, and non-stationary waveforms. The key idea is to incorporate the available information over time to increase spike clustering consistency, using the clustering results from interval T_{k-1} to improve the clustering of the subsequent set of data sampled during interval T_k, etc.

This strategy is based on the optimization of a Gaussian mixture model (GMM) via expectation-maximization (EM) [26,43]. Assuming that the analysis of the data in the interval T_{k-1} has yielded a reasonable clustering result, the model parameters estimated from interval T_{k-1} provide a Bayesian prior for the clustering of data in interval T_k. Thus, clustering is effected as a maximum *a posteriori* (MAP) method rather than maximum likelihood (ML) method. Additionally, the model's statistics from the preceding interval provide the initial values (or *seed clusters*) for the EM computation. Importantly, the prior is implemented such that clustering will likely succeed even if the preceding results were incorrect or if different neurons' signals are recorded during the two intervals. Not only does this procedure provide more consistent clustering results, but it provides data association across recording intervals (neuron tracking) "for free," as it quantifies the probability that a given cluster found in interval T_k is associated with a cluster found in interval T_{k-1}. A Bayesian technique for choosing the best mixture model class is embedded in this approach as well.

Several authors have contributed refinements to mixture model optimization for spike sorting, validating the approach's importance in the field. The early work of Lewicki [28] implemented a Bayesian approach on the full waveform within a GMM, where the prior on the waveform is set by the user. Shoham et al. [29] focused on the non-Gaussianity of clusters, noting that the larger tails of the t-distribution may better model clusters under the effects of non-stationarity, but this adjustment is typically unnecessary for short recording intervals. Kim and Kim [30] use an EM-optimized GMM, suggesting a new method for feature extraction using projection pursuit based on negentropy maximization (PP/NEM) and a heuristic model selection technique meant to replace traditional information criteria such as AIC and BIC; the method presented in this chapter remains agnostic to choice of feature space and presents a more principled model selection technique that utilizes available prior information. Wood et al. [31] generate seed clusters for their GMM via spectral clustering and select their model based on a greedy optimization according to the resulting decoding accuracy of a neuroprosthesis; the seed cluster strategy presented in Section 3.3.4 should be more accurate than theirs because it incorporates prior information, and the model selection of Section 3.3.5 can operate in any setting, not just when used during the training session of a neuroprosthesis.

In general, none of the above algorithms integrate information over time to improve clustering accuracy and consistency like the method proposed here. Bar-Hillel et al. [32] are the only others to include neighboring clustering results, and the only others to address novel clustering and tracking procedures together. They divide a long recording into smaller time frames and cluster the spikes in each time frame via an EM-optimized GMM, using a repetitive "mixing" of solutions to generate seed clusters based on results from neighboring time frames. However, the EM implementation is still based on ML optimization rather than MAP optimization, and their non-causal, computationally intensive method is designed for offline processing and hence not applicable to the real-time applications that motivate this work.

The clustering method presented here remains compatible with a wide variety of techniques for the upstream processes of spike detection and feature selection, such as those proposed in above works[2]. In particular, several options for feature spaces have been recently been proposed, such as the PP/NEM strategy in [30] and those based on wavelet-based representations [21,23,64,65]. In this thesis, results are obtained and presented in a 2-dimensional PCA basis, as this feature space persists as a common practice in the neuroscience community, but the methods may be applied to data in any feature space where Gaussian distribution of a cluster's points is a reasonable assumption.

3.2 ML Optimization of Mixture Models via EM

The classical clustering technique based on ML optimization of a mixture model [26, 43] has been the basis for several spike sorting algorithms. An underlying assumption of this approach is that each neuron produces spikes whose waveform features vary according to a probability distribution, and thus each generating neuron may be represented as a component in the mixture model. For example, if the ith data point (spike observation in feature space) $y_i \in \mathbb{R}^d$ was generated by the gth neuron (belongs to component, or "cluster," $\mathcal{C}_g$) with associated distribution parameters θ_g, then it is governed by the probability density $p\big(y_i \,|\, i \in \mathcal{C}_g, \theta_g\big)$. If using a Gaussian PDF, denoted $f_{\mathcal{N}}$, the distribution parameters are the mean and covariance matrix, i.e., $\theta_g = \{\mu_g, \Sigma_g\}$, and the corresponding density is

$$p\big(y_i|i \in \mathcal{C}_g, \theta_g\big) = f_{\mathcal{N}}\big(y_i|\mu_g, \Sigma_g\big) \equiv \frac{1}{\sqrt{\det(2\pi\Sigma_g)}} \exp\left(-\frac{1}{2}(y_i - \mu_g)^T \Sigma_g^{-1}(y_i - \mu_g)\right) . \tag{3.1}$$

[2] The model selection method, a subordinate contribution of this chapter, also may be easily exchanged.

Including all N data points in the recording interval and all mixture components $g = 1, ..., G_m$, the *mixture likelihood*, $\mathcal{L}_M$, of the model parameters given the data is:

$$\mathcal{L}_M(\Theta_m) = p\big(Y|\Theta_m, \mathcal{M}_m\big) = \prod_{i=1}^{N} \sum_{g=1}^{G_m} \pi_g f_g\big(y_i|\theta_g\big) \, , \tag{3.2}$$

where:

- $Y = \{y_i\}_{i=1}^{N}$ is the set of all spike observations (as represented in feature space).
- $\mathcal{M}_m$ is the mth model class under consideration in the current recording interval, which dictates the model order G_m (i.e., the number of individual neurons contributing to the signal), the form of the gth probability density f_g (typically Gaussian), and the form of the model parameters Θ_m, which include θ_g and π_g.
- π_g is the mixture weight of component $\mathcal{C}_g$, i.e., the prior probability that an observed spike was generated by gth source neuron, with $\pi_g \geq 0$ and $\sum_{g=1}^{G_m} \pi_g = 1$.

If one knew the actual mixture parameters Θ_m that governed the above model, then each spike y_i could be assigned to the cluster $\mathcal{C}_g$ whose component likelihood $\pi_g f_g\big(y_i \mid \theta_g\big)$ is the greatest. However, no closed-form solution for the optimal mixture parameters exists using (3.2) without already knowing which component neuron generated each spike. Thus, the expectation-maximization (EM) algorithm [66] is typically applied to estimate the parameters, using the following technique. The data Y are considered "incomplete" and are augmented by Z, the set of component-label vectors $z_i = (z_{i1}, ..., z_{iG_m})$ that indicate spike membership to a particular cluster,

$$z_{ig} = \begin{cases} 1 & \text{if spike observation } y_i \text{ belongs to cluster } \mathcal{C}_g \\ 0 & \text{otherwise} \end{cases} .$$

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

$$l_{\text{CD}}\big(\Theta_m|Y, Z, \mathcal{M}_m\big) = \sum_{i=1}^{N} \sum_{g=1}^{G_m} z_{ig} \log\left[\pi_g f_g\big(y_i|\theta_g\big)\right] \, . \tag{3.3}$$

The EM algorithm iterates between an E-step to calculate the conditional expectation $\hat{z}_{ig} = E[z_{ig} \mid y_i, \hat{\Theta}_m] \in [0, 1]$ using the current parameter estimates[3], and an M-step to find the parameter estimates $\hat{\Theta}_m$ that maximize (3.3) given $\hat{z}_{ig}$. This iteration guarantees (under weak conditions) a monotonically increasing $\mathcal{L}_M(\Theta_m)$ (3.2) and is continued until a predetermined convergence criterion [26]. Thus, the algorithm produces locally optimal mixture parameters $\hat{\Theta}_m$, as well as the clustering of spikes by assigning each spike y_i to a component $\mathcal{C}_{g^*}$ via $g^* = \arg\max_g \hat{z}_{ig}$.

[3] The symbol "ˆ" will be used to denote an estimated quantity.

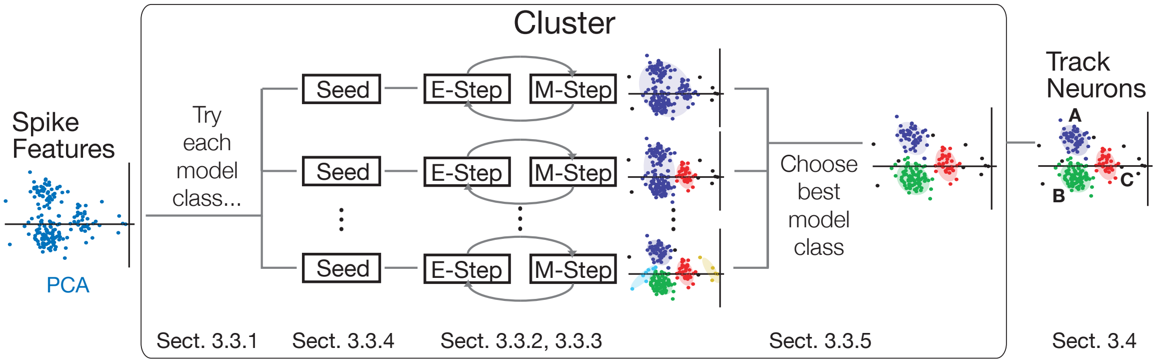

Figure 3.1: Structure of the clustering procedure for a single recording interval. Spikes are clustered using EM optimization over several possible model classes; the best model class selected. Then neurons are tracked by associating the clusters from the current interval to the previous interval.

3.3 MAP Clustering for Neuron Tracking

Figure 3.1 outlines the general flow of the spike sorting process, after the spikes recorded during interval T_k have been projected to feature space. Note that the EM iterations must be initialized by "seed clusters," or an initial guess of the data partitions (see Sect. 3.3.4). Also, the EM algorithm assumes that the model class—most importantly, the number of clusters G_m—is known *a priori*, but this is not feasible for spike sorting. A typical workaround is to apply EM to several model classes $\mathcal{M}_m$, $m = 1, ..., \bar{M}$, varying G_m among them, and then evaluating the results of each model to select the best.

The primary technical innovations in this chapter lie in four parts. First, the proposed procedure converts the EM algorithm to MAP optimization (rather than ML) of a GMM for the purpose of improved clustering and tracking throughout a recording session. Although MAP optimization via EM has previously been proposed for generic clustering cases (for example [43]), a mixture prior appropriate to spike sorting is explicitly derived, along with the resulting EM adjustments. Second, the method uses prior clustering results to provide appropriate seed clusters, thereby increasing the chances of avoiding poor local optima in the EM process. The process to generate good seeds of different model orders allows for phenomena commonly encountered in clustering neural data over time. Similarly, the model selection procedure incorporates information from the preceding interval while still admitting changes in the number of recorded neurons. Finally, the clustering solution inherently provides a simple tracking method to associate clusters over consecutive recording intervals.

Remark 3.1. A few considerations should be balanced when selecting the interval duration Δ. A short duration minimizes non-stationarity effects (which complicate the clustering task). However, a short duration may result in very few spike samples per cluster, which decreases the chance of proper clustering as well as the confidence of the cluster parameter estimates (though the MAP approach helps mitigate these issues). As a rule of thumb, for $d = 2$, it is best to aim for a minimum of 10–20 spikes per neuron in the interval (corresponding to an average firing rate as low as 1–2 Hz in the $\Delta = 10$ second duration used in the experiments discussed in Section 3.5). $\Diamond$

To begin the derivation of the MAP algorithm, first incorporate the sequential nature of the data sampling process to establish the Bayesian framework for parameter estimation (determining model parameter estimates $\hat{\Theta}_m$ and thus cluster membership Z) and model selection (determining the most appropriate number of clusters, $\hat{G}$). Let $Y^k = \{y_i\}_{i=1}^N$ denote all spike observations in the kth recording interval T_k and $Y^{1:k} = \{Y^1, ..., Y^k\}$ denote all data from the 1st through the kth recording intervals. The MAP parameter estimates can be naturally derived from Bayes' Rule:

$$p\big(\Theta_m^k | Y^{1:k}, \mathcal{M}_m\big) \propto \underbrace{p\big(Y^k | \Theta_m^k, \mathcal{M}_m\big)}_{\text{likelihood, Eq. (3.5)}} \underbrace{p\big(\Theta_m^k | Y^{1:k-1}, \mathcal{M}_m\big)}_{\text{prior, Eq. (3.10)}} \ , \tag{3.4}$$

where Θ_m^k denotes the mixture model parameters for the mth model during T_k and the likelihood's unnecessary conditioning on $Y^{1:k-1}$ has been suppressed.

3.3.1 Model Classes

Many model classes are possible within the framework used in this thesis. For simplicity, the following will be assumed for all sets of model classes under consideration:

- The set of model classes must allow for different possible numbers of neurons, $G_m = 1, ..., G_{\max}$, in the signal, since the number of neurons recorded during T_k is not known *a priori*.
- A Gaussian distribution, whose PDF is denoted $f_{\mathcal{N}}$, is used to account for the variability in each neuron's signals.
- In addition to the G_m Gaussian components in the mixture model, an *outlier* distribution is used to capture false positives of the spike detector (discussed in Section 2.3). The PDF of the outlier component, which is assigned component label $g = 0$, is traditionally either uniform or an origin-centered Gaussian with large covariance.

There are many parsimonious models of the covariance matrices Σ_g of Gaussian distributions. For example, Celeux and Govaert [67] parameterize the covariance matrix based on its eigenvalue decomposition $\Sigma_g = \lambda_g D_g A_g D_g^T$, with factors describing the volume (λ_g), shape (A_g), and orientation (D_g) of the corresponding constant-deviation ellipsoids. Some or all of these factors may be constrained to be equal across all clusters if a parsimonious model is desired.

Remark 3.2. The results in Section 3.5 and Section 4.5 employ the following model choices[4]:

[4] Although the focus here is on ensuring a range of model orders G_m is tested by evaluating different model classes, other model characteristics, such as the form of the component PDF f_0 or the model of the covariance matrices, may also be varied within the set of model classes used in the analysis of a single interval. The model selection procedure in Section 3.3.5 chooses the best, with a penalty for over-parameterization.

- The maximal number of neurons in an interval is assumed to be four, $G_{\max} = 4$.
- A shared-volume parsimonious covariance model is used rather than a fully variable model. This choice helps to avoid spurious clusters of very few spikes, which are especially damaging to the model in the degenerate case when a small set of (nearly) collinear points are clustered together. Thus, $\Sigma_g = \lambda\, C_g$, where $C_g \equiv D_g A_g D_g^T$.
- The outlier mixture component consists of an average of two distributions: (1) a Gaussian centered at the origin $f_{\mathcal{N}}\big(y_i\,|\,\mathbf{0},\Xi\big)$, whose isotropic covariance Ξ is scaled to be K times the covariance of a noise sample[5]; (2) a uniform distribution with magnitude $\frac{1}{V}$ defined over the rectangular volume of the data, $V = \prod_{j=1}^{d} \left(\max_i y_{i,j} - \min_i y_{i,j}\right)$. Thus,

  $$f_0\big(y_i\,|\,\theta_0\big) = \frac{1}{2}\left(f_{\mathcal{N}}\big(y_i\,|\,\mathbf{0},\Xi\big) + \frac{1}{V}\right) .$$

  This PDF models the tendency of the wavelet-based spike detection algorithm [55] to have a greater density of false positives near the origin (in PCA features), though some large amplitude outliers also occur. The parameter value $K = 4$ is chosen to sufficiently capture the near-origin outliers typical of the spike detector while not falsely classifying true spikes as outliers.

The method below may apply to many other choices of mixture models than the one defined above, and thus the development that follows remains general. ♢

Incorporating the outlier component, Gaussian clusters, and the parameters' time-dependence, the mixture likelihood (3.2) can be rewritten as

$$\begin{aligned} p\big(Y^k\,|\,\Theta_m^k, \mathcal{M}_m\big) &= \prod_{i=1}^{N}\sum_{g=0}^{G_m} \pi_g^k f_g\big(y_i\,|\,\theta_g^k\big) \\ &= \prod_{i=1}^{N}\left(\pi_0^k f_0\big(y_i\,|\,\theta_0^k\big) + \sum_{g=1}^{G_m} \pi_g^k f_{\mathcal{N}}\big(y_i\,|\,\mu_g^k, \Sigma_g^k\big)\right) , \end{aligned} \tag{3.5}$$

where θ_0^k contains the (constant) parameter(s) of the outlier distribution and $\pi_0^k = 1 - \sum_{g=1}^{G_m} \pi_g^k$ since the mixture weights must sum to unity. Thus, the set of independent parameters for the Gaussian mixture model is $\Theta_m^k = \{\mu_g^k, \Sigma_g^k, \pi_g^k\}_{g=1}^{G_m}$.

Remark 3.3. Two notational remarks are in order:

- Although the matrix Σ_g^k is treated as a single parameter for brevity, the parameter set Θ_m^k actually includes only the independent elements of the symmetric matrix, which will depend on the chosen parsimonious covariance model.

[5] The *noise* is defined by random extractions of the signal $\mathcal{S}$ that do not contain spikes; these extractions are projected to feature space for the covariance calculation.

- In most cases, the k-dependence of variables is suppressed when they will only be used to refer to the current recording interval, for example, $y_i \equiv y_i^k$, $V \equiv V^k$, $\mathcal{M}_m \equiv \mathcal{M}_m^k$, etc.

$\diamondsuit$

Incorporating hidden data Z as in Section 3.2, the complete data log-likelihood is similar to before:

$$l_{\mathrm{CD}}\big(\Theta_m^k | Y^k, Z, \mathcal{M}_m\big) = \sum_{i=1}^{N} \sum_{g=0}^{G_m} z_{ig} \log \left[\pi_g^k f_g\big(y_i | \theta_g^k\big)\right] \ . \tag{3.6}$$

3.3.2 Prior on Cluster Location

Next, construct an appropriate prior on the model parameters Θ_m^k based on the clustering results from interval T_{k-1}. The same prior will be used for all model classes under consideration in the current interval T_k. The model parameters are assumed to be independent across mixture components and across each parameter; therefore,

$$p\big(\Theta_m^k | Y^{1:k-1}, \mathcal{M}_m\big) = \prod_{g=1}^{G_m} p\big(\mu_g^k|\cdot\big)\, p\big(\Sigma_g^k|\cdot\big)\, p\big(\pi_g^k|\cdot\big)\,, \tag{3.7}$$

where the factors on the right are the prior probability densities of the respective mixture model parameters with the same conditioning as on the left-hand side.

Most important to the practical issue of cluster consistency and neuron tracking is the location of each cluster center, μ_g. Since the cluster covariance Σ_g and the mixture weight π_g associated with a given neuron may vary substantially across sampling intervals, diffuse priors may be implemented for these less informative model elements (though this method may be adjusted to use informative priors for these parameters as well). Figure 3.2 illustrates the use of previous cluster results for constructing priors on the means.

To establish priors on the cluster center locations, the gth cluster mean μ_g^k in T_k is sought near to *any* of the cluster centers found in T_{k-1}, without regard to which one, and thus a Gaussian mixture is used to represent all of the cluster means found in T_{k-1}. To allow for the possibility that $\mathcal{C}_g$ represents a *new* neuron that was not recorded in T_{k-1}, a uniform distribution component is included as well.

Definition 3.1. The ***mixture prior on a cluster mean*** is

$$\begin{aligned} p\big(\mu_g^k | Y^{1:k-1}, \mathcal{M}_m\big) &= \sum_{j=0}^{\hat{G}^{k-1}} \omega_j^k\, f_j\big(\mu_g^k | \psi_j^{k|k-1}\big) \\ &= \frac{\omega_0^k}{V} + \sum_{j=1}^{\hat{G}^{k-1}} \omega_j^k\, f_{\mathcal{N}}\big(\mu_g^k | \hat{\mu}_j^{k-1}, S_j^{k-1}\big)\,, \end{aligned} \tag{3.8}$$

where the zeroth component is uniform over the observation volume V and the remaining components

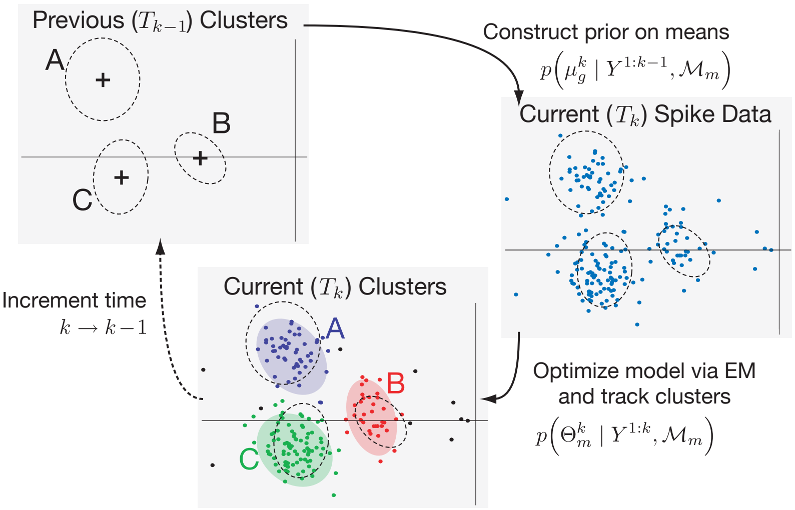
Previous (T_{k-1}) Clusters
A
B
C
Construct prior on means
$p\left(\mu_g^k \mid Y^{1:k-1}, \mathcal{M}_m\right)$
Current (T_k) Spike Data
Increment time
$k \to k-1$
Current (T_k) Clusters
A
B
C
Optimize model via EM
and track clusters
$p\left(\Theta_m^k \mid Y^{1:k}, \mathcal{M}_m\right)$

Figure 3.2: The Bayesian clustering cycle. Clusters identified on the previous recording interval T_{k-1} are used to construct priors on the locations (i.e., means) of clusters on the current recording interval T_k. The current data are clustered via maximum *a posteriori* (MAP) optimization of a Gaussian mixture model via expectation-maximization (EM), and the new clusters are matched to previous clusters to "track" neurons.

are Gaussian distributions for all $\hat{G}^{k-1}$ clusters estimated to exist in interval T_{k-1}. The symbol $\psi_j^{k|k-1}$ denotes the parameters of the jth mixture component for interval T_k based on T_{k-1}. For the Gaussian distributions, the parameter $\hat{\mu}_j^{k-1}$ is the estimated value of the jth cluster mean in T_{k-1}, and S_j^{k-1} is the covariance associated with the estimation that the current mean μ_g^k is in the same location as the prior mean $\hat{\mu}_j^{k-1}$. In this model, $S_j^{k-1} \equiv R_j^{k-1} + Q^{k-1}$, where $R_j^{k-1} = \frac{1}{n_j^{k-1}} \Sigma_j^{k-1}$ is the measurement covariance matrix associated with the estimation of $\hat{\mu}_j^{k-1}$ (n_j is the number of data points in cluster $\mathcal{C}_j$) and the empirically-determined covariance matrix Q^{k-1} accounts for effects, such as electrode movement, that cause a cluster to drift around in feature space[6]. The mixture weight ω_j^k is defined as

$$\omega_j^k = \begin{cases} \frac{1}{c}\lambda_0 & j = 0 \\ \frac{1}{c}\mathsf{P}_{\mathsf{d},j} & j = 1, \ldots, \hat{G}^{k-1} \end{cases} \tag{3.9}$$

where λ_0 is the combined expected number of newly appearing neurons and false clusters[7] in the recording interval, $\mathsf{P}_{\mathsf{d},j}$ is the probability of detecting the jth neuron found in T_{k-1}, and c is the normalization constant. The parameters λ_0 and $\mathsf{P}_{\mathsf{d},j}$ are set by the user. ⧫

Remark 3.4. This choice of mixture weight ω_j^k, which sets the prior probability of assigning a cluster

[6] Q_j^k and R_j^k are treated more formally in Chapter 4 as the covariance matrices of the Gaussian process and measurement noise, respectively, within the context of a Kalman Filter. Also, the calculation of R_j^k used here is optimistic, since it assumes that the mean signal is stationary over the recording interval and that all measurement noise is due to the estimation of the cluster mean from its member spikes (not including uncertainty in the measurement of the spikes themselves).

[7] A *false cluster* is a spurious grouping of observations that does not represent a neuron, usually consisting of probable outliers rather than the true spikes. Its inclusion models the realistic possibility that not every cluster represents a distinct neuron that should be tracked.

to the jth component, arises from the following rationale. Assume that each neuron found in T_{k-1} is detected according to a Bernoulli trial[8] with probability $\mathsf{P}_{\mathsf{d},j}$. Assume also that the number of newly appearing neurons and false clusters are each Poisson-distributed with respective rates λ_ν and λ_ϕ, and that each of these may appear anywhere in the volume V with uniform probability; thus the $j = 0$ component in the mixture prior must capture both these new neurons and false clusters. Then, each mixture weight ω_j^k is determined by the *expected* number of clusters in each component. By the above assumptions:

$$E[\text{number of clusters in } j\text{th component} \mid j = 1, \ldots, \hat{G}^{k-1}] = \\ E[\text{number of clusters representing the } j\text{th neuron}] = \mathsf{P}_{\mathsf{d},j}$$

and

$$E[\text{number of clusters in uniform (zeroth) component}] = \\ E[\text{number of clusters representing new neurons}] + E[\text{number of false clusters}] = \lambda_\nu + \lambda_\phi \,.$$

Finally, to obtain (3.9), define $\lambda_0 \equiv \lambda_\nu + \lambda_\phi$ and note that the constant c is required by the constraint on mixture weights $\sum_j \omega_j^k = 1$. ◊

Remark 3.5. Clearly, the prior statistics and current data must be expressed in the same coordinates. Thus, if using data-dependent features such as PCA, to obtain μ_j^{k-1} and Σ_j^{k-1}, the spike waveforms from T_{k-1} are projected to the feature space of T_k, and then the prior clusters' statistics are recalculated in this space. ◊

Incorporating (3.8) into (3.7), the complete prior on the mixture parameters is

$$p\big(\Theta_m^k \,|\, Y^{1:k-1}, \mathcal{M}_m\big) = C \prod_{g=1}^{G_m} \sum_{j=0}^{\hat{G}^{k-1}} \omega_j^k \, f_j\big(\mu_g^k \,|\, \psi_j^{k|k-1}\big) \,, \tag{3.10}$$

where $C \equiv \prod_{g=1}^{G_m} p\big(\Sigma_g^k \,|\, Y^{1:k-1}, \mathcal{M}_m\big)\, p\big(\pi_g^k \,|\, Y^{1:k-1}, \mathcal{M}_m\big)$ is a constant representing the diffuse priors on the parameters Σ_g^k and π_g^k.

3.3.3 Extending EM to Account for Cluster Location Priors

Note that the prior (3.10) resembles the mixture likelihood (3.5) and would in fact share the same difficulty of maximization. Thankfully, the same solution approach can be used: add "hidden" variables and optimize via EM.

[8] This formulation allows the probability of detection to vary across known neurons (perhaps according to their firing rates or signal quality) and across intervals (to allow for electrode movement). Although the results presented in Section 3.5 use a common P_d, the general formulation is utilized in Chapter 4.

Definition 3.2. Let $\mathcal{Z} = \{\zeta_{gj}\}$ be the set of ***cluster association indicators***, hidden data that specify whether the cluster $\mathcal{C}_j^{k-1}$ found in T_{k-1} is related to the current cluster $\mathcal{C}_g^k$ in T_k, or, ideally,

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

♦

Employing the classical complete-data approach, the cluster association indicators and the mixture prior defined in Section 3.3.2 determine the *complete-data* log prior on the mixture parameters:

$$\log p\big(\Theta_m^k, \mathcal{Z} | Y^{1:k-1}, \mathcal{M}_m\big) = \sum_{g=1}^{G_m} \sum_{j=0}^{\hat{G}^{k-1}} \zeta_{gj} \log \Big[\omega_j^k f_j\big(\mu_g^k | \psi_j^{k|k-1}\big)\Big] + \log C\,. \tag{3.11}$$

Rewriting (3.4) to include the hidden spike membership indicators Z as well as the cluster association indicators $\mathcal{Z}$ gives the complete-data posterior,

$$p\big(\Theta_m^k, \mathcal{Z} | Y^{1:k}, Z, \mathcal{M}_m\big) \propto p\big(Y^k, Z | \Theta_m^k, \mathcal{Z}, \mathcal{M}_m\big)\, p\big(\Theta_m^k, \mathcal{Z} | Y^{1:k-1}, \mathcal{M}_m\big)\,. \tag{3.12}$$

As it is convenient to work with the log-posterior, take the logarithm of (3.12) and substitute in (3.6) and (3.11),

$$\begin{aligned} &\log p\big(\Theta_m^k, \mathcal{Z} | Y^{1:k}, Z, \mathcal{M}_m\big) = \\ &\qquad \sum_{i=1}^{N} \sum_{g=0}^{G_m} z_{ig} \log \big[\pi_g^k f_g\big(y_i | \theta_g^k\big)\big] + \sum_{g=1}^{G_m} \sum_{j=0}^{\hat{G}^{k-1}} \zeta_{gj} \log \Big[\omega_j^k f_j\big(\mu_g^k | \psi_j^{k|k-1}\big)\Big] + D\,, \end{aligned} \tag{3.13}$$

where D is a constant. This *complete-data log-posterior* (3.13) is the object equation of the EM algorithm's iterations, which follow.

3.3.3.1 E-Step

As in the classical EM algorithm, given the parameter estimates from the M-step, the expectation of each spike membership indicator, $\hat{z}_{ig}$, is:

$$\hat{z}_{ig} = \frac{\hat{\pi}_g^k f_g\big(y_i | \hat{\theta}_g^k\big)}{\sum_{n=0}^{G_m} \hat{\pi}_n^k f_n\big(y_i | \hat{\theta}_n^k\big)}\,.$$

Recall that $f_g\big(y_i | \hat{\theta}_g^k\big)$ is a Gaussian distribution with parameters $\hat{\theta}_g^k = \{\hat{\mu}_g^k, \hat{\Sigma}_g^k\}$ for the components $g = 1, \ldots, G_m$ and an outlier density for the zeroth mixture component. The expectation of the other hidden data, the cluster association indicators, i.e., $\hat{\zeta}_{gj} = E\Big[\zeta_{gj} \mid Y^{1:k}, \hat{\Theta}_m^k\Big]$, has an analogous form:

$$\hat{\zeta}_{gj} = \frac{\omega_j^k f_j\big(\hat{\mu}_g^k|\psi_j^{k|k-1}\big)}{\sum_{l=0}^{\hat{G}^{k-1}} \omega_l^k f_l\big(\hat{\mu}_g^k|\psi_l^{k|k-1}\big)} \,. \tag{3.14}$$

3.3.3.2 M-Step

Since the prior term in (3.13) is independent of the parameters π_g and Σ_g, these estimates remain the same as the classical ML clustering version. For the mixture weights,

$$\hat{\pi}_g^k = \frac{n_g}{N} \,,$$

where $n_g = \sum_{i=1}^N \hat{z}_{ig}$, and for the shared-volume form of the covariance matrix [67],[9]

$$\hat{\Sigma}_g^k = \lambda^k \frac{W_g^k}{|W_g^k|^{\frac{1}{d}}} \,,$$

where

$$\lambda^k = \frac{\sum_{g=1}^{G_m^k} |W_g^k|^{\frac{1}{d}}}{N}$$

and

$$W_g^k = \sum_{i=1}^N \hat{z}_{ig}(y_i - \hat{\mu}_g^k)(y_i - \hat{\mu}_g^k)^T \,.$$

Maximizing (3.13) with respect to μ_g, results in the estimate:

$$\hat{\mu}_g^k = \left[\sum_{i=1}^N \hat{z}_{ig}(\hat{\Sigma}_g^k)^{-1} + \sum_{j=1}^{\hat{G}^{k-1}} \hat{\zeta}_{gj}(S_j^{k-1})^{-1}\right]^{-1} \cdot \left[\sum_{i=1}^N \hat{z}_{ig}(\hat{\Sigma}_g^k)^{-1} y_i + \sum_{j=1}^{\hat{G}^{k-1}} \hat{\zeta}_{gj}(S_j^{k-1})^{-1}\hat{\mu}_j^{k-1}\right] \,, \tag{3.15}$$

in contrast to the ML estimation of the cluster center location,

$$\hat{\mu}_g^k = \frac{\sum_{i=1}^N \hat{z}_{ig} y_i}{\sum_{i=1}^N \hat{z}_{ig}} .$$

Note that Equation (3.15) has the form of a weighted average of the data points y_i with (fuzzy) membership to cluster $\mathcal{C}_g$ and the prior means $\hat{\mu}_j^{k-1}$ (fuzzily) affiliated to cluster $\mathcal{C}_g$, with the weights governed by the respective covariance matrices.

Remark 3.6. A minor drawback to the MAP parameter calculation is that (3.15) is a function of the parameters $\hat{\Sigma}_g^k$, implying the need to simultaneously solve the equations for the parameters $\hat{\mu}_g^k$ and $\hat{\Sigma}_g^k$. Alternatively, one may use an approximation for $\hat{\Sigma}_g^k$ to solve (3.15), such as its value from the previous EM iteration, and then find $\hat{\Sigma}_g^k$ in the usual way. ♢

[9] The ML equation assuming a fully variable covariance is $\hat{\Sigma}_g = \frac{1}{n_g}\sum_{i=1}^N \hat{z}_{ig}(y_i - \hat{\mu}_g^k)(y_i - \hat{\mu}_g^k)^T$. See [67] for other estimations using parsimonious covariance models.

3.3.4 Generating Seed Clusters

The EM algorithm requires initial values to seed its iterations. A key issue is the choice of these seed clusters, as the EM algorithm is highly susceptible to finding local optima near its initial values. Assuming again that the clusters found in T_{k-1} provide a good starting point, an obvious seeding strategy is to group the current data points according to the closest prior cluster, where "closest" is determined by the (squared) Mahalanobis distance between the ith data point y_i in T_k and the jth cluster center estimated from T_{k-1}:

$$d_j^2(y_i) = (y_i - \hat{\mu}_j^{k-1})^T (\hat{\Sigma}_j^{k-1})^{-1} (y_i - \hat{\mu}_j^{k-1})\,. \tag{3.16}$$

Recall that the EM algorithm is applied to a range of candidate model classes, with varying model order (numbers of clusters). The primary complication arises in cases where the candidate model order G_m is different from $\hat{G}^{k-1}$, the model order estimated in T_{k-1}. Such differences can arise, for example, when neurons go silent or new neural signals are introduced between sampling intervals. Each of the three possible relations between G_m and $\hat{G}^{k-1}$ require a different approach.

3.3.4.1 Case $G_m = \hat{G}^{k-1}$

The seed assignment process assigns each observation to the closest prior cluster: each y_i is assigned to the jth cluster, where j is the index that minimizes $d_j^2(y_i)$ in (3.16).

3.3.4.2 Case $G_m < \hat{G}^{k-1}$

The goal here is to produce good clustering seeds when $\Delta G = \hat{G}^{k-1} - G_m$ neuron(s) disappear (or perhaps become indistinguishable in the current feature space) between sampling intervals. To produce appropriate seeds, all $\binom{\hat{G}^{k-1}}{G_m}$ combinations of the $\hat{G}^{k-1}$ prior clusters are evaluated to determine which set of G_m prior clusters minimizes the sum of the squared Mahalanobis distance. This process tests the elimination of possible prior cluster(s), keeping the best to inform the current data set. These tests can be completed quickly, as the number of neurons is typically small. The left column of Figure 3.6 displays a seeding example with $\hat{G}^{k-1} = 3$ and $G_m = 2$.

3.3.4.3 Case $G_m > \hat{G}^{k-1}$

In this case, $\Delta G = G_m - \hat{G}^{k-1}$ "extra" seed clusters must be generated. Such a situation can occur when ΔG new neurons have been detected and a new cluster must be created for each. Another possibility is that the prior interval's clustering result was incorrect (with multiple neurons inaccurately grouped into one cluster) and the current clustering interval must rectify this error.

The spikes from T_k are first assigned to the $\hat{G}^{k-1}$ prior clusters, as in the first case above, after which the cluster that is most likely to contain multiple neurons is divided (see the right column of Figure 3.6 for an example with $\hat{G}^{k-1} = 1$ and $G_m = 2$). Since such a group is likely to have a larger data spread, the group with the largest average point-to-centroid Euclidean distance is chosen to divide. This cluster's points are projected onto its principal axis and then split between the adjacent points that have the largest distance between them along this line (see Figure 3.6F/G). This is essentially a one-step divisive hierarchical clustering technique.[10] The above identification and splitting of groups is repeated as necessary for $\Delta G > 1$.

3.3.5 Selecting the Model Class $\mathcal{M}_m$

The model selection step is based on a Bayesian approach as well, with the model probability taking the form:

$$P\big(\mathcal{M}_m | Y^{1:k}\big) = \frac{1}{E}\, \underbrace{p\big(Y^k | Y^{1:k-1}, \mathcal{M}_m\big)}_{\text{model evidence}}\, \underbrace{P\big(\mathcal{M}_m | Y^{1:k-1}\big)}_{\text{model prior}}, \tag{3.17}$$

where E is a normalizing constant. This probability (3.17) is difficult to compute because the evidence $p\big(Y^k | Y^{1:k-1}, \mathcal{M}_m\big)$ theoretically requires an integration over all possible parameters.

However, Laplace's method for asymptotic approximation of integrals (see Appendix A) can be employed to estimate a value of the evidence term while evaluating only at the MAP parameters $\hat{\Theta}^k_m$:

$$p\big(Y^k | Y^{1:k-1}, \mathcal{M}_m\big) \approx p\big(Y^k | \hat{\Theta}^k_m, \mathcal{M}_m\big)\, p\big(\hat{\Theta}^k_m | Y^{1:k-1}, \mathcal{M}_m\big)\, (2\pi)^{\eta_m/2}\, |\mathbf{H}(\hat{\Theta}^k_m)|^{-1/2}, \tag{3.18}$$

where η_m is the number of independent parameters in model $\mathcal{M}_m$. The first factor is the likelihood of the Gaussian mixture (3.5). The other factors, collectively known as the Ockham factor since they penalize the complexity of the model parameterization, include the parameter prior (3.10) and the Hessian matrix,

$$\mathbf{H}(\hat{\Theta}^k_m) = -\nabla_{\Theta^k_m} \nabla_{\Theta^k_m} |_{\hat{\Theta}^k_m}\ \log\, p\big(Y^k | \Theta^k_m, \mathcal{M}_m\big)\, p\big(\Theta^k_m | Y^{1:k-1}, \mathcal{M}_m\big),$$

which has an analytical expression for the model classes under consideration, provided in Appendix B. Most popular model selection approaches, such as the Akaike Information Criterion (AIC) and Bayes Information Criterion (BIC)[11], are essentially approximations to (3.18) and specific to the maximum likelihood method [68]. For the MAP clustering applications like the one in this chapter, Laplace's method naturally incorporates the prior on Θ^k_m. Additionally, because it makes fewer approximations than the AIC and BIC, calculation of model evidence by Laplace's method is more

[10] In practice the split location is constrained to the middle 90% of points in the group. Otherwise, the largest difference between points is likely to be between an *outlier* and a true cluster, and thus a seed cluster of only outlier(s) would result. (How much to limit the split location depends on the experimental conditions and spike detector's false positive rate.)

[11] $BIC \equiv 2 l_M(\hat{\Theta}^k_m \mid Y^k, \mathcal{M}_m) - \eta_m \log N$, for maximized mixture log-likelihood l_M, and number of independent model parameters η_m.

capable of distinguishing the best model in "close calls" common in clustering noisy neural data.

The model class prior $P\big(\mathcal{M}_m|Y^{1:k-1}\big)$ in (3.17) is the model selection output from T_{k-1}, under the assumption that the model class is constant. However, there exists some probability that the model class changes (e.g., neural signal sources appear or disappear). Thus, a weighted mixture of the previous result and a uniform prior is appropriate:

$$P\big(\mathcal{M}_m|Y^{1:k-1}\big) \leftarrow \alpha\, P\big(\mathcal{M}_m|Y^{1:k-1}\big) + (1-\alpha)\frac{1}{\bar{M}} \ ,$$

where $\bar{M}$ is the total number of model classes under consideration. This places a "forgetting factor" on the prior, governed by the parameter α, and ensures a nontrivial probability of each model class at every sampling interval. (The results in Section 3.5 use $\alpha = 0.95$.)

3.4 Tracking Clusters Across Intervals

Ultimately the goal in spike sorting of successive recording intervals is to "track" individual neurons—that is, to associate specific neurons with specific clusters over time. Viewing this as a data association task on the means, the quantity $\hat{\zeta}_{gj}$ already encodes the probability that current cluster $\mathcal{C}_g^k$ is associated with prior cluster $\mathcal{C}_j^{k-1}$, *relative* to all $\hat{G}^{k-1}+1$ components in the prior (3.10). Each current cluster $\mathcal{C}_g^k$ is therefore matched to a prior cluster $\mathcal{C}_{j^*}^{k-1}$ via $j^* = \arg\max_j \hat{\zeta}_{gj}$. Thus, at the completion of the EM iterations, in addition to the model parameters $\hat{\Theta}_m^k$ and the cluster memberships $\hat{z}_{ig}$, the algorithm also yields cluster associations $\hat{\zeta}_{gj}$ for tracking.

When $\mathcal{C}_g^k$ is matched to the uniform distribution, it is considered a new neuron, highlighting the importance of a uniform component in the prior. Without the uniform distribution, $\mathcal{C}_g^k$ would be matched to the closest $\mathcal{C}_j^{k-1}$ (even if it is not close in absolute measures), and some *ad hoc* procedure would be required to verify it does not represent a new neuron. Including the uniform distribution effectively places a data-dependent minimum threshold on the Mahalanobis distance allowed to match $\mathcal{C}_g^k$ to prior cluster $\mathcal{C}_j^{k-1}$.

Disappearing neurons are identified when prior clusters are not matched to any current clusters. Additionally, *multiple* current clusters $\mathcal{C}_g^k$ may match the same prior cluster $\mathcal{C}_j^{k-1}$, marking a "split" of the neuronal signal components. A single-match nearest neighbor approach could be used, but does not fit as naturally with the mixture prior used in Definition 3.1. Further consideration of these issues is included in the more sophisticated tracking procedure of Chapter 4.

3.5 Experimental Results

The proposed MAP algorithm was applied to recordings from macaque parietal cortex, collected in acute recording sessions with platinum-iridium, 1.5 MΩ-impedance electrodes. Spikes were detected from the recorded voltage stream according to a wavelet matching approach [55], aligned by their minimum, and projected onto a two-dimensional PCA feature space prior to clustering.

As noted earlier, EM optimization of a Gaussian mixture model with ML parameters has shown its effectiveness in many clustering applications [26] and specifically spike sorting [27–32]. Thus, the proposed method is compared to such a technique, which has been used for over two years in hundreds of recording sessions in the electrode positioning algorithm. This method was previously chosen for use in the electrode positioning algorithm due to its success compared to other spike sorting options, especially as the application requires real-time computation and robustness when only small amounts of data are available.

In the implementation of the contrasting ML approach, seed clusters are generated from a standard hierarchical agglomerative technique and model order is selected according to Bayesian information criterion (BIC), following the suggestions of [57]. Both the MAP and ML implementations use the same common-volume parsimonious models of the components' covariance matrices and the same "background" mixture component to capture outliers. Note that, in comparison to the standard ML method of Section 3.2, this ML implementation benefits greatly from such decisions, which were informed by extensive experience with using that approach. Below, consecutive sampling intervals are examined in detail and then views of algorithm performance over longer time frames are provided.

3.5.1 Detail: Sequence of Consecutive Recording Intervals

Figures 3.3 and 3.4 display clustering results over a sequence of twelve consecutive recording intervals, chosen to highlight how the MAP algorithm enables neuron tracking, especially as compared to alternatives. Each sampling interval lasts 10 seconds, with separating intervals of about 25 seconds during which no signals are sampled. For consistent visualization, the same PCA feature space is used to present the results of each interval (rather than the PCA features of the individual intervals, in which the clustering process took place). Although it is impossible to know the actual spike–neuron associations conclusively, the results are compared to a best-effort "ground truth" manual clustering of the data, as determined by an expert thorough examination of *both* the spikes' waveforms and PCA features (whereas the automated clustering uses only PCA features). In addition to MAP and ML algorithm results, a k-means clustering result is also presented, with the number of clusters k manually selected to match the number of clusters in the ground truth results. Listed for each interval in these figures is the percentage of spikes correctly classified, calculated as follows: Each cluster is matched to the "truth" cluster sharing the most spikes, and the number of spikes these clusters have in common is considered correctly classified. Finally, each cluster is labeled with a

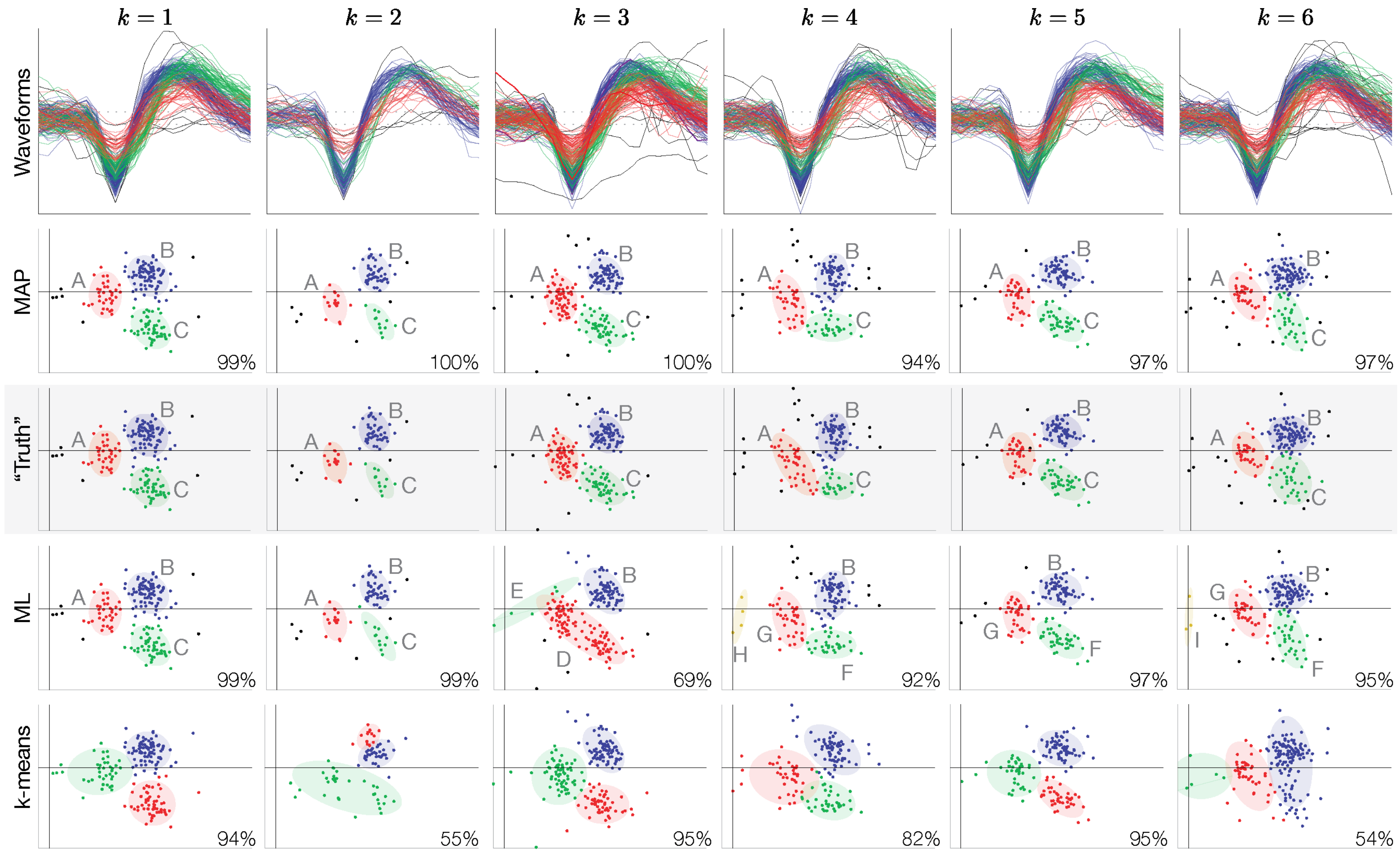

Figure 3.3: Intervals 1–6 of clustering results over twelve consecutive recording intervals. Rows: (1) Extracted, aligned waveforms (colored by MAP result); (2) Results from the proposed MAP algorithm; (3) Manual “ground truth”; (4) Results from the baseline (ML) algorithm; (5) Results from k-means, with $k = 3$. Percentage of spikes classified correctly and neuron ID letters are also listed.

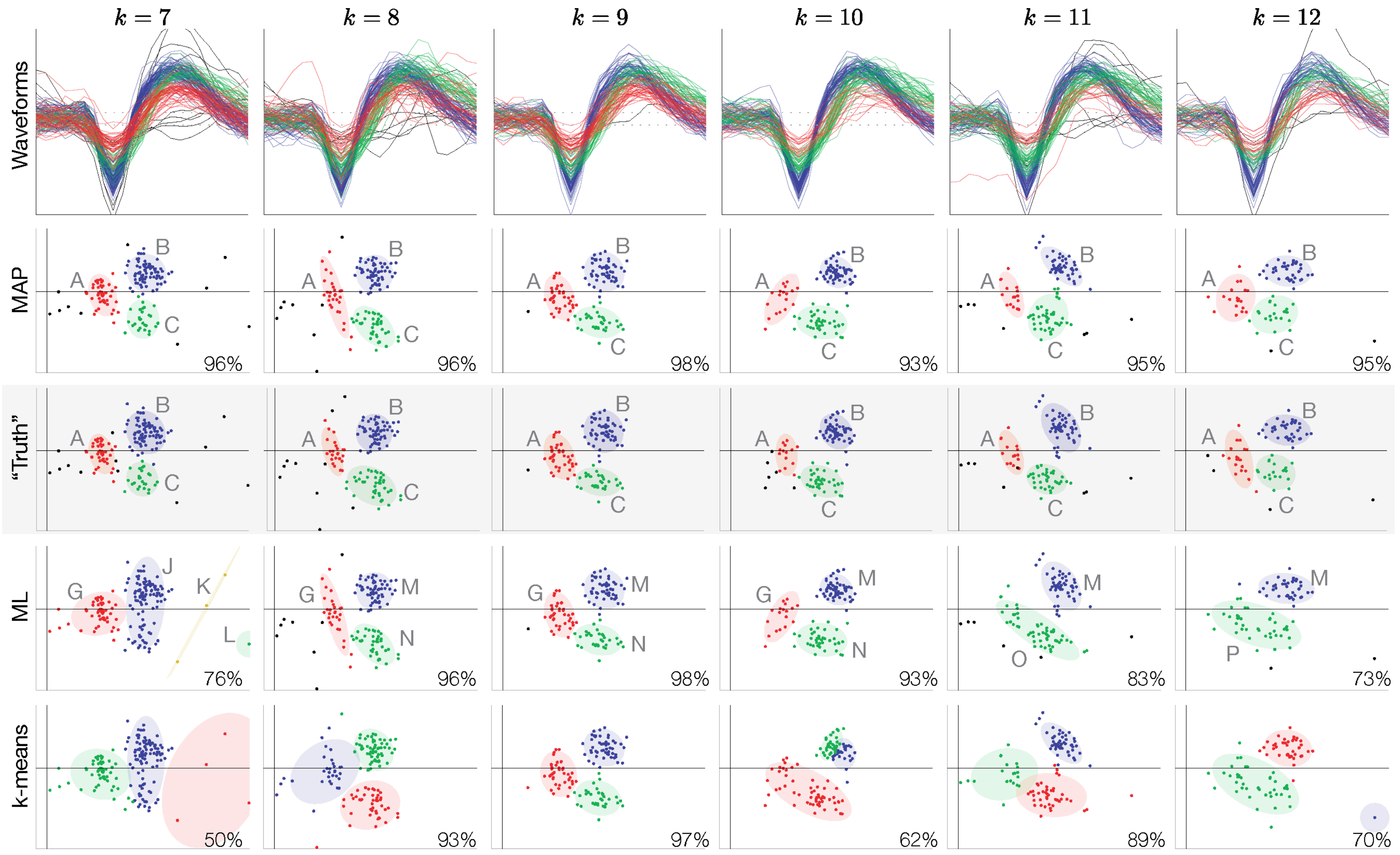

Figure 3.4: Intervals 7–12 of clustering results over twelve consecutive recording intervals. Rows: (1) Extracted, aligned waveforms (colored by MAP result); (2) Results from the proposed MAP algorithm; (3) Manual "ground truth"; (4) Results from the baseline (ML) algorithm; (5) Results from k-means, with $k = 3$. Percentage of spikes classified correctly and neuron ID letters are also listed.

"neuron ID," indicating the neuron that it tracks[12].

Table 3.1: Cluster Statistics of Selected Intervals

k	MAP					ML				
	g	Tr.	n_g	Err.	ΔFR	g	Tr.	n_g	Err.	ΔFR
2	1	A	49	0%	0%	1	A	48	2%	2%
	2	B	12	0%	0%	2	B	13	8%	8%
	3	C	17	0%	0%	3	C	17	0%	0%
3	1	A	73	0%	0%	1	A	76	4%	4%
	2	B	53	0%	0%	2	—	5	n/a	n/a
	3	C	65	0%	0%	3	C	115	86%	77%
4	1	A	73	4%	1%	1	A	74	3%	3%
	2	B	32	23%	23%	2	B	32	23%	23%
	3	C	36	17%	12%	3	C	36	17%	12%
						4	—	4	n/a	n/a
7	1	A	74	4%	4%	1	A	104	35%	35%
	2	B	29	7%	7%	2	—	1	n/a	n/a
	3	C	48	7%	7%	3	C	53	18%	18%
						4	—	3	n/a	n/a
11	1	A	49	9%	9%	1	A	54	0%	0%
	2	B	41	14%	14%	2	B	53	53%	47%
	3	C	16	7%	7%					
12	1	A	39	0%	0%	1	A	40	3%	3%
	2	B	22	10%	10%	2	C	41	105%	105%
	3	C	20	20%	0%					

Table 3.1 provides a detailed view of the intervals where MAP and ML results differed significantly. For these intervals, Table 3.1 lists a) Tr., the "truth" cluster to which the gth cluster was matched; b) n_g, the number of spikes in gth cluster; c) Err., the percentage of falsely classified spikes for this cluster; d) ΔFR, the percentage difference in firing rate between the cluster and its matching truth cluster. Here the error is defined as Err. $= \frac{\text{MC+FP}}{n_{g,\text{truth}}}$, where MC is the number of missed classifications and FP is the number of false positives.

Ostensibly, spikes from the same three neurons (labeled A, B, and C) persist through the twelve sampling intervals of Figures 3.3 and 3.4, as determined in the "truth" clusters. The clustering challenge is difficult, however, as the spike waveform features are not highly separated and the firing rates (and thus numbers of data points) are sometimes low. Notice that the MAP algorithm consistently identifies three clusters in roughly the same PCA position. The ML algorithm often provides good results, but some intervals show incongruous (though statistically sound) results,

[12] Because the ML method does not include a natural data association process, the following procedure was used to test its neuron tracking ability. The clusters from T_1 are assigned neuron IDs (A, B, C, ...). Thereafter, a cluster in T_k is associated with the nearest cluster from T_{k-1}, provided its mean lies within 2 standard deviations of the prior mean location, using the covariance Q^k (the measure of expected movement of cluster means between intervals, discussed in Section 3.3.2, with the same value used in the MAP algorithm). If no match is found, a new track is created (new ID assigned). Note that the presented MAP tracking results are identical when using this procedure or when using the procedure in Section 3.4.

seemingly more volatile to noise variations. Meanwhile, the k-means solution is unreliable, even with the advantage of knowing the model order *a priori*.

Even a small number of intervals with poor results significantly impacts the ability to track neurons over time. For example, at $k = 3$, the ML method groups most spikes from neurons A and C into a single cluster, whose mean is relatively distant from the means of interval T_2. When attempting to associate the clusters across these intervals, this result is interpreted as the loss of neurons A and C and the appearance of a new neuron (D) in T_3 (rather than tracking neurons A and C from T_2, as the MAP method does). Then, when the spikes from neurons A and C are (mostly) correctly classified by the ML method during T_4, they are considered as "new" neurons G and F since their mean locations are appreciably removed from the prior mean. Similar errors occur also at $k = 7$, $k = 11$, and $k = 12$ and thus prevent neuron tracking in the ML method. Table 3.2 lists all "neuron tracks" from this sequence of intervals for the MAP and ML clusters. Note that the MAP algorithm results in one track per neuron lasting across all twelve intervals, whereas the ML algorithm cannot track the three neurons and also generates many spurious tracks.

Table 3.2: Neuron Tracks

Cluster	Neuron	Start	Duration
Method	ID	k	Δk
MAP	A	1	12
	B	1	12
	C	1	12
ML	A	1	6
	B	1	2
	C	1	2
	D	3	1
	E	3	1
	F	4	3
	G	4	7
	H	4	1
	I	6	1
	J	7	1
	K	7	1
	L	7	1
	M	8	5
	N	8	3
	O	11	1
	P	12	1

3.5.2 Gross Measures of Cluster Consistency

The consistency of the clustering outcome is a primary benefit of the proposed algorithm, as evidenced in Figure 3.3. Although it is difficult to compellingly quantify this advantage, one metric to examine is the change in the number of clusters from interval to interval. Taking $\Psi =$

$\sum_{s=1}^{S}\sum_{k=2}^{K_s}|\hat{G}^k - \hat{G}^{k-1}|$ over all time intervals k of each recording session s for which the proposed algorithm was applied provides a quantitative measure of "inconsistency"—note, however, that many changes in the value of G are correct as the number of recorded neurons may vary over the recording session[13]. Examining a set of 100 consecutive recording sessions, comprising about one month of recordings and 21 914 total sampling intervals, $\Psi = 3516$ for the MAP algorithm, compared to $\Psi = 17\,646$ for the ML algorithm, an 80% decrease.

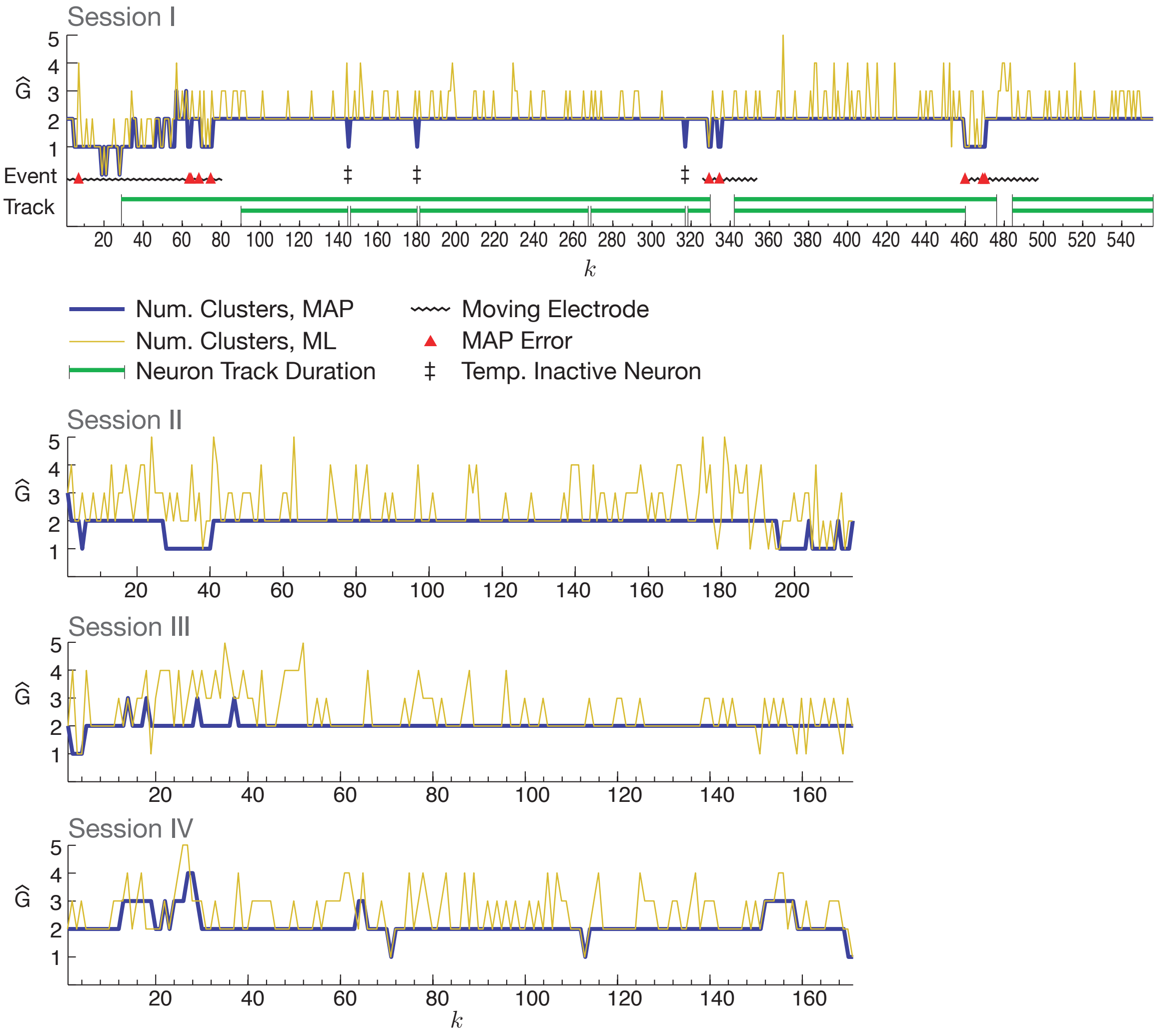

Figure 3.5: Number of estimated clusters $\hat{G}$ over time for four entire recording sessions, comparing consistency of MAP and ML clustering algorithms. For Session I, the duration of several selected neuron tracks is marked, as well as several "event" types that may provide insight into why tracking failed during some intervals.

Several example plots of the estimated number of clusters $\hat{G}$ over an entire session are shown in Figure 3.5. Clearly, the MAP algorithm provides a much more consistent model, though some spurious changes in the number of clusters are evident. Additional detail is provided for Session I,

[13] A recording *session* is a single experimental trial, consisting of many sampling intervals from a single electrode.

including the period over which the neurons were successfully tracked using the proposed method. Often, the periods in which $\hat{G}$ varies significantly correspond to commanded motions of the electrode as it attempts to isolate a neuron. A change in $\hat{G}$ for a single interval sometimes results from a temporarily inactive neuron, and sometimes indicates a mistake by the MAP algorithm. These errors are usually related to a lack of spike waveform separability in PCA space or coincidental alignments of the spike detector's false positives (outliers).

3.5.3 Changing Numbers of Clusters

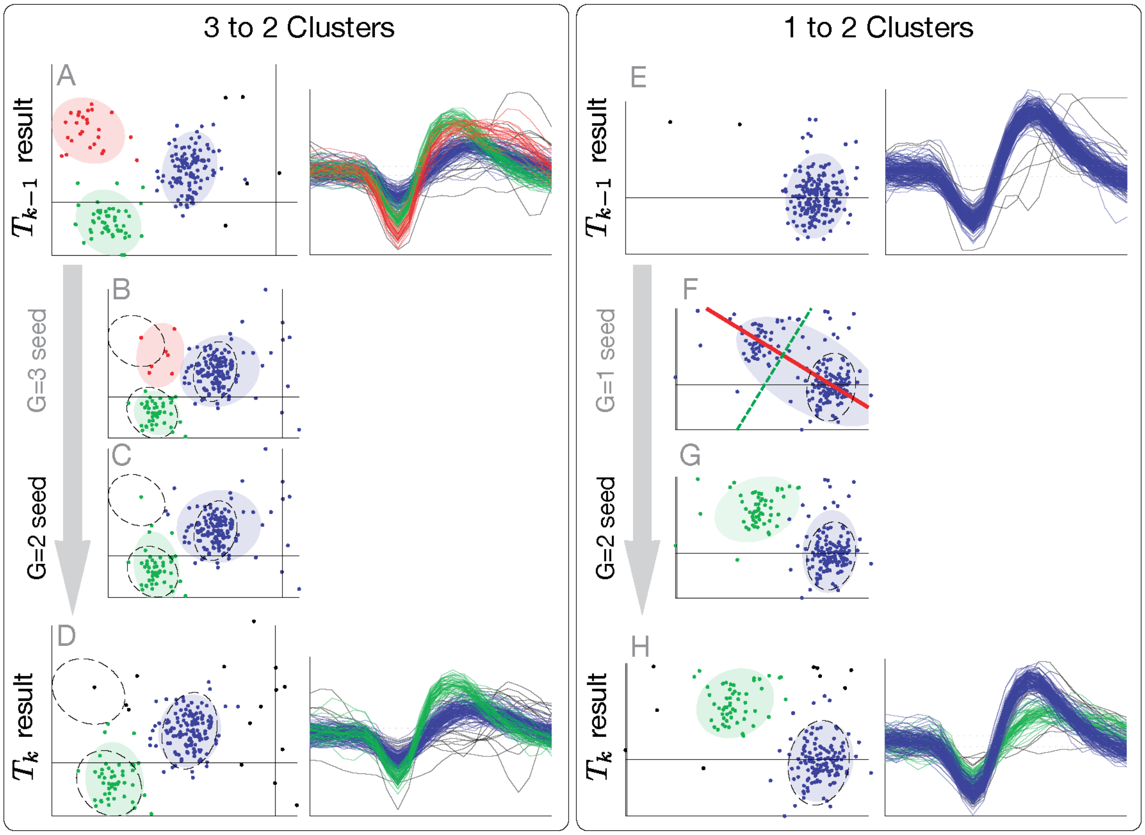

Figure 3.6: Examples of consecutive intervals with changing numbers of neurons. Left column: Decrease from $\hat{G}^{k-1} = 3$ to $\hat{G}^k = 2$ clusters. Right column: Increase from $\hat{G}^{k-1} = 1$ to $\hat{G}^k = 2$ clusters. Black dashed 2-sigma ellipses show locations of prior clusters. B/C: Seed clusters for $G = 2$ by keeping best two of the three prior clusters. F/G: Seed clusters for $G = 2$ by splitting the one prior cluster along its principal axis (think red line), at the point of largest gap (dashed green line).

The importance of properly detecting the appearance or disappearance of neurons in the recorded signal has motivated several decisions in developing the MAP algorithm. The columns of Figure 3.6 present a detailed view of two examples involving transitions to fewer and to more clusters. In both cases, although $\hat{G}^{k-1}$ is different from the number of neurons in T_k, the MAP algorithm determines the correct number of clusters. In each case, the $\hat{G}^k = 2$ model class was chosen with probability greater than 99%, demonstrating that strong evidence far outweighs the model prior. Also shown in this figure are plots of the seed clusters for the selected model class and how those seed clusters are generated from the T_{k-1} result.

3.6 Discussion

The application results presented in Section 3.5 show how the MAP algorithm properly integrates information over time to provide more consistent clustering results, which enables tracking neurons from interval to interval. Although the procedure focuses on providing more consistent results, it also performs well when the prior is not similar to the current clusters. The prior's construction as a mixture of densities effectively influences the posterior cluster locations but assumes neither a certain number of clusters nor the *a priori* association of *particular* current and prior clusters. Thus, the algorithm is not unduly biased by the prior when evidence suggests the appearance (or disappearance) or neurons.

The MAP algorithm is more likely to avoid poor local optima because of its seeding method and because the prior on the cluster means better guides the EM process. (Although the resulting optimization is not guaranteed to be the global optimum of the posterior, it tends to be the *desired* solution.) Also, the model selection procedure is quite effective because (a) the model evidence increases when the parameters are near those of the last interval, as influenced by the MAP EM approach, (b) the model prior biases the result toward a consistent number of clusters, and (c) using Laplace's method for model evidence requires fewer approximations than other methods.

The contribution from generating good seed clusters dominates when there are many data points and/or when the covariance S_j^{k-1} is large (both situations are more likely to arise during longer recording intervals). For short sampling intervals (with relatively few data points but effectively stationary signals), the use of cluster location priors during EM plays a stronger role, and enables the same clusters to be identified with relatively few data points.

Because the motivation for this work is a real-time application to autonomous electrode positioning and brain-machine interfaces, computational considerations are important. The total processing time for each sampling interval in Section 3.5's results averaged ~ 2 seconds using non-optimized MATLAB code, well within the needs of the electrode positioning algorithm. The main computational burden is the calculation of the Hessian matrix, which may be eliminated by using other estimates, such as BIC, to approximate model evidence (instead of Laplace's method) while maintaining most benefits of the MAP approach. In this case, the average time per interval drops to about 0.25 seconds. Note that, when using the same model selection method (e.g., BIC), the MAP algorithm in fact executes about 40% faster than the ML version—although the MAP method is more complex, it usually requires fewer EM iterations to converge because the seeding strategy creates initial conditions closer to the local optimum.

In conclusion, this chapter has detailed a Bayesian clustering algorithm to optimize a mixture model via EM. In addition to constructing an appropriate prior on cluster locations and adjusting the traditional EM approach to incorporate this term, it incorporates a new process for generating seed clusters and a suitable model class selection method. As a whole, this technique provides more consistent clusters and enables the association of clusters across consecutive time intervals, specifically quantifying the probability of these associations in the expectation of the cluster association indicators $\hat{\mathcal{Z}}$. While this chapter has shown how $\hat{\mathcal{Z}}$ may be used in a simple nearest neighbor tracking method, the more robust tracking method presented in Chapter 4 incorporates information from several recent intervals and maintains multiple "hypotheses" of possible cluster results and cluster associations, rather than selecting the best model class and best cluster associations after every interval.

Chapter 4

Multiple Hypotheses Tracking for Clusters

This chapter introduces a method for robustly tracking targets whose locations are inferred from clusters of measurements in successive sampling intervals. This new procedure, termed *MHTC* for *multiple hypothesis tracking of clusters*, maintains several possibilities for how the data should be clustered and for how each cluster should be associated to a particular neuron. The MAP clustering technique of Chapter 3, along with its inherent measures of cluster associations across time intervals, provides a key component of the MHTC algorithm. While the previous chapter demonstrated a simple nearest neighbor tracker easily implemented in conjunction with the MAP clustering procedure, such a solution succeeds only when the clusters are regular. MHTC offers a more robust solution, demanded in situations when, for instance, a neuron momentarily ceases firing, an interval's clustering result contains an error, or the signals of different neurons are difficult to distinguish for a time.

The MHTC algorithm employs a delayed decision-making framework that evaluates a history of many recent sampling intervals to determine tracking probabilities, enabling it to overcome the transient effects that could otherwise cause the loss of a neuron track. Additionally, MHTC utilizes a recursive filter for estimating the state of each neuron, explicitly identifies false clusters, and includes a more sophisticated model selection technique. Although the focus of this chapter is the development of the MHTC algorithm for applications to neuronal data, the method may be applied to tracking any targets that are measured via "groups" of observations per recording interval (or scan).

The relevant target tracking context for this solution is reviewed in Section 4.1. The MHTC algorithm is presented in Section 4.2, supported by further mathematical detail included in Section 4.3. Section 4.4 details a particular implementation of MHTC for the spike sorting application motivating this thesis, the results of which are exhibited in Section 4.5.

4.1 Multitarget Tracking and Multiple Hypothesis Tracking

Tracking the identities of several neurons throughout a recording session can be viewed as a problem of *multitarget tracking* (MTT). In traditional MTT, the locations of several objects of interest (targets) are measured in sequential “scans” of an observation volume. Using these data, MTT solutions combine a filter for estimating the target states and a data association technique for assigning the current measurements to known targets. Modern MTT solutions typically allow for changing numbers of targets across scans, false measurements (clutter), and missed detections (temporary occlusions). This field of study has been historically motivated by military applications (e.g., radar scans to locate airplanes) and, more recently, computer vision.

Let us briefly review the foremost classical MTT data association methods [35, 37, 69]. These solutions govern which measurements are assigned to putative targets and used to update the state estimates of those targets (typically through a Kalman Filter). In common MTT practice, an *exclusivity* principle is usually enforced, under which each target may generate at most one measurement and each measurement can represent only a single target. Under this assumption, a set of legal *data association hypotheses* may be defined, where each hypothesis assigns every measurement to an existing target (or possibly designates it as a new target or false measurement). The global nearest neighbor (GNN) approach [37], the simplest and perhaps most widely applied method, chooses the single best hypothesis at each new scan, where the combined distance of the measurement locations to the predicted target locations is minimized (usually defined through a squared Mahalanobis distance). The descriptor “global” distinguishes GNN from the (greedy) nearest neighbor solution, where each best assignment is made successively.

Next in increasing complexity and accuracy is joint probabilistic data association (JPDA) [36,70], an “all neighbors” association approach, which updates a target using all the measurements near its predicted location, weighted by likelihood of the assignment. Essentially, this technique considers many possible data association hypotheses and *combines* them after every scan.

Finally, multiple hypothesis tracking (MHT), attributed to Reid [38], maintains many possible data association hypotheses and propagates the corresponding target state estimates for each hypothesis, implicitly deferring decisions in anticipation that subsequent data measurements will resolve any ambiguity. MHT is generally accepted as the preferred data association mechanism for modern MTT systems [39] but, because each hypothesis spawns a new set of child hypotheses at each new data scan, the MHT approach may result in a combinatorial explosion of hypotheses. A key recent development in MHT implementations is the use of an algorithm, originally due to Murty [71], to generate only the L-best hypotheses [40,41,72,73]. This technique (see Section 4.3.3) obviates the need to enumerate all possible hypotheses, thus maintaining computational feasibility even for large numbers of measurements and targets. Figure 4.1 illustrates an example hypothesis tree structure

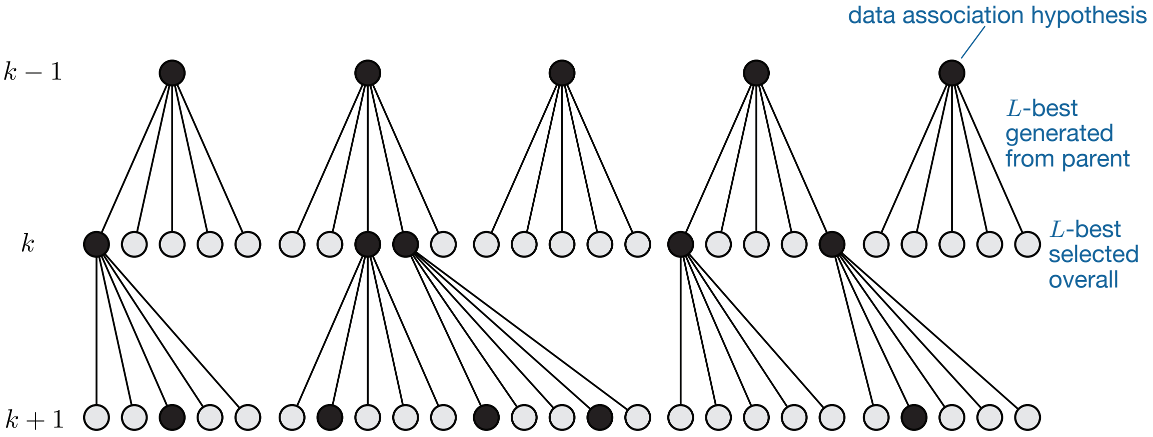
data association hypothesis
$k-1$
k
$k+1$
L-best generated from parent
L-best selected overall

Figure 4.1: Traditional MHT tree structure, displaying L-best hypothesis technique, with $L = 5$. Black circles represent surviving data association hypotheses at each time step.

using an L-best hypotheses technique. Suppose, at time $(k-1)$, L hypotheses exist. Then with the data arriving at time k, each of these parent hypotheses spawns the L best associations conditioned on the parent hypothesis being correct. The resulting L^2 total hypotheses generated at time k are guaranteed to contain the best L hypotheses overall, which are selected for propagation to time $(k+1)$. A variety of other techniques may also be implemented to keep the hypothesis growth in check, as discussed in Section 4.4.1.

The clustering solution proposed in Chapter 3 effectively implemented a simplified multitarget tracker, utilizing one-step prediction of a stationary target location (PCA coordinates of a cluster mean) and a (non-exclusive) nearest neighbor rule for data association. Thus, it may be considered a "single hypothesis" solution, where the best clustering result and the best data association assignments are taken after every interval. The MHTC algorithm developed in this chapter is a multiple hypothesis approach to combined clustering and tracking, propagating not only multiple data association hypotheses but also multiple hypotheses on how the data should be clustered (i.e., different mixture models).

The key differentiator of MTT for extracellular recording versus traditional applications is the multitude of observations (spikes) per target (neuron) in each scan (sampling interval); that is, the neuron "target" may be observed only through the population of spikes it generates during every sampling interval. The measurements of neuron target location are therefore not received directly from the sensor but only through the estimated means of many spike observations. Further, the uncertainty inherent in the clustering problem greatly complicates this factor, as correctly assigning spike observations to their generating neurons is essential for accurately estimating cluster means, and even the *number* of measurements (clusters) is unknown *a priori*.

While the above MTT literature, and especially the MHT framework, are leveraged for the MHTC algorithm, no previous work directly addresses the above problem, in which multiple hypotheses of clustering models represent possible new measurement sets. The similarities to existing "(interacting) multiple model" methods are largely semantic—these solutions account for maneuvering targets by considering various models of the dynamical systems representing the targets [36,74], whereas in MHTC the term "model" usually refers to the mixture model used to cluster data observations. Interestingly, a previous application of MHT to classify action potentials does exist, due to Hansson et al. [75]; however, their problem statement is completely different from the one addressed by MHTC. Their goal was to identify separate C-fibers in human skin nerves with characteristic (but changing) response latencies after experimentally induced electrical stimuli, a problem of (one-dimensional) tracking of a spike time "marker" over successive trials with no clustering requirement.

4.2 Integrating Clustering into an MHT Framework

This section introduces the MHTC methodology for using an MHT framework to track targets based on clusters of observations, with an emphasis on applying this technique for spike sorting.

4.2.1 Definitions

4.2.1.1 Target Tracking and Hypothesis Terminology

To conform with MTT convention, each cluster mean $\hat{\mu}_g \in \hat{\Theta}_m$ will often be called a *measurement* and the term *target* describes a putative neuron, whose *position* is the coordinates of its mean waveform shape. When integrating clustering into a multiple hypothesis tracking framework, two types of hypotheses exist: model hypotheses and data association hypotheses. A *model hypothesis*, $\mathcal{M}_m$, dictates the mixture model class used to cluster the spikes recorded during a particular interval[1], which will lead to corresponding optimized model parameters $\hat{\Theta}_m$ and expected cluster association indicators $\hat{\mathcal{Z}}$, as in Chapter 3. A *data association hypotheses*, $h_l = \{\tau_l, \nu_l, \phi_l\}$, indexed by l, assigns each cluster in a given model hypothesis to a target (or marks it as spurious): The set τ_l contains the assignments of the model's clusters to known targets, $\tau_l = \{(g_1, j_1), \ldots, (g_{N_\tau}, j_{N_\tau})\}$, where each indexed pair (g, j) matches the gth cluster to the jth neuron target; $\nu_l = \{g_1, \ldots, g_{N_\nu}\}$ contains the indices of the model's clusters that are identified as new neurons; and $\phi_l = \{g_1, \ldots, g_{N_\phi}\}$ holds the the indices of *false clusters* (spurious groupings of outliers or similar clustering errors)[2] in the current model. Note that N_τ, N_ν, and N_ϕ are the respective cardinalities of the sets τ_l, ν_l, and ϕ_l. A *legal* hypothesis must assign every measurement to only a single target (or classify it as false) and may only assign at most one measurement to each target. Thus, note that the total number of measurements is $G_m = N_\tau + N_\nu + N_\phi$.

[1] The notations in this paragraph actually have time dependence but are always be used to refer to the current (kth) interval T_k only, and thus the superscript k is suppressed for readability.

[2] The term *false cluster* is equivalent to *clutter*, *spurious measurement*, or *false alarm* used in other literature.

Together, a data association hypothesis and its parent model hypothesis form a particular *joint hypothesis*[3] at time k, $H_l^k = \{\mathcal{M}_{m(l)}, h_l\}$. The joint hypothesis H_l^k thus postulates a complete set of data associations for interval T_k, including the spike–cluster (observation–measurement) associations in $\mathcal{M}_{m(l)}$ and the cluster–neuron (measurement–target) associations in h_l. A particular joint hypothesis is combined with its *parent hypothesis* $H_{\rho(l)}^{1:k-1}$ to define a *global hypothesis*, $H_l^{1:k} = \{H_l^k, H_{\rho(l)}^{1:k-1}\}$, which includes the full history of all model and data association hypotheses from the first through kth intervals. Note that the subscripts $m(l)$ and $\rho(l)$ are used to indicate the index of the model or global hypothesis, respectively, that is the parent of the lth data association hypothesis; similarly, in a slightly abusive notation, $\rho(m)$ may also indicate the parent global hypothesis of the mth model hypothesis. Finally, it is convenient to define Ω^k as the set of all surviving global hypotheses $\{H_l^{1:k}\}_{l=1}^L$ and all data $Y^{1:k}$, which thus provides all relevant measured and hypothesized information about time k: $\Omega^k = \big\{\{H_l^{1:k}\}_{l=1}^L, Y^{1:k}\big\}$.

4.2.1.2 Dynamical System Model

After measurements are assigned under the hypothesis h_l, they are used to update the neuron's *track*, its sequence of estimated positions (waveform means). More generally, we may consider the neuron's *state*, which, depending on user preferences and experimental paradigms, might include neuron characteristics such as neuron waveform covariance, firing rate, or "velocity" (rate of change of the mean waveform between intervals) as well as position.

Remark 4.1. Recall that clustering operates in *feature space*, and thus the measurements are made in this feature basis. Ideally, the neuron state would simply be tracked in this feature space, but this is only possible for invariant feature spaces. Using PCA, for instance, results in a data-dependent features space, where the basis changes from interval to interval. In this case, the tracking problem is ill-posed because the statistics derived from the data in T_{k-1} (in basis $\mathcal{B}^{k-1}$) cannot be readily transformed into the current feature space ($\mathcal{B}^k$). For now, it is easiest to consider the position to be in the full waveform space, rather than the reduced-dimension feature space, and this issue will be further addressed in Section 4.4. ◊

Let x_j^k be the state of the jth target (neuron) at time k (more precisely, its average state over recording interval T_k), and presume the measurements consist of the means estimated during the clustering process, $\hat{\mu}_j^k$. The neuron's dynamics are modeled by a simple linear, discrete-time, Gauss–Markov system:

$$\begin{aligned} x_j^k &= F^{k-1}\, x_j^{k-1} + v_j^{k-1} \\ \hat{\mu}_j^k &= H^k\, x_j^k + w_j^k \end{aligned} \tag{4.1}$$

[3] The terms *joint hypothesis* and *model hypothesis*, as well as the detail that measurements and positions are statistical means, are new in this thesis. The other terms defined in this section originate in previous work, such as [36, 37, 69].

where F^{k-1} is the transition matrix of the system and H^k is the measurement matrix. The process noise v_j^k and measurement noise w_j^k are *iid*, zero-mean, Gaussian random processes, respectively governed by the covariance matrices Q_j^k and R_j^k:

$$v_j^k \sim \mathcal{N}(0, Q_j^k) \quad \text{and} \quad w_j^k \sim \mathcal{N}(0, R_j^k)\,.$$

In this model, the process noise v_j^k accounts for all non-stationarity of the neurons' waveforms between recording intervals. In the sections that follow, the Kalman Filter is used for state estimation, as it is the optimal recursive filter for the above model [36].

4.2.1.3 Probability Models

Given a set of targets in the parent hypothesis from $H_{\rho(l)}^{1:k-1}$, the probabilities of the existence and location of new measurements in interval k are modeled as follows. The occurrence that the jth existing target is *detected* (i.e., produces a cluster) is considered a Bernoulli trial with probability $\mathsf{P}_{\mathsf{d},j}$:

$$P\big(\delta_{j,l} | H_{\rho(l)}^{1:k-1}, Y^{1:k-1}\big) = \big(\mathsf{P}_{\mathsf{d},j}\big)^{\delta_{j,l}} \, \big(1 - \mathsf{P}_{\mathsf{d},j}\big)^{1-\delta_{j,l}}\ , \tag{4.2}$$

where $\delta_{j,l}$ indicates whether the jth target from the parent hypothesis $H_{\rho(l)}^{1:k-1}$ is tracked under hypothesis h_l,

$$\delta_{j,l} = \begin{cases} 1 & \text{if the } j\text{th target is tracked under } h_l \\ 0 & \text{otherwise} \end{cases}\,.$$

If the target is detected, the associated measurement is expected to appear near the target's predicted location with a Gaussian distribution,

$$p\big(\mu_g^k | (g,j) \in \tau_l, h_l, H_{\rho(l)}^{1:k-1}, Y^{1:k-1}\big) = f_{\mathcal{N}}\big(\mu_g^k | \hat{\mu}_j^{k|k-1}, S_j^k\big)\,, \tag{4.3}$$

where the mean and covariance are provided by the Kalman filter (see Section 4.2.3), and recall that $\tau_l \in h_l$ contains the indexed pairs assigning measurements to targets for the data association hypothesis h_l.

The numbers of new targets or false clusters appearing in a given time interval are each modeled by the Poisson distribution with respective rates λ_ν and λ_ϕ:

$$P\big(N_\nu\big) = \frac{(\lambda_\nu)^{N_\nu}\, e^{-\lambda_\nu}}{N_\nu!} \tag{4.4}$$

$$P\big(N_\phi\big) = \frac{(\lambda_\phi)^{N_\phi}\, e^{-\lambda_\phi}}{N_\phi!}\,. \tag{4.5}$$

If a measurement originates from a new target or false cluster, it may arise anywhere in the obser-

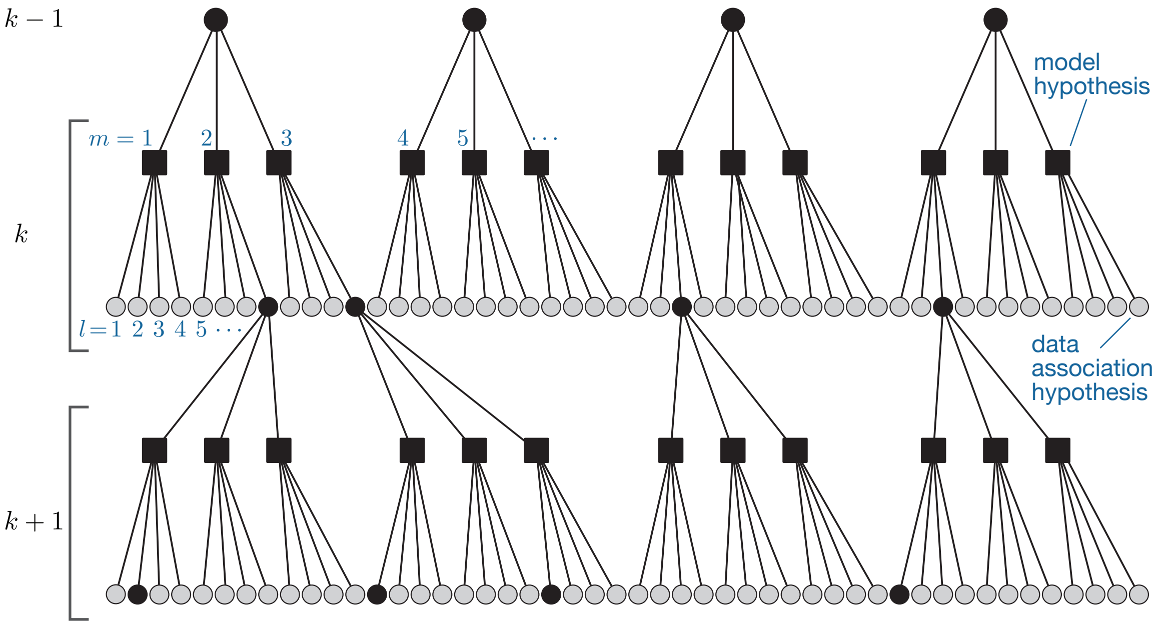

Figure 4.2: MHTC hypothesis tree structure, illustrating the integration of model hypotheses into the traditional MHT framework, using $L = 4$ and $\bar{M} = 3$. Squares represent model hypotheses (i.e., clustering output) and black circles represent surviving data association hypotheses at each time step.

vation volume V with a uniform PDF:

$$p\big(\mu_g^k | g \in \{\nu_l, \phi_l\}, h_l, H_{\rho(l)}^{1:k-1}, Y^{1:k-1}\big) = \frac{1}{V}\,. \tag{4.6}$$

The parameters $\mathsf{P}_{\mathsf{d},j}$, λ_ν, and λ_ϕ are set by the user and may vary across sampling intervals. The above probability models are incorporated into the hypothesis probability calculations in Section 4.3 and Section 4.A.

4.2.2 Hypothesis Tree Structure

As shown in Figure 4.2, the MHTC algorithm extends the traditional MHT tree to include model hypotheses as well as data association hypotheses. If L global hypotheses exist at time $(k-1)$ and we consider $\bar{M}$ model classes for each parent hypothesis, then $(L\bar{M})$ model hypotheses are formed at time k, each of which is optimized according to the MAP EM procedure of Chapter 3. As in Figure 4.1, only the L best data association hypotheses are generated from each parent, but in MHTC they are formed from a parent model hypothesis rather than directly from a parent data association hypothesis. To end the hypothesis management at time k, the best L global hypotheses are selected from the $(L^2\bar{M})$ that have been generated. Section 4.2.3 provides further detail on the above process.

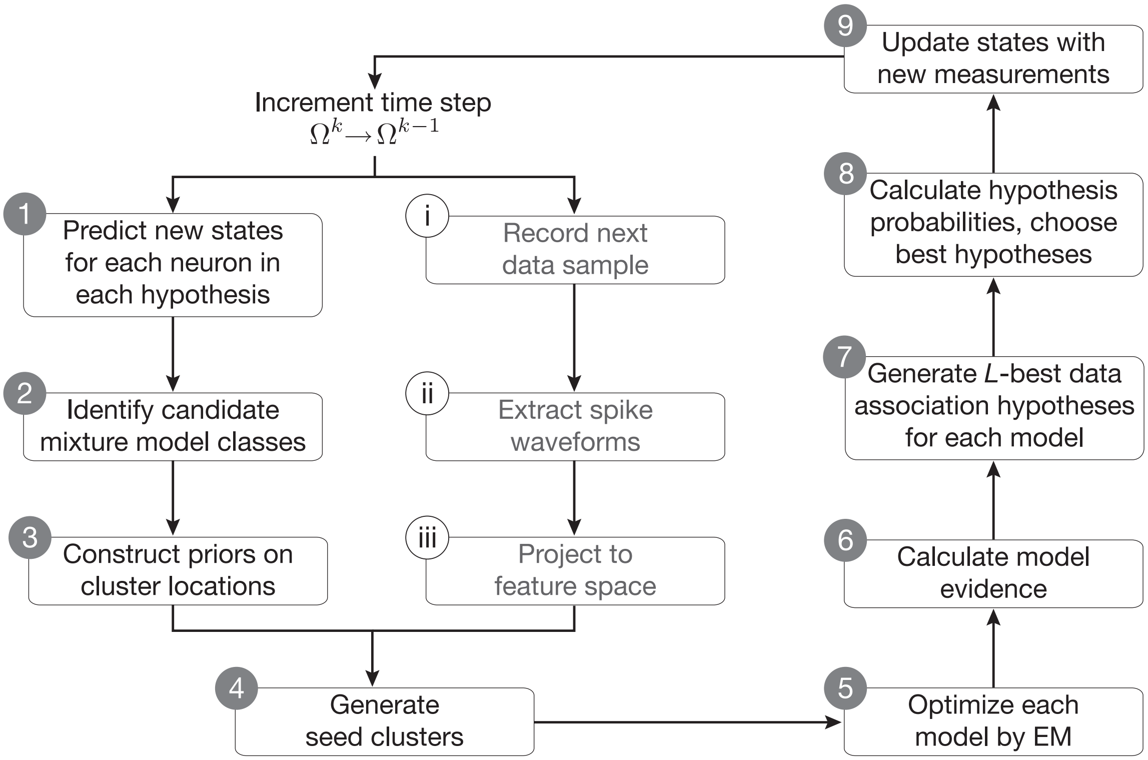
9
Update states with new measurements
Increment time step
$\Omega^k \rightarrow \Omega^{k-1}$
1
Predict new states for each neuron in each hypothesis
i
Record next data sample
8
Calculate hypothesis probabilities, choose best hypotheses
7
Generate *L*-best data association hypotheses for each model
2
Identify candidate mixture model classes
ii
Extract spike waveforms
6
Calculate model evidence
3
Construct priors on cluster locations
iii
Project to feature space
4
Generate seed clusters
5
Optimize each model by EM

Figure 4.3: MHTC process diagram. See text for description of each step. Steps 1–9 indicate core clustering and hypothesis tracking procedures, whereas steps i–iii are for data acquisition only.

Remark 4.2. Comparing Figure 4.1 and Figure 4.2, it appears that introducing the clustering process into the MHT framework (via the model hypotheses) has exacerbated the combinatorial increase in the number of hypotheses. Indeed, at every time increment, the number of hypotheses now expands twice in MHTC. However, because the numbers of measurements and targets are usually small (typically 1–4), the data association hypotheses can be formed and evaluated without much cost; on the other hand, each model hypothesis is relatively expensive, as it must be clustered. Also, although a relatively small fraction, $(\frac{1}{L\bar{M}})$, of the generated hypotheses survive to the next step, generating all $(L^2\bar{M})$ guarantees these are the best available, notwithstanding the possibility that, for example, the $(L+1)$th global hypothesis from time $(k-1)$ could have spawned a hypothesis that would be in the top L hypotheses at time k. Selection of the parameter L, then, must balance the expansion of hypotheses with the aim of ensuring that all viable ones are maintained; in spike sorting applications, the hypothesis probability usually falls drastically over the first few hypothesis and thus good results may be obtained even with small L. ◊

4.2.3 Overview of the MHTC Process

This section walks through the MHTC process of the combined clustering and multiple hypothesis tracking, as illustrated in Figure 4.3. Steps 2–6 are similar to the procedure detailed in Chapter 3, but some of these steps require a reformulation in the context of the MHT framework and the notations and models provided in Section 4.2.1.

Step 1. Given Ω^{k-1} (all the measurements and hypotheses up to T_{k-1}), the first step is to predict the measured locations (means) of the target neurons. Based on the system model in Section 4.2.1.2, the Kalman Filter equations for prediction may be written as follows for the jth neuron target, given the estimates from the last step's measurement update (see Step 9):

$$\hat{x}_j^{k|k-1} = F^{k-1}\,\hat{x}_j^{k-1|k-1} \qquad \text{state prediction} \tag{4.7}$$

$$\Lambda_j^{k|k-1} = F^{k-1}\,\Lambda_j^{k-1|k-1}{F^{k-1}}^T + Q^{k-1} \qquad \text{state prediction covariance} \tag{4.8}$$

$$\hat{\mu}_j^{k|k-1} = H^k\,\hat{x}_j^{k|k-1} \qquad \text{measurement prediction}\,. \tag{4.9}$$

Step 2. For every parent hypothesis in Ω^{k-1}, a set of mixture model classes $\{\mathcal{M}_m\}$ is generated, which will be used to cluster the current data Y^k. The same mixture of Gaussians (with the outlier distribution for component $g = 0$) described in Section 3.3.1 is used, with corresponding likelihood function:

$$p\big(Y^k|\Theta_m^k, \mathcal{M}_m\big) = \prod_{i=1}^{N}\left(\pi_0^k f_0\big(y_i|\theta_0^k\big) + \sum_{g=1}^{G_m}\pi_g^k f_{\mathcal{N}}\big(y_i|\mu_g^k, \Sigma_g^k\big)\right). \tag{4.10}$$

Recall that in Chapter 3, the procedure clustered every model class in a predetermined set $m = 1, ..., \bar{M}$, which most importantly captured a range of numbers of clusters $G_m = 1, ..., G_{\max}$. If L hypotheses have survived from interval $(k-1)$, this approach would require $L\bar{M}$ model optimizations. Many of these models will be very unlikely; for example, the probability of changing from one cluster on interval T_{k-1} to four clusters on interval T_k may be very low (depending on the parameters λ_ν and λ_ϕ). Thus, to save the computation inherent in optimizing and evaluating "dead end" model classes, the probability of each model class arising from its parent hypothesis is calculated and then tested against a threshold β:

$$P\big(\mathcal{M}_m|H_{\rho(m)}^{1:k-1}, Y^{1:k-1}\big) > \beta\,. \tag{4.11}$$

An expression for this probability is derived in Appendix 4.A.1. Model classes that do not pass this thresholding test are discarded; surviving model classes become the model hypotheses.

Step 3. In each model hypothesis, a prior on the model parameters is constructed in a way similar to Section 3.3.2:

$$\begin{aligned} p\big(\Theta_m^k \,|\, Y^{1:k-1}, \mathcal{M}_m\big) &= C \prod_{g=1}^{G_m} \sum_{j=0}^{\hat{G}^{k-1}} \omega_j^k \, f_j\big(\mu_g^k \,|\, \psi_j^{k|k-1}\big) \\ &= C \prod_{g=1}^{G_m} \left(\frac{\omega_0^k}{V} + \sum_{j=1}^{\hat{G}^{k-1}} \omega_j^k f_{\mathcal{N}}\big(\mu_g^k \,|\, \hat{\mu}_j^{k|k-1}, S_j^k\big) \right) . \end{aligned} \tag{4.12}$$

Recall from Definition 3.1 that the prior on each mean is a mixture of Gaussians centered on the predicted measurements, with a uniform component ($j = 0$) to capture new neurons and false clusters, and that C is a constant representing the diffuse priors on the other elements of Θ_m^k (i.e., π_g^k and Σ_g^k). The symbol $\psi_j^{k|k-1}$ denotes the parameters for the jth mixture component; $\hat{\mu}_j^{k|k-1}$ and S_j^k may now be more formally (and more generally) defined as, respectively, the predicted measurement (4.9) and *innovation covariance* from the Kalman Filter:

$$S_j^k = H^k \, \Lambda_j^{k|k-1} \, {H^k}^T + R^k \, . \tag{4.13}$$

Steps i–iii. These steps collectively acquire and prepare new neural data for the clustering and tracking procedures. As before, the spike waveforms are recorded from an electrode over a sampling interval T_k, extracted from the electrode's voltage trace, and projected onto a d-dimensional feature space. The clustering and data association procedures operate on the feature space representation of the spikes, $Y^k = \{y_i\}$, as the reduced dimensionality greatly simplifies the clustering computation.

Steps 4–5. For each model hypothesis, model parameters are seeded and optimized using the clustering procedure detailed in Section 3.3. Note that, like the prior, the seeding method should use the predicted locations of the clusters $\hat{\mu}_j^{k|k-1}$ (4.9) rather than simply their locations in the previous step. The EM process maximizes the posterior, which now is written to include the parent hypothesis:[4]

$$p\big(\Theta_m^k \,|\, \mathcal{M}_m, H_{\rho(m)}^{1:k-1}, Y^{1:k}\big) \propto p\big(Y^k \,|\, \Theta_m^k, \mathcal{M}_m\big) \, p\big(\Theta_m^k \,|\, \mathcal{M}_m, H_{\rho(m)}^{1:k-1}, Y^{1:k-1}\big) \, . \tag{4.14}$$

Note the factors on the right-hand side are the same as Equations (4.10) and (4.12). Step 5 thus provides the MAP parameter estimates $\hat{\Theta}_m^k$ and the expectation of the spike membership indicators $\hat{Z}$ and of the cluster association indicators $\hat{\mathcal{Z}}$. Section 4.3.3 details how $\hat{\mathcal{Z}}$ can be used to generate data association hypothesis probabilities.

Step 6. The evidence of each model hypothesis, $p\big(Y^k \mid \mathcal{M}_m, H_{\rho(m)}^{1:k-1}, Y^{1:k-1}\big)$, may now be calculated, preferably via Laplace's method (3.18). (If necessary to reduce the computational burden, use of the AIC, BIC, or other approximation may again be substituted for Laplace's method.) Because

[4] In the likelihood expression, the conditioning on $Y^{1:k-1}$ and $H_{\rho(m)}^{1:k-1}$ is unnecessary.

the next step results in an expansion of the number of hypotheses from every model hypothesis, one may now prune extremely unlikely models for computational saving using the model evidence. However, because the spike sorting application involves few targets and measurements, the results shown in this thesis do not use such a reduction of model hypotheses at this step.

Step 7. The core step in MHT generates the data association hypotheses, $h_l = \{\tau_l, \nu_l, \phi_l\}$. As noted earlier, it is desirable to utilize Murty's L-best ranked linear assignment algorithm to produce only the best data association hypotheses from each parent cluster hypothesis, obviating the need for full enumeration of all possible data associations. This technique requires careful formulation of the probability calculations so that Murty's algorithm can operate on a matrix of assignment likelihoods $\mathbf{A}$. Section 4.3 derives an appropriate expression of $\mathbf{A}$ and describes the hypothesis generation in more detail.

Step 8. Suppose that a total of $\tilde{M}$ model hypotheses exist at this time, each of which has now spawned L data association hypotheses. From the $(\tilde{M}L)$ hypotheses that have been generated, the most probable L global hypotheses must be selected. Evaluating each model and data association hypothesis together with its parent hypothesis $H_{\rho(l)}^{1:k-1}$, the probability of each new global hypothesis $P\big(H_l^{1:k} \mid Y^{1:k}\big)$ can be calculated, as detailed in Section 4.3.1. This step provides the set of best global hypotheses in Ω^k.

Step 9. Finally, for each $H_l^{1:k} \in \Omega^k$, the hypothesized data associations h_l, along with the optimized parameters Θ_m^k of the corresponding model hypothesis, are used to update the Kalman Filter:

$$K_j^k = \Lambda_j^{k|k-1} \, H^{k^T} \left(H^k \, \Lambda_j^{k|k-1} \, H^{k^T} + R_j^k \right)^{-1} \qquad \text{Kalman filter gain} \tag{4.15}$$

$$\hat{x}_j^{k|k} = \hat{x}_j^{k|k-1} + K_j^k \, (\hat{\mu}_j^k - \hat{\mu}_j^{k|k-1}) \qquad \text{updated state estimate} \tag{4.16}$$

$$\Lambda_j^{k|k} = (I - K_j^k \, H^k) \Lambda_j^{k|k-1} \qquad \text{updated state covariance} \, . \tag{4.17}$$

4.3 Probability Calculations

This section develops expressions for the probabilities necessary for the MHTC algorithm and formulates the data association problem for use of Murty's L-best assignment algorithm.

4.3.1 Global Hypothesis Probability

The key probability to be determined for MHTC is that of a global hypothesis given all collected data, $P\big(H_l^{1:k} \mid Y^{1:k}\big)$, the basis of the final hypothesis selection for time k (in Step 9). The expression for this probability includes all relevant measures about the parent hypothesis, model hypothesis, and data association hypothesis.

Proposition 4.1. *The* ***global hypothesis probability*** *given the data* $Y^{1:k}$ *may be expressed as*

$$P\big(H_l^{1:k}|Y^{1:k}\big) \approx \frac{1}{\mathcal{C}} \frac{\mathcal{P}_{1,l}\,\mathcal{P}_{2,l}}{\sum_{n\in\Gamma}\mathcal{P}_{1,n}\,\mathcal{P}_{2,n}}\,\mathcal{P}_3\,\mathcal{P}_4\,\mathcal{P}_5\,, \tag{4.18}$$

where $\mathcal{C}$ *is a normalization constant,* Γ *is the set of indices of all legal data association hypotheses given the model hypothesis* $\mathcal{M}_{m(l)}$, *and the other factors are listed below:*

$\mathcal{P}_{1,l} \equiv p\big(\{\hat{\mu}_g^k\}_{g=1}^{G_{m(l)}}\,\vert\,H_l^{1:k}, Y^{1:k-1}, \hat{\mu}_g^k \in \hat{\Theta}_{m(l)}^k\big)$	*likelihood of cluster means under* h_l
$\mathcal{P}_{2,l} \equiv p\big(H_l^k\,\vert\,H_{\rho(l)}^{1:k-1}, Y^{1:k-1}\big)$	*joint hypothesis prior*
$\mathcal{P}_3 \equiv P\big(\mathcal{M}_{m(l)}\,\vert\,H_{\rho(l)}^{1:k-1}, Y^{1:k-1}\big)$	*model hypothesis prior*
$\mathcal{P}_4 \equiv p\big(Y^k\,\vert\,\mathcal{M}_{m(l)}, H_{\rho(l)}^{1:k-1}, Y^{1:k-1}\big)$	*model evidence*
$\mathcal{P}_5 \equiv P\big(H_{\rho(l)}^{1:k-1}\,\vert\,Y^{1:k-1}\big)$	*probability of parent hypothesis.*

Recall that $H_l^k = \{\mathcal{M}_{m(l)}, h_l\}$ *is a joint hypothesis (of both the model and data association hypotheses) and* $H_{\rho(l)}^{1:k-1}$ *is the parent global hypothesis of* H_l^k. ♦

Proof. First, $P\big(H_l^{1:k}|Y^{1:k}\big)$ may be decomposed using Bayes' Rule:

$$P\big(H_l^{1:k}|Y^{1:k}\big) = \frac{1}{\mathcal{C}}\,p\big(Y^k|H_l^{1:k}, Y^{1:k-1}\big)\,P\big(H_l^{1:k}|Y^{1:k-1}\big)\,, \tag{4.19}$$

where $\mathcal{C} = p\big(Y^k \mid Y^{1:k-1}\big)$ is independent of a particular hypothesis[5]. Recalling that $H_l^{1:k} = \{h_l, \mathcal{M}_m, H_{\rho(l)}^{1:k-1}\}$, the last factor on the right-hand side is easily broken down via the chain rule:

$$\begin{aligned}
P\big(H_l^{1:k}|Y^{1:k-1}\big) &= P\big(H_l^k|H_{\rho(l)}^{1:k-1}, Y^{1:k-1}\big)\,P\big(H_{\rho(l)}^{1:k-1}|Y^{1:k-1}\big)\\
&= P\big(h_l|\mathcal{M}_{m(l)}, H_{\rho(l)}^{1:k-1}, Y^{1:k-1}\big)\,P\big(\mathcal{M}_{m(l)}|H_{\rho(l)}^{1:k-1}, Y^{1:k-1}\big)\,P\big(H_{\rho(l)}^{1:k-1}|Y^{1:k-1}\big)\\
&\triangleq P\big(h_l|\mathcal{M}_{m(l)}, H_{\rho(l)}^{1:k-1}, Y^{1:k-1}\big)\,\mathcal{P}_3\,\mathcal{P}_5\,.
\end{aligned} \tag{4.20}$$

[5] At this point, the normalization constant $\mathcal{C}$ is theoretically a sum over all feasible hypotheses. In practice, $\mathcal{C}$ never requires explicit calculation; if normalization is desired, $\mathcal{C}$ may be considered a sum over the L hypotheses surviving after Step 8, as those are the only ones under consideration.

Denoting $\cdot = \{\mathcal{M}_{m(l)}, H_{\rho(l)}^{1:k-1}, Y^{1:k-1}\}$ for convenience, the other factor of (4.19) is

$$\begin{aligned}
p\big(Y^k | H_l^{1:k}, Y^{1:k-1}\big) &= p\big(Y^k | h_l, \cdot\big) && \text{(4.21a)}\\
&= P\big(h_l | Y^k, \cdot\big) \, \frac{p\big(Y^k | \cdot\big)}{P\big(h_l | \cdot\big)} && \text{(4.21b)}\\
&\approx P\big(h_l | \hat{\Theta}_m^k, \cdot\big) \, \frac{p\big(Y^k | \cdot\big)}{P\big(h_l | \cdot\big)} && \text{(4.21c)}\\
&= \frac{p\big(\hat{\Theta}_{m(l)}^k | h_l, \cdot\big) \, P\big(h_l | \cdot\big)}{\sum_{n \in \Gamma} p\big(\hat{\Theta}_{m(l)}^k | h_n, \cdot\big) \, P\big(h_n | \cdot\big)} \, \frac{p\big(Y^k | \cdot\big)}{P\big(h_l | \cdot\big)} && \text{(4.21d)}\\
&= \frac{p\big(\hat{\Theta}_{m(l)}^k | h_l, \cdot\big) \, p\big(h_l, \mathcal{M}_{m(l)} | H_{\rho(l)}^{1:k-1}, Y^{1:k-1}\big)}{\sum_{n \in \Gamma} p\big(\hat{\Theta}_{m(l)}^k | h_n, \cdot\big) \, p\big(h_n, \mathcal{M}_{m(l)} | H_{\rho(l)}^{1:k-1}, Y^{1:k-1}\big)} \, \frac{p\big(Y^k | \cdot\big)}{P\big(h_l | \cdot\big)} && \text{(4.21e)}\\
&\triangleq \frac{\mathcal{P}_{1,l} \, \mathcal{P}_{2,l}}{\sum_{n \in \Gamma} \mathcal{P}_{1,n} \, \mathcal{P}_{2,n}} \, \frac{\mathcal{P}_4}{P\big(h_l | \mathcal{M}_{m(l)}, H_{\rho(l)}^{1:k-1}, Y^{1:k-1}\big)} \, . && \text{(4.21f)}
\end{aligned}$$

Above, the equalities (4.21b) and (4.21d) are obtained by simple use of Bayes' Rule[6]. In (4.21c), Laplace's method has been applied to essentially replace the conditioning on Y^k with conditioning on $\hat{\Theta}_m^k$; the details of this approximation may be found in Appendix A.3. Intuitively, this step is important because the data association hypothesis h_l is evaluated based on the cluster means in $\hat{\Theta}_m^k$ rather than on the spike observations Y^k directly. In (4.21e), the numerator and denominator have been multiplied by $P\big(\mathcal{M}_{m(l)} | H_{\rho(l)}^{1:k-1}, Y^{1:k-1}\big)$[7], and it should be recognized that

$$P\big(h_l | \mathcal{M}_{m(l)}, H_{\rho(l)}^{1:k-1}, Y^{1:k-1}\big) \, P\big(\mathcal{M}_{m(l)} | H_{\rho(l)}^{1:k-1}, Y^{1:k-1}\big) = P\big(h_l, \mathcal{M}_{m(l)} | H_{\rho(l)}^{1:k-1}, Y^{1:k-1}\big) \triangleq \mathcal{P}_{2,l} \, .$$

Note that $m(l) = m(n)$ and $\rho(l) = \rho(n) \; \forall n \in \Gamma$, since Γ includes only the data association hypotheses generated from $\mathcal{M}_{m(l)}$, so all parent model hypotheses and parent global hypotheses are identical—this fact ensures that $P\big(\mathcal{M}_{m(l)} \,|\, H_{\rho(l)}^{1:k-1}, Y^{1:k-1}\big)$ may be combined as above in every term of the sum. Finally, (4.21f) rewrites the equation with the notation defined in Proposition 4.1, noting that the PDFs for $\{\pi_g\}_g$ and $\{\Sigma_g\}_g$ in $p\big(\hat{\Theta}_{m(l)}^k | h_n, \cdot\big)$ have no dependence on h_n and thus cancel in the numerator and denominator.

Substituting (4.20) and (4.21f) into (4.19) results in:

$$\begin{aligned}
P\big(H_l^{1:k} | Y^{1:k}\big) &\approx \frac{1}{\mathcal{C}} \, \frac{\mathcal{P}_{1,l} \, \mathcal{P}_{2,l}}{\sum_{n \in \Gamma} \mathcal{P}_{1,n} \, \mathcal{P}_{2,n}} \, \mathcal{P}_4 \frac{P\big(h_l | \mathcal{M}_{m(l)}, H_{\rho(l)}^{1:k-1}, Y^{1:k-1}\big)}{P\big(h_l | \mathcal{M}_{m(l)}, H_{\rho(l)}^{1:k-1}, Y^{1:k-1}\big)} \, \mathcal{P}_3 \, \mathcal{P}_5\\
&= \frac{1}{\mathcal{C}} \, \frac{\mathcal{P}_{1,l} \, \mathcal{P}_{2,l}}{\sum_{n \in \Gamma} \mathcal{P}_{1,n} \, \mathcal{P}_{2,n}} \, \mathcal{P}_3 \, \mathcal{P}_4 \, \mathcal{P}_5 \, . \qquad \square
\end{aligned}$$

[6] In (4.21d), the denominator is written as a sum rather than as $p\big(\hat{\Theta}_{m(l)}^k | \mathcal{M}_{m(l)}, H_{\rho(l)}^{1:k-1}, Y^{1:k-1}\big)$ to avoid confusion, because this latter expression looks identical to the parameter prior (4.12). These PDF values are different, since the denominator of (4.21d) assumes an exclusivity principle in uniquely assigning measurements and targets, whereas in the parameter prior no restrictions prevent multiple "assignments," as it is a mixture model.

[7] This step provides a more convenient form of the equations used to determine the most likely data association hypotheses in Section 4.3.2.

Equation (4.18), combined with the expressions for its components, is a key contribution of this chapter, providing the mechanism for evaluating a global hypothesis given all collected data. $\mathcal{P}_{1,l}$ and $\mathcal{P}_{2,l}$ are together used to compute the plausibility of particular data association hypotheses, as derived and discussed in Section 4.3.2 and Section 4.3.3. The model hypothesis prior, $\mathcal{P}_3$, is developed in Appendix 4.A.1 and is also used in Step 2 of the MHTC algorithm to prune unlikely model classes before the parameter optimization process. The expression for $\mathcal{P}_4$, the model evidence, has already been discussed in Section 3.3.5. $\mathcal{P}_5$ is simply the global hypothesis probability of the parent hypothesis, calculated after the previous interval T_{k-1}.

Remark 4.3. The careful reader may have noticed that (4.18) requires the normalizing sum over all legal data association hypotheses from a particular model class (the set denoted by Γ), but it is stated in Section 4.2 that only the L best hypotheses are generated. However, the contribution of the hypotheses worse than the Lth best is often negligible for spike sorting applications, even for small L. Thus, normalizing using only the L best from Γ is a very good approximation. $\lozenge$

4.3.2 Data Association Hypothesis Plausibility

When generating the L-best data association hypotheses $\{h_l\}$ from each model hypothesis $\mathcal{M}_m$ in Step 7 of the MHTC algorithm, only the product $(\mathcal{P}_{1,l}\,\mathcal{P}_{2,l})$ needs to be examined, as all other factors in (4.18) are identical for a given model hypothesis. Thus, this product is referred to as the *data association hypothesis plausibility*—it is proportional to the (posterior) probability but is technically neither a likelihood nor a normalized probability.

$\mathcal{P}_{1,l}$ is the likelihood of the data association hypothesis $h_l = \{\tau_l, \nu_l, \phi_l\}$ given the measurements $\{\hat{\mu}_g^k\}_{g=1}^{G_{m(l)}} \in \hat{\Theta}_{m(l)}^k$. Based on the probability models defined in Section 4.2.1.3,

$$\begin{aligned} \mathcal{P}_{1,l} &\equiv p\big(\{\hat{\mu}_g^k\}_{g=1}^{G_{m(l)}} \,|\, H_l^{1:k}, Y^{1:k-1}, \hat{\mu}_g^k \in \hat{\Theta}_{m(l)}^k\big) \\ &= \Bigg[\prod_{(g,j)\in\tau_l} f_{\mathcal{N}}\big(\hat{\mu}_g^k \,|\, \hat{\mu}_j^{k|k-1}, S_j^k\big)\Bigg] \Bigg[\prod_{g\in\nu_l} \frac{1}{V}\Bigg] \Bigg[\prod_{g\in\phi_l} \frac{1}{V}\Bigg] . \end{aligned} \tag{4.22}$$

The joint hypothesis prior $\mathcal{P}_{2,l}$ gives the probability of a particular joint hypothesis H_l^k without knowledge of the new measurements. This probability is based on the number of measurements of each type and which existing neuron targets have been tracked (see the derivation in Appendix 4.A.2):

$$\begin{aligned} \mathcal{P}_{2,l} &\equiv P\big(H_l^k \,|\, H_{\rho(l)}^{1:k-1}, Y^{1:k-1}\big) \\ &= A_m \Bigg[\prod_{j=1}^{N_t} \left(\mathsf{P}_{\mathsf{d},j}\right)^{\delta_{j,l}} \left(1-\mathsf{P}_{\mathsf{d},j}\right)^{1-\delta_{j,l}}\Bigg] \left(\lambda_\nu\right)^{N_\nu} \left(\lambda_\phi\right)^{N_\phi} , \end{aligned} \tag{4.23}$$

where A_m is a constant for each model hypothesis and N_t is the number of targets in hypothesis $H_{\rho(l)}^{1:k-1}$. Recall from Section 4.2.1.3 that the hypothesized numbers of new neurons N_ν and false clusters N_ϕ are Poisson distributed with respective rates λ_ν and λ_ϕ, and $\mathsf{P}_{\mathsf{d},j}$ is the probability of detecting the jth existing target (indicated by the binary-valued $\delta_{j,l}$).

Combining (4.22) and (4.23), the best data association hypotheses from a particular model may be generated using the quantity:

$$\mathcal{P}_{1,l}\,\mathcal{P}_{2,l} = A_m \left[\prod_{j=1}^{N_t} \left(\mathsf{P}_{\mathsf{d},j}\right)^{\delta_{j,l}}\ \left(1-\mathsf{P}_{\mathsf{d},j}\right)^{1-\delta_{j,l}}\right]\left[\prod_{(g,j)\in\tau_l} f_{\mathcal{N}}\left(\hat{\mu}_g^k|\hat{\mu}_j^{k|k-1}, S_j^k\right)\right]\left(\frac{\lambda_\nu}{V}\right)^{N_\nu}\left(\frac{\lambda_\phi}{V}\right)^{N_\phi}. \tag{4.24}$$

4.3.3 Formulation for Hypothesis Generation via Murty's Algorithm

As noted earlier, by utilizing Murty's algorithm for ranked linear assignment, one can avoid enumerating all legal data association hypotheses and instead generate only the L-best hypotheses directly. This section shows how to formulate the data association problem such that Murty's algorithm may be applied and additionally demonstrates how the necessary probabilities have largely been calculated in the clustering procedure, further establishing a natural connection between MHTC's clustering and tracking procedures.

As previously recognized by Danchick and Newnam [72], the problem of generating legal hypotheses by mapping current measurements to known targets can be thought of as a problem of weighted bipartite graph matching, with the group of measurements and the group of targets as the two disjoint sets. The primary goal of this section is to construct a cost matrix for the corresponding linear assignment problem (which must also include the notions of new targets and false clusters), where the total cost of an assignment hypothesis is equivalent to (4.24).

Let $\mathbf{A} \in \mathbb{R}^{G_m \times N_t + 2G_m}$ be the *data association matrix*, where the rows are the G_m current measurements (cluster means) and the columns represent the N_t existing targets, G_m possible new targets, and G_m possible false clusters. (Since each measurement may be independently assigned as a new target or false clusters, and only unique assignments are allowed, new targets and false clusters each require a column for each measurement.) The elements of this matrix, $[a_{gj}]$, essentially define the likelihood of assigning the gth measurement to the jth target.

Proposition 4.2. *Define the data association matrix as*

$$\mathbf{A} \equiv [a_{gj}] \equiv \left[\begin{array}{ccccc} a_{11} & \cdots & a_{1N_t} & & \\ \vdots & \ddots & \vdots & \operatorname{diag}(\alpha_1^\nu, \ldots, \alpha_{G_m}^\nu) & \operatorname{diag}(\alpha_1^\phi, \ldots, \alpha_{G_m}^\phi) \\ a_{G_m 1} & \cdots & a_{G_m N_t} & & \end{array}\right], \tag{4.25}$$

where

$$\begin{aligned} a_{gj} &= \frac{1}{1-\mathsf{P}_{\mathsf{d},j}}\hat{\zeta}_{gj} && g=1,\ldots,G_m\,;\, j=1,\ldots,N_t \\ \alpha_g^{\nu} &= \frac{\lambda_\nu}{\lambda_\nu+\lambda_\phi}\hat{\zeta}_{g0} && g=1,\ldots,G_m \\ \alpha_g^{\phi} &= \frac{\lambda_\phi}{\lambda_\nu+\lambda_\phi}\hat{\zeta}_{g0} && g=1,\ldots,G_m\,, \end{aligned}$$

where, as defined in Section 3.3.3,

$$\hat{\zeta}_{gj} = \frac{\omega_j^k f_j\big(\hat{\mu}_g^k|\psi_j^{k|k-1}\big)}{\sum_{l=0}^{\hat{G}^{k-1}} \omega_l^k f_l\big(\hat{\mu}_g^k|\psi_l^{k|k-1}\big)}\,. \tag{4.26}$$

Then the probability of a legal data association hypothesis h_l is proportional to the product of the elements of $\mathbf{A}$ assigned by h_l; that is,

$$\mathcal{P}_{1,l}\,\mathcal{P}_{2,l} = \mathcal{D}\prod_{(g,j)\in\tilde{h}_l} a_{gj} \quad , \tag{4.27}$$

where $\mathcal{D}$ is a constant for each model hypothesis and $\tilde{h}_l$ is simply another way of labeling the assignments in h_l:

$$\tilde{h}_l \equiv \tau_l \cup \{(g, g+N_t) : g\in\nu_l\} \cup \{(g, g+N_t+G_m) : g\in\phi_l\}\,.$$

♦

Proof. Recall from Definition 3.1 in Section 3.3.3 that $f_0 = \frac{1}{V}$ and $f_j = f_{\mathcal{N}}$ for $j=1,\ldots,\hat{G}^{k-1}$, and that

$$\omega_j^k = \begin{cases} \frac{1}{c}\lambda_0 & j=0 \\ \frac{1}{c}\mathsf{P}_{\mathsf{d},j} & j=1,\ldots,\hat{G}^{k-1} \end{cases}\,,$$

where $\lambda_0 \equiv \lambda_\nu + \lambda_\phi$. Denoting the denominator in Equation 4.26 by

$$b_g \equiv \sum_{l=0}^{\hat{G}^{k-1}} \omega_l^k\, f_l\big(\hat{\mu}_g^k|\psi_l^k\big)\,, \tag{4.28}$$

the expected cluster association indicators may be written

$$\hat{\zeta}_{gj} = \begin{cases} \frac{\lambda_\nu+\lambda_\phi}{b_g}\frac{1}{V} & j=0 \\ \frac{\mathsf{P}_{\mathsf{d},j}}{b_g} f_{\mathcal{N}}\big(\hat{\mu}_g^k|\hat{\mu}_j^{k|k-1}, S_j^k\big) & j=1,\ldots,\hat{G}^{k-1} \end{cases}\,.$$

Using the above set of definitions and equalities, Equation (4.24) can be rearranged:

$$\begin{aligned}
\mathcal{P}_{1,l}\,\mathcal{P}_{2,l} &= A_m \left[\prod_{j=1}^{N_t} \left(\mathsf{P}_{\mathsf{d},j}\right)^{\delta_{j,l}} \left(1-\mathsf{P}_{\mathsf{d},j}\right)^{1-\delta_{j,l}}\right] \left[\prod_{(g,j)\in\tau_l} f_{\mathcal{N}}\big(\hat{\mu}_g^k \,|\, \hat{\mu}_j^{k|k-1}, S_j^k\big)\right] \left[\prod_{g\in\nu_l} \frac{\lambda_\nu}{V}\right] \left[\prod_{g\in\phi_l} \frac{\lambda_\phi}{V}\right] \\
&= A_m \left[\prod_{g=1}^{G_m} b_g\right] \left[\prod_{j=1}^{N_t} 1-\mathsf{P}_{\mathsf{d},j}\right] \left[\prod_{(g,j)\in\tau_l} \frac{\mathsf{P}_{\mathsf{d},j}\, f_{\mathcal{N}}\big(\hat{\mu}_g^k \,|\, \hat{\mu}_j^{k|k-1}, S_j^k\big)}{b_g\,(1-\mathsf{P}_{\mathsf{d},j})}\right] \left[\prod_{g\in\nu_l} \frac{\lambda_\nu}{b_g\,V}\right] \left[\prod_{g\in\phi_l} \frac{\lambda_\phi}{b_g\,V}\right] \\
&= A_m B_m C_{\rho(l)} \left[\prod_{(g,j)\in\tau_l} \frac{1}{1-\mathsf{P}_{\mathsf{d},j}} \hat{\zeta}_{gj}\right] \left[\prod_{g\in\nu_l} \frac{\lambda_\nu}{\lambda_\nu+\lambda_\phi} \hat{\zeta}_{g0}\right] \left[\prod_{g\in\phi_l} \frac{\lambda_\phi}{\lambda_\nu+\lambda_\phi} \hat{\zeta}_{g0}\right] \\
&= A_m B_m C_{\rho(l)} \left[\prod_{(g,j)\in\tilde{h}} a_{gj}\right], \qquad (4.29)
\end{aligned}$$

where $B_m \equiv \prod_{g=1}^{G_m} b_g$, the product of the terms in (4.28), depends only on the model hypothesis and $C_{\rho(l)} \equiv \prod_{j=1}^{N_t}(1-\mathsf{P}_{\mathsf{d},j})$ is constant for each parent hypothesis. Thus, $\mathcal{D} \triangleq A_m B_m C_{\rho(l)}$ is constant for all data association hypotheses generated from the same model hypothesis, and (4.29) is equivalent to (4.27). □

To employ Proposition 4.2 in Murty's algorithm, use the data association matrix $\mathbf{A} \equiv [a_{gj}]$ (4.25) to define the linear assignment cost matrix $\mathbf{A}^* = -[\log a_{gj}]$, where the elements of $\mathbf{A}$ that are zero are instead replaced by a suitably large upper bound. Murty's algorithm may then be applied to L^* to generate the L-best data association hypotheses for every model hypothesis $\mathcal{M}_m$.

Proposition 4.2 also implies that the global hypothesis probability (4.18) may be rewritten:

$$P\big(H_l^{1:k} | Y^{1:k}\big) \approx \frac{1}{\mathcal{C}} \frac{\prod_{(g,j)\in\tilde{h}_l} a_{gj}}{\sum_{n\in\Gamma}\prod_{(g,j)\in\tilde{h}_n} a_{gj}}\,\mathcal{P}_3\,\mathcal{P}_4\,\mathcal{P}_5\,, \qquad (4.30)$$

so that the assignment costs calculated during the data association hypothesis generation step are used directly to evaluate the global hypothesis probability. Note that the constant $\mathcal{D}$ never requires calculation, as it cancels from the numerator and denominator.

4.4 Implementation

This section provides details of the implementation that produced the results to be seen in Section 4.5. These details fall into two general categories: hypothesis management strategies that maintain computational efficiency and model and parameter choices that suit the particular spike sorting application.

4.4.1 Hypothesis Management

While MHT is generally recognized as the preferred MTT solution, the exponential increases in memory and processing requirements of the ideal case (maintaining all hypotheses for all time) make that implementation intractable. Thus, many methods have been developed to increase the computational efficiency, such as gating, hypothesis ratio pruning, N-scan pruning, track-oriented hypothesis management, and spatially disjoint hypothesis trees[8] [39, 40, 76]. The application of Murty's ranked linear assignment algorithm to generate only the L-best hypotheses was a significant advancement and reduced the need for these previous techniques [37]. Generally, the above methods result in a sub-optimal approach in which many data association hypotheses are either eliminated or never considered, thus negating the possibility that these hypotheses will later appear as the best one in MHT's delayed decision making. However, these investigators have shown that excellent results may still be obtained under such assumptions.

Two key hypothesis management strategies implemented in this work have already been discussed: generation of only the L-best hypotheses via Murty's algorithm (in Step 7 of MHTC) and pruning model classes before EM optimization by a probability threshold β (in Step 2). Another technique is required to delete obsolete tracks, so that only targets that are still likely to generate measurements are considered. For this purpose, a track is removed from a hypothesis after K_{miss} consecutive missed detections [37].

A hypothesis may be considered a "duplicate" if its assignments are identical to another hypothesis' for the most recent K_{miss} intervals. In this case, the global hypothesis with the lesser probability has a negligible chance at "overtaking" the leading hypothesis with a similar assignment history, and is thus deleted. This method is very similar to the commonly-implemented N-scan pruning (with $N = K_{\text{miss}}$), which examines the portions of the hypothesis tree branching from the nodes at time $(k - N)$ and keeps only the branch has the greatest probability. N-scan pruning thus assumes that any ambiguity at time $(k - N)$ is resolved by time k and makes an irrevocable decision to make the most probable node from time $(k - N)$ the new root node. When using the L-best hypotheses approach, not only does the removal of duplicate hypotheses save computation, but it also is important for the diversity of possible hypotheses. If duplicate hypotheses are not removed, then all L hypotheses being maintained will often become very similar, and other possibilities, which might later prove to be more probable, would be prematurely discarded.

[8] *Gating* refers to a constraint under which only those measurements who lie within some distance of the target's predicted location are considered for association, eliminating unlikely pairings. *Spatially disjoint hypothesis trees* define groups of neighboring targets whose measurement assignments may be considered separately because they are sufficiently spatially distinct (e.g., the gates of targets in different groups do not overlap). These target groups are commonly referred to as *clusters*, but the use of the term in MHT literature is unrelated to the use in this thesis, in which a cluster is a set of observations that define a single measurement.

4.4.2 Model and Parameter Choices

To complete the model described in Section 4.2.1.2, the form of the state vector and the matrices F, H, Q, and R must be chosen. For the results presented in Section 4.5, the state is simply the neuron location (i.e., the neuron's mean spike waveform during interval T_k), $x_j^k \in \mathbb{R}^{d_\mathsf{w}}$, where d_w is the dimension of the full waveform. Note that while the measurements are expressed in feature space, $\mu_j^k \in \mathbb{R}^d$, the neurons are tracked in the full waveform space because of the variation in PCA coordinates across sample intervals (see Remark 4.1). Thus,

$$F^k = \mathbf{I}_{\mathbf{d}_\mathsf{w}} \, \forall k \quad \text{and} \quad H^k = \mathbf{B}^k \, ,$$

where $\mathbf{I}_{\mathbf{d}_\mathsf{w}}$ is the d_w-dimensional identity matrix and $\mathbf{B}^k \in \mathbb{R}^{d \times d_\mathsf{w}}$ is the matrix representation of the PCA basis $\mathcal{B}^k$ (i.e., the rows are the first d eigenvectors of covariance matrix of Y^k). As in Chapter 3, the process noise covariance Q^k is empirically determined to account for waveform non-stationarity, and the measurement noise is set to $R_j^k = \frac{1}{n_j}\Sigma_j^k$.

Table 4.1 lists the parameters in the MHTC algorithm for the results that are summarized in the following section. The detection probability $\mathsf{P}_{\mathsf{d},j}$ is a function of the target neuron's firing rate, r_j^k; the function is constructed so that the detection probability is minimally $\mathsf{P}_{\mathsf{d},\min}$ and increases with r_j^k to asymptotically approach $\mathsf{P}_{\mathsf{d},\max}$:

Table 4.1: Parameter Choices in MHTC Implementation

Parameter	Value
L	8
λ_ν	0.01
λ_ϕ	0.015
$\mathsf{P}_{\mathsf{d},j}$	function of neuron firing rate
$\mathsf{P}_{\mathsf{d},\max}$	0.98
$\mathsf{P}_{\mathsf{d},\min}$	0.75
β	0.001
$G_{\max}$	4
K_{miss}	5

$$\mathsf{P}_{\mathsf{d},j}^{k+1}(r_j^k) = \frac{1}{2}\left(q(r_j^k) + \mathsf{P}_{\mathsf{d},j}^k\right) \, , \qquad q(r_j^k) \equiv \mathsf{P}_{\mathsf{d},\max} - \frac{1}{a r_j^k + b} \, ,$$

where the parameters $a = 0.6$ and $b = 4$ have been chosen to scale and shift the function $q(r_j^k)$ such that $q(0) = \mathsf{P}_{\mathsf{d},\min}$ and $q(10) = \frac{1}{2}(\mathsf{P}_{\mathsf{d},\max} + \mathsf{P}_{\mathsf{d},\min})$.

4.5 Experimental Results

The proposed MHTC algorithm was applied (offline) to extracellular recordings obtained from macaque parietal cortex in acute recording sessions, as in Chapter 3. Below, a detailed view of a short recording session is provided first, examining the capabilities and behavior of the MHTC algorithm. Then, results from several more recording sessions are provided to demonstrate the efficacy of MHTC in various spike sorting scenarios. Although the ground truths for these data sets are unknown, several details are called out that imply the results' veracity (and in some cases show failure modes).

Session I represents a relatively simple, short recording (29 sampling intervals over about 17 minutes) during which at first one and then two neuronal signal sources are discernible. Figure 4.4 displays the "tracks" of the best global hypothesis estimated by MHTC. Under this hypothesis, two neurons, labeled A and B, are tracked over nearly the entire session. In the first five time steps of the session, the electrode advances about 30 microns, during which time the signals of neuron B become distinguishable. Over the subsequent retraction of the electrode, the signal quality (SNR) of B increases until it eventually becomes the dominant neuron[9]. Figure 4.5 shows a detailed view of these tracks for intervals 1–7 and 22–28. From $k = 22$ to $k = 28$, the electrode is in motion and a significant change occurs in the waveforms, especially of neuron B. Figure 4.5B illustrates the MHTC algorithm's ability to track neurons over such changes and shows that, even though the neuron tracks cross in the second principal component, *PC 2* (Figure 4.4), the clusters are reasonably well separated and the MHTC result appears to be correct.

In Figure 4.5A, the initiation of neuron B's track over the first seven steps is shown, as well as the tracking of neuron A's signals. Note that during the missed detections at $k = 3$ and $k = 4$, the covariance of the predicted measurement grows (and, not shown, the probability of detection decreases); when a cluster appears at $k = 6$ near the predicted location, it is matched to neuron B. Also notice the MHTC algorithm's ability to detect a neuron with very few spikes, in $k = 6$.

As illustrated by the hypothesis tree in Figure 4.6 and the hypothesis ranks in Figure 4.13, the selection of model hypotheses and data associations for intervals 1–7 benefited from the delayed decision making inherent to the MHTC framework, as the best global hypothesis was not always the leading global hypothesis[10]. Figure 4.6 shows the history of all global hypotheses that survived at $k = 7$, $\{H_l^{1:7}\}_l \in \Omega^7$ (but not all hypotheses for the preceding intervals). In each model hypothesis, the log-evidence $\log \mathcal{P}_4 \equiv \log p\big(Y^k \mid \mathcal{M}_{m(l)}, H_{\rho(l)}^{1:k-1}, Y^{1:k-1}\big)$ is given. For each data association

[9] Note the hysteretic nature of the extracellular signal versus electrode position is evident, as the electrode position at $k = 0$ and at $k = 25$ are nearly the same but the signals at these times are much different. These effects, as well as those of non-stationary signals without electrode movement and probable tissue decompression effects, are noticeable throughout the results in this section.

[10] Assume that global hypotheses are ordered according to their probability; i.e., $H_l^{1:k}$ refers to the lth best global hypothesis at interval T_k. The term *leading* global hypothesis is used to indicate the hypothesis with the highest rank at a particular time interval—$H_1^{1:k}$ is the leading global hypothesis at time k. The *best* global hypothesis is the one that has the highest final probability (i.e., the leading global hypothesis at the end of the recording session).

hypothesis, the following is displayed:

- the hypothesis' cluster–neuron assignments of h_l in the format $[a_1, a_2, \ldots]$, where a_g is the neuron ID to which the gth cluster is assigned ($g = 1$ for the blue cluster and $g = 2$ for the green cluster), $a_g = N$ indicates assignment as a new neuron, and $a_g = F$ designates a false cluster;

- the log-probability (roughly) of these assignments—actually $\frac{\mathcal{P}_{1,l}\,\mathcal{P}_{2,l}}{\sum_{n\in\Gamma}\mathcal{P}_{1,n}\,\mathcal{P}_{2,n}}\,\mathcal{P}_3$ from (4.18);

- the log-probability of the resulting global hypothesis $H_l^{1:k}$, stated as the difference from the log-probability of the leading hypothesis; and the rank of this global hypothesis compared to the others at that interval.

The history of the best global hypothesis (i.e., the one shown in Figures 4.4 and 4.5) is indicated by the thicker black lines and borders, along the right side. Note this final solution is the leading hypothesis (rank 1) at $k = 7$ but is ranked third from time $k = 2$ to $k = 4$; such behavior is not possible in simpler algorithms, such as nearest neighbor approaches, and is a key benefit of MHT. Several other interesting details about the operation of MHTC may be extracted from examination of the hypothesis tree:

- The best model (the one with greatest model evidence, $\mathcal{P}_4$) at $k = 2$ has two clusters. However, the single cluster model is chosen in the leading global hypothesis $H_1^{1:2}$ since the parent hypothesis has only one track.

- Similarly, at $k = 6$ the model evidence supports a single cluster, yet the (eventual) best global hypothesis $H_2^{1:6}$ supports the two cluster model because it includes a detection of neuron A, whose continued existence is supported in T_7. Thus, although neuron A is "nearly silent" during T_6, it is still detected by the best hypothesis.

- The influence of the parameter prior in the model evidence is noticeable in the models at $k = 5$, with the higher evidence corresponding to the model with clusters in both predicted locations.

- At $k = 5$, the hypotheses $H_2^{1:5}$ and $H_3^{1:5}$ have significantly greater probability than $H_6^{1:5}$. However, these former two hypotheses are eliminated at $k = 7$ because they are duplicates of the best global hypothesis, in the sense discussed in Section 4.4.

Session II, for which the best hypothesis is shown in Figure 4.7 with further details in Figure 4.8, is a significantly longer, more challenging data set, containing 198 sampling intervals (almost 2 hours). During most of Session II's sampling intervals, the spikes of (probably) two units are detected but are difficult to discriminate from each other. As apparent in Figure 4.7, although the signals from neurons A and B and neurons B and C are poorly separated, the MHTC algorithm steadily tracks them for most of the session. In this case, the second principal component (PC 2) provides little useful information, as a large overlap exists between signals in this coordinate. The following details are notable:

- After the electrode movement at $k = 100$, the signals from neurons A and B essentially merge in feature space. The MHTC algorithm is sometimes limited by the upstream processes and the indistinguishability of low SNR neuronal signals. At the bottom of Figure 4.7, the plots of the spikes in basis $\mathcal{B}^k$ show the data as "seen" by the clustering algorithm. Although the MHTC algorithm identifies two reasonable clusters in $k = 101$ and $k = 113$, these plots show the difficulty of identifying two separate groups from spikes in feature space over these intervals. Thus, although tracks A and C likely belong to the same neuron, this set of intervals "confuses" the tracking algorithm. Still, tracking each of these neurons for the duration shown would be considered very successful in most electrophysiology experiments.

- The sequence from $k = 117$ to $k = 129$, detailed in Figure 4.8, again demonstrates the MHTC algorithm's tracking signals over rather large feature space displacements correlated with electrode motion. Note that the SNR of the neurons increases dramatically as the electrode advances, implying that the electrode is approaching the neurons and may be in danger of damaging them. When the electrode retracts, the signal changes substantially again and returns to a similar state as during $k < 100$.

Results from four more sessions are also provided, further characterizing the MHTC algorithm's abilities as summarized below.

- Figure 4.9 shows, for reference, a session (Session III) in which only a single target is identified. One steady neuron is isolated over the entire session, which lasts just over one hour.

- Session IV, shown in Figure 4.10, represents a very successful session with two neurons tracked, even through frequent changes of electrode position. (Track C, with few actual detections, may result from spurious groupings of outliers, or a temporarily detectable low-SNR neuron. Because it never threatens to become the dominant neuron, it can be ignored.) The detail panels for $k = 131 - 132$ and $k = 170 - 171$ are presented to demonstrate MHTC's recovery from clustering errors.

- In Session V, Figure 4.11, neuron A is tracked for the entire session (over two hours), with a large total electrode displacement (it is very likely that tissue decompression kept the neuron near the electrode tip as the electrode retracted). Somewhat remarkably, a second neuron (B) is also tracked for most of the session, even though a large number of missed detections (due to the intermittent firing of the neuron) makes this challenging.

- Finally, Figure 4.12 provides an example (Session VI) in which three neurons contribute to the recorded signals, tracked by the MHTC algorithm with moderate success. After the electrode (and signals) "settle down" after about $k = 80$, the three distinct sets of spikes are distinguishable (see $k = 90$ panel). The MHTC algorithm is able to maintain the identity of these neurons for the remainder of the session, even as the electrode's retraction from the cells causes them to be increasingly difficult to discern. Several intervals with probable errors, especially when the spikes of different neurons are clustered together, are visible, but the algorithm recovers.

Figure 4.13 shows the rank histories of the best global hypotheses for the above sessions. Note that some, especially the "easy" sessions with steady tracks, typically use the best hypothesis at each interval, whereas the more complex intervals, with overlapping clusters, track initiations, and the like, more fully utilize the delayed decision making capability of the MHTC framework.

Finally, the tracks computed according to two other clustering/tracking methods are provided in Figures 4.14 and 4.15. The methods used to generate these results are the same as used for the cluster sequence in Figures 3.3 and 3.4. Figure 4.14 uses the MAP clustering/tracking procedure that is the subject of Chapter 3, where tracking is made by taking the maximum $\hat{\zeta}_{gj}$ values in each interval (a nearest neighbor approach) rather than propagating multiple hypotheses. Figure 4.15 employs the ML clustering approach used for comparison in Chapter 3; at each time interval, the ML optimization of the Gaussian mixture is computed, then clusters are matched across intervals by a standard nearest neighbor approach. Overall, these methods fail to track neurons over intervals of missed detections and significant non-stationarity and generate more spurious tracks than MHTC.

4.6 Discussion

The results above demonstrate the usefulness of the MHTC algorithm in spike sorting applications. MHTC includes the MAP clustering technique presented in Chapter 3, but the use of the MHT framework enables a more robust tracking solution than the simple nearest neighbor approach obtained by simply taking the maximum $\hat{\zeta}_{gj}$ value. Many of the recording sessions presented above (for example, any that contain a missed detection, indicated in the figures by a small circle) demonstrate the MHTC algorithm's ability to maintain tracks despite signal variability, and may be compared to the single hypothesis, nearest neighbor result in Figure 4.14. Session V represents an extreme example of robustness to intermittent neuron firing, for instance, and Session IV showed that the MHTC method can recover after spikes are poorly clustered in a particular interval.

The hypothesis ranks in Figure 4.13 indicate how a hypothesis that may seem less likely in the current interval may be supported by future data and thus eventually prove to be the best hypothesis overall. Thus, any point in these plots with a rank greater than one supports the value of the delayed decision making process enabled by propagating multiple hypotheses. On the other hand, Session III's consistent use of the leading hypothesis at every interval implies that a simpler framework may have sufficed, but this easy clustering and tracking scenario is unusual in neural data.

It is difficult to determine the "false positive" occurrences in neuronal tracks. That is, when the algorithm claims that a track lasting K intervals exists, how can one verify that this track contains the signals of only a single neuron? This predicament stems largely from the fact that neuron tracks generally initiate and terminate with low SNR signals, since the moving electrode first experiences the neuron's signals from a distance, then usually gets closer, and then sometimes drifts away. When drifting away, the electrode may encounter another neuron's signals and mistake them for those of the first neuron, since low SNR signals of two neurons are particularly difficult to discriminate in the 2-dimensional feature space. This issue is mediated by the electrode positioning algorithm's goal of maximizing SNR: Since the high-SNR neurons are the most important to track and the algorithm is continually repositioning the algorithm to maintain high SNR, false positives in the low SNR neuronal signals may be insignificant.

A related tracking issue is balancing the ability to track neurons over reasonable "jumps" in feature space and yet avoid matching a cluster of a different neuron to the track. This behavior is principally governed by the process noise Q^k, which models all non-stationarity of the signal—if Q^k is too small, a neuron's cluster that moves in between intervals due to non-stationarity may be incorrectly marked as a new neuron or false cluster; if Q^k is too large, false positives as described above are more likely to occur. The feature space displacements shown in Figure 4.7 are near the upper limit of what may be tracked using the Q^k of this implementation.

One possible solution to reduce false positives and decrease the sensitivity on the parameter Q^k is to expand the state space x_j^k and/or measure more of these states. A related idea, often encountered in visual tracking, is to take advantage of "features" of the neuron targets (other than of the waveform shape) that may aid in identifying them. Examples include:

- The neuron's firing rate may be used if it is presumed to remain nearly constant between intervals or if its expected changes can be estimated. In the highly structured scenario of a scientific experiment, the neuron's tuning (its firing rate related to spatial preferences) may be estimated and used for tracking, as the subject's experimental activity would be known. Perhaps even in unstructured paradigms, a correlation between firing rate and LFP may exist and be incorporated to help identify the neuron.

- Other waveform features, such as wavelet coefficients, may be useful for capturing more of the waveform shape. These features could be utilized as part of the state vector for tracking, even if they are not incorporated into the clustering feature space.

- In some recording sessions, it is evident that the cluster's feature space location across several intervals follows a steady trend in a particular location (see Figure 4.7), suggesting that the neuron has a feature space "velocity," which could be included in the state vector to better predict the location of a cluster. However, such behavior has been insufficiently studied to know if this inclusion would improve tracking. Counter-example cases have been observed in which a cluster at first has a steady trend (away from the feature space origin, roughly corresponding to increasing SNR) and then the direction reverses. This can occur even with consistent electrode movement as the electrode passes by the neuron.

As suggested by the last example, correlating certain parameters to electrode movement may improve tracking. For example, a significant change in electrode position would likely increase the process noise Q^k and the expected numbers of new neurons λ_ν. Physiologically, the probability of detecting a neuron $\mathsf{P}_{\mathsf{d},j}$ would also likely vary according to electrode movement and SNR as well as firing rate (as implemented). The above changes are easily integrated into the MHTC framework as presented in the previous sections.

Computationally, a few opportunities for increased efficiency exist. Most notably, perhaps not all model hypotheses need to be clustered separately. As evident in Figure 4.6, the spike-cluster assignments in different model hypotheses are often similar (or identical)—the only influence that would make two optimized models (with the same number of clusters G_m) different is the prior on the cluster means, which originates from the tracks in the parent hypothesis. However, if the priors from different parent hypotheses are similar, a single clustering operation for this set may be used to approximate the results from multiple parent hypotheses. As the EM clustering procedure and, especially, the related Hessian calculation dominate the computational expense, this may result in significant saving. Alternatively, the Hessian calculation can be avoided by using a different approximation to model evidence, such as AIC or BIC, as in Chapter 3, but this may degrade the accuracy of the tracking algorithm. With the implementation detailed above, processing of each interval in non-optimized MATLAB code ranged between from about 2–20 seconds, with an average near 10 seconds.

In conclusion, MHTC's Bayesian approach to combined clustering and target tracking maintains multiple hypotheses (both for cluster models and measurement data association) so that future data can help resolve current ambiguities. This method provides a more robust spike clustering and neuron tracking solution, capable of preserving the identities of particular neurons through intermittent firing of neurons, significant changes in mean waveform, and temporary errors in clustering.

4.A Supporting Probability Calculations

4.A.1 Model Class Prior Probability

This section provides the expression for the prior probability of a particular model hypothesis $\mathcal{P}_3 \equiv P\big(\mathcal{M}_m | H^{1:k-1}_{\rho(m)}, Y^{1:k-1}\big)$. The model hypothesis prior is used in Step 2 of the MHTC algorithm to determine which model classes should be clustered via EM and also in Equation 4.18 to determine the probability of a global hypothesis.

First, note that $\mathcal{M}_m$ includes the number of clusters G_m, so that

$$P\big(\mathcal{M}_m | H^{1:k-1}_{\rho(m)}, Y^{1:k-1}\big) = P\big(\mathcal{M}_m | G_m, H^{1:k-1}_{\rho(m)}, Y^{1:k-1}\big)\, P\big(G_m | H^{1:k-1}_{\rho(m)}, Y^{1:k-1}\big)\,. \tag{4.31}$$

Assume that each model hypothesis generated from the parent $H^{1:k-1}_{\rho(m)}$ has a unique number of clusters G_m; then $P\big(\mathcal{M}_m | G_m, H^{1:k-1}_{\rho(m)}, Y^{1:k-1}\big) = 1$.

Remark 4.4. The procedure below is easily adjusted if this assumption of a unique model class for each G is removed. An appropriate distribution for $P\big(\mathcal{M}_m \,|\, G_m, H^{1:k-1}_{\rho(m)}, Y^{1:k-1}\big)$ would then simply be supplied by the user. A possible choice would be a uniform distribution $\frac{1}{M_G}$, where M_G is the number of model hypotheses that share the same number of clusters G. ◊

Employing the Law of Total Probability gives

$$P\big(G_m | H^{1:k-1}_{\rho(m)}, Y^{1:k-1}\big) = \sum_{N_\tau=0}^{N_t} P\big(G_m | N_\tau, H^{1:k-1}_{\rho(m)}, Y^{1:k-1}\big) P\big(N_\tau | H^{1:k-1}_{\rho(m)}, Y^{1:k-1}\big)\,.$$

The first factor above arises from the combined number of new targets and false clusters. Recalling the definitions and models in Section 4.2.1.3, and that fact that $G_m = N_\tau + N_\nu + N_\phi$,

$$\begin{aligned} P\big(G_m | N_\tau, H^{1:k-1}_{\rho(m)}, Y^{1:k-1}\big) &= P\big(G_m - N_\tau | N_\tau, H^{1:k-1}_{\rho(m)}, Y^{1:k-1}\big) \\ &= P\big(N_\nu + N_\phi\big) \\ &= \frac{(\lambda_0)^{G_m - N_\tau}\, e^{-\lambda_0}}{(G_m - N_\tau)!}\,, \end{aligned}$$

where $\lambda_0 \equiv \lambda_\nu + \lambda_\phi$. (Recall that the sum of N Poisson distributions with rates λ_i follows a Poisson distribution with rate $\sum_{i=1}^{N} \lambda_i$.)

The second factor is the probability of a particular number of tracked targets N_τ, essentially counting the combinations of detected targets that can total to N_τ[11]:

$$P\big(N_\tau | H^{1:k-1}_{\rho(m)}, Y^{1:k-1}\big) = \sum_{\boldsymbol{\delta}\in\Upsilon} \prod_{j=1}^{N_t} (\mathsf{P}_{\mathsf{d},j})^{\delta_j}\ (1-\mathsf{P}_{\mathsf{d},j})^{1-\delta_j}\ ,$$

where $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_{N_t})$ and $\Upsilon = \{\boldsymbol{\delta} : \sum_j \delta_j = N_\tau\}$.

Thus,

$$\mathcal{P}_3 = \sum_{N_\tau=0}^{N_t} \left[\frac{(\lambda_0)^{G_m - N_\tau}\, e^{-\lambda_0}}{(G_m - N_\tau)!} \sum_{\boldsymbol{\delta}\in\Upsilon} \prod_{j=1}^{N_t} (\mathsf{P}_{\mathsf{d},j})^{\delta_j}\ (1-\mathsf{P}_{\mathsf{d},j})^{1-\delta_j} \right]. \tag{4.32}$$

4.A.2 Derivation of Hypothesis Prior

Below is the derivation of $\mathcal{P}_{2,l} \equiv P\big(H^k_l \,|\, H^{1:k-1}_{\rho(l)}, Y^{1:k-1}\big)$ used in the data association plausibility calculation in Section 4.3.2. Begin with the decomposition:

$$\mathcal{P}_{2,l} = P\big(\mathcal{M}_m, h_l | H^{1:k-1}_{\rho(l)}, Y^{1:k-1}\big) = P\big(\mathcal{M}_m | h_l, H^{1:k-1}_{\rho(l)}, Y^{1:k-1}\big)\, P\big(h_l | H^{1:k-1}_{\rho(l)}, Y^{1:k-1}\big)\,. \tag{4.33}$$

Suppose again that each model $\mathcal{M}_m$ encodes a different number of total measurements G_m (see Remark 4.4). Then, since $G_m = N_\tau + N_\nu + N_\phi$, the total number of measurements is encoded in h_l, so

$$P\big(\mathcal{M}_m | h_l, H^{1:k-1}_{\rho(l)}, Y^{1:k-1}\big) = P\big(\mathcal{M}_m | G_m, H^{1:k-1}_{\rho(l)}, Y^{1:k-1}\big) = 1\,.$$

Without knowing the current data Y^k, the probability of h_l depends only on how many of each type of measurement are hypothesized and which of the existing targets from $H^{1:k-1}_{\rho(l)}$ are tracked. Let $\boldsymbol{\delta}_l = (\delta_{1,l}, \ldots, \delta_{N_t,l})$ be the vector of variables indicating whether the existing targets are detected under the lth hypothesis (inferred by comparing τ_l to $H^{1:k-1}_{\rho(l)}$). Since $\boldsymbol{\delta}_l$, N_ν, and N_ϕ are implicitly contained in h_l, Equation (4.33) can now be written

$$\begin{aligned}
\mathcal{P}_{2,l} &= 1 \cdot P\big(h_l, \boldsymbol{\delta}_l, N_\nu, N_\phi | H^{1:k-1}_{\rho(l)}, Y^{1:k-1}\big) \\
&= P\big(h_l | \boldsymbol{\delta}_l, N_\nu, N_\phi, H^{1:k-1}_{\rho(l)}, Y^{1:k-1}\big)\, P\big(\boldsymbol{\delta}_l, N_\nu, N_\phi | H^{1:k-1}_{\rho(l)}, Y^{1:k-1}\big) \\
&= P\big(h_l | \boldsymbol{\delta}_l, N_\nu, N_\phi\big)\, P\big(\boldsymbol{\delta}_l | H^{1:k-1}_{\rho(l)}, Y^{1:k-1}\big)\, P\big(N_\nu\big)\, P\big(N_\phi\big)\,,
\end{aligned} \tag{4.34}$$

where unnecessary conditioning has been removed.

The first factor of (4.34) is based on the number of ways one can assign the specific measurements

[11] For homogeneous detection probabilities, this reduces to a binomial distribution, $P\big(N_\tau \mid H^{1:k-1}_{\rho(m)}, Y^{1:k-1}\big) = \binom{N_t}{N_\tau} (\mathsf{P}_\mathsf{d})^{N_\tau}\ (1-\mathsf{P}_\mathsf{d})^{N_t - N_\tau}$.

for h_l if one knows the number of each type and which existing targets are tracked:

$$\begin{aligned} P\big(h_l|\boldsymbol{\delta}_l, N_\nu, N_\phi\big) &= P\big(\tau_l, \nu_l, \phi_l|\boldsymbol{\delta}_l, N_\nu, N_\phi\big) \\ &= \underbrace{P\big(\phi_l|\tau_l, \nu_l, \boldsymbol{\delta}_l, N_\nu, N_\phi\big)}_{=1\,\text{(known)}}\ \underbrace{P\big(\tau_l|\nu_l, \boldsymbol{\delta}_l, N_\nu, N_\phi\big)}_{\text{1/num. permutations}}\ \underbrace{P\big(\nu_l|\boldsymbol{\delta}_l, N_\nu, N_\phi\big)}_{\text{1/num. combinations}} \\ &= \left(\mathrm{P}^{G_m-N_\nu}_{N_\tau}\right)^{-1} \left(\mathrm{C}^{G_m}_{N_\nu}\right)^{-1} \\ &= \frac{(G_m-N_\nu-N_\tau)!}{(G_m-N_\nu)!}\,\frac{(G_m-N_\nu)!N_\nu!}{G_m!} \\ &= \frac{N_\nu!N_\phi!}{G_m!}\,. \end{aligned} \tag{4.35}$$

The remaining factors in (4.34) rely on the probability models for the numbers of each type of measurement, as defined in Section 4.2.1.3: Since the probability of tracking each target is an independent Bernoulli trial with (inhomogeneous) probability $\mathsf{P}_{\mathsf{d},j}$, then

$$P\big(\boldsymbol{\delta}_l|H^{1:k-1}_{\rho(l)}, Y^{1:k-1}\big) = \prod_{j=1}^{N_t} \left(\mathsf{P}_{\mathsf{d},j}\right)^{\delta_{j,l}}\,\left(1-\mathsf{P}_{\mathsf{d},j}\right)^{1-\delta_{j,l}}\ , \tag{4.36}$$

and $P\big(N_\nu\big)$ and $P\big(N_\phi\big)$ are simply the Poisson distributions (4.4) and (4.5). Combining (4.35), (4.36), (4.4), and (4.5) into (4.34) results in

$$\mathcal{P}_{2,l} = \frac{N_\nu!N_\phi!}{G_m!}\left[\prod_{j=1}^{N_t}\left(\mathsf{P}_{\mathsf{d},j}\right)^{\delta_{j,l}}\,\left(1-\mathsf{P}_{\mathsf{d},j}\right)^{1-\delta_{j,l}}\right]\frac{(\lambda_\nu)^{N_\nu}\,e^{-\lambda_\nu}}{N_\nu!}\,\frac{(\lambda_\phi)^{N_\phi}\,e^{-\lambda_\phi}}{N_\phi!}\ ,$$

or, letting $A_m \equiv \frac{\exp(-\lambda_\nu-\lambda_\phi)}{G_m!}$,

$$\mathcal{P}_{2,l} = A_m\left[\prod_{j=1}^{N_t}\left(\mathsf{P}_{\mathsf{d},j}\right)^{\delta_{j,l}}\,\left(1-\mathsf{P}_{\mathsf{d},j}\right)^{1-\delta_{j,l}}\right](\lambda_\nu)^{N_\nu}\,(\lambda_\phi)^{N_\phi}\,.$$

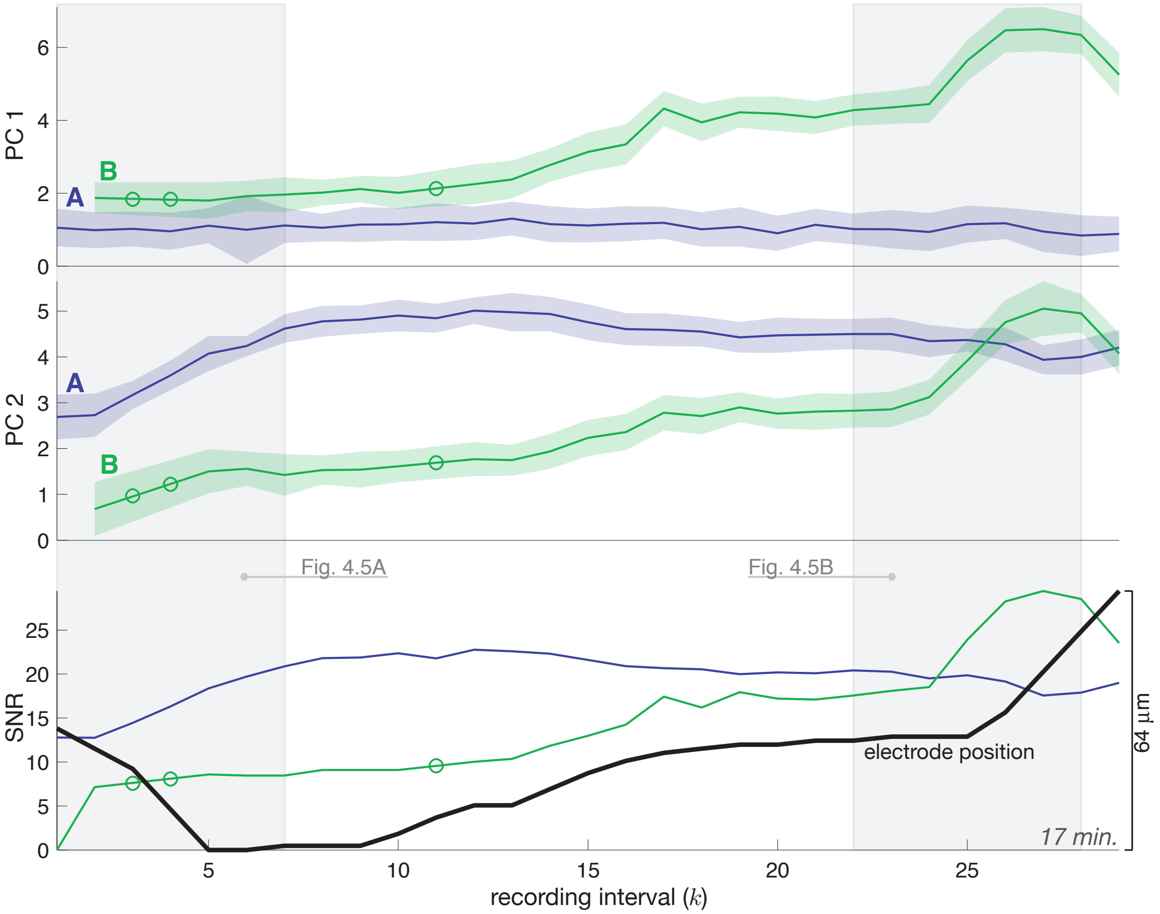

Figure 4.4: MHTC tracks, Session I. The **top two rows** show the tracks in the first and second principal components, respectively. (For representation, these coordinates were obtained by PCA on the set of spike waveforms over the *entire* recording session (basis $\mathcal{B}^{1:k}$), whereas the MHTC algorithm operates in the PCA coordinates derived from the spike waveforms in each interval, $\mathcal{B}^k$.) Each track is represented by its mean waveform (thick line), with a filled area showing two standard deviations of its assigned spikes, and is labeled by an ID (capital letter). Scale is shown on these y-axes for reference but has no clear physical interpretation due to the PCA projection. The **bottom row** plots the SNR of each tracked neuron and the position of the electrode over the recording session (with scale on right side). Circles along the SNR and track lines indicate a missed detection for that target neuron. The shaded intervals are further examined in Figure 4.5

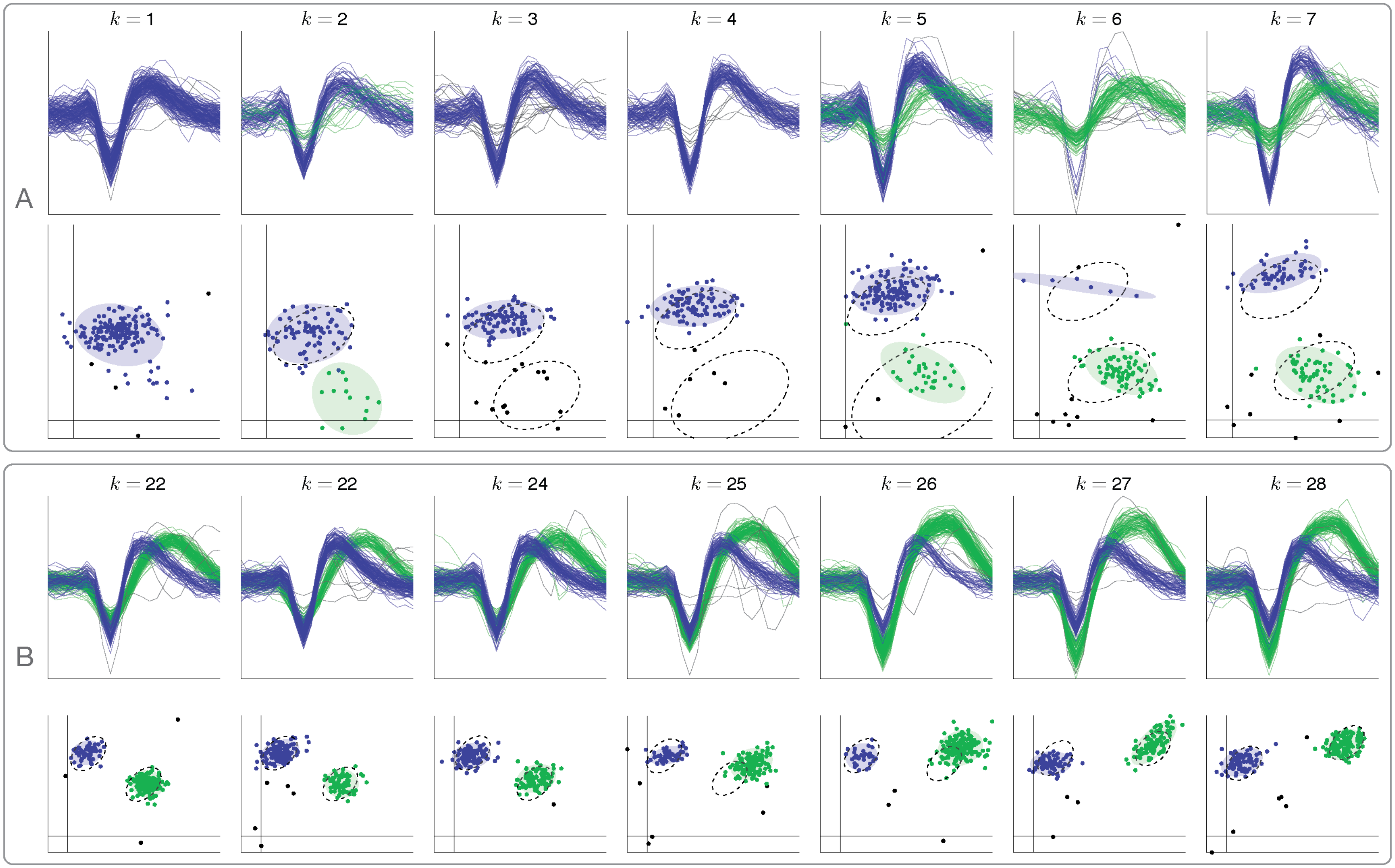

Figure 4.5: Cluster sequence detail, Session I. Subfigures (A) and (B) show the clusters (model hypotheses) from the consecutive intervals highlighted in Figure 4.4, in the same PCA basis. Clusters are indicated with color and filled 2-sigma ellipses; black points indicate classification as “outliers;” black, dashed lines show 2-sigma ellipses of the predicted mean.

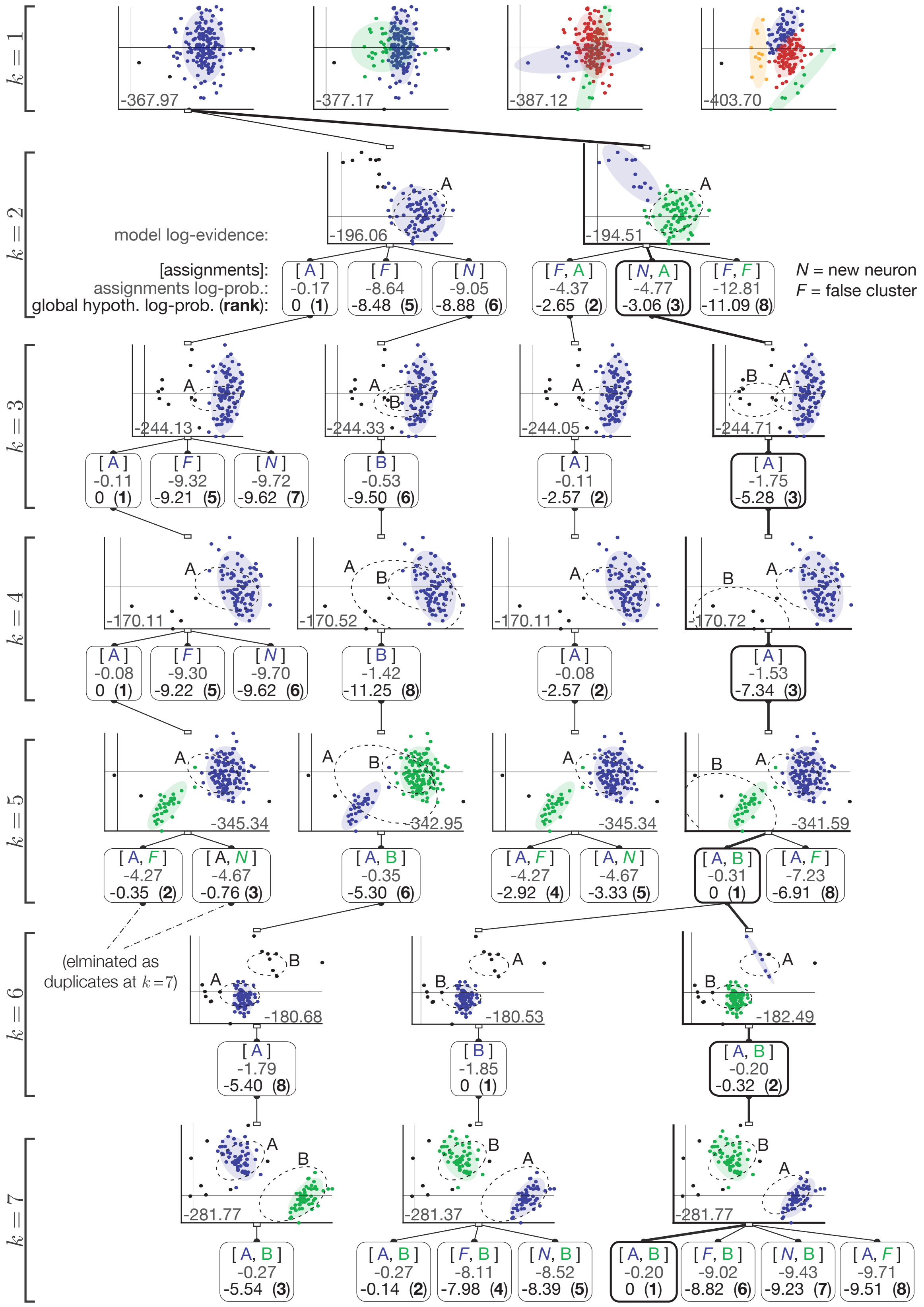

Figure 4.6: Hypothesis tree, Session I. Cluster results are shown in each interval's own PCA basis. Black ellipses show predicted target locations, labeled by target ID. See text for further description.

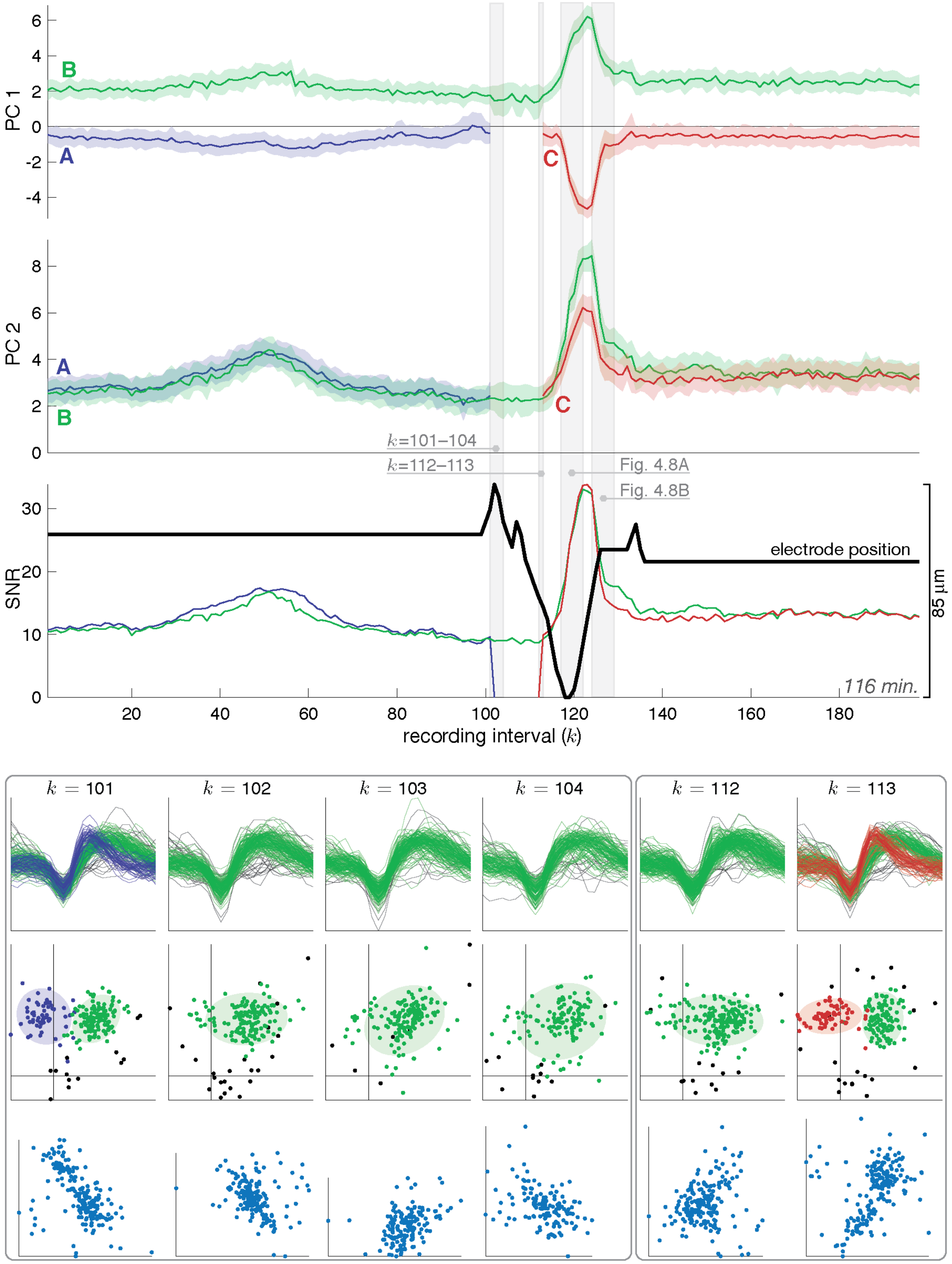

Figure 4.7: MHTC tracks, Session II. See Figure 4.4 for plot conventions The three rows of the detail call outs contain: the spike waveforms; clustered spikes in PCA basis used in the above track plots, $\mathcal{B}^{1:k}$; unclustered spikes in the PCA basis of interval k, $\mathcal{B}^k$.

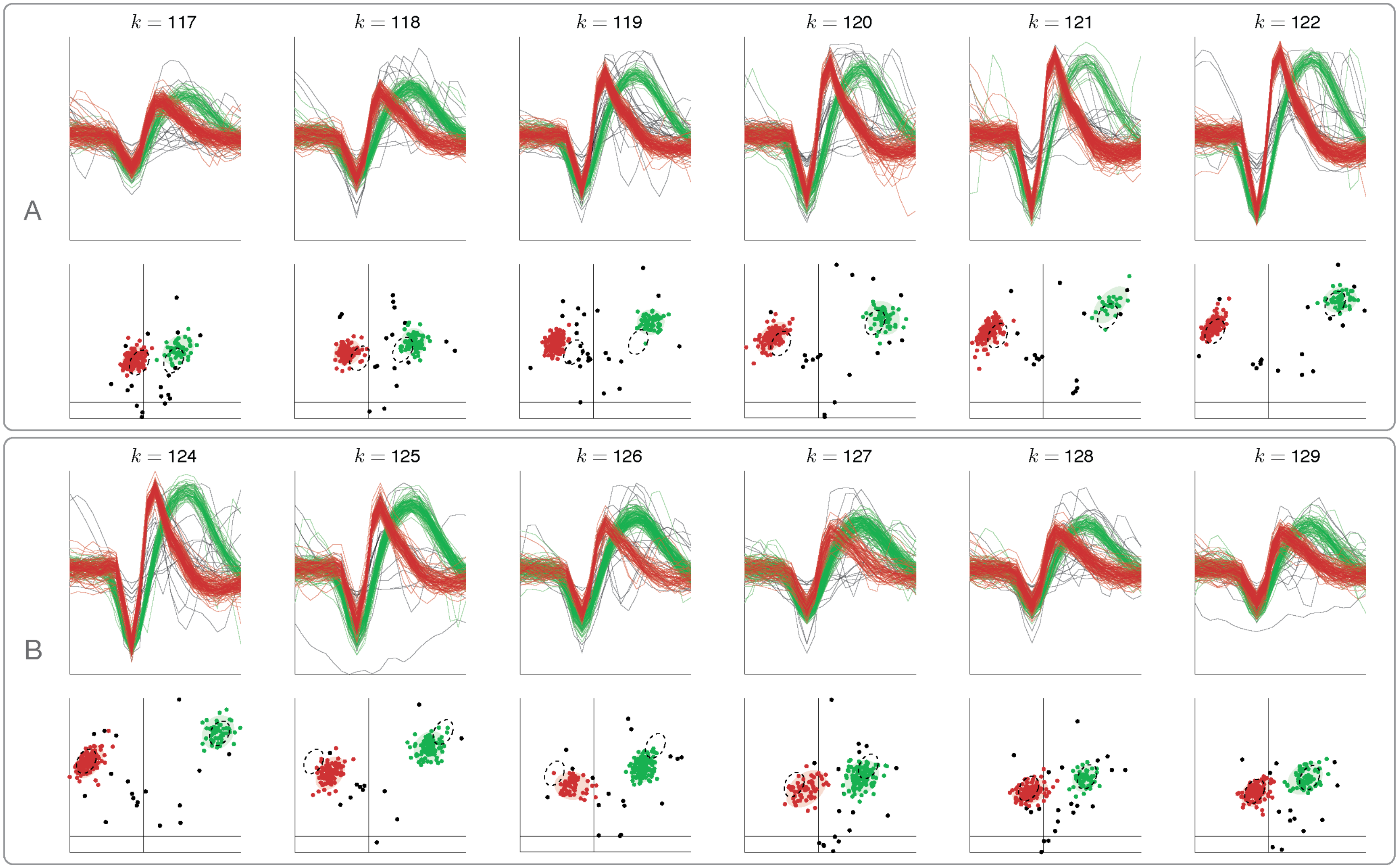

Figure 4.8: Cluster sequence detail, Session II. Subfigures (A) and (B) show the clusters (model hypotheses) from the consecutive intervals highlighted in Figure 4.7, in the same PCA basis. Clusters are indicated with color and filled 2-sigma ellipses; black points indicate classification as “outliers;” black, dashed lines show 2-sigma ellipses of the predicted mean.

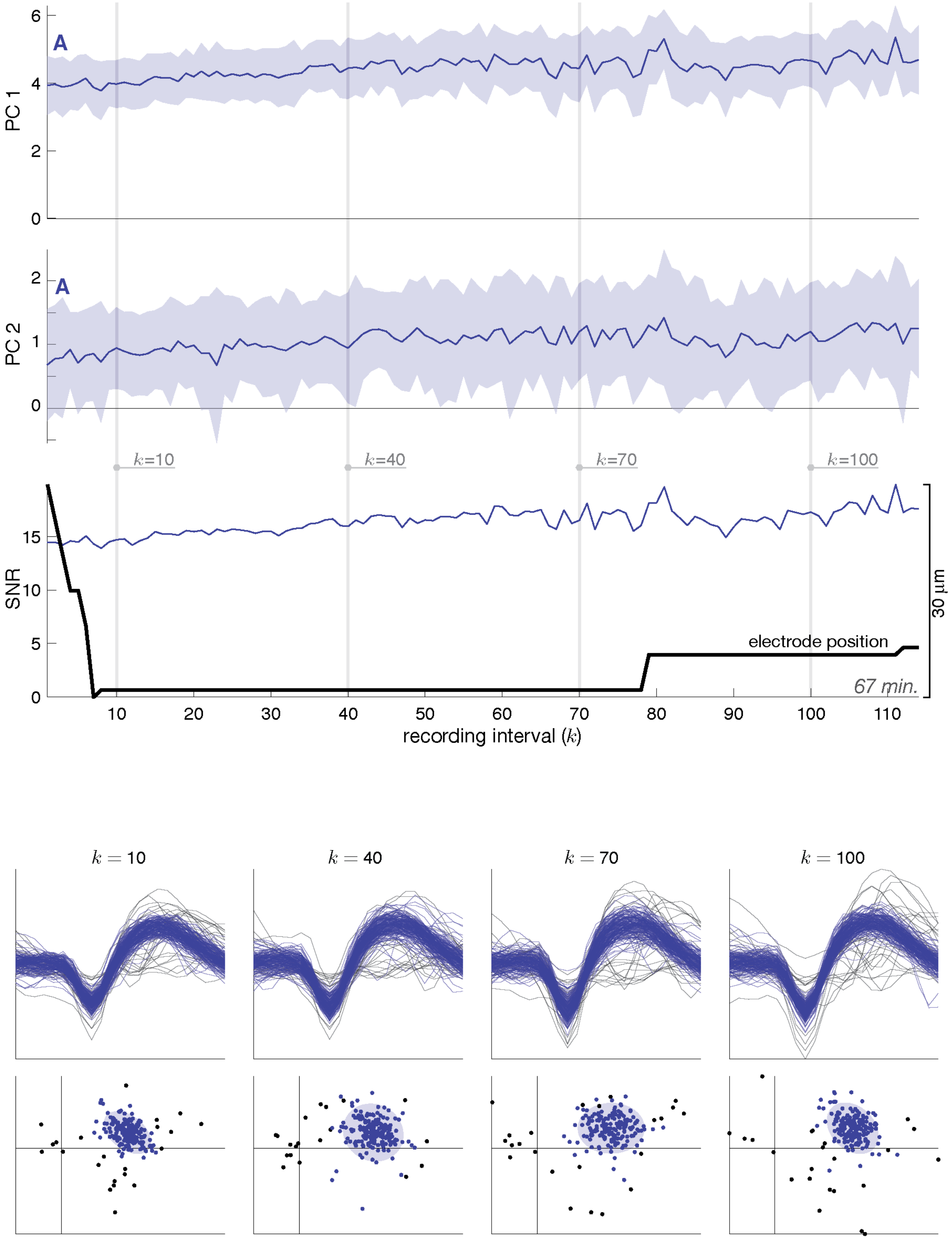

Figure 4.9: MHTC tracks, Session III. See Figure 4.4 for plot conventions. Detail panels show the waveforms and PCA representations of the spikes from labeled intervals.

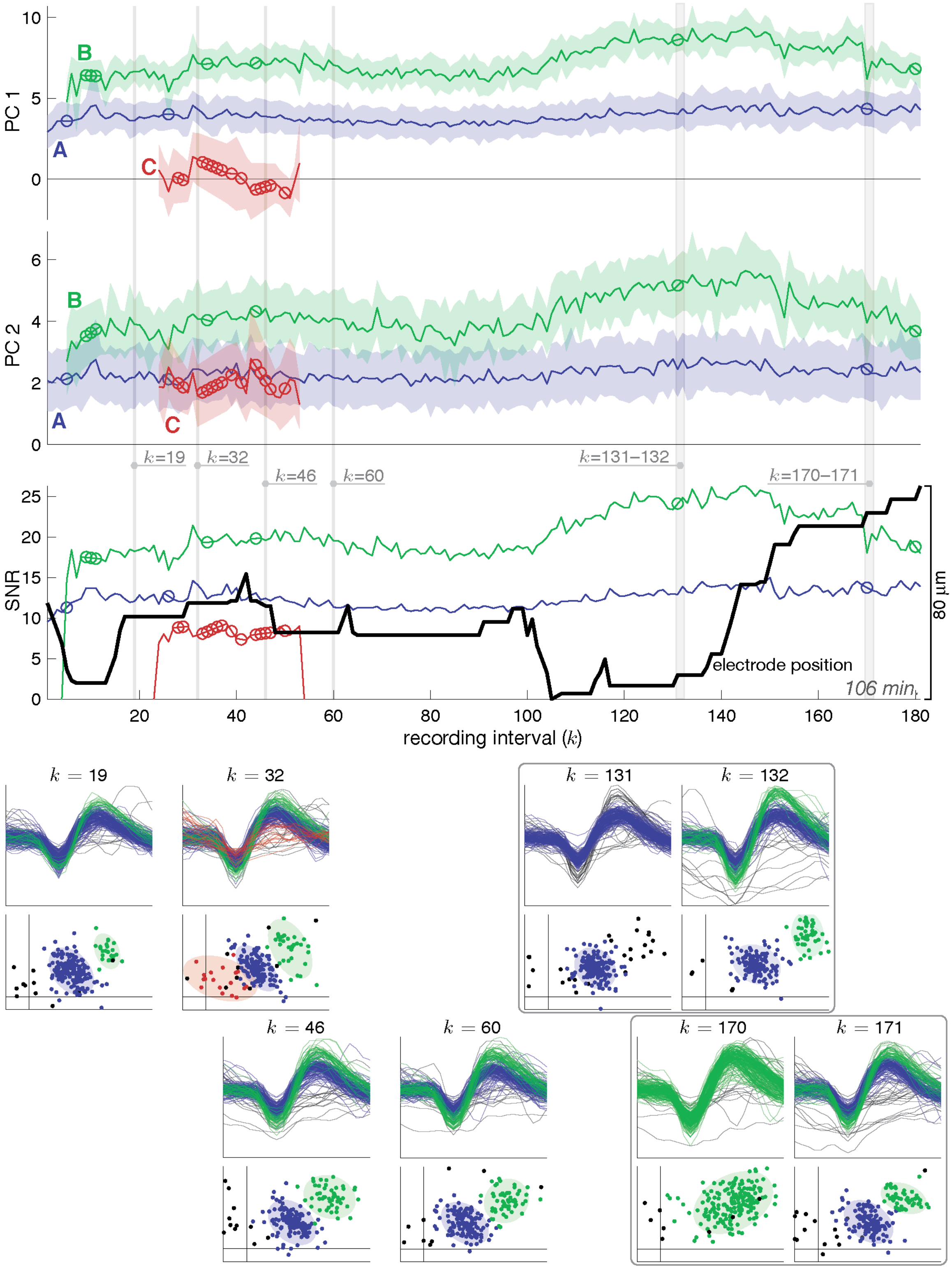

Figure 4.10: MHTC tracks, Session IV. See Figure 4.4 for plot conventions. Detail panels show the waveforms and PCA representations of the spikes from labeled intervals.

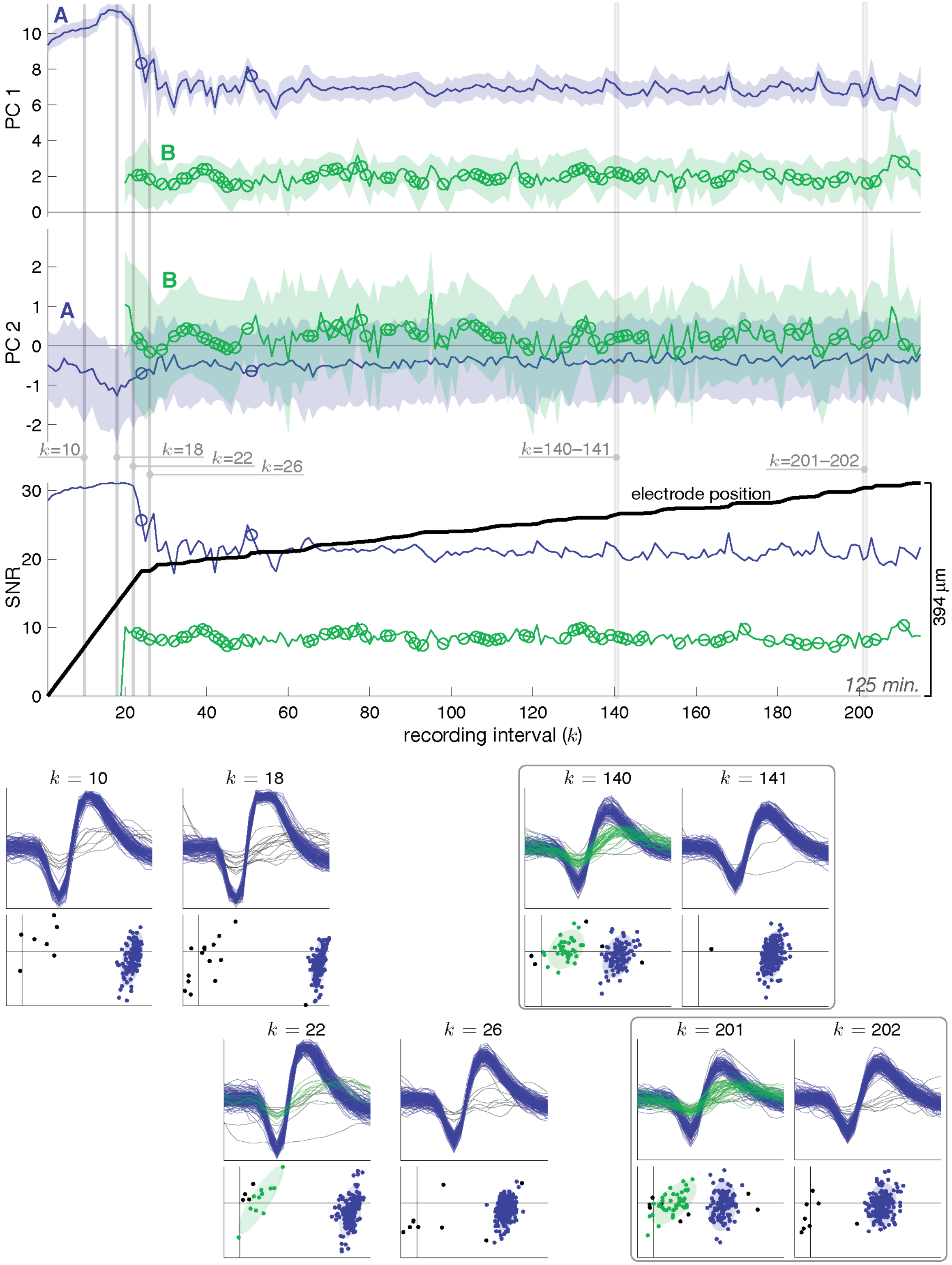

Figure 4.11: MHTC tracks, Session V. See Figure 4.4 for plot conventions. Detail panels show the waveforms and PCA representations of the spikes from labeled intervals.

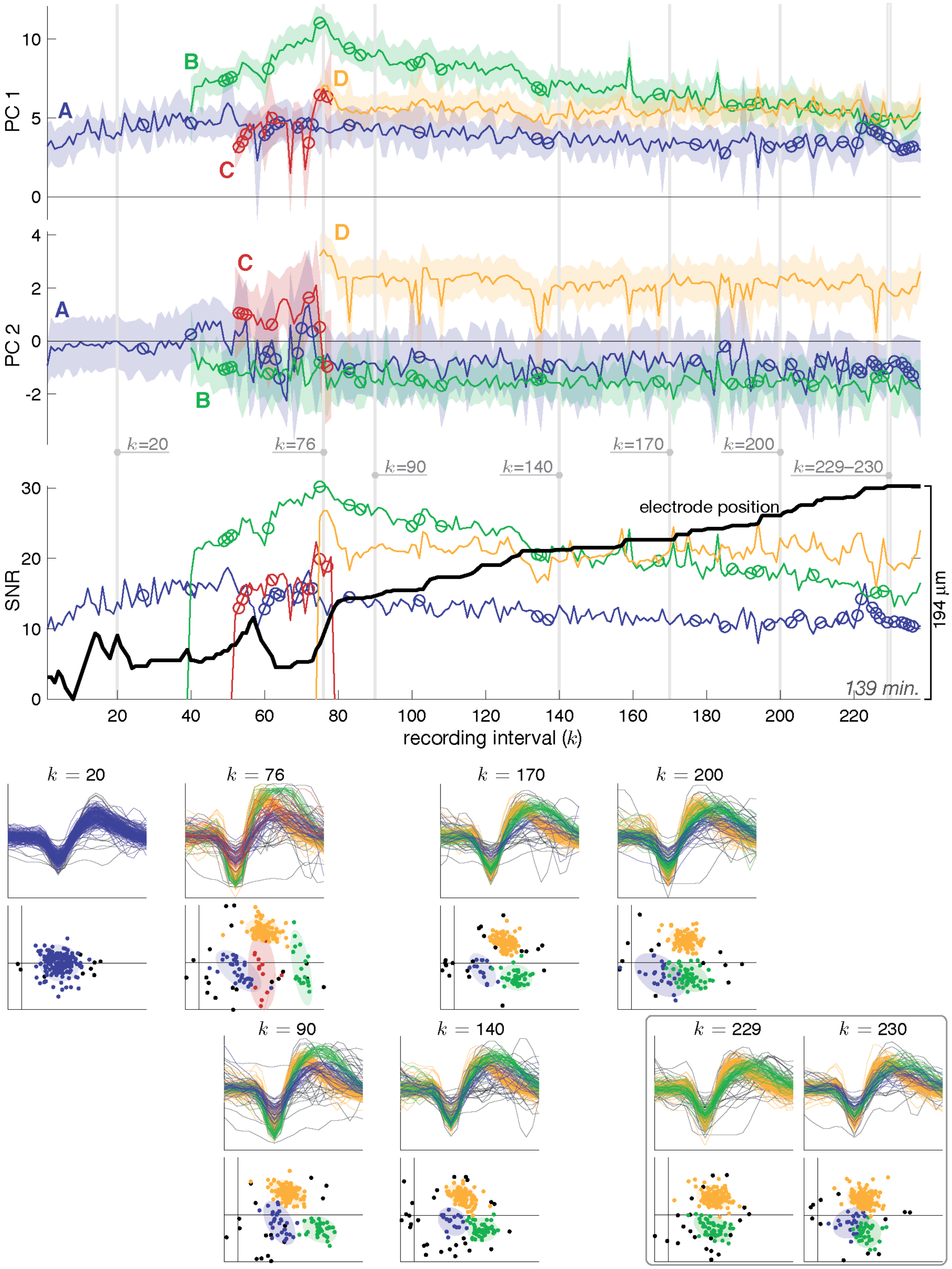

Figure 4.12: MHTC tracks, Session VI. See Figure 4.4 for plot conventions. Detail panels show the waveforms and PCA representations of the spikes from labeled intervals.

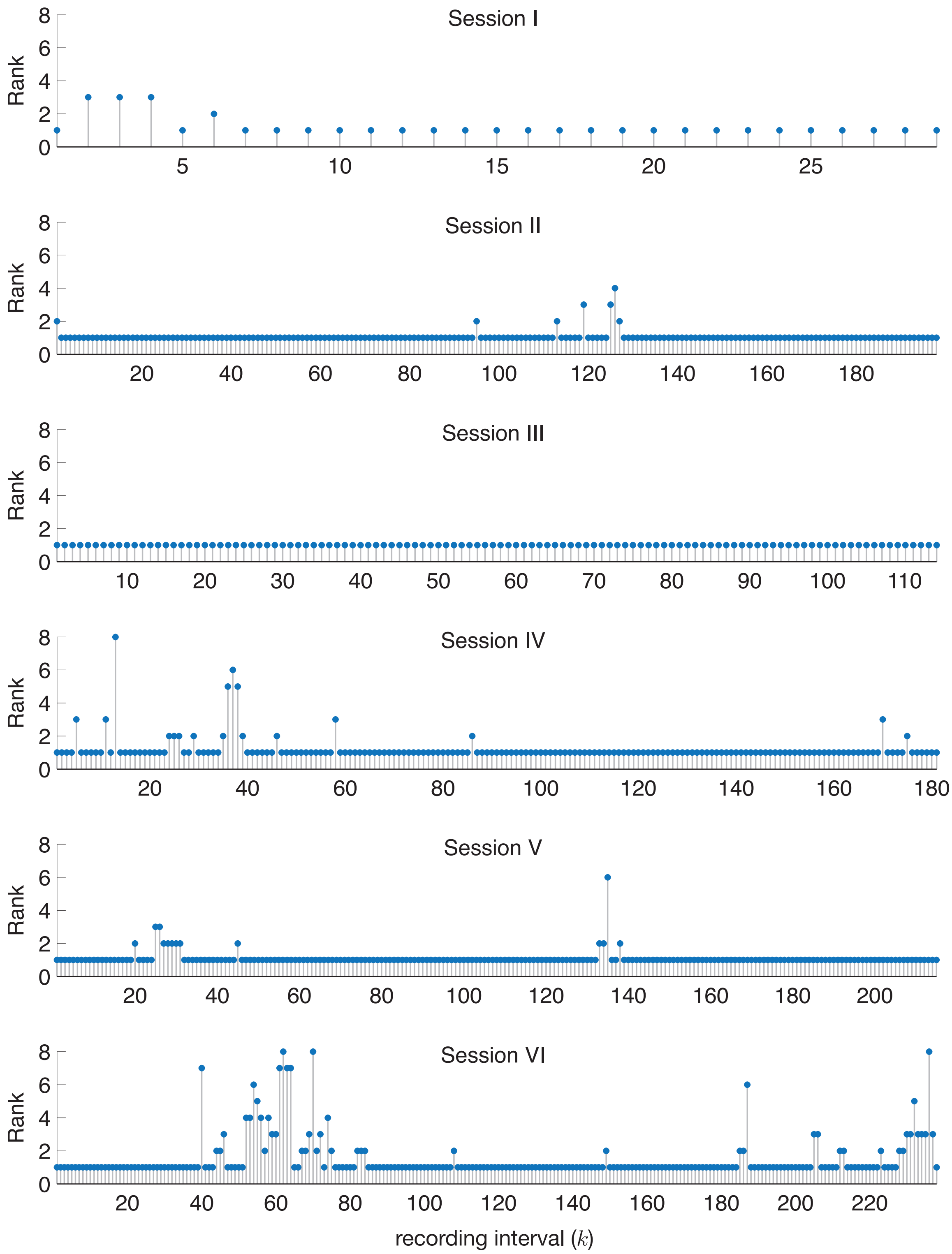

Figure 4.13: Rank of best global hypothesis, Sessions I–VI. The hypotheses whose rank histories are plotted above correspond to the tracks displayed in previous figures for Sessions I–VI.

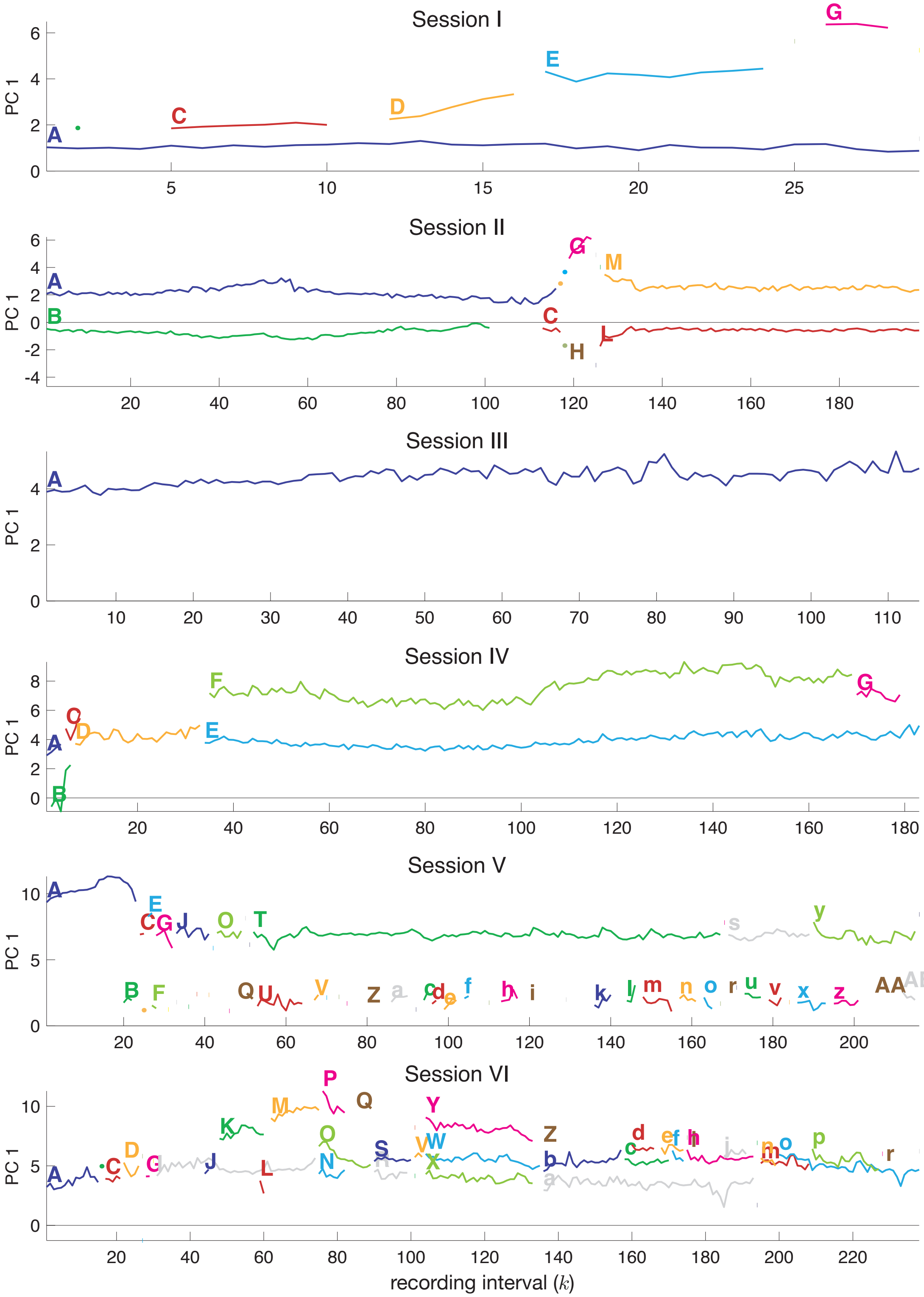

Figure 4.14: Nearest neighbor comparison (MAP clusters), Sessions I–VI. First principal components of tracks in each session are displayed; tracks were determined by the simple nearest neighbor tracking method presented in Chapter 3. Neuron ID labels are suppressed for tracks lasting only one interval.

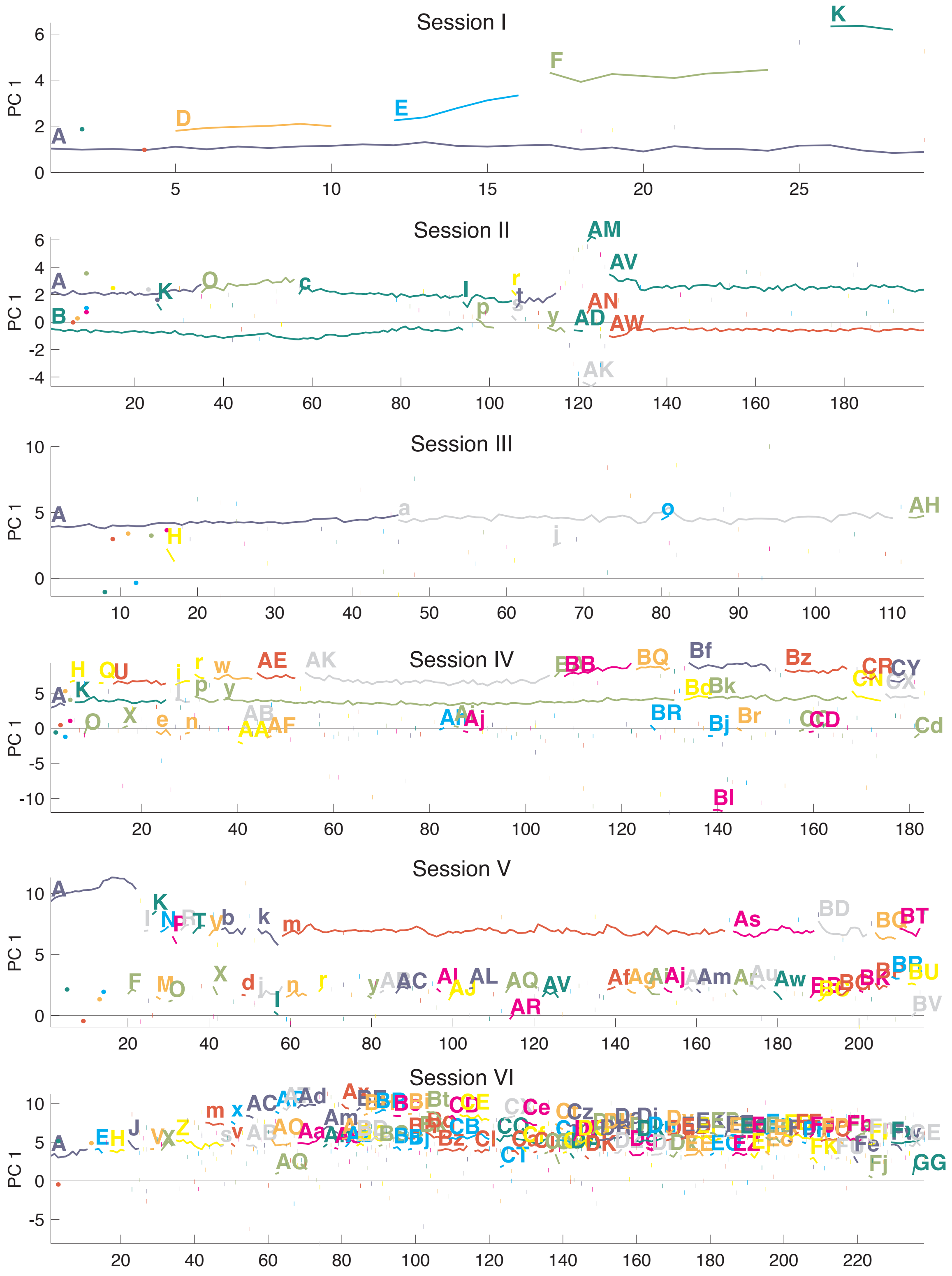

Figure 4.15: Nearest neighbor comparison (ML clusters), Sessions I–VI. First principal components of tracks in each session are displayed; tracks were determined by nearest neighbor association after ML optimization of a Gaussian mixture model. Neuron ID labels are suppressed for tracks lasting only one interval.

Chapter 5

A Semi-Chronic Robotic Multi-Electrode Microdrive

This chapter presents a custom mechanism used to finely position three independent recording electrodes in neural tissue for extracellular recordings. This electrode microdrive is novel primarily in its small size, enabling it to be used "semi-chronically"—that is, for days or weeks at a time—rather than only in daily acute sessions like other motorized microdrives. Additionally, the microdrive is specifically designed for operation with the autonomous electrode positioning algorithm discussed in Section 2.2, though it may also be used under manual control. In this capacity, this mesoscale microdrive serves as a testbed for the positioning algorithm and as a means to develop the specifications for future miniaturized implantable devices (such as in [54]). The current design is also immediately useful to the neuroscience research community for longer-term electrophysiology experiments that cannot be carried out with currently existing microdrives. See Section 2.1 for a review of existing microdrives and extracellular recording practices; the device presented in this chapter represents the next generation of the first motorized miniature microdrive reported in [7].

5.1 Goals and Challenges

The microdrive must be capable of *semi-chronic* experiments, allowing an unrestricted non-human primate to behave normally with free movement and comfort without significant risk of injury while the device is installed, possibly for days or weeks at a time. To more easily integrate into current electrophysiological experiments, the microdrive should affix to a standard cranial recording chamber used in non-human primate research (see Section 2.1.1). These requirements principally imply a much smaller size and mass than available from commercial microdrives, as shown by the comparison in the Figure 5.1

Additionally, the resulting compactness and proximity of all the electrical pathways can increase noise and interference in the recorded signal. Finally, size limitations also restrict the number

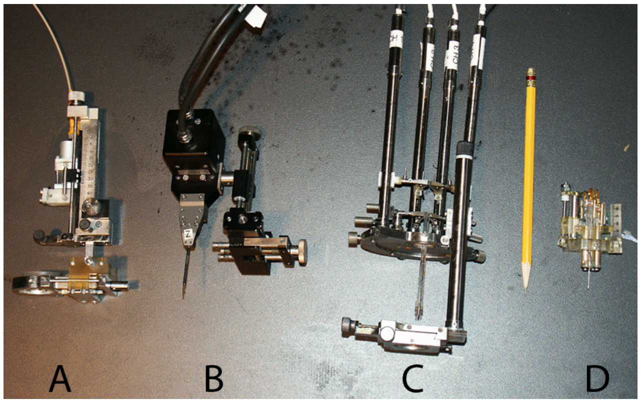

Figure 5.1: Photograph of (D) the miniature robotic microdrive prototype alongside three commercial microdrives: (A) FHC, (B) Thomas Recording, (C) NAN Instruments. A pencil (∼18 cm) is included for scale.

of actuators, and hence recording electrodes, that can be packaged in the device. Physical and biochemical protections must ensure the device and subject will not harm each other. The device must be secure against leaks and impacts, and the interior of the cranial chamber must be sealed from debris (and the fingers of the animal subject). The electrodes must not advance into neural tissue unless commanded to do so. Also, material biocompatibility must be achieved.

The needs of experimenters also stipulate design requirements. For instance, most electrophysiologists desire the flexibility to insert electrodes over a range of locations within the recording chamber, to explore multiple brain areas and perhaps to avoid damaging an area by repeatedly piercing the same tissue. Also, robustness and ease of use considerations often determine the long-term success of the device; in particular, the design should minimize the effort required to prepare the drive for use (especially loading the electrodes) and clean, sterilize, and maintain its parts.

5.2 Design

Figure 5.2 shows a schematic diagram of the microdrive mechanism. The microdrive's central structure is the main body, which encases three piezoelectric linear actuators and furnishes mountings for the electrode guide tube and 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. This is particularly important when the electrode position is to be computer-controlled.

Each linear actuator moves a *carrier*, to which the electrodes are attached both electrically and mechanically. The electrodes consist of platinum-iridium wires coated with glass along their length (except at the recording tip and the back end) for electrical insulation (Alpha Omega Co., USA).

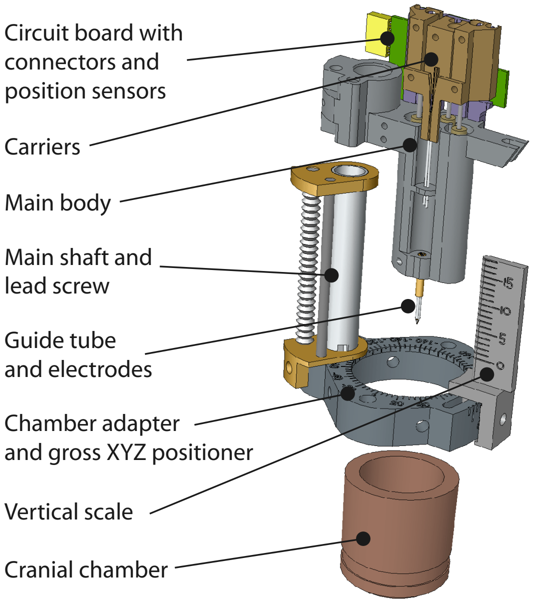
Circuit board with connectors and position sensors
Carriers
Main body
Main shaft and lead screw
Guide tube and electrodes
Chamber adapter and gross XYZ positioner
Vertical scale
Cranial chamber

Figure 5.2: Exploded view of electrode microdrive structure.

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 signals from the electrode are then 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 body assembly is held to a chamber adapter via a main shaft. Rotation of the body assembly around the main shaft axis combined with rotation of the chamber adapter on the chamber rim sets the guide tubes and the electrodes over any location within a 12 mm diameter circular area inside the chamber.

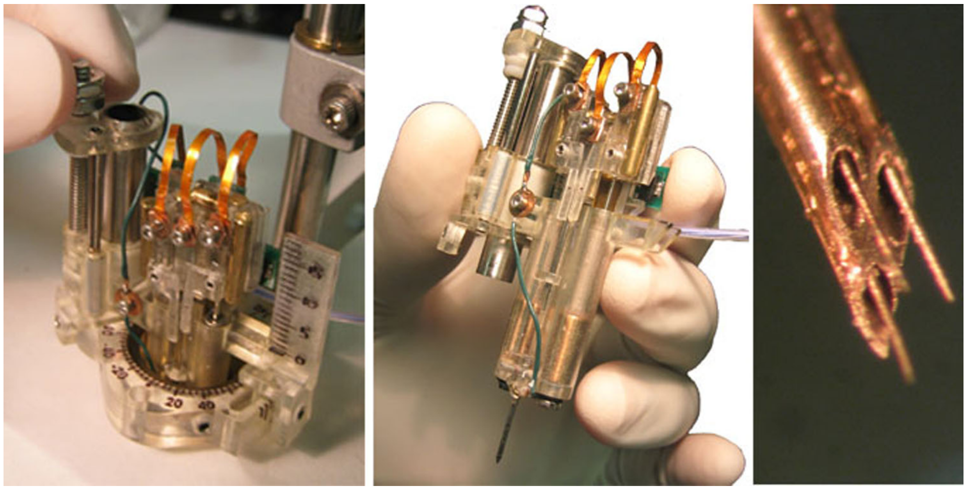

Figure 5.3: Photographs of the electrode microdrive.

After setting the desired planar position within the chamber, the microdrive is lowered by manual turning of the vertical lead screw, until the guide tube just pierces the dura, which is the tough layer of tissue protecting the brain. The microdrive's custom guide tube—consisting of three stainless steel pieces of hypodermic tubing, honed together to a sharp point—protects the fragile electrodes (see Figure 5.3) during this process. This gross vertical lowering of the guide tube is critical and can be challenging, as it is often difficult to tell when the dura has been pierced and lowering the guide tube too much can damage brain tissue. To this end, the microdrive was designed to maximize visual and tactile feedback during this operation. The design allows the experimentalist a rough view of the point of insertion and includes clear vertical markings that show insertion depth. Teflon bearings were used to minimize friction and increase the movement's smoothness.

Once the guide tube is in the correct position above the brain, the electrodes are deployed by activating the piezoelectric actuators. For semi-chronic use, structural elements are locked into place with set screws, and a cover can be placed over the entire assembly for protection against impact and fiddling by the subject.

5.3 Manufacturing

Building electromechanical devices at small scales sometimes defies traditional methods of manufacturing. Relying on machining and assembling individual parts causes loss of space in providing enough material for fasteners, connectors, and sealing, resulting in bulkier devices. This approach also compromises long term reliability, as fasteners can work themselves loose and seals can leak over time. Even if the central components are made using MEMS technology, as future devices may contain, these must be housed and connected through an overall structure that also faces the same difficult conditions.

As an alternative, the microdrive was manufactured using stereolithography (SLA), a layered manufacturing process in which parts are built up in 0.1 mm thick layers by the repeated deposition of material unto a substrate. One advantage of this choice is the ability to create compact designs that are not encumbered by fasteners or connectors or that require additional sealing between assembled components. Structures with complex geometry that would be impossible to make with traditional methods can be made as one solid piece. Other advantages include the relatively short time between design iterations and the ability to custom-modify each device to fit a particular patient or implantation site with relative ease.

One constraint, however, is that the materials used in these processes are often not biocompatible. In the work presented here, an additional step is required to coat the parts made by SLA with Parylene. Parylene is the generic name for a family of thermoplastic polymers that can be deposited using room-temperature low pressure chemical vapor deposition (LPCVD). Parylene is known to be biocompatible (it is a US Pharmacy class-VI implantable material), and is commonly used for coating of biomedical devices such as pacemakers. A thin film of 20 microns of Parylene was coated on the parts using a commercial LPCVD machine (Part number SCS PDS2010E Labcoter).

Certain steps were also taken to account for the limitations in precision of the SLA process. Because of the finite beam size of the computer-controlled laser that solidified the layers of the UV-curable plastic used in the process, actual dimensions varied approximately 0.1 mm from the specified value. These variations, however, as well as the thickness of the Parylene coating, were found to be relatively consistent, and were accounted for in dimensioning the model.

These variations posed a problem for the slider and turn screw mechanism for gross XYZ adjustment of the microdrive. In this case, variations in the slider joint would cause play in the movement of the drive, which would cause damage to the tissue when inserting the guide tube through the dura. To this end, these joints were design with Teflon bearing inserts whose fit could be adjusted with small set-screws to achieve the right amount of joint precision and smoothness needed when advancing the guide tube.

Finally, the geometry of the parts was designed such that areas of high stress were reinforced to account for the flexibility of the SLA plastic (available plastics for SLA have mechanical properties that approximate ABS plastic).

5.4 Improvements from Previous Prototype

The microdrive presented in this chapter offers significant improvements over a previous prototype [7]. First, the new prototype's design reduced the time needed for electrode loading and device cleanup by more than half, owing to the accessibility and lack of assembly required of key components. Second, the design greatly increases ease and reliability of dura penetration, due to the improved visual and tactile feedback. Third, the use of SLA parts increased robustness to breakage and leakage of biological fluids, and made the microdrive components easier to repair, replace, and also revise. Finally, the new design reduced the total weight by nearly half from the previous design, primarily due to the elimination of metal modules and fasteners. Taken together, these solutions to practical challenges have enabled this miniature electrode microdrive to overcome issues of usability and flexibility, which are key barriers to adoption of such technology within the electrophysiology community.

5.5 Experimental Results

This section presents data obtained by the microdrive to verify its effectiveness in extracellular recordings and in particular to demonstrate its use as part of the robotic electrode positioning system. These data sets were recorded in acute experimental sessions in macaque parietal cortex, with the microdrive affixed to a standard cranial chamber and autonomously controlled by the electrode positioning algorithm (see Chapter 2). Signals, recorded by glass-insulated platinum-iridium electrodes of approximately 1.5 MΩ impedance at 1 kHz, were amplified and filtered (Plexon, Inc.) and then interfaced to the controlling computer via an analog-to-digital data acquisition card (National Instruments).

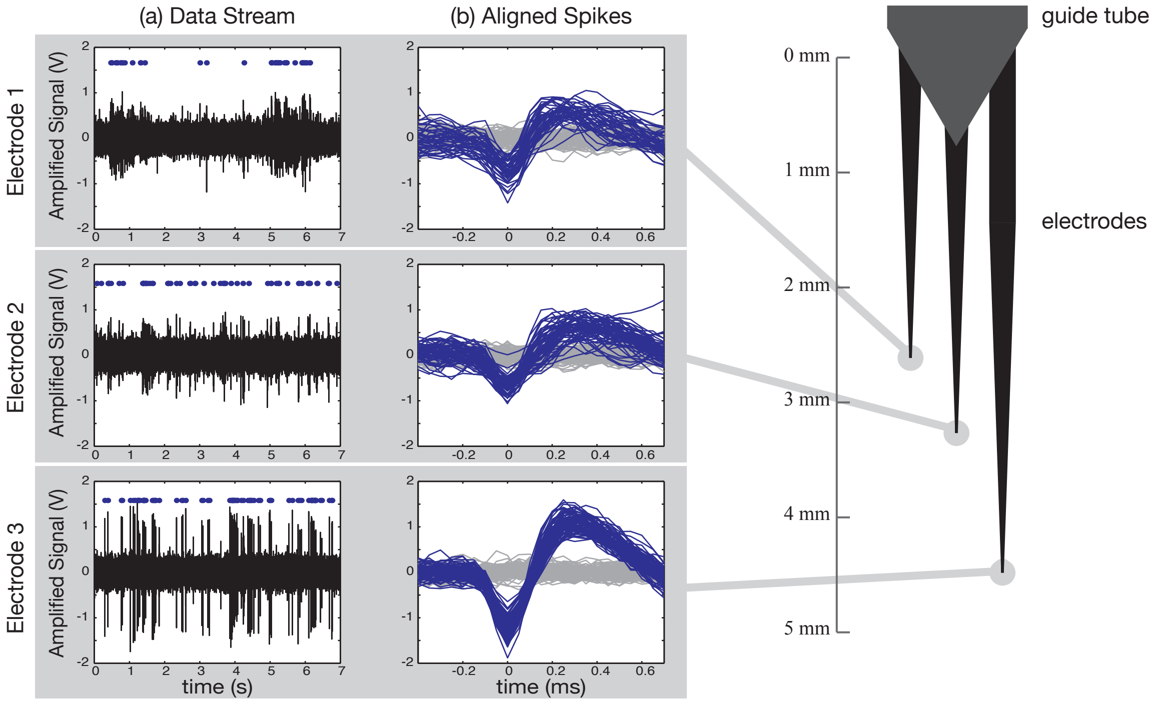

Figure 5.4: Simultaneous recordings from the microdrive's three electrodes.

Figure 5.4 shows neural data recorded simultaneously by the microdrives three electrodes. Graphs in column (a) plot several seconds of the filtered data stream over time, sampled at 20 kHz, with blue dots above the voltage trace at times when spikes were detected by the method in [55]. Column (b) shows close-up views of these detected spike waveforms, aligned by their minimum with noise samples shown in gray underlying the spike waveforms. In this recording, electrode 3 has isolated a strong neuronal signal (large SNR), as qualitatively evident in comparing the waveforms to the noise sample. The diagram on the right side of the figure indicates the relative depths of the three electrodes at the time of the recordings.

Figure 5.5 and Figure 5.6 show two examples of the robotic electrode microdrive autonomously achieving neuron isolations—that is, finding the electrode position that optimizes the neuronal signal via the algorithm discussed in Chapter 2, for which this microdrive was explicitly designed. These figures include information indicating the operation of autonomous electrode positioning algorithm over twelve consecutive recording intervals, with the recorded spike waveforms and their

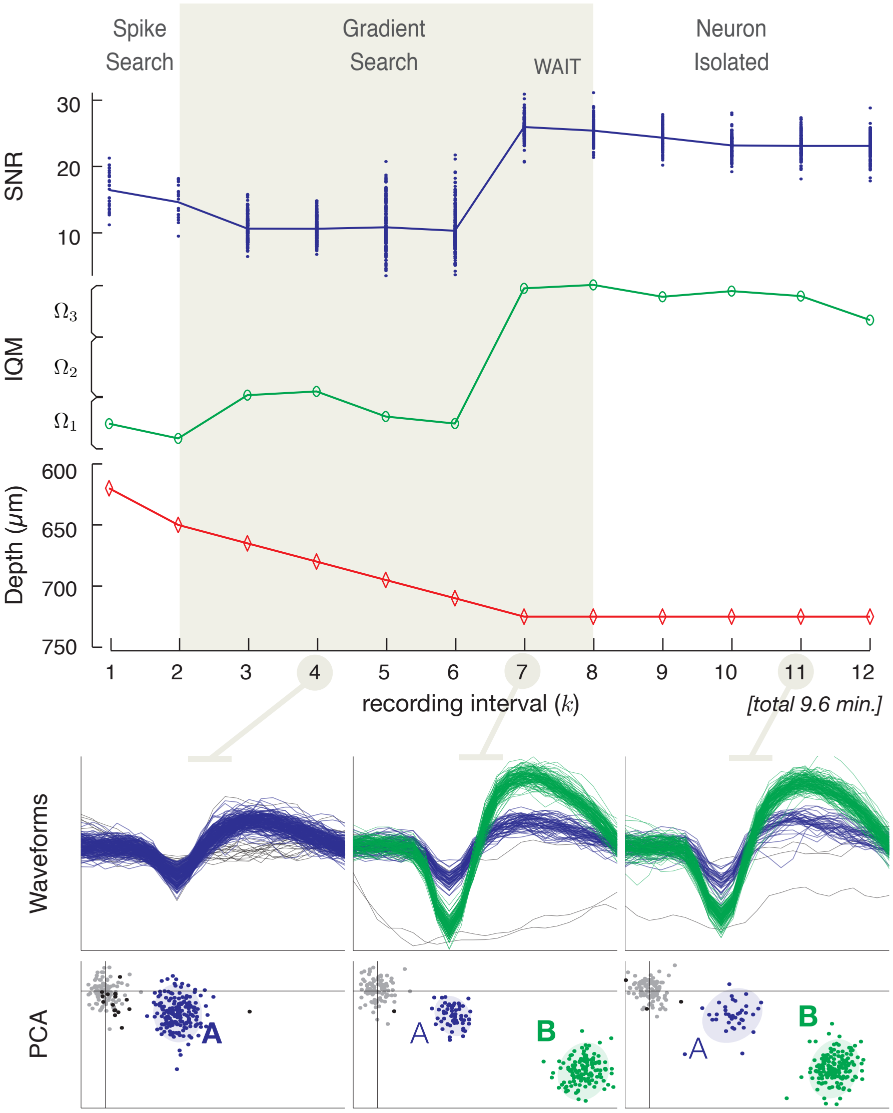

Figure 5.5: Example neuron isolation by robotic microdrive (Isolation by IQM).

2-dimensional PCA projections shown for particular recording intervals, which again demonstrate the strong signals obtained via the microdrive.

For the reader interested in the operation of the electrode positioning algorithm over these intervals, the top two graphs plot the SNR and isolation quality metric (IQM; see Table 2.1) of the dominant neuron, and the third graph indicates the electrode position at each interval. The state of the supervisory finite state machine (SFSM) is shown above. The detail plots show sample clustering/tracking results. The cluster marked by the bold letter is the one associated to the dominant neuron, for which the SNR and IQM is shown above. In Figure 5.5, a transition to the *Neuron Isolated* state of the SFSM is invoked because the electrode suddenly picked up the signals of a strong new neuron at $k = 7$; this neuron's IQM is very high, indicating that further electrode

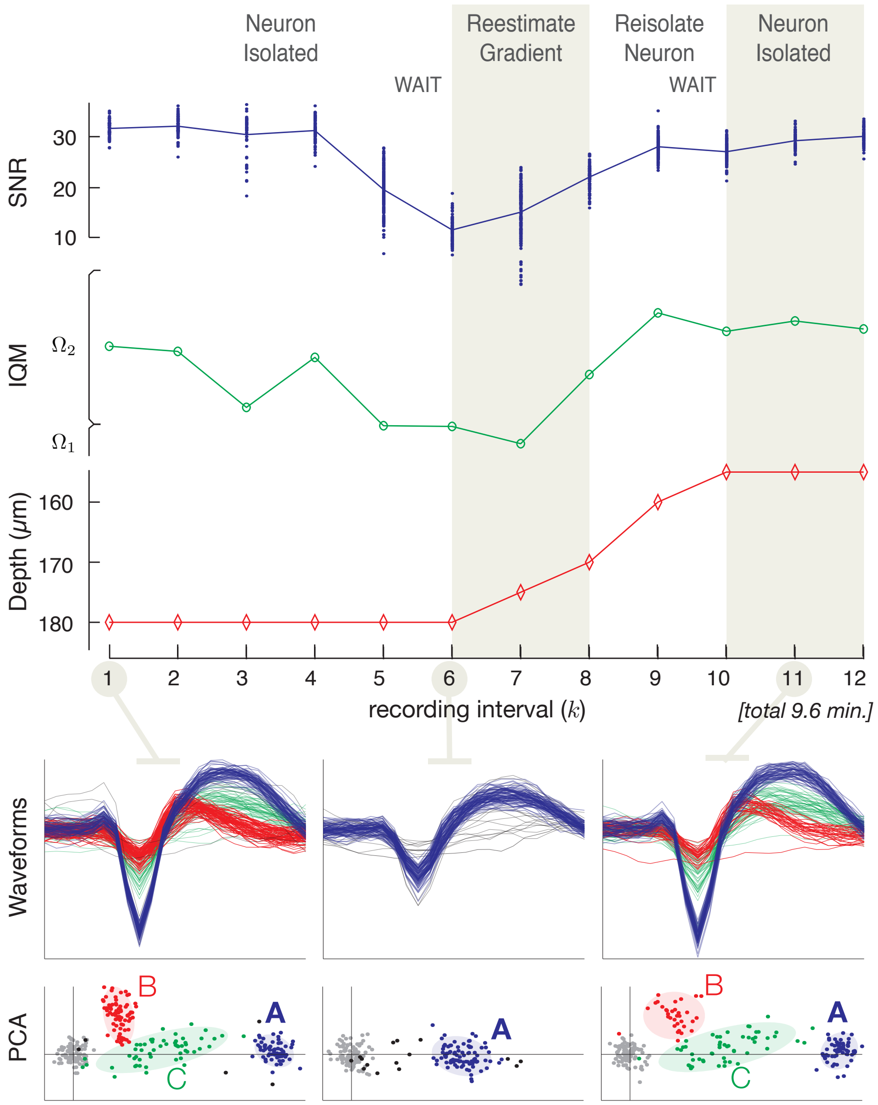

Figure 5.6: Example neuron isolation by robotic microdrive (Re-isolation).

advancement may damage it. Figure 5.6 shows a “re-isolation” path: the SNR of the dominant neuron drastically decreases over the first six intervals (even though the electrode is stationary), so the control algorithm retracts the electrode (tissue decompression being the likely culprit) until the SNR reaches a level equivalent to the original isolation.

Chapter 6

Conclusion

6.1 Summary of Thesis Contributions

The primary contribution of this thesis is the *multiple hypothesis tracking for clusters* (MHTC) algorithm, a multitarget tracking solution for targets observed through *groups* of measurements in each processing interval or scan. The MHTC algorithm may be thought of in two parts: a Bayesian clustering algorithm for associating observations to clusters in each interval (Chapter 3) and a probabilistic supervisory system that maintains several hypotheses of cluster models and cluster–target data associations across intervals (Chapter 4).

The novel clustering method of Chapter 3 provides significantly more consistent results than previously available methods, due to its incorporation of information from the preceding intervals as a prior for the current interval. This maximum *a posteriori* (MAP) method optimizes a Gaussian mixture model via expectation-maximization (EM). In addition to constructing an appropriate prior on cluster locations and adjusting the traditional EM approach to incorporate this term, this thesis presents a new process for generating seed clusters and a suitable model class selection method. As a whole, this technique enables the association of clusters across consecutive time intervals and thus the tracking of neurons whose signals persist over many adjacent recording intervals. The MAP clustering technique often succeeds even in situations having low firing rates (few samples per cluster), low signal-to-noise ratio, poor cluster separability, and non-stationary waveforms.

The MAP clustering method fits naturally into the MHTC framework that manages multiple possibilities for how to cluster the data and how to assign clusters to existing targets, as a new target, or as false clusters, as described in Chapter 4. The MHTC algorithm similarly utilizes a Bayesian probabilistic approach that incorporates information over successive time intervals, explicitly modeling the plausibility that neurons will appear or disappear on a signal and tracking putative neurons over changes in their waveform. Because it incorporates the delayed decision making logic of multiple hypothesis tracking (MHT), the MHTC algorithm uses future data to resolve clustering or data association ambiguities in the current interval. This novel combination of clustering and tracking in a single solution robustly maintains the identities of neurons, meeting the needs of challenging neural recordings.

Another contribution of this thesis is the robotic neural interface hardware presented in Chapter 5. This electrode microdrive, which serves as a testbed for the autonomous electrode positioning algorithm described in Chapter 2, is novel in its small size, aimed at enabling the microdrive to be used in semi-chronic experiments. These investigations, lasting days or weeks at a time, will aid in developing the specifications for future miniaturized implantable devices and for long-term operation of the control algorithm.

6.2 Opportunities for Future Work

While the preceding chapters have presented novel contributions to improve the acquisition and processing of neuronal signals, it is appropriate to suggest the most important next steps that may further advance this work. A goal of the initial MHTC implementation in this thesis was to limit unnecessary complexity and, to some degree, maintain generality for multiple applications, demonstrating usefulness of the algorithm even without carefully chosen representations or parameters. Some natural next steps, therefore, include more thorough consideration of specializing the implementation for spike sorting as discussed below.

Perhaps the most important area to examine is the choice of feature space. While a 2-dimensional PCA basis often provides sufficient information for clustering, other representations may capture features better suited to separating neuronal spike waveforms [21, 30]. (In the author's experience, simply expanding the PCA basis to include the first three or more components can actually produce worse results than using two components, perhaps grouping similarities along "noise directions.") Even if retaining the use of PCA bases, some additional pre-processing steps may make the data easier to sort [34, 60]. From the perspective of tracking, using an invariant feature space would be helpful, reducing the dimensionality of the state space (which is now the full waveform) and adding a more convenient representation for users; however, these considerations are relatively minor, especially as the general formulation in this thesis handles time-varying feature spaces without much additional computational burden.

The neuron characteristics used to represent its state in the dynamical system model, as well as the choice of various parameter values, may also be inspected further. As noted in Section 4.6, incorporating the neuron's firing rate or feature space "velocity" may improve the tracker. Additionally, some parameters, such as the detection probability and process noise covariance, may be time-varying and better modeled as a function of SNR and/or change in electrode position.

Finally, the electrode positioning algorithm (discussed in Chapter 2) may be enhanced by fully incorporating the reliable clustering and tracking information provided by the MHTC algorithm to improve its optimization strategy. Recall that, before the work presented in this thesis, the positioning algorithm declared the neuron with the highest SNR the *dominant neuron* after each time interval and attempted to maximize its *SNR curve*, without maintaining an explicit sense of "neuron identity." If neurons are successfully tracked (i.e., their identities are maintained across recording intervals), the control algorithm can construct and properly maintain multiple SNR curves, one for each neuron, as more physiologically appropriate. The control algorithm can then use this multiplicity of SNR curves in several ways: Most simply, it can "switch" between the SNR curves of different neurons instead of either (a) building up a single SNR curve that is erroneously based on the combined signals of disparate units, or (b) throwing out a previously used SNR curve and starting over when it switches neurons. Armed with multiple and distinct SNR curves, the optimization function of the control algorithm may even be reconsidered. For example, perhaps the electrode can be placed to simultaneously record the distinct signals of *two* different neurons, finding an optimal compromise between signal quality and multiple isolations; such "multi-single-unit" recording would increase the output from every electrode.

6.3 Neural Interfaces and Other Applications

Although this thesis is principally motivated by the development of an autonomous neural interface for neuroprosthetic devices, other disciplines may also benefit from this work. First, the neuroscience research community is constantly seeking to improve methods for single-unit recording and is especially interested in maintaining neuron isolations for long periods of time; thus, a chronic or semi-chronic autonomous electrode positioning system (which utilizes the thesis' signal processing methods and microdrive) may facilitate new paradigms for scientific experiments as well as help enable clinical neuroprostheses. Second, the challenge of spike sorting exists for virtually all extracellular neural recordings, not just those in which electrodes are autonomously controlled; the MHTC algorithm can be employed to automate the tedious and often inaccurate manual processes by which these neural data are typically classified. Using MHTC for spike sorting, in addition to providing efficiency and perhaps accuracy benefits, may also enable experimenters to detect the signals of more than one neuron at a time, as Chapter 4 showed several multi-single-unit results.

Finally, the new framework provided by MHTC may be applicable to fields outside of electrophysiology. Multitarget tracking methods have been increasingly applied in disciplines such as robotics, computer vision, oceanography, and biomedical research [35]. The MHTC algorithm provides a general solution for tracking targets observed through probabilistically distributed groups of measurements over successive intervals or scans, regardless of their source.

In conclusion, the novel neural signal tracking methods and recording hardware contributed by this thesis provide the scientific community with tools to better investigate brain activity at a cellular level and are critical components of an autonomous neural interface. Because future neuroprostheses will likely rely on the calibrated signals of particular neurons, the ultimate goal of this implantable neural interface is to optimally position many independently movable electrodes, each capable of tracking individual neurons indefinitely. Thus, the advances in this thesis hopefully provide steps along the path to a practical neuroprosthetic system.

Appendix A

Laplace's Method

This appendix reviews Laplace's method for asymptotic approximation of integrals and shows how this technique is applied in Chapter 3 and Chapter 4.

A.1 Review of Laplace's Method

This section summarizes Laplace's method and the essential theory that supports the method, closely following [77], except that the multidimensional case is presented, as in [26] and [78]. Consider the integral

$$I(N) \equiv \int_R f(\theta) \exp[-\phi_N(\theta)]\, d\theta \,, \tag{A.1}$$

where $\theta \in \mathbb{R}^\eta$ can take values over the region $R \subset \mathbb{R}^\eta$ and $\phi_N(\theta) \equiv N\phi(\theta)$. Both f and ϕ must be sufficiently smooth for the operations that follow.

Suppose that the unique absolute minimum of ϕ occurs at $\hat{\theta} \in R$ so that $\nabla_\theta \phi_N(\hat{\theta}) = \vec{0}$ and $\mathbf{H}(\hat{\theta}) \equiv \nabla_\theta \nabla_\theta \phi_N(\hat{\theta})$, the Hessian matrix evaluated at $\hat{\theta}$, is positive definite. Suppose also that $\phi_N(\hat{\theta}) \neq 0$. Then *Laplace's formula* states that, for large N, the integral in (A.1) can be approximated as:

$$I(N) \approx f(\hat{\theta}) \exp[-\phi_N(\hat{\theta})]\, (2\pi)^{\eta/2}\, |\mathbf{H}(\hat{\theta})|^{-1/2} \,. \tag{A.2}$$

The intuition behind Laplace's formula relies on the fact that as $N \to \infty$ the integrand in (A.1) becomes increasingly peaked at $\hat{\theta}$ and thus only values within an arbitrarily small neighborhood of this point need to be considered. The last two factors in (A.2) are introduced by a second-order Taylor series expansion of ϕ_N at the minimum $\hat{\theta}$,

$$\phi_N(\theta) \approx \phi_N(\hat{\theta}) - \frac{1}{2}\big(\theta - \hat{\theta}\big)^T \mathbf{H}(\hat{\theta})\big(\theta - \hat{\theta}\big) \,, \tag{A.3}$$

and a subsequent use of the equality

$$\int \exp\left[-\frac{1}{2}(\theta-\hat{\theta})^T\mathbf{H}(\hat{\theta})(\theta-\hat{\theta})\right] = (2\pi)^{d/2}\,|\mathbf{H}(\hat{\theta})|^{-1/2}\,. \tag{A.4}$$

(One may notice that the integrand in (A.4) is, apart from a normalizing constant, equal to a Gaussian density with mean $\hat{\theta}$ and covariance matrix equal to the inverse of $\mathbf{H}(\hat{\theta})$.)

The application of Laplace's formula (A.2) is called *Laplace's method.* For a more rigorous discussion, as well as verification that the approximation (A.2) asymptotically approaches the integral (A.1) as $N \to \infty$, the reader is directed to [77].

A.2 Application for Model Evidence

To find the model evidence $p\big(Y^k|Y^{1:k-1},\mathcal{M}_m\big)$ by Laplace's method, as given in Equation (3.18), begin with the Law of Total Probability,

$$\begin{aligned} J &\equiv p\big(Y^k|Y^{1:k-1},\mathcal{M}_m\big) \\ &= \int_R p\big(Y^k|\Theta_m^k,Y^{1:k-1},\mathcal{M}_m\big)\,p\big(\Theta_m^k|Y^{1:k-1},\mathcal{M}_m\big)d\Theta_m^k \\ &= \int_R p\big(Y^k|\Theta_m^k,\mathcal{M}_m\big)\,p\big(\Theta_m^k|Y^{1:k-1},\mathcal{M}_m\big)d\Theta_m^k\,, \end{aligned} \tag{A.5}$$

where $R \subset \mathbb{R}^{\eta_m}$ denotes the entire parameter space and $\Theta_m^k \in \mathbb{R}^{\eta_m}$. To match the form of Equation (A.1) to the form of Eq. (A.5), let

$$f(\Theta_m^k) \equiv 1 \tag{A.6}$$

$$\phi_N(\Theta_m^k) \equiv -\log\left[p\big(Y^k|\Theta_m^k,\mathcal{M}_m\big)\,p\big(\Theta_m^k|Y^{1:k-1},\mathcal{M}_m\big)\right]\,. \tag{A.7}$$

Note that $\phi_N(\Theta_m^k)$ is proportional to the negative log-posterior probability of the parameters (see Eq. (3.4)), and thus the optimized parameter set $\hat{\Theta}_m^k$ is a minimizing point of $\phi_N(\Theta_m^k)$. Also, ϕ_N scales (roughly linearly) with N, the number of data points in Y^k, which we expect to be large. Applying Laplace's formula then leads to

$$J \approx p\big(Y^k|\hat{\Theta}_m^k,\mathcal{M}_m\big)\,p\big(\hat{\Theta}_m^k|Y^{1:k-1},\mathcal{M}_m\big)\,(2\pi)^{\eta_m/2}\,|\mathbf{H}(\hat{\Theta}_m^k)|^{-1/2}\,, \tag{A.8}$$

where

$$\mathbf{H}(\hat{\Theta}_m^k) = -\nabla_{\Theta_m^k}\nabla_{\Theta_m^k}|_{\hat{\Theta}_m^k}\,\log\,p\big(Y^k|\Theta_m^k,\mathcal{M}_m\big)\,p\big(\Theta_m^k|Y^{1:k-1},\mathcal{M}_m\big)\,.$$

Remark A.1. Although it is demonstrated only that $\hat{\Theta}_m^k$ is a local minimum, rather than the *unique absolute* minimum, of ϕ_N, the method still provides a valuable measure for model evidence. Prevailing information criteria such as the AIC and BIC are essentially approximations to (A.8) [68]. As there is considerable support for use of BIC in this context [26, 57], and Laplace's method requires fewer assumptions than BIC, it is reasonable to assume that Laplace's method will perform well in practice. See [78] for further discussion of using Laplace's method for approximating model evidence under cases of varying model identifiability. ◊

A.3 Application for Data Association Hypothesis Likelihood

This section details how Laplace's method was used to make the approximation, in Eq. (4.21c), that $P\big(h_l|Y^k,\cdot\big) \approx P\big(h_l|\hat{\Theta}^k_m,\cdot\big)$. As in Section A.2, begin with the Law of Total Probability,

$$P\big(h_l|Y^k,\cdot\big) = \int_R P\big(h_l|Y^k,\Theta^k_m,\cdot\big)\, p\big(\Theta^k_m|Y^k,\cdot\big) d\Theta^k_m \,. \tag{A.9}$$

By Bayes' Rule, the second term in the integrand is

$$p\big(\Theta^k_m|Y^k,\cdot\big) = \frac{p\big(Y^k|\Theta^k_m,\cdot\big)\, p\big(\Theta^k_m|\cdot\big)}{p\big(Y^k|\cdot\big)} = \frac{p\big(Y^k|\Theta^k_m,\cdot\big)\, p\big(\Theta^k_m|\cdot\big)}{\int_R p\big(Y^k|\Theta^k_m,\cdot\big)\, p\big(\Theta^k_m|\cdot\big) d\Theta^k_m} \,.$$

Thus, Equation (A.9) can be expressed as

$$P\big(h_l|Y^k,\cdot\big) = \frac{\int_R P\big(h_l|Y^k,\Theta^k_m,\cdot\big)\, p\big(Y^k|\Theta^k_m,\cdot\big)\, p\big(\Theta^k_m|\cdot\big) d\Theta^k_m}{\int_R p\big(Y^k|\Theta^k_m,\cdot\big)\, p\big(\Theta^k_m|\cdot\big) d\Theta^k_m} \equiv \frac{I}{J} \,. \tag{A.10}$$

We now evaluate the numerator I and denominator J of Equation (A.10) separately using Laplace's method. For I, let

$$f(\Theta^k_m) \equiv P\big(h_l|Y^k,\Theta^k_m,\cdot\big) \tag{A.11}$$

$$\phi_N(\Theta^k_m) \equiv -\log\left[p\big(Y^k|\Theta^k_m,\cdot\big)\, p\big(\Theta^k_m|\cdot\big)\right] \,. \tag{A.12}$$

Note that this function $\phi_N(\Theta^k_m)$ is the same as is Section A.2, and thus Laplace's method may be applied with the optimized parameters $\hat{\Theta}^k_m$:

$$I \approx P\big(h_l|Y^k,\hat{\Theta}^k_m,\cdot\big)\, p\big(Y^k|\hat{\Theta}^k_m,\cdot\big)\, p\big(\hat{\Theta}^k_m|\cdot\big)(2\pi)^{\eta_m/2}|\mathbf{H}_m(\hat{\Theta}^k_m)|^{-1/2} \,.$$

The integral J is exactly the same as is Section A.2:

$$J \approx p\big(Y^k|\hat{\Theta}^k_m,\cdot\big)\, p\big(\hat{\Theta}^k_m|\cdot\big)\, (2\pi)^{\eta_m/2}\, |\mathbf{H}(\hat{\Theta}^k_m)|^{-1/2} \,. \tag{A.13}$$

Combining these expressions yields:

$$P\big(h_l|Y^k,\cdot\big) \equiv \frac{I}{J} \approx P\big(h_l|Y^k,\hat{\Theta}^k_m,\cdot\big)\,. \tag{A.14}$$

Finally we note that the probability of h_l is independent of Y^k if we are already conditioning on $\hat{\Theta}^k_m$, since the latter is what is used to form the hypothesis. To show this another way, note that the mixture model governing the data Y^k is entirely determined by Θ^k_m and $\mathcal{M}_m$ (h_l has no influence) and thus $p\big(Y^k|h_l,\hat{\Theta}^k_m,\cdot\big) = p\big(Y^k|\hat{\Theta}^k_m,\cdot\big)$. Then, using Bayes' Rule,

$$P\big(h_l|Y^k,\hat{\Theta}^k_m,\cdot\big) = \frac{p\big(Y^k|h_l,\hat{\Theta}^k_m,\cdot\big)P\big(h_l|\hat{\Theta}^k_m,\cdot\big)}{p\big(Y^k|\hat{\Theta}^k_m,\cdot\big)} = P\big(h_l|\hat{\Theta}^k_m,\cdot\big)\,.$$

Thus,

$$P\big(h_l|Y^k,\cdot\big) \approx P\big(h_l|\hat{\Theta}^k_m,\cdot\big)\,,$$

which is the result used in Eq. (4.21c).

Appendix B

Hessian Matrix for Model Evidence

This appendix calculates the derivatives required to compute the Hessian matrix

$$\mathbf{H}(\Theta) = -\nabla_\Theta \nabla_\Theta \, \log p\big(Y^k | \Theta, \mathcal{M}\big) \, p\big(\Theta | Y^{1:k-1}, \mathcal{M}\big) \, , \tag{B.1}$$

which is used in Equation (3.18) for approximating the model evidence via Laplace's method. The model evidence is used both in the model selection procedure of Chapter 3 and as a measure of model hypothesis plausibility in Chapter 4. The equations derived in this appendix are specific to a 2-dimensional Gaussian mixture model with a common-volume covariance model (see Remark 3.2) and an outlier component (of any form). This appendix provides intermediary steps used to obtain the Hessian's derivatives so that they may be adapted to similar models.

B.1 Preliminaries

This section presents the problem of finding the Hessian derivatives and divides the effort into more manageable components. Additionally, some useful identities from matrix calculus are recalled.

B.1.1 Problem Statement and Decomposition

The Hessian matrix (B.1) consists of mixed second derivatives with respect to the model parameters Θ. For the model used in this theses (Remark 3.2),

$$\Theta = \big\{ \{\pi_g, \mu_g, c_{11,g}, c_{12,g}\}_{g=1}^{G}, \lambda \big\} \ , \tag{B.2}$$

where $\pi_g \in \mathbb{R}$ and $\mu_g \in \mathbb{R}^d$ are respectively the mixture weights and means of the gth mixture component. The scalar parameters $c_{11,g}$, $c_{12,g}$, and λ define the covariance matrix of the gth component, $\Sigma_g \in \mathbb{R}^{d \times d}$, under the common-volume constraint: $\Sigma_g = \lambda \, C_g$. $C_g = D_g \, A_g \, D_g^T$, where D_g is the matrix of normalized eigenvectors of Σ_g and A_g is the diagonal matrix of normalized eigenvalues of

Σ_g [67]. Note that C_g, like Σ_g, is symmetric and has $\frac{d(d+1)}{2} - 1$ independent elements; restricting now to the case $d = 2$, these constraints lead to:

$$C_g = \begin{bmatrix} c_{11,g} & c_{12,g} \\ c_{12,g} & c_{22,g} \end{bmatrix} , \qquad c_{22,g} = \frac{c_{12,g}^2 + 1}{c_{11,g}} \, . \tag{B.3}$$

Note that the correlation coefficient $\rho_{12,g} = \frac{\sigma_{12,g}}{\sqrt{\sigma_{11,g}\sigma_{22,g}}} = \frac{c_{12,g}}{\sqrt{c_{11,g}c_{22,g}}}$ may be used as a parameter rather than $c_{12,g}$, in which case, $c_{22,g} = \frac{1}{c_{11,g}(1-\rho_{12,g}^2)}$.

It is convenient to consider the derivatives of the likelihood and prior terms separately:

$$\mathbf{H}(\Theta) = -\left(\mathbf{H}^l(\Theta) + \mathbf{H}^p(\Theta)\right) , \tag{B.4}$$

where

$$\mathbf{H}^l(\Theta) \equiv \nabla\nabla \log p\big(Y^k|\Theta, \mathcal{M}\big) \tag{B.5}$$

and

$$\mathbf{H}^p(\Theta) \equiv \nabla\nabla \log p\big(\Theta|Y^{1:k-1}, \mathcal{M}\big) \, . \tag{B.6}$$

The martices $\mathbf{H}^l$ and $\mathbf{H}^p$ are derived in Sections B.2 and B.3.

B.1.2 Useful Matrix Calculus Identities

For vector x and symmetric martix Σ, with elements σ_{ij}:

$$\frac{\partial}{\partial x} x^T \Sigma x = (\Sigma + \Sigma^T)x = 2\Sigma x \tag{B.7}$$

$$\frac{\partial}{\partial x^T} x^T \Sigma x = x^T(\Sigma^T + \Sigma) = 2x^T\Sigma \tag{B.8}$$

$$\frac{\partial}{\partial \sigma_{ij}} \Sigma^{-1} = -\Sigma^{-1} \frac{\partial \Sigma}{\partial \sigma_{ij}} \Sigma^{-1} \tag{B.9}$$

$$\frac{\partial}{\partial \sigma_{ij}} \log(\det(\Sigma)) = \mathrm{tr}(\Sigma^{-1} \frac{\partial \Sigma}{\partial \sigma_{ij}}) \, . \tag{B.10}$$

B.1.3 Derivatives of Gaussian PDF

The following derivatives of the Gaussian density $f_{\mathcal{N}}\big(y_i|\mu, \Sigma\big)$ will be helpful later.

Means

$$
\begin{aligned}
\frac{\partial}{\partial\mu}(f_{\mathcal{N}}(y_i|\mu,\Sigma)) &= f_{\mathcal{N}}(y_i|\mu,\Sigma)\frac{\partial}{\partial\mu}\left[-\frac{1}{2}(y_i-\mu)^T\Sigma^{-1}(y_i-\mu)\right] \\
&= f_{\mathcal{N}}(y_i|\mu,\Sigma)\left[-\frac{1}{2}(-\mathbf{I})2\Sigma^{-1}(y_i-\mu)\right] \\
&= f_{\mathcal{N}}(y_i|\mu,\Sigma)\Sigma^{-1}(y_i-\mu) \\
\frac{\partial}{\partial\mu^T}(f_{\mathcal{N}}(y_i|\mu,\Sigma)) &= f_{\mathcal{N}}(y_i|\mu,\Sigma)(y_i-\mu)^T\Sigma^{-1}
\end{aligned}
$$

Covariance Matrix Elements

$$
\begin{aligned}
&\frac{\partial}{\partial c_{ij}}(f_{\mathcal{N}}(y_i|\mu,\lambda C)) \\
&\qquad = -\frac{1}{2}f_{\mathcal{N}}(y_i|\mu,\lambda C)\left[\frac{\partial}{\partial c_{ij}}\log(\lambda^d\det(C)) + (y_i-\mu)^T\frac{\partial}{\partial c_{ij}}(\lambda C)^{-1}(y_i-\mu)\right] \\
&\qquad = \frac{1}{2}f_{\mathcal{N}}(y_i|\mu,\Sigma)\left[\frac{1}{\lambda}(y_i-\mu)^TC^{-1}\frac{\partial C}{\partial c_{ij}}C^{-1}(y_i-\mu) - \operatorname{tr}(C^{-1}\frac{\partial C}{\partial c_{ij}})\right]
\end{aligned}
$$

Data Variable

$$
\begin{aligned}
\frac{\partial}{\partial y_i}(f_{\mathcal{N}}(y_i|\mu,\Sigma)) &= f_{\mathcal{N}}(y_i|\mu,\Sigma)\frac{\partial}{\partial\mu}\left[-\frac{1}{2}(y_i-\mu)^T\Sigma^{-1}(y_i-\mu)\right] \\
&= f_{\mathcal{N}}(y_i|\mu,\Sigma)\left[-\frac{1}{2}(I)2\Sigma^{-1}(y_i-\mu)\right] \\
&= -f_{\mathcal{N}}(y_i|\mu,\Sigma)\Sigma^{-1}(y_i-\mu)
\end{aligned}
$$

B.2 Derivatives of Log-Likelihood Term

The goal of this section is to derive $\mathbf{H}^l$ so that it may be used in (B.4). Recall that the likelihood (3.5) is

$$
p(Y^k|\Theta,\mathcal{M}) = \prod_{i=1}^{N} p(y_i|\Theta,\mathcal{M}),
$$

and let

$$p\big(y_i|\Theta,\mathcal{M}\big) = \sum_{g=0}^{G} \pi_g f_g\big(y_i|\theta_g\big) \tag{B.11a}$$

$$= \pi_0 f_0\big(y_i|\theta_0\big) + \sum_{g=1}^{G} \pi_g f_{\mathcal{N}}\big(y_i|\mu_g,\Sigma_g\big) \tag{B.11b}$$

$$= \Big(1-\sum_{g=1}^{G}\pi_g\Big) f_{0,i} + \sum_{g=1}^{G}\pi_g \Phi_{g,i}\,, \tag{B.11c}$$

where the last equality has used the fact that $\pi_0 \notin \Theta$ is chosen to satisfy the mixture weight's constraint to sum to unity and the shorthand notation

$$f_{0,i} \equiv f_0\big(y_i|\theta_0\big) \qquad \Phi_{g,i} \equiv f_{\mathcal{N}}\big(y_i|\mu_g,\Sigma_g\big)$$

is used hereafter. Note also that $\theta_0 \notin \Theta$ because these parameters are selected by the user and not optimized by the EM algorithm. Next, it is helpful to further decompose the problem by letting

$$\mathbf{H}^l_i \equiv \nabla\nabla\, l_i \equiv \nabla\nabla \log p\big(y_i|\Theta,\mathcal{M}\big)\,,$$

where $l_i \equiv \log p\big(y_i|\Theta,\mathcal{M}\big)$, so that

$$\mathbf{H}^l \equiv \nabla\nabla \log p\big(Y^k|\Theta,\mathcal{M}\big) = \sum_{i=1}^{N}\mathbf{H}^l_i\,. \tag{B.12}$$

The matrix $\mathbf{H}^l_i$ has a block form:

$$\mathbf{H}^l_i = \begin{bmatrix} \mathbf{H}^l_{11,i} & \mathbf{H}^l_{12,i} & \cdots & \mathbf{H}^l_{1G,i} & \mathbf{L}_{1,i} \\ {\mathbf{H}^l_{12,i}}^T & \mathbf{H}^l_{22,i} & \cdots & \mathbf{H}^l_{2G,i} & \mathbf{L}_{2,i} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ {\mathbf{H}^l_{1G,i}}^T & {\mathbf{H}^l_{2G,i}}^T & \cdots & \mathbf{H}^l_{GG,i} & \mathbf{L}_{G,i} \\ {\mathbf{L}_{1,i}}^T & {\mathbf{L}_{2,i}}^T & \cdots & {\mathbf{L}_{G,i}}^T & \mathbf{l}_i \end{bmatrix}, \tag{B.13}$$

where

$$\mathbf{H}^l_{gh,i} \equiv \begin{bmatrix} \frac{\partial^2 l_i}{\partial\pi_g\partial\pi_h} & \frac{\partial^2 l_i}{\partial\pi_g\partial\mu_h^T} & \frac{\partial^2 l_i}{\partial\pi_g\partial c_{11,h}} & \frac{\partial^2 l_i}{\partial\pi_g\partial c_{12,h}} \\ \frac{\partial^2 l_i}{\partial\mu_g\partial\pi_h} & \frac{\partial^2 l_i}{\partial\mu_g\partial\mu_h^T} & \frac{\partial^2 l_i}{\partial\mu_g\partial c_{11,h}} & \frac{\partial^2 l_i}{\partial\mu_g\partial c_{12,h}} \\ \frac{\partial^2 l_i}{\partial c_{11,g}\partial\pi_h} & \frac{\partial^2 l_i}{\partial c_{11,g}\partial\mu_h^T} & \frac{\partial^2 l_i}{\partial c_{11,g}\partial c_{11,h}} & \frac{\partial^2 l_i}{\partial c_{11,g}\partial c_{12,h}} \\ \frac{\partial^2 l_i}{\partial c_{12,g}\partial\pi_h} & \frac{\partial^2 l_i}{\partial c_{12,g}\partial\mu_h^T} & \frac{\partial^2 l_i}{\partial c_{12,g}\partial c_{11,h}} & \frac{\partial^2 l_i}{\partial c_{12,g}\partial c_{12,h}} \end{bmatrix} \tag{B.14}$$

and

$$\mathbf{L}_{g,i} \equiv \begin{bmatrix} \frac{\partial^2 l_i}{\partial \pi_g \partial \lambda} \\ \frac{\partial^2 l_i}{\partial \mu_g \partial \lambda} \\ \frac{\partial^2 l_i}{\partial c_{11,g} \partial \lambda} \\ \frac{\partial^2 l_i}{\partial c_{12,g} \partial \lambda} \end{bmatrix} \tag{B.15}$$

and

$$\mathbf{l}_i \equiv \frac{\partial^2 l_i}{\partial \lambda^2} \, . \tag{B.16}$$

Thus the problem now consists of finding the second derivatives in (B.14), (B.15), and (B.16). First consider the general form of these equations given l_i: For each element $t \in \Theta$:

$$\frac{\partial l_i}{\partial t} = \frac{\partial}{\partial t} \log \sum_{l=0}^{G} \pi_l f_{l,i} = \frac{1}{\sum_l \pi_l f_{l,i}} \sum_h \frac{\partial}{\partial t} \pi_h f_{h,i} \, .$$

Selecting any two elements $t \in \Theta$ and $s \in \Theta$, the form of all the second derivative terms is:

$$\begin{aligned} \frac{\partial^2 l_i}{\partial t \partial s} &= \frac{\partial^2}{\partial t \partial s} \log \sum_{l=0}^{G} \pi_l f_{l,i} \\ &= \frac{\partial}{\partial s} \left[\frac{1}{\sum_l \pi_l f_{l,i}} \frac{\partial}{\partial t} \sum_m \pi_m f_{m,i} \right] \\ &= -\frac{1}{\left(\sum_l \pi_l f_{l,i}\right)^2} \left[\frac{\partial}{\partial t} \sum_m \pi_m f_{m,i} \right] \left[\frac{\partial}{\partial s} \sum_n \pi_n f_{n,i} \right] + \frac{1}{\sum_l \pi_l f_{l,i}} \frac{\partial^2}{\partial t \partial s} \sum_m \pi_m f_{m,i} \, . \end{aligned} \tag{B.17}$$

For the elements in $\mathbf{H}^l_{gh,i}$, the fact that most parameters are related to a particular Gaussian component simplifies (B.17) to:

$$\frac{\partial^2 l_i}{\partial t_g \partial s_h} = -\frac{1}{\left(\sum_l \pi_l f_l\right)^2} \left[\frac{\partial}{\partial t_g} \sum_m \pi_m f_{m,i} \right] \left[\frac{\partial}{\partial s_h} \sum_n \pi_n f_{n,i} \right] + \delta_{gh} \frac{1}{\sum_l \pi_l f_l} \frac{\partial^2}{\partial t_g \partial s_g} (\pi_g \Phi_g) \tag{B.18}$$

where δ_{gh} is the Kronecker delta, so the last term on the right-hand side is needed only when taking both derivatives with respect to the same mixture component (i.e., in $\mathbf{H}^l_{11,i}$, $\mathbf{H}^l_{22,i}$, etc.).

Thus, all first and second derivatives of $\sum_m \pi_m f_{m,i}$ are required to complete the derivation of $\mathbf{H}^l_i$. These derivatives are developed in Section B.2.1 and Section B.2.2.

B.2.1 First Derivatives of $\sum_m \pi_m f_{m,i}$

Using the Gaussian derivatives already provided in B.1.3, the following results are easily reached.

Mixture Weights

$$\frac{\partial}{\partial \pi_g} \sum_m \pi_m f_{m,i} = \Phi_{g,i} - f_{0,i}$$

Means

$$\frac{\partial}{\partial \mu_g} \sum_m \pi_m f_{m,i} = \pi_g \Phi_{g,i} \Sigma_g^{-1} (y_i - \mu_g)$$
$$\frac{\partial}{\partial \mu_g^T} \sum_m \pi_m f_{m,i} = \pi_g \Phi_{g,i,i} (y_i - \mu_g)^T \Sigma_g^{-1}$$

(Co)variances

$$\frac{\partial}{\partial c_{ij,g}} \sum_m \pi_m f_{m,i} = \frac{1}{2} \pi_g \Phi_{g,i} \left[\frac{1}{\lambda} (y_i - \mu_g)^T C_g^{-1} \frac{\partial C_g}{\partial c_{ij,g}} C_g^{-1} (y_i - \mu_g) - \mathrm{tr}(C_g^{-1} \frac{\partial C_g}{\partial c_{ij,g}}) \right]$$

where

$$C = \begin{bmatrix} c_{11} & c_{12} \\ c_{12} & \frac{c_{12}^2 + 1}{c_{11}} \end{bmatrix}$$

$$\frac{\partial C_g}{\partial c_{11,g}} = \begin{bmatrix} 1 & 0 \\ 0 & -\frac{c_{12,g}^2 + 1}{c_{11,g}^2} \end{bmatrix}$$
$$\frac{\partial C_g}{\partial c_{12,g}} = \begin{bmatrix} 0 & 1 \\ 1 & 2\frac{c_{12,g}}{c_{11,g}} \end{bmatrix}$$

Covariance Volume

$$\begin{aligned} \frac{\partial}{\partial \lambda} \sum_m \pi_m f_{m,i} &= \sum_g -\frac{1}{2} \pi_g \Phi_{g,i} \left[\frac{\partial}{\partial \lambda} \log(\det(\lambda C_g)) + (y_i - \mu)^T \frac{\partial}{\partial \lambda} (\lambda C_g)^{-1} (y_i - \mu) \right] \\ &= \sum_g -\frac{1}{2} \pi_g \Phi_{g,i} \left[\frac{\partial}{\partial \lambda} \log \lambda^d + \frac{\partial}{\partial \lambda} \lambda^{-1} (y_i - \mu)^T C_g^{-1} (y_i - \mu) \right] \\ &= \sum_g \frac{1}{2} \pi_g \Phi_{g,i} \left[\frac{1}{\lambda^2} (y_i - \mu)^T C_g^{-1} (y_i - \mu) - \frac{d}{\lambda} \right] \\ &= \sum_g \frac{1}{2\lambda} \pi_g \Phi_{g,i} \left[(y_i - \mu)^T \Sigma_g^{-1} (y_i - \mu) - d \right] \end{aligned}$$

B.2.2 Second Derivatives of $\sum_m \pi_m f_{m,i}$

Below, the component-dependence (i.e., the g or h subscript) is often suppressed for clarity. This does not introduce ambiguity since for the derivatives in $\mathbf{H}_{gh,i}^l$ the second derivative is needed only when the parameters belong to the same component (as in (B.18)) and for the derivatives in $\mathbf{L}_{g,i}$ there is only one component. The following list provides the required derivatives for the last term in Equation (B.18).

Mixture Weights – Mixture Weights

$$\frac{\partial^2}{\partial \pi_g \partial \pi_h} \sum_m \pi_m f_{m,i} = 0$$

Means – Means

$$\begin{aligned}
\frac{\partial^2}{\partial \mu \partial \mu^T} \sum_m \pi_m f_{m,i} &= \frac{\partial}{\partial \mu} \left[\pi \Phi_i (y_i - \mu)^T \Sigma^{-1} \right] \\
&= \left[\frac{\partial}{\partial \mu} \pi \Phi_i \right] (y_i - \mu)^T \Sigma^{-1} + \pi \Phi_i \left[\frac{\partial}{\partial \mu} (y_i - \mu)^T \Sigma^{-1} \right] \\
&= \pi \Phi_i \Sigma^{-1} (y_i - \mu)(y_i - \mu)^T \Sigma^{-1} + \pi \Phi_i (-\mathbf{I}) \Sigma^{-1} \\
&= \pi \Phi_i \left[\Sigma^{-1} (y_i - \mu)(y_i - \mu)^T \Sigma^{-1} - \Sigma^{-1} \right]
\end{aligned}$$

(Co)variances – (Co)variances

$$
\begin{aligned}
&\frac{\partial^2}{\partial c_{ij}\partial c_{kl}} \sum_m \pi_m f_{m,i} \\
&\quad = \frac{\partial}{\partial c_{kl}} \Bigg(\frac{1}{2}\pi\Phi_i \left[\frac{1}{\lambda}(y_i-\mu)^T C^{-1}\frac{\partial C}{\partial c_{ij}} C^{-1}(y_i-\mu) - \mathrm{tr}(C^{-1}\frac{\partial C}{\partial c_{ij}}) \right] \Bigg) \\
&\quad = \frac{1}{2} \left[\frac{\partial}{\partial c_{kl}}\pi\Phi_i \right] \left[\frac{1}{\lambda}(y_i-\mu)^T C^{-1}\frac{\partial C}{\partial c_{ij}} C^{-1}(y_i-\mu) - \mathrm{tr}(C^{-1}\frac{\partial C}{\partial c_{ij}}) \right] \\
&\qquad + \frac{1}{2}\pi\Phi_i \frac{\partial}{\partial c_{kl}} \left[\frac{1}{\lambda}(y_i-\mu)^T C^{-1}\frac{\partial C}{\partial c_{ij}} C^{-1}(y_i-\mu) - \mathrm{tr}(C^{-1}\frac{\partial C}{\partial c_{ij}}) \right] \\
&\quad = \frac{1}{2} \left[\frac{1}{2}\pi\Phi_i \left(\frac{1}{\lambda}(y_i-\mu)^T C^{-1}\frac{\partial C}{\partial c_{kl}} C^{-1}(y_i-\mu) - \mathrm{tr}(C^{-1}\frac{\partial C}{\partial c_{kl}}) \right) \right] \\
&\qquad\quad \cdot \left[\frac{1}{\lambda}(y_i-\mu)^T C^{-1}\frac{\partial C}{\partial c_{ij}} C^{-1}(y_i-\mu) - \mathrm{tr}(C^{-1}\frac{\partial C}{\partial c_{ij}}) \right] \\
&\qquad + \frac{1}{2}\pi\Phi_i \Bigg[\frac{1}{\lambda}(y_i-\mu)^T \left(\frac{\partial C^{-1}}{\partial c_{kl}}\frac{\partial C}{\partial c_{ij}}C^{-1} + C^{-1}\frac{\partial^2 C}{\partial c_{ij}\partial c_{kl}}C^{-1} + C^{-1}\frac{\partial C}{\partial c_{ij}}\frac{\partial C^{-1}}{\partial c_{kl}} \right)(y_i-\mu) \\
&\qquad\quad - \mathrm{tr}(\frac{\partial C^{-1}}{\partial c_{kl}}\frac{\partial C}{\partial c_{ij}} + C^{-1}\frac{\partial^2 C}{\partial c_{ij}\partial c_{kl}}) \Bigg] \\
&\quad = \frac{1}{2}\pi\Phi_i \Bigg(\frac{1}{2} \left[\frac{1}{\lambda}(y_i-\mu)^T C^{-1}\frac{\partial C}{\partial c_{kl}} C^{-1}(y_i-\mu) - \mathrm{tr}(C^{-1}\frac{\partial C}{\partial c_{kl}}) \right] \\
&\qquad\quad \cdot \left[\frac{1}{\lambda}(y_i-\mu)^T C^{-1}\frac{\partial C}{\partial c_{ij}} C^{-1}(y_i-\mu) - \mathrm{tr}(C^{-1}\frac{\partial C}{\partial c_{ij}}) \right] \\
&\qquad + \frac{1}{\lambda}(y_i-\mu)^T \Bigg(-C^{-1}\frac{\partial C}{\partial c_{kl}}C^{-1}\frac{\partial C}{\partial c_{ij}}C^{-1} + C^{-1}\frac{\partial^2 C}{\partial c_{ij}\partial c_{kl}}C^{-1} \\
&\qquad\qquad - C^{-1}\frac{\partial C}{\partial c_{ij}}C^{-1}\frac{\partial C}{\partial c_{kl}}C^{-1} \Bigg)(y_i-\mu) \\
&\qquad - \mathrm{tr}\left(-C^{-1}\frac{\partial C}{\partial c_{kl}}C^{-1}\frac{\partial C}{\partial c_{ij}} + C^{-1}\frac{\partial^2 C}{\partial c_{ij}\partial c_{kl}} \right) \Bigg) \\
&\quad = \frac{1}{2}\pi\Phi_i \Bigg(\frac{1}{2} \left[\frac{1}{\lambda}(y_i-\mu)^T C^{-1}\frac{\partial C}{\partial c_{kl}} C^{-1}(y_i-\mu) - \mathrm{tr}(C^{-1}\frac{\partial C}{\partial c_{kl}}) \right] \\
&\qquad\quad \cdot \left[\frac{1}{\lambda}(y_i-\mu)^T C^{-1}\frac{\partial C}{\partial c_{ij}} C^{-1}(y_i-\mu) - \mathrm{tr}(C^{-1}\frac{\partial C}{\partial c_{ij}}) \right] \\
&\qquad - \frac{1}{\lambda}(y_i-\mu)^T C^{-1} \left(\frac{\partial C}{\partial c_{kl}}C^{-1}\frac{\partial C}{\partial c_{ij}} - \frac{\partial^2 C}{\partial c_{ij}\partial c_{kl}} + \frac{\partial C}{\partial c_{ij}}C^{-1}\frac{\partial C}{\partial c_{kl}} \right) C^{-1}(y_i-\mu) \\
&\qquad + \mathrm{tr}\left(C^{-1}\frac{\partial C}{\partial c_{kl}}C^{-1}\frac{\partial C}{\partial c_{ij}} - C^{-1}\frac{\partial^2 C}{\partial c_{ij}\partial c_{kl}} \right) \Bigg)
\end{aligned}
$$

where for the 2D case,

$$\begin{aligned}
\frac{\partial^2 C}{\partial c_{11}^2} &= \frac{\partial}{\partial c_{11}} \begin{bmatrix} 1 & 0 \\ 0 & -\frac{c_{12}^2+1}{c_{11}^2} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 2\frac{c_{12}^2+1}{c_{11}^3} \end{bmatrix} \\
\frac{\partial^2 C}{\partial c_{11} \partial c_{12}} &= \frac{\partial}{\partial c_{12}} \begin{bmatrix} 1 & 0 \\ 0 & -\frac{c_{12}^2+1}{c_{11}^2} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & -2\frac{c_{12}}{c_{11}^2} \end{bmatrix} \\
\frac{\partial^2 C}{\partial c_{12}^2} &= \frac{\partial}{\partial c_{12}} \begin{bmatrix} 0 & 1 \\ 1 & 2\frac{c_{12}}{c_{11}} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & \frac{2}{c_{11}} \end{bmatrix}
\end{aligned}$$

Volume – Volume

$$\begin{aligned}
\frac{\partial^2}{\partial \lambda^2} \sum_m \pi_m f_{m,i} &= \frac{\partial}{\partial \lambda} \left(\sum_g \frac{1}{2} \pi_g \Phi_{g,i} \left[\frac{1}{\lambda^2} (y_i - \mu_g)^T C_g^{-1} (y_i - \mu_g) - \frac{d}{\lambda} \right] \right) \\
&= \sum_g \frac{1}{2} \left[\frac{\partial}{\partial \lambda} \pi_g \Phi_{g,i} \right] \left[\frac{1}{\lambda^2} (y_i - \mu_g)^T C_g^{-1} (y_i - \mu_g) - \frac{d}{\lambda} \right] \\
&\quad + \sum_g \frac{1}{2} \pi_g \Phi_{g,i} \frac{\partial}{\partial \lambda} \left[\frac{1}{\lambda^2} (y_i - \mu_g)^T C_g^{-1} (y_i - \mu_g) - \frac{d}{\lambda} \right] \\
&= \sum_g \frac{1}{2} \left(\frac{1}{2} \pi_g \Phi_{g,i} \left[\frac{1}{\lambda^2} (y_i - \mu_g)^T C_g^{-1} (y_i - \mu_g) - \frac{d}{\lambda} \right] \right) \\
&\quad \cdot \left[\frac{1}{\lambda^2} (y_i - \mu_g)^T C_g^{-1} (y_i - \mu_g) - \frac{d}{\lambda} \right] \\
&\quad - \sum_g \pi_g \Phi_{g,i} \left[\frac{1}{\lambda^3} (y_i - \mu_g)^T C_g^{-1} (y_i - \mu_g) - \frac{d}{2\lambda^2} \right] \\
&= \sum_g \pi_g \Phi_{g,i} \left(\frac{1}{4} \left[\frac{1}{\lambda^2} (y_i - \mu_g)^T C_g^{-1} (y_i - \mu_g) - \frac{d}{\lambda} \right]^2 \right. \\
&\quad \left. - \left[\frac{1}{\lambda^3} (y_i - \mu_g)^T C_g^{-1} (y_i - \mu_g) - \frac{d}{2\lambda^2} \right] \right) \\
&= \sum_g \pi_g \Phi_{g,i} \frac{1}{\lambda^2} \left(\frac{1}{4} \left[(y_i - \mu_g)^T \Sigma^{-1} (y_i - \mu_g) - d \right]^2 \right. \\
&\quad \left. - \left[(y_i - \mu_g)^T \Sigma^{-1} (y_i - \mu_g) - \frac{d}{2} \right] \right)
\end{aligned}$$

Means – Weights

$$\begin{aligned}\frac{\partial^2}{\partial\mu\partial\pi}\sum_m \pi_m f_{m,i} &= \frac{\partial}{\partial\mu}\left(\Phi_i - f_{0,i}\right)\\ &= \Phi_i\Sigma^{-1}(y_i-\mu)\end{aligned}$$

(Co)variances – Weights

$$\begin{aligned}\frac{\partial^2}{\partial c_{ij}\partial\pi}\sum_m \pi_m f_{m,i} &= \frac{\partial}{\partial c_{ij}}\left(\Phi_i - f_{0,i}\right)\\ &= \frac{1}{2}\Phi_i\left[\frac{1}{\lambda}(y_i-\mu)^T C^{-1}\frac{\partial C}{\partial c_{ij}}C^{-1}(y_i-\mu) - \mathrm{tr}(C^{-1}\frac{\partial C}{\partial c_{ij}})\right]\end{aligned}$$

Volume – Weights

$$\begin{aligned}\frac{\partial^2}{\partial\lambda\partial\pi}\sum_m \pi_m f_{m,i} &= \frac{\partial}{\partial\lambda}\left(\Phi_i - f_{0,i}\right)\\ &= \frac{1}{2\lambda}\Phi_i\left[(y_i-\mu)^T\Sigma_g^{-1}(y_i-\mu) - d\right]\end{aligned}$$

Means – (Co)variances

$$
\begin{aligned}
&\frac{\partial^2}{\partial\mu\partial c_{ij}} \sum_m \pi_m f_{m,i} \\
&\qquad= \frac{\partial}{\partial\mu}\left(\frac{1}{2}\pi\Phi_i\left[\frac{1}{\lambda}(y_i-\mu)^T C^{-1}\frac{\partial C}{\partial c_{ij}}C^{-1}(y_i-\mu) - \mathrm{tr}(C^{-1}\frac{\partial C}{\partial c_{ij}})\right]\right) \\
&\qquad= \frac{1}{2}\left[\frac{\partial}{\partial\mu}\pi\Phi_i\right]\left[\frac{1}{\lambda}(y_i-\mu)^T C^{-1}\frac{\partial C}{\partial c_{ij}}C^{-1}(y_i-\mu) - \mathrm{tr}(C^{-1}\frac{\partial C}{\partial c_{ij}})\right] \\
&\qquad\quad+ \frac{1}{2}\pi\Phi_i\frac{\partial}{\partial\mu}\left[\frac{1}{\lambda}(y_i-\mu)^T C^{-1}\frac{\partial C}{\partial c_{ij}}C^{-1}(y_i-\mu) - \mathrm{tr}(C^{-1}\frac{\partial C}{\partial c_{ij}})\right] \\
&\qquad= \frac{1}{2}\pi\Phi_i\Sigma^{-1}(y_i-\mu)\left[\frac{1}{\lambda}(y_i-\mu)^T C^{-1}\frac{\partial C}{\partial c_{ij}}C^{-1}(y_i-\mu) - \mathrm{tr}(C^{-1}\frac{\partial C}{\partial c_{ij}})\right] \\
&\qquad\quad+ \frac{1}{2}\pi\Phi_i\left[\frac{1}{\lambda}(-\mathbf{I})2C^{-1}\frac{\partial C}{\partial c_{ij}}C^{-1}(y_i-\mu)\right] \\
&\qquad= \frac{1}{2}\pi\Phi_i\frac{1}{\lambda}\left[\frac{1}{\lambda}(y_i-\mu)^T C^{-1}\frac{\partial C}{\partial c_{ij}}C^{-1}(y_i-\mu) - \mathrm{tr}(C^{-1}\frac{\partial C}{\partial c_{ij}})\right]C^{-1}(y_i-\mu) \\
&\qquad\quad- \pi\Phi_i\frac{1}{\lambda}C^{-1}\frac{\partial C}{\partial c_{ij}}C^{-1}(y_i-\mu) \\
&\qquad= \pi\Phi_i\frac{1}{\lambda}\Bigg(\frac{1}{2}\left[\frac{1}{\lambda}(y_i-\mu)^T C^{-1}\frac{\partial C}{\partial c_{ij}}C^{-1}(y_i-\mu) - \mathrm{tr}(C^{-1}\frac{\partial C}{\partial c_{ij}})\right]C^{-1} \\
&\qquad\quad- C^{-1}\frac{\partial C}{\partial c_{ij}}C^{-1}\Bigg)(y_i-\mu)
\end{aligned}
$$

Means – Volume

$$
\begin{aligned}
\frac{\partial^2}{\partial\mu\partial\lambda}\sum_m \pi_m f_{m,i} &= \frac{\partial}{\partial\mu}\left(\frac{1}{2}\pi\Phi_i\left[\frac{1}{\lambda^2}(y_i-\mu)^T C^{-1}(y_i-\mu) - \frac{d}{\lambda}\right]\right) \\
&= \frac{1}{2}\pi\Phi_i\Sigma^{-1}(y_i-\mu)\left[\frac{1}{\lambda^2}(y_i-\mu)^T C^{-1}(y_i-\mu) - \frac{d}{\lambda}\right] \\
&\quad+ \frac{1}{2}\pi\Phi_i\frac{1}{\lambda^2}2C^{-1}(y_i-\mu)(-\mathbf{I}) \\
&= \pi\Phi_i\Bigg(\frac{1}{2\lambda}C^{-1}(y_i-\mu)\left[\frac{1}{\lambda^2}(y_i-\mu)^T C^{-1}(y_i-\mu) - \frac{d}{\lambda}\right] \\
&\quad- \frac{1}{\lambda^2}C^{-1}(y_i-\mu)\Bigg) \\
&= \frac{1}{\lambda^2}\pi\Phi_i\left(\frac{1}{2}\left[\frac{1}{\lambda}(y_i-\mu)^T C^{-1}(y_i-\mu) - d\right] - 1\right)C^{-1}(y_i-\mu) \\
&= \frac{1}{\lambda}\pi\Phi_i\left(\frac{1}{2}\left[(y_i-\mu)^T\Sigma^{-1}(y_i-\mu) - d\right] - 1\right)\Sigma^{-1}(y_i-\mu)
\end{aligned}
$$

<u>(Co)variances – Volume</u>

$$\begin{aligned}
\frac{\partial^2}{\partial c_{ij}\partial\lambda}\sum_m \pi_m f_{m,i} &= \frac{\partial}{\partial c_{ij}}\left(\frac{1}{2}\pi\Phi_i\left[\frac{1}{\lambda^2}(y_i-\mu)^T C^{-1}(y_i-\mu)-\frac{d}{\lambda}\right]\right)\\
&= \frac{1}{2}\left[\frac{\partial}{\partial c_{ij}}\pi_g\Phi_g\right]\left[\frac{1}{\lambda^2}(y_i-\mu)^T C^{-1}(y_i-\mu)-\frac{d}{\lambda}\right]\\
&\quad+\frac{1}{2}\pi\Phi_i\left[\frac{\partial}{\partial c_{ij}}\frac{1}{\lambda^2}(y_i-\mu)^T C^{-1}(y_i-\mu)\right]\\
&= \frac{1}{2}\frac{1}{2}\pi\Phi_i\left[\frac{1}{\lambda}(y_i-\mu)^T C^{-1}\frac{\partial C}{\partial c_{ij}}C^{-1}(y_i-\mu)-\mathrm{tr}(C^{-1}\frac{\partial C}{\partial c_{ij}})\right]\\
&\qquad\cdot\left[\frac{1}{\lambda^2}(y_i-\mu)^T C^{-1}(y_i-\mu)-\frac{d}{\lambda}\right]\\
&\quad-\frac{1}{2}\pi\Phi_i\frac{1}{\lambda^2}(y_i-\mu)^T C^{-1}\frac{\partial C}{\partial c_{ij}}C^{-1}(y_i-\mu)\\
&= \frac{1}{2\lambda^2}\pi\Phi_i\Bigg(\frac{1}{2}\left[(y_i-\mu)^T C^{-1}\frac{\partial C}{\partial c_{ij}}C^{-1}(y_i-\mu)-\lambda\ \mathrm{tr}(C^{-1}\frac{\partial C}{\partial c_{ij}})\right]\\
&\qquad\cdot\left[\frac{1}{\lambda}(y_i-\mu)^T C^{-1}(y_i-\mu)-d\right]\\
&\quad-(y_i-\mu)^T C^{-1}\frac{\partial C}{\partial c_{ij}}C^{-1}(y_i-\mu)\Bigg)
\end{aligned}$$

B.2.3 Note on Covariance Elements

As noted earlier, several parameterizations of the covariance matrix Σ_g are possible. Above, the derivatives with respect to c_{ij} (and λ) were derived. However, one may wish to choose the the correlation coefficient ρ_{ij} instead of c_{ij} as the parameters representing the off-diagonal elements of the matrix C. Recall that $c_{ij} = \frac{\rho_{ij}}{\sqrt{1-\rho_{ij}^2}}$. Thus, by the chain rule, we have

$$\frac{\partial}{\partial\rho_{ij,g}}(\cdot) = \frac{\partial c_{ij,g}}{\partial\rho_{ij,g}}\frac{\partial}{\partial c_{ij,g}}(\cdot) = \left[\left(1-\rho_{ij,g}^2\right)^{-1/2}+\rho_{ij,g}^2\left(1-\rho_{ij,g}^2\right)^{-3/2}\right]\frac{\partial}{\partial c_{ij,g}}(\cdot)$$

B.3 Derivatives of Log-Prior Term

The goal of this section is to derive $\mathbf{H}^p$, the other term required in (B.4). Recall that the prior (3.10) is

$$p\big(\Theta^k|Y^{1:k-1},\mathcal{M}\big) = C\prod_{g=1}^{G}p\big(\mu_g^k|Y^{1:k-1},\mathcal{M}\big)\,,$$

where C is a constant consisting of the uniform distributions on the parameters Σ_g^k and π_g^k and

$$\begin{aligned} p\big(\mu_g^k|Y^{1:k-1},\mathcal{M}\big) &= \sum_{j=0}^{\hat{G}^{k-1}} \omega_j^k\, f_j\big(\mu_g^k|\psi_j^{k|k-1}\big) && \text{(B.19a)} \\ &= \frac{\omega_0^k}{V} + \sum_{j=1}^{\hat{G}^{k-1}} \omega_j^k\, f_{\mathcal{N}}\big(\mu_g^k|\hat{\mu}_j^{k|k-1}, S_j^k\big) && \text{(B.19b)} \\ &= \frac{\omega_0^k}{V} + \sum_{j=1}^{\hat{G}^{k-1}} \omega_j^k \Phi_j\big(\mu_g^k\big)\,, && \text{(B.19c)} \end{aligned}$$

where the last equality has used the notation

$$\Phi_j\big(\mu_g^k\big) \equiv f_{\mathcal{N}}\big(\mu_g^k|\hat{\mu}_j^{k|k-1}, S_j^k\big)\,.$$

Recall that

$$\mathbf{H}^p \equiv \nabla\nabla \log p\big(\Theta^k|Y^{1:k-1},\mathcal{M}\big) = \nabla\nabla \sum_{g=1}^{G} \log p\big(\mu_g^k|Y^{1:k-1},\mathcal{M}\big)\,. \tag{B.20}$$

$\mathbf{H}^p$ is block diagonal, as μ_g is the only parameter from Θ in the prior and all cross derivatives across the mixture components are zero:

$$\mathbf{H}^p = \begin{bmatrix} \mathbf{H}_1^p & \mathbf{0} & \cdots & \mathbf{0} & \vec{0} \\ \mathbf{0} & \mathbf{H}_2^p & \cdots & \mathbf{0} & \vec{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{H}_G^p & \vec{0} \\ \vec{0}^T & \vec{0}^T & \cdots & \vec{0}^T & 0 \end{bmatrix}, \tag{B.21}$$

where

$$\begin{aligned} \mathbf{H}_g^p &\equiv \nabla_{\tilde{\theta}_g^k}\nabla_{\tilde{\theta}_g^k} \sum_{g=1}^{G} \log p\big(\mu_g^k|Y^{1:k-1},\mathcal{M}\big) && \text{(B.22)} \\ &= \begin{bmatrix} 0 & \vec{0}^T & 0 & 0 \\ \vec{0} & \frac{\partial^2}{\partial\mu_g^k \partial{\mu_g^k}^T} \log p\big(\mu_g^k|Y^{1:k-1},\mathcal{M}\big) & \vec{0} & \vec{0} \\ 0 & \vec{0}^T & 0 & 0 \\ 0 & \vec{0}^T & 0 & 0 \end{bmatrix}, && \text{(B.23)} \end{aligned}$$

where $\theta_g^k \equiv [\pi_g^k \, {\mu_g^k}^T \, c_{11,g}^k \, c_{12,g}^k]^T$. Thus, to complete $\mathbf{H}^p$, it remains only to find the derivatives for

$$\frac{\partial^2}{\partial \mu_g^k \partial {\mu_g^k}^T} \log p\big(\mu_g^k | Y^{1:k-1}, \mathcal{M}\big) \, ,$$

which follow.

First Derivative

$$\begin{aligned}
\frac{\partial}{\partial \mu_g^k} \log p\big(\mu_g^k | Y^{1:k-1}, \mathcal{M}\big) &= \frac{\partial}{\partial \mu_g^k} \log \sum_{j=0}^{\hat{G}^{k-1}} \omega_j^k f_j(\mu_g^k) \\
&= \frac{1}{\sum_m \omega_m^k f_m(\mu_g^k)} \frac{\partial}{\partial \mu_g^k} \sum_j \omega_j^k f_j(\mu_g^k) \\
&= -\frac{1}{\sum_{m=0}^{\hat{G}^{k-1}} \omega_m^k f_m(\mu_g^k)} \sum_{j=1}^{\hat{G}^{k-1}} \omega_j^k \Phi_j(\mu_g^k) \left(S_j^k\right)^{-1} \left(\mu_g^k - \hat{\mu}_j^{k|k-1}\right)
\end{aligned}$$

Second Derivative

$$\begin{aligned}
&\frac{\partial^2}{\partial \mu_g^k \partial {\mu_g^k}^T} \log p\big(\mu_g^k | Y^{1:k-1}, \mathcal{M}\big) \\
&\quad = -\frac{\partial}{\partial \mu_g} \left[\frac{1}{\sum_{m=0}^{\hat{G}^{k-1}} \omega_m^k f_m(\mu_g^k)} \sum_{j=1}^{\hat{G}^{k-1}} \omega_j^k \Phi_j(\mu_g^k) \left(S_j^k\right)^{-1} \left(\mu_g^k - \hat{\mu}_j^{k|k-1}\right) \right]^T \\
&\quad = -\frac{1}{\sum_m \omega_m^k f_m(\mu_g^k)} \left[\frac{\partial}{\partial \mu_g^k} \sum_j \omega_j^k \Phi_j(\mu_g^k) \left(\mu_g^k - \hat{\mu}_j^{k|k-1}\right)^T \left(S_j^k\right)^{-1} \right] \\
&\qquad - \left[\frac{\partial}{\partial \mu_g^k} \frac{1}{\sum_m \omega_m^k f_m(\mu_g^k)} \right] \sum_j \omega_j^k \Phi_j(\mu_g^k) \left(\mu_g^k - \hat{\mu}_j^{k|k-1}\right)^T \left(S_j^k\right)^{-1} \\
&\quad = -\frac{1}{\sum_m \omega_m^k f_m(\mu_g^k)} \sum_j \omega_j^k \left[\frac{\partial \Phi_j(\mu_g^k)}{\partial \mu_g^k} \left(\mu_g^k - \hat{\mu}_j^{k|k-1}\right)^T + \Phi_j(\mu_g^k) \, \mathbf{I} \right] \left(S_j^k\right)^{-1} \\
&\qquad + \frac{1}{\left(\sum_m \omega_m^k f_m(\mu_g^k)\right)^2} \left[\sum_n \omega_n^k \frac{\partial \Phi_n(\mu_g^k)}{\partial \mu_g^k} \right] \left[\sum_j \omega_j^k \Phi_j(\mu_g^k) (\mu_g^k - \hat{\mu}_j^{k|k-1})^T \left(S_j^k\right)^{-1} \right] \\
&\quad = \frac{1}{\sum_{m=0}^{\hat{G}^{k-1}} \omega_m^k f_m(\mu_g^k)} \Bigg(\sum_{j=1}^{\hat{G}^{k-1}} \omega_j^k \Phi_j(\mu_g^k) \left[\left(S_j^k\right)^{-1} \left(\mu_g^k - \hat{\mu}_j^{k|k-1}\right) \left(\mu_g^k - \hat{\mu}_j^{k|k-1}\right)^T + \mathbf{I} \right] \left(S_j^k\right)^{-1} \\
&\qquad - \frac{1}{\sum_{m=0}^{\hat{G}^{k-1}} \omega_m^k f_m(\mu_g^k)} \left[\sum_{n=1}^{\hat{G}^{k-1}} \omega_n^k \Phi_n(\mu_g^k) \left(S_n^k\right)^{-1} \left(\mu_g^k - \hat{\mu}_n^{k|k-1}\right) \right] \\
&\qquad \cdot \left[\sum_{j=1}^{\hat{G}^{k-1}} \omega_j^k \Phi_j(\mu_g^k) (\mu_g^k - \hat{\mu}_j^{k|k-1})^T \left(S_j^k\right)^{-1} \right] \Bigg)
\end{aligned}$$

Bibliography

[1] R. A. Andersen, J. W. Burdick, S. Musallam, B. Pesaran, and J. G. Cham. Cognitive neural prosthetics. *Trends in Cognitive Sciences*, 8(11):486–493, November 2004.

[2] Dawn M. Taylor, Stephen I. Helms Tillery, and Andrew B. Schwartz. Direct cortical control of 3D neuroprosthetic devices. *Science*, 296(5574):1829–1832, June 2002.

[3] Mijail D. Serruya, Nicholas G. Hatsopoulos, Liam Paninski, Matthew R. Fellows, and John P. Donoghue. Brain-machine interface: Instant neural control of a movement signal. *Nature*, 416(6877):141–142, March 2002.

[4] Johan Wessberg, Christopher R. Stambaugh, Jerald D. Kralik, Pamela D. Beck, Mark Laubach, John K. Chapin, Jung Kim, S. James Biggs, Mandayam A. Srinivasan, and Miguel A. L. Nicolelis. Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature*, 408(6810):361–365, November 2000.

[5] Parag G. Patil and Dennis A. Turner. The development of brain-machine interface neuroprosthetic devices. *Neurotherapeutics*, 5(1):137–146, January 2008.

[6] Richard A Andersen, Sam Musallam, and Bijan Pesaran. Selecting the signals for a brain-machine interface. *Current Opinion in Neurobiology*, 14(6):720–726, December 2004.

[7] Jorge G. Cham, Edward A. Branchaud, Zoran Nenadic, Bradley Greger, Richard A. Andersen, and Joel W. Burdick. Semi-chronic motorized microdrive and control algorithm for autonomously isolating and maintaining optimal extracellular action potentials. *Journal of Neurophysiology*, 93:570–579, January 2005.

[8] Zoran Nenadic and Joel W. Burdick. A control algorithm for autonomous optimization of extracellular recordings. *IEEE Transactions on Biomedical Engineering*, 53(5):941–955, May 2006.

[9] Edward A. Branchaud. *An Algorithm for the Autonomous Isolation of Neurons in Extracellular Recordings.* PhD thesis, California Institute of Technology, June 2006.

[10] F. Wood, M. J. Black, C. Vargas-Irwin, M. Fellows, and J. P. Donoghue. On the variability of manual spike sorting. *IEEE Transactions on Biomedical Engineering*, 51(6):912–918, 2004.

[11] Michale S. Fee. Active stabilization of electrodes for intracellular recording in awake behaving animals. *Neuron*, 27(3):461–468, September 2000.

[12] 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*, 94(1):5–17, December 1999.

[13] R. Christopher deCharms, David T. Blake, and Michael M. Merzenich. A multielectrode implant device for the cerebral cortex. *Journal of Neuroscience Methods*, 93(1):27–35, October 1999.

[14] Michale S. Fee and Anthony Leonardo. Miniature motorized microdrive and commutator system for chronic neural recording in small animals. *Journal of Neuroscience Methods*, 112(2):83–94, December 2001.

[15] Jerald D. Kralik, Dragan F. Dimitrov, David J. Krupa, Donald B. Katz, Dana Cohen, and Miguel A. L. Nicolelis. Techniques for long-term multisite neuronal ensemble recordings in behaving animals. *Methods*, 25(2):121–150, October 2001.

[16] Suri Venkatachalam, Michale S. Fee, and David Kleinfeld. Ultra-miniature headstage with 6-channel drive and vacuum-assisted micro-wire implantation for chronic recording from the neocortex. *Journal of Neuroscience Methods*, 90(1):37–46, August 1999.

[17] Bart P. Vos, Mike Wijnants, Sofie Taeymans, and Erik De Schutter. Miniature carrier with six independently moveable electrodes for recording of multiple single-units in the cerebellar cortex of awake rats. *Journal of Neuroscience Methods*, 94(1):19–26, December 1999.

[18] 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*, 52(8):1470–1477, August 2005.

[19] Michale S. Fee, Partha P. Mitra, and David Kleinfeld. Automatic sorting of multiple unit neuronal signals in the presence of anisotropic and non-Gaussian variability. *Journal of Neuroscience Methods*, 69(2):175–188, November 1996.

[20] M. Salganicoff, M. Sarna, L. Sax, and G.L. Gerstein. Unsupervised waveform classification for multi-neuron recordings: a real-time, software-based system. I. Algorithms and implementation. *Journal of Neuroscience Methods*, 25(3):181–187, October 1988.

[21] Eyal Hulata, Ronen Segev, and Eshel Ben-Jacob. A method for spike sorting and detection based on wavelet packets and Shannon's mutual information. *Journal of Neuroscience Methods*, 117(1):1–12, May 2002.

[22] F. Ohberg, H. Johansson, M. Bergenheim, J. Pedersen, and M. Djupsjobacka. A neural network approach to real-time spike discrimination during simultaneous recording from several multi-unit nerve filaments. *Journal of Neuroscience Methods*, 64(2):181–187, February 1996.

[23] R. Quian Quiroga, Z. Nadasdy, and Y. Ben-Shaul. Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. *Neural Computation*, 16(8):1661–1687, 2004.

[24] Tetyana I. Aksenova, Olga K. Chibirova, Oleksandr A. Dryga, Igor V. Tetko, Alim-Louis Benabid, and Alessandro E. P. Villa. An unsupervised automatic method for sorting neuronal spike waveforms in awake and freely moving animals. *Methods*, 30(2):178–187, June 2003.

[25] Carlos Vargas-Irwin and John P. Donoghue. Automated spike sorting using density grid contour clustering and subtractive waveform decomposition. *Journal of Neuroscience Methods*, 164(1):1–18, 2007.

[26] Geoffrey McLachlan and David Peel. *Finite Mixture Models.* Wiley Interscience, 2000.

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

[28] Michael S. Lewicki. Bayesian modeling and classification of neural signals. *Neural Computation*, 6:1005–10030, 1994.

[29] Shy Shoham, Matthew R. Fellows, and Richard A. Normann. Robust, automatic spike sorting using mixtures of multivariate t-distributions. *Journal of Neuroscience Methods*, 127(2):111–122, August 2003.

[30] Kyung Hwan Kim and Sung June Kim. Method for unsupervised classification of multiunit neural signal recording under low signal-to-noise ratio. *IEEE Transactions on Biomedical Engineering*, 50(4):421–431, 2003.

[31] Frank Wood, M. Fellows, J.P. Donoghue, and M. J. Black. Automatic spike sorting for neural decoding. In *IEEE Int. Conf. Engineering in Medicine and Biology Society (EMBS)*, 2004.

[32] Aharon Bar-Hillel, Adam Spiro, and Eran Stark. Spike sorting: Bayesian clustering of non-stationary data. *Journal of Neuroscience Methods*, 157(2):303–316, October 2006.

[33] R. K. Snider and A. B. Bonds. Classification of non-stationary neural signals. *Journal of Neuroscience Methods*, 84(1-2):155–166, October 1998.

[34] A. A. Emondi, S. P. Rebrik, A. V. Kurgansky, and K. D. Miller. Tracking neurons recorded from tetrodes across time. *Journal of Neuroscience Methods*, 135:95–105, 2004.

[35] G. W. Pulford. Taxonomy of multiple target tracking methods. *IEE Proceedings — Radar, Sonar & Navigation*, 152(5):291–304, 2005.

[36] Yaakov Bar-Shalom and Thomas E. Fortmann. *Tracking and Data Association.* Academic Press, Inc., Orlando, FL, 1988.

[37] Samuel S. Blackman and Robert Popoli. *Design and Analysis of Modern Tracking Systems.* Artech House, Norwood, MA, 1999.

[38] Donald B. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, December 1979.

[39] S. S. Blackman. Multiple hypothesis tracking for multiple target tracking. *IEEE Aerospace and Electronic Systems Magazine*, 19(1):5–18, 2004.

[40] Ingemar J. Cox and Sunita L. Hingorani. An efficient implementaion of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):138–150, February 1996.

[41] R. Danchick and G. E. Newnam. Reformulating reid's mht method with generalised murty k-best ranked linear assignment algorithm. *IEE Proceedings — Radar, Sonar & Navigation*, 153(1):13–22, 2006.

[42] Yaakov Bar-Shalom and William Dale Blair, editors. *Multitarget-Multisensor Tracking: Applications and Advances*, volume 3. Artech House, Boston, MA, 2000.

[43] Peter Cheeseman and John Stutz. Bayesian classification (AutoClass): Theory and results. In U.M. Fayyard, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, chapter 6, pages 61–83. AAAI/MIT Press, Cambridge, MA, 1996.

[44] Eric R. Kandel, James H. Schwartz, and Thomas M. Jessell, editors. *Principles of Neural Science.* McGraw-Hill, New York, fourth edition, 2000.

[45] D. R. Kipke, R. J. Vetter, R. J. Vetter, J. C. Williams, J. C. Williams, and J. F. Hetke. Silicon-substrate intracortical microelectrode arrays for long-term recording of neuronal spike activity in cerebral cortex. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):151–155, June 2003.

[46] Gyorgy Buzsaki. Large-scale recording of neuronal ensembles. *Nature Neuroscience*, 7(5):446–451, May 2004.

[47] Charles M. Gray, Pedro E. Maldonado, Mathew Wilson, and Bruce McNaughton. Tetrodes markedly improve the reliability and yield of multiple single-unit isolation from multi-unit recordings in cat striate cortex. *Journal of Neuroscience Methods*, 63(1–2):43–54, December 1995.

[48] Roger Lemon. *Methods for Neuronal Recording in Conscious Animals*. John Wiley and Sons, 1984.

[49] Patrick J. Rousche and Richard A. Normann. Chronic recording capability of the Utah intracortical electrode array in cat sensory cortex. *Journal of Neuroscience Methods*, 82(1):1–15, July 1998.

[50] Justin C. Williams, Robert L. Rennaker, and Daryl R. Kipke. Long-term neural recording characteristics of wire microelectrode arrays implanted in cerebral cortex. *Brain Research Protocols*, 4(3):303–313, December 1999.

[51] I. Porada, I. Bondar, W. B. Spatz, and J. Kruger. Rabbit and monkey visual cortex: more than a year of recording with up to 64 microelectrodes. *Journal of Neuroscience Methods*, 95(1):13–28, January 2000.

[52] Cees J. J. Avezaat and John H. M. van Eijndhoven. The role of the pulsatile pressure variations in intracranial pressure monitoring. *Neurosurgical Review*, 9(1):113–120, 1986.

[53] J. N. Turner, W. Shain, D. H. Szarowski, M. Andersen, S. Martins, M. Isaacson, and H. Craighead. Cerebral astrocyte response to micromachined silicon implants. *Experimental Neurology*, 156(1):33–49, 1999.

[54] C. Pang, Y. C. Tai, J. W. Burdick, and R. A. Andersen. Electrolysis-based diaphragm actuators. *Nanotechnology*, 17(4):S64–S68, 2006.

[55] Zoran Nenadic and Joel W. Burdick. Spike detection using the continuous wavelet transform. *IEEE Transactions on Biomedical Engineering*, 52(1):74–87, January 2005.

[56] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley Interscience, second edition, 2000.

[57] Chris Fraley and Adrian E. Raftery. How many clusters? Which clustering method? Answers via model-based cluster analysis. *Computer Journal*, 41(8), 1998.

[58] Kenneth D. Harris, Hajime Hirase, Xavier Leinekugel, Darrell A. Henze, and Gyorgy Buzsaki. Temporal interaction between single spikes and complex spike bursts in hippocampal pyramidal cells. *Neuron*, 32(1):141–149, 2001.

[59] Alex Zviagintsev, Yevgeny Perelman, and Ran Ginosar. Algorithms and architectures for low power spike detection and alignment. *Journal of Neural Engineering*, 3(1):35–42, January 2006.

[60] Maneesh Sahani. *Latent Variable Models for Neural Data Analysis*. PhD thesis, California Institute of Technology, June 1999.

[61] Kenneth D. Harris, Darrell A. Henze, Jozsef Csicsvari, Hajime Hirase, and Gyorgy Buzsaki. Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. *Journal of Neurophysiology*, 84(1):401–414, July 2000.

[62] Christophe Pouzat, Ofer Mazor, and Gilles Laurent. Using noise signature to optimize spike-sorting and to assess neuronal classification quality. *Journal of Neuroscience Methods*, 122:43–57, 2002.

[63] M. D. Linderman, B. Gilja, G. Santhanam, A. Afshar, S. Ryu, T. H. Meng, and K. V. Shenoy. Neural recording stability of chronic electrode arrays in freely behaving primates. Program No. 13.7. In *Abstract Viewer / Itinerary Planer*. Society for Neuroscience, Atlanta, GA, 2006.

[64] Juan Carlos Letelier and Pamela P. Weber. Spike sorting based on discrete wavelet transform coefficients. *Journal of Neuroscience Methods*, 101(2):93–106, September 2000.

[65] Alexey Pavlov, Valeri A. Makarov, Ioulia Makarova, and Fivos Panetsos. Separation of extracellular spikes: When wavelet based methods outperform the principle component analysis. *Lecture Notes in Computer Science*, 3561:123–132, October 2005.

[66] 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*, 39(1):1–38, 1977.

[67] G. Celeux and G. Govaert. Gaussian pasimonious clustering models. *Pattern Recognition*, 28:781–793, 1995.

[68] James L. Beck and Ka-Veng Yuen. Model selection using response measurements: Bayesian probabilistic approach. *Journal of Engineering Mechanics*, 130(2):192–203, 2004.

[69] Lawrence D. Stone, Carl A. Barlow, and Thomas L. Corwin. *Bayesian Multiple Target Tracking*. Artech House, Boston, MA, 1999.

[70] T. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, 8(3):173–184, July 1983.

[71] Katta G Murty. An algorithm for ranking all the assignments in order of increasing cost. *Operations Research*, 16:682–687, 1968.

[72] R. Danchick and G. E. Newnam. A fast method for finding the exact N-best hypotheses for multitarget tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 29(2):550–560, April 1993.

[73] I. J. Cox, M. L. Miller, R. Danchick, and G. E. Newnam. A comparison of two algorithms for determining ranked assignments with application to multitarget tracking and motion correspondence. *IEEE Transactions on Aerospace and Electronic Systems*, 33(1):295–301, 1997.

[74] E Mazor. Interacting multiple model methods in target tracking: A survey. *IEEE Transactions on Aerospace and Electronic Systems*, 34(1):103–123, January 1998.

[75] B. Hammarberg, C. Forster, and E. Torebjork. Parameter estimation of human nerve c-fibers using matched filtering and multiple hypothesis tracking. *IEEE Transactions on Biomedical Engineering*, 49(4):329–336, 2002.

[76] Thomas Kurien. *Multitarget-Multisensor Tracking: Advanced Applications*, chapter "Issues in the design of practical multitarget tracking algorithms", pages 43–83. Artech House, Boston, MA, 1990.

[77] Norman Bleistein and Richard A. Handelsman. *Asymptotic Expansions of Integrals.* Dover, New York, 1986.

[78] J. L. Beck and L. S. Katafygiotis. Updating models and their uncertainties. I: Bayesian statistical framework. *Journal of Engineering Mechanics*, 124(4):455–461, April 1998.