On neural encoding: its estimation, application, and development

Benjamin James Lansdell

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2017

Reading Committee:
Adrienne Fairhall, Chair
Jose Nathan Kutz
Eric Shea-Brown
Chet Moritz

Program Authorized to Offer Degree:
Applied Mathematics

University of Washington

Abstract

On neural encoding: its estimation, application, and development

Benjamin James Lansdell

Chair of the Supervisory Committee:
Professor Adrienne Fairhall
Department of Physiology and Biophysics

The spiking activity of neurons encodes information about sensory stimuli and about planned or executed motor outputs. An important problem in computational neuroscience is the development of predictive models that describe the relationship between neural spiking and sensory inputs and motor outputs. This dissertation explores different aspects and applications of neural encoding.

First, a common family of models used to capture this relationship – known as linear non-linear Poisson (LNP) models – is described. Natural stimuli and motor outputs often exhibit strong auto-correlations, which presents challenges to model fitting. Ways of dealing with these issues are discussed, and an application is presented in primary motor cortical data from a non-human primate while performing a grip-and-reach task.

Determining the nature of encoding in primary motor cortex is a long-standing problem in studies of the motor system. For instance, it is unclear if primary motor neurons are best thought of as encoding information about movement kinetics or kinematics. These issues carry weight in the design of brain-computer interfaces (BCIs). A study into how primary motor cortical activity adapts to a coordination task which involves concurrent control of a BCI and motor output is presented. The task, known as a dual-control BCI, requires a monkey implanted with an intracortical electrode array in primary motor cortex control one axis of a cursor through brain control and the other axes through wrist control. During the dual control task, an effective connectivity analysis shows that the units directly controlling the cursor specifically dissociate

their from other units. Further, factors such as control unit tuning to wrist motion do not predict task performance. Only the control units variability is shown to be predictive of performance, which is useful when considering unit selection in such BCIs.

Second, this dissertation examines a phenomenon that occurs during the developing nervous system. Correlated spontaneous activity in the developing retina, known as retinal waves, is known to have a role in the receptive field development in higher visual areas. However the exact role of this activity in development is unknown – whether the form of the spontaneous activity actively drives the development specific features, or whether it merely plays a permissive role. This dissertation presents a reaction-diffusion model of retinal waves mediated through cholinergic signaling. This allows for a semi-analytic analysis to identify when the retinal medium is excitable and can support wave activity, which can be useful in determine appropriate pharmacological manipulations in order to better study retinal waves' role in development.

Finally, advances in optical imaging mean that it is possible now to record from comprehensive populations of neurons from an entire brain or organism. Such studies promise to reveal much about how neural activity encodes and processes information. Hydra, a cnidarian, is a promising, novel model organism for such studies. Their nervous system is structured as a diffuse nerve net, which permits whole-animal calcium imaging of their neural activity. In order to extract this activity and relate it to behavior the neurons must be registered and tracked throughout imaging. Deformable object tracking is a challenging computer vision problem. This dissertation presents two methods that may aid in this neuron tracking problem – one based on an extended Kalman filter accelerated through a GPU implementation, and the other based on a combination of multi-frame optic flow and image registration methods.

Acknowledgments

I would like to first thank my advisor. I have had the privilege of working with Adrienne Fairhall, and deeply appreciate her support, encouragement, keen scientific insight, and tireless work promoting and enabling computational neuroscience research at UW. Also, thank you to Nathan Kutz for introducing me to the field of computational neuroscience, for his patience and understanding, and for the opportunity to visit Harvard during his sabbatical year.

To the remainder of my committee for their time and feedback: thanks to Eric Shea-Brown, Emily Fox, and particularly to Chet Moritz for his support, encouragement, and making me feel at home in his lab.

The work presented here is the combined effort of many researchers. I would like to thank my collaborators: Kevin Ford, Rob Robinson, Charlie Matlack, Eb Fetz, Cooper Mellema, Rafael Yuste, Chris Dupre, John Szymanski, Shuting Han, Anatoly Buchin, Amnon Horowitz and Ryan McGee. And particularly Ivana Milovanovic, for the countless hours spent debugging MATLAB

code together, and for putting up with my efforts to learn SQL and incorporate it into our analysis – OK, maybe it *was* overkill...

Thanks to Pedro Maia, Eli Shlizerman and Julijana Gjorgjieva for discussion and feedback on the retinal waves work.

Thanks to the many people I have had the pleasure of engaging in scientific and philosophical discussion throughout my time in graduate school. Particularly to Alex Cayco Gajic, Alison Duffy and Braden Brinkman.

Thanks to the Fairhall lab members, and the AMATH staff and faculty for providing a supportive work environment. To Olga Trichtchenko, Mauricio Del Razo and Donsub Rim for their friendship. To the baristas at Voxx Coffee and Cafe Solstice – the appeal of Solstice's too loud, too crowded work environment continues to elude me, yet there I am.

To Elizabeth Embick.

Finally to my parents and brother for all their support.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

Acronyms / Abbreviations

$\hat{\theta}$	an estimator of θ
$\mathbb{E}_\theta$	expectation with respect to distribution possessing parameter θ (i.e. to measure $dF(x;\theta) = f(x;\theta)dx$)
$\mathbb{I}_A(\omega)$	indicator function of event A
$\mathbb{V}_\theta$	variance with respect to distribution possessing parameter θ
$\mathbf{x}$	a column vector
$\mathcal{I}$	Fisher information matrix

θ_* the true parameter value

$A, A_{:,n}, A_{n,:}$ a matrix, the nth column, and the nth row of the matrix

$F_X(x)$ df of random variable X

$f_X(x), p_X(x)$ pdf/pmf of random variable X

$L, \mathcal{L}$ likelihood and log-likelihood

X a random variable

x a particular observation/outcome of the random variable X

1

Overview

"*By reasoning, I understand computation. And to compute is to collect the sum of many things added together at the same time, or to know the remainder when one thing has been taken from another.*"

—Thomas Hobbes, *De Corpore*, 1655

1.1 The computing brain

The Greek word for spirit, *pneuma*, is part of a long history of conceptualizing living and thinking organisms through contemporary technological metaphors. Pneuma also means breath, or wind, and reflects a relationship between the water-based machines of ancient Greece, fountains

and water clocks, and conceptions of the soul. Other technologies beyond antiquity also serve as examples: clockwork mechanisms during the enlightenment, steam engines and hydraulic machines during the Victorian era, relay switches and telegraphs from early last century, and the digital computer of the last 70 years. Each technology has been used as a metaphor to understand the brain's function and its relation to cognition [63]. This thesis will explore various models within the field of *computational neuroscience* that rely, in part, on this most recent, computational metaphor. Before describing the aims of this thesis a brief overview of this discipline is provided.

Computational neuroscience is the study of the function of nervous systems and brains, both through using computers as tools to simulate that which cannot be modeled otherwise and through using metaphors of computation [59, 207]. What exactly does it mean for something to be a computer? This can be debated, but a common definition is the following. A physical system implements a given computation if there are some elements of that system that map usefully to some variables of interest[59]. This mapping and the evolution of the system then results in the manipulation of the variables of interest in such a way that some calculation/computation is performed. Thus, in general, we have some relations between physical states and variables x and z such that the evolution of the system honors the relation

$$z = a(x). \tag{1.1}$$

Such an arrangement could be said to be computing the function $a(x)$. Note that this definition does not stipulate that the physical system implementing the computation possess any digital behavior: a sun dial is a good example of an analog computer. Nor does a computer have to be human-made: suitably interpreted, many natural systems can be seen to be performing computations. A ball rolling and settling at the bottom of a hill is performing an 'energy minimization' computation. But such natural computations are by no means trivial – the process of evolution through natural selection can be seen as a type of 'fitness' optimizing computation, and one that computer scientists have found useful to mimic in solving general optimization problems.

Given this definition, neuroscientist David Marr proposed his well known *tri-level analysis* as a framework to analyze the computational capabilities of neural *circuits*. Marr proposed that a given neural circuit can be analyzed at three different levels: we can first ask what *computation* is being performed, second we can ask what is the *algorithm* that is being used to perform the computation, and third we can ask what is the physical *implementation* of this algorithm[176]. Marr's framework inspired many founders of the field, including Hopfield, Sejnowski, and Poggio from the 1980s onward.

1.2 Representations in neural activity

The definition of computation given above details two necessary components: 1) a physical system capable of generating sufficiently rich dynamics, and 2) the existence of a mapping between the system's states and some variables of interest. The existence of such a mapping is necessary for claiming that a system *represents* a variable of interest, and it is the manipulation of these representations that are the basis of computation. An oft-quoted tag-line for philosopher Jerry Fodor's book *The Language of Thought* is "No computation without representation."[89] When it comes to the brain, in general we would like to know: what elements of an organism's environment, internal processes, and motor responses are represented by the nervous system? And how are these elements represented? In relation to Equation (1.1), we could phrase these questions as: what does variable x represent, and what is the relation between x and physical states of the nervous system. An example might be the velocity and position of the eyes, and its representation in the spiking rate of neurons in the oculomotor system.

Does any relationship, or mapping, between some variable and states of the nervous system count as representational? In nervous systems many different components can be found to reliably correlate with external or physiologically relevant internal variables. This is true even within individual neurons: membrane potential, synaptic strength, intracellular calcium concentration, and processes involving G-coupled protein receptors correlate with physiologically relevant

variables, and can all be studied for their role in single cell computations. A compendium of such single cell computations is provided in Koch 1998 [145]. But part of the definition of computation given above is the phrase 'variables of interest', which highlights that what counts as computational has an interest-relative component. We only view something as performing a computation if we are interested in some of its outputs or the variables involved in the computation. A ball rolling to the bottom to the hill is only an interesting computation if the landscape, and where its minimum is, can thought of as representing some variable of interest, otherwise it's just a ball on a hill. This means that the existence of a correlation is not sufficient for calling something a representation. More specific and detailed theories of representation can be provided, and debated. The review by Piccinini and Shagrir 2014 [207], and references therein, provides a good starting point for discussion on these issues (and an example of a detailed, 'bottom-up' theory of representation can be found in Lloyd 1989 [164]).

Regardless of these more philosophical issues, it remains the case that a necessary step in understanding a given neuronal computation relies on characterizing the relationship between a nervous system's state and candidate variables of interest. In this thesis focus will be given to the relationships between spiking activity of neurons and sensory inputs and motor outputs. This is known as the neural coding, or encoding, question: how does neural activity *encode* information about the environment, motor output and internal variables? Slightly more specifically, the goal is to describe the relationship between a candidate variable $x(t)$ and neural output $y(t)$, typically given as a set of spike times for a neuron:

$$y(t) \sim f[x(t)]. \tag{1.2}$$

Determining this relationship is complicated by the fact that neurons, when presented with the same input $x(t)$, will produce varying outputs – due to many factors, neurons are noisy. For this reason the relationship between $x(t)$ and $y(t)$ is statistical in nature.

Answering these neural encoding questions is arguably more straightforward in sensory systems, in part because encoding of environmental variables – e.g. visual stimuli, variables more under an experimentalist's control – can be better probed experimentally. Indeed, the *receptive field* of neurons in the visual system continues to provide the archetype for this type of relationship [177, 81] . Hubel and Weisel, by presenting moving gratings in particular directions and visual field locations over repeated trials, were able to determine a relationship between grating location, direction and activity of visual cortical neurons of the cat [129]. Since their foundational work, significant effort has been devoted to the identification of encoding relationships of individual neurons (a detailed overview, of which this thesis provides a contribution, can be found in Aljadeff et al 2016[7]). Non-visual senses can similarly be studied and encoding relationships identified (e.g. [235, 172]).

1.3 Changing models of encoding in contemporary neuroscience

Yet the preceding description leaves open the question of how it is we are to identify representations in the nervous system. There is no general recipe, of course; it requires looking at the specifics of the system under consideration. Knowledge of the types of computations the system may be involved in is required to generate hypotheses. And therein is the challenge.

Comparisons between encoding models in the visual system and the motor system serve to illustrate this challenge. From studies of the visual system, an influential idea first articulated by Barlow is that of the feature detector. In studies of frog vision, Barlow was the first to note the center-surround nature of receptive fields of the frog's retinal ganglion cells – they respond strongly to small spots on a light background. Frogs eat flies, and the hypothesis that Barlow put forward is that retinal ganglion cells act as fly detectors. Thus by proposing a computational role in feature detection, the activity of these neurons could be said to *represent* the presence or absence of a fly in a particular part of the animal's visual field. The idea of course can be generalized to

generic objects and features in a visual scene. Barlow further proposed that individual neurons therefore coded for perceptually significant events. [177]

The feature detector is a conceptually straightforward and appealing computation, and also one that is compatible with single-unit recordings available at the time of its conception. Thus while the notion that individual neurons' activity relates so to directly to perception has been tempered, the idea that neurons represent different components of a sensory input through their spiking activity has remained. Indeed hierarchical models of visual processing based on neurons representing increasingly abstract and invariant visual features have been successful in describing many aspects of the visual system [234]. The general notion has been fruitful in teasing apart the function of the visual system, and in testing more specific hypotheses about the nature of this encoding – sparse coding [196], predictive coding [214], etc.

In contrast, encoding models in primary motor cortex are more ambiguous: it remains unclear what motor output parameters cortical neurons can best be said to represent. In a sense, motor cortex was one of the first areas of cortex to be studied[1], and yet debate continues today about its function. In the 1960s Evarts, performing some of the first electrophysiological recordings in behaving primates, noted a relationship between single joint forces applied and neural activity. This lead to the notion that motor cortex represents movement *kinetics* [80, 231]. However early studies focused typically on single joint movements in conjunction with single-unit recordings[226]. In 1982 Georgopoulos [99] constructed a now famous center-out task, which involved reaching from a central position outward to one of eight targets. Compared to previous studies, the task involved multiple joint motion. Georgopoulos noted that individual neurons responded in a way related to the direction of the reach motion. Each neuron exhibiting cosine tuning: its firing rate was a cosine function of the reach direction. Such a model provides each neuron with a 'preferred reach direction', the direction of maximal firing rate. The findings lead to the notion that motor cortical activity also represents limb movement *kinematics* – limb positions, velocities and accelerations.

[1] Prior to the mid 1800s, it was thought the cortex was largely functionally irrelevant. In what now appears rather barbaric, Fritsch and Hitzig, in 1870, restrained live dogs and removed their skulls. This enabled them to stimulate regions of the dogs' cortex. When specific regions were stimulated they observed movement of specific parts of the dogs' contralateral limbs. In this way they were the first to identify the motor cortex.

Georgopoulos also noted that, while the activity of any one unit could not be used to reliably determine motion direction, the sum of a population of unit's preferred directions, weighted by their firing rate, could be used to predict motion direction. This is known as the population vector, and is a type of neural *decoding*. It provides a useful starting point in the design of brain-computer interfaces.

Further complication is provided by studies from Graziano [113, 112], in which electrical stimulation is applied to sites in motor cortex at higher currents and for longer durations than previously performed. Graziano observed that stimulation of the same site for an extended duration lead to drawn-out, multi-joint reaching and grasping movements. Thus motor cortex may additionally represent ethologically relevant actions. However later studies suggest that this effect may be the result of intra-cortical stimulation blocking and replacing natural neural activity [52].

Each of the above experimental paradigms provides evidence that motor cortical activity encodes different features of motor behavior. Indeed motor cortex does encode each of these, under different circumstances, as the information about each different aspect of movement can be significantly correlated to neural activity. Yet interpreting these relationships in a single computational framework is challenging.

Thinking of motor cortical activity as representing movement in a similar way to how visual cortex may represent visual information would suggest that primary motor neurons represent different elements of motor control of different limbs, and that the mixture of neurons active at any one time combine in some way to produce an action related to the actively represented motions – e.g. the population vector. Yet this cannot account for why it is that different encoding relationships become apparent under different experimental paradigms – the population vector does not work all of the time (see, for example, the analysis of Sanger 1994 [222]).

Different computational theories may thus be required. Some candidate theories involve viewing the motor system as implementing an optimal control system [275, 229, 226]. A recent idea, enabled by multi-electrode array recordings and optical imaging, proposes that motor cortex

implements a type of dynamical system related to the dynamics of reaches and motions performed. A specific instantiation of this idea, tested in the context of a delayed-reach task – a target is presented and after a delay period is reached for, possibly while avoiding obstacles – is that a planning phase configures motor cortical activity to be in some initial state, and that during the reach this state acts as initial conditions for the dynamical system, which then unfolds according to its own dynamics [58, 57].

This description of encoding models in motor cortex serves to illustrate two points: first, that understanding how neural activity encodes information about particular external variables is not sufficient to understand its role in neural computation; and second, that the types of theories that can be considered is often constrained by the experimental paradigms, recording technologies and datatypes available to us. Dynamical system theories of motor cortical activity were not considered until the advent of multi-electrode arrays, nor were the arrays originally constructed to test such theories.

Technology now allows for the simultaneous recording of hundreds or even thousands of neurons, in some cases in conjunction with non-trivial behaviors. This greatly expands the class of neural computational theories that can be tested, and thus in the future will serve to better understand how sensory inputs and motor outputs are represented in the nervous system. An important component of this is understanding how neurons encode these inputs and outputs, and how these relationships change as technologies advance.

1.4 Thesis aims and contributions

This thesis aims to explore the notion of neural encoding in three ways: its estimation, application, and development. First, in **Chapter 2**, a well-known class of statistical models commonly applied to encoding problems, the linear non-linear Poisson (LNP) model, is reviewed. Fitting these methods to naturalistic data requires some care, as their estimation often can be poorly conditioned. Thus particular attention is given to these issues.

However, directly relating neural activity to external parameters may be less informative in some cases. In some cases internally generated neural dynamics may play a significant role, and need to be accounted for in statistical descriptions of neural activity. The advent of high dimensional multi-unit recordings permits estimation of these dynamics. Thus the second part of the thesis describes augmentations to these LNP-type models to multi-unit recordings. In **Chapter 3** we review models that allow estimation of interactions between recorded units, referred to as either *functional* or *effective connectivity*. As described above, issues of encoding are particularly ambiguous when it comes to motor cortical activity, and multi-unit recordings have the potential to provide insight in ways not possible from single-unit recordings. **Chapter 4** provides a demonstration of an extension of our encoding models to network models in primary motor cortical recordings from a monkey while performing a 3D grip and reach task. Finally, in **Chapter 5**, an application of these connectivity and encoding models is provided in order to better understand a type of concurrent use brain computer interface (BCI), termed a dual-control BCI.

Third, the thesis contributes work related broadly to the development of neural encoding – that is, how it is that nervous systems develop to encode information about the environment. Normative models of neural circuits propose a principle that nervous systems should operate by, and from this principal derive properties that can be compared against actual neural recordings. Such analysis can confirm or reject principles nervous systems operate under, and thus provide insight into *why* it has the structure is does. Additional insight into why a circuit has the structure it does can be obtained by studying instead the developmental processes which lead to its ultimate form. Receptive fields in primary visual cortex can be studied in this way. During development, spontaneous waves of depolarizations in retinal ganglion cells are known to contribute to the formation of receptive fields of primary visual cortex. In **Chapter 6**, we develop a model of this spontaneous activity, known as *retinal waves*. The model is based on reaction-diffusion equations which can be analyzed in a semi-analytic fashion, which allows for the identification of criteria for when the retinal cell network supports wave activity. In principle this allows for predictions

about which pharmacological manipulations of the retinal network will alter/abolish retinal waves, which can be used to probe the waves' effect on the development of receptive fields.

Finally, this thesis considers a computer vision problem of importance to calcium image analysis in freely behaving animals. Focus is given to tracking the cniderian, Hydra, as it undergoes semi-constrained, approximately two-dimensional motion. However, in general, advancing technologies in neuroscience are rapidly allowing 'whole-animal' imaging studies to be performed in a variety of small animals, in conjunction with behavior. An important algorithmic prerequisite to such studies is being able to track their behavior, to facilitate more robust neuron tracking, and to perform behavioral analysis. To that end, we develop methods that can track deformable objects over extending video sequences. **Chapter 7** provides an overview of methods from the deformable object tracking literature. **Chapter 8** provides a high dimensional state space model made computationally feasible through an alternate formulation of the extended Kalman filter suitable to implement on a GPU. The model is shown to track deformable objects that do not undergo rapid motion or extreme deformations. In order to handle these cases a robust, long-term tracking method is developed based on multi-frame optic flow and image segmentation methods. This method is presented in **Chapter 9**.

Chapter 10 provides a summary and outlook of the presented work.

1.5 Published work

The proceeding work has formed the basis for contributions to the following proceedings and publications:

1. B Lansdell, K Ford, JN Kutz (2014). "A reaction-diffusion model of cholinergic retinal waves." *PLoS Comput Biol* 10 (12), e1003953.

2. R Pang, B Lansdell, A Fairhall (2016). "Dimensionality reduction in neuroscience." *Current Biology* 26 (14), R656–R660.

3. J Aljadeff, B Lansdell, A Fairhall, D Kleinfeld (2016). "Analysis of neuronal spike trains, deconstructed." *Neuron* 91 (2), 221-259.

4. B Lansdell, I Milovanovic, A Fairhall, E Fetz, C Moritz (2016). "Neural activity in a simultaneous BCI and manual task." *Proceedings of the 6th International Brain-Computer Interface Meeting*. Monterey CA, USA.

5. B Lansdell, I Milovanovic, C Mellema, EE Fetz, AL Fairhall, CT Moritz (2017). "Reconfiguring motor circuits for a joint manual and BCI task." (*submitted for publication* – arXiv preprint arXiv:1702.07368)

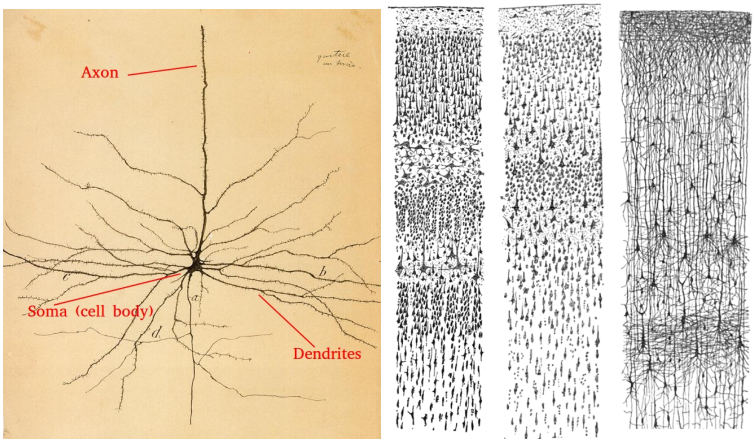
2

Neural encoding models

"*Then do you now model the form of a multitudinous, many-headed monster, having a ring of heads of all manner of beasts, tame and wild, which he is able to generate and metamorphose at will?*"

Plato, *The Republic*

This Chapter provides an overview of an important statistical modeling paradigm in computational neuroscience – one that centers around the linear non-linear Poisson (LNP) model, its extensions, and its applications. Particular focus is given to issues related to its estimation. While this or similar frameworks have arguably found most success in sensory neuroscience, it has proved a useful tool in modeling neurons of many brain regions.



(a) (b)

Figure 2.1: (a) Sketch of a single neuron. The morphology of neurons varies greatly, but their structure can usually be roughly described as being tree-like. Many incoming root-like elements, *dendrites*, connect to the cell body, *soma*, from which extends a longer, thicker, and generally less branched trunk-like structure, *the axon*. (b) Within the brain single neurons are embedded in a densely packed networks of neurons, in which the axons of neurons interface with the dendrites of other neurons, through *synapses*. These drawings are by Ramon e Cajal.

2.1 Introduction

2.1.1 A (very) brief overview of what neurons do

Neurons are the cells of animals that coordinate motion and share sensory information rapidly over large distances. They do so (in general) via electrical impulses called action potentials, *spikes* in membrane potential, that propagate along an extension of the cell body called an *axon*. These impulses travel (in general) without significant loss of amplitude or shape down the axon. The axon contains contacts, *synapses*, with neighboring cells' *dendrites*. Dendrites are influenced by incoming action potentials, which then affect that cell's probability of sending its own spikes down its own axon to cells that it synapses with (Figure 2.1). Many properties of the synapses, dendrites, the cells themselves, and the environment they're embedded in, determine how this probability of spiking is affected as the result of input – the action potentials that a neuron outputs are some *processed* form of the action potentials it receives as input. As discussed in Chapter 1, neurons, individually and collectively, are thus said to perform *computations* on their inputs that

ultimately dictate the organism's decisions and behavior. Because of this all-or-none nature, it is often sufficient to describe the *information* that one neuron sends to another entirely in terms of the times at which the spikes were generated or received by the sending or receiving neuron. With that in mind, we can consider how this may be modeled mathematically.

2.1.2 A class of models of neurons

Let a neuron's activity then be characterized as a set of spike times:

$$y(t) = \sum_{i=1}^{Y(T)} \delta(t - t_i),$$

where $Y(T)$ gives the total number of spikes up to a time T and $\delta(\cdot)$ is the Dirac delta function; thus $y(\cdot)$ is a point process and $Y(\cdot)$ is its counting process. In sensory neuroscience a key question to ask is: how does the sequence of spikes $y(t)$ depend on a sensory input $x(t)$?

The most general formulation is simply:

$$y(t) \sim f[x(t)], \tag{2.1}$$

for some functional $f[\cdot]$, and some stationary input $x(t)$[1]. This is a (very) broad class of models, so to make the problem tractable and interpretable the functional $f[\cdot]$ is approximated or parameterized in some way. For instance, we can assume that a neuron's response depends on a set of linear functionals designating 'features' $k(\tau)$ within the stimulus that the neuron preferentially responds to. This restricts the space of functionals to those of a particular form:

$$y(t) \sim f(k * x(t)), \tag{2.2}$$

[1]Strictly stationary: all finite-dimensional distributions are time-translation invariant – $f_k(t_1, \ldots, t_k) = f(t_1 + \tau, \ldots, t_k + \tau)$ for all $\tau \in \mathbb{R}$ and $k \in \mathbb{Z}$. Second-order, weakly stationary: the first and second dimensional joint distributions are time-translation invariant. In this text, and often elsewhere, stationary will be taken to mean second-order stationary. Wide-sense stationary: a process for which the first and second moments are time translation invariant. Wide-sense stationarity is sufficient to ensure that the space $\{x(t)\}$ is a Hilbert space with inner product $\langle X_t, Y_t \rangle = \mathbb{E}(X_t Y_t)$.[147]

where $*$ denotes the convolution: $a * b = \int a(\tau)b(t-\tau)\,d\tau$. In this case we are projecting the stimulus onto a subspace spanned by $k(\tau)$, by which the output $f[\cdot]$ becomes simply a function of this projection, and the model identification problem becomes one of determining functions (not functionals) $f(\cdot)$ and $k(\cdot)$ – a dramatic reduction in model complexity! We then think of the convolution $k * x(t)$ acting as a 'feature detector', as an operation which pinpoints times t at which the stimulus contains the feature $k(\tau)$.

This Chapter will explore models of the form (2.2), those which characterize a neuron's activity as a non-linear function of a linear operation on the stimulus input. They are in part inspired by Weiner/Volterra models, which expand the functional $f[\cdot]$ into its Wiener series and whose coefficients can then be estimated directly from the observed cross-correlations[2]. This basic 'linear, non-linear' idea has been extensively applied in many settings within neuroscience; the text Spikes: Exploring the Neural Code provides a clear and detailed exposition [217, 54].

A central problem in this setting is of course the estimation of both $f(\cdot)$ and $k(\cdot)$. In this section we review various implementations of this framework and how the estimation of the linear filters $k(\tau)$ is performed in each case. As it is an important statistical issue, we focus on how

[2]The theory of non-linear systems identification studies exactly equations of the form (2.1), which can be expressed in terms of a Volterra series (or an equivalent Weiner series). The Volterra series takes the form

$$y(t) = h^{(0)} + \int_{\mathbb{R}} h^{(1)}(\tau_1)x(t-\tau_1)\,d\tau_1 + \int_{\mathbb{R}^2} h^{(2)}(\tau_1,\tau_2)x(t-\tau_1)x(t-\tau_2)\,d\tau_1 d\tau_2 + \dots$$

for Volterra kernels $h^{(i)}$. We can also write

$$y(t) = H_0x(t) + H_1x(t) + H_2x(t) + \dots$$

for Volterra operators H_i. The Volterra operators are not orthogonal, in the sense that $\langle H_ix(t), H_jx(t)\rangle = \int_{\mathbb{R}} H_ix(t)H_jx(t)\,dt \neq 0$ for $i \neq j$, which makes estimation of the kernels in terms of observed (co)variances more difficult. Each term can be orthogonalized in a process analogous to the Gram-Schmidt orthogonalization procedure, and the result is the Wiener series:

$$y(t) = G_0x(t) + G_1x(t) + G_2x(t) + \dots$$

where Wiener operators are defined in a way that $\langle G_ix(t), G_jx(t)\rangle = \int_{\mathbb{R}} G_ix(t)G_jx(t)\,dt = 0$ for $i \neq j$. This has the benefit that the kernels can estimated directly from the observed cross-correlations:

$$k^{(n)}(\tau_1,\dots,\tau_n) = \frac{1}{n!A^n}y(t)x(t-\tau_1)\bar{}\dots x(t-\tau_n).$$

this estimation procedure is affected by the stimulus statistics, in particular on the effect of inputs possessing strong auto-correlation on the time-scale on which the neuron's activity is modeled.

2.1.3 Moving from continuous to discrete time frameworks

While we think of spikes as point events, suggesting they be indexed by a continuous time variable, in practice we will perform computations with a countable set of time bins. Let the following allow us to convert between a continuous set of spike times $\{t_j\}_{j=1}^{J}$ and a discrete set of binned spikes: assume there are K timebins such that $T = K\Delta t$. Denote by Y_k the number of spikes that occur up to time t_k for $t_k = k\Delta t$ and by y_k the number of spikes that occur in the interval $[t_{k-1}, t_k)$. Let $\mathbf{Y}_{1:k}$ denote the k-vector whose ith element records the number of spikes up to time t_i, and let $\mathbf{y}_{1:k}$ denote the k-vector in which the ith element represents the number of spikes that occurred in timebin i. When the length of these vectors does not need to be specified $\mathbf{Y}$ and $\mathbf{y}$ will be used. It's common but not always necessary to make Δt small enough that only one spike can be attributed to a neuron within the interval Δt, approximately 1-2ms, in which case the output vector becomes simply a binary sequence indicating the occurrence of a spike or not.

2.1.4 Spike trains as stochastic processes

It's useful to express the problem in terms of a stochastic process for which significant theory and estimation machinery exists. The Poisson process is the foundational point process, in the sense that the likelihood function of a very broad class of point processes has the same form as the humble Poisson process's likelihood. Here we describe the relation between a continuous point process and its discrete equivalent.

2.1.4.1 Continuous-time Poisson process

A homogeneous Poisson process can be defined by its measure of a set of J intervals $(a_j, b_j]$ with $a_j < b_j \leq a_{j+1}$:

$$\Pr(Y(a_j, b_j] = n_j) = \prod_{j=1}^{J} \frac{[\lambda(b_j - a_j)]^{n_j}}{n_j!} e^{-\lambda(b_j - a_j)}. \tag{2.3}$$

Given this form, the likelihood of a specif instantiation of events, given by times $\{t_j\}_{j=1}^{J}$, can be obtained by considering the set of intervals $S = \{(t_j - \Delta, t_j]\}_{j=1}^{J}$ on which, if Δ is made small enough, each contain one event, while the complementary set of intervals, denoted S^C, each contain no events. This gives

$$\begin{aligned}\Pr\left(\{Y(t_j - \Delta, t_j] = 1\}, \{Y(t_j, t_{j+1} - \Delta] = 0\}\right) &= \left(\prod_{j=1}^{J} \frac{[\lambda\Delta]^1}{1!} e^{-\lambda\Delta J}\right) e^{-\lambda T - \lambda\Delta J} \\ &= [\lambda\Delta]^J e^{-\lambda T}\end{aligned}$$

which, if we let $\Delta \to 0$ and divide by Δ^j provides, the density:

$$\begin{aligned}L_{(0,T]}(J; t_1, \ldots, t_J) =& \lambda^J e^{-\lambda T} \\ =& \exp\left\{-\lambda T + J \log \lambda\right\}.\end{aligned} \tag{2.4}$$

More generally, an inhomogeneous Poisson process is one defined by a time varying rate $\lambda(t)$, in which case we define

$$\Lambda(a, b) = \int_a^b \lambda(t')dt.$$

The probability of a set of events on disjoint intervals (2.3) then becomes

$$\Pr(Y(a_j, b_j] = n_j) = \prod_{j=1}^{J} \frac{\left[\int_{a_j}^{b_j} \lambda(t)\, dt\right]^{n_j}}{n_j!} e^{-\int_{a_j}^{b_j} \lambda(t)\, dt} \tag{2.5}$$

and the likelihood (2.4) becomes

$$\begin{aligned} L_{(0,T]}(J; t_1, \ldots, t_J) =& e^{-\int_0^T \lambda(t)\, dt} \prod_{j=1}^{J} \lambda(t_j) \\ &= \exp\left\{ -\int_0^T \lambda(t)\, dt + \int_0^T \log \lambda(t)\, dN(t) \right\} \\ &= \exp\left\{ -\int_0^T \lambda(t)\, dt + \sum_{j=1}^{J} \log \lambda(t_j) \right\}, \end{aligned} \tag{2.6}$$

where we have simply replaced in each expression $\lambda(b_j - a_j)$ with $\int_{a_j}^{b_j} \lambda(t)\, dt$ – the process maintains the property that the joint distributions are still Poisson, and disjoint intervals are independent. This substitution highlights that in moving to an inhomogeneous process we have done nothing more than rescale time according to $t \to u(t) \equiv \Lambda(0, t]$, which means that if we undo this transformation, by defining

$$\tilde{Y}(t) = Y(u^{-1}(t)),$$

then we obtain a homogeneous Poisson process with unit rate. Remarkably, this rescaling is in fact possible for a broad range of point processes, highlighting the fact that the Poisson process is in some sense the canonical point process (see Papangelouo 1972 [200] for a precise statement).

Indeed, quite generally, a regular stochastic point process is fully characterized by its rate, or conditional intensity function (CIF), λ, which can be defined as

$$\lambda(t|H(t)) = \lim_{\Delta t \to 0} \frac{\Pr(Y(t + \Delta t) - Y(t) \geq 1 | H(t))}{\Delta t}$$

and likelihood (2.6), where $H(t)$ represents the history of the process up to time t. See Daley and Vere-Jones 2003 [62] (p.23 and p.233) for more details.

2.1.4.2 Discrete-time Poisson processes

If Equation (2.5) is applied to a set of regular intervals $(t_{j-1}, t_j]$, we obtain

$$\Pr(\mathbf{Y}_{1:N}) = \Pr(Y(t_{k-1}, t_k] = y_k) = \prod_{k=1}^{N} e^{-\Lambda(t_{k-1},t_k]} \frac{[\Lambda(t_{k-1}, t_k)]^{y_k}}{y_k!} \tag{2.7}$$

We make two observations that are useful:

1. If we let $\lambda_k = \Lambda(t_{k-1}, t_k]$ then above probability becomes

$$\Pr(\mathbf{Y}_{1:N}) = \prod_{k=1}^{N} e^{-\lambda_k} \frac{\lambda_k^{y_k}}{y_k!} \tag{2.8}$$

   and we can instead consider a discrete-time version of the Poisson process, in which we need only specify λ_k for each timebin k, instead of having to specify the continuous rate $\lambda(t)$. If we only care about what happens on timescales at or above the duration Δt then this description may be entirely sufficient.

2. Over a short enough interval Δt, the probability of more than one spike occurring is an order smaller than the probability of one spike occurring within Δt, in which case the Poisson process (2.8) is approximately equivalent to a Bernoulli process. This can be seen by noting that for small enough Δt the rate is approximately

$$\lambda_k = \Lambda(t_{k-1}, t_k] = \lambda(t_k)\Delta t + O(\Delta t^2)$$

   and therefore

$$Y_k \sim \mathrm{Pn}(\lambda(t_k)\Delta t).$$

This means that

$$\begin{aligned}\Pr(Y_k = 0) =& e^{-\lambda\Delta t} = 1 - \lambda\Delta t + O(\Delta t^2)\\ \Pr(Y_k = 1) =& \lambda\Delta t e^{-\lambda\Delta t} = \lambda\Delta t + O(\Delta t^2)\\ \Pr(Y_k \geq 2) =& O(\Delta t^2),\end{aligned}$$

and thus to first order in Δt we can describe the likelihood of a particular set of events $\mathbf{y}$ by a Bernoulli process in which at each timebin an event either occurs or does not with probability $\lambda\Delta t$:

$$\Pr(y_{1:N} = \mathbf{y}) = \prod_{k=1}^{N} [\lambda(t_k|H_k)\Delta t]^{y_k}[1 - \lambda(t_k|H_k)\Delta t]^{1-y_k} + o(\Delta t^J).$$

Refer to Daley and Vere-Jones 2003 [62] for more details (see also Berman and Turner 1992 [29]); conversely, the continuous likelihood (2.6) is derived from the discrete likelihood (2.8) in Truccolo et al. 2005 [265].

2.1.4.3 Discrete-time linear non-linear Poisson (LNP) processes

Returning to models of spike trains, in later sections we'll formulate the generic model (2.2) as a probabilistic model of the form

$$\mathbf{y}_{1:k} \sim \mathrm{Pn}(\lambda_{1:k}(x)), \quad \text{with } \lambda_k = f\left(\sum_{i=0} k_i x_{k-i}\right), \tag{2.9}$$

where, as above, if Δt is taken to be small enough we can interpret this as

$$\lambda_k = \Pr(y_k = 1).$$

In words, the model applies a linear filter to the input stimulus, stipulates the mean firing rate of the neuron is some non-linear function of this filtered stimulus, and that the neuron fires

according to an inhomogeneous Poisson process according to this rate. This linear, non-linear Poisson cascade model is accordingly abbreviated an LNP model.

2.2 Modeling approaches in well-posed settings

When assuming a general non-linearity $f(\cdot)$, the characterization of $k(\cdot)$ is often tackled with two approaches: moment-based methods utilizing either the first or second moment of the spike-triggered stimulus ensemble, or likelihood-based methods involving specification of the distribution of $\mathbf{y}$ given stimulus $\mathbf{x}$. We describe these approaches here, before discussing issues related to temporally correlated inputs.

2.2.1 Spherical noise and the spike triggered average

When the input process is elliptically symmetric, a consistent estimator can be obtained simply through computation of the spike-triggered average (STA): [131, 40]

$$\hat{\mathbf{k}}_{STA} = \frac{1}{J}\sum_{j=1}^{J}\mathbf{x}_{t_j} = \frac{1}{J}\sum_{n=1}^{N} y_n \mathbf{x}_n,$$

where $\mathbf{x}_{t_j} \in \mathbb{R}^P$ is the stimulus immediately preceding the jth spike, and the output y_n is presumed to be simply a binary element indicating the occurrence of a spike or otherwise. The STA is simply the average stimulus preceding the spike. Chichilnisky 2001 [54] presents a simple argument that the STA provides a consistent and unbiased estimator for $\hat{\mathbf{k}}$ in the case of radially symmetric stimuli. We can infact extend this result to cases of elliptically symmetric stimuli by simply transforming the stimuli to by spherically symmetric – that is, by 'whitening' the stimulus. We encountered this exact idea in the linear case. Recalling

$$\hat{\mathbf{k}}_{LS} = (X^T X)^{-1} X^T \mathbf{y}, \tag{2.10}$$

then, when the stimulus is radially symmetric, the matrix $X^T X$ is a multiple of the identity matrix (Appendix A.1), which means the least squares estimator is

$$\hat{\mathbf{k}}_{LS} = \sigma_X^2 X^T \mathbf{y}.$$

When $\mathbf{y}$ is a binary vector of spike occurrences the least squares estimator is then proportional to the STA. In cases when X is elliptically symmetric, the estimator (2.10) is in fact based on the STA taken from a whitened version of the stimulus.[3] This is discussed in more detail in Theunissen 2001[262]. Chichilnisky 2001 further shows how to estimate the non-linearity $f(.)$ once the filter $k(.)$ has been estimated. The same methodology is applied to a BCI dataset in Shoham et al. 2005 [241]. Paninski 2003 gives convergence and consistency results for the STA [198], including cases where the estimator is optimal, which are described below.

It is worth pointing out that the STA corresponds to an estimate of the first term of the Weiner series. However, STA-based methods allow for the estimation of a large family of non-linearities, provided the stimulus is symmetric, so these methods are in a sense more general than Weiner/Volterra cascade methods. Indeed, models based on estimation of the Weiner series allow for estimation of only the first few non-linear terms in the expansion (after which data requirements for reliable estimation become prohibitive), whereas STA-based methods can in principal recover

[3] The argument runs as follows: begin by expression X in its QR factorization

$$\begin{aligned} X =& Q \begin{bmatrix} R \\ 0 \end{bmatrix}, \text{ or } X_R R = Q \begin{bmatrix} I \\ 0 \end{bmatrix} R \\ \Rightarrow X_R =& Q \begin{bmatrix} I \\ 0 \end{bmatrix}, \text{ or } X_r = X R^{-1} \\ \Rightarrow (X_r^T X_r) =& I \end{aligned}$$

so that the transformed data, X_R, is spherically symmetric. Then

$$\begin{aligned} \mathbf{k}_{LS} =& (X^T X)^{-1} \left(X^T \mathbf{y}\right) = (R^T R)^{-1} \mathbf{k}_{STA} \\ =& (R^T R)^{-1} R^T \left(X_r^T \mathbf{y}\right) = R^{-1} \mathbf{k}_{RSTA} \end{aligned}$$

which shows the least squares estimator $\mathbf{k}_{LS}$ is equal to the STA of the whitened data, $\mathbf{k}_{RSTA}$, expressed in the original basis. It also shows that $R^{-T} \mathbf{k}_{STA} = \mathbf{k}_{RSTA}$. Note that R is a Cholesky factor of the matrix $X^T X$. See [227, 201].

non-linearities of arbitrarily high order with relatively little data. This is important because neurons often exhibit saturating non-linearities which demand a high order approximation.

2.2.2 Gaussian noise and the spike triggered covariance

The spike triggered covariance (STC) is the second centered moment of the spike-triggered stimulus ensemble:

$$\hat{\Lambda} = \frac{1}{J}\sum_{n=1}^{N} y_n(\mathbf{x}_n - \mu)^T(\mathbf{x}_n - \mu)$$

again assuming a binary output y_n. The STC analysis is used to identify eigenvectors for Λ whose eigenvalues are significantly different to the entire stimulus covariance matrix. Paninski 2003 shows that $\hat{\Lambda}$ is a consistent estimator of the second moment when the stimulus is Gaussian [198]. This is actually a more strict requirement than the consistency requirement for the STA estimator (spherical or elliptical). In STC, attention is given to both the largest and smallest principal components of $\hat{\Lambda}$, which correspond to excitatory and suppressive stimulus directions respectively [227]. These methods are reviewed in Simoncelli and Paninski 2004 and Schwartz et al 2006[243, 227]

Both STA and STC analyses can be thought of as 'method-of-moment' type approaches, which focus on equating population and samples means but do not postulate a likelihood or probability model. In the case of a linear or quadratic model with Gaussian errors, such a likelihood model can be proposed and related to the estimators $\mathbf{k}_{STA}$, as was discussed in Sections 2.2.1 and 2.3. Other relations between the STA and an MLE exist. For example, Park and Pillow 2011 [201] observe that estimating parameters of the non-linear Poisson model

$$\lambda = f(\mathbf{x}) = \exp\left(\frac{1}{2}\mathbf{x}^T C\mathbf{x} + b^T\mathbf{x} + a\right)$$

for $y \sim \text{Pn}(\lambda)$ using an *expected likelihood* maximization scheme, in which the expectation is taken with respect to the stimulus distribution, corresponds to STA and STC estimation of b and C respectively, under constraints on C, b and a and under their respective consistent stimulus

conditions. The advantage of such a framework is that it provides a natural extension to cases in which stimuli are not symmetric or Gaussian, and that, by describing a likelihood-based model, a lot of associated likelihood and Bayesian theory and methodology becomes available. We describe one more such relation between an MLE and the STA, as discussed in Paninski 2004 [199], in Section (2.2.4).

2.2.3 Non-Gaussian noise and maximally informative dimensions

As just hinted at, stimuli are rarely Gaussian or elliptically symmetric, which prompts the development of methods that can handle estimation of $k(.)$ with arbitrary stimulus ensemble statistics. Information theoretic approaches are appropriate in cases of arbitrary distributions [198]. One common generalization of STA-based analysis to arbitrary stimulus distributions is a maximally informative dimensional (MID) analysis [238]. The idea is to identify axes which maximize the Kullback-Liebler (KL) divergence between the spike-triggered and entire stimulus ensemble when projected onto this axis:

$$\hat{\mathbf{k}}_{MID} = \text{argmax}_k D_{KL}\left[P(\mathbf{k}\cdot\mathbf{x}|\text{spike})\|P(\mathbf{k}\cdot\mathbf{x})\right] \tag{2.11}$$

This method works for any stimulus distribution, but has the drawback that it is computationally quite intensive to compute. In fact, estimation of $\mathbf{k}$ under LNP models has the property of an MID estimator, which is discussed in more detail below.

2.2.4 LNP models

The LNP model was defined in (2.9), recall that it consists simply of a linear filter applied to the stimulus, followed by a non-linear operation to obtain a firing rate which then generates a Poisson-distributed number of spikes according to this rate. The LNP model is attractive because it is simple enough to be efficiently estimated through likelihood maximization or related techniques and general enough that it can be applied to a variety of systems. There are cases when it is not

appropriate, for instance when the Poisson nature of the neuron is unreasonable. Extensions for dealing with that particular issue can be dealt with by using a GLM discussed below. Maximum likelihood estimation of $k(\cdot)$ has the benefit that it is reasonable to perform in cases when the stimulus distribution is neither symmetric nor Gaussian (e.g. "naturalistic" stimuli).

To make this more exact, consider the data being generated by a GLM having distribution $P_{\mathbf{k}^*}(\mathbf{x})$. If our model has the same form as the data generation process (same distribution family and non-linearity) then standard likelihood theory tells us the MLE is consistent – $\hat{\mathbf{k}}_{MLE} \to^P \mathbf{k}^*$. More generally, if the data generation process follows a distribution $Q_{\mathbf{k}^*}(\mathbf{x})$ obtained from a different non-linearity, g, then Paninski 2004 [199] proves that the ML estimate of $\mathbf{k}$ for the LNP model is still consistent given a log-concave non-linearity f and elliptically symmetric stimulus $\mathbf{x}$ (an extension of results found in Li and Duan 1989 and White 1982 [158, 269]). Further, he provides a condition under which the LNP likelihood is convex and can be efficiently maximized through any convex optimization method: $f(\cdot)$ is convex and log-concave and that the parameter space $\mathcal{K}$ is convex. Finally, Paninski notes also that in an LNP model with an exponential non-linearity, the STA provides a consistent and optimal estimator, assuming spherically or elliptically symmetric stimulus.[4]

For an LNP model in which the non-linearity f is modeled as a piece-wise constant function, Williamson et al 2015[273] show that the information each spike contains about the stimulus,

[4]The argument goes as follows[199]: the likelihood for an LNP model with non-linearity $f(.)$ is

$$\log p(\mathbf{y}|\mathbf{x}, \theta) = \sum_i y_i \log f(k \cdot x_i) - \Delta \sum_i f(k \cdot x_i) + c$$

Solving for the MLE requires taking the derivative of this likelihood with respect to k and setting it equal to zero. We find it satisfies:

$$\sum_i y_i \frac{f'\left(\hat{\mathbf{k}}_{MLE} \cdot \mathbf{x_i}\right)}{f\left(\hat{\mathbf{k}}_{MLE} \cdot \mathbf{x}_i\right)} \mathbf{x}_i = \int p(\mathbf{x}(t)) f'\left(\hat{\mathbf{k}}_{MLE} \cdot \mathbf{x}(t)\right) \mathbf{x}(t)\, dt$$

When x_i is spherically symmetric the RHS term is proportional to $\hat{\mathbf{k}}_{MLE}$ (as per[54]), giving

$$\sum_i y_i \frac{f'\left(\hat{\mathbf{k}}_{MLE} \cdot \mathbf{x_i}\right)}{f\left(\hat{\mathbf{k}}_{MLE} \cdot \mathbf{x}_i\right)} \mathbf{x}_i \propto \hat{\mathbf{k}}_{MLE},$$

where the LHS is a form of weighted STA. If $f'/f(\cdot)$ is a constant then $\hat{\mathbf{k}}_{MLE}$ is exactly proportional to the STA. This ratio is in fact a constant precisely when the non-linearity $f(\cdot)$ is exponential!

given a set of filters in matrix K, is equivalent to

$$\hat{I}_{SS}(K) = \frac{1}{n_{\text{sp}}}\mathcal{L}_{lnp}(\theta; D) - \frac{1}{n_{sp}}\mathcal{L}_{lnp}(\theta_0; D)$$

where $\mathcal{L}_{lnp}(\theta_0; D)$ represents the likelihood of a null spiking model in which the spiking is unrelated to the stimulus – a homogeneous Poisson firing model. This term does not depend on the model parameters K and hence maximizing the likelihood of the LNP model is equivalent the maximization the MID method performs (Equation (2.11)).

2.2.5 Generalized linear models

In cases where the non-linearity $f(\cdot)$ and the distribution of the number of spikes given the expected rate $\lambda = f(\cdot)$ is specified in advance, the model takes the form of a generalized linear model (GLM). In the statistics literature, GLMs are an extension of linear models and provide a candidate class of models for the above question. They are useful in that they allow a lot of techniques from the theory of linear models to be applied under a broader set of assumptions. The general aim is to describe the relationship between a set of N *observations* $\mathbf{y} = \{y_i\}_{i=1}^{N}$ and a set of K *covariates* contained in an $N \times K$ data matrix $X - K$ covariate values for each of the N datapoints. In a linear model, the following assumptions are made:

1. the expected value of a set of observations $\mathbb{E}(\mathbf{y}|X, \beta)$ is taken to be a linear sum of the covariates

   $$\mathbb{E}(\mathbf{y}|X, \beta) = X\beta,$$

   for a set of unknown parameters $\beta = \{\beta_j\}_{j=1}^{K}$

2. the vector of residuals ($\epsilon = \mathbf{y} - X\beta$) is assumed to be a normally distributed random variable with zero mean and a constant variance.

A generalized linear model extends this in two ways:

1. The expected value of the observations is taken to be a non-linear function of the sum of the covariates:

$$\mathbb{E}(\mathbf{y}) = g^{-1}(X\beta)$$

   The function $g(.)$ is called the *link* function.

2. The vector of residuals is taken to be a random variable whose distribution belongs to the exponential family

A general exponential family distribution is any distribution that can be written in the form

$$f(y;\theta) = \exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\theta)\right\}.$$

The exponential family of distributions includes a lot of commonly used distributions, including the normal distribution, the Gamma, the negative binomial, the binomial, and the Poisson.

In the computational neuroscience literature, GLM often means more specifically an extension of an LNP model with a specified non-linearity and a spike history, or auto-regressive, term. Following Eden et al 2004 [74] we can specify a GLM which approximates a point process, and is thus appropriate for modeling spike trains, as follows: recall from (2.7) the likelihood of a point process can be written as

$$P(\mathbf{y}_{1:N}|\lambda_{1:N}) = \exp\left[\sum_k -\lambda(t_k)\Delta t + \log(\lambda(t_k)\Delta t)\mathbf{y}_k\right] + o(\Delta t)$$

where Δt is taken to be small enough that $\lambda_k = \Lambda(t_{k-1}, t_k] = \lambda(t_k)\Delta t$ as before. Under the assumption that

$$\log(\lambda_k) = \sum x_{kj}\beta_j, \tag{2.12}$$

then the likelihood can be written as

$$P(\mathbf{y}_{1:N}|\lambda_{1:N}) = \exp\left[\sum_k -e^{\sum x_{kj}\beta_j} + \sum x_{kj}\beta_j y_k\right],$$

and the point process with CIF given by (2.12)corresponds to a GLM with a Poisson distribution and a log-link function. The log-likelihood takes the form

$$\mathcal{L}(\mathbf{k};\mathbf{x},\mathbf{y}) = \sum_k -e^{\sum x_{kj}\beta_j} + \sum x_{kj}\beta_j \Delta N_k \tag{2.13}$$

Thus, by approximating point processes of this form as a GLM we are able to use the analyses and inference machinery that exists for these models and apply them to spike trains. Note that the spike history term often incorporated into GLMs need not be elliptically symmetric or Gaussian distributed [198], meaning neither STC or STA approaches may be valid for their estimation. In that case MLE or a regularized modification are more appropriate, as discussed below.

2.2.6 Hawkes processes

As mentioned, a common component of GLMs used in computational neuroscience is a spike history, or auto-regressive, component:

$$g(\lambda_k) = \sum_{j=1}^{P} y_{k-j}k_j + \text{other covariates},$$

thus the firing rate for a neuron depends on its own spiking history. In some sense, the continuous-time equivalent of these GLMs is the point processed called the *Hawkes* process [120]. Recalling that

$$\lambda(t)dt = \mathbb{E}(dY(t)|\mathcal{F}_t)$$

describes the conditional intensity then the linear Hawkes process is defined as

$$\lambda(t) = \nu + \int_{-\infty}^{t^-} k(t-u)dY(u).$$

Thus, as with the GLM the process depends on its own history through the kernel $k : \mathbb{R}^+ \to \mathbb{R}$. The log-likelihood of the Hawkes process is one of the form:

$$-\int_0^T \lambda(t)\, dt + \sum_{j=1}^{J} \log \lambda(t_j)$$

as in (2.4), and similar to (2.13). Indeed, such models of spiking neural networks were some of the first applications considered for Hawkes processes [120]. (see [55, 33, 35] for other examples).

As in the GLM, the non-linear Hawkes process can be extended to have a non-linear dependence on its filtered history:

$$\lambda = \phi\left(\nu + \int_{-\infty}^{t^-} k(t-u) dY(u)\right)$$

for some rectifying non-linearity $\phi : \mathbb{R} \to \mathbb{R}^+$. The stability of such processes was considered in [35]: a condition that required for stability is that the non-linearity should be α-Lipschitz continuous, with $\alpha \leq 1$. Note in particular that a common choice for the GLM, $\phi = \exp$, is not α-Lipschitz continuous. A work-around in this case to ensure stability is to take a piece-wise exponential-linear function. This form of the Hawkes process is considered in models of genomic transcription regulatory elements in Carstensen 2010[43]. Penalized maximum likelihood is considered in Hansen 2010 and Hansen 2015 [115, 116], in which numerous connections to auto-regressive GLMs of the type considered in this document are made.

2.3 Illustration of estimation under temporally correlated input

Having reviewed a range of models for neural coding, we describe turn describe issues in e begin with the simplest form of Equation (2.2), the case when $f(\cdot)$ is linear:

$$y(x_{1:n}) \sim \sum_{i=0}^{P} k_i x_{n-i} \tag{2.14}$$

for a zero mean stationary process x_n. Note that the process x_n can be assumed to be zero mean without loss of generality, as the data can always be mean-centered. A more general version of

this model is explored in Theunissen *et al* 2001 [262]. There are obvious problems with this model, in particular the fact that the model allows for a negative firing rate depending on values of k, a and x. Nonetheless, it provides intuition for the effects of correlated input. We can learn most of what we need from this model, so describe it in detail.

For convenience in later discussion, we'll assume (2.14) is described by a kind of moving average model:

$$y_n = \sum_{i=0}^{P-1} k_i x_{n-i} + \epsilon_n$$

for IID white noise process ϵ_n of zero mean and finite variance. The process x_n, beyond being stationary, is unspecified. Stationarity implies the existence of an auto-covariance function

$$\Gamma_X(t) = \langle x_{n+t}, x_n \rangle \qquad \forall n.$$

More compactly, let X be the $(N \times P)$ matrix whose (i,j)th entry corresponds to x_{i-j+1} – the ith row corresponds to the history of x preceding timebin and including i. Equation (2.14) then becomes

$$\mathbf{y} = X\mathbf{k} + \epsilon.$$

Without assuming the precise form of distribution ϵ we can estimate $\mathbf{k}$ through a method of moments-type estimation (e.g. through the Yule-Walker equations[5]):

$$\begin{aligned} X^T\mathbf{y} &= X^T X\mathbf{k} \\ \Rightarrow \hat{\mathbf{k}}_{LS} &= (X^T X)^{-1} X^T \mathbf{y} \end{aligned} \tag{2.15}$$

[5]Note that the Yule-Walker estimator is separate to the Yule-Walker equations. The Yule-Walker estimator uses a slightly different estimator for $\hat{C}_{xx}$ and $\hat{C}_{xy}$. Refer to Lutkehpuhl 2007 [169] for details.

which, as notationally preempted, is infact the same as the least squares estimator:

$$\hat{\mathbf{k}}_{LS} = \text{argmin}_{\mathbf{k}} \|\mathbf{y} - X\mathbf{k}\|_2^2 .$$

Observations

1. Since $\mathbf{x}$ is a stationary stochastic process the $P \times P$ matrix $(X^T X)$ is an estimator of the auto-covariance matrix of process x up to time delays $P-1$, $C_{XX}(i,j) = \Gamma(|i-j|)$, which we'll denote $\hat{C_{xx}}$, and $X^T\mathbf{y}$ is an estimator of the cross-covariance vector $\hat{C_{xy}}$.

2. In this case we can show that C_{xx} is a scalar multiple of the identity matrix if the distribution $f_X(\mathbf{x})$ is radially symmetric.

2.3.1 How do auto-correlations effect estimation of $\mathbf{k}$?

The effect of correlated inputs then becomes apparent by considering properties of the matrix

$$\hat{C}_{xx} = X^T X,$$

which represents the a 'sensitivity' matrix of sorts. If we were to additionally assume a Gaussian white noise error term $\epsilon \sim \mathcal{N}(0, \sigma_Y^2)$ then the least squares estimator would of course also be the maximum likelihood estimator, from which we could compute confidence intervals based on standard likelihood theory. In fact, the least squares estimator in this case would be consistent, unbiased, and asymptotically efficient. None of these features stop the estimator from having a high variance on finite samples, however. To determine that we consider the conditioning of the Fisher information matrix, in this case $\hat{C}_{XX}$.

Estimator uncertainty

For comparison to later extensions, we'll consider $\hat{\mathbf{k}}_{LS}$ as an MLE, in which case asymptotic likelihood theory gives

$$\sqrt{K}(\hat{\mathbf{k}_{LS}} - \mathbf{k}) \sim \mathcal{N}\left(0, \left(\hat{C}_{xx}\right)^{-1} \sigma_Y^2\right) \tag{2.16}$$

as $K \to \infty$. This enables us to compute marginal $100\alpha\%$ confidence intervals for each component k_i as

$$\hat{k}_{LS,i} - z_{\alpha/2}\text{se}(\hat{k}_{LS,i}) \leq k_i \leq \hat{k}_{LS,i} + z_{\alpha/2}\text{se}(\hat{k}_{LS,i}), \tag{2.17}$$

where $z_x = \Phi^{-1}(1-x)$, with Φ^{-1} being the inverse of the distribution function of the standard normal distribution, and $\text{se}(\hat{k}_{LS,i})$ is the square root of the ith diagonal element of the matrix $\left(\hat{C}_{xx}\right)^{-1} \sigma_Y^2$.[6]

A sign of strong auto-correlation in X is that $\Gamma(p)$ decays slowly over the $0 < p < P-1$, such that at least some rows of $C_{XX}(i,\cdot) = \Gamma(|i - .|)$ will be roughly collinear and that C_{XX} will be poorly conditioned. We see in (2.17) that $\hat{C_{xx}}$ being close to singular results in increased parameter uncertainty.

The behavior is also made clear by considering the form of the cost function that $\hat{\mathbf{k}}_{LS}$ minimizes:

$$C(\mathbf{k}) = \left\|\mathbf{y} - \tilde{X}\mathbf{k}\right\|^2$$

[6]Note that really we should be defining a joint confidence *set* in the P dimensional parameter space of $\mathbf{k}$, instead of a set of P marginal confidence *intervals*. This amounts to approximating the joint confidence region by a P dimensional hypercube, which may be appropriate if each component of $\hat{\mathbf{k}}_{LS,i}$were independent. However, the actual joint confidence region described by (2.16) is an ellipsoid defined by

$$C_N = \left\{\mathbf{x} \in \mathbb{R}^P : (\hat{\mathbf{k}_{LS}} - \mathbf{x})^T X^T X (\hat{\mathbf{k}_{LS}} - \mathbf{x}) \leq \frac{\chi_P^2(1-\alpha)\sigma_x}{\sqrt{N}}\right\} \tag{2.18}$$

(when σ_x^2 is presumed to be known, or

$$C_N = \left\{\mathbf{x} \in \mathbb{R}^P : (\hat{\mathbf{k}_{LS}} - \mathbf{x})^T X^T X (\hat{\mathbf{k}_{LS}} - \mathbf{x}) \leq F_{P,N-P}(1-\alpha)s^2 P\right\}$$

when σ_x^2 is also estimated from the data, by $s^2 = \epsilon^T\epsilon/(N-P)$.) This step is rarely taken in practice however, because the confidence set is typically harder to visualize and interpret. This is discussed further in Draper and Smith 1998 [71]. The argument presented here is unchanged whether (2.17) or (2.18) is used.

or

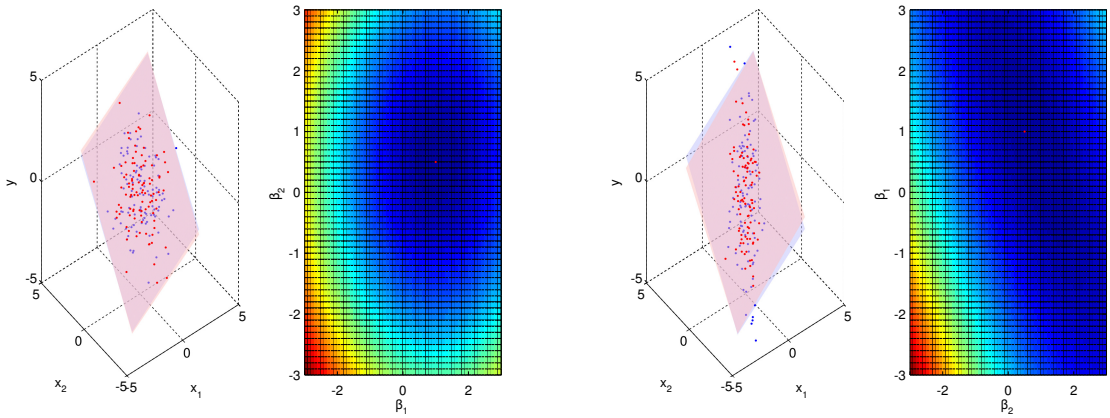
$$C(\mathbf{k}) = \sum_{n=1}^{N}(y_n - X_{ni}k_i)^2 = \sum_{n=1}^{N} y_n^2 - 2y_n X_{ni}k_i + X_{ni}^2 k_i^2 \sim -2\mathbf{y}^T X\mathbf{k} + \mathbf{k}^T X^T X\mathbf{k}.$$

We note that each observation $X_{n,:}$ contributes a rank one matrix $(X_{n,:})^T X_{n,:}$ to the covariance matrix $X^T X$, such that the paraboloid can be thought of as a weighted sum of these rank one matrices. The optimal estimate lies at the minimum of this weighted sum of soft constraints. Each observation contributes a quadratic function $(y_n - X_{n,:}\mathbf{k})^2$ which penalizes estimates $\mathbf{k}$ that lie parallel to the vector $X_{n,:}$, thus we expect the estimate $\hat{\mathbf{k}}$ to be more constrained in these directions. Note that if vectors tend to $X_{n,:}$ lie in a particular direction (are collinear in two or more predictors), then the estimator $\hat{\mathbf{k}}$ will be poorly constrained in the direction perpendicular to this axis. Another way to state the same thing is, if predictors i and j are strongly linearly correlated, $ax_{:,i} + bx_{:,j} \sim 0$, then any parameter whose relevant components lie parallel to (a, b) (perpendicular to $(x_{:,i}, x_{:,j})$) will have very little bearing on the model fit, as any such (k_i, k_j) will satisfy $k_i x_{:,i} + k_j x_{:,j} \approx 0$ and make very little contribution to the cost function. Figure 2.2 contains a demonstration of this phenomenon.

Spectral considerations

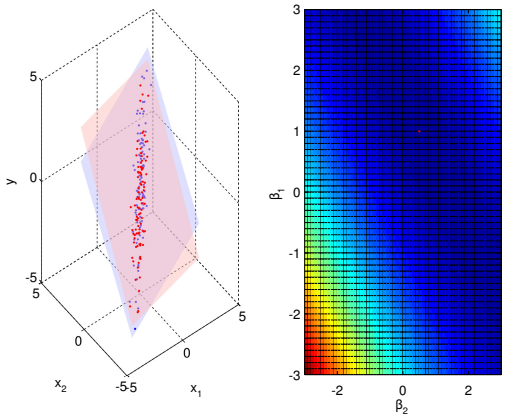
Insight can further be gained by considering the spectral representation of the series. Since C_{XX} is the autocovariance matrix of a stationary process, it is a Toeplitz matrix

$$C_{XX} = \begin{bmatrix} \Gamma_X(0) & \Gamma_X(1) & \cdots & \Gamma_X(P-1) \\ \Gamma_X(1) & \Gamma_X(0) & \Gamma_X(1) & \vdots \\ \vdots & \Gamma_X(1) & \ddots & \\ \Gamma_X(P-1) & \cdots & & \Gamma_X(0) \end{bmatrix}.$$



(a) a

(b) a



(c) a

Figure 2.2: Least square estimation of data generated by a linear regression model $y = x_1\beta_1 + x_2\beta_2 + \epsilon$ when x_1 and x_2 are correlated. As the correlation coefficient between $x_1$and x_2 increases (a)→(c), fitting a plane (with normal vector $(\hat{\beta}_1, \hat{\beta}_2)$) becomes increasingly ill-posed. The left panels show two independent sets of observations generated according to the linear model, as well as, in the same color, the plane of best fit according to the least squares estimate. Higher correlations results in increased variability in the estimates $\hat{\beta}$ from sample to sample. Shown in the right panel is the cost function $\|\mathbf{y} - X\mathbf{k}\|^2$ whose minimum is to be taken as the least squares estimate. Red point shows the true parameter values. Regions of low curvature correspond to regions of high uncertainty. Correlated inputs result in a highly elliptical cost function, containing axes which are relatively unconstrained.

This is significant because the eigendecomposition of the Toeplitz matrix relates to the discrete Fourier transform of the covariance function $\Gamma_X(n)$. The eigenvectors are of the form:

$$\mathbf{q}_k = [1, \exp(i\omega_k), \exp(2i\omega_k), \cdots, \exp((P-1)i\omega_k)]^T,$$

and the eigenvalues are of the form

$$\lambda_k = \sum_{p=0}^{P-1} \Gamma_X(p) \exp(i\omega_k p),$$

for $\omega_k = \frac{2\pi k}{P}$. The (in this case discrete) Fourier transform of the auto-covariance function is known as the spectral density, $\Lambda_X(k) = \lambda_k$, which represents the power of the stimulus at each frequency ω_k. Thus the decomposition

$$C_{xx} = Q\Lambda_X Q^*$$

for eigenvectors $\mathbf{q}_k$ and eigenvalues λ_k provides a useful intuition: since λ_k represents the power of the stimulus at ω_k , we observe that C_{xx} being badly conditioned (condition number $\kappa_C = |\lambda_{max}|/|\lambda_{min}|$) corresponds to the power of the stimulus across frequencies varying over large orders of magnitude – that is, there are some frequencies which are relatively under represented, which results in $\hat{\mathbf{k}_{LS}}$ being undersampled at those frequencies. Of course, if x_n is white noise then $\kappa_C = 1$ and the stimulus represents all frequencies equally.

As a very simple, informal example of the above discussion, consider data being generated by the AR(1) model

$$x_t = \alpha x_{t-1} + \eta_t \tag{2.19}$$

for $0 \ll \alpha < 1$, and η_t a white noise process with variance σ_X^2 . This has an MA(∞) representation

$$x_t = \sum_{i=0}^{\infty} \alpha^i \eta_{t-i}$$

and hence auto-covariance relation

$$\Gamma_X(n) = \alpha^n \sigma_X^2.$$

Thus for α close to one the time series has a slowly decaying auto-covariance. What happens if we try fitting an MA(P) model for y_n to the x_n data obtained from (2.19)? The spectral density of the x_n series is given by

$$\lambda_k = \sum_{p=0}^{P-1} \sigma_X^2 \alpha^p \exp(i\omega_k p) = \sigma_X^2 \frac{1-\alpha^P}{1-\alpha\exp(-i\omega_k)} \text{ (geometric formula)}$$

which tells us, assuming P is even for simplicity, that

$$\begin{aligned}\lambda_0 &= \frac{1-\alpha^P}{1-\alpha} \\ \lambda_{P/2} &= \frac{1-\alpha^P}{1+\alpha}\end{aligned}$$

and hence the condition number of the covariance matrix is a least the ratio of these two eigenvalues, that is

$$\kappa(C_{XX}) > \frac{1+\alpha}{1-\alpha}$$

which tends to ∞ as $\alpha \to 1$. Hence, for a large number of samples we expect our estimated $\hat{C}_{XX}$ matrix will tend to an ill-conditioned matrix. In fact in this case the conditioning doesn't depend on the order P; any $P > 1$ will be over-parameterized and ill-conditioned for α close to one.

2.4 Dealing with ill-posed problems

Even in cases where the STA or MLE provides an efficient, consistent and unbiased estimator, the parameter estimation problem can be an ill-posed one, as was demonstrated in the case of a linear model in Section 2.3. Correlated inputs tend to create effective over-parameterizations of the data,

or over-fitting. In this section we describe how this manifests in GLMs before discussing ways to remedy this over-fitting.

In general, the Fisher information for a GLM is

$$\mathcal{I}(\beta) = X^T W X/\phi$$

where W is given by:

$$W_{ii} = [\mathbb{V}(\theta_i)]^{-1} \left(\frac{d^2\theta}{d\eta^2}\right)_i, \quad W_{ij} = 0, \text{ for } i \neq j,$$

that is, it is the matrix having each observation's inverse variance on the diagonal. In the case of a Poisson likelihood with log link function W has the form

$$W_{ii} = e^{-X_{ij}\beta_j}, \qquad \phi = 1 \tag{2.20}$$

As before, based on the maximum likelihood estimator's asymptotic normality, we can specify confidence intervals based on:

$$\sqrt{K}(\hat{\beta} - \beta) \sim \mathcal{N}(0, \mathcal{I}(\beta)^{-1})$$

for large number of samples K. The form of $\mathcal{I}(\beta)$ is similar to the linear model case (Section (2.3.1)), thus we expect correlated inputs to manifest in a similar way when estimating β. In fact, exactly as before, the Fisher information matrix can be split into a sum of rank one matrices, one for each observation:

$$\mathcal{I}(\beta) = \sum_{k=1}^{K} \left(X_{k,:}\right)^T X_{k,:} W_{kk}$$

weighted by the predicted variance of that observation (e.g. Equation (2.20)). Thinking of the Fisher information as a second-order approximation to the likelihood function at $\hat{\mathbf{k}}_{MLE}$ then each of these observations serves as a quadratic penalty to the likelihood that constrains estimates

parallel to $X_{n,:}$. Correlated, or collinear, predictors result in unconstrained parameter estimates in directions perpendicular to $X_{n,:}$. Figure 2.3 demonstrates estimation of a simple GLM is performed with correlated input data, in equivalence to the linear example.

2.4.1 Overfitting and maximum a posteriori estimation

Over-fitting in a likelihood framework is simple to understand: we have found parameters that satisfy

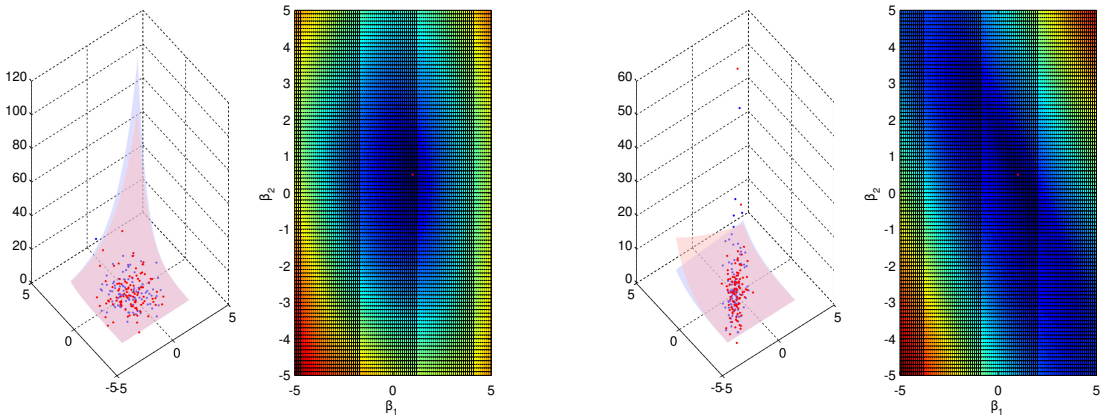
$$\mathcal{L}(\hat{\mathbf{k}}_{MLE}; \mathbf{x}, \mathbf{y}) = \max_{\mathbf{k}} \mathcal{L}(\mathbf{k}; \mathbf{x}, \mathbf{y})$$

which is a metric that can always be improved simply by the addition of more parameters. Indeed, for $\mathbf{k}$ having the same dimensionality as $\mathbf{y}$ we can construct a model that fits the observed data *exactly*. But this is not the aim of constructing a model – the MLE by itself is slightly misaligned with the goal of constructing of a parsimonious model that explains $\mathbf{y}$. Fitting the data exactly corresponds to fitting both the expected trend, $\langle \mathbf{y} \rangle = \lambda(\mathbf{x})$, which we wish to explain, and fitting what we consider errors, $\mathbf{y} - \lambda(\mathbf{x})$, that are perturbations around that trend not of interest. We expect that, with repeated sampling of data generated under the same conditions, the trend component will remain the same, while the error component will vary from sample to sample. This suggests that if we want a model that captures the trend and not the errors, a better metric to judge a model by is how well it predicts novel samples of data that it was not trained on – its *predictive,* or out of sample, error, as opposed to its *training*, or within sample, error.

One way of achieving this would be to search for an estimator that maximizes the likelihood of $\mathbf{k}$ given not the data $\mathbf{y}$ but instead the expected value $\lambda(\mathbf{x})$:

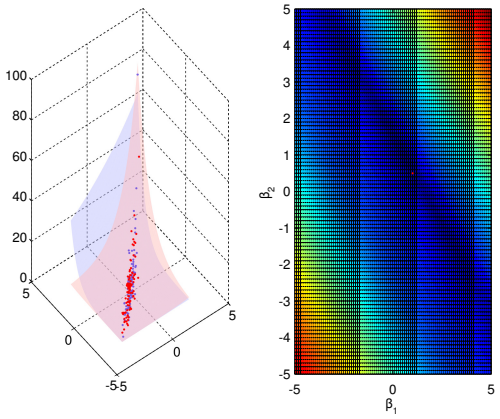
$$\hat{\mathbf{k}}_{AIC} = \text{argmax}_{\mathbf{k}} \mathcal{L}(\mathbf{k}; \mathbf{x}, \lambda).$$

This would work if we did observe the true λ, but we don't. In the case of linear models we can estimate λ, however, which will result in what is known as the Akaike information criterion (AIC)



(a) a

(b) a



(c) a

Figure 2.3: Estimation of LN Poisson regression model $y \sim \mathrm{Pn}(\exp[x_1\beta_1 + x_2\beta_2])$ when x_1 and x_2 are correlated. Gaussian (elliptically symmetric) stimulus data of different correlation coefficients is used to estimate (β_1, β_2). The left panels show two independent sets of observations generated according to the GLM, as well as, in the same color, the plane of best fit according to the ML estimate. Higher correlations results in increased variability in the estimates $\hat{\beta}$ from sample to sample. Shown in the right panel is the cost function $\|\mathbf{y} - X\mathbf{k}\|^2$ whose minimum is to be taken as the ML estimate. Red point shows the true parameter values. Regions of low curvature correspond to regions of high uncertainty. Correlated inputs result in a highly elliptical cost function, containing axes which are relatively unconstrained. Compared with the linear model, the cost function is no longer elliptically symmetric. Shown also is the STA estimator, which, for this particular LNP model is is asymptotically equivalent to the ML estimate.

estimator[5]:

$$\hat{\mathbf{k}}_{AIC} = \text{argmax}_{\mathbf{k}} 2\left[\mathcal{L}(\mathbf{k};\mathbf{x},\mathbf{y}) - \|\mathbf{k}\|_0\right]. \tag{2.21}$$

where $\|\mathbf{k}\|_0$ simply indicates the total number of parameters in vector $\mathbf{k}$ (refer to Wood 2006 for the derivation[279]; though denoted as a norm it is not, it is a quasi-norm at best). The AIC estimator is a very simple to implement method that penalizes unnecessary model complexity.[7] It can further be shown to be asymptotically equivalent to performing leave-one-out-cross-validation (LOOCV) [254], and thus can be seen as a correction of the observed training error to provide an estimate of prediction error. The same AIC estimator can be derived in a way that applies not just to linear models but to any ML estimation problem.

There are a number of other ways to counter this problem, many of which can be interpreted in a Bayesian framework in which we are bringing prior knowledge to the estimation problem that penalizes extraneous parameters in the model and which thus help prevent overfitting. This translation of an ill-posed problem to a well-posed one through extra assumptions or information is known as *regularization*. In particular, given a prior distribution on the space of parameter estimates, $p_K(\mathbf{k})$, and we can find an estimate $\hat{\mathbf{k}}_{MAP}$ that maximizes the a posteriori probability

$$\hat{\mathbf{k}}_{MAP} = \text{argmax}_{\mathbf{k}} \log p(\mathbf{k}|\mathbf{x};\mathbf{y}) = \text{argmax}_{\mathbf{k}} \left[\mathcal{L}(\mathbf{k};\mathbf{x},\mathbf{y}) + \log p_K(\mathbf{k})\right].$$

Different choices of prior penalizing function $Q(\mathbf{k}) \equiv \log p_K(\mathbf{k})$ correspond to different regularizing assumptions and are discussed below.

2.4.2 Information criteria

The AIC (2.21) is an example of myriad related information criteria:

[7] Its simplicity must go some way to explaining its popularity – Akaike's paper [5] is one of the top 100 papers cited of all time!

- Bayesian information criterion (BIC)[228] instead maximizes

$$\hat{\mathbf{k}}_{BIC} = \text{argmax}_{\mathbf{k}} \left[2\mathcal{L}(\mathbf{k}; \mathbf{x}, \mathbf{y}) - \|\mathbf{k}\|_0 \log N \right]$$

  based on information theoretic arguments.

- Deviance information criterion (DIC) is a Bayesian method that uses the expected deviance (or log-likelihood) defined as

$$\bar{D} = -2\mathbb{E}(\mathcal{L}(\mathbf{k})|\mathbf{x}, \mathbf{y})$$

  to identify the number of effective parameters as

$$p_D = \bar{D} + 2\mathcal{L}(\mathbb{E}(\mathbf{k})|\mathbf{x}, \mathbf{y})$$

  from which the DIC estimator is given as

$$\hat{\mathbf{k}}_{DIC} = \text{argmax}_{\mathbf{k}} + 2p_D$$

- Hannan-Quinn information criterion [211] is a method used to estimate the optimal order of an auto-regressive model, and takes the form

$$\hat{\mathbf{k}}_{HQIC} = \text{argmax}_{\mathbf{k}} 2 \left[\mathcal{L}(\mathbf{k}; \mathbf{x}, \mathbf{y}) - \frac{1}{N} \|\mathbf{k}\|_0 \log \log N \right].$$

In addition to these, there is a corrected version of the AIC (AICc), the Watanabe Akaike IC (WAIC), and the focused information criterion (FIC). In general, information criteria can provide a principled solution to the model selection problem which can in some cases be preferable to Wald or likelihood ratio hypothesis tests of nested models. For a general framework to understand the relation between various information criteria and different versions of cross-validation refer to Gelman et al 2013 [98].

The AIC and BIC are used in model selection of an LNP model of primary motor neurons encoding hand/wrist motion in Shoham et al 2005 [241].

2.4.3 Gaussian priors

A simple prior to incorporate is the Gaussian prior:

$$Q(\mathbf{k}) = \mathbf{k}^T C \mathbf{k}$$

Which, in the linear estimation case, results in a log-posterior:

$$\log p(\mathbf{k}|\mathbf{x}, \mathbf{y}) = -\frac{1}{2\sigma^2}\|\mathbf{y} - X\mathbf{k}\|_2^2 - \frac{1}{2}\mathbf{k}^T C \mathbf{k}$$

The regularized least squares estimator can then be specified exactly

$$\hat{\mathbf{k}}_{RLS} = (X^T X + \sigma^2 C)^{-1} X^T \mathbf{y}$$

Different forms of C have different properties. A matrix C that represents the Laplacian, or second derivative, can be used to estimate filters that vary smoothly. This was implemented in a model of receptive field filters of primary visual cortical neurons in Smyth et al 2003[245]. See also the report by Rad and Paninski 2010 [213] that discusses estimation of such models. Sahani and Linden 2003[220] also discusses this model in the context of estimation of spatiotemporal and spectrotemporal filters.

2.4.4 Ridge regularization

Also known as Tikhonov regularization, and first studied in the statistical estimation context by Hoerl and Kennard 1970 [126] under the name ridge regression, here the matrix C is of the form

$\Gamma^T\Gamma$ such that the log-prior becomes

$$Q(\mathbf{k}) = \lambda\|\Gamma\mathbf{k}\|_2^2.$$

A particularly simple form of Gaussian prior is to take C to be a scalar multiple, λ, of the identity matrix. The effect of this prior on the conditioning of the problem can be determined using the singular value decomposition. If

$$\hat{\mathbf{k}}_{RR} = (X^TX + \sigma^2 I)^{-1}X^T\mathbf{y}$$

then it can be shown that if $X = U\Sigma V^*$ and that

$$\hat{\mathbf{k}}_{RR} = VDU^*\mathbf{y}$$

where

$$D_{ii} = \frac{\sigma_i}{\sigma_i^2 + \alpha^2}.$$

Thus the conditioning of X^TX is improved.

Ridge regression is applied in the estimation of spectro-temporal receptive fields in a model of neurons in rat auditory cortex in Machens, Wehr and Zador 2003 [170].

2.4.5 LASSO and Group LASSO

More generally, a broad class of priors are known to enforce sparsity but have nicer properties as estimators than some of the information criteria discussed above. For instance, if our goal is penalize parameter vectors with a lot of non-zero components then something like the 'zero-norm' seems appropriate (as in (2.21)). However, $\|\mathbf{k}\|_0$ is not convex which makes the resulting optimization more complicated. A well known alternative then is to take its convex envelope, the L^1 norm, which simplifies the optimization while maintaining its tendency towards sparsity. A

convex prior maintains the convexity of the MAP estimation, provided, of course, the likelihood is also convex. The estimator takes the form

$$\hat{\mathbf{k}}_{L1} = \text{argmax}_{\mathbf{k}} \mathcal{L}(\mathbf{k}; \mathbf{x}, \mathbf{y}) - \lambda \|\mathbf{k}\|_1 \tag{2.22}$$

for some scale parameter λ the can be determined, for instance, through cross-validation. In the context of linear regression, this is known as a LASSO (least absolute shrinkage and selection operator) estimator [263]. In some cases this convex envelope can in fact be shown to give exactly the same estimators as the $\|\mathbf{k}\|_0$ penalty[69].

A similar approach can be taken in models that contain groups of parameters, in which the sparsity is more natural to enforce on a group-parameter level rather than an individual parameter level. For instance, a GLM may be specified for a network of neurons, in which each unit's rate may depend on the activity of all other neurons in the network:

$$\lambda_n^{(i)} = g\left(\sum_l \sum_j \psi_{lj}^{(i)} y_{n-j}^{(i)} + \sum_j k_j^{(i)} x_{n-j}\right)$$

where the connectivity strengths from unit j to i at time-lag l are given by $\psi_{jl}^{(i)}$. In this model it makes sense to enforce sparsity for each group of parameters that indicate a connection between a pair of units. For this a 'group LASSO' approach can be taken:

$$\hat{\mathbf{\Theta}}_{GL1}^{(i)} = \text{argmax}_{\mathbf{k},\Psi} \mathcal{L}(\mathbf{k}, \Psi; \mathbf{x}, \mathbf{y}^{(i)}, Y) - \lambda \|\Psi\|_{2,1} - \gamma \|\mathbf{k}\|_1$$

in which the parameter $\Theta^{(i)} = \left\{\mathbf{k}^{(i)}, \Psi^{(i)}\right\}$ contains the stimulus filters and matrix of connectivities associated with unit i. Here the $\mathrm{L}^{2,1}$ norm indicates the L^2 norm taken with respect to the rows of the connectivity matrix, followed by the L^1 norm on the resulting vector:

$$\|\Psi\|_{2,1} = \sum_i \|\psi_{i,:}\|_2$$

Such group LASSO methods have been used for inference of network connectivities Bolstad, van Veen and Nowak 2011[32], and Shojaie and Michailidis 2010[242], and is similar to what is performed by Pillow et al 2008 [208].

2.4.6 Reduced parameter spaces and alternative bases

Another method to enforce expected characteristics of $\mathbf{k}$ is to express the filter in terms of a different set of basis functions $\left[\mathbf{k}'_1, \ldots, \mathbf{k}'_{p'}\right] \equiv K'$ which possess some desired behavior. For instance, smooth basis functions if a smooth filter $\mathbf{k}$ is desired. This could be incorporated into a Bayesian framework, if so desired, but since this amounts to simply making $p(\mathbf{k})$ flat for $\mathbf{k} \in \text{span}(K')$ and zero otherwise, it isn't particularly interesting or computationally practical as a prior. Instead, the stimulus can be projected on to the basis[8] K':

$$\begin{aligned} K'X'^T \approx & X^T \\ \Rightarrow X'^T \approx & \left(K'^T K'\right)^{-1} K'^T X^T \end{aligned}$$

where the pseudoinverse is used because K' is not square. The filter is found through MLE in the new basis.

There are many examples of this technique being used, for example a common choice is a 'stretched cosine' basis like:

$$k_i(t) = \begin{cases} \frac{1}{2}\left(\cos\left[a\log(t-\psi_i) - \phi_i\right] + 1\right), & \phi_i - \pi < a\log(t-\psi_i) < \phi_i + \pi; \\ 0, & \text{otherwise;} \end{cases} \tag{2.23}$$

[8] The term basis is used a little generously here since the span of K' is a proper subset of the span of K, not the entire space ($p' \ll p$). Dimensionality reduction is the name of the game.

which provides high temporal precision near the start of the filter (near the time of the spike) and less precision away from the time of the spike, after which it varies smoothly[9]. Pillow et al 2008 [208] uses this form in a GLM of retinal ganglion cells (RGCs), and Keat et al 2001[139] use this form in models of RGCs and LGN neurons. Other examples include Hermite polynomials in a model of V1 neurons [268], Zernicke basis in a model of place cells in hippocampus [19], and cubic splines in a model of supplementary eye field neurons [137].

Along the lines of reduced stimulus or parameter spaces we mention two other possibilities. First, the stimulus space can be reduced to a subspace which captures a large proportion of its variance. This amounts to a principle component analysis (PCA) of the matrix X:

$$X = U^* \Sigma V$$

Such an analysis is performed with a linear model in the Gaussian process factor analysis of Yu et al 2010 [284], and has been used to characterize low-dimensional spaces of neural activity in primary motor cortex. This is discussed further in Aljadeff et al 2016 [7].

Second, if a spatio-temporal filter is to be identified, $k(x,t)$, then a large reduction in the parameter space is to assume that the spatial and temporal components are separable

$$k(x,t) = k_X(x) k_T(t).$$

which is described in Ahrens et al 2008 [4].

[9]For binsize Δt, if we choose parameters as follows:

$$\begin{aligned}\phi_1 &= a \log\left\{\Delta t \left[1 - \exp(-\pi/a)\right]^{-1}\right\} \\ \phi_j &= \phi_{j-1} + \pi \\ \psi &= -\exp\left(\frac{\phi_0}{a}\right)\end{aligned}$$

then the resulting basis will be such that $k_0(t)$ and $k_1(t)$ have their peaks at $t = 0$ and $t = \Delta t$ respectively, meaning the basis is as finely resolved as possible at the time of the spike, while becoming less resolved for larger times. The parameter a then sets the rate at which this resolution decreases.

3

Effective connectivity in neuronal networks

3.1 Introduction

Having multi-unit recordings available facilitates understanding neural dynamics and interactions between recorded neurons that can supplement the understanding observed by models considered in the previous chapter. One way of studying such interactions is through estimating measures of correlation or *connectivity*. The appropriate notion of connectivity depends on the circumstance being considered, and on the data that is available. Following Sporns, we can distinguish between a few notions of connectivity *[247]*. In some cases, it is possible to estimate the strength of anatomical connections between neurons, whether synaptic connections, or connectivity on a

larger scale. Such connectivity is directed, and is assumed to not be mediated by unobserved processes. This is referred to as *structural connectivity*.

In other cases this information is unavailable and only statistical measures of relation can be obtained. Measures that estimate statistical dependence between two units without considering the other observed units can be termed *functional connectivity*. These methods may be directed or undirected connections, and may be mediated through unobserved processed. Correlation or coherence are examples of functional connectivity. Measures that estimate statistical dependence between two units when all other units' activity has been considered, thereby attempting to provide a measure that indicates a more specific interaction between two units, can be called *effective connectivity*. These methods are directed, and may also be mediated through unobserved processes. Conditional Granger causality or transfer entropy are examples of effective connectivity. We note that other authors use functional and effective connectivity differently, sometimes interchangeably (e.g. [257, 133, 143, 253, 251, 250, 246]).

The determination of effective connectivity is a well-studied problem. In this Chapter we first review the mathematical methods that are applied to address the problem of inferring directed influences between variates more generally. This can be phrased in the context of a 'Granger causality'-type framework, typically invoking time series theory, or it can be phrased as a network inference problem, invoking theory of graphical models and regularization techniques. Applications to neuroscience are then discussed.

3.2 Causal inference review

3.2.1 Granger causality

The motivation behind what is now referred to as Granger causality can be traced to Weiner, whose observation that "a causal influence should manifest in improving the predictability of the

driven process when the driving process is observed"[272] provided inspiration to Granger. Thus, Granger causality [110] is based on two properties often associated with a causal relationship [1]:

1. A cause precedes an effect in time
2. Information in a cause's past must improve the prediction of the effect above and beyond information contained in the collective past of all other measured variables (including the effect).

Based on these properties, Granger causality can be formulated in a straightforward manner in the context of (vector) auto-regressive (VAR) time series. Let $\mathbf{x} = (x_1(t), x_2(t))$ be a stationary time series describe by the following VAR:

$$x_1(t) = \sum_{j=1}^{P} A_{11,j} x_1(t-j) + \sum_{j=1}^{P} A_{12,j} x_2(t-j) + \epsilon_1(t)$$
$$x_2(t) = \sum_{j=1}^{P} A_{21,j} x_1(t-j) + \sum_{j=1}^{P} A_{22,j} x_2(t-j) + \epsilon_2(t)$$

Given a set of time series data, the test for Granger causality from $x_1 \to x_2$ can be phrased as a test that the set of parameters $A_{21,j} \equiv 0$, that is, that the series x_2 is independent of the series x_1:

$$x_2`(t) = \sum_{j=1}^{P} A_{22,j} x_2(t-j) + \epsilon_2`(t)$$

for error terms $\epsilon_1(t), \epsilon_2(t)$ and $\epsilon_2'(t)$ being normally distributed having variance (or covariance matrix, in case the series are generalized to multivariate) Σ_1, Σ_2 and Σ_2', respectively. A number of such tests exist, however a typical measure of Granger causality is

$$\mathcal{F}_{x_1 \to x_2} \equiv \log \frac{|\Sigma'_{xx}|}{|\Sigma_{xx}|}$$

[1] Causality is philosophically loaded word, and 'Granger causality' should not be misinterpreted as asserting anything strong in this regard; on its own it does not provide an appropriate statistical framework to answer questions about 'true' causality. It is a measure focused on quantifying predictability of one variable based on information from another; some advocate the term Granger prediction as a way of avoiding any philosophical issues. Refer to Chicharro and Ledberg 2012[53] for a discussion on how the formulation of causality described here may relate to other formulations, in the context of neuroscience.

which under the null hypothesis of Granger non-causality:

$$H_0 : A_{21,j} = 0 \quad \forall j$$

follows a χ^2 distribution having P degrees of freedom[101, 102].This measure has the benefit that it represents the log-likelihood ratio between the full and reduced MLE estimated models (a result that can be traced to [270]), which provides an obvious route for generalizing Granger causality to multivariate and non-linear/non-Gaussian models (see for instance the discussion Barnett and Bossomaier 2012 [21]), some of which are described below.

A remarkable feature of this measure of Granger causality is that it admits a spectral representation:

$$\mathbf{f}_{x\to y}(\lambda) = \log \left| \frac{S_{xx}(\lambda)}{S_{xx}(\lambda) - H_{xy}(\lambda)\Sigma_{y|x}H_{xy}(\lambda)^*} \right|$$

where S is the cross-power spectral density, and H is the transfer function. Similar to the power spectral density, it can be thought of as a spectral decomposition of the time-domain Granger causality

$$\mathcal{F}_{x\to y} = \frac{1}{2\pi} \int \mathbf{f}_{x\to y}(\lambda)\, d\lambda.$$

The spectral representation can facilitate more efficient computations and a principled spectral analysis of Granger-causality, compared with the more ad hoc approach of band-pass filtering the time series to frequencies of interest and then testing for Granger causality. This can be useful in analysis of LFP, EEG and MEG datasets. A more detailed review of the theory in multivariate Granger causality can be found in Barnett and Seth 2014[22], along with a MATLAB toolbox.

There are many instances where application of Granger-causality must be applied with care. For instance, the method as formulated above can only be applied to stationary time series, a condition that may be hard to meet in some datasets. Toda and Yamamota [264] suggest a correction to Granger-causality in the case that the series are co-integrated [2].

[2] A time series x_t is *integrated* to order d if d applications of the difference operation $(1 - L)^d$ yields a process that is stationary. Cointegrated series are series integrated to an order above zero, for which there exists a linear

3.2.2 Transfer entropy, and generalizations of Granger causality

A more general notion of directed information, based on information theory and known as transfer entropy, is defined as

$$\mathcal{T}_{x\to y} = H(Y_t|Y_{t-1:t-L}) - H(Y_t|Y_{t-1:t-L}, X_{t-1:t-L})$$

where $H(X)$ denotes the Shannon entropy. Transfer entropy can be interpreted as a the mutual information between two processes X_t and Y_t conditioned on Y_t's history. It can thus can be interpreted as the increase in information obtained about Y_t from observing X_t's history over and above information already contained in Y_t's history. This is similar in spirit to Granger causality, and infact, when X_t and Y_t are described by a VAR model, then transfer entropy reduces to Granger causality [21]. More generally, a correspondence can be made between transfer entropy and the log-likelihood ratio statistic of a full and reduced ergodic Markov process, suggesting that transfer entropy is in some sense the extension of Granger-causality to non-Gaussian Markov processes [21] (see also Amblard and Michel 2012 [10]).

A related notion to transfer entropy is directed entropy [10], which is a similar information theoretic quantity measured directed dependencies. It differs from transfer entropy in that it allows for instantaneous coupling between variables. These methods have been extensively applied in neuroscience [271]. For instance [163] and [133].

Alternative extensions to Granger causality are possible. For instance, Chen et al 2004 proposed an extended Granger causality method appropriate for non-linear systems. The idea is simply to apply Granger causality to data within a clusters in which the dynamics may be well approximated as a linear system. These clusters are defined simply in terms of the data points being close to one another. An average of each Granger causality statistic within each cluster may then be taken as a global estimate of extended Granger causality.

combination which is integrated to a lower order. This is an indication that the two series are related to a common, non-stationary input.

An alternative approach is to perform a non-linear transformation to the lagged data in terms of radial basis functions:

$$x_t = \alpha^T \Phi(x_{t-1}^m) + \epsilon_x$$
$$y_t = \beta^T \Psi(y_{t-1}^m) + \epsilon_y$$

A comparison between these methods and standard, linear Granger causality can be found in Lungarella et al 2007 [167]. Other possible extensions include non-linear Granger causality using kernel methods are possible [173].

3.2.3 Point process analogues

One approach to extending Granger causality to point processes is with the notion of *local independence* [67]. This is defined in relation to a *marked point process* (MPP), in which events are assigned both a time and one of a finite number of types. That is, let $\mathcal{E} = \{e_1, \ldots, e_K\}$ denote an event space with $K < \infty$, then the MPP consists of events (t_i, e_i) for $t_i \in \mathcal{T}$, the *time space*, and $e_i \in \mathcal{E}$. Then the counting processes for events of type k are given by

$$N_k(t) = \sum_{s=1}^{\infty} \mathbb{I}(t_s < t; e_s = k), \quad k = 1, \ldots, K$$

defining a multivariate point process. The sigma algebra generated by events of type k, that is

$$\mathcal{F}_t^k = \sigma \left\{ N_k(s) | s \leq t \right\},$$

is the *internal filtration* of counting process N_k. Define

$$\Lambda_k(dt) = \mathbb{E}(N_k(dt) | \mathcal{F}_{t-})$$

then, in the bivariate case ($K = 2$), a process N_1 is said to be *locally independent* of N_2 if Λ_1 is measurable with respect to $\mathcal{F}_t^1$. That is, its immediate, or short-term prediction $\Lambda_1(t)$ does not depend on information besides that which is contained in its own history. Refer to Florens and Fougere 1996 [88] for an alternative extension to continuous time processes, or Gegout-Petit et al 2010[97].

An alternative approach to extending Granger causality to a point process is through Granger causality's spectral representation. A point process also admits a spectral representation which relates to its covariance structure, as with discrete-time stationary processes. An approach adopted by Nedungadi et al 2009 [190] is thus to simply take Geweke's expression for spectral Granger causality, and apply it to the spectra estimated by a point process. This is in some sense an ad hoc approach, as it is not clear that any point process version of Granger causality described above possesses a spectral representation that matches the spectral representation of Granger causality as applied to discrete-time data. A similar approach is adopted by Dhamala 2008 [66].

3.2.4 'Graphical Granger' models

An alternative approach to inferring which nodes of a network are connected to each other is not through hypothesis testing but through regularization. The idea is to pose a network model that is to be estimated with an assumption of sparse connectivity. This has a number of benefits over the Granger-causality estimation procedure described above. In particular, Granger-causality can require the separate estimation of a network model for every putative connection between pairs of nodes to be tested. For large networks this quickly becomes prohibitive. Second, imposing a sparse prior facilitates estimation of high-dimensional networks possessing a large nodes and relatively few observations – the potential $n \ll p$ case. These cases arise often in bioinformatics and in neuroscience, among others.

This is the approach taken by Bolstad et al 2011 [32], applied to a VAR network model:

$$\mathbf{x}_t = \sum_{n=1}^{p} A_n \mathbf{x}_{t-n} + \epsilon_t$$

where A represents coupling between components of $\mathbf{x}$. The resulting optimization problem takes the form

$$\hat{\mathbf{\Theta}}^{(i)}_{GL1} = \text{argmax}_{\mathbf{k},\Psi} \frac{1}{n} \|\mathbf{y} - \mathbf{X}\beta\| - \lambda \sum_{i=1}^{N} \|\beta_i\|_2.$$

This is an example of a Group LASSO method (discussed also in Section 2.4.5). As a generic feature selection method, group LASSO is described by Yuan and Lin 2006 [286]. A number of variation on this basic idea suited to network inference exist. Such methods, and the standard, exhaustive Granger method, fall into the category of 'graphical Granger' models[12]. A comparison of a number of these methods is performed by Haufe et al 2009 [118] and by Shojaie and Michailidis 2010[242].

This idea has been applied to a point process Hawkes model using a group sparse inference of network topology, [202]. Similar ideas have been applied in gene expression networks [166].

3.3 Applications to neuroscience

These methods have been applied extensively in neuroscience, both for networks of spiking neurons (e.g [236]) and more generally. Barnett and Seth 2014 [22] provides a review of Granger causality applied to neuroscience data. It is used, for instance, on continuous data such as fMRI data, EEG data, or MEG data.

An important consideration in neuroscience is that its complexity means assumptions on which connectivity methods are based may not always be satisfied. Thus application to neuroscience data requires significant care. In this regard we note that comparisons between network inference methods are often performed with simulated datasets in which the simulated data matches the model underlying the network inference method (e.g. simulated data from a VAR model as a test of Granger causality – [242]). It is particularly valuable to test the performance of the methods on data generated not from, for example, a VAR, but from a realistic simulation of the phenomena being considered. Work along these lines includes spectral Granger causality being considered from an analytic and simulation-based perspective in relation to networks of neural mass models

by Friston et al 2014[93]. A network of Morris-Lecar neurons and its relation to spectral Granger causality was considered by Boykin et al 2012[34]. Lusch et al [168]considers Granger causality accuracy on a network of coupled Kuramoto oscillators.

When applied to spiking data, a number of Hawkes process models have been proposed to infer connectivity of neurons (e.g.[161, 148, 216]). Hawkes processes have the benefit that they are arguably the most appropriate mathematical framework to model spike trains. They are, by definition, able to capture phenomena at an arbitrarily high temporal resolution, which may be necessary to capture some behavior of neurons. The spike history filter may need to be characterized at quite a high temporal resolution, for instance, in order to fully characterize both the neuron's refractory period and other behaviors such as bursting. Failure to do so can result in a model without a sufficiently strong refractory period, which will be more likely to be unstable in simulation. Thus their discrete-time GLM equivalent of the Hawkes process can sometimes run into stability issues due to discretization errors. However, Hawkes processes can be more difficult to estimate and deal with analytically. Further, since ultimately all data is discretized to some extent, either through binning or through the finite precision of computers, in practice the difference between the two models can be more a quantitative one, not qualitative. Thus Granger-causality and graphical Granger approaches remain more common [149][143][66, 190][124].

Finally we mention that a more direct approach that doesn't rely on any of the above methods is just interpreting the filters of a GLM that represent the connections between neurons (e.g. [208, 144]). This is in some sense the simplest approach, and in some cases can be shown to be more accurate than methods such as Granger causality[144].

4

A GLM of primary motor encoding in a grip and reach motor task

This chapter provides an example of the above encoding and network inference methods. The GLM framework 2.2.5 can be readily extended to network implementations of M neurons [208, 265]. Each neuron is considered to be driven by a filtered stimulus, its own spiking history, and also the filtered activity of the rest of the neurons. If

$$\alpha_i^i(t)$$

for $i, j = 1, \ldots, M$ is the filter acting on the spiking history of neuron j driving neuron i, then the model for the ith neuron is

$$\lambda_t^i = \exp(\sum_k x_{tk}\beta_k + \sum_j \sum_\tau \alpha_{\tau j}^i y_{t-\tau}^j + \mu). \tag{4.1}$$

The incorporation of such network filters has been shown to improve the capability of the model to account for correlations between neurons in a retinal population [208]. While it is tempting to interpret the network filters as capturing, for example, synaptic or dendritic filtering of direct inter-neuronal connections, these terms cannot be taken to imply that two neurons are anatomically connected. For example, correlations might arise from a common input that is not taken into account through the stimulus feature vector [150, 208, 11].

As a demonstration of the above principles we present an example using a brain-computer interface dataset, provided by Engelhard et al [76]. In this study the relation between gamma oscillations, neuronal synchrony and behavior was investigated. In particular, neural oscillations in the low-gamma range (30-50Hz) was used as the basis of a BCI. Monkeys were successfully trained to increase low-gamma power activity to control a cursor to obtain a reward, demonstrating volitional control of LFP oscillations. The increase in low-gamma activity was observed coincident with increases in spiking synchrony. The particular data that we focus on here comes from manual control trials performed before and after the brain control trials, in which the cursor reach task is performed manually by manipulating a robotic arm and gripping handle. This analysis is published in Aljadeff et al 2016 [7], and here is reproduced with permission.

4.1 Behavioral task

The grip-and-reach task involved first moving the cursor to a central position, followed by gripping the handle with sufficient force. Once gripping at the center, after a variable wait time, a target would appear randomly in one of 8 locations. Following another wait of a variable time, the origin would disappear, acting as a go signal, after which the monkey was now able to perform the reach

movement. Grip on the handle had to be maintained through the duration of the trial. A successful trial required reaching the target within a set time limit. Once the target was reached, the monkey needed to hold the cursor at the target for 700ms, and to release its grip on the handle. Following a successful trial, the monkey would receive a reward, and after an inter-trial period the next trial would begin.

4.2 Trial and spiking data

Analysis is based on one ~90 minute recording of a monkey performing the grip-and-reach task. While performing the task, spikes are recorded from 9 single units in the contralateral primary motor cortex, using an intracortical multi electrode array (Blackrock Utah array). Spiking data are binned into millisecond intervals, while both cursor data and grip force are sampled at 100Hz. Figure 4.1 shows example trial data.

4.3 Model

The GLM consists of feature vectors that relate to hand motion, as measured by a cursor trajectory and grip force (Figure 4.1A), that were modeled with Gaussian-bump basis functions. Since motor neurons can encode future motor outputs, the stimulus feature encodes both past and future relationships relative to the current time bin (Figure 4.1B). Similar choices with GLMs have been previously applied to neurons in motor cortex [241, 265, 221]. Lastly, we used raised cosine basis functions (Equation 2.23) for the spike history filters and the coupling filters for the histories of other neurons in the network. The cursor position and grip data vary over hundreds of milliseconds to seconds (Figure 4.1C), while the spike history data vary on the order of milliseconds. Capturing effects on these disparate timescales within the same model requires some care, as the values of the data have a non-Gaussian distribution and are highly temporally correlated; as noted, this correlation can result in uninterpretable high frequencies in the feature vectors. This requires some form of regularization. Here, we used only a limited number of basis vectors

that sparsely sample the stimulus at regular intervals, with the interval size on the order of the stimulus autocorrelation timescale (Figure 4.1C).

The fitting was performed only on data within the movement phases of the trials, i.e., from the beginning of the ‘‘Go signal’’ to the end of ’’Grip pressed’’ (gray bands in Figure 4.1B). In order to avoid unnecessary coupling terms, a group LASSO [286] penalty is applied to the sets of parameters representing connections between neurons. This takes the form (Equation 2.22).

$$\text{argmax}_\theta \left(\log \mathcal{L}(\Theta) - \kappa \sum_{i \neq j}^{M} \|\Theta_{ij}\|^2 \right) \tag{4.2}$$

where $\{\Theta_{ij}\}$ are the parameters representing the coupling from neuron j to neuron i. A similar penalty is applied in prior work [208]. The penalized likelihood is still convex and ensures global convergence.

4.4 Validation

Here the model is validated by splitting the data into a training set representing 80% of the total data and performing cross-validation. A test set representing a contiguous block of 20% of the total data, or 4 min of recording, is used for validation. We take a value $\kappa = 100$ in our network analysis (Equation 4.2); smaller values decreased the log-likelihood while larger values reduced all coupling terms to near zero. We then calculated the predicted rate for the models, used in the log-likelihood estimate (Equation 2.13), by averaging repeated simulations of the GLM given the same stimulus. This validation process was repeated five times, selecting a non-overlapping 20% of the data for testing each time. Combining the likelihood and coherence estimates over all the individual estimates allows the mean likelihood and the standard error of the mean likelihood to be determined for both the coupled and uncoupled models.

Prior work found coupling terms in a network GLM (Equation 4.1) that could be interpreted as functional interaction kernels between cells [208]. In that study, model validation of each neuron was done using the stimulus and the recorded activity of the remainder of the cells. This

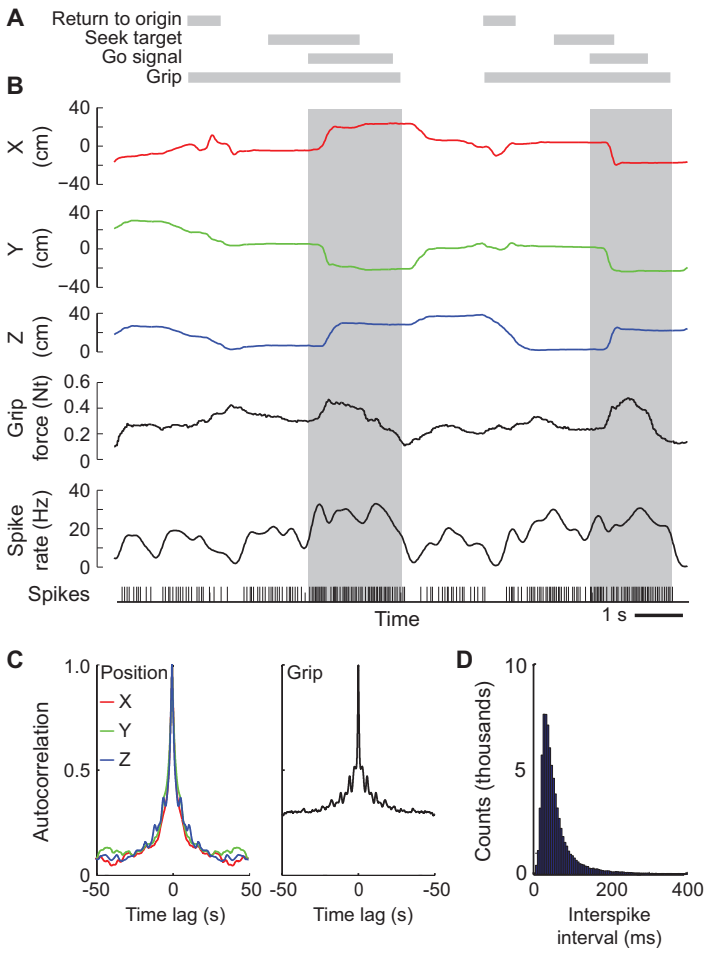


Figure 4.1: Summary of the three-dimensional monkey-based reach task with spike data from example unit. Analysis is based on a single ~90 min recording session in which the monkey performed the task; both cursor motion and grip force are recorded. **A** Grip-and-reach task involves first moving the cursor to a central position, followed by gripping the handle with sufficient force. Once gripping at the center, after a variable wait time, a target appears randomly in one of eight locations. Following another wait of a variable time, the cue at the origin disappears, acting as a go signal, after which the monkey may perform the reach movement. Grip on the handle has to be maintained through the duration of the trial. A successful trial requires reaching the target within a set time limit. Once the target is reached, the monkey needs to hold the cursor at the target for 700 ms and to release its grip on the handle. Following a successful trial, the monkey receives a reward, and after an inter-trial period, the next trial begins. **B** Measured cursor position and grip force. **C** Stimulus autocorrelation. **D** Distribution of inter-spike intervals shows a clear refractory period. Methods: Spikes were recorded from single isolated units in the contralateral cortex to the task arm using an intracortical multi electrode array (Blackrock Utah array) implanted in the arm region of M1. Spiking data were binned into millisecond intervals, while both cursor data and grip force are sampled at 100 Hz. Of the isolated units, we selected those that showed no evidence of contamination based on inspection of the inter-spike interval distribution. Analysis was performed from the time of the Go signal until the grip was released; see gray band in **B**.

procedure is equivalent to fitting a single-cell model where the stimulus is expanded to include the spiking history of the rest of the network. As a practical matter, this procedure has value when one is interested in the precise timing of coupling between cells, e.g. to find whether neurons are anatomically connected [100]. However, expanding the stimulus to encompass the spiking history of the rest of the network stands in contrast to validation of a GLM that represents a network with feedback between neurons, for which the spike histories are based solely on simulations and the only external variable is the stimulus. We use the full network approach in our validation procedure.

4.5 Filters estimation and simulations

With respect to a representative example cell (Figure 4.1), we find that the history filters are the same for the coupled and uncoupled cases (Figure 4.2A), coupling terms are present on a variety of timescales, (Figure 4.2B), and the stimulus feature vectors are altered in magnitude by the coupling (Figure 4.2C). Interestingly, for all cells in the network, the log-likelihood of the model evaluated for the observed spike train shows overall a negligible difference between the coupled and uncoupled models (Figure 4.2D). This is consistent with studies of coupled GLMs applied to retina data[208], in which the addition of coupling terms yields no observable benefit to predicting the average rate given the same stimulus.

More information can be gleaned from the coherence[1] between the predicted rate and the observed spike train than from the log-likelihood. Significant spectral power in both coupled and uncoupled cases only occurred for low frequencies, i.e., 0–1 Hz, and the coupled model had

[1]Coherence measures the relation between two signals in the spectral domain. Between two stationary processes $x(t)$ and $y(t)$ it has the form

$$C_{xy}(\lambda) = \frac{\left\langle \tilde{x}(\lambda)\tilde{y(\lambda)} \right\rangle}{\sqrt{\left\langle |\tilde{x}(\lambda)|^2 \right\rangle \left\langle |\tilde{y}(\lambda)|^2 \right\rangle}}$$

where $\tilde{x}(\lambda)$ and $\tilde{y}(\lambda)$ represent the Fourier transforms of $x(t)$ and $y(t)$. The term $\left\langle \tilde{x}(\lambda)\tilde{y(\lambda)} \right\rangle$ is therefore an estimate of the cross-spectral density, and the denominator terms represent auto-spectral densities of $x(t)$ and $y(t)$. The magnitude of the coherence is a measure of how much one signal follows another at a given frequency. The phase of the coherence is related to the relative lag at which one signal follows another. [7]

a significantly higher coherence in this range for some cells (Figure 4.2E). This increase was statistically significant and particularly strong for our example cell (red ellipse in Figure 4.2F) and one other cell (blue ellipse in Figure 4.2F) and barely significant in three other cells (green ellipses in Figure 4.2F). Thus network interactions through the spike history terms of neighboring cells improve the ability to predict the spike trains for some cells in this dataset.

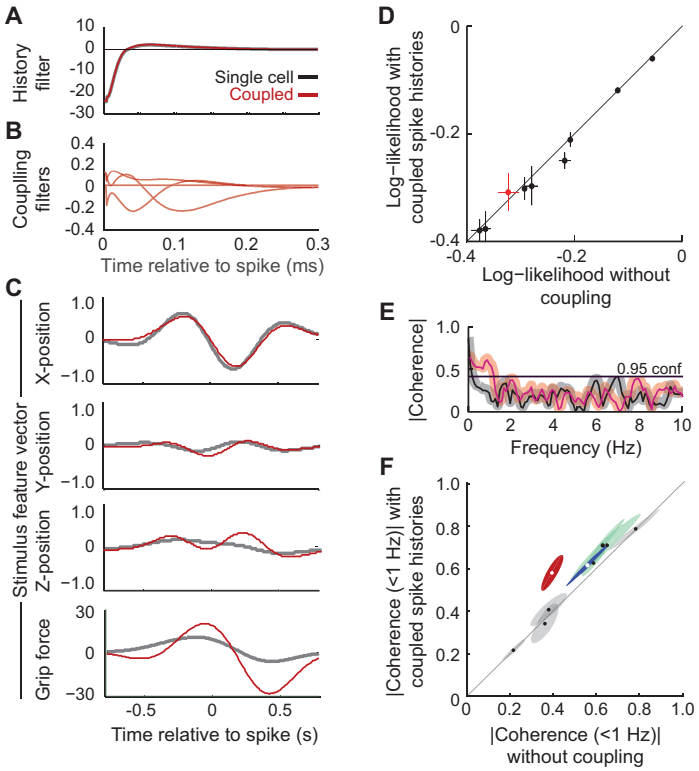


Figure 4.2: Network GLM features and validation for monkey reach data using the interval between the start of the go signal and the end of the trial. **A** Spike history filter for sample unit 36, one of nine concurrently recorded units in our analysis. The nine were chosen as those out of 45 units with no extra spikes in the refractory period of the inter-spike interval. Red curve shows result for the coupled model (k = 100 in 52), and black curve shows the filter in the absence of coupling between units; in this case the two curves are indistinguishable. **B** Spike history filters from eight neighboring cells for the coupled model (k = 100) (Equations 52 and 53). The history filter was expanded in the basis of Equations 46 and 47 using B =5, t0 = 0.005 s, t1 = 0.4 s, and t2 = 2.0 s. The coupling terms are non-zero for three neighbors. **C** Feature vectors calculated for the network, i.e., coupled (red), and single cell, i.e., uncoupled GLM (black). **D** Scatterplot of log-likelihood between predicted spike rate and observed spike train for the coupled and uncoupled model. The red dot refers to the data in **A–C**; the bars are one standard error jack-knife estimates. **E** The spectral coherence, calculated as an average over 100 trials with a 0.5 Hz bandwidth, for the network, i.e., coupled (red), and single cell, i.e., uncoupled GLM (black). The band is one standard error. **F** Scatterplot of the coherence between predicted spike rate and observed spike train for the coupled and uncoupled models. The ellipses are one standard error jack-knife estimates. The red ellipse refers to the data in **A–C** and **E**. The red and blue data are significant at the level of three standard errors (0.01), the green data barely significant at two standard errors (0.05), and the gray data have no significant improvement from coupling to the network.

5

Reconfiguring motor circuits for a joint manual and BCI task

In this chapter we consider an application of the above described encoding and connectivity models to a type of concurrent use, or dual-control brain computer interface (BCI). This work is a collaboration between Chet Moritz, Ivana Milovanovic, Adrienne Fairhall, Eberhard Fetz, and Cooper Mellema. It is being prepared for publication, with a preprint available [154].

Designing BCIs that can be used in conjunction with ongoing motor behavior requires an understanding of how neural activity co-opted for brain-control interacts with existing neural circuits. BCIs may be used to regain lost motor function after stroke or traumatic brain injury, for instance, provided that a neural control signal can be extracted from a single spared hemisphere. Such an approach, however, requires that neural activity controlling the unaffected limb is dissociated from activity controlling the BCI. In this study we investigate how primary motor cortex

accomplishes simultaneous BCI control and motor control in a task that explicitly requires both activities to be driven from the same brain region (i.e. a 'dual control task'). Single unit activity was recorded from intracortical multi-electrode arrays while a non-human primate performed this dual-control BCI task. We observe broad changes in tuning of both units used to drive the BCI directly ('control units') and units that do not directly control the BCI ('non-control units'). Using transfer entropy, a measure of effective connectivity, we observe control-unit-specific dissociation from other units. Through an analysis of variance we find that the intrinsic variability of the control units has a significant effect on task proficiency. In particular, at least one control unit must have a high intrinsic variability to allow cursor control and target acquisition. Surprisingly, the degree of co-tuning or connectivity with other units did not affect dual-control task proficiency. Thus, motor cortical activity is flexible enough to perform novel BCI tasks that require active decoupling of natural associations to wrist motion.

5.1 Introduction

Broad application of brain-computer interfaces (BCIs) for control of neural prostheses requires understanding how brain circuits can simultaneously engage in competing tasks. The complexity of motor cortical neuron representations, however, creates uncertainty about how BCI control could be decoupled from ongoing motor control. Concurrent control of natural movement and BCI output may be useful in cases of brain injury. For example, when the motor cortex of one hemisphere is damaged by stroke or trauma, a BCI-controlled neural prosthetic may be driven by the intact hemisphere. In this case cortical circuits in the intact hemisphere may be tasked with both producing movement of the unaffected limb, while also restoring function to the affected, ipsilateral limb using a BCI. Even in the absence of injury, future industrial applications may seek to control supernumerary artificial limbs or external tools using cortex presumably already involved in normal behavior.

Current state-of-the-art BCI control is achieved via intra-cortical brain signals. Typical BCIs in humans and animals are derived from population coding based on real or imagined movement [61, 103, 3, 125] . BCI studies observe that brain-control mappings which utilize activity observed during the natural motor repertoire are most effective [132, 219]. However utilizing natural mappings between population activity and movement may pose a challenge when BCI control is required to be independent of ongoing motor activity.

A possible approach, then, is to take advantage of biofeedback and conditioning paradigms that allow for volitional control of individual neurons. Studies demonstrate that monkeys can be trained to volitionally control individual neurons in such a fashion [84, 86, 85, 184, 183], with similar results also observed in humans [3, 61]. In these tasks, there is evidence that neural activity can be chosen independently of natural movement association [85, 184, 183, 155]; furthermore, motor cortex cell activity can be operantly dissociated from activity of associated muscles [83], reflecting the remarkable adaptability of motor cortical activity. These findings suggest that next-generation BCI design may be able to better leverage the brain's inherent adaptability.

Most BCI studies to date focus on control of external devices alone. In these tasks, the subjects do not concurrently perform motor actions, thus freeing the activity of neurons or single units controlling the BCI to be modified as needed to perform the task. Indeed, in these cases, continued use of motor cortex cells to control a BCI resulted in a decrease in correlation to movement, suggesting decoupling of the cells from motor control [83, 42]. It is unknown whether and how such a decoupling occurs when natural motor control is engaged concurrently with BCI tasks.

While subpopulations of neurons encoding ipsilateral motion are a candidate control source for the BCI following stroke [237], a robust interface may require recording from the larger population of neurons involved in natural, contralateral control. When a single brain region must simultaneously control a BCI and ongoing limb movement, do neurons strongly associated with contralateral motion make poor control units due to the potential interference imposed by concurrent hand movement? Assuming that the cortical circuits sampled by the recording array

have some causal role in motor control, it seems that at least some sampled neurons could present challenges for simultaneous manual and BCI control.

Previous studies of the activity of units used in control (control units) and the non-control units in the surrounding population (non-control units) reveal widespread changes in activity and tuning preferences in a variety of tasks and task perturbations [258, 42, 156, 134, 94, 95, 109]. Hwang et al 2013 [132] observed changes in preferred direction of both control and non-control units when a BCI decoder's output is rotated, suggesting a global reconfiguration of the motor cortical network. In addition to these changes, other studies showed specific changes for the control units compared with those not involved in the control [155, 95]. How do these unit specific effects manifest in a BCI paradigm where control units may be additionally constrained by their role in ongoing movement? In this study we directly address these questions using a 'dual-control' BCI task. In this task both BCI control and ongoing motor output are driven by the same cortical region, requiring control units to dissociate their activity from the circuits controlling the movement. The task requires that a monkey acquire a target in two dimensions, one dimension controlled by a BCI while the other is controlled using natural motor control of the wrist contralateral to the recorded hemisphere (Figure 5.1).

The dual control task allows us to investigate how primary motor cortical neural activity can be decoupled from natural motion. A previous dual-control BCI study demonstrated that task performance was independent of control units' prior association to wrist motion [181]. Other concurrent-use BCIs demonstrate similar robustness, albeit often with some adaptation period required [197, 23]. Here we extend these studies by recording activity of both control and non-control units during dual control. We examine whether dissociation from wrist motion occurs specifically for control units, and whether this dissociation is associated with changes in tuning and intrinsic variability at the population level. We also consider the *effective connectivity* of neurons in the network, which quantifies the extent to which firing of each neuron can be predicted from that of the others. Using these measures, we compare the motor cortical activity observed during

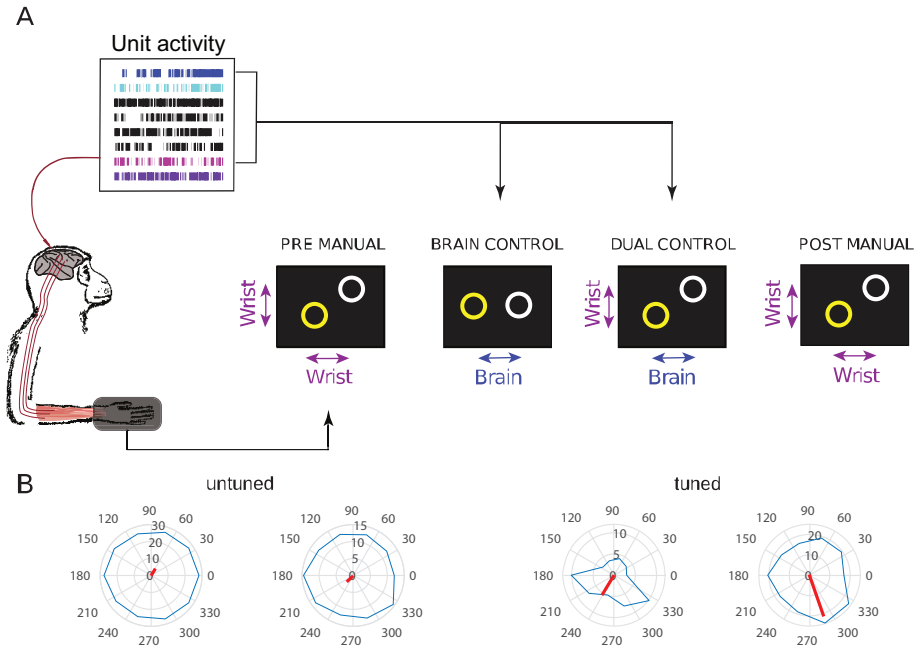


Figure 5.1: Dual control experimental setup. A: Dual control experimental setup. Isolated primary motor unit activity controls a brain-control axis, while contralateral wrist torque determines a manual-control axis. Monkey first performed a 2D manual wrist task. The monkey then performed a 1D brain control task, in which cursor velocity was determined by neural activity of two units. Then the monkey performed a dual control task, where one axis is determined by wrist torque and the other by neural activity. Finally, the monkey performed a second 2D manual task. B: Linear encoding models were used to identify preferred direction and modulation depth (tuning angle and tuning strength) of each unit. Two untuned units shown left have firing rate independent of wrist velocity direction, while two tuned units shown right respond preferentially to wrist velocity in particular directions. Radius indicates mean firing rate (Hz). Red line indicates tuning angle and strength.

the dual-control task to a standard brain-control task which requires the monkey to acquire targets only through BCI control.

Along with the clinical goals for BCI development, studying brain plasticity in the presence of a BCI has the potential to elucidate the nature of motor network function [108]. While traditional approaches propose that motor cortical activity encodes motion kinetics – a representational view of cortical activity– contemporary theories consider motor cortical activity to be better described as a high dimensional interacting system in which is embedded a potentially lower-dimensional dynamical system whose features are related to muscle activities directly [180, 75, 57, 58]. Here, our goal is to characterize as effectively as possible any changes in the population neural activity

underlying a dual-control and non dual-control BCI. Thus, we examine both the properties of traditional representational features such as tuning curves, as well as dynamical features related to effective connectivity and low dimensional manifolds. This allows us to draw on a significant BCI literature to contrast the dual-control BCI paradigm with other paradigms.

To determine baseline tuning, variability and inferred connectivity of the recorded population, the monkey first performed a manual control (MC) target-pursuit task that involved acquiring targets through two degree-of-freedom wrist motion. When compared to this manual control task, we observe widespread changes in tuning of both control and non-control units for the brain-control and dual-control tasks. We find that in the dual control task both tuning to wrist movement and measures of effective connectivity between co-tuned units are more similar to those observed during natural motion. A notable exception is that effective connections to control units significantly decrease during the dual-control task. This suggests that the dual-control task requires control unit-specific dissociation from other units. We further find that the intrinsic variability of the control units has a significant effect on how the task is performed. Specifically, at least one control unit must have a high intrinsic variability to allow cursor control and target acquisition. The degree of co-tuning and strong effective connectivity with other units has no effect on the dual control task proficiency.

5.2 Methods

One macaque nemestrina monkey was trained to perform a random target-pursuit motor task [181]. He began each daily session by controlling the cursor with isometric wrist torque (manual control, MC), then progressed to using the aggregate neural activity of two single units to control a cursor in one dimension (brain-control, BC). Subsequently, he used the same neural activity to control the BCI in one dimension, while simultaneously using isometric wrist torque of the contralateral forelimb to control the cursor in a second orthogonal dimension [181] (dual-control,

DC; Figure 5.1). All procedures were approved by the University of Washington IACUC, and in accordance with National Institutes of Health 'Guide for the Care and Use of Laboratory Animals'.

The monkey was required to move the cursor towards the target and maintain the cursor within the target for at least 1s to receive a reward. Targets appeared randomly, and a 0.5s break was provided between trials. For each trial there was a time-out period of 40s. Each day a new pair of units was chosen to control the BCI. If directionally tuned units were used that day, units with approximately opposite preferred directions were selected and paired to control the BCI. In order to ensure decoupling of unit activity and hand control in the dual control task, visual feedback of one of these modalities had to be rotated to achieve independent degrees of freedom on the monitor. Since the monkey was overtrained on the manual control task, we made a deliberate decision to preserve the relation between wrist torque and cursor movement. Instead, we rotated the visual feedback of the units' firing rate by 90 degrees relative to their preferred direction, even for units with little to no directional preference.

All units with a mean firing rate above 5 Hz recorded from the hemisphere contralateral to arm movement were analyzed. The torque signals were smoothed with a Gaussian kernel of width 300ms and a linear tuning model was fit to each unit's firing rate as a function of wrist torque. Overall performance was quantified as the number of targets per minute acquired throughout the entire recording.

5.2.1 BCI DECODER

The cursor position along the BCI axis, x_t, was determined as a linear function of N control neurons' smoothed activity, y_t^n, $N = 2$, with gain α_n and a running estimate of baseline firing b_n^t:

$$x_t = \sum_{n=1}^{N} \alpha_n (y_t^n - b_t^n) \tag{5.1}$$

where the baseline firing rate is defined as

$$b_t^n = b_{t-1}^n(1-\gamma) + y_t^n\gamma \tag{5.2}$$

The baseline update rate is given by γ ($0 < \gamma < 1$). Spikes were binned at 60Hz. Spike trains were smoothed with a Gaussian kernel of width 100ms.

5.2.2 Linear tuning model

The kinematic encoding model to wrist motion is defined for each unit y^n simply through the linear relation

$$y_t^n = \alpha^1 x_{t+\tau}^1 + \alpha^2 x_{t+\tau}^2 + c + \epsilon \tag{5.3}$$

for velocity along task axes x^1 and x^2, for predetermined time lag τ, and for Gaussian noise ϵ. Here an offset of $\tau = 75$ms was used (Figure S5). Spikes are binned at 25 Hz. This time is consistent with previous studies [241] and was, on average, the time lag of the maximum cross-correlation between cursor and neural activity. For most units, a velocity encoding model was found to yield higher cross-correlation between torque and neural activity than a position encoding model, so the results presented here are based on a velocity encoding model. Velocity is computed from position information using a cubic spline [241].

Note that the model is equivalent to a cosine encoding model through a simple transformation:

$$y_t^n = \alpha r\cos(\theta - \theta_{pref}) + c + \epsilon \tag{5.4}$$

for $\alpha = \sqrt{(\alpha^1)^2 + (\alpha^2)^2}$, $\theta_{pref} = \tan^{-1}(\alpha^2/\alpha^1)$, $r = \sqrt{(x^1)^2 + (x^2)^2}$ and $\theta = \tan^{-1}(x^2/x^1)$. From this it is clear we can interpret α/c as a measure of modulation depth. We will refer to θ_{pref} as tuning angle and α as tuning strength.

5.2.3 Assessing effective connectivity with transfer entropy

Transfer entropy [224] is defined as the conditional mutual information between an observed time series Y_t and the history of a candidate related series X_t, conditional on the history of X_t:

$$H_{X\to Y} = I(Y_t|Y_{t-1}, Y_{t-T}) - I(Y_t|Y_{t-1}, Y_{t-T}, X_{t-1}, X_{t-T}) \tag{5.5}$$

As with many information theoretic quantities, $H_{X\to Y}$ is expensive to compute. An approximation tailored for spiking data is used, in particular the transfer entropy toolbox [133].

The method was validated using a synthetically generated dataset and compared with other common effective connectivity measures. Specifically it was compared to linear Granger causality, Poisson-process Granger causality, and correlation. On the synthetic data transfer entropy was most accurate. More details are provided in the supplementary methods.

Transfer entropy is computed with spikes binned at 5ms intervals. Six minutes of recording data in each condition is used. Connections extend up to 30 time-bins into the past (150ms).

The connectivity analysis is performed as follows. Within each session, between recording conditions (MC→BC and MC→DC), changes in connectivity are compared for different populations of pairs of units. Recalling transfer entropy is a directed measure of connectivity, the populations of pairs of units compared are: to control units and to non-control units; co-tuned units to control units and non-co-tuned units to control units; and co-tuned units to control units and co-tuned units to non-control units. For a given session, generally no more than two pairs of units could be identified within 45 degrees of the unit of interest (e.g. the control unit). Thus in comparisons involving co-tuned units, for each session only the top two co-tuned units were selected. These are defined as the two unit pairs having the closest tuning angle. All other pairs of units in the session count as not co-tuned. In non-co-tuned populations generally many more than two pairs of units can be identified within a session. Thus to ensure these larger populations are better sampled, but still balanced with populations of co-tuned units, the selection of two units is bootstrap sampled 50 times for each session. Such generated populations are combined over

all sessions. All co-tuned populations are selected on the basis of tuning during manual control recordings.

5.2.4 Intrinsic variability and Gaussian-process factor analysis

Gaussian process factor analysis [285] was used to identify a 15 dimensional subspace in the manual control recordings, consistent with previous analyses into the dimensionality of motor cortical data [285, 138]. Data were binned at 2ms intervals. For each recording, 300s of data were used. Of the entire 15-dimensional space identified, the top five dimensions were shown to accommodate for, on average, 98% of neural covariability in the data. The principal angle was identified using the subspace function in MATLAB.

5.2.5 Granger-causality cursor control metric

Given the form of the linear decoding model (Equation 5.1), it is reasonable to model the cursor trajectory as a moving average time series:

$$x_t = \sum_{n=1}^{N} \sum_{p=1}^{P} k_p y_{t-p}^n + c \tag{5.6}$$

for some model order P to be determined. Spikes are binned at 25Hz, and cursor data is smoothed with a cubic spline, as per the linear tuning model [241]. From the linear model we compute the maximum likelihood estimate given both neurons, $\hat{\beta}_{MLE}^n$, and the maximum likelihood estimate not given neuron j, $\hat{\beta}_{MLE}^{n/j}$. The difference between the two log-likelihoods of the MLE models provides a Granger-causality type [22] metric

$$\mathcal{G}_{j \to i} = 2\mathcal{L}(\hat{\beta}_{MLE}^i; \mathbf{x}, \mathbf{y}) - 2\mathcal{L}(\hat{\beta}_{MLE}^{i/j}; \mathbf{x}, \mathbf{y}) \tag{5.7}$$

that quantifies the influence that one neuron has on the cursor trajectory. The likelihood ratio is often used as the basis for hypothesis testing, though the quantity itself can be interpreted as a

form of transfer entropy, provided the data is Gaussian distributed [20]. The Granger causality toolbox MVGC [22] was used to perform this analysis. The approach is well suited since the cursor position is, by definition, a moving average model of the neural data. The method assumes a covariance stationary time series. By construction (Equation 5.1) the cursor trajectory has this property, provided that the neural data is second-order stationary. This was confirmed by checking the spectral radius of the vector autoregressive model was less than one.

To verify the notion that $\mathcal{G}_{j\to i}$ has quantitative meaning we generated synthetic data from neural recordings. A cursor trajectory was generated as the weighted sum of the two units' decoded trajectories, according to (Equation 5.1). Following this we computed $\mathcal{G}_{j\to i}$ for different weightings. A strong monotonic relationship exists between $\mathcal{G}_{j\to i}$ and the weight that determines how much each unit contributed to the cursor trajectories (Figure S4), demonstrating that the control metric can recover which unit contributes more to cursor control.

5.2.6 Analysis

Analysis was performed in MATLAB 2016a. Code is available online at https://github.com/benlansdell/dualbci.

5.3 Results

5.3.1 Dual control task better maintains tuning to wrist motion compared to brain control task

In our previous study [181], we observed that a monkey was able to use all tested pairs of single cortical units to control a BCI task, independently of ongoing contralateral limb movement. Overall performance in this dual-control task was similar, regardless of the control units' directional tuning. During both the brain control and dual control tasks, we measure how units change their tuning to wrist motion and compare this tuning to that observed during the manual control task. Our goal was to understand the overall differences in neural activity that underlie performance of the

brain-control task compared to the dual-control task where wrist motion is also required. We characterize the relationship between unit activity and wrist motion by fitting a simple linear model which provides a measure of a unit's correlation with wrist velocity, given by a tuning angle and tuning strength (see Methods). We then analyzed changes in tuning to wrist motion between different tasks. Compared to the manual control task, we observed significant changes in average tuning angle ($p < 0.001$; two sided t-test, Figure 5.2A,B) and average tuning strength ($p < 0.001$, Figure 5.2C) for units in both the brain control and dual control task. To test the possibility that 90 degrees of visual feedback rotation could by itself result in observed changes in tuning angle, we also conducted a set of trials in which visual feedback was not rotated (Figure S2). In both conditions we observed similar results, suggesting that tuning changes are due to the dual control task rather than the visual rotation.

We asked whether control unit tuning was specifically modified in the two tasks. We found that during the brain control task, 56% of control units and 72% of non-control units did not significantly change their tuning angle compared to the manual control task. However, in dual control, unit tuning angles were more constrained by wrist motion: 79% of control units and 81% of the non-control units retained a similar tuning angle as in manual control (Figure 5.2B). Thus, the significantly greater change in the tuning angle of the control units compared to non-control units in the brain control task (p=0.008, two sided t-test) was not observed in dual control, where tuning angle changed similarly for both types of units (p=0.699, two sided t-test).

To assess whether the tuning changes in previous brain control trials persisted, the monkey performed another manual control task after the dual control task. During this post-manual control session, the majority of units had the same tuning angle as in the first manual control task (correlation coefficient 0.99; Figure 5.2A), suggesting that the network dynamically reconfigured in a reversible way. This reversibility is similar to that seen in previous BCI studies [95].

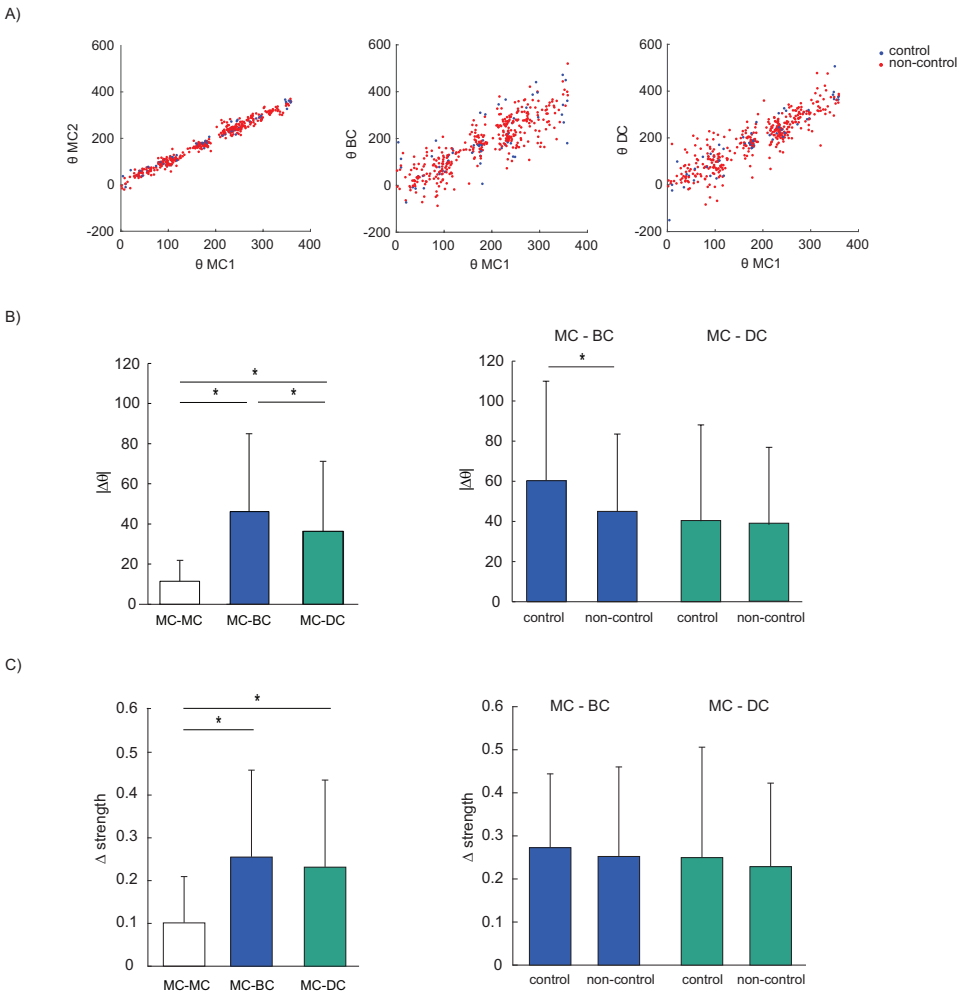


Figure 5.2: Changes in encoding between dual control and brain control tasks. A: Change in each unit's tuning angle (θ) between conditions. Each point represents one unit within one set of recordings. All angles on the y axis shifted to within $\pm$ 180 degrees of x axis value. Plot shows 69 days of recordings, with a total of 597 units in all sessions and an average of 8.652 units per session. Only units tuned, with a R^2 above 0.01, were included. Compared to the manual control task (MC1), the population of units showed significant changes in their tuning angle in both the brain (BC) and dual control task (DC). Unit tuning angle is more constrained by wrist motion in dual control than in brain control task. During the second manual control session (MC2), the majority of units reverted back to their original tuning angle in MC1. B: (left) Change in tuning angle (θ) for all units between conditions. Units changed tuning angle significantly between brain control and manual control and between dual control and manual control task (Two sided t-test; both $p \ll 0.001$). (right) Change in tuning angle for control and non-control units between conditions. Units controlling the BCI change their tuning angle significantly more than non-control units during BC (t-test; $p = 0.008$). In the dual control task, BCI units did not change their tuning angle significantly more than non-BCI units ($p = 0.699$). All values are mean $\pm$ standard deviation. C: (left) Change in tuning strength for all units between conditions. Compared to the manual control task (MC1), the population of units showed significant changes in their tuning strength in both the BC and DC task ($p \ll 0.001$). (right) Tuning strength changes similarly for control and non-control units in both tasks.

5.3.2 Dual control task selectively dissociates control unit activity from co-tuned units

We next considered how the relationships between pairs of units depended on the task. To evaluate this, we used a measure of effective connectivity, transfer entropy. The transfer entropy of two random processes, from X_t to Y_t, measures the information about Y_t obtained by observing the history of X_t, conditioned on the history of Y_t (see Methods). In this paper we use the term effective connectivity to mean a directed relationship in which one neuron's spike train is useful in predicting another's [193, 247]. We note that, though directed, transfer entropy is purely a statistical relationship and does not imply any synaptic or anatomical connection.

We investigated the relation between control units and co-tuned unit activity during the dual control task. As described above, in dual control most units maintain their association with wrist motion observed in manual control. However, the control units are required to modulate their activity independently of this association. Thus we might expect that they will subsequently exhibit a weaker relation with units with which they were previously co-tuned, and that this will manifest as a change in the transfer entropy between control and co-tuned units. According to this hypothesis, the dual-control task requires control unit activity to become specifically less dependent on co-tuned neural activity. We therefore used a directed measure of connectivity to measure this dependence.

We focused on changes in connectivity that occurred between manual control and brain or dual control tasks for different subpopulations of unit pairs (see Methods). Overall, we observed a widespread change in the transfer entropy between manual control and the brain and dual control conditions. This change was smallest between the pre- and post- manual conditions ($R^2 = 0.420$, with the brain and dual control conditions showing larger changes in transfer entropy (Figure 5.3A). The changes for both brain control ($R^2 = 0.212$) and dual control ($R^2 = 0.221$) were similar (Figure 5.3A).

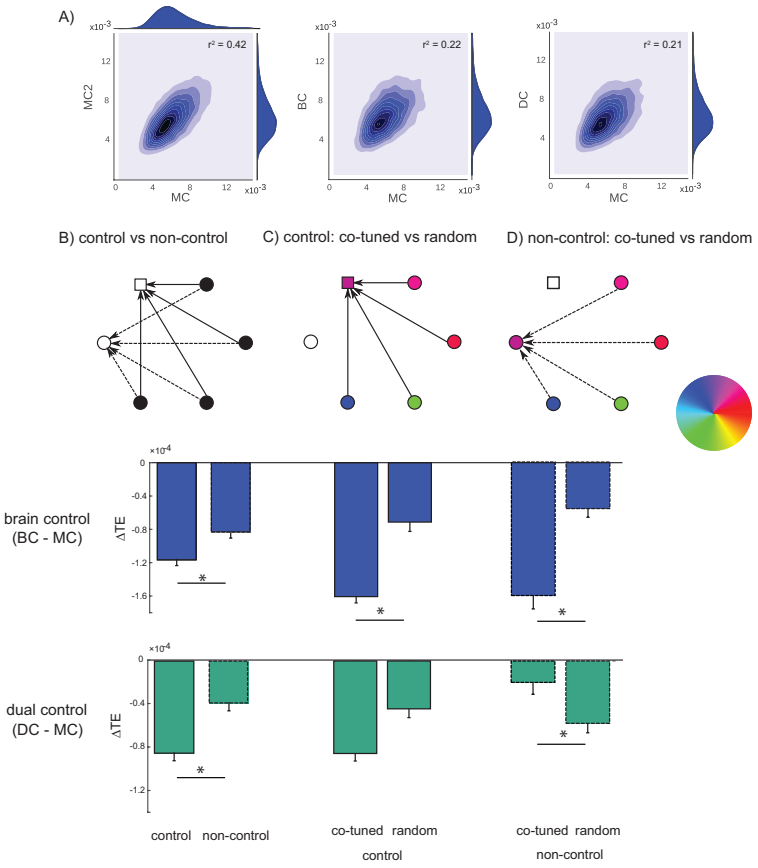


Figure 5.3: Connectivity analysis using transfer entropy. A: Contour density plot showing change in transfer entropy between tasks. Changes between first and second manual control tasks $R^2 = 0.423$, manual control to brain control $R^2 = 0.212$, manual control to dual control $R^2 = 0.221$. B: Top: Cartoon demonstrating comparisons in change in transfer entropy between tasks for an example, six unit network with one control unit. Square indicates control unit, circles indicate non-control units. Here effective connectivity to control (solid arrows) and to non-control units (dashed arrowed) are compared. Bottom: connectivity changes significantly more between manual and brain control task for control units than non-control units ($p \ll 0.001$ [Wilcoxon rank sum]); connectivity also changes significantly more between manual and dual control task for control vs non-control units ($p \ll 0.001$ [rank sum]). All changes are significantly different from zero (all $p \ll 0.001$ [signed rank test]) C: Change in transfer entropy to control units between tasks for both co-tuned units and randomly selected units. Here color indicates tuning angle, per the color wheel to right. Connectivity to the control units changed significantly more between manual and brain control task for co-tuned units than for randomly selected units ($p \ll 0.001$ [rank sum]); connectivity to the control units does not change significantly more between manual and dual control task for co-tuned units vs randomly selected units ($p = 0.528$ [rank sum]). All changes were significantly different from zero ($p \ll 0.001$ [signed rank test]) D: Change in transfer entropy to non-control units between tasks for both co-tuned units and randomly selected units. Connectivity changed significantly more between manual and brain control task for co-tuned units than for randomly selected units ($p \ll 0.001$ [rank sum]); connectivity changes significantly between manual and dual control task for randomly selected units compared with co-tuned units ($p \ll 0.001$ [rank sum]). Changes for co-tuned units between manual and dual control were not significantly different from zero ($p = 0.145$ [signed rank test]).

We then compared the change in connectivity for control units versus non-control units. We observed that between brain and manual control, control units decreased their connectivity significantly more than non-control units. We observed the same trend when comparing changes in connectivity between dual and manual control (Figure 5.3B). Thus, in both brain and dual control tasks, the control units become more independent from other units than non-control units. In other words, both tasks result in some control unit-specific dissociation.

We wondered if this control unit dissociation was specific to units that were co-tuned with the control unit. Indeed, comparing changes in connectivity to the control unit between brain and manual control showed a significant difference in the connectivity change for previously co-tuned units versus randomly selected units ($p \ll 0.001$ [rank sum]; Figure 5.3C). However, comparing such changes in connectivity to the control unit between dual control and manual control does not reveal a significant difference ($p = 0.529$ [rank sum]). Thus, control units become more independent from co-tuned units than from other units in the brain control task, but not in the dual control task.

This result is counterintuitive, given that the structure of the dual-control task was designed to effect the opposite result – control unit specific independence. The possibility remains, however, that the brain control task leads to a higher degree of change in co-tuned units in general, and that the effect has nothing to do with whether or not a control unit is involved. Thus, we also examined changes in connectivity to non-control units, from either co-tuned units or from non-co-tuned units (Figure 5.3D). In the brain control task, connectivity decreases significantly more for co-tuned units than for non-co-tuned units. This is similar to the result obtained in Figure 5.3C for control units. In dual control, however, connectivity changes significantly less for co-tuned units than for non-co-tuned units. In fact, between manual and dual control the overall change in connectivity for co-tuned units to non-control units is not significantly different from zero ($p = 0.145$). Thus, in dual control transfer entropy between co-tuned units is maintained, unless one of the units is a control unit. In brain control, however, there is an overall decrease in connectivity for co-tuned units, independently of whether a control unit is involved.

This result is consistent with the tuning angle findings of the previous section, in which greater changes in tuning angle were observed in brain control compared to the changes in dual control (Figure 5.2B). Together these findings suggest that the manual component of the dual-control task holds both the tuning angles and connectivities between co-tuned units in similar states to that observed in manual control, with the notable exception of connectivities involving the control units.

5.3.3 Neural variability constrains brain control proficiency

Despite the insight that the connectivity and tuning analysis provide into the neural activity that underlies the dual control task, neither analysis was predictive of task proficiency. This is consistent with previous studies showing remarkable flexibility of motor cortical activity. Previous studies also demonstrate that neural variability measured during natural behavior – intrinsic variability – is a constraint on learning and proficiency of brain control tasks [219]. Thus we investigated the role of intrinsic variability of the control neurons in relation to the variability of the entire neural population.

We began by identifying a low-dimensional space that characterizes a high proportion of the variance observed during natural movements from the manual control recordings. We used Gaussian process factor analysis (GPFA) to identify this space, which we term the intrinsic space (similar to the intrinsic manifold proposed by Sadtler et al 2014 [219]). In line with this terminology, we term the variance of each recorded unit during the manual control task its intrinsic variability. When a brain control task is performed, the activity from just two neurons determines the cursor position. The activity of these two neurons thus comprises a brain-control space (Figure 5.4A). Based on Sadtler et al 2014 [219], we hypothesized that more proficient brain-control and dual-control would occur when the brain-control space and the intrinsic space share some amount of alignment. We observed that in the brain control task, but not in the dual control task, performance is significantly higher when the brain-control and intrinsic spaces are indeed non-orthogonal (Figure 5.4B; brain-control $p = 0.023$, dual-control $p = 0.13$ – two sided t-test).

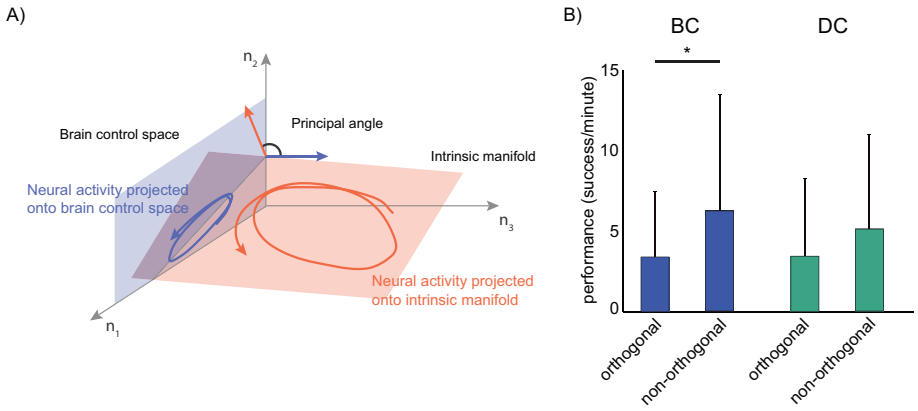


Figure 5.4: Intrinsic variability constrains performance. A: Cartoon demonstrating variability analysis. For simplicity here we assume only three neurons are recorded. At each recorded time bin, we collect three observations, the number of spikes occurring within the time bin of each of the recorded neurons, and place this point within a three dimensional 'neural activity space'. In this way we collect a cloud of points representing the activity of all neurons throughout the entire recording. We identify axes which represent a lower-dimensional subspace which accounts for the largest proportion of the variance of the data. Here Gaussian Process Factor Analysis (GPFA) is used to identify this subspace, and is used to identify such a low-dimensional space in conjunction with an assumption of temporal continuity between successive recordings of each neuron. This space is called the 'intrinsic manifold', and here is constructed based on manual control recordings. Units chosen for brain control and dual control tasks provide an n-dimensional subspace, where n is the number of neurons used in the decoder (here $n = 2$). The alignment between the brain-control space and the intrinsic space is the 'principal angle'. For clarity, only the neural trajectories projected onto the two low-dimensional spaces are shown, not the full-dimensional neural trajectory. B: Alignment between brain-control space and intrinsic manifold relates to performance in brain control and dual control. In brain control, performance is significantly higher when the two spaces are non-orthogonal, i.e. when there is some degree of overlap between the two spaces. In dual control, the same trend exists, though it is not significant ($p = 0.13$; two sided t-test).

Intrinsic variability of control units affects contributions to cursor control and task proficiency

Given this result, we sought further insight into how the task was being performed by focusing specifically on the intrinsic variability of the control units. In both the brain control and dual control tasks, the decoder (Eq 5.1) is set up so that neurons which have a high variance exert more effect on the cursor trajectory. We hypothesized that a high amount of variance is necessary for control over the cursor. If this is indeed the case then, when control units are selected with lower intrinsic variability, the monkey is required to increase unit variance in order to accurately perform the target pursuit task. If the monkey is unable to change the control units' variance, then we might expect that in these cases he will be less able to control the cursor and will acquire targets at a lower rate.

To measure each unit's contribution to the target acquisition task, we constructed a metric which quantifies the amount of influence each control unit has over the cursor trajectory. Since the cursor trajectory was determined through a moving average model weighing the contribution from each control unit, we selected a Granger causality-type metric (see Methods). We validated this metric using synthetic cursor data generated through a weighted contribution of two neurons (Figure S4). We observed that a high value of this control metric for at least one of the control neurons is required for dual control task proficiency (Figure 5.5A).

In addition, we observed that a high control metric is related to a high intrinsic variability of the control unit (Figure 5.5B). This indicates that the task is predominantly controlled by the unit with higher intrinsic variation. As suggested by the relation to cursor control and performance, there is a significant, positive relation between control unit intrinsic variability and performance, in both brain control (not shown) and dual control (Figure 5.5C).

These results suggest that a control unit's intrinsic variability is a harder constraint on both brain-control and dual-control task performance than either its tuning or its transfer entropy with

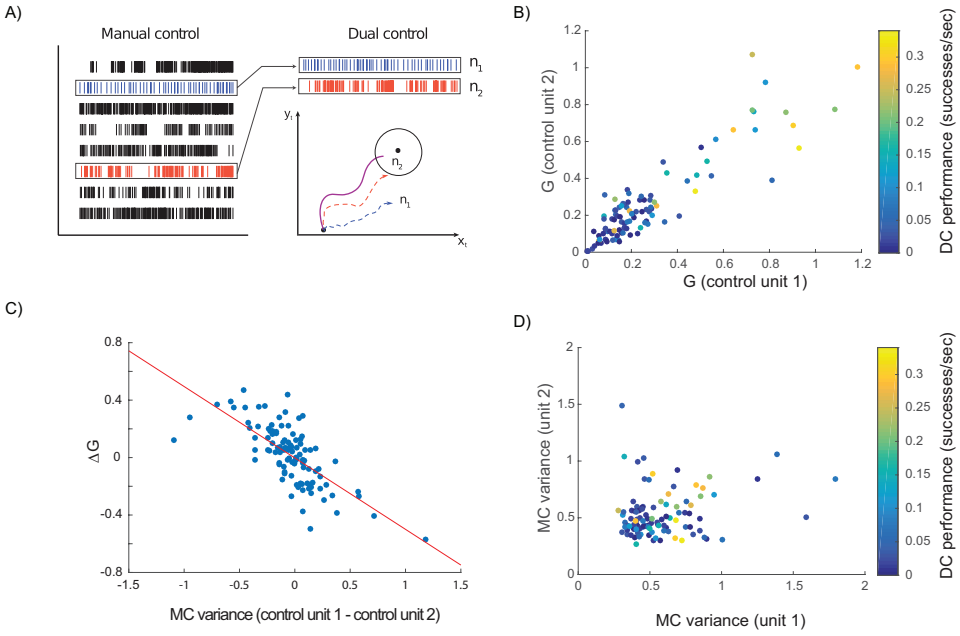


Figure 5.5: Contributions of cursor motion by each control unit, measured by Granger causality score $\mathcal{G}$. A: Control units chosen based on their tuning and activity in manual control recording (left panel) exert different levels of control over cursor in dual control recording (right panel). Dashed lines represent cursor motion contribution from individual control units, while solid purple line represents their aggregate. B: High levels of control positively correlated with high performance indicated by warmer colors clustering to the right (F-test, $p \ll 0.001$). C: Granger causality analysis identifies extent that individual control units exert control over cursor trajectory. The relative variance of the control units in manual control (MC) significantly influences the amount of control that each unit exerts over the cursor trajectory in dual control (DC), as identified by the Granger causality analysis ($p \ll 0.001$, $R^2 = 0.500$). D: The variance of control units observed during manual control is significantly positively correlated with performance in the dual control task. (F-test, $p = 0.016$).

other units. Therefore, when selecting units to control a BCI, intrinsic variability should be given greater consideration than tuning or connectivity to other units.

5.4 Discussion

In this study we sought to understand the factors that affect dual-control BCI proficiency and that may inform control unit choice in concurrent BCI tasks. Together our analyses suggest that factors such as effective connectivity and control unit's tuning strength and direction have minimal impact on dual control performance, while the unit's intrinsic variability has a significant effect.

5.4.1 Differences between linear tuning in brain control and dual control

In both brain and dual control contexts we observed a population-wide change of tuning angles between conditions, compared to tuning to naturalistic motion. These changes are suggestive of a global reconfiguration of the motor cortical network for different motor tasks, as seen in many previous BCI studies [258, 42, 156, 134, 94, 95, 109].

We also observed that tuning angles are significantly different between brain control and dual control. This finding is similar to observations made in a previous concurrent control study [197]. The authors found that producing static force with the arm during the BCI task perturbs the neural map formed during the standard BCI task. Interestingly, we found that during dual control, tuning angles were more similar to tuning observed during natural motion. Consistent with this finding, Suminski et al [255] reported that when the arm is passively moved in concert with a brain-controlled cursor, decoding performance improved. This suggests that tuning curves change less between manual and brain control conditions when the arm is actively involved in the task. It is possible that proprioceptive feedback inherent in arm movement is responsible for driving uniformly distributed tuning angles aligned with arm movement directions.

In contrast to the brain control task, control-unit-specific changes in tuning were not observed in the dual control task. Greater modification of control units' tuning angles in the brain control task appears to be consistent with results from other studies in which visuomotor rotation was applied to a subset of units controlling the cursor [134, 50]. Applied visuomotor rotation resulted in the rotated subpopulation shifting their tuning angles more than the non-rotated population. Conversely, in the dual control task, tuning angles changed similarly for both control and non-control units, as observed in [132]. One explanation for this difference is that the manual component of the dual control task constrains neural activity in such a way that dissociation can only be achieved at a broader population level, or that the increased task complexity requires more coordinated activity of neurons in the population.

Previous BCI studies additionally show differential tuning strength of control versus non-control units [95, 155], while here we report that both control and non-control units undergo similar changes in tuning strength in the brain control and dual control task. Interestingly, Ganguly et al. observed no differential tuning strength of control and non-control groups of units during the initial learning process, but only after 2-3 days of performance with the same units. Similarly, Law et al [155] saw a stronger effect in late compared to early trials. This suggests that the short amount of practice time in the present study for each pair of control units may be insufficient to affect modulation depth.

5.4.2 Insights from effective connectivity

We sought a better understanding of the changes in network activity underlying the dual control task compared with the brain control task. We observed a selective decrease in effective connectivity between co-tuned units and the control unit in dual control, and a non-selective decrease in connectivity between co-tuned units in brain control. This suggests that the dissociation required in the dual control task does affect interactions between individual units in the network.

Effective connectivity, sometimes also referred to as functional connectivity, has been used to study properties of motor cortical networks using directed information in [212, 51] and the

Granger causality approach of [143]. In this study we chose to use transfer entropy as a non-parametric measure of effective connectivity, as it performed best on a synthetic validation dataset. A drawback of all these methods is that they are unable to distinguish between direct interactions and the effect of hidden, or latent inputs to recorded units. Given the 400 μm electrode separation of the Blackrock Utah array used here, direct synaptic connections between units are unlikely [244] and all relations between units may be mediated through unobserved connections. A more appropriate model then may be one that separates latent and direct connectivity components [206, 230, 171]. Such an analysis is the focus of future work.

5.4.3 Intrinsic variability constraints

In the brain control task we found that performance is constrained by each unit's variability, and how well aligned the brain control space is to the units' natural firing patterns. Other studies have also reported that factors relating to motor tuning may not be the most relevant when selecting BCI control units. For instance, Yu et al 2009 showed that BCI performance was better when low-dimensional latent dynamics were inferred and utilized, independent of any kind of external movement or task parameter [285]. Another study showed that trial-by-trial spike predictions based only on the firing rates of simultaneously recorded motor cortex neurons tend to outperform predictions based on external parameters [252].

Constraints provided by intrinsic neural variability have been identified before [219, 132] in populations of 10s to near 100 units. In this study the 'control space' is only two dimensional, whereas previous studies have explored the concept in wider control spaces consisting of 10-100 units. The low-dimensionality of the present study may facilitate insight into these larger control spaces. In Sadtler et al 2014 [219], for instance, within-manifold perturbations were less detrimental to performance than 'outside manifold' perturbations. In the present study the 'control space' is deliberately limited to two dimensions to facilitate insight into how a focused population of neurons must change their activity to accomplish the dual control task. It is possible that the role of variability of control units also explains these previous results. That is, the intrinsic

manifolds are predominantly defined by axes parallel to highly varying/firing units, and it is these units that dominate the manifold, not linear combinations of units. In other words, correlations between units may not be significant; within/outside manifold perturbations become simply the inclusion or otherwise of varying units that are able to contribute to the control of the BCI. This possibility requires further study.

5.4.4 Outlook

Our results demonstrate that dissociation of motor cortical units required by a dual control task does not occur with individual neuron specificity, but through task-related changes in both the control and non-control populations. This and previous work [219, 132] suggests that, provided specific constraints are taken into account, the motor cortex can flexibly adapt to a variety of challenging control tasks. Tasks that require dissociations from established correlations between neural activity and movement can be learned even over the course of a short recording session. Alongside their clinical value, BCI paradigms provide insight into the physiological principles that guide motor control [108] by allowing the relevant populations for motor control and learning to be directly observed. Despite recent findings, a lack of consensus remains over the appropriate framework to understand primary motor cortical activity. In the present study we relied on a variety of analyses utilized in both representational and dynamical perspectives of encoding in motor cortex. We found that each analysis revealed different insights into how primary motor cortex learns a novel coordination task.

5.5 Supporting information

S1 Appendix. Validating connectivity methods. There are a number of candidate measures of effective or functional connectivity between units, including Granger causality, transfer entropy and correlation. Undirected, statistical relationships, may be referred to instead as functional connectivity [193, 247]. Before applying these analyses to brain-control recording data we sought

to understand each method's success in identifying effective/functional connectivity in a synthetic dataset with known connectivity. For this, we generated a dataset using a point process model [152]. This model was designed to simulate an ensemble of cortical neurons in motor cortex, and has been used previously for making predictions about brain-BCI interactions. The candidate methods were standard covariance, Granger causality analysis based on a Poisson process GLM [143], Granger causality analysis based on a linear firing rate model, and transfer entropy [133]. Details for each method are below.

A simulated spiking network [152] of 30 Izhikevich neurons is divided into 3 densely interconnected groups with some inter-group connections. As common input, the neurons are stimulated with a rectified sinusoidal input. The input was a 2 Hz sinusoidal envelope with a 100 Hz envelope amplitude. The simulation was run for 3 min, with a baseline firing rate of 1 Hz (producing on average 17 Hz activity in simulation).

For our Transfer entropy analysis, we used the Transfer Entropy Toolbox [133] with a 1ms bin size, a max delay of 20 ms, 1-3 time bins and 1-3 past time bins to calculate the transfer entropy between neurons. We then normalized the Transfer entropy by z-scoring.

For our linear Granger causality analysis, using MLE we fit a linear model of the activity of all neurons 1-20 ms in the past (including the past history of the unit being predicted) in order to predict the activity of unit i. We then removed unit j from the model and re-fit the model. The change in deviance (normalized by z-scoring) between the old and new model was used as our measure of the effective connectivity between units. For our GLM granger causality analysis, we performed the same analysis with an Poisson model with an exponential nonlinearity rather than a linear model, again normalizing our results by z-scoring.

We also calculated the covariance between units and normalized by z-scoring.

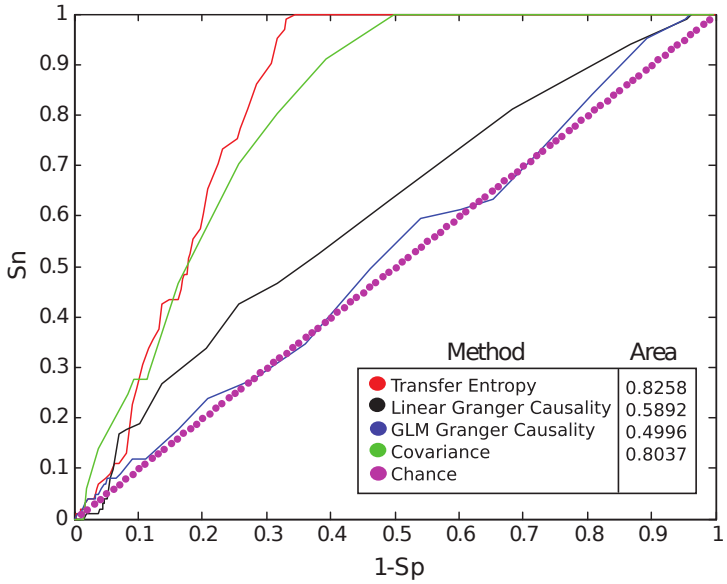


Figure 5.6: Functional connectivity validation.

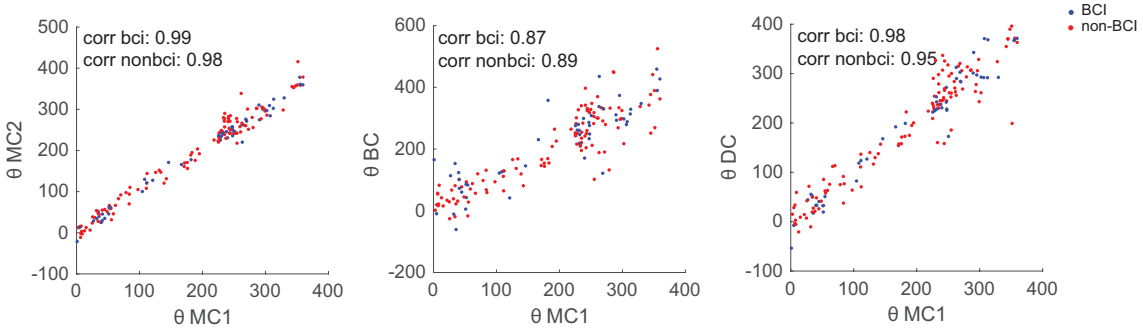


Figure 5.7: Change in tuning angle in unrotated sessions. Change in tuning angle between conditions for control units that were displayed in their preferred direction on the screen, rather than rotating by 90 degrees as in the remainder of the study to permit dual control. Changes are similar to those observed in the 90 degree rotated dataset (Figure 5.2) – overall larger between manual control and brain control tasks than manual control and dual control tasks.

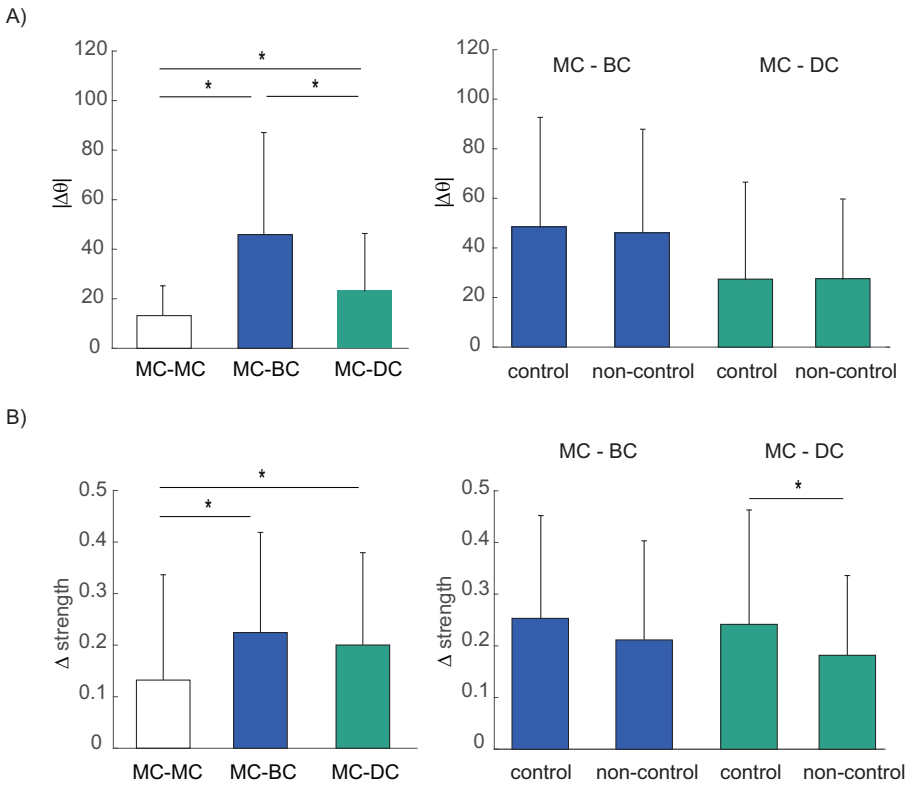


Figure 5.8: Tuning changes during unrotated sessions. A: Tuning angle. Left: all p $\ll 0.001$ [two sided t-tests]. Right: non-significant. B: Tuning strength. Left: MC-BC $p \ll 0.001$, MC-DC $p = 0.0016$ [two sided t-tests] Right: MC-DC $p = 0.0491$ [two sided t-test]

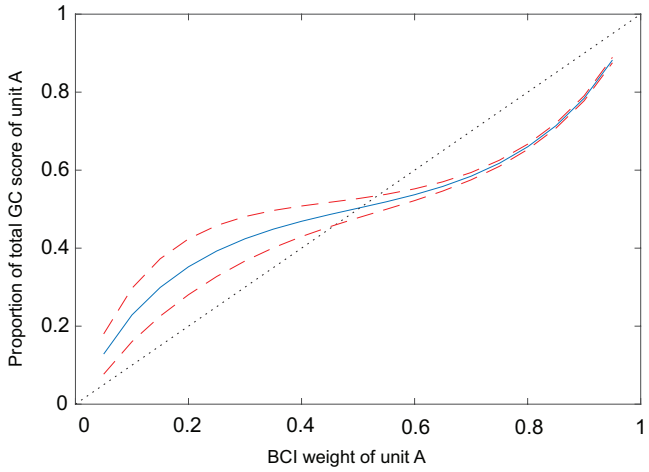


Figure 5.9: Validation of Granger-causality cursor control metric. Synthetic cursor trajectories are generated according to neural data from two control units, according to the decoder (5.1). Contribution weights for each units are varied. 108 trajectories are generated for each weighting and cursor control metric, $\mathcal{G}_{j\to i}$, is computed. Blue curve indicates mean proportion of control score as a function of weight; red curve indicates standard deviation; dotted black line indicates identity line.

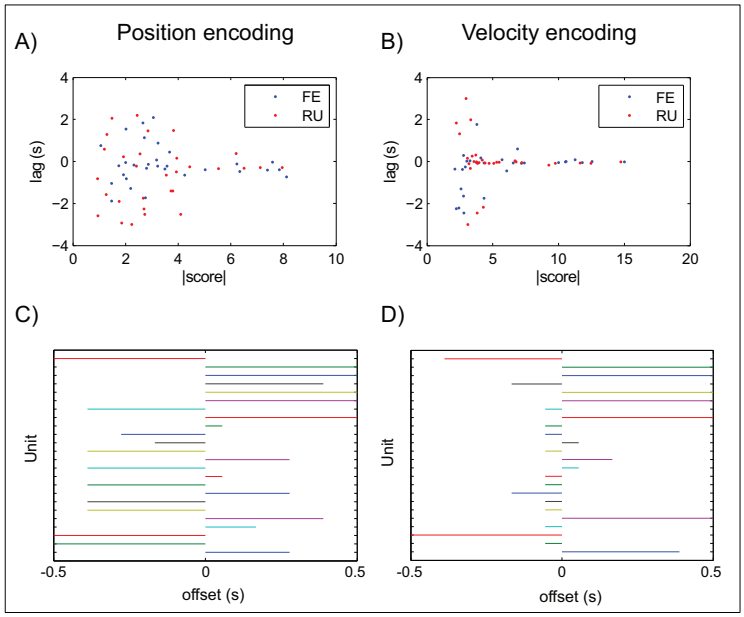


Figure 5.10: Optimal time lags for simple tuning models. Peak of cross-correlation between filtered firing and position flexion-extension (FE) and radial-ulnar (RU) torque output gives an optimal encoding lag for each channel. The absolute value of the z-scored peak provides a simple encoding 'strength'. A: The relation between score and optimal lag for direct FE and RU output – related to cursor positions. B: the same relation for time-differenced torque output – related to cursor velocity. Scores below 4 are wildly varying in optimal lag, while scores above 4 are more tightly constrained to near zero. The 'velocity encoding' responses is optimal near a lag of zero. Similarly, fitting a simple GLM Poisson model with a stimulus filter of length 1 (only one time point) for different time point values queries at what time delay the neuron's firing is most informative about the stimulus. The change in each model's likelihood when the stimulus filter is included gives an indication of the encoding at this time lag. The optimal time lag can be computed by taking the maximum in this change in likelihood. C: The optimal time lag for a set of units under a position encoding model. D: Optimal time lag for a set of units under a velocity encoding model. The most common optimal time lag was approximately -100ms – indicating spiking occurs before the motor output. As a compromise between these two sets of analyses, a fixed time lag of -75ms was used in the main text.

6

A reaction-diffusion model of cholinergic retinal waves

In this chapter we develop a model of spontaneous activity in the developing nervous system. This work is a collaboration with Kevin Ford and Nathan Kutz, and has been published in Lansdell et al 2014[153]. Here it is reproduced.

Prior to receiving visual stimuli, spontaneous, correlated activity in the retina, called retinal waves, drives activity-dependent developmental programs. Early-stage waves mediated by acetylcholine (ACh) manifest as slow, spreading bursts of action potentials. They are believed to be initiated by the spontaneous firing of Starburst Amacrine Cells (SACs), whose dense, recurrent connectivity then propagates this activity laterally. Their inter-wave interval and shifting wave boundaries are the result of the slow after-hyperpolarization of the SACs creating an evolving mosaic of recruitable and refractory cells, which can and cannot participate in waves, respec-

tively. Recent evidence suggests that cholinergic waves may be modulated by the extracellular concentration of ACh. Here, we construct a simplified, biophysically consistent, reaction-diffusion model of cholinergic retinal waves capable of recapitulating wave dynamics observed in mice retina recordings. The dense, recurrent connectivity of SACs is modeled through local, excitatory coupling occurring via the volume release and diffusion of ACh. In addition to simulation, we are thus able to use non-linear wave theory to connect wave features to underlying physiological parameters, making the model useful in determining appropriate pharmacological manipulations to experimentally produce waves of a prescribed spatiotemporal character. The model is used to determine how ACh mediated connectivity may modulate wave activity, and how parameters such as the spontaneous activation rate and sAHP refractory period contribute to critical wave size variability.

6.1 Introduction

Throughout the nervous system, correlated spontaneous activity drives developmental programs [182, 30]. Within the retina, these events manifest as slow, spreading waves of depolarizations and are thus termed retinal waves. Waves are observed in a variety of species and progress through three stages of development [276, 91]. They have been implicated in numerous developmental processes, including retinotopic map refinement [130] and the eye-specific segregation of retinal projections into layers of the thalamus [281]. In mammals, waves mediated by acetylcholine (cholinergic waves, or stage II waves) are the best characterized. They exhibit a slow wavefront velocity, random initiation site and direction of propagation, an interwave interval (IWI) which lasts tens of seconds, and constantly shifting wave boundaries. Precisely determining their role in development requires a controlled manipulation of these properties, which in turn requires a sound theoretical understanding of the mechanisms responsible for their generation. However, the complexity of their dynamics means that determining the connection between the generation and resulting spatiotemporal patterns of wave activity and underlying physiology is by no means

intuitive and is reliant on computational modeling. This paper develops a biophysically consistent, yet simplified, conductance based model of the developing retina that is able to produce physiological waves, in order to elucidate this connection.

Retinal waves are believed to be initiated by spontaneous depolarizations of Starburst amacrine cells (SACs) whose processes reside in the inner plexiform layer (IPL) of the retina. In mice these spontaneous depolarizations are sparse: they occur roughly once every 15 minutes per SAC [92], where as in rabbit the rate is closer to once every 30 seconds. Following initiation, the dense, recurrent connectivity of the SAC network laterally propagates activity through cholinergic synapses [287]. After depolarizing, SACs exhibit a slow after-hyperpolarization current due to a cyclic-AMP sensitive, calcium-activated potassium channel [92, 288], which persists for tens of seconds. Consequently, waves propagate over only a finite domain of the retina, with their boundaries determined by regions still refractory from previous wave activity. A number of computational studies have tested this wave generation hypothesis, notably the models by Godfrey *et al* 2007 [107], Hennig *et al* 2009 [121] and subsequent study by Ford *et al* 2012 [92] (refer to the reviews Gjorgjieva and Eglen 2011 [104] and Godfrey and Eglen 2009 [105] for more information on previous computational studies).

A striking feature of retinal waves is their variety of sizes, speeds and shapes, which differs from the more stereotyped behavior of spontaneous activity in other developing brain regions, for example in neocortex [2, 9]. Despite numerous theoretical studies, it is not clear what properties of the recurrent developing SAC network most contribute to these spatiotemporal patterns. Indeed, retinal waves sizes appear to be distributed according to a power-law[121], analogous to other examples of spontaneous activity in the nervous system. In cortex, such activity has been extensively studied as an example of a critical state phase-transition [26]. Additionally, diffuse release of ACh has been detected at the inner limiting membrane coincident with wave activity [92], suggesting that, like later stage waves mediated by glutamate [31], extra-synaptic diffusion may play a role in cholinergic wave propagation, though this has not been tested theoretically or experimentally.

In this work, we develop a theoretical, reaction-diffusion framework that integrates the key biophysical processes, including increased excitation due to acetylcholine diffusion and the slow after hyper-polarization of the SACs, to characterize spontaneous wave dynamics in the developing retina. This provides a framework to address the above questions of critical dynamics and mechanisms of wave generation and lateral propagation. The model is consistent with the above described mechanisms, but its purpose is to be as simple as possible while still being able to capture the defining spatiotemporal properties of retinal waves. We demonstrate the model is capable of producing physiological waves, as observed in mice [92], after which we address two questions. First, what are the conditions for spontaneous activity in the amacrine cell layer to propagate laterally as waves? We derive necessary conditions for propagating activity using techniques from non-linear dynamics that are made available through the use of a reaction-diffusion model. These conditions are shown to be consistent with known pharmacological and genetic manipulations. Second, once conditions for propagation are met, what are the conditions required for the spatiotemporal patterns of retinal wave activity to take the form of avalanches, as observed in multielectrode array recordings of stage II waves in a variety of species [121]? This is answered by appeal to a model of critical behaviour in a canonical model of forest fires.

6.2 Methods

6.2.1 The model

SACs are assumed to obey dynamics based on a Morris-Lecar [121] model, which includes a quasi-stationary voltage dependent calcium conductance. The voltage dynamics are described by

$$C_m V_t = -g_{Ca}(V - V_{Ca}) - g_K(V - V_K) - g_L^M(V - V_L) - g_{ACh}(V - V_{syn}) - g_n^M N(V - V_{Ca}),$$

where

$$
\begin{aligned}
g_{Ca}(V) &= \frac{1}{2} g_{Ca}^{M} \left[1 + \tanh\left(\frac{V - V_1}{V_2}\right)\right], \\
g_K(R) &= g_K^M R, \\
\Lambda(V) &= \cosh\left(\frac{V - V_3}{2V_4}\right), \\
R_\infty(V) &= \frac{1}{2}\left[1 + \tanh\left(\frac{V - V_3}{V_4}\right)\right],
\end{aligned}
$$

are the standard Morris-Lecar auxiliary functions. The model also includes an ACh conductance ($g_{ACh}(A)$) which depends on the local concentration of acetylcholine (A):

$$
g_{ACh}(A) = g_{ACh}^{M} \frac{\delta A^2}{1 + \delta A^2}.
$$

Additionally, a slow AHP variable (S) is activated according to a voltage-dependent function $G(V)$ with time scale τ_{ACh}:

$$
\begin{aligned}
S_t &= \gamma G(V) - \frac{S}{\tau_S} \\
G(V) &= \frac{1}{1 + \exp[-\kappa(V - V_0)]}.
\end{aligned}
$$

The sAHP variable enters the dynamics through its effect on refractory variable R as follows

$$
\tau_R R_t = \Lambda(V)(R_\infty - R) + \alpha S(1 - R),
$$

where $\Lambda(V)$ is defined above as part of the Morris-Lecar equations.

Cells are indexed by a continuous spatial parameter x such that the state space is described by the tuple $(V(x,t), R(x,t), S(x,t), A(x,t))$, where $V(x,t)$ is the membrane potential. On spiking,

cells release ACh at a voltage dependent rate $\beta G(V)$ which diffuses with coefficient D:

$$A_t = D\nabla^2 A + \beta G(V) - \frac{A}{\tau_{ACh}}$$
$$G(V) = \frac{1}{1+\exp[-\kappa(V-V_0)]}.$$

Coupling between SACs occurs only through the diffusion of ACh.

To induce spontaneous depolarizations, a noisy, excitatory current is included in simulations:

$$g_n^M N(V - V_{Ca}),$$

where at each time-step N is a Bernoulli random variable ($N \sim B(1,p)$). Refer to Table 6.1 for the parameter values used. The value of p is chosen such that the mean waiting time between sponataneous noisy channel openings matches approximately the rate of intrinsic spontaneous bursts observed in mouse retina recordings [92].

Computational Methods

Numerical integration was performed using a time-splitting method, in which the diffusion term was computed using a locally one dimensional Crank-Nicolson method and the reaction term was computed using a two-stage Runge-Kutta method. Simulations were computed in MATLAB (version 7.14.0; R2012a. Natick, Massachusetts: The Mathworks, Inc., 2012) using a 64×64 grid with a fixed step size of 1ms. The stochastic conductance variable was updated every 10ms. Simulation data used in statistical analysis was from 2500s of simulation on a 4mm^2 domain ($L = 2$mm), following a 500s warm-up period. The dimensions chosen mean that one grid point has a length of approximately 30μm – roughly the density of SACs in the inner plexiform layer [41]. Simulations were performed on a dual 6-core Intel Xeon 3.07Ghz machine with 24GB RAM, running Ubuntu 12.04. MATLAB code to run simulations is available for download at: https://github.com/benlansdell/retinalwaves.

Simulation statistics

Statistical analysis is performed as follows: snapshots of the simulation are taken every 10 time steps (10ms). To remove boundary effects, grid points 5 units or less from the boundary are not considered for analysis. All grid points whose potential is above a threshold of -60mV are labeled as active. All active points adjacent to one another are assigned a common wave number. For each set of cells assigned that number: the time from the first active cell to the last active cell is its wave duration, and the total number of cells is its size. The wave speed is calculated as in Blankenship *et al* 2009 [31]: the wave initiation and termination points are identified and the path the wave front travels between these two points is computed. This allows the maximum wave front speed for that wave to be computed, and the distance this wave front travels divided by the wave duration gives an average wave speed. To give better estimates of the wave front speed, waves which involve the activation of fewer than 50 cells or whose duration is less than 1s are ignored, and waves which do not have a distinct initiation point and are therefore the result of two waves colliding are also omitted (Figure S1). The interwave-intervals are computed by considering all above threshold wave activity, collisions or otherwise, and, for each grid point, measuring the time between successive threshold crossings. A minimum IWI of 2s was imposed.

A note on thresholds and smoothing voltages

Previous models of retinal waves consisted of two layers: an amacrine cell layer, which initiated and laterally propagated activity, and a ganglion cell layer, which acted as a smoothed 'read out' layer [82]. Subsequent models did away with the ganglion cell layer, and modeled only SACs [107, 121]. However, without this read-out later, wave analysis based directly on thresholding SAC potentials is noisy, making wave front tracking more difficult and wave 'collisions' more common. In order to compare simulated waves to recorded waves, Godfrey *et al* 2007 [107] thus compute a calcium response variable, in which wave fronts are smoothed. Our simulations (Figure 6.2A), and experiments [92] show highly variable participation in wave activity on a cell-cell

basis. Thus to compare our simulations to calcium imaging data, which is based on filtered RGC activity, we lower the threshold for which a cell is counted as participating in a wave to the ACh release threshold (near -55 mV), instead of its spiking threshold. This thus reflects the 'general excitation' of the region, not individual SAC activity, which makes it better suited for comparison with calcium imaging data. With this threshold, smoothing our data did not have a significant effect on wave labeling, so was not implemented (Figure S2).

Bifurcations

AUTO [68] was used for the numerical continuation of traveling fronts and bifurcation analysis.

6.2.2 Mathematical analysis

In order to analyze the model we first perform non-dimensionalisation, the parameters and scalings are outlined in Table 6.2. Non-dimensional dynamic variables are named in lower-case. To non-dimensionalise we make the following change of variables:

$$V = V_{Ca}v, \quad R = r, \quad A = \tau_{ACh}\beta a, \quad S = \tau_S \gamma s$$

and scale time (dimensioned quantity here denoted by a capital T) and space (capital X) such that

$$T = \frac{C_m}{g_K^M}t, \quad X = \sqrt{\frac{DC_m}{g_K^M}}x.$$

Then:

$$
\begin{aligned}
v_t &= -\tilde{g}_{Ca}(v)(v-1) - r(v-v_K) - \tilde{g}_L^M(v-v_L) - \tilde{g}_{ACh}(a)(v-v_{ACh}),\\
r_t &= \frac{1}{\tilde{\tau}_R}[\lambda(v)(r_\infty(v)-r) + \tilde{\alpha}s(1-r)],\\
s_t &= \frac{1}{\tilde{\tau}_S}(g(v)-s),\\
a_t &= \nabla^2 a + \frac{1}{\tilde{\tau}_{ACh}}(g(v)-a),
\end{aligned}
$$

with

$$
\begin{aligned}
g(v) &= \frac{1}{1+\exp[-\tilde{\kappa}(v-v_0)]}\\
\tilde{g}_{Ca}(v) &= \frac{1}{2}\tilde{g}_{Ca}^M\left[1+\tanh\left(\frac{v-v_1}{v_2}\right)\right]\\
\tilde{g}_{ACh}(a) &= \tilde{g}_{ACh}^M\frac{\tilde{\delta}a^2}{1+\tilde{\delta}a^2}\\
\lambda(v) &= \cosh\left(\frac{v-v_3}{2v_4}\right)\\
r_\infty(v) &= \frac{1}{2}\left[1+\tanh\left(\frac{v-v_3}{v_4}\right)\right].
\end{aligned}
$$

To proceed, let $\epsilon = 1/\tilde{\tau}_R$, $\epsilon\theta = 1/\tilde{\tau}_S$ and $\alpha = \pi$

$$
\begin{aligned}
f(v,r,a) &= -\tilde{g}_{Ca}(v)(v-1) - r(v-v_K) - \tilde{g}_L^M(v-v_L) - \tilde{g}_{ACh}(a)(v-v_{ACh}),\\
h(v,r,s) &= \frac{1}{\tilde{\tau}_R}[\lambda(v)(r_\infty(v)-r) + \tilde{\alpha}s(1-r)],\\
k(x,v) &= -x + g(v),
\end{aligned}
$$

to give:

$$\begin{aligned} v_t &= f(v, r, a), \\ r_t &= \epsilon h(v, r, s), \\ s_t &= \epsilon\theta k(s, v), \\ a_t &= \nabla^2 a + \frac{k(a, v)}{\tilde{\tau}_{ACh}}, \end{aligned} \tag{6.1}$$

which we analyze as a fast-slow system as is standard (refer to Ermentrout and Terman 2010 [77], or Keener and Sneyd 2001 [140]). The methods described below make no attempt to rigorously establish the existence of traveling fronts or waves.

Wave fronts

This scaling of space and time is appropriate in regions where the diffusion term is relevant, thus the system obtained by letting $\epsilon \to 0$

$$\begin{aligned} v_t &= f(v, a; r), \\ a_t &= \nabla^2 a + \frac{k(a, v)}{\tilde{\tau}_{ACh}}, \end{aligned} \tag{6.2}$$

provides an approximate description of the wave fronts and backs. The variable r is here considered a parameter. To construct wave front solutions in one spatial dimension, change coordinates to a frame moving with speed c:

$$x' = x - ct, \quad t' = t$$

so that

$$\begin{aligned} v_{t'} - cv_{x'} &= f(v, a; r), \\ a_{t'} - ca_{x'} &= \nabla^2 a + \frac{k(a, v)}{\tilde{\tau}_{ACh}}. \end{aligned}$$

A traveling front corresponds to the stationary solutions:

$$\begin{aligned} 0 &= f(v, a; r) + cv' + \epsilon v'', \\ 0 &= ca' + a'' + \frac{k(a, v)}{\tilde{\tau}_{ACh}}. \end{aligned} \tag{6.3}$$

where $' = \frac{d}{dx'}$. A bounded wave front solution, if it exists, corresponds to a *heteroclinic* orbit connecting the rest fixed point to the excited fixed point of Equation 6.3. In order to obtain this heteroclinic orbit, Equation 6.2 was simulated in MATLAB with a sigmoid function connecting the rest and excited fixed points for initial data. Once the solution had sufficient time to converge to the traveling front solution, its profile was saved, its wave speed calculated, and these two pieces of data put into the AUTO subpackage homcont to perform continuation on the system described in Equation 6.3. This allows for the computation of $c(r)$ – the wave speed as a function of refractory state.

Excitability thresholds

We seek to determine when the medium is sufficiently excitable to support a traveling front solution. In two variable neuronal systems such as Fitzhugh-Nagumo or Morris-Lecar, in which case the fast dynamics are one dimensional, an energy argument can provide conditions under which the medium is excitable (*e.g.* Ermentrout and Terman 2010 [77]). This is not possible since the fast dynamics are here two dimensional. So, let r_0 be the unique fixed point of Equation 6.1. We aim to find parameters for which $c(r_0) > 0$, which, assuming the front we are studying is excited to the left and at rest to the right, will correspond to a propagating front. Parameters for which $c(r_0) < 0$ will generate a receding front, and thus parameters for which $c(r_0) = 0$ corresponds to the transition between excitable and not. This is found using AUTO.

These results are compared to model simulations without the noise channel. The simulations are performed for different parameters for a simulated time of 10s, with an initial condition in which the domain is at rest besides a small cluster of grid points on one side of the domain.

Excitability is determined by measuring if activity above an 'excited' threshold is observed on the other side of the domain – from which we infer that a wave must have traversed the simulated retina.

Relation between one dimensional analysis and two dimensional simulations

Our analysis is performed in one spatial dimension only. Moving to two dimensions has the following effect: the Laplacian in radially symmetric polar coordinates is

$$\nabla^2 = \frac{\partial^2 f}{\partial r^2} + \frac{1}{r}\frac{\partial f}{\partial r},$$

which, for large r, approximates the one dimensional Laplacian operator. Thus we expect, for waves of large radius, the results we find for one spatial dimension will apply. Indeed, performing the above simulations to determine excitability threshold with either a small cluster of initially excited cells (thus simulating a 2D, but radially symmetric solution) or with an entire strip of initially excited cells (thus simulating essentially the 1D dynamics) showed negligible difference.

6.3 Results

6.3.1 A physiological reaction-diffusion model of cholinergic retinal waves

As in previous models [121, 92], individual SACs are modeled according to Morris-Lecar dynamics [185], with an additional sAHP current activated by depolarization and subsequent calcium influx. The sAHP current generates recovery times on the order of a minute. Each cell is assigned the same recovery time scale, and the dynamics are such that recovery is minimally activity-dependent (Figure 6.1A). Previous theoretical models [107, 121] and experimental observations [288] show

that larger depolarizations increase sAHP duration. The lack of a strongly activity-dependent refractory period in our model does not affect the its ability to generate physiological waves. A noisy, excitatory current induces spontaneous depolarizations (Figure 6.1B) at an average rate of once every 15 minutes per isolated cell, consistent with recordings in mice (Figure 6.1C).

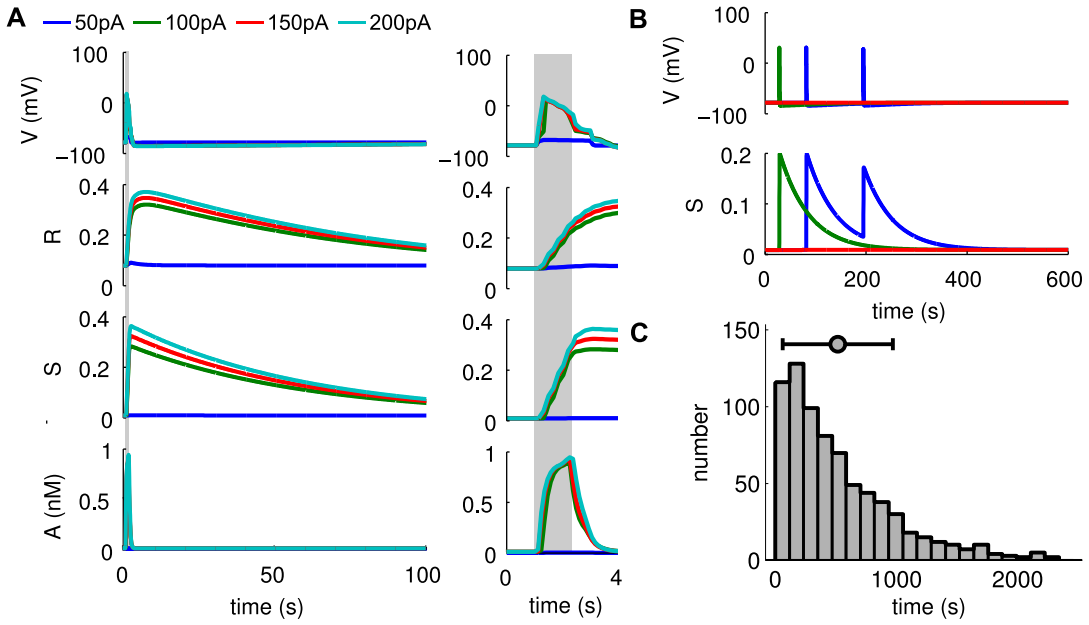


Figure 6.1: Isolated SAC dynamics. Starburst amacrine cells obey Morris-Lecar dynamics with voltage term V, refractory variable R, sAHP variable S and acetylcholine concentration A. **A.** Time course of individual SAC dynamics following current injections of indicated size, injected at $t = 1$s for 1.5s (shown in grey). Model SACs' refractory period shows dependence on amount of current injected. **B.** Single cell dynamics showing sparse spontaneous depolarizations. Different colors represent different SACs. **C.** Distribution of inter-event intervals in stochastic simulation of isolated SAC. A threshold of -50 mV was employed. Bar plot shows mean and standard deviation.

The dense synaptic connectivity of the SAC network and the potential role for extra-synaptic transmission suggests that a model based on local coupling between cells is appropriate. As such, coupling between amacrine cells is modeled via the volume release and diffusion of acetylcholine (ACh), and an excitatory post-synaptic current dependent on the local concentration of ACh. By taking a continuum limit of this amacrine cell network, a reaction-diffusion model is thus described (Methods). By modelling lateral excitation as a diffusion process the model is mathematically tractable. It should be emphasized that the diffusion of ACh is best thought of as an effective

diffusion process, representing the combination of synaptic and extra-synaptic excitation through acetylcholine. Below we will demonstrate the inclusion of long-range connections, in addition to local, 'diffusive', ones has minimal impact in simulations. The model is described in more detail in the Methods, but its notation is established here. For a SAC located at x, its membrane potential at time t is described by $V(x,t)$. Three dynamic variables regulate the membrane potential: $A(x,t)$, the extra-cellular ACh concentration which provides an excitatory current to the SAC; $S(x,t)$, a calcium-dependent potassium channel which provides an inhibitory, slow after hyperpolarization (sAHP) current; and $R(x,t)$, an inhibitory potassium channel modeled as in the original Morris-Lecar model. The vector

$$\mathbf{v} = (V, A, R, S) \tag{6.4}$$

then specifies the state of the system and obeys the following dynamics

$$\mathbf{v}_t = \mathbf{f}(\mathbf{v}) + \mathbf{D}\nabla^2\mathbf{v}.$$

The matrix $\mathbf{D}$ is a diagonal matrix whose specified entries indicate the diffusion coefficient of each dynamic variable

$$\mathbf{D} = \begin{pmatrix} 0 & & & \\ & D & & \\ & & 0 & \\ & & & 0 \end{pmatrix},$$

and whose blank entries represent zeros. The function $\mathbf{f}(\mathbf{v})$ specifies the cell-intrinsic dynamics.

Simulations show qualitatively that the model produces realistic waves (Figure 6.2A; Movie S1). Waves propagate without bias in their initiation region or direction (due to the translational and rotational symmetry of the equations), occur on average once every 60s, propagate at an average speed of 150 μm per second, and exhibit a broad distribution of wave sizes, all of which are consistent with *in vitro* recordings of mice retinal waves [92] (Figure 6.2B). The uniform

distribution of initiation points is expected given the homogeneity of the connectivities (diffusion coefficient) of the model. Some studies show that both *in vivo* and *in vitro* recordings contain a directional bias in propagation [1, 248], which could be modelled with a drift-diffusion model. However, since there is presently no physiological model for how this directionality occurs, we do not attempt to address these issues here.

The spontaneous firing of each cell is modeled as a Poisson process, which makes it simple to fix the per-cell spontaneous activation rate to be consistent with the recordings of Ford *et al* 2012 [92]. The mean firing rate is set to be low, such that adjacent spontaneous activations are exceedingly rare, and hence waves are the result of a single SAC depolarization. However, the mean rate is sufficiently high to desynchronize the network, such that the correlation of activity of neighboring SACs quickly decays as a function of distance (Figure 6.2C). Indeed, the correlation of both refractory variables (S and R) decays faster than either the voltage or acetylcholine variables, indicating neighboring SACs can exist in different refractory states and thus exhibit variable participation in waves [92, 121]. These features are consistent with physiological waves observed in mice. Further, by assuming a continuum model, the dimensionality of the dynamics can be drastically reduced, when compared with a network model. This feature is exploited in the following section.

6.3.2 Necessary conditions for wave propagation

Having established the model produces realistic waves, in this section we determine under what conditions propagating activity exists. For this note that previous studies predict that the evolving mosaic of retinal wave activity is a result of the refractory period of SACs, which delineates shifting boundaries of future wave activity [92]. If previous wave activity is indeed responsible for the finite spatial extent of retinal waves then, were the amacrine cell network in a homogeneously recovered state, activity would spread across the entire retina without dissipation. Under this hypothesis, a prerequisite for the amacrine cell network supporting propagating retinal wave

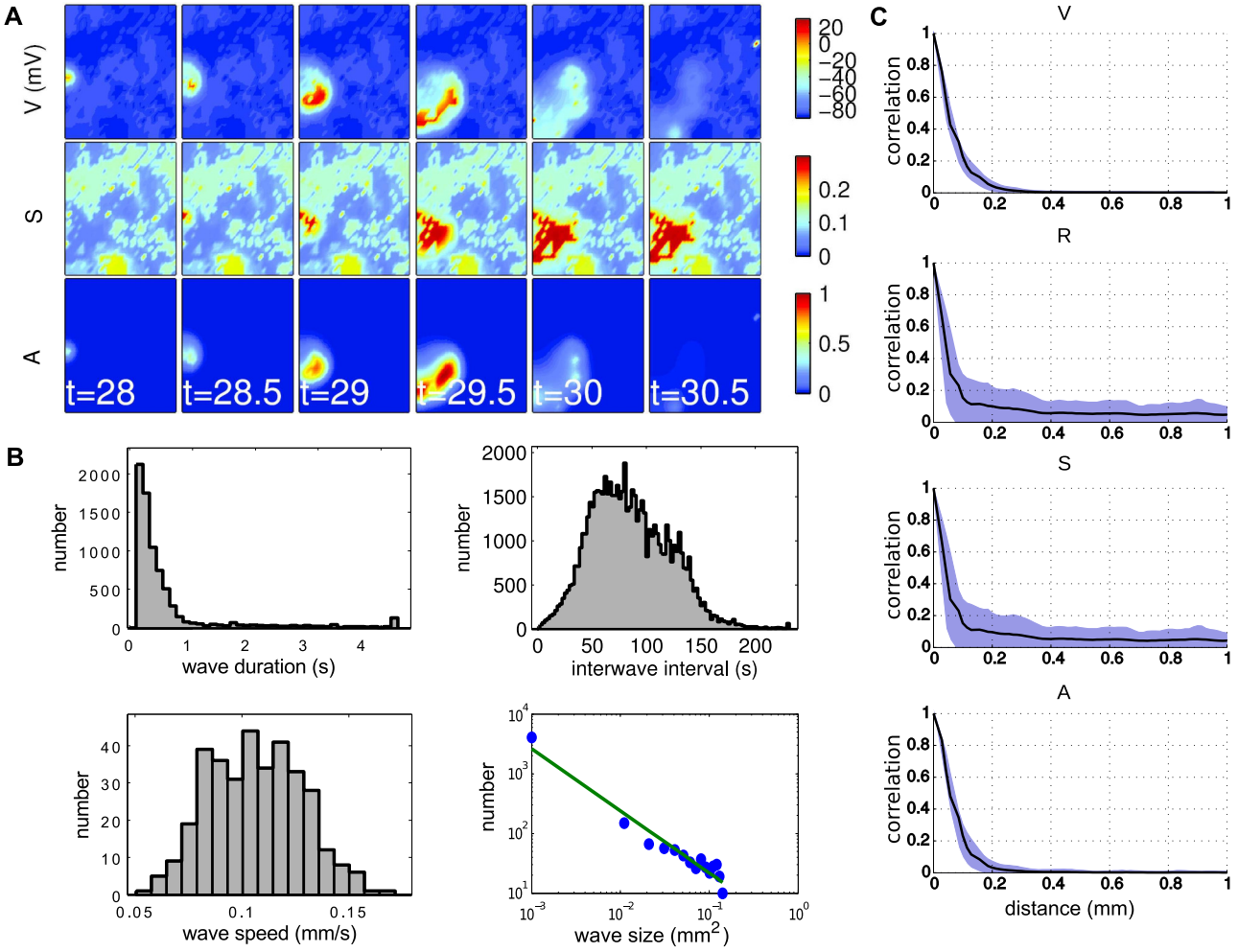


Figure 6.2: Model produces realistic cholinergic waves. **A.** Network dynamics showing spatiotemporal patterns of retinal waves **B.** Distribution of wave sizes, speeds, durations and inter-wave intervals from 2500s of simulation. Mean wave size is 0.017 mm^2($\pm$0.059mm^2), mean wave speed is 0.11mm/s ($\pm$0.022mm/s), mean wave duration is 0.63s ($\pm$0.90s), and mean inter-wave interval is 49s ($\pm$25s). **C.** SACs exhibit variable participation in waves. Pearson correlation coefficient between a cell in the center of the domain and all other cells. The correlation coefficient for each variable is plotted as a function of euclidean distance between cells. Computed using one 2500s simulation, with activity recorded every 0.1s. Solid curve represents a loess moving average estimate of mean correlation as a function of distance. Shaded region highlights all points within one standard deviation of this mean curve.

activity is thus that it should support traveling wave solutions were it provided an infinite spatial domain at rest – or that it should be an excitable medium.

To investigate parameters in which our retinal waves model is excitable is this sense, the amacrine cell network is studied as a non-linear reaction-diffusion system, in which we seek

parameters under which traveling wave solutions can be constructed. The method is outlined briefly here, and described in further detail in Methods. We take advantage of the fact that the voltage and acetylcholine variables change on a faster timescale than either refractory variables. Following a singular-perturbation analysis outlined by Keener and Sneyd, 2001 [140], (treated in a number of other texts also, *e.g.* Ermentrout and Terman 2010 [77]), the dynamics can thus be broken into a fast and slow system:

$$\begin{aligned} v_t &= f(v, r, a), \\ r_t &= \epsilon h(v, r, s), \\ s_t &= \epsilon\theta k(s, v), \\ a_t &= \nabla^2 a + \frac{k(a, v)}{\tilde{\tau}_{ACh}}, \end{aligned} \tag{6.5}$$

where the lower-case variables denote dimensionless quantities corresponding to their upper-case equivalents of Equation 6.4. Singular perturbation analysis is used to study systems for which dynamics on more than one timescale need to be considered. For such systems, attempting to find approximate solutions by neglecting very fast or very slow components would result in an incorrect description of the dynamics. Here, the parameter ϵ represents the ratio of the fast and slow timescales and is to be considered small (for the default parameters used in simulations $\epsilon \approx 0.001$). The model requires singular perturbation techniques because the small parameter affects the leading order derivative terms. The fast-system models the dynamics during the jump from the rest state to the excited (depolarized) state, or from the excited state to the refractory state. In one spatial dimension $\nabla^2 = \partial_{xx}$, and the zeroth-order ($\epsilon \to 0$) dynamics are therefore

$$\begin{aligned} v_t &= f(v, r, a), \\ a_t &= a_{xx} + \frac{k(a, v)}{\tilde{\tau}_{ACh}}. \end{aligned}$$

In a moving frame with speed $c(r)$, stationary solutions which connect the rest and excited fixed points are *heteroclinic* orbits which represent traveling front solutions. The basic idea of the wave front construction is outlined in Figure 6.3.

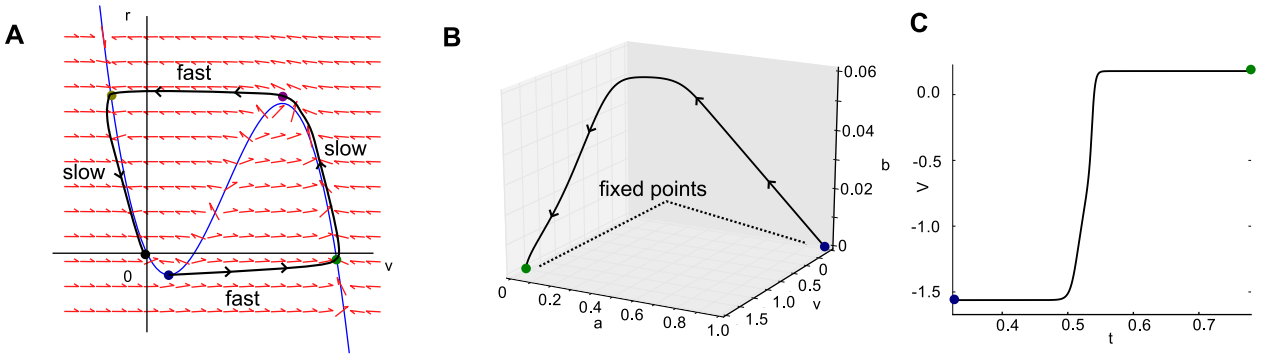


Figure 6.3: Construction of traveling wave-front. **A.** Fast-slow dynamics in the canonical Fitzhugh-Nagumo model of action potential generation. Black curve represents a trajectory of an action potential through phase space, in which a fast transition occurs between the rest (blue dot) and excited state (green dot), followed by slow excited dynamics (green to purple dot), another fast transition between the excited and refractory state (purple to yellow), and slow dynamics while refractory (yellow to black). Red arrows represent flow lines, and the blue curve is the $V = 0$ nullcline which defines the slow manifold ($R = 0$ nullcline not drawn for clarity). **B.** The fast system here is described by three dynamical variables (v, a, and $b \equiv a'$). Shown here is the trajectory connecting the rest (blue) and excited (green) fixed points, defining the wavefront. **C.** Temporal voltage dynamics of the wave front.

Parameters for which a heteroclinic orbit exists for a positive speed c are parameters which support propagating activity. Parameters at the transition between a propagating, positive speed, traveling front and a receding, negative speed, traveling front (that is, when $c = 0$) represent an excitability threshold. Figure 6.4 demonstrates this excitability threshold over a two dimensional parameter space, computed using the above framework and, for comparison, computed using numerical simulations of the model for different values of ϵ (the separation of fast and slow time scales). The analysis and simulations exhibit the same general behavior. The analysis method presented here is a more direct method for determining wave propagation because it is not necessary to set up and solve numerically a solution for each point in parameter space to be queried. Refer to Methods for more detail on the numerical simulations.

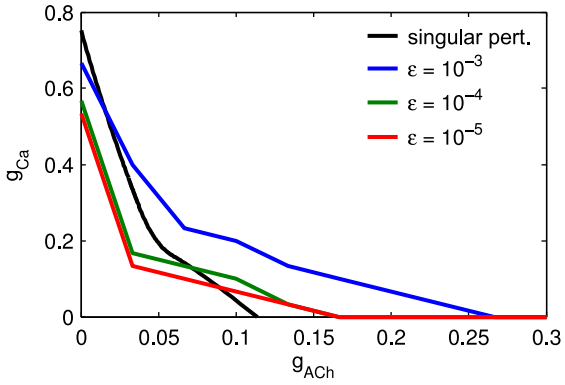


Figure 6.4: Parameter regimes which produce propagating activity. Numerical determination of retinal wave excitability threshold for different timescales ϵ and excitability threshold determined through singular perturbation analysis, both as functions of g_{ACh}, g_{Ca}. Each point on each curve indicates a point in parameter space in which the wavefront transitions from propagating to receding. Points in parameter space below each curve are therefore not excitable, while those above are excitable.

Both analysis and simulation demonstrate that excitability is sensitive to the value of both maximal conductances g^M_{ACh} and g^M_{Ca}. Further, excitability is independent of all aspects of the model not involved in the fast dynamics. This means that neither the spontaneous activation rate nor the duration of either refractory variable have an effect on wave propagation. Indeed, this is consistent with data presented in Ford *et al* 2012 [92], which shows manipulations affecting the strength of the sAHP current have minimal effect on measured wave speed. In the sense defined here, excitability is also independent of the diffusion coefficient D, as it is scaled out of the dimensionless equations used to compute the excitability thresholds. Thus, beyond assuming a non-zero coefficient, the diffusion rate merely determines the speed at which waves propagate. The existence of spontaneous, propagating activity in the developing retina is thus determined by factors regulating individual SACs excitability and their synaptic strength. It is important to understand that this analysis is only for the case of a domain uniformly at rest. Of course, we would like to consider the existence of waves within a medium that is not uniformly recovered but for which some of the domain may be refractory from previous activity. Our analysis provides a necessary condition for the existence of propagating activity in this more general case: a network incapable of supporting wave propagation at rest is a network incapable of supporting wave

activity when some of its cells are refractory. In this more general case, waves propagate not only if SACs are sufficiently excitable, but also if neighboring SACs are sufficiently recovered from prior depolarizations. The wave speed is then modulated by the refractory state as well (*e.g.* Figure 6.5).

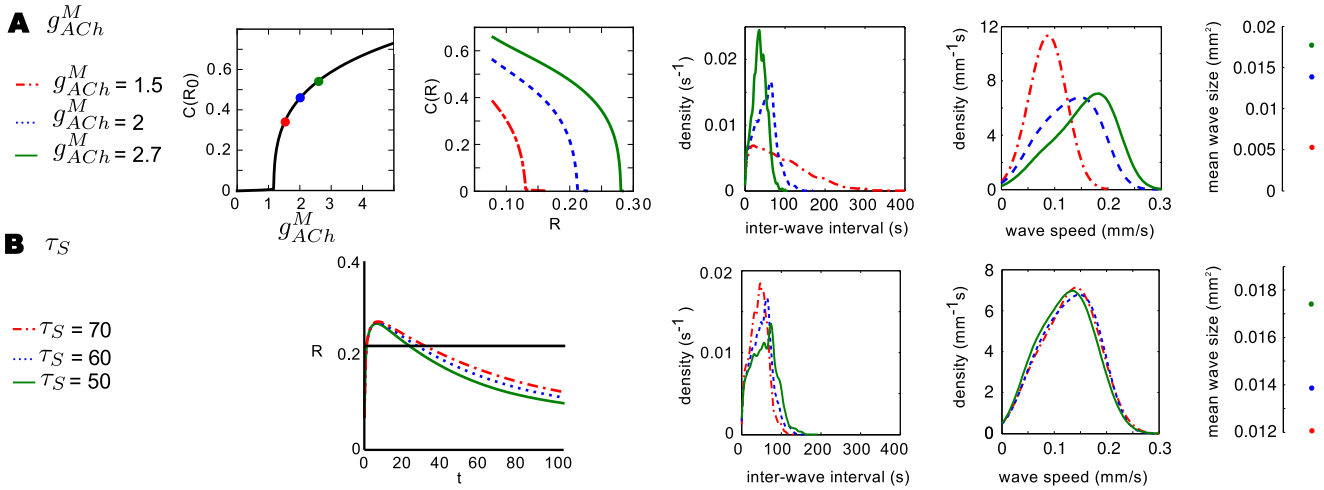


Figure 6.5: Modeling biophysical manipulations. **A** .Synaptic connection strength g^M_{ACh} is varied. Sub plots from left to right: speed of wave front $c(R_0)$ at rest (when $R = R_0$) as a function of conductance g^M_{ACh}, velocity indicates maximum wave-front speed since $R \geq R_0$ and $C(R)$ is monotonically decreasing, point at which c becomes zero represents excitability threshold; wave-front speed as function of refractory variable R for three different values of g^M_{ACh}; from 5000s of simulation of model with indicated values of g^M_{ACh} interwave-inteval; wave speed distribution; and mean wave size. **B.** Sub plots from left to right: dynamics of refractory variable R of individual SAC following depolarization with different sAHP timescales τ_S, black line indicates refractory value above which $C(R) \leq 0$ and thus represents an absolute refractory time period in which SAC is not sufficiently excitable to participate in future wave activity; from 5000s of simulation of model with indicated values of $\tau_S$inter-wave interval; wave speed distribution; and mean wave size.

6.3.3 Waves under physiological manipulations

The pharmacological or genetic manipulation of retinal waves forms a major component of their experimental study. In this section we show how two such common manipulations are recapitulated by our model and use the above analysis to interpret the effects of these manipulations. Using

a biophysically inspired model is beneficial since model parameters have clear, experimentally determinable values.

6.3.3.1 Synaptic connection strength

We first investigate the effect of nicotinic acetylcholine receptor (nAChR) agonists/antagonists by varying the maximal synaptic conductance g^M_{ACh} (see Methods for definition). Our analysis shows both the wave speed at rest and the wave speed as a function of refractory state R are highly sensitive to changes in g^M_{ACh} (Figure 6.5A). A 25% reduction in g^M_{ACh} lowers the $C(R)$ curve significantly, making the medium less excitable and waves more easily blocked by encounters with refractory cells. This manifests in simulations as a reduction in wave size, and a change in the inter-wave interval: from sharply peaked at a mean of approximately 50s to a monotonically decaying function similar to the individual SAC spontaneous firing rate (Figure 6.1), indicating that wave activity is localized and activations are primarily caused by spontaneous activity, not wave activity. This is indeed observed in pharmacological studies of mice, chick and turtle retina [232, 233, 18], which produce both a decrease in wave frequency and a decrease in wave size following treatment with nAChR antagonists.

Additionally, a study by Xu *et al* 2011 [281] used transgenic mice in which only about half of SACs express functional nicotinic acetylcholine receptors, which reduces the effective coupling of the network. In these mice, wave frequency and speed were unchanged, whereas wave size was significantly reduced, as the analysis of our model predicts.

Conversely, a 25% increase in g^M_{ACh}results in an increase in wave speed both from rest and as a function of refractory state, indicating that wave activity is more robust and less likely to be blocked by encounters with refractory cells. Indeed, in simulations this results in an increase in wave size and a more frequent and sharply peaked inter-wave interval distribution.

6.3.3.2 Modulation of sAHP current

We also investigate the effect on waves from varying the sAHP current. Experimentally, this is achieved through stimulating the cAMP second-messenger pathway via forskolin. We study this effect by varying the sAHP timescale τ_S. Since τ_S does not effect the fast time scale dynamics, our singular perturbation analysis does *not* provide insight into how τ_S effects wave speed or propagation. However, using the model's excitability criteria does allow us to compute the duration of an absolute refractory period, which is the length of the period in which activation, either spontaneous or through incident wave activity, results in subsequent a refractory state that does not support a propagating wave. This period is shown to be on the order of 30 seconds, in accordance with experimental findings [18]. (Figure 6.5B)

Retinal wave simulation *does* show how τ_S affects wave properties. An increase (respectively decrease) in τ_S results in an increase (respectively decrease) in the interwave interval and a minimal change in wave size and wave speed (Figure 6.5B). The reduction in IWI is expected given the decrease in absolute refractory period. The minimal change in wavespeed is also expected. Further, the moderate changes in τ_S presented here should not affect significantly the wave size (larger changes in τ_S through which different wave behavior regimes are explored are investigated below). These results agree with the study of Ford *et al* 2012 [92], which found that treatment with 1μM of forskolin resulted in roughly a 50% reduction in IWI, and only a moderate reduction in wave speed. The same increase in wave frequency and decrease in underlying sAHP current are also observed in ferrets and rabbit [288, 249].

Recently, Ford *et al* 2013 [90] proposed that the sAHP current is established by the two-pore potassium channel TREK1. In this study TREK1 knockout mice are shown to exhibit retinal waves with a significantly reduced (approximately halved) interwave intervals. The present model, in which wave frequency is shown to be increased by a reduction of the timescale of sAHP recovery, is consistent with these findings.

It is possible the effect of forskolin is on the strength of the sAHP current, not on its decay rate, thus we also investigated the effect of changing the strength parameter α, which determines

how sensitive each SAC is to the slow refractory state S. The effect was comparable to varying τ_S – a decrease in wave frequency and slight change in wave speed and size following an increase in α (Figure S3).

In summary, the preceding sections have demonstrated that our model produces physiological retinal waves, and that the existence of propagating activity can be understood by studying the model as an excitable medium. This allows for the effect of common genetic and pharmacological manipulations to be predicted.

6.3.4 Spatio-temporal patterns of propagating activity

We now turn our attention to a second question: given parameters in which propagating activity exists, what model parameters determine their form? Indeed, the spatiotemporal patterns of cholinergic retinal waves are often similar, despite differences in a range of physiological parameters, in a range of species. How is this similarity of form maintained given variations in physiology? Given parameters in which propagating activity exists, this section addresses the nature of the resulting spatiotemporal patterns. Hennig *et al* 2009 [121] notes that, both in MEA recordings and computer simulation, physiological waves take the form of avalanches, or power-law sized events. Thus it has been proposed that the developing amacrine cell network is in a critically configured state – a transition between a locally and globally connected network. Such neuronal avalanches have been observed and extensively studied in the both *in vitro* [205] and *in vivo* [26] cortex.

Here, we make an analogy between retinal waves and a model of self-organized criticality (SOC) [16], allowing us to determine for which parameters avalanches may be observed. A well-studied example of a complex system potentially demonstrating self-organized criticality is the Drossl-Schwabl forest fire model (DS-FFM) [72, 15], imagined as a grid in which each unit is either occupied (by a tree), ignited (burning tree), or empty (ash, burnt tree). At each discrete time step: 1) an occupied grid point spontaneously ignites with probability f, 2) burning trees ignite their occupied neighbors, 3) burning trees become empty and 4) empty sites regrow a tree with probability p. The analogy between the forest and the retina is clear: amacrine cells spontaneously

fire at some rate and excite their nearest recruitable neighbors to also fire, after which these cells experience a slow recovery time, which proceeds at some average rate. For a simulated lattice of n^2 cells, representing $L^2\text{mm}^2$ of retina (in our simulations $n = 64$ and $L = 2\text{mm}$), a simple heuristic derivation shows a relationship between f and p of the DS-FFM and the retinal wave model parameters:

$$f = \frac{\pi n^2 c^2 \tau^2 \hat{f}}{L^2}, \quad p = \frac{\tau}{\rho},$$

for wave speed c, a per cell spontaneous firing rate $\hat{f}$, burst duration τ, an effective refractory period ρ, and length of domain L. These values are determined either directly from model parameters ($\hat{f}$, L, n), from the model analysis (c and ρ), or from simulation (τ).

The heuristic derivation is as follows: to estimate f and p from retinal wave simulations, and relate it to the DS-FFM, we rescale time and space so that one time unit is the duration of a burst during wave activity (τ); in this way after one time unit every active grid point is now refractory. And rescale space such that the area of one 'lattice point' is the area covered by all grid points excited as a result of one point spontaneously firing (initiating a wave) ($\pi c^2 \tau^2$); in this way, during one time unit, an active lattice point induces the nearest (and only the nearest) excitable lattice points to become active. When scaled in this way, the dynamics of our retinal wave simulations approximate the dynamics described by the rules of the DS-FFM. The probability of 'regrowth' (recovery) is $p = \tau/\rho$, and the probability of a lattice point spontaneously firing is assumed to be the probability of a single simulated cell firing, multiplied by the number of cells that are included in that lattice point ($\pi c^2 \tau^2 n^2 / L^2$).

In the DS-FFM, SOC is expected when [72]

$$(f/p)^{-1/2} \ll p^{-1} \ll f^{-1},$$

or, for the present model, when

$$\left[\frac{\pi n^2 c^2 \tau \hat{f} \rho}{L^2}\right]^{-1/2} \ll \frac{\rho}{\tau} \ll \frac{L^2}{\pi n^2 c^2 \tau^2 \hat{f}}. \tag{6.6}$$

In this regime, retinal wave, or forest fire, sizes are characterized by a power-law distribution with scaling exponent of approximately $\theta = -1.15$ (simulation based [111], theoretical based [165, 122]). Parameters which have the largest and most direct impact on f and p are the per cell spontaneous firing rate $\hat{f}$, and the slow refractory variable τ_S (refer to Figure 6.5): these are the parameters which best determine when criticality may be observed. Figure 6.6 demonstrates that, within the region described by Equation (6.6), wave sizes distributions approximately follow a power law with an estimated exponent close to the expected $\theta = -1.15$. Conversely, simulations performed outside this parameter region are sub-critical and do not follow an approximate power-law. The same behaviour is observed when the distribution of wave duration (or lifetimes) are considered (Figure S4). This is further made clear by looking at the correlation in voltage activity between cells of a given distance from one another (Figure 6.6C). Parameters for which power-laws are observed produce an initially high, but sharply decaying correlation function, while the sub-critical parameter set produces significantly less correlated activity. This is indicative of the smaller, more localized wave activity expected in a sub-critical system.

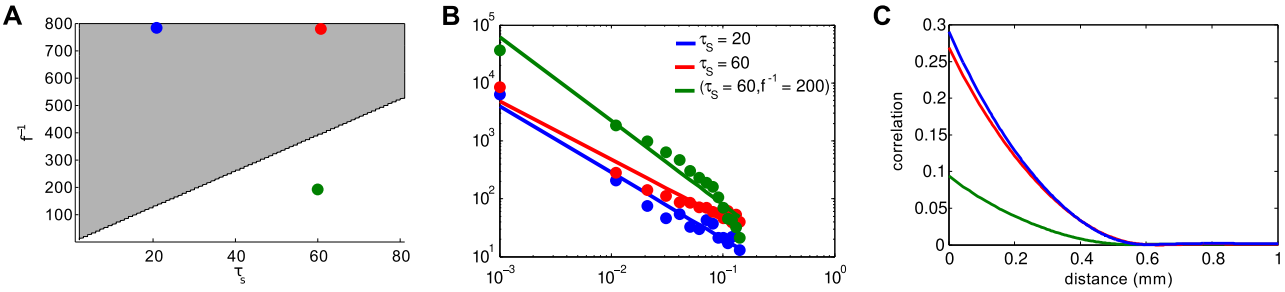


Figure 6.6: Power-law distributed wave-size retinal waves. **A.** Parameter space in which avalanches are expected (gray, Equation 6.6) and three sample points **B.** Wave size distributions (points) following 5000s of simulation on a 128×128 domain for specified values of τ_S and f^{-1}. Solid lines represent log-linear least-squares lines of best fit, having slopes: $\theta = -1.45$ ($R^2 = 0.95$, green), $\theta = -1.01$ ($R^2 = 0.95$, red) and $\theta = -1.14$ ($R^2 = 0.95$, blue) **C.** Correlation in membrane potential between cells of a given distance apart.

The relationship between the rate of spontaneous firing $\hat{f}$, and the slow refractory variable τ_S, dictates whether the network exists in a critical state. This simple inverse relationship (Figure 6.6) can explain how robust waves with similar spatiotemporal characteristics exist in very different

parameter regimes. Experiments in mice indicate that SACs depolarize infrequently but exhibit sAHPs lasting as long as two minutes [92], whereas experiments in rabbit [288] show a spontaneous depolarization rate an order of mangitude higher and sAHPs that are substantially shorter. While these two systems have cellular parameters that differ on an order of magnitude, their collective network activity is very similar. This can be explained by the fact that the ratio of $\hat{f}$ to τ_S is conserved across species.

This analysis can thus be used to predict, given particular parameters for a particular species, whether or not the wave-size distribution will be critical or sub-critical. Further, since both parameters varied here, $\hat{f}$ and τ_S, have no bearing on the excitability of the medium, the analysis of the spatio-temporal patterns of the amacrine cell network separates neatly into two stages. Firstly, if the medium is not excitable then only small, localized activity is expected. If the medium is excitable then, secondly, the spatio-temporal patterns expected will depend on the value of both $\hat{f}$ and τ_S, as discussed here.

6.4 Discussion

6.4.1 Comparison to previous studies

Close collaboration with experimentalists means computational modeling of retinal waves has been notably successful (refer to the reviews Gjorgjieva and Eglen 2011 [104] and Godfrey and Elgen 2009 [105]). Models have helped identify SACs as the cell layer which both initiates and laterally propagates waves [82], and have highlighted the importance of an activity-dependent refractory period to the generation of physiological waves [107]. Most recently, the models of Hennig *et al* 2009 [121] and Ford *et al* 2012 [92] (henceforth referred to by their first authors) have investigated how the cell-intrinsic spontaneous firing rate and the duration of each cell's refractory period contributes to physiological waves.

Godfrey and Hennig both propose the importance of a refractory period that scales with the degree of excitation. The Godfrey model was deterministic, and thus was reliant on the activity

dependent refractory periods to make the dynamics chaotic and therefore consistent with observed waves – variable in size, and with shifting wave boundaries.

In the Hennig model, the competition between the synchronizing force of waves and the desynchronizing force of spontaneous activity is balanced at physiological wave parameters. This was adjusted in the Ford model to match the observed low spontaneous activation rate observed in mice SACs. Physiological waves were then only observed when each cell's refractory time scale was allowed to vary randomly (a hypothesis also pursued in Feller *et al* 1997). Ford hypothesizes that this cell to cell variability is thus a necessary component of the developing amacrine cell network.

We find that neither hypothesis is necessary in producing realistic waves – our model does not contain an activity-dependent refractory variable or cell to cell variability in parameters. Our model is stochastic, consisting of sparse spontaneous depolarizations with enough strength to initiate a wave by themselves. These strong, sparse depolarizations are sufficient to induce variability in refractory periods. It remains to be shown how this might change in other species whose spontaneous depolarization rate is much higher, and in which cell-cell variability or activity-dependent refractory periods may play a more important role.

6.4.2 Experimental validation of the model

The model makes several predictions. Firstly, it provides a way to systematically investigate the effect of different parameters on both wave propagation and spatiotemporal patterns. One way to confirm the model experimentally would then be to match (more closely than was able to be performed in the present study) the parameters of the model to observed wave statistics such that in a quantitative fashion the effect of particular pharmacological manipulations (*e.g.* nAChR antagonists) could be reliably reproduced. The test would then be to predict the outcome of a different pharmacological manipulation (*e.g.* cholinesterase inhibition) that the model was not calibrated against. A simpler validation would be the following: a prediction the model makes is that certain parameters affect only the existence of propagating activity, while certain parameters

affect only the form of that propagating activity (power-law distributed or otherwise). Thus manipulations that affect only parameters within one of these classes should have no effect on behaviours determined by parameters in the other class. For instance, the only effect changes to parameters of the sAHP current, according to our analysis, is on the form of the activity – spiral waves, large domain-covering waves, *etc* – not on their existence.

Finally, the model predicts the existence of power-law distributed activity within a particular parameter range based on an analogy to a forest fire model. There are many different models of self-organized critical systems which have slightly different rules and slightly different statistics, and it is not clear that the forest fire model indeed is the best analog. These different models exhibit different sub-sampling artifacts, when only a subset of cells are observed, and these effects thus provide a way of teasing apart different SOC candidates [209]. Examining sub-sampling effects in our model's simulations, and in high-density recordings of *in-vivo* or *in-vitro* retinal waves, would provide a way of determining which model exhibiting criticality is most appropriate.

6.4.3 On the extra-synaptic diffusion of ACh

Previous retinal wave studies have hypothesized an extra-synaptic agent as responsible for wave propagation: Burgi *et al* 1994 [39], for instance, noted that the relatively slow wave speed is consistent more with an extra-synaptic agent than either gap junction or direct synaptic stimulation, and proposed extracellular potassium as a source of lateral excitation. This particular hypothesis was later discredited. However diffuse action of ACh within the IPL has been observed coincident with wave activity, suggesting that the volume transmission of acetylcholine may instead play a role [92, 256]. Thus, although direct recurrent synaptic connections between SACs both in mice and in rabbit have been observed [287, 92], there is evidence that the diffuse release of ACh is responsible for wave propagation: synaptic currents recorded during waves last several seconds beyond depolarization, suggesting excitation via an extra-synaptic agent; and retinal waves drive ON and OFF RGCs [278] and cells in the inner nuclear layer [277], despite the restricted stratification of

SAC processes within the IPL that do not form direct contacts with all cells, again suggesting propagation via volume transmission.

It is worth pointing out that extra-synaptic neurotransmission is observed throughout the developing nervous system to produce correlated spontaneous activity, and thus may be a general phenomenon regulating a diverse range of developmental programs. This may happen when the development of neurotransmitter release through synaptic vesicle fusion precedes the expression of the uptake transporters and breakdown enzymes required to prevent spillover into the extra-cellular space. For a the review of the role of extra-synaptic stimulation in generating spontaneous activity refer to Kerschensteiner 2013 [142].

The role of volume transmission has been explored in cortex [8], hippocampus [45], as well as extensively in stage III (glutamatergic) retinal waves [6, 31, 87]. In stage III waves extra-synaptic glutamate mediates lateral connections between bipolar cells.

The model presented here represents a theoretical validation that volume release of ACh is able to mediate stage II retinal waves. However, further experimental validation is needed beyond the above observations and this model. It should therefore be emphasized that the motivation of writing a model based on diffusion of acetylcholine is not to test the role of volume transmission. The dense lateral connectivity observed between SACs (synaptic or otherwise) means that lateral excitation is well approximated by a diffusion process – the model's diffusion of acetylcholine should be interpreted as a phenomenological, or effective, diffusion, not as a literal diffusion process based on the net Brownian motion of molecules which allows for singular-perturbation analysis to be performed.

Indeed, the extent of a SAC's arborization is only approximately 100 microns, so that even if synaptic terms were explicitly included their connectivity would be well-modelled by a short-range Gaussian weight function similar to the Gaussian kernel that results from modelling interactions as occurring through a diffusion process. The only cells in the retina with long range connections are other types of GABAergic amacrine cells, which do not alter waves since GABA antagonists

play a minor role in shaping the spatiotemporal features of waves. For these reasons models with long-range connectivity were not pursued in this study.

6.4.4 Stage II retinal waves and development

Retinal waves are one example of spontaneous activity in the nervous system instructing developmental processes. A number of processes appear to rely on the activity generated by retinal waves. Indeed, they have been shown to coordinate activity throughout the visual system [1]. In animals in which waves are blocked, the formation of the retinotopic map [49], the separation of retinogeniculate projections into eye-specific layers of the LGN [281], and the formation of ocular dominance and orientation selectivity [130] (and references therein) are all adversely effected.

Retinal waves are thus believed to provide a naturalistic stimulus from which downstream connections can be refined and stablized. However, it remains unclear how important the spatiotemporal patterns of the activity are, or whether any activity, regardless of its properties (size, speed, typical shape), would suffice to provide input for correct development to occur. Hennig *et al* 2009 [121], for instance, suggests that the critical form of the retinal waves are a useful feature, because, by definition, critical-state activity consists of events possessing no intrinsic length scale, they thereby provide an input without bias to any particular feature length. Godfrey *et al* 2009 [106], on the other hand, investigates this question explicitly in a model of retinotopic map refinement and finds that the only feature which affects the degree of refinement is the spatial correlation activity function – the size, speed and frequency of the waves otherwise had no effect.

If correlation structure of retinal activity is the main factor regulating downstream developmental processes, then how is this structure dependent on the spatiotemporal properties of retinal waves? Are scale-free, critical 'avalanches' the only form of activity that provide the necessary correlation structure? In other areas (*i.e.* cortex), a network operating at a critical state can be shown to be optimal for information processing, and for maintaining a high dynamic range [25, 240]: there is some functional role for criticality (as discussed in Shew and Plenz 2013 [239]), but its significance remains controversial [27]. Retinal waves provide a useful study into

the functional role of criticality, as they occur within well defined and understood circuitry where the above questions can be asked and thus critical activity can (potentially) be assigned a specific developmental role.

6.4.5 Self-organized criticality

Our study focused in particular on a forest fire model extensively studied in the context of complex systems and critical phenomena for its potential demonstration of self-organized criticality (SOC). This connection between correlated activity within the central nervous system and the DS-FFM has been noted before (*e.g.* Buice and Cowan 2009 [38]). The DS-FFM model is not without its issues: while initial theoretical results [72, 165] provided justification for numerically observed scaling behavior, subsequent, more extensive, numerical analysis [111] demonstrate the power-law scaling behavior to be transient and not reflective of the 'true' asymptotic behavior. Indeed, a variety of scaling parameter (τ) estimates have been observed for different parameter values and lattice sizes (refer to the summary presented in Pruessner 2012 [210]).

In theory, critical behavior is expected as the ratio $\theta = f/p$ tends to infinity. However, this limit corresponds to lightning strikes becoming increasingly rare, such that in simulations, for sufficiently large θ, strikes are rare enough that the entire domain recovers by the time the next strike hits, and the resulting dynamics are simply that the fire spreads across the entire domain. Thus, in practice, critical behavior is to be expected when θ is large, yet the characteristic length scale remains small compared to the lattice size. In addition to this behavior, for sufficiently large p values spiral wave fronts are observed [60]. These behaviours – propagating 'critical', stationary rotating spirals and radially symmetric domain-covering waves are the three regimes of spatially extending activity observed in simulation of the present model, and as classified in previous models [41]. The latter two behaviours are likely to occur in parameter regimes where the spontaneous activation rate is sparse, thus creating large scale structures, and where, relative to the wave-front speed, the rate of recovery is either too fast (spiral waves) or too slow (domain-covering radial waves). Though more systematic study is needed in these cases, we argue the principle value of

the analogy to the DS-FFM is not evidence of critical behavior in the developing retina (as it is not clear the DS-FFM is critical in a strict sense), but is the insight it provides into when and how these different regimes will be observed.

6.4.6 Wider context

This work shows that a model based on lateral excitation through the diffusion of ACh can account for physiological retinal waves, and can predict how wave dynamics consequently depend on biophysical parameters. Specifically, by determining when the retinal network is excitable and through analogy to a canonical forest-fire model, the spatiotemporal patterns observed in different parameter regimes can be understood. This framework is shown to be consistent with previous experimental studies.

We note that the nature of this analysis is novel: the singular perturbation and traveling front construction analysis presented here has not been extended before, from the well studied and more stereotyped waves studied in cortex, to the considerably more complicated wave behavior observed in the developing retina.

A number of avenues for future work present themselves. First, the reaction-diffusion component of the model can be naturally applied to glutamatergic waves, for which there is more significant evidence that diffusion is indeed a key excitatory process. Mechanistic models of stage III waves have recently been developed [6, 87], and would benefit from computational modeling to establish their validity. It is proposed in stage III waves that lateral excitation occurs via the diffusion of extra-synaptic glutamate released by bipolar cells in the ganglion cell layer, and that GABAergic connections from amacrine cells to bipolar and ganglion cells provides a source of inhibition that is absent during stage II waves. Changes to the present model to a model of stage III waves would therefore be significant, however the framework to both analyze and simulate the resulting model would remain the same.

Second, the connection between between criticality and development can also be explored in much more depth than was performed in this study. We believe the approaches developed here may also be applied to other forms of spontaneous and correlated activity in the nervous system.

Tables

Table 6.1: Parameters for Morris-Lecar model.

Parameter	Value	Notes	Parameter	Value	Notes
C_m	0.160 nF	Hennig	V_2	20 mV	Hennig
V_{Ca}	50 mV	Hennig	V_3	-25 mV	Sneyd
V_K	-90 mV	Hennig	V_4	40 mV	Sneyd
V_{syn}	50 mV		k	2	new
V_N	50 mV	Hennig	κ	0.2 mV^{-1}	new
V_L	-70 mV		V_0	-40 mV	new
g^M_{ACh}	2 nS		D	0.01 mm^2s^{-1}	new
g^M_{Ca}	10 nS	Hennig	β	5 $\text{nM}\cdot\text{s}^{-1}$	new
g^M_K	30 nS	Hennig	δ	800 nM^{-2}	new
g^M_L	3 nS		τ_R	5 s	Hennig
V_1	-20 mV	Hennig	τ_{ACh}	0.2 s	new
V_n	50mV		τ_S	60 s	
α	2	new	μ	5	
L	2 mm	approx 1 cell per grid point	γ	0.3 s^{-1}	new

Table 6.2: Dimensionless parameters for Morris-Lecar model.

Parameter	Relation	Value	Parameter	Relation	Value
v_K	V_K/V_{Ca}	-1.8	v_0	V_0/V_{Ca}	-0.8
v_{ACh}	V_{ACh}/V_{Ca}	1	v_1	V_1/V_{Ca}	-0.4
v_L	V_L/V_{Ca}	-1.4	v_2	V_2/V_{Ca}	0.4
$\bar{g}_{Ca}$	g^M_{Ca}/g^M_K	1/3	v_3	V_3/V_{Ca}	-0.5
$\bar{g}_{ACh}$	g^M_{ACh}/g^M_K	1/15	v_4	V_4/V_{Ca}	0.8
$\bar{g}_L$	g^M_L/g^M_K	1/10	$\tilde{\kappa}$	κV_{Ca}	10
$\frac{1}{\tilde{\tau}_R}$	$C_m/(\tau_R g^M_K)$	0.001067	$\tilde{D}$	$(C_m D)/(g^M_K L^2)$	1.333×10^{-5}
$\frac{1}{\tilde{\tau}_{ACh}}$	$C_m/(\tau_{ACh} g^M_K)$	0.02667	$\tilde{\delta}$	$\delta\tau^2_{ACh}\beta^2$	800
$\frac{1}{\tilde{\tau}_S}$	$C_m/(\tau_S g^M_K)$	8.88×10^{-5}	$\tilde{\alpha}$	$\alpha\tau_S\gamma$	36

Supporting Figures

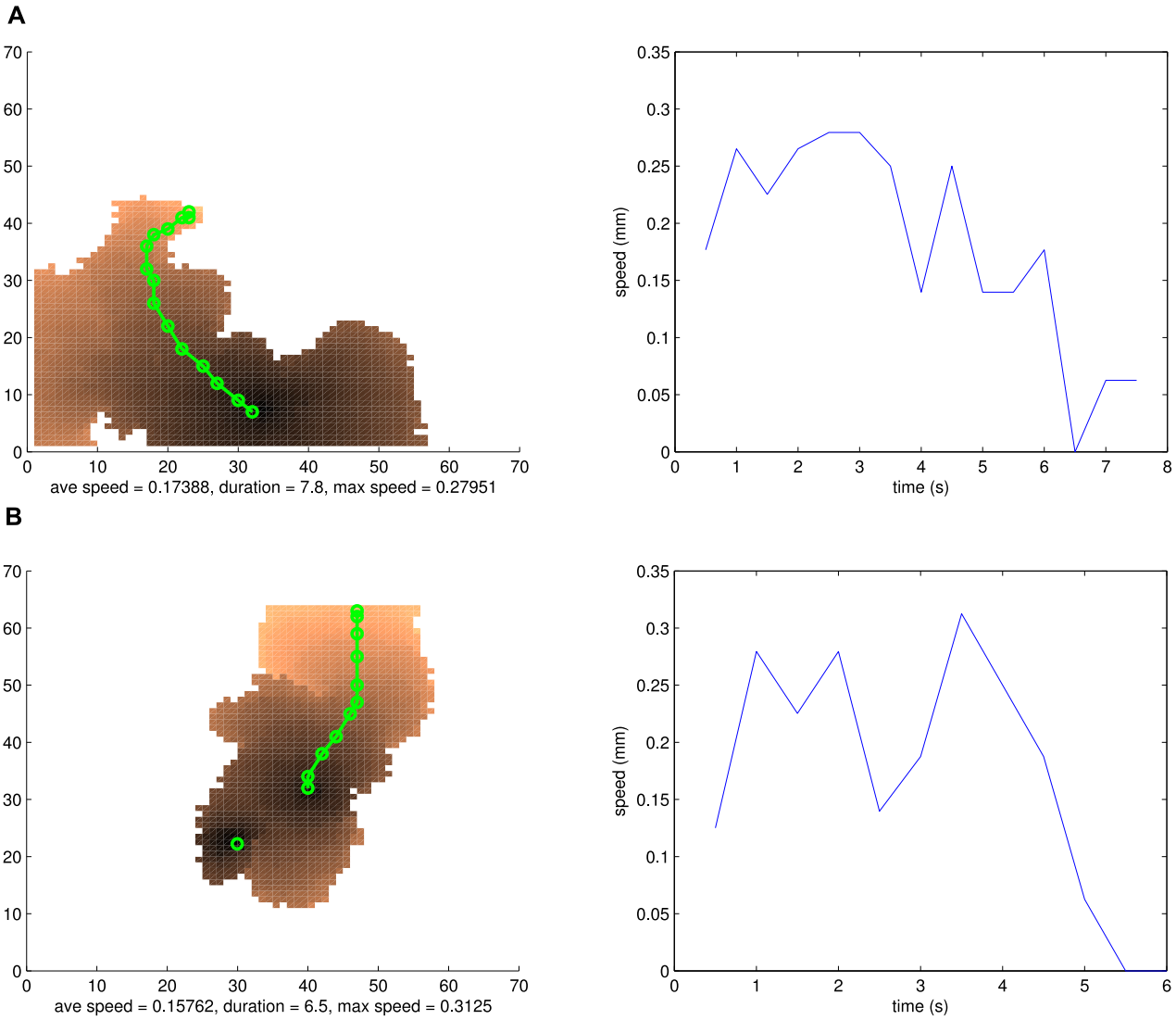


Figure 6.7: Determining speed of wave front for two example waves. Black corresponds to earlier times, and orange to later. At the latest time the wave is active, the wave front which connects the most distal point from the initiation point is tracked. Starting at this final point and moving backward in units of 0.5s, the next closest active point to the current one draws the wave front trajectory in reverse, as demonstrated in **A.**. This process does not make sense if a wave is the result of a collision, as shown in **B.**. Waves that are involved in a collision have more than one start point (additional start point shown as single green circle), meaning there is ambiguity in how to apply the wave speed algorithm. The wave speeds for these waves (shown in right subplot) are not counted in our analysis.

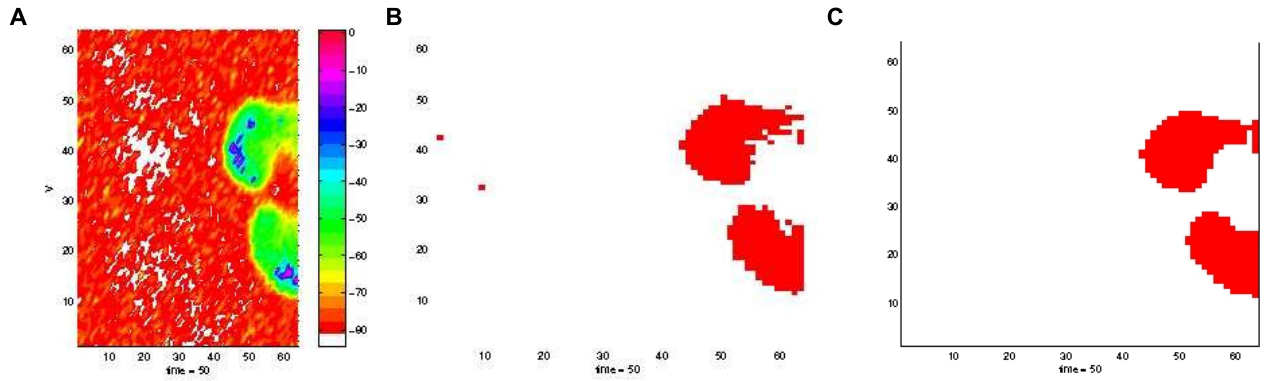


Figure 6.8: Effect of thresholds and smoothing on wave labeling. Snapshot of a thresholded and subsequently labeled wave from voltage data **A.** without **B.** and with **C.** smoothing. Threshold was set to -60 mV to produce wave forms which, by eye, match waves shown in voltage data.

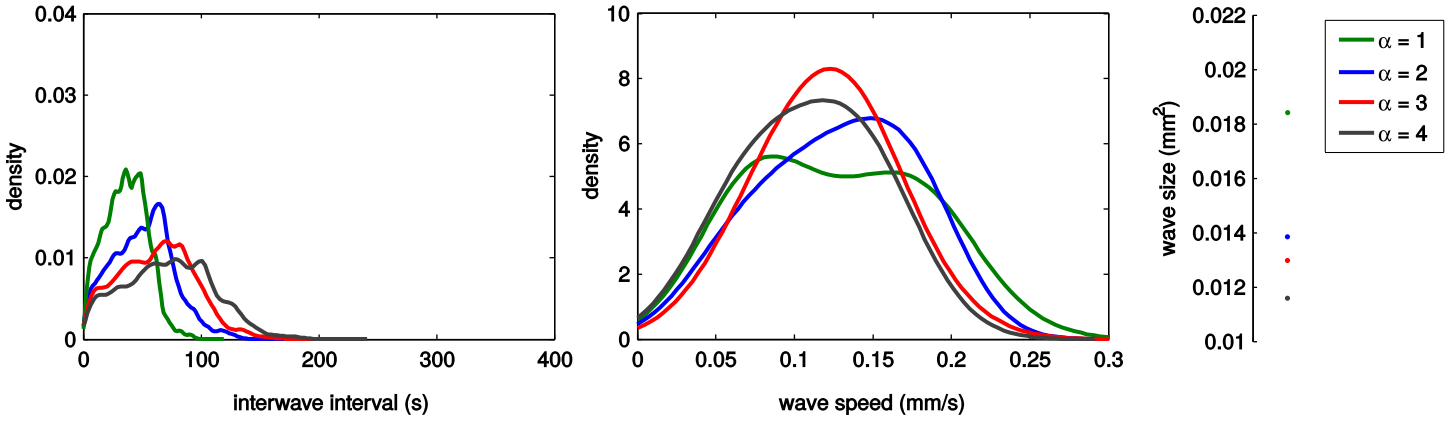


Figure 6.9: Effect of varying sAHP sensitivity parameter α. Statistics following 5000s of simulation of model with indicated values of α – interwave interval; wave speed distribution; and mean wave size. Comparison to Figure 6.5 shows varying α has a similar effect to varying the timescale τ_S. Note that varying the sAHP sensitivity parameter α or varying the rate of activation parameter γ have identical impact on the model – as can be seen from the dimensional analysis summaryized in Table 6.2.

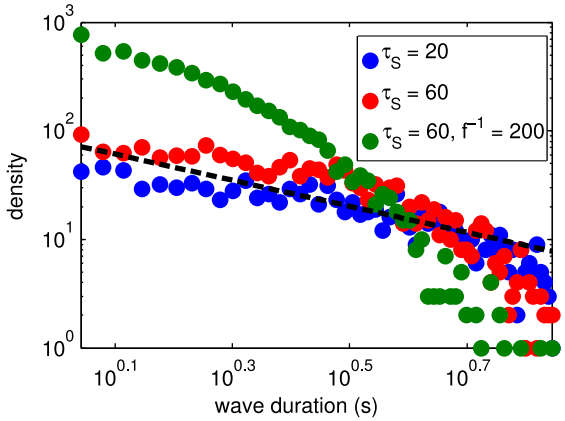


Figure 6.10: Distributions of wave duration (lifetime). Distribution of wave duration on logarithmic scale following 5000s simulation on a 128x128 grid point domain for specified values of τ_S and f^{-1}. For reference, the dotted black curve shows a theoretical distribution of the form $p(s) \sim s^{-\alpha}$ for $\alpha = -1.2$ as observed in large-scale simulations of the DS-FFM model [210]. Comparison to Figure 6.6 shows that SOC is not predicted for the parameters drawn in green, which indeed is clearly sub-critical.

7

Deformable object tracking

7.1 Introduction

This chapter provides background material on state-space models used in deformable object tracking. The neuroscience application that concerns us is described in Chapter 8. The chapter also provides background material on physical modeling methods used in object tracking, background material on common dense optic flow methods, and provides details of previous relevant object tracking algorithms.

7.2 State space model background

A discrete-time, continuous state-space model can be described by the following stochastic process,

$$\{(\mathbf{x}_k, \mathbf{y}_k)\}_{k\in\mathbb{Z}^*},$$

where $k \in \mathbb{Z}^*$ indexes time and random variables $\mathbf{x}_k \in \mathbb{R}^N$ and $\mathbf{y}_k \in \mathbb{R}^M$ share the probability space $(\Omega, \mathbb{F}, \mathbb{P})$. The system obeys non-linear dynamics and a non-linear observation process, with generic, zero-mean noise processes ϵ_k and η_k:

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{f}(\mathbf{x}_k, \epsilon_k)\\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k, \eta_k).\end{aligned}$$

Let $\mathbf{y}_{1:k} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k\}$ denote the history of the observation process up to time k. We want to estimate the density, $p_X(\mathbf{x}_k|\mathbf{y}_{1:k})$, defined to satisfy

$$\mathbb{P}(\mathbf{x}_k \in \mathcal{B}|\mathbf{y}_{1:k}) = \int_{\mathcal{B}} p_X(x|\mathbf{y}_{1:k})\, dx,$$

which specifies the state of the process given the entire observation history up to a time k. In this case the model satisfies the Markovian property, thus using Bayes' rule and the Chapman-Kolmogorov equation we can split the problem into a familiar 'predictor-corrector' type form:

$$p_X(\mathbf{x}_{k+1}|\mathbf{y}_{1:k+1}) \propto p_Y(\mathbf{y}_{k+1}|\mathbf{x}_{k+1}, \mathbf{y}_{1:k}) p_X^-(\mathbf{x}_{k+1}|\mathbf{y}_{1:k}) \qquad \text{(Bayes' rule)}, \tag{7.1}$$

with

$$p_X^-(\mathbf{x}_{k+1}|\mathbf{y}_{1:k}) = \int_{\mathbb{R}^N} dx\, p_X(\mathbf{x}_{k+1}|x, \mathbf{y}_{1:k}) p_X(x|\mathbf{y}_{1:k}). \tag{7.2}$$

The notation $p_X^-(\cdot)$ is used to highlight the distribution of $\mathbf{x}_{k+1}$ given observation information *only* up to the previous time step k. Thus (7.2) represents a *prediction* about state $\mathbf{x}_{k+1}$ given only observations up to k and (7.1) represents an adjustment to (7.2) once the observation information at $k+1$ is taken into account, in the form of the likelihood function

$$p_Y(\mathbf{y}_{k+1}|\mathbf{x}_{k+1}, \mathbf{y}_{1:k}) \equiv \mathcal{L}(\mathbf{x}_{k+1}|\mathbf{y}_{1:k}; \mathbf{y}_{k+1}).$$

Equations (7.1) and (7.2) describe the 'optimal non-linear filter', and are known as the *prediction* and *filtering* distributions, respectively. Typically analytic expressions for the state distributions are unavailable or intractable and require some form of approximation. The particular formulations relevant here are introduced below, in the context of models used for tracking a 2D deformable body in video.

7.2.1 The linear Kalman filter (KF)

We assume that $\mathbf{f}(.)$ and $\mathbf{h}(.)$ are both linear and ϵ_k and η_k are both zero-mean, Gaussian additive noise processes having variance σ_ϵ^2 and σ_η^2 respectively:

$$\mathbf{x}_{k+1} = F\mathbf{x}_k + \mathbf{c} + \epsilon_k,$$
$$\mathbf{y}_k = H\mathbf{x}_k + \mathbf{d} + \eta_k.$$

In this case the problem becomes the well-known Kalman filter. There are many ways to derive the Kalman filter, we present one way that comes immediately from (7.1) and (7.2) and that extends easily to other cases. Loosely it is based on the derivation for the point process filter presented in Truccolo et al 2005 [265]. We assume that the distribution $p_X(\mathbf{x}_k|\mathbf{y}_{1:k})$ is Gaussian for all k. This a useful simplification because only the first and second moment then characterize the entire distribution, and because given only information about the first and second moments, the

Gaussian distribution is the least informative (has maximum entropy). Given this assumption, all distributions of the problem are Gaussian, so we define the following first and second moments, which together characterize the entire system:

$$
\begin{aligned}
p_X(\mathbf{x}_{k+1}|\mathbf{y}_{1:k+1}) &= \text{Gaussian with mean } \hat{\mathbf{x}}_{k+1} \text{and covariance matrix } W_{k+1} \\
p_Y(\mathbf{y}_{k+1}|\mathbf{x}_{k+1}, \mathbf{y}_{1:k+1}) &= \text{likelihood function given by the observation model, here Gaussian} \\
p_X(\mathbf{x}_k|\mathbf{y}_{1:k}) &= \text{Gaussian with mean } \hat{\mathbf{x}}_k \text{and covariance matrix } W_k \\
\int_{\mathbb{R}^N} dx\, p_X(\mathbf{x}_{k+1}|x, \mathbf{y}_{1:k}) p_X(x|\mathbf{y}_{1:k}) &= \text{a convolution of two Gaussians, thus Gaussian with} \\
&\quad \text{mean } \hat{\mathbf{x}}^-_{k+1} \text{and covariance matrix } W^-_{k+1}.
\end{aligned}
$$

Using the observation model the likelihood function is Gaussian:

$$
p_Y(\mathbf{y}_{k+1}|\mathbf{x}_{k+1}, \mathbf{y}_{1:k+1}) \sim \mathcal{N}(H\hat{\mathbf{x}}_{k+1} + \mathbf{d}, HW^-_{k+1}H^T + \sigma^2_\eta),
$$

and the prediction variables are:

$$
\hat{\mathbf{x}}^-_{k+1} = F\hat{\mathbf{x}}_k + \mathbf{c} \tag{7.3}
$$

$$
W^-_{k+1} = FW_kF^T + \sigma^2_\epsilon. \tag{7.4}
$$

Thus it remains to determine $\hat{\mathbf{x}}_{k+1}$ and W_{k+1} in terms of the previous timestep's estimate and the prediction $\hat{\mathbf{x}}^-_{k+1}, W^-_{k+1}$. This we obtain from (7.1) by taking the log of both sides:

$$
\begin{aligned}
-\frac{1}{2}\left(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}\right)^T \left(W_{k+1}\right)^{-1} \left(\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}\right) = &-\frac{1}{2}\left(\mathbf{x}_{k+1} - \hat{\mathbf{x}}^-_{k+1}\right)^T \left(W^-_{k+1}\right)^{-1} \left(\mathbf{x}_{k+1} - \hat{\mathbf{x}}^-_{k+1}\right) \\
&- \frac{1}{2}\left(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}\right)^T \left(\sigma^2_\eta\right)^{-1} \left(\mathbf{y}_{k+1} - \hat{\mathbf{y}}^-_{k+1}\right),
\end{aligned} \tag{7.5}
$$

where $\hat{\mathbf{y}}_{k+1}^{-}$ represents the predicted observation, given the current state prediction:

$$\hat{\mathbf{y}}_{k+1}^{-} = H\hat{\mathbf{x}}_{k+1} + \mathbf{d}.$$

Equation (7.5) is a functional relation in $\mathbf{x}_{k+1}$, thus we can determine $\hat{\mathbf{x}}_{k+1}$ and W_{k+1} by substituting convenient values of $\mathbf{x}_{k+1}$ and solving. Here we Taylor expand the LHS and RHS of (7.5) around $\mathbf{x}_{k+1} = \hat{\mathbf{x}}_{k+1}^{-}$ and equate derivatives at first and second order. By doing this we obtain the update equations:

$$(W_{k+1})^{-1} = \left(W_{k+1}^{-}\right)^{-1} + H^T \left(\sigma_\eta^2\right)^{-1} H \tag{7.6}$$

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1}^{-} + W_{k+1} H^T \left(\sigma_\eta^2\right)^{-1} \left(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}^{-}\right) \tag{7.7}$$

which can be rearranged to be in a more standard Kalman filter form using the Woodbury matrix identity:

$$W_{k+1} = (I - K_{k+1}H)W_{k+1}^{-} \tag{7.8}$$

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1}^{-} + K_{k+1}(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}^{-}), \tag{7.9}$$

with *Kalman gain* matrix

$$K_{k+1} = W_{k+1}^{-} H^T \left[HW_{k+1}^{-} H^T + \sigma_\eta^2\right]^{-1}. \tag{7.10}$$

(Refer to Appendix A.2 for the quick calculation.)

7.2.2 The extended Kalman filter (EKF)

The above derivation admits a simple extension to approximate the case where dynamics and observation processes $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ are non-linear, with noise processes ϵ_k and η_k still additive and Gaussian. The extended Kalman filter [274] simply linearizes both $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$ at the current

and predicted state estimate. That is, we simply approximate

$$\mathbf{f}_{\mathbf{x}_0}(\mathbf{x}) \equiv \mathbf{f}(\mathbf{x}_0) + F_{\mathbf{x}_0}(\mathbf{x} - \mathbf{x}_0), \quad \mathbf{h}_{\mathbf{x}_0}(\mathbf{x}) \equiv \mathbf{h}(x_0) + H_{\mathbf{x}_0}(\mathbf{x} - \mathbf{x}_0)$$

with

$$F_{\mathbf{x}_0} \equiv \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_0}, \quad H_{\mathbf{x}_0} \equiv \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_0}.$$

The prediction equations(7.3) ,(7.4) become:

$$\hat{\mathbf{x}}_{k+1}^{-} = \mathbf{f}_{\hat{\mathbf{x}}_k}(\hat{\mathbf{x}}_k) \tag{7.11}$$

$$W_{k+1}^{-} = F_{\hat{\mathbf{x}}_k} W_k F_{\hat{\mathbf{x}}_k}^T + \sigma_\epsilon^2, \tag{7.12}$$

where we have expanded the process non-linearity around the previous estimate $\hat{\mathbf{x}}_k$. In this case the approximation and the function are identical: $\mathbf{f}(\hat{\mathbf{x}}_k) = \mathbf{f}_{\hat{\mathbf{x}}_k}(\hat{\mathbf{x}}_k)$, but care is taken to denote the distinction between the two for the iterated EKF, below. The update equations(7.6) ,(7.7) become:

$$(W_{k+1})^{-1} = \left(W_{k+1}^{-}\right)^{-1} + H_{\hat{\mathbf{x}}_{k+1}^{-}}^T \left(\sigma_\eta^2\right)^{-1} H_{\hat{\mathbf{x}}_{k+1}^{-}} \tag{7.13}$$

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1}^{-} + W_{k+1} H_{\hat{\mathbf{x}}_{k+1}^{-}}^T \left(\sigma_\eta^2\right)^{-1} \left(\mathbf{y}_{k+1} - \mathbf{h}_{\hat{\mathbf{x}}_{k+1}^{-}}(\hat{\mathbf{x}}_{k+1}^{-})\right) \tag{7.14}$$

where we have expanded the observation non-linearity around the prediction estimate $\hat{\mathbf{x}}_{k+1}^{-}$.

7.2.3 The iterated extended Kalman filter (IKF)

The common extension in the case of moderately non-linear systems is the iterated EKF. Is is achieved through iterating approximations of $\mathbf{h}(\cdot)$ at each update step, and so is in general able to provide a better estimate than the standard EKF [274, 119]. It is based on the idea that the approximations to the non-linearity $\mathbf{h}_{\mathbf{x}_k}(\cdot)$ will be more accurate after the measurement incorporation has taken place, and that this can then provide a better state estimate, which in turn can provide a better estimation of $\mathbf{h}_{\mathbf{x}_k}(\cdot)$, etc. Thus in the update step we should instead use the

approximation

$$\mathbf{h}(\mathbf{x}) \approx \mathbf{h}_{\hat{\mathbf{x}}_{k+1}}(\mathbf{x}).$$

This gives the update relation

$$\begin{aligned}\hat{\mathbf{x}}_{k+1} &= \hat{\mathbf{x}}_{k+1}^{-} + W_{k+1} H_{\hat{\mathbf{x}}_{k+1}}^{T} \left(\sigma_\eta^2\right)^{-1} \left(\mathbf{y}_{k+1} - \mathbf{h}_{\hat{\mathbf{x}}_{k+1}}(\hat{\mathbf{x}}_{k+1}^{-})\right) \\ &= \hat{\mathbf{x}}_{k+1}^{-} + W_{k+1} H_{\hat{\mathbf{x}}_{k+1}}^{T} \left(\sigma_\eta^2\right)^{-1} \left(\mathbf{y}_{k+1} - \mathbf{h}(\hat{\mathbf{x}}_{k+1}) - H_{\hat{\mathbf{x}}_{k+1}}(\hat{\mathbf{x}}_{k+1}^{-} - \hat{\mathbf{x}}_{k+1})\right)\end{aligned}$$

which is a non-linear equation in $\hat{\mathbf{x}}_{k+1}$ we can solve for through iteration. For update at step $k+1$ we let j denote the iteration number, set $\hat{\mathbf{x}}_{k+1}^{0} = \hat{\mathbf{x}}_{k+1}^{-}$ and then iterate over successive values of j:

$$\left(W_{k+1}^{j}\right)^{-1} = \left(W_{k+1}^{-}\right)^{-1} + H_{\hat{\mathbf{x}}_{k+1}^{j}}^{T} \left(\sigma_\eta^2\right)^{-1} H_{\hat{\mathbf{x}}_{k+1}^{j}} \tag{7.15}$$

$$\hat{\mathbf{x}}_{k+1}^{j+1} = \hat{\mathbf{x}}_{k+1}^{0} + W_{k+1}^{j} H_{\hat{\mathbf{x}}_{k+1}^{j}}^{T} \left(\sigma_\eta^2\right)^{-1} \left(\mathbf{y}_{k+1} - \mathbf{h}(\hat{\mathbf{x}}_{k+1}^{j}) - H_{\hat{\mathbf{x}}_{k+1}^{j}}(\hat{\mathbf{x}}_{k+1}^{-} - \hat{\mathbf{x}}_{k+1}^{j})\right) \tag{7.16}$$

This can be repeated until the change in $\hat{\mathbf{x}}_{k+1}^{j}$ is sufficiently small between iterations

$$\|\hat{\mathbf{x}}_{k+1}^{j+1} - \hat{\mathbf{x}}_{k+1}^{j}\| < \epsilon.$$

Following convergence at iteration J the covariance matrix is updated

$$\left(W_{k+1}\right)^{-1} = \left(W_{k+1}^{-}\right)^{-1} + H_{\hat{\mathbf{x}}_{k+1}^{J}}^{T} \left(\sigma_\eta^2\right)^{-1} H_{\hat{\mathbf{x}}_{k+1}^{J}}$$

and the state is set to $\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1}^{J}$. Given the form of the iteration, it is not surprising that the IKF can be shown to be equivalent to performing Gauss-Newton iteration to obtain the maximum a posteriori state estimate [28].

7.2.4 The particle filter (PF)

In the case where the non-linearities $\mathbf{f}$ and $\mathbf{h}$ or the noise distributions prove too severe or non-Gaussian we can turn to approximating the prediction and filtering distributions through Monte Carlo methods. Approximating (7.1) and (7.2) through random sampling methods is known as sequential Monte Carlo (SMC), or the *particle filter*. For the particle filter, we assume we have an approximation of the filtering distribution at time $k-1$. This takes the form of a set

$$\mathbf{X}_{k-1} = \{\mathbf{x}_{k-1,j}\}_{j=1}^{J}$$

of J points, or particles, drawn from $p_X(\mathbf{x}_{k-1}|\mathbf{y}_{k-1})$. Thus

$$p_X(\mathbf{x}_{k-1}|\mathbf{y}_{k-1}) \approx \sum_{j=1}^{J} \delta(\mathbf{x}_{k-1} - \mathbf{x}_{k-1,j}).$$

We simply use the process model to update these points to time k to obtain an estimate of the prediction distribution :

$$p_X^-(\mathbf{x}_k|\mathbf{y}_{k-1}) \approx \sum_{j=1}^{J} \delta(\mathbf{x}_k - \mathbf{f}(\mathbf{x}_{k-1,j})), \qquad j = 1, \ldots, J.$$

Once we have an estimate of the prediction distribution, an estimate for the updated filtering distribution is obtained by performing a weighted resampling of $\left\{\mathbf{x}_{k,j}^-\right\}_{j=1}^{J}$, with weights proportional to each sample's likelihood:

$$w_j = \mathcal{L}(\mathbf{x}_{k,j}^-; \mathbf{y}_k).$$

Roughly, this works because

$$\begin{aligned} p_X(\mathbf{x}_k|\mathbf{y}_k) =& \frac{1}{b_k} p_Y(\mathbf{y}_k|\mathbf{x}_k) p_X^-(\mathbf{x}_k|\mathbf{y}_{k-1}) \\ \approx& \sum_{j=1}^{J} \frac{1}{b_k} p_Y(\mathbf{y}_k|\mathbf{x}_k) \delta(\mathbf{x}_k - \mathbf{f}(\mathbf{x}_{k-1,j})) \\ =& \frac{1}{b_k} \sum_{j=1}^{J} w_j \delta(\mathbf{x}_k - \mathbf{f}(\mathbf{x}_{k-1,j})), \end{aligned}$$

where the final RHS term can be approximated by resampling from $\mathbf{X}_k^-$ with weights w_j. The particle filter proceeds by alternately propagating the filter distribution through the process model and then resampling according to the observation likelihoods. The particle filter can be proven to be unbiased. Standard references include: Doucet et al 2011 [70], Kitagawa 1987 and Arulampalam et al 2002 [13].

Implemented simply as above, the particle filter struggles in high dimensions, when the number of dimensions is of comparable magnitude to the number of particles. Such cases exacerbate the particle filter's *degeneracy* problem: the resampling stage can only *diminish* the number of separate particles – once a particular trajectory $\mathbf{x}_{k,j}$ is not represented at a particular time k there is no way through resampling for that trajectory to be recovered at any subsequent time. Adjustments to the particle filter for high dimensional models alter the resampling stage to counter degeneracy. Note that these problems don't arise in Gaussian or Gaussian mixture models, such as the ensemble Kalman filter (EnKF) in which the non-point like nature of the distributions is able to generate necessary 'jitter' at each time step.

7.3 Dense optical flow

One generic way of tracking elements in an image between two frames without any modeling or physical priors is known as *optic flow*. A sparse set of correspondence can be identified between two frames, which would involve a sparse optic flow problem, while a dense of correspondence points for every pixel in an image is a dense optic flow problem. The basic assumption of most

optic flow methods is that of brightness constancy. We assume that local variations in image intensity are the result of a pixel being translated some amount at the next frame. This being the case, if $I^t(\mathbf{x})$ represents an image pixel $\mathbf{x} = (x, y)$ at frame t then the following relation holds:

$$I^t(x, y) = I^{t+1}(x + u, y + v) \tag{7.17}$$

for the flow vector $(u, v) = \mathscr{F}(\mathbf{x}, t)$, which we wish to identify. Linearizing the gray value constancy assumption yields the well-known optic flow constraint:

$$I_x u + I_y v + I_t = 0. \tag{7.18}$$

Superscripts indicating time bin will be dropped when only one time bin is relevant. Different optic flow methods add different smoothness and constancy assumptions on top of (7.17) or (7.18) to estimate the optic flow field $\mathscr{F}$. For instance, the Lucas-Kanade method assumes that the flow vector is constant within a (small) region of interest, defined by a set of m pixels $\{\mathbf{x}_i\}_{i=1}^m$, yielding the linear system:

$$\begin{bmatrix} I_x(\mathbf{x}_1) & I_y(\mathbf{x}_1) \\ I_x(\mathbf{x}_2) & I_y(\mathbf{x}_2) \\ \vdots & \\ I_x(\mathbf{x}_m) & I_y(\mathbf{x}_m) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -I_t(\mathbf{x}_1) \\ -I_t(\mathbf{x}_2) \\ \vdots \\ -I_t(\mathbf{x}_m) \end{bmatrix}$$

that is solved by least squares. This is an example of a sparse optic-flow method, in that the vector (u, v) is only estimated for a small number of regions of an image.

The alternative is estimation of the flow field $\mathscr{F}(\mathbf{x}, t)$ for the entire image – so-called dense optic flow estimation. Here the canonical method is known as the Horn-Schunck method [127], and is based on a variational minimization problem. Ideally, we would like to minimize the energy functional:

$$E[\mathscr{F}] = \int \left[I^{t+1}(\mathbf{x} + \mathscr{F}) - I^t(\mathbf{x})\right]^2 + \lambda \|\nabla_{\mathbf{x}} \mathscr{F}\|_F^2 \, d\mathbf{x},$$

where the first term represents discrepancy between the gray value constancy and the second term represents a smoothness constraint on the flow field with regularization parameter λ, $\|\cdot\|_F$ denotes the Frobenius norm. This (in general) results in a non-convex optimization problem in $\mathscr{F}$, due to the non-linearity of I. Luckily, the linearization of I at $\mathbf{x}$ yields the functional:

$$E[\mathscr{F}] = \int \left[I_x u + I_y v + I_t\right]^2 + \lambda \|\nabla_{\mathbf{x}} \mathscr{F}\|_F^2 \, d\mathbf{x},$$

which is convex in $\mathscr{F}$ and which can be minimized using gradient descent. The Horn-Schunck quadratic penalty terms are generally too harsh for typical flow data, resulting in flow fields that are too smoothed, and which fail to capture discontinuities in the flow field which occur at object boundaries. More robust estimation can be achieved using L^1-like penalties.

For instance, a more recent model, implemented in the analysis below, is the energy functional used by Brox, Papenburg and Weickert 2004 [36]:

$$E_{data}[\mathscr{F}] = \int_\Omega \Psi\left(\|I^{t+1}(\mathbf{x}+w) - I^t(\mathbf{x})\|^2 + \gamma \|\nabla I^{t+1}(\mathbf{x}+w) - \nabla I^t(\mathbf{x})\|^2\right) d\mathbf{x}$$

and

$$E_{smooth}[\mathscr{F}] = \int_\Omega \Psi\left(\|\nabla 3u\|^2 + \|\nabla 3v\|^2\right) dx$$

with that the total energy functional is

$$E[\mathscr{F}] = E_{data} + \alpha E_{smooth}.$$

The function $\Psi(\cdot)$ is taken to be $\Psi(s) = \sqrt{s+\epsilon^2}$, which results in a modified L^1 minimization. The small constant $\epsilon > 0$ is added to ensure that, once other linearizations are performed, the optimization is convex.

More generally, we can interpret the energy functional as coming from a Bayesian framework, that is, as a combination of a likelihood of $\mathscr{F}$ given images I^t and I^{t+1} and prior probability of flow $\mathscr{F}$:

$$p(\mathscr{F}|I^{t+1}, I^t) \propto p(I^{t+1}|\mathscr{F}, I^t) \cdot p(\mathscr{F}|I^t)$$

where $p(\mathscr{F}|I^t)$ is often taken as just $p(\mathscr{F})$. If we describe the flow likelihood given the next frame as an independent, Gaussian variable, we have:

$$p(I^{t+1}|\mathscr{F}, I^t) \propto \exp\left[\frac{1}{2\sigma^2}\sum_{ij}\Big(I^{t+1}(i,j) - I^t(i+u, j+v)\Big)^2\right]$$

The Horn-Schunck model corresponds to a prior probability that represents a separable Markov random field.

7.4 Deformable object tracking

Instead, prior physical or object specific modes can be utilized to aid tracking. In this case, a common problem in computer vision is tracking the position and pose of a deformable object. Here we discuss approaches to this problem, focusing on face tracking in particular – an important and well studied computer vision problem. The general field of object tracking is vast, so the below approaches provide only a small sample of some relevance to our problem.

7.4.1 Local estimation of structure from motion using state-space models

Using state space models described above objects can be tracked in an online fashion, in which a running estimate is updated frame-by-frame. These methods tend to be computationally less expensive, but can be less robust, as can be more perturbed by occlusions and other unmodeled effects. Throughout the late 1980s through to 2010 a number of state space models have been developed for this purpose. Kalman filters provide the basis for many online tracking problems. In this setup a state space of head poses and positions is defined, which is then estimated recursively

(updated frame by frame). In these cases an observation model is specified. Estimation is often made possibly by confining the range of head poses to a small dimensional space defining 'eigenfaces' [266], or 'eigenheads' [135, 157].

7.4.2 With physical models

Alternatively others employ more complex physics based approaches, including skin and muscle dynamics, to regularize the tracking problem. Terzopoulos, Witkin and Kass 1987[261] were the first to include physics based models to generate overconstrained 3D tracking of deformable models. The original paper in particular was concerned with objects with a particular symmetry, to further constrain the estimation. The physics model was implemented with a finite difference scheme. Horowitz and Pentland 1991[203] use the FEM method to construct a physical model of a deformable mesh which is then matched to the optic flow through a Kalman filter. The method decomposes the object into a set of free-vibration modes. They constrain the problem by only looking at a truncated number of these models, neglecting the high frequency modes. Terzopoulos and Water 1993 [260] specify a more complete model of the muscles of the face. How these muscles relate to the motion of distinctive contours (lips, eyebrows, hairline, etc) can then be identified in the image and used to infer the muscle activity. Metaxas and Terzopoulos 1993 [179] specifies a Lagrangian formulation of a soft-body physics model. They model the discrepancy of model and observations as manifesting as forces that deform the body to match it with observations and show that this can be made equivalent to a Kalman filter.

Finally, we mention Essa and Pentland 1994 [78]. Here, as in Horowitz and Pentland above, an FEM method is specified and a physics model in terms of muscle activation is described. This analysis doesn't assume a low dimensional deformation space for the motion, instead the flow is matched to the object by assuming a face template. They also specify a set of muscles which are proposed to create the movement, and a Kalman filter is used for inference. After the geometric motion has been inferred with the Kalman filter, they then propose the muscles as a set of control inputs and, assuming the control is optimal, can infer some muscle strains that caused the motion.

Such a framework allows for more detailed facial expression classification in terms of muscle activations, rather than appearances alone[79].

7.4.3 Template matching

Finally, cases in which the object being tracked is not deformable but static may be solved by *template matching*. Dellaert et al 1998 [65] estimates both an object's pose and its texture in a two stage Kalman filter. The first stage is an IEKF for estimating the pose given a fixed texture. The second stage estimates the texture given the pose, implemented as a normal KF. In Shodl et al 1998[223] a textured head is fit to a set of images. A 6 parameter rigid 3D head model is rendered using OpenGL and the difference between the rendered image and the input frame used as an error measure to be minimized. In Cascia et al 2000[44] a cylindrical model is fit to the face. Only the position and orientation of the head are estimated. The texture that is used as the reference is only estimated once, at the initialization stage, and is not updated later.

A combination of all these methods is in some sense the method known as G-flow presented in Marks et al 2010 [175, 174, 186]. A deformable 3D object using morph bases is specified. They specify a generative model which takes the 3D model and produces a set of output pixels. Part of the model is a texture map that is to be learned at the model proceeds. Thus the method is a combination of 'flow' based methods which only use information about pairs of frames at a time, and template-based methods which use information collected over the entire video sequence at a time. The authors split the problem into two inferences: the pose estimation given a fixed texture and texture estimation given a fixed pose, similar to [65]. The pose estimation is performed with a Rao-Blackwellized particle filter and a linear Gaussian filter is used for the texture estimation.

8

High-dimensional state space tracking model through GPU acceleration

8.1 Motivation

Large-scale recordings of neural activity in freely behaving animals promises to uncover much about the relationship between neural activity and behavior. Particularly exciting is the possibility of recording intracellular calcium of whole organisms in conjunction with unconstrained behavior (e.g. in the worm C. Elegans: Nguyen et al 2015, Venkatachalam et al 2016 and in the cnidarian Hydra: Dupre and Yuste 2015[192, 73, 267]). Yet tracking neurons throughout such recordings is a significant computer vision problem. Current approaches in both planar and volumetric imaging rely on semi-automatic, curated tracking algorithms, which must be manually checked for consistency [192, 56, 73, 267]. However, when a full complement of behavior is under investigation,

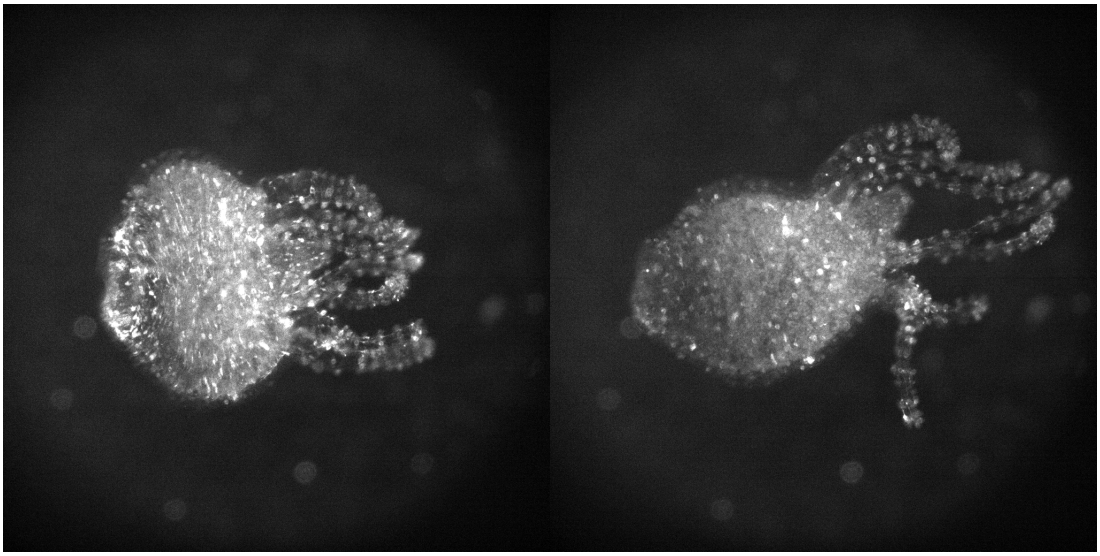


Figure 8.1: Images of *Hydra vulgaris* in which GCaMP6s is expressed in neurons.

these approaches do not scale to larger organisms or to longer video sequences. In these cases robustly tracking neurons remains a significant bottleneck, which motivates the need for principled augmentations to current cell tracking algorithms. In the application considered here, we focus on cases where neuron tracking may be aided by additionally tracking the body of the animal. Tracking the body is also a prerequisite for other behavioral analyses, and of course for the important task of relating the neural activity to the organism's behavior.

8.1.1 Hydra, the model cnidarian

In this and next chapter, we consider and neuron and body tracking problem in Hydra, a cnidarian (Figure 8.1). Cnidaria are an exclusively aquatic phylum of animals. The phylum includes coral, sea anenomes, jellyfish, and the, somewhat lesser known, hydra. Cnidaria are one of the most distantly related organisms from ourselves to possess a nervous system. Estimates of the last common ancestor (LCA) between cnidaria and ourselves place the divergence time somewhere around 824MYA, though range between 611MYA and 1035MYA[1] (Figure **??**). Thus as animals they differ significantly from most animals familiar to us. They do not possess bilateral symmetry

[1]As a comparison the estimated LCA between ourselves and fish (chondrichthyes) is estimated to be around 465MYA. Estimates obtained from http://timetree.org, accessed April 10th 2017.

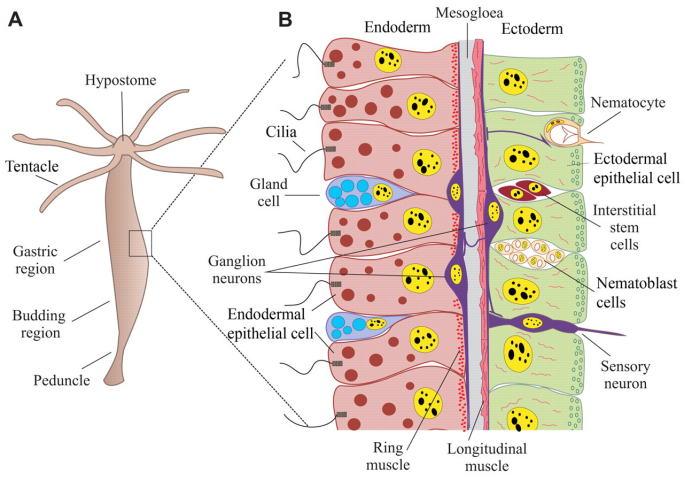


Figure 8.2: Anatomy of Hydra. Reproduced from Technau and Steele 2012 [259].

(are members or radiata, instead of bilateria), possess only two cell layers (diploblastic, instead of triploblastic), do not possess both a mouth and anus (just one 'stome'). Structurally, cnidaria are very simple compared with higher animals. Cnidaria are named for their cnidocytes (or nematocytes)[2], barbs that can be used to paralyze and catch prey. Their nervous systems possess no apparent ganglia, no central nervous system, nor do they possess brains – rather the nervous system consists of a dispersed nerve net throughout their body and tentacles.

Hydra is suitable for studying basic principles of nervous system function for a number of reasons. First, its nerve net means that calcium imaging can be used to observe the neural activity of a significant proportion of the animal. Its body size (500μm to 1.5cm) means that it is small enough fit within the field of view of a traditional microscope. The Hydra does not noticeably age, making the generation of genetic lines easier to maintain. The Hydra also regenerates, making it less susceptible to photodamage during imaging.

8.1.2 Approach

In aiding the neuron tracking problem in Hydra, constraints obtained from the its pose may be useful. As the hydra is a highly elastic, jellyfish-like organism, here we seek to develop model-based motion estimation methods that can be used for tracking highly deformable objects. The

[2] *Cnida* can be traced to the Latin cnide, for nettle, and the Greek knide, of the same meaning.

model is based on a GPU implementation of a high dimensional extended Kalman filter. The GPU implementation allows significantly higher dimensional state-space model to be considered than previous work. While optical flow could be considered alone as a prior on neuron tracking, the approach adopted here allows for more general considerations to be incorporated in a principled way, also, including physical constraints and edge constraints – keeping the neurons within the outline of the animal.

Of course, deformable object tracking is an important problem in computer vision more generally. It finds application in medical imaging, face tracking, video compression and augmented reality. The methods developed here will share some of these applications.

8.2 Model

Similar to Dellaert et al 1998 [65], we propose a two stage Kalman filter approach. We present a model that is well suited to high M, reasonably high N models. Here we describe the application to tracking a deformable body from video data. The observed data takes the form of video pixel intensities $\mathbf{z}_k \in \mathbb{R}^m$ for images of dimensions m_x, m_y, such that $m = m_x \times m_y$, the pre-computed dense optic flow field $\mathbf{j}_k^x, \mathbf{j}_k^y \in \mathbb{R}^m$ generated between adjacent frames $\mathbf{z}_k$ and $\mathbf{z}_{k+1}$, and an object mask, $\mathbf{m}_k \in \mathbb{R}^m$, denoting the pixels assigned to the object being tracked:

$$\mathbf{y}_k \equiv (\mathbf{z}_k, \mathbf{j}_k^x, \mathbf{j}_k^y, \mathbf{m}_k)^T \in \mathbb{R}^M,$$

with $M = 4m$. We assume the subject is easily separable from the background and, given an initial parameterization of its pose, wish to track that parameterization throughout the video.

The state space:

$$\mathbf{x}_k \equiv (x_1, y_1, \ldots, x_n, y_n, v_1^x, v_1^y, \ldots, v_n^x, v_n^y)^T \equiv \begin{bmatrix} \mathbf{q}_k \\ \mathbf{v}_k \end{bmatrix} \in \mathbb{R}^N,$$

represents the positions and velocities of a set of n mesh points describing the object, such that $N = 4n$. To initialize the model the points are Delaunay triangulated to generate a set of P faces, $\mathcal{D} = \{\mathcal{D}_p\}_{p=1}^{P}$, where each $\mathcal{D}_p$ is a triplet (d_p^1, d_p^2, d_p^3) indicating the face's corresponding vertex numbers, oriented clockwise, the mesh contains I edges. Let V denote the set of vertices. Each face is assigned a texture based on the initial appearance and parameterization of the model, $\mathcal{T} = \{\mathcal{T}_p\}_{p=1}^{P}$. The initial texture will be referred to as the *reference texture*. Each $\mathcal{T}_p$ consists of UV coordinates:

$$\mathcal{T}_p = (t_p^1, t_p^2, t_p^3).$$

where each t_p^i indexes a point in the reference texture. To begin with, we consider a model in which these textures remain constant throughout the video, though this can be generalized.

8.2.1 Process model

8.2.1.1 Constant velocity dynamics

The prediction distribution embodies the physics of the states. In this case, the states represent a three dimensional object projected onto two dimensions, and as such do not obey an intuitively mechanical model [3]. To keep matters simple and to remain as agnostic as possible about the dynamics of the mesh points, we begin by implementing a constant velocity dynamical system, though this will be generalized later. The dynamics are therefore described as the following linear system:

$$\mathbf{x}_k = F\mathbf{x}_{k-1} + G\mathbf{a}_k,$$

for

$$F = \begin{bmatrix} I_{2n} & (\Delta t)I_{2n} \\ & I_{2n} \end{bmatrix}, \qquad G = \begin{bmatrix} (\Delta t)^2 I_{2n}/2 \\ (\Delta t)I_{2n} \end{bmatrix},$$

[3] Though, given a simple perspective model, it can be easily shown that any low-dimensional, in terms of morph bases, three dimensional mesh can be realized with a (possibly higher-dimensional) two-dimensional mesh [280].

with I_n representing the $n \times n$ identity matrix, (Δt) representing the timestep size and empty matrix entries representing zeros. Process $\mathbf{a}_k \in \mathbb{R}^{2n}$ represents a zero-mean, Gaussian distributed acceleration applied independently to each vertex, and thus a unit's worth affects each velocity component by Δt but each position component by $(\Delta t)^2/2$. The model is therefore

$$\mathbf{x}_k = F\mathbf{x}_{k-1} + \epsilon_k,$$

where $\epsilon_k \sim \mathcal{N}(0, Q)$ and

$$Q = GG^T\sigma_\epsilon^2 = \begin{bmatrix} (\Delta t)^4 I_{2n}/4 & (\Delta t)^3 I_{2n}/2 \\ (\Delta t)^3 I_{2n}/2 & (\Delta t)^2 I_{2n} \end{bmatrix} \sigma_\epsilon^2.$$

This model assumes the least about the underlying dynamics.

8.2.1.2 Mass-spring dynamics

However there are some constraints that we do want to enforce, even if not a literal mechanical model is to be implemented. The mesh must always maintain its continuity, with mesh points never crossing edges. Thus here we use a mass-spring like model, similar to [162]. Based on implicitly solving the dynamics equations. Some other possibilities include[187][225] [188].

The constant velocity dynamics model provides some minimal regularization to prevent gross deformations and unrealistic poses, however, beyond penalizing large changes in frame-to-frame velocity of each vertex, the dynamics do nothing to hold the object together. While we wish to consider objects that are highly deformable, and so want to make few physical assumptions about the object being tracking, its elasticity, etc, additional assumptions about the tracked object's physics do help fill in individual vertex motion in cases when visual cues do not help update its position. Other deformable tracking algorithms do employ simple constant velocity dynamics, but these models compensate for the decrease in regularization by dramatically reducing the

dimension of the pose-space – e.g. restricting to a small number of deformation modes or morph bases. In our case, morph bases are not known and a small number of deformation modes

For this purpose, many other deformable object tracking algorithms employ a complete finite element method solution. In our case, in which large deformations are possible, a linear elasticity assumption may not be sufficient. Non-linear elasticity models are computationally more expensive. To keep a computationally simple physics model we do not implement such models. Our models, being a 2D projection of a 3D object are not physically consistent in the way normally considered by finite elements.

A mass-spring model (MSM) obeys non-linear dynamics:

$$\mathbf{x}_k = F\mathbf{x}_{k-1} + \mathbf{f}(\mathbf{x}_{k-1}) + \epsilon_k,$$

for force function $\mathbf{f} : \mathbb{R}^N \to \mathbb{R}^N$ which depends only on the mesh positions $\mathbf{y}_k$:

$$\mathbf{f}(\mathbf{x}_k) = \begin{bmatrix} \mathbf{0}_{2n} \\ (\Delta t)M^{-1}f(\mathbf{q}_k) \end{bmatrix},$$

where $\mathbf{0}_{2n}$ is the $2n \times 2n$ zero matrix, (Δt) is time step size, M^{-1} is the mass of each mesh point, and $f : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$. Additive Gaussian noise ϵ_k is defined as in the constant velocity case.

If $f(\mathbf{x}_k)$ represents Hookean spring interactions the model takes the following form. Let $\mathbf{l}_i$ represent tension on an edge i given by $\mathbf{d_i} = \mathbf{q}_{i,1} - \mathbf{q}_{i,2}$, so that

$$\mathbf{l}_i = -\kappa(1 - l_0/\|\mathbf{d}_i\|)\mathbf{d}_i$$

for rest length l_0 and spring stiffness κ. Let $L(\mathbf{q}_k) = \Downarrow_i \mathbf{l}_i \in \mathbb{R}^{2I}$ be the vertical concatenation of each edge tension, and $\mathbf{d} = \Downarrow_i \mathbf{d}_i$ be the vertical concatenation of all the edge vectors. Then the force applied to each node is given by

$$f(\mathbf{q}_k) = (K' \otimes I_2)L(\mathbf{q}_k),$$

for incidence matrix $K' \in \mathbb{R}^{n \times I}$, where $K'_{ij} = 1$ if edge j points to node i and $K'_{i'j} = -1$ if edge j points from node i', and $K'_{ij} = 0$ otherwise, and for Kronecker product $\otimes$ and 2×2 identity matrix I_2. The direction is arbitrary but, of course, must be consistent with each $\mathbf{l}_i$. In what follows let $K \equiv K' \otimes I_2$.

As presented the model implements simple forward Euler integration, which is unstable for large forces/high spring tension. Thus we present a formulation of the dynamics for implicit, backward Euler integration:

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \begin{bmatrix} (\Delta t)\mathbf{v}_k \\ (\Delta t)M^{-1}f(\mathbf{q}_k) \end{bmatrix},$$

which is a non-linear equation in $\mathbf{x}_k$. We can use Newton's method to solve for $\mathbf{x}_k$:

$$g(\mathbf{x}_k; \mathbf{x}_{k-1}) \equiv \mathbf{x}_k - \mathbf{x}_{k-1} - \begin{bmatrix} (\Delta t)\mathbf{v}_k \\ (\Delta t)M^{-1}f(\mathbf{q}_k) \end{bmatrix} = 0$$

We must compute the Jacobian of f:

$$\frac{\partial f}{\partial \mathbf{q}} = K\frac{\partial L}{\partial \mathbf{q}}.$$

In the case of Hookean springs we have:

$$\begin{aligned} L &= -\kappa \left[\text{diag}_i(1 - l_o/\|\mathbf{d}_i\|) \otimes I_2\right] \mathbf{d} \\ \text{(or equivalently)} &= -\kappa \text{diag}(\mathbf{d}) \left[\Downarrow_i (1 - l_o/\|\mathbf{d}_i\|) \otimes \mathbf{1}_{2\times 1}\right] \\ \Rightarrow \frac{\partial L}{\partial \mathbf{q}} &= -\kappa \left[\text{diag}_i(1 - l_o/\|\mathbf{d}_i\|) \otimes I_2\right] \frac{\partial \mathbf{d}}{\partial \mathbf{q}} \\ &\quad - \kappa \text{diag}(\mathbf{d}) \frac{\partial}{\partial \mathbf{q}} \left[\Downarrow_i (1 - l_o/\|\mathbf{d}_i\|) \otimes \mathbf{1}_{2\times 1}\right]. \end{aligned} \tag{8.1}$$

Since we have that $\mathbf{d} = K^T\mathbf{q}$ then the first derivative term of (8.1) is simply:

$$\frac{\partial \mathbf{d}}{\partial \mathbf{q}} = K^T.$$

The second derivative term of (8.1) is, for an edge i:

$$\frac{\partial}{\partial \mathbf{q}}\left[1 - \frac{l_0}{\|\mathbf{d}_i\|}\right] = -l_0\frac{\partial}{\partial \mathbf{q}}\left[\frac{1}{\|\mathbf{d}_i\|}\right].$$

Let

$$\mathbf{d}_i = (K^T)_{[2i:2i+1,:]}\mathbf{q} \doteq (K^T)_i\mathbf{q},$$

which implies

$$\|\mathbf{d}_i\| = \left([(K^T)_i\mathbf{q}]^T(K^T)_i\mathbf{q}\right)^{1/2}$$

and therefore that

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{q}}\left[\frac{1}{\|\mathbf{d}_i\|}\right] &= \frac{-1}{\|\mathbf{d}_i\|^2}\frac{\partial\|\mathbf{d}_i\|}{\partial \mathbf{q}} \\
&= -\frac{1}{2}\frac{1}{\|\mathbf{d}_i\|^3}\frac{\partial(\mathbf{d}_i^T\mathbf{d}_i)}{\partial \mathbf{q}} \\
&= -\frac{1}{2}\frac{1}{\|\mathbf{d}_i\|^3}\frac{\partial}{\partial \mathbf{q}}\left(\mathbf{q}^T[(K^T)_i]^T\left(K^T\right)_i\mathbf{q}\right) \\
&= -\frac{1}{\|\mathbf{d}_i\|^3}\mathbf{q}^T[(K^T)_i]^T\left(K^T\right)_i.
\end{aligned}$$

Putting everything together we have:

$$\frac{\partial L}{\partial \mathbf{q}} = -\kappa\left[\text{diag}_i(1 - \frac{l_0}{\|\mathbf{d}_i\|}) \otimes I_2\right]K^T - \kappa l_0\text{diag}(\mathbf{d})\left[\Downarrow_i \frac{1}{\|\mathbf{d}_i\|^3}\mathbf{q}^T[(K^T)_i]^T\left(K^T\right)_i \otimes \mathbf{1}_{2\times 1}\right]$$

which implies that

$$\begin{aligned}\frac{\partial f}{\partial \mathbf{q}} = &-\kappa K \left[\text{diag}_i(1 - \frac{l_0}{\|\mathbf{d}_i\|}) \otimes I_2\right] K^T \\ &- \kappa l_0 K \text{diag}(\mathbf{d}) \left[\Downarrow_i \frac{1}{\|\mathbf{d}_i\|^3} \mathbf{q}^T [(K^T)_i]^T \left(K^T\right)_i \otimes \mathbf{1}_{2\times 1}\right]\end{aligned} \tag{8.2}$$

and that

$$\frac{\partial g}{\partial \mathbf{x}} = I_N - (\Delta t) \begin{bmatrix} \mathbf{0}_{2n} & I_{2n} \\ M^{-1} \frac{\partial f}{\partial \mathbf{q}} & \mathbf{0}_{2n} \end{bmatrix}. \tag{8.3}$$

Note that the first term of $\partial f / \partial \mathbf{q}$ is a weighted graph Laplacian, similar in form to the derivation presented in [162].

8.2.2 Observation model

8.2.2.1 Enforcing edge constraints

We include as an additional constraint that the contour of the state should match, at least roughly, the contour of the image, to mitigate severe tracking drift. Other tracking methods incorporating contour information into the estimation include Brox, Rosenhahn and Cremers 2008[37]. Particle-filter based method, G-Flow [175], implements something quite similar, in that it explicitly models the background process, thus penalizing states for which there is discrepancy between the background and foreground pixels (as delineated by the object's contour). The problem of identifying and tracking the contour alone can be tackled with so-called 'active-contours', or snakes [136].

Here we add, as an additional observation, the mask of the image identifying the tracked object. For the cases we consider, simple thresholding on image intensity suffices to identify the mask in the observed frames, but other foreground/background subtraction methods may be appropriate in other cases.

8.2.2.2 The model

The observation process $\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k; \mathbf{D}, \mathbf{T})$ involves rendering a predicted frame, a predicted optic flow field and an object mask. The function $\mathbf{h}(\cdot)$ thus returns the predicted image $\mathbf{z}_k$, given by

$$\mathbf{z}_k(i,j) = \mathbf{T}(U^{-1}(i,j;\mathbf{x}_k\mathbf{D})),$$

for screen coordinates (i,j) and UV mapping function $U : \mathbb{R}^2 \to \mathbb{R}^2$ that takes UV coordinates and current pose and returns screen coordinates:

$$U(u,v;\mathbf{x}_k,\mathbf{D}) = (i,j).$$

Function $\mathbf{T} : \mathbb{R}^2 \to \mathbb{R}$ returns the intensity for a given UV texture coordinate. Rendering each face D_p within the frame uses linear interpolation of the initialized texture T_p and the current vertices.

The horizontal flow field is given by

$$\mathbf{j}_k^x(i,j) = \mathcal{F}^x(U^{-1}(i,j;\mathbf{x}_k,\mathbf{D})),$$

where $\mathcal{F}^x(\cdot)$ determines the flow by linearly interpolating the velocities at the vertices of the face indicated by inverse UV mapping $U^{-1}(\cdot)$. The vertical flow field $\mathcal{F}^y(\cdot)$ is given similarly.

Finally, to penalize models that drift outside of the observed image contour, we render a mask

$$\mathbf{m}_k(i,j) = \mathbf{I}(U^{-1}(i,j;\mathbf{x}_k,\mathbf{D})),$$

Computation of $\mathbf{y}_k$ is thus an expensive operation, though fortunately one that can be parallelized and executed on a GPU. Indeed the above description is a roundabout way of describing a rendering pipeline easily implementable with OpenGL.

8.2.2.3 Static observation noise

The observation covariance is considered to be a diagonal matrix

$$\sigma_\eta^2 = \begin{bmatrix} \sigma_z^2 I_m & & & \\ & \sigma_j^2 I_m & & \\ & & \sigma_j^2 I_m & \\ & & & \sigma_m^2 I_m \end{bmatrix},$$

where error in image observations are independent of error in the optic flow observations. Note that by changing the balance between σ_z^2, σ_j^2 and σ_m^2 we change the relative contribution of the template matching, optic flow and contour matching to the update step. Similar to the particle filter based method, G-flow [175], the model thus represents a generalization of both template matching based methods and optic flow based object tracking methods. Consistent, unbiased estimation of σ_η^2 could in principle be performed, for instance using an approach proposed by Mehra 1970[178]. Another option is to take a bootstrapping method such as Kondermann et al 2008 or Kybic et al 2011 [146, 151]. However, such approaches prove difficult in the high-dimensional cases. Thus, variances σ_j^2, σ_z^2 and σ_m^2 are considered parameters to be estimated based on model performance in test sets presented below.

8.2.2.4 Adaptive observation noise

More generally, we can have observation noise that depends on the state:

$$\sigma_\eta^2(\mathbf{x}) = \begin{bmatrix} \mathrm{diag}(\sigma_z^2(\mathbf{x})) & & & \\ & \mathrm{diag}(\sigma_{jx}^2(\mathbf{x})) & & \\ & & \mathrm{diag}(\sigma_{jy}^2(\mathbf{x})) & \\ & & & \mathrm{diag}(\sigma_m^2) \end{bmatrix},$$

for each $\sigma_z^2 : \mathbb{R}^N \to \mathbb{R}^m$, $\sigma_{jx}^2 : \mathbb{R}^N \to \mathbb{R}^m$ and $\sigma_{jy}^2 : \mathbb{R}^N \to \mathbb{R}^m$. This may be appropriate when there is a known relationship between the optic flow observation and the underlying state dynamics. This relationship which is explored below.

8.2.3 GPU implementation

The iterated form of the KF update,

$$\left(W_{k+1}^j\right)^{-1} = \left(W_{k+1}^-\right)^{-1} + H_{\hat{\mathbf{x}}_{k+1}^j}^T \left(\sigma_\eta^2\right)^{-1} H_{\hat{\mathbf{x}}_{k+1}^j} \tag{8.4}$$

$$\hat{\mathbf{x}}_{k+1}^{j+1} = \hat{\mathbf{x}}_{k+1}^0 + W_{k+1}^j H_{\hat{\mathbf{x}}_{k+1}^j}^T \left(\sigma_\eta^2\right)^{-1} \left(\mathbf{y}_{k+1} - \mathbf{h}(\hat{\mathbf{x}}_{k+1}^j)\right) - W_{k+1}^j H_{\hat{\mathbf{x}}_{k+1}^j}^T \left(\sigma_\eta^2\right)^{-1} H_{\hat{\mathbf{x}}_{k+1}^j} (\hat{\mathbf{x}}_{k+1}^- - \hat{\mathbf{x}}_{k+1}^j) \tag{8.5}$$

was derived because it is particularly convenient to parallelize and implement on a GPU. The matrix H represents a type of 'Jacobian image' [65], and in this case must be computed numerically, which is expensive. Further, it is $M \times N$, which for reasonable resolution images and mesh sizes is somewhere between cumbersome and totally infeasible – explicit construction must be avoided. One approach, taken by Dellaert et al 1998 [65], is to execute the updates pixel by pixel, instead of updating based on the entire image at once, iterating until the change in the updates is sufficiently small.

This is unnecessary here since, fortunately, H^T has the form:

$$H^T = \left[\begin{array}{cccc} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_2}{\partial x_1} & \cdots & \frac{\partial y_M}{\partial x_1} \\ \frac{\partial y_1}{\partial x_2} & & & \frac{\partial y_M}{\partial x_2} \\ \vdots & & & \vdots \\ \frac{\partial y_1}{\partial x_N} & \frac{\partial y_2}{\partial x_N} & \cdots & \frac{\partial y_M}{\partial x_N} \end{array}\right]_{\mathbf{x}=\hat{\mathbf{x}}_{k+1}^-} \approx \left[\begin{array}{cccc} \frac{(\tilde{y}_1(x_1)-\hat{y}_1)}{\Delta x} & \frac{(\tilde{y}_2(x_1)-\hat{y}_2)}{\Delta x} & \cdots & \frac{(\tilde{y}_M(x_1)-\hat{y}_M)}{\Delta x} \\ \frac{(\tilde{y}_1(x_2)-\hat{y}_1)}{\Delta x} & & & \frac{(\tilde{y}_M(x_2)-\hat{y}_{1M})}{\Delta x} \\ \vdots & & & \vdots \\ \frac{(\tilde{y}_1(x_N)-\hat{y}_1)}{\Delta x} & \frac{(\tilde{y}_2(x_N)-\hat{y}_2)}{\Delta x} & \cdots & \frac{(\tilde{y}_M(x_N)-\hat{y}_M)}{\Delta x} \end{array}\right]_{\mathbf{x}=\hat{\mathbf{x}}_{k+1}^-},$$

in which each row represents the 'differential image' given a change in state space, and which therefore permits a matrix-free approach. The component of the state update, $\mathcal{Z} = H^T(\sigma_\eta^2)^{-1}(\mathbf{y}_{k+1} -$

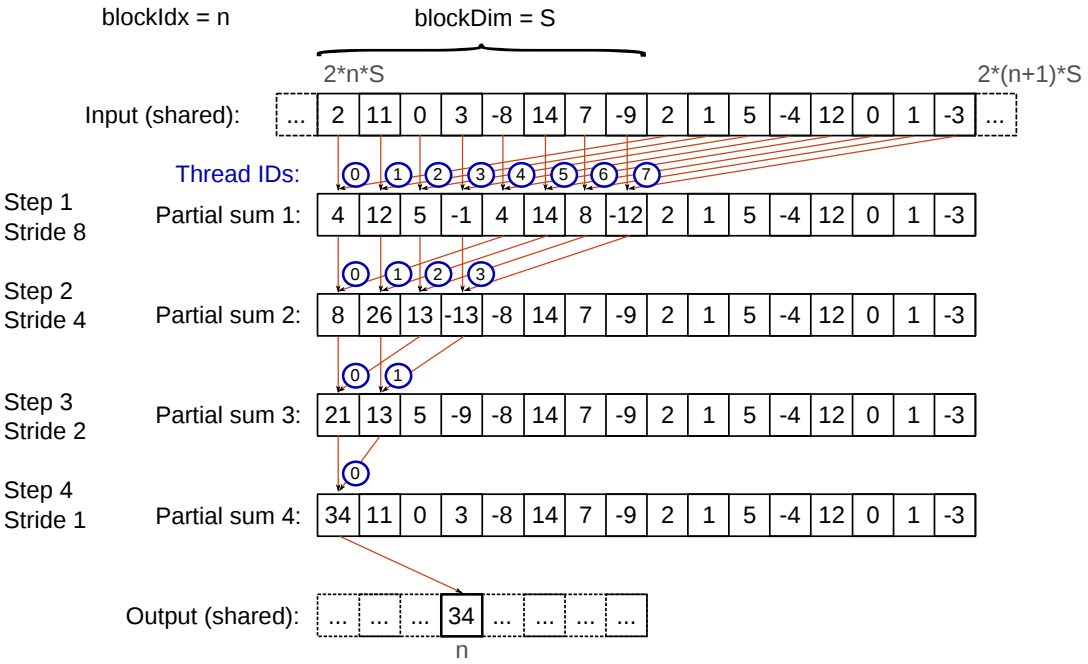


Figure 8.3: Computing a dot product using sequential addressing parallel reduction in CUDA. In words, N blocks of size S (a power of 2) are used to compute the reduction (dot product). The input vector, of length $2NS$ (padded with zeros if needed), is loaded into shared memory. Within each block, a step consists of each active thread summing two elements. Subsequent steps sum the preceding totals in a way that leaves all elements of the block added together at the final step. The block sum is written to an output vector and the N block sums are totaled on the CPU.

$\hat{\mathbf{y}}_{k+1}^{-}$), can thus be computed element-by-element in N rendering operations:

$$\mathcal{Z}_i = \sum_{j=1}^{M} H_{ij}^T (\mathbf{y}_{k+1,j} - \hat{\mathbf{y}}_{k+1,j}^{-}) / \sigma_{\eta,j}^2, \qquad i = 1, \ldots, N,$$

assuming that the measurement covariance σ_n^2 is diagonal having entries $\left\{\sigma_{\eta,m}^2\right\}_{m=1}^{M}$. Rendering $\hat{\mathbf{y}}_{k+1}^{-}$ and each $\tilde{\mathbf{y}}(x_i)$ in OpenGL and performing the dot product on the GPU, using CUDA (Compute Unified Device Architecture) or OpenCL, is fast and avoids GPU-CPU transfers. Parallel reduction [117] computes the dot product in $\mathcal{O}(\log M)$ parallel steps, so is well suited to large M (high resolution) observation spaces. The method is summarized in Figure 8.3, adapted from Harris et al 2007 [117].

The final term in Equation 8.5 and the update to W_{k+1} involves construction of the $N \times N$ matrix $\mathcal{H} = H^T(\sigma_\eta^2)^{-1}H$ which, naively, involves $\mathcal{O}(N^2)$ rendering operations to compute.

However, the elements of $\mathcal{H}$ take the form:

$$\mathcal{H}_{ij} = \sum_{k=1}^{M} H_{ik}^T H_{jk}^T / \sigma_{\eta,k}^2,$$

and thus each element represents the inner product between the differential images when state components x_i and x_j are varied. When i and j refer to vertices that are not connected according to the mesh $\mathcal{D}$ then they share no common face and their differential images will be orthogonal – $\mathcal{H}_{ij}$ will be zero.[4]

Therefore only elements of $\mathcal{H}$ involving vertices that are connected according to the mesh $\mathcal{D}$ need to be computed. The Delaunay triangulation used means vertices are only connected to their neighbors and thus the number of connected vertices scales approximately linearly with the total number of vertices. This means $\mathcal{H}$ can be computed in $\mathcal{O}(N)$ rendering operations also. The reduction to compute each $\mathcal{H}_{ij}$ can similarly be performed on the GPU with CUDA or OpenCl, increasing speed and saving GPU-CPU transfer.

The simple parameterization of the state space in terms of explicit vertex positions and velocities – instead of in terms of 'morph bases' or finite element method (FEM) deformation modes – is highly redundant, introducing many degrees of freedom that are unlikely needed. Fortunately, this redundancy can be taken advantage of via the resulting sparsity in $H^T(\sigma_\eta^2)^{-1}H$, which goes some way to offsetting the increased dimensionality of the state space – again relative to a model based on morph bases or similar.

8.2.3.1 Multiple perturbations with one render

Numerically estimating the matrix H involves perturbing the vertex positions and velocities by small increments. Significant gains can be made if multiple, independent perturbations are made within one render operation. For both the Jacobian and Hessian calculation this can be achieved by noting that the changes to images obtained by perturbing vertices that are not connected to

[4] Note that provided the noise covariance matrix is diagonal, the result holds independent of the specific values of $\sigma_{\eta,k}^2$. That is, orthogonal vectors under one inner product, $\langle \mathbf{u}, \mathbf{v} \rangle_S = \sum \mathbf{u}^T S \mathbf{v}$, for diagonal S, will be orthogonal under any other inner product $\langle \mathbf{u}, \mathbf{v} \rangle_{S'}$ of the same form, since all we're doing is rescaling coordinate axes.

one another in the mesh will occur in disjoint regions of the image. For the Jacobian calculation, we partition the vertices into groups such that each vertex within a group does not share an edge – and so can be perturbed independently within the one render operation. For this a queue is setup, consisting of each perturbation that needs to be made, and from which the partitions are generated (Algorithm 1).

Mathematically, we can express the idea as follows. Call E one such partitioning of the vertices, V, i.e.:

$$E = \{e_j\}_{j=1}^{J}, \quad e_j = \{v_i : v_i\}, \quad \cup_{j=1}^{J} e_j = V, \quad e_i \cap e_j = \emptyset \forall i \neq j.$$

Then for each element of the partition, $e_j = \left\{v_1^j, \ldots v_{d_j}^j\right\}$, let $B^j \in \mathbb{Z}^{n \times M}$ be the matrix of masks:

$$B^j = \begin{array}{r} \text{row } v_1^j \rightarrow \\ \\ \\ v_2^j \rightarrow \\ \\ v_{d_j}^j \rightarrow \\ \\ \end{array} \begin{bmatrix} \mathbf{m}_1 & \ldots \\ 0 & \ldots \\ \vdots & \ldots \\ \mathbf{m}_2 & \ldots \\ \vdots & \\ \mathbf{m}_{d_j} & \ldots \\ 0 & \ldots \end{bmatrix},$$

where $\mathbf{m}_i$ is the mask indicating whether each pixel in the frame is or is not affected by a perturbation in vertex v_i^j, that is, it is one for the pixels inside the faces $D(i)$ which contain vertex v_i^j and zero otherwise. We will perturb x and y coordinates separately and so define

$$B_x^j = B^j \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad B_y^j = B^j \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Recall our goal is to compute

$$\mathcal{Z} = H^T (\sigma_\eta^2)^{-1} (\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}^-).$$

We're free to compute entries in $\mathcal{Z}_j$ corresponding to vertices in e_j simultaneously. This is achieved by perturbing, separately in x and y, all vertices in e_j and computing $H^j \in \mathbb{R}^{M\times 1}$ from the one render operation, and then splitting it into the corresponding entries in $\mathcal{Z}_j$ using the mask matrices B_x^j and B_y^j:

$$\mathcal{Z} = \sum_j^J B_x^j \text{diag}(H_x^j)\text{diag}(\sigma_\eta^2)^{-1}(\mathbf{y} - \hat{\mathbf{y}}) + B_y^j \text{diag}(H_y^j)\text{diag}(\sigma_\eta^2)^{-1}(\mathbf{y} - \hat{\mathbf{y}}).$$

Perturbations to velocities are made in the same fashion. Thus, the computation of $\mathcal{Z}$ involves only $\mathcal{O}(J)$ rendering operations, instead of $\mathcal{O}(n)$. In fact, J will be roughly a constant as a function of n.

Note that the B^j are large matrices in themselves, but fortunately do not need to be constructed separately in practice. In practice the one image, $\mathcal{B}^j \in \mathbb{Z}^{1\times M}$ can be formed, which contains the superposition of all the other masks, something like:

$$\mathcal{B}^j = \begin{bmatrix} 1 & 2 & \dots & n \end{bmatrix} B^j,$$

which provides an image of the masks 'colored' by the numbers $1, \dots, n$ and whose coloring can be used to separate the perturbation components in the CUDA summation code.

Note that a simple parallel reduction here is no longer possible, as the data is categorized by its face. The calculation performed is thus more akin to counting elements of a given class, as in the creation of a histogram.

For the Hessian calculation we use the following similar algorithm (Algorithm 2). This saves a large number of render operations, making the renders sub-linear in the number of mesh points, and thereby allowing large meshes to be used.

8.2.4 Implementation

Run on computer with NVIDIA Tesla Fermi C2075 cGPUs, 32GB RAM. x2 6-core Intel Xeon E5-2640, 2.5GHz. Source code can be found at https://github.com/benlansdell/hydra

Algorithm 1 Computation of Jacobian×residual

$q =$ all mesh points needing to be perturbed
while q is not empty **do**
    $r = []$
    for $y \in q$ **do**
        if y is independent of all points in r **then**
            Add y to r, remove y from q
        end if
    end for
    Perturb all x-axis points in r
    Render
    Compute entries in Jacobian $\times$ residual vector
    Perturb all y-axis points in r
    Render
    Compute entries in Jacobian $\times$ residual vector
end while

Algorithm 2 Computation of Hessian

$q =$ all pairs of mesh points needing to be perturbed according to $\mathcal{D}$
while q is not empty **do**
    $r = []$
    for $(y_1, y_2) \in q$ **do**
        if both y_1 and y_2 are independent of all points in r **then**
            Add (y_1, y_2) to r, remove (y_1, y_2) from q
        end if
    end for
    Perturb all x-axis points in r
    Render
    Compute entries in $\mathcal{H}$
    Perturb all y-axis points in r
    Render
    Compute entries in $\mathcal{H}$
end while

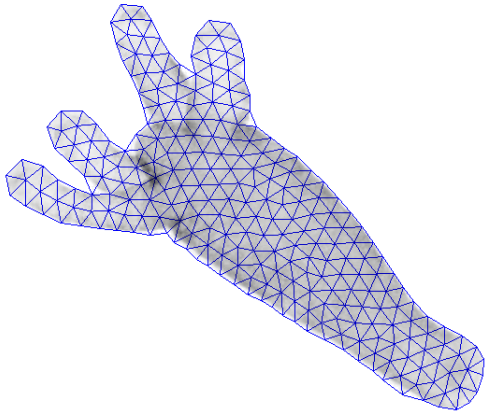


Figure 8.4: Delaunay triangulation initialization of mesh for tracking Hydra. Simple thresholding on image intensity provides an outline of the Hydra, after which DistMesh is used to generate a set of evenly distributed points in the interior and Delaunay triangulated. Image inverted.

8.3 Results

For each video the model is initialized using Delaunay triangulation. The outline of the Hydra is identified through simple thresholding on image intensity. The DistMesh method [204] is then used to generate a set of evenly distributed points in the interior which are then Delaunay triangulated. (Figure 8.4)

Following mesh initialization, the first frame of the video is used to construct the reference texture. The Kalman filter can then be executed, integrating each frame to provide an updated state estimate. The method is tried on test videos in which GCaMP6s is expressed in epithelial cells (which act as). On the test mesh demonstrated here, the model integrates approximately one frame per minute, for a mesh consisting of approximately 80 points. The method works well when Hydra motion is relatively slow, which elongations and tentacle waving taking place over 10s of frame. (First ~130 frames of Figure 8.5) The method struggles however in cases where the Hydra contracts its body rapidly over a few frames (~135 frame of Figure 8.5). Unfortunately this

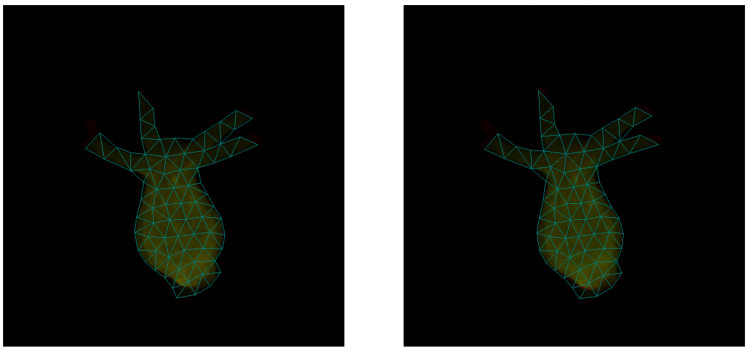
behaviour is common – to a first approximation the Hydra's behavior consists of bursts of rapid whole body contraction and slower elongation.

8.4 Discussion

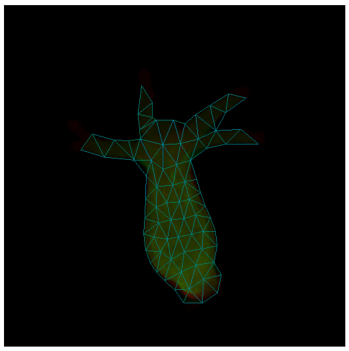
Thus the method works well for slow motion but fails for rapid motion. Further, the parameterization allows discontinuous (overlapping faces) meshes, which are unstable. These problems could likely be mitigated through using lower-dimensional parameterizations in terms of morph bases, either from a FEM model or from knowledge of typical Hydra positions. However, this may represent restrictive assumptions on the behavior of the Hydra we do not want to make – since its the behavior of the Hydra we are trying to identify. Nonetheless, a compromise between model stability and flexibility may exist.

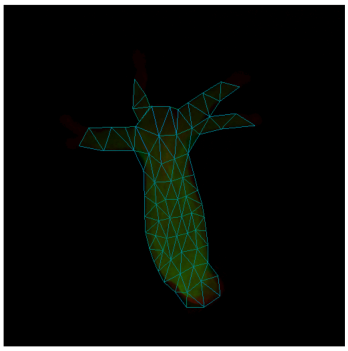
In addition to such a fix, numerous avenues need to be explored before this method is a demonstrably viable solution to the tracking problem. In particular, its time complexity as a function of mesh size and image resolution needs to be tested. The 80 frame mesh demonstrated here is acceptable for the muscle imaged videos tested, but more detailed meshes of hundreds of points would be required for videos in which neural activity is imaged. Further, its performance needs to be tested on datasets where ground truth data is available in order to validate the method.

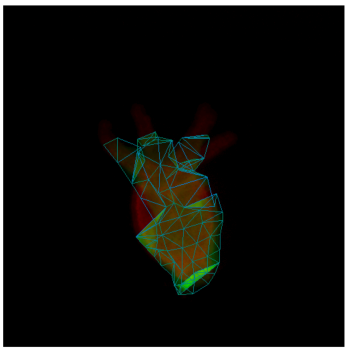
Nonetheless, the method provides a novel GPU implementation of the Kalman filter, reformulated in order to take advantage of GPU computations. Specifically, the GPU was utilized to perform the rendering and the subsequent matrix-matrix matrix-vector reduction related to the rendered images during updates. However the remainder of the implementation take place on CPU, in traditional serial fashion. It is worth mentioning that there is some work into more complete parallel implementations of Kalman filters utilizing GPU architecture. For instance Huang et al 2011 [128] perform matrix multiplications involved in the Kalman filter on GPU. In particular their method inverts the observation covariance matrix, which is $M \times M$ and thus takes the largest proportion of time in both CPU and GPU implementations. In the model presented here, the


Frame 15

Frame 45


Frame 75


Frame 105


Frame 135

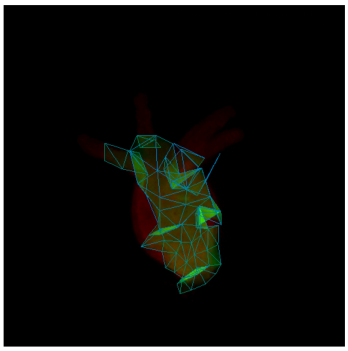

Frame 165

Figure 8.5: Example of Hydra tracking through 200 frames (20 seconds). Motion includes slow elongation of Hydra, which is capture well by the method. The rapid contraction (in a few frames, near frame 135) is difficult for the method to capture. This results in a discontinuous mesh – faces overlapping – which is an unstable configuration that cannot be recovered from.

largest matrix inversion that needs to be performed is $N \times N$ which, for large observation spaces (i.e. images), is significantly faster. Thus the method presented here should be implementable and faster in cases when there is a large observation space, such as in image processing applications, and in which is it reasonable to assume the observation noise is uncorrelated. Other work exploring similar ideas can be found in Rodr et al 2012 [218] and Lin et al 2016 [160].

8.5 Model notation

Please see Table 8.1.

Symbol	Description
Basics	
$\mathbf{x}$	state variable, $\mathbb{R}^N = \mathbb{R}^{4n}$
$\mathbf{q}$	state positions, $\mathbb{R}^{2n}$
$\mathbf{v}$	state velocities, $\mathbb{R}^{2n}$
$\mathbf{y}$	observation variable, $\mathbb{R}^M$
$\mathbf{z}$	observed image variable, $\mathbb{R}^m$
$\mathbf{j}_x, \mathbf{j}_y$	observed flows, $\mathbb{R}^m$
$\mathbf{m}$	observed mask, $\mathbb{R}^m$
$\mathcal{F}$	flow field observation function
$\mathbf{I}$	mask observation function
M, N	observation and state dimensions
Kalman filter	
$\hat{\mathbf{x}}_k, W_k$	estimated state and covariance
F, H	prediction and observation Jacobian
$\mathcal{Z}$	matrix vector product $\mathcal{Z} = H^T(\sigma_\eta^2)^{-1}(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}^-)$
$\mathcal{H}$	matrix-matrix product $H^T H$
$\sigma_\eta^2, \sigma_\epsilon^2$	observation Gaussian noise variance, additive acceleration Gaussian noise, diagonal matrices
$\sigma_z^2, \sigma_j^2, \sigma_m^2$	observation Gaussian noise variance components, scalars
$\mathbf{f}, \mathbf{h}$	non-linear dynamics and observation process
Q	process covariance matrix
$\hat{\mathbf{x}}_k^-, W_k^-$	predicted state and covariance matrix
$\mathbf{T}, U$	reference texture, UV mapping function for current pose
$\mathbf{D}$	set of mesh faces
V	set of mesh vertices
Mass-spring dynamics	
L	
$\mathbf{d}$	
l_i	
g	implicit equation solved with Newton's method
K	incidence matrix
l_0	rest spring length
κ	spring stiffness constant
M	vertex mass
E	partition of vertices into groups in disconnected, $E = \{e_j\}$
B^j	matrix with rows corresponding to masks for each vertex perturbed in subset e_j

Table 8.1: Model notation

9

A long term tracking method robust to occlusion and large deformation

9.1 Introduction

The preceding chapter described a state space model to do deformable object tracking. It was shown to work in cases of slow motion or not extreme deformations. By formulating the model as a recursively estimated filter an inherent instability is constructed. As soon as the tracking model significantly deviates from the object being tracked, it is difficult for it to recover. The frame-by-frame integration of the Kalman filter provides no mechanism to reinitialize its tracking procedure when needed.

The problem may be better tackled, then, not as a problem of *tracking*, but as one of *registration*. Along these lines, by having a mechanism that can recognize when the Hydra is in a similar

position, we can register frames that may be temporally distant but positionally related to each other. This provides a mechanism to, in a sense, reinitialize the tracking procedure at appropriate moments throughout a video. This chapter develops this basic idea into a tracking method that is robust to severe deformations and can be applied to long video sequences.

9.2 Background

9.2.1 A primal-dual algorithm for convex optimization

Before describing the method and the relevant image segmentation and optic flow methods we introduce some necessary material from convex optimization. Let X, Y be two finite-dimensional real vector spaces having inner produce spaces and norms $\langle ., . \rangle_X$,$\langle ., . \rangle_Y$ and $\|x\|_X^2 = \langle x, x \rangle_X$, $\|y\|_Y^2 = \langle y, y \rangle_Y$, respectively. Generally, whether the inner product applies to X or to Y can be inferred from the context so will not be denoted. Further let K be a linear map $K : X \to Y$.

The problem we solve has the general form:

$$\min_{x \in X} \max_{y \in Y} \langle Kx, y \rangle + G(x) - F^*(y),$$

where

$$G : X \to [0, \infty) = \text{lower semi-continuous (LSC), convex}$$
$$F^* : Y \to [0, \infty) = \text{LSC, convex.}$$

Note that this is the primal-dual version of the optimization:

$$\min_{x \in X} F(Kx) + G(x)$$

and in dual form:

$$\max_{y \in Y} -(G^*(-K^*y) + F^*(y)).$$

We denote by $(\hat{x}, \hat{y})$ the solution, which can be shown to satisfy:

$$\begin{aligned} K\hat{x} \in &\partial F^*(\hat{y}) \\ -(K^*\hat{y}) \in &\partial G(\hat{x}). \end{aligned}$$

In the above F^* and G^* represent the convex conjugates of functions $F(y)$ and $G(x)$, respectively, and K^* denotes the adjoint operator of K. We define the resolvent, or proximal, operator (the resolvent is a generalization of proximal operator for maximal monotone operators, but this distinction won't concern us here) of a convex function F as:

$$\begin{aligned} x &= (I + \tau\partial F)^{-1}(y) \\ &= \text{argmin}_x \left\{ \frac{\|x - y\|^2}{2\tau} + F(x) \right\} \\ &= \text{prox}_F(y; \tau) \end{aligned}$$

Note that if F is an indicator function then $\text{prox}_F(y; \tau)$ is Euclidean projection onto the convex set indicated by F.

Given the above preliminaries we describe what is sometimes called the Chambolle algorithm [48, 47, 46]. Generically, a proximal point algorithm has iterates of the following form:

$$x_{n+1} = (I + \lambda_n \partial F)^{-1}(x_n)$$

for convex function F. This can be seen as solving the 'root finding' equation

$$0 \in \partial F(x)$$

through a sort of backward Euler (implicit) gradient descent method.

The above may be generalized to a system of equations:

$$0 \in T(x_{n+1}) + M^{-1}(x_{n+1} - x_n).$$

The Chambolle algorithm has a similar structure to the proximal point algorithm, and is exactly a proximal point algorithm for certain parameter choices. In particular, choosing

$$T = \begin{pmatrix} \partial G(x_{n+1}) + K^* y_{n+1} \\ \partial F(y_{n+1}) - K x_{n+1} \end{pmatrix} \quad \text{and} \quad \text{M=} \begin{pmatrix} \mathcal{T}^{-1} & -K^* \\ -\theta K & \Sigma^{-1} \end{pmatrix}.$$

gives the following algorithm:

Algorithm 3 Chambolle's algorithm

Require: $\tau, \sigma \geq 0, \tau\sigma L^2 < 1, \theta \in [0,1], (x^0, y^0) \in X \times Y, \bar{x}^0 = x^0$

1: **while** not converged **do**
2: $y^{n+1} = (I + \sigma \partial F^*)^{-1}(y^n + \sigma(K\bar{x}^n))$
3: $x^{n+1} = (I + \tau \partial G)^{-1}(x^n - \tau K^* y^{n+1})$
4: $\bar{x}^{n+1} = x^{n+1} + \theta(x^{n+1} - x^n)$
5: **end while**

Thus application of the algorithm is straightforward, requiring simply computing the resolvent operators – the algorithm is therefore useful when these can be computed efficiently. We can easily generalize the algorithm to solve slightly more general problems of the form:

$$\min_{x \in X} \max_{y \in Y} \langle Kx, y \rangle + \langle g, x \rangle - \langle h, y \rangle + G(x) - F^*(y),$$

in which the algorithm simply becomes:

Algorithm 4 Slightly generalized Chambolle's algorithm

Require: $\tau, \sigma \geq 0, \tau\sigma L^2 < 1, \theta \in [0,1], (x^0, y^0) \in X \times Y, \bar{x}^0 = x^0$

1: **while** not converged **do**
2: $\quad y^{n+1} = (I + \sigma\partial F^*)^{-1}(y^n + \sigma(K\bar{x}^n - h))$
3: $\quad x^{n+1} = (I + \tau\partial G)^{-1}(x^n - \tau(K^* y^{n+1} - g))$
4: $\quad \bar{x}^{n+1} = x^{n+1} + \theta(x^{n+1} - x^n)$
5: **end while**

9.2.2 Applications to imaging

Let an image have dimensions $M \times N$, so that $X \in \mathbb{R}^{M \times N}$. Then we have the following operators

$$\langle u, v \rangle = \sum_{ij} u_{ij} v_{ij}$$

$$\nabla : X \to Y$$

$$(\nabla u)_{ij} = \begin{pmatrix} (\nabla u)^1_{ij} \\ (\nabla u)^2_{ij} \end{pmatrix}$$

$$\text{where} \quad (\nabla u)^1_{ij} = \begin{cases} u_{i+1,j} - u_{ij}, & i < M; \\ 0 & i = M \end{cases}$$

$$(\nabla u)^2_{ij} = \begin{cases} u_{i,j+1} - u_{ij}, & j < N; \\ 0 & j = N \end{cases}$$

With these, one can show that

$$\langle \nabla u, p \rangle_Y = \langle u, -\text{div} p \rangle,$$

where, importantly, we must use *backward* differencing for the div operator if we used *forward* differencing for ∇u. Stability of the algorithm depends on the norm L of the operator K:

$$L^2 = \|\nabla\|^2 = \|\text{div}\|^2 = 8.$$

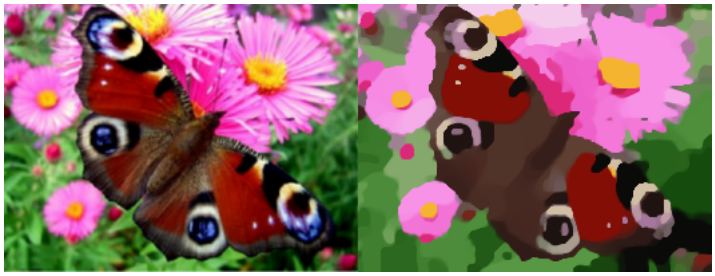


Figure 9.1: Example of image segment ion. The Mumford-Shah model estimates regions of piecewise-constant color such that the total variation (perimeter of the partition) is minimized. Here $k = 16$ colors used.

9.2.3 Review of Mumford-Shah segmentation

The basic idea behind Mumford-Shah image segmentation is to fix a set of k colors we are going to segment an image with in a way that avoids high spatial variability in color, while maintaining sharp transitions between regions of different color. The solution is to use a total-variation (TV) penalty. We let $\{c_i\}^k$ be our set set of colors and let $g(\mathbf{x}) : X \to \mathbb{R}_+$ be the function representing our image. Let $\mathbf{f} \in \mathbb{R}_+^k$ be the vector of distance between the image and each of our colors:

$$f_l = \lambda \int |g(\mathbf{x}) - c_l|^2 \, d\mathbf{x} \qquad l = 1, \ldots, k,$$

a measure of the error incurred by selecting color l for a given region of the image.

Let $\mathbf{u}(\mathbf{x}) \in \mathbb{R}^k$ be our vector that weighs contributions from each color c_i, to produce a final pixel intensity at $\mathbf{x}$. The segmented image can the be generated either as:

$$I(\mathbf{x}) = \mathbf{u}(\mathbf{x})^T \mathbf{c},$$

or as

$$I(\mathbf{x}) = \mathbf{c}(\text{argmax}_l u_l(\mathbf{x})).$$

We want the vector $\mathbf{u}$ to be *sparsely* varying as a function of $\mathbf{x}$. We can set this up as the following optimization problem

$$\min_{u \in U} \frac{1}{2} \sum_{l=1}^{k} \int_{\Omega} |Du_l| + \sum_{l=1}^{k} \int_{\Omega} u_l f_l \, dx \tag{9.1}$$

subject to particular constraints described below.

Why TV minimization? In Equation 9.1 $\int_{\Omega} |Du|$ denotes the total variation of the function u, with Du being the *distributional derivative* of u. For sufficiently smooth functions, $u \in W^{1,1}(\Omega)$, it reduces to:

$$\int_{\Omega} |Du| = \int_{\Omega} |\nabla u| \, dx.$$

There is a nice relation between total variation regularization and partition perimeter minimization. This is due to the fact that

$$\int_{\Omega} |Du| = \int_{-\infty}^{\infty} \text{Per}(\{u > s\}, \Omega) \, ds,$$

where $\text{Per}(\{u > s\}, \Omega)$ denotes the perimeter of the set $\{x : x \in \Omega, u(x) > s\}$. Thus the total variation of u can be thought of as equivalent to the sum of the surfaces of its level sets, and there is therefore a relation between minimizing TV and minimizing the length of the boundaries of regions of contiguous color.

Application to image segmentation With this in mind, let $R = \{R_l\}_{l=1}^{k}$ denote a partition of the domain Ω, with each partition having color c_i. The piece-wise constant, Mumford-Shah segmentation problem is then to solve

$$\min_{(R_l)_{l=1}^{k}, (c_l)_{l=1}^{k}} \frac{1}{2} \sum_{l=1}^{k} \text{Per}(R_l; \Omega) + \frac{\lambda}{2} \sum_{l=1}^{k} \int_{R_l} |g(x) - c_l|^2 \, dx.$$

The original specification treats the colors c_l as unknowns. This is related to a model from statistical physics known as the *Potts model*, a type of multi-spin Ising model. This problem can be

shown to be NP-hard [48]. However a convex problem can be obtained by treating c_l as known. (One easy way to choose sensible colors is to first choose a k and to perform k-means clustering to pick the c_ls.) We instead focus on a convex relaxation, the *vector* total variation minimization:

$$\min_{(u_l)_{l=1}^k} J(u) + \sum_{l=1}^{k} \int_\Omega u_l f_l \, dx, \quad u_l(x) \geq 0, \sum_{l=1}^{k} u_l(x) = 1, \forall x \in \Omega \tag{9.2}$$

where

$$f_l = \frac{\lambda}{2}|g(x) - c_l|$$

and $u(x)$ lies in the unit simplex:

$$u(x) \in U = \left\{ u_l(x) \geq 0, \sum_{l=1}^{k} u_l(x) = 1 \right\} \forall x \in \Omega$$

In the two color case ($k = 2$) this minimization and subsequent argmax projection in fact gives the exact same solution as the original MS problem, while for $k > 2$ the relation is not exact [194].

Different choices of energy $J(u)$ are possible. The simplest is to choose

$$J_1(u) = \frac{1}{2} \sum_{k=1}^{l} \int_\Omega |Du_l|,$$

and a more principled choice is

$$J_2(u) = \int_\Omega \Psi(Du),$$

where

$$\Psi(Du) = \sup_q \{\langle p, q : |q_l - q_m| \leq 1, 1 \leq l < m \leq k \rangle\}.$$

In this implementation we use the simpler functional $J_1(u)$. The derivative has the form:

$$|Du| = |\nabla u| = \sqrt{((\nabla u)_{ij}^1)^2 + ((\nabla u)_{ij}^2)^2}.$$

Thus, the minimization has the form

$$\min_{u\in U} \frac{1}{2}\sum_{l=1}^{k} |\nabla u| + \sum_{l=1}^{k} \langle u_l, f_l\rangle$$

From this we can identify $G(u) = \delta_U(u), g = f, F(p) = \frac{1}{2}\sum_{l=1}^{k} |p_l|$ and we can apply the Chambolle algorithm to the problem. Note that the conjugate of a norm is the indicator function of the dual norm unit ball. So we have

$$F^*(p) = \delta_P(p)$$

with

$$P = \left\{ p \in Y^k : \|p_l\|_\infty \leq \frac{1}{2} \right\}$$

Dualizing the $|\nabla u|$ gives the primal-dual problem:

$$\min_{u=(u_l)_{l=1}^k} \max_{p=(p_l)_{l=1}^k} \left(\sum_{l=1}^{k} \langle \nabla u_l, p_l\rangle + \langle u_l, f_l\rangle \right) + \delta_U(u) - \delta_P(p). \tag{9.3}$$

where the constraint has been expressed in terms of indicator function $\delta_U(u)$. As above, U is the pixel-wise unit simplex:

$$U = \left\{ u \in X^k : (u_l)_{i,j} \geq 0, \sum_{l=1}^{k} (u_l)_{i,j} = 1 \right\}$$

and P is the set

$$P = \left\{ p \in Y^k, \|p_l\|_\infty \leq \frac{1}{2} \right\}.$$

To apply the algorithm we need the proximal operators

$$\begin{aligned} \text{prox}_G(\tilde{u};\tau) &= \text{argmin}_u \left\{ \frac{\|u-\tilde{u}\|^2}{2\tau} + \delta_U(u) \right\} \\ &= \text{(per pixel) projection onto unit simplex} \end{aligned}$$

and

$$\begin{aligned}\text{prox}_{F^*}(\tilde{p};\sigma) &= \text{argmin}_p \left\{ \frac{\|p-\tilde{p}\|^2}{2\sigma} + \delta_P(p) \right\} \\ &= \text{(per pixel) projection onto unit balls}\end{aligned}$$

Thus, specifically

$$p_{ij}^l = \frac{\tilde{p}_{ij}^l}{\max(1, 2|\tilde{p}_{ij}^l|)}$$

with

$$|p_{ij}^l| = \sqrt{(p_{ij}^1)^2 + (p_{ij}^2)^2}.$$

Since these are both pixel-wise operators the implementation of the algorithm is straightforward on a GPU.

9.3 Long term tracking with multi-image optic flow and segmentation

Dense tracking of an object over an extended video sequence is an important problem in computer vision. Dense tracking attempts to find correspondence between points in a sequence of video frames. A common problem encountered in any tracking method is the accumulation or errors as the length of the video is increased – tracking drift. Few existing methods provide drift-robust tracking of generic objects throughout video. This Chapter provides a novel method of mitigating tracking drift.

Optical flow is one method of solving tracking problems. It enables a dense set of correspondence points to be estimated at sub-pixel accuracy. The method is typically applied to pairs of frames, though multi-frame optic flow estimation methods also exist. Optical flow, besides image brightness constancy, and motion continuity, makes few assumptions about the details of the

objects tracked, which makes it useful for deformable object tracking. It has thus found use in medical imaging and computer assisted surgery.

Previous methods to mitigate tracking drift include DeCarlo et al [64], which enforces objects stay within a detected contour.

More recent approaches include the works by Beeler et al [24] and Li et al [159]. Beeler et al [24] developed the notion of 'anchor frames' in order to robustly track faces. The model makes an assumption that the face will return to a neutral position, thus providing a set of 'anchor' positions. These frames can be identified and tracking throughout a video can then be broken into tracking for smaller blocks of video between these anchor frames. Similarly, Li et al also uses the notion of an anchor frame. However it generalizes the notion to be able to deal not just with faces, but also generic objects. Their extension relies on the ability to recognize not just anchor frames, but also smaller, local, 'anchor' patches which may be similar to one another and which can be used to perform long-term registration.

This chapter represents a further generalization to the method presented in Li et al. Their method only uses a single reference frame. However, not all parts of an object may be visible in a single reference frame, owing to self-occlusion or obstruction, and large deformations may make the appearance of an object at other times unrecognizable based on a single reference frame. In applications in which the object does not return to a 'neutral' position this becomes important. Robust tracking of such objects may require more than one reference frame. Our approach does not make a distinction between anchor frames and anchor patches, treating all correspondences as being potentially either localized or global.

9.3.1 The method

Similar to Beeler et al [24] and Li et al [159], the method breaks down a video into a set of blocks. Multi-frame optic flow [96] is applied independently to each block, using the initial frame of the block as the reference. These sets of frames, similar to video compression, will be referred to as interframes. Then, using continuity between adjacent interframes, and using registration

methods between the interframes and a set of reference frames we register paths that are tracked within each block to points in one of the reference frames, thereby performing registration and tracking. When combined with multi-frame optic flow, or other temporal registration/tracking methods, we can think of applying this to a set of regularly spaced frames from a video. If we can successfully recognize regions of frames which correspond to parts of the reference frames then we can perform registration and tracking of objects over extended video sequences, which may include occlusions and large deformations. By dividing the problem into two sub-problems – local tracking over short intervals, and registration throughout the entire video – the method's performance does not degrade as a function of video length.

The method relies on a way to register regions of two generic – not necessarily adjacent – frames with one another, and an accurate measure of how well we can do this. Similar frames can be mapped to one another using optic flow methods, which is the approach we take here. The method is therefore constrained by the performance and applicability of the optic flow method chosen. We use the *DeepFlow* [215] method to performs large-displacement optic flow estimation. It operates on the basis of a semi-dense set of correspondence points identified through a deep convolutional neural network (known as *DeepMatching* [215]). It provides state-of-the-art optic flow estimation, and the DeepMatching method can handle large deformations and repetitive textures [215].

To measure how well we can represent regions of one frame in terms of another we use a simple forward-backward optic flow estimation metric (applied in occlusion estimation methods and other related problems:[282, 141, 195, 17]). That is, between two images k and l let the estimated flow fields be $g_{kl}(x) \in \mathbb{R}^2$ and $g_{lk}(x) \in \mathbb{R}^2$, and let

$$f_{kl}(x) = |g_{lk}(g_{kl}(x)) - x|$$

be a measure of the error in optic flow tracking at a point x. The measure is based on the assumption that if we have estimated the flow fields accurately at a point x then g_{kl} and g_{lk} should be inverses (Figure 9.2).

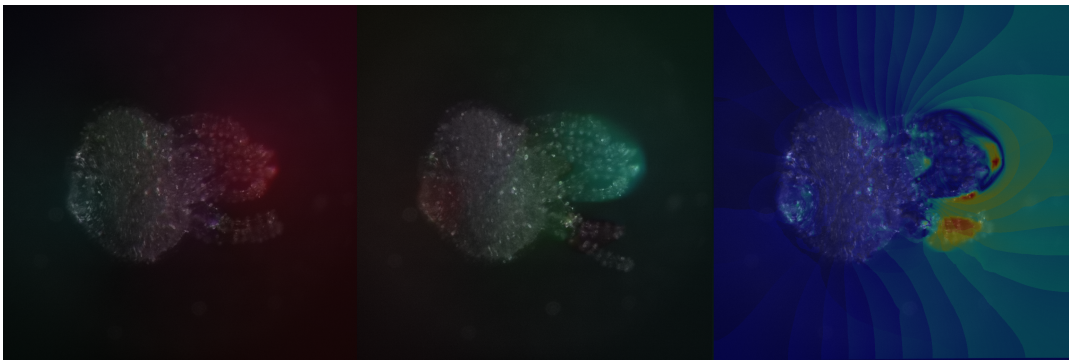


Figure 9.2: Left and center panels show forward and backward, respectively, optic flow estimation (g_{kl} and g_{lk}) overlaid on the two source video frames. Hue and saturation represent the direction and magnitude of the flow, respectively. The right panel shows the error magnitude $f_{kl}(x)$. Note that one tentacle was occluded in one but not the other source frame, and that this is reflected by the error field f_{kl}.

Note that, of course, each image reconstructs itself perfectly:

$$f_{l,l}(x) = 0, \forall 1 \leq l \leq K, x \in \Omega.$$

We use $f_{kl}(x)$ as a basis for determining identifications between regions of an interframe and a reference frame. One simple approach might be, at a position x, to take the reference frame having smallest error $\text{argmin}_k f_{kl}(x)$. However, simply choosing the minimum in this fashion provides pixel-level, and potentially noisy, registration. Ideally we would like larger, continuous regions to be registered with a reference frame. This motivates the use of total variation image segmentation methods described above.

Thus let $u_{kl}(x)$ be the proportion we use image k to represent image l at position x (of image l). Then, we can think of our reconstruction error as

$$\mathcal{L} = \frac{\lambda}{2} \sum \sum \int_\Omega u_{kl}(x) f_{kl}(x)\, dx = \frac{\lambda}{2} \sum \sum \langle u_{kl}, f_{kl} \rangle.$$

We impose a total variation penalty, and so we seek to find

$$\min_{u \in U} \frac{1}{2} \sum_{k,l=1}^{K} \int_\Omega |Du_{kl}|\, dx + \frac{\lambda}{2} \langle u_{kl}, f_{kl} \rangle,$$

with

$$U = \left\{ u^{kl}(x) \geq 0, \quad \sum_{k=1}^{K} u_{kl}(x) = 1 \forall l, x \in \Omega \right\}.$$

This, of course, simply breaks into K independent problems of the form solved in the previous section. But we want to use fewer than K frames to perform our reconstruction. Indeed, if we were using all K frames then the solution is trivial

$$u_{kk}(x) = 1 \quad \forall x, k$$

and thus $\mathcal{L} = 0$.

Here, we employ two strategies to identify fewer than K reference frames:

1. **Approach 1:** Choose by hand a set of reference frames $K < L$ and perform image segmentation on each non-reference image.

2. **Approach 2:** Use a group LASSO penalty to perform the segmentation and reference frame selection simultaneously.

Approach 1 is a straightforward application of the image segmentation method to each frame and needs no further attention. It may be feasible or desirable for some applications, and is of course the simplest approach.

Approach 2 requires the additional of a group LASSO penalty, which does need further attention. The penalty takes the form of a $L_{2,1}$ norm:

$$\sum_{k=1}^{K} \left[\sum_{l=1}^{K} (u_{ij}^{kl})^2 \right]^{1/2}$$

thus

$$\mathcal{L} = \sum_{k=1}^{K} \int_{\Omega} \frac{1}{2} \sum_{l=1}^{K} |Du^{kl}| \, dx + \frac{\lambda_1}{2} \sum_{l=1}^{K} \langle u^{kl}, f^{kl} \rangle + \frac{\lambda_2}{2} \left[\sum_{l=1}^{K} (u_{ij}^{kl})^2 \right]^{1/2}$$

and, as above, we seek

$$\min_{u \in U} \mathcal{L}(u; \lambda_1, \lambda_2).$$

We explore one way of computing this minimization. To begin, we make the identification

$$\begin{aligned} F(p) &= \sum_{k=1}^{K} \frac{1}{2} \int_{\Omega} \sum_{l=1}^{K} |Du^{kl}| \, dx \\ g(u) &= \sum_{k=1}^{K} \frac{\lambda_1}{2} \sum_{l=1}^{K} \langle u^{kl}, f^{kl} \rangle \\ G(u) &= \sum_{k=1}^{K} \frac{\lambda_2}{2} \left[\sum_{l=1}^{K} (u_{ij}^{kl})^2 \right]^{1/2} + \delta_U(u) \end{aligned}$$

from which it is clear we can apply the Chambolle algorithm. We need the resolvents. We have

$$F^*(p) = \delta_P(p)$$

with

$$P = \left\{ p \in Y^k : \|p^{kl}\|_\infty \leq 1/2, \forall k, l \right\}$$

Thus this resolvent, as before, takes the form of a projection onto unit balls per pixel. The resolvent for $G(u)$ is now more involved. We need to solve the minimization

$$\text{prox}_G(\tilde{u}) = \text{argmin}_u \left\{ \frac{\|u - \tilde{u}\|_F^2}{2\sigma} + \delta_U(u) + \frac{\lambda_2}{2} \sum_{k=1}^{K} \left[\sum_{l=1}^{K} (u_{ij}^{kl})^2 \right]^{1/2} \right\}$$

which takes the form of a constrained group LASSO problem [114].

We solve this with ADMM. We let v be our dual variable, such that we seek

$$\min_{u,v} \sum_{k=1}^{K} \int_{\Omega} \frac{1}{2} \sum_{l=1}^{K} |Du^{kl}| \, dx + \frac{\lambda_1}{2} \sum_{l=1}^{K} \langle u^{kl}, f^{kl} \rangle + \frac{\lambda_2}{2} \left[\sum_{l=1}^{K} (v_{ij}^{kl})^2 \right]^{1/2}$$

subject to the constraint $u = v$. To do this we form the augmented Lagrangian:

$$\mathcal{L}(u, v, \Lambda) = \frac{\|u - \tilde{u}\|_F^2}{2\sigma} + \delta_u(u) + \frac{\lambda_2}{2} \sum_{k=1}^{K} \left[\sum_{l=1}^{K} (v_{ij}^{kl})^2 \right]^{1/2} + \langle \Lambda, u - v \rangle + \frac{\rho}{2} \|u - v\|_F^2$$

where we solve, in turn:

$$u_{k+1} = \text{argmin}_u \mathcal{L}(u, v_k, \Lambda_k) \tag{9.4}$$

$$v_{k+1} = \text{argmin}_v \mathcal{L}(u_{k+1}, v, \Lambda_k) \tag{9.5}$$

$$\Lambda_{k+1} = \Lambda_k + \rho(u_{k+1} - v_{k+1}) \tag{9.6}$$

The first step, (9.4), is a simple per-pixel projection onto the unit simplex. The final step,(9.6), is of course trivial. The second step,(9.5), requires our attention. This step has the form of a group LASSO optimization. The proximal operator the group LASSO operator known, so this can be solved.

Specifically, for (9.4), we solve:

$$u_{k+1} = \text{argmin}_u \frac{\|u - \tilde{u}\|_F^2}{2\sigma} + \delta_U(u) + \langle \Lambda, u \rangle + \frac{\rho}{2}\|u - v\|_F^2$$

where we have dropped irrelevant terms to the minimization. We can express this as a per-pixel per image optimization:

$$u^{.l}_{k+1(ij)} = \text{argmin}_u \frac{\|u^{.l}_{ij} - \tilde{u}^{.l}_{ij}\|_F^2}{2\sigma} + \delta_{U^{.l}_{ij}}(u^{.l}_{ij}) + \langle \Lambda^{.l}_{ij}, u^{.l}_{ij} \rangle + \frac{\rho}{2}\|u^{.l}_{ij} - v^{:.}_{ij}\|_F^2$$

which we write as

$$u^{.l}_{k+1(ij)} = \text{argmin}_u \sum_{k=1}^{K} \left((u^{kl}_{ij} - \tilde{u}^{kl}_{ij})^2 + \frac{\rho}{2}(u^{kl}_{ij} - v^{kl}_{ij})^2 + \Lambda^{kl}_{ij} u^{kl}_{ij} \right) + \delta_{U^{.l}_{ij}}(u^{.l}_{ij}).$$

Each of the k terms simplifies to (dropping the indices in the RHS for brevity)

$$\begin{aligned}
(u^{kl}_{ij} - \tilde{u}^{kl}_{ij})^2 + \Lambda^{kl}_{ij} u^{kl}_{ij} + \frac{\rho}{2}(u^{kl}_{ij} - v^{kl}_{ij})^2 &= u^2 - 2u\tilde{u} + \tilde{u}^2 + \Lambda u + \frac{\rho}{2}(u^2 - 2uv + v^2) \\
&= (1 + \frac{\rho}{2}) \left[u^2 + \frac{(\Lambda - 2\tilde{u} - \rho v)u}{(1 + \rho/2)} \right] + \text{terms not involving u} \\
&= (1 + \frac{\rho}{2}) \left[u + \left(\frac{(\Lambda - 2\tilde{u} - \rho v)}{(1 + \rho/2)} \right) \right]^2 + \text{terms not involving u}
\end{aligned}$$

Which tells us we seek (again in vector form)

$$\text{argmin}_u \|u - \frac{(2\tilde{u} + \rho v - \Lambda)}{2+\rho}\|_F^2 + \delta_U(u)$$

which, as stated above, is a projection onto the unit simplex.

For (9.4), we solve

$$\begin{aligned} v_{n+1} &= \text{argmin}_v \mathcal{L}(u_{n+1}, v, \Lambda_n) \\ &= \text{argmin}_v \frac{\lambda_2}{2} \sum_{k=1}^{K} \left[\sum_{l=1}^{K} (v_{ij}^{kl})^2 \right]^{1/2} - \langle \Lambda, v \rangle + \frac{\rho}{2} \|u - v\|_F^2 \end{aligned}$$

We again rearrange, dropping indices for brevity, to obtain

$$\begin{aligned} \frac{\rho}{2}(u-v)^2 - \Lambda v &= \frac{\rho}{2}u^2 - \rho uv + \frac{\rho}{2}v^2 - \Lambda v \\ &= \frac{\rho}{2}\left[v^2 - 2uv + 2\Lambda v/\rho\right] + \text{terms not involving v} \\ &= \frac{\rho}{2}\left[v - (u + \Lambda/\rho)\right]^2 + \text{terms not involving v} \end{aligned}$$

giving

$$\text{argmin}_v \frac{\rho}{2}\|v - (u + \Lambda\rho)\|_F^2 + \frac{\lambda_2}{2} \sum_{k=1}^{K} \left[\sum_{l=1}^{K} (v_{ij}^{kl})^2 \right]^{1/2}$$

which is in group LASSO proximal operator form. In general the group LASSO proximal operator is

$$\text{prox}_{GL}(W; \rho, \lambda_2)_k = \frac{W_{\cdot\cdot}^{k\cdot}}{\|W_{\cdot\cdot}^{k\cdot}\|_F} \left(\|W_{\cdot\cdot}^{k\cdot}\|_F - \frac{\lambda_2}{2\rho} \right)_+ .$$

In our case we use $\text{prox}_{GL}(u + \Lambda\rho; \rho, \lambda_2)$. So with this we can compute the ADMM iterates.

The other possible approach, instead of breaking down the Chambolle step into a ADMM sub-optimization, is perhaps to dualize the variable in the LASSO term. However this possibility requires further exploration.

Regardless of whether approach 1 or 2 is used, the resulting method can now be summarized as follows: (Figure9.3)

1. Split video into equal sized blocks. Perform multi-frame optic flow separately on each block.

2. Use DeepMatching and DeepFlow to identify blocks that are positionally related to one another. Construct DeepFlow error signal $f_{ij}(x)$.

3. From each block select a reference frame. By hand (Approach 1) or in a semi-automatic fashion (Approach 2).

4. Register regions of interframes with reference frames using image segmentation.

5. Using multi-frame optic flow (Step 1) and registration (Step 4), associate each locally tracked path with a point in one of the reference frames.

9.3.2 Continuity between adjacent interframes

The above described method only performs registration between interframes and reference frames, not between interframes and other interframes. However in some cases the tracking between adjacent interframes may be significantly better than a direct registration between interframes and reference frames. So this section presents an extension to the method that allows for paths to be registered either directly with a reference frame, or to be registered with a point in an adjacent interframe. Note that, to perform tracking through-out the whole video, each path will be associated with a point in a reference frame, the question is through which interframes this ultimate registration is to be made? This extension complicates the segmentation procedure.

First, between adjacent interframes we measure the multi-frame optic flow tracking error through the same forward and reverse flow metric. Designate this error by

$$c_{lk}(x) = |d_{lk}(d_{kl}(x)) - x|$$

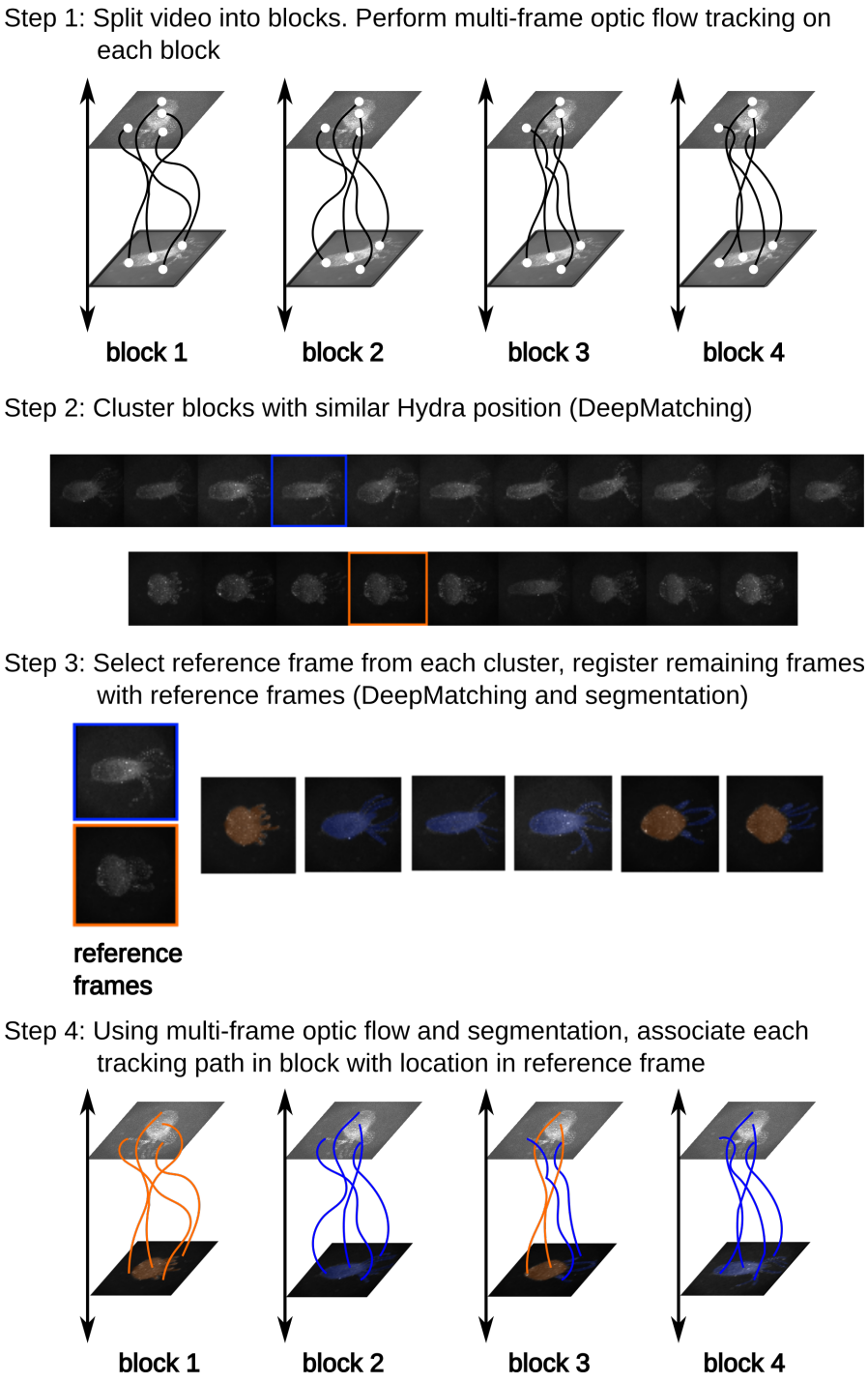


Figure 9.3: Proposed long-term tracking method.

where $l = k + 1$ or $l = k - 1$ (i.e. an adjacent interframe), and $d_{lk}(x)$ is the estimation of the optic flow as performed by the multi-frame optic flow method for the block between interframes l and k. By allowing registrations between interframes we extend the number of possible routes an association can be made with a point in a reference frame. Indeed we now need to consider all possible routes an association could be made.

To make this concrete, consider the following simple example. Assume that blocks 5 and 7 are designated reference frame blocks. A point x in block 2's interframe can be associated with a point in block 5 or 7 through one of the following composite registrations:

- Associate with block 5 straight through multi-frame optic flow continuation $d_{45}(d_{34}(d_{23}(x)))$
- Associate with block 4 through multi-frame optic flow continuation, then with either block 5 or 7 through DeepMatching estimation $g_{45}(d_{34}(d_{23}(x)))$ or $g_{47}(d_{34}(d_{23}(x)))$
- Associate with block 3 through multi-frame optic flow continuation, then with either block 5 or 7 through DeepMatching estimation $g_{35}(d_{23}(x))$ or $g_{37}(d_{23}(x))$
- Associate with block 5 or 7 through DeepMatching estimation $g_{25}(x)$ or $g_{27}(x)$
- Associate with block 1 through multi-frame optic flow continuation, then with either block 5 or 7 through DeepMatching estimation $g_{15}(d_{21}(x))$ or $g_{17}(d_{12}(x))$

The process is summarized in Figure 9.4.

In general, the number of paths that need to be considered will scale roughly linearly with the number of interframes. Precisely, for K interframes and N reference frames $\{n_i\}_{i=1}^{N}$ and an interframe k, the number of paths that need to be considered is

$$N(\min_{n_i>k}(n_i) - \max_{n_i<k}(n_i)).$$

For long videos this may be an infeasible number of paths to consider. A workaround in this case would be to cap the number of multi-frame optic flow continuations that are permitted at some

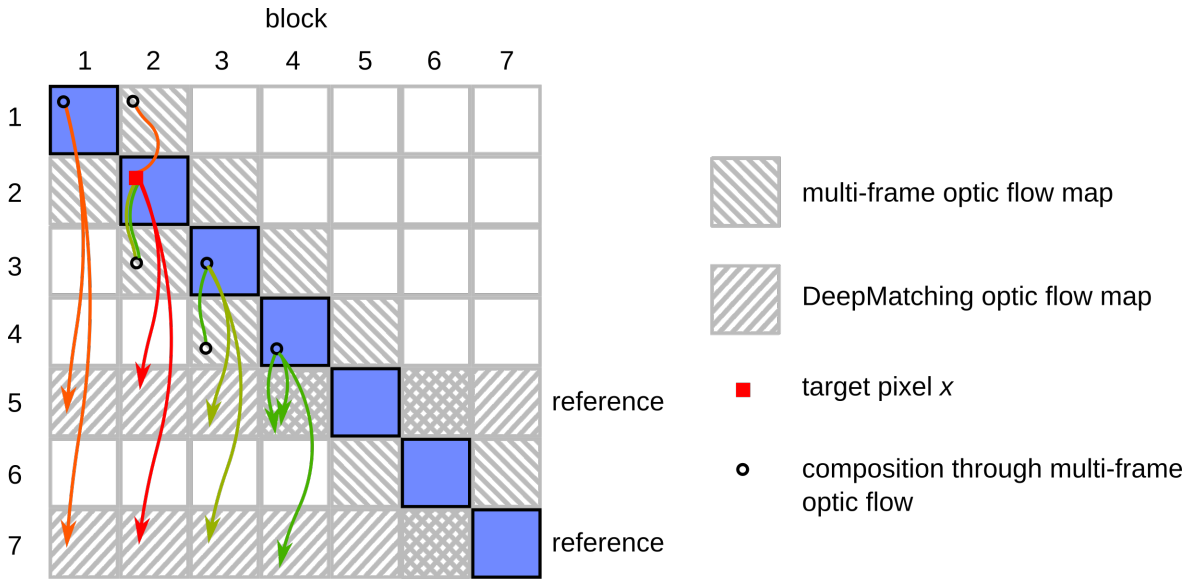


Figure 9.4: Example registration method with 7 blocks and 2 reference frames. A target pixel x (red square) from the interframe of block 2 is registered with one of the two reference frames (5 or 7). A combination of registration through composition of multi-frame optic flow tracking from one adjacent block to the next, or direct registration with the reference frames through a two-frame optic flow method (here DeepMatching), means multiple paths can be taken to register pixel x. These paths are shown as separately colored lines. An error measure for each path is determined through the sum of the individual errors of each of the registrations that form the path composition. Image segmentation is then perform on these error signals to select the path taken for each target pixel in each interframe.

value n_{max}. In which case the maximum number of paths that would need to be considered for any interframe k would be

$$2Nn_{max}.$$

Once the set of paths to be considered has been identified, a method to choose which path to choose is required. For this we need to extend our error measure $f_{kl}(x)$ and $d_{kl}(x)$ to compositions. The simplest approach, for a composite registration, is to take the sum of the errors of each of the individual maps estimated. Thus, a general composition from interframe l to reference frame k may consist of p multi-frame optic flow continuations, $\{q_i\}_{i=0}^{p}$ where $q_0 = l$ and $q_i = q_{i-1} + 1 \forall i$ or $q_i = q_{i-1} - 1 \forall i$, following by a DeepMatching registration:

$$g_{q_p,k} \circ d_{q_{p-1},q_p} \, [p \text{ times}] \circ d_{q_0,q_1}(x).$$

With this composition we can associate the error:

$$\tilde{f}_{kl}(x) = f_{q_p,k}(x_{p+1}) + \sum_{i=1}^{p} d_{q_{i-1},q_i}(x_i)$$

where x_i is the position of the constructed registration after i compositions.

We note that two simple extensions are possible here. First, we may weigh errors from one method more than the other, thus favoring registrations made through one more than the other. For instance, we may value continuity between adjacent blocks higher, in which case we may want to weight errors made through the multi-frame optic flow tracking *less*. We can introduce a simple weight α_M to accommodate this. Second, we may take simple non-linear functions of the two contributions to the error signal. For example, choosing a sigmoidal function or simple threshold function in one or both of the terms can encourage behavior such as: follow the multi-frame optic flow tracking, until errors exceed a given threshold, after which resort to the DeepMatching

registration. This may be desirable in some cases. This gives the more general form:

$$\tilde{f}_{kl}(x) = \Gamma\left[f_{q_p,k}(x_{p+1})\right] + \alpha_M \Lambda\left[\sum_{i=1}^{p} d_{q_{i-1},q_i}(x_i)\right],$$

for non-linear scalar functions $\Gamma(\cdot)$ and $\Lambda(\cdot)$. After this error signal is constructed for each of the interframe, image segmentation and registration can proceed as above.

9.4 Results

The method is applied to an extended video sequence with GCaMP6 labeled neurons in Hydra, consisting of 5000 frames (500 seconds). Interframes are selected every 250 frames. To determine how well these interframes can be matched to one another, the DeepMatching error metric is computed between every pair of interframes, and the measure $f_{ij}(x)$ is averaged over each entire image. This provides a frame-wide measure of frame similarity. This similarity matrix has a periodic structure that reflects the periodicity in the Hydra's behavior – it alternatives at fairly regular intervals between a contracted and elongated state. The similarity matrix is clustered to reveal two dominant clusters, corresponding in general to a contracted and elongated pose, confirming that the measure does indeed recognize when the Hydra is in a similar position (Figure 9.5 and Figure9.6).

In a shorter segment of video, two reference frames are selected by increasing the group LASSO penalty parameter until only two frames are used for the registration. Once reference frames are chosen the image segmentation is performed. Some frames are best continued and some are best registered to the reference frames. This is combined with the multi-frame optic flow results to obtain tracking through-out the entire video (Figure 9.7).

The video demonstrates that the method is able to successfully recognize when the Hydra has return to its initial contracted configuration and associate points at these later times to points at the earlier times, regardless of the magnitude and duration of any intervening motion – thus providing a robust tracking approach.

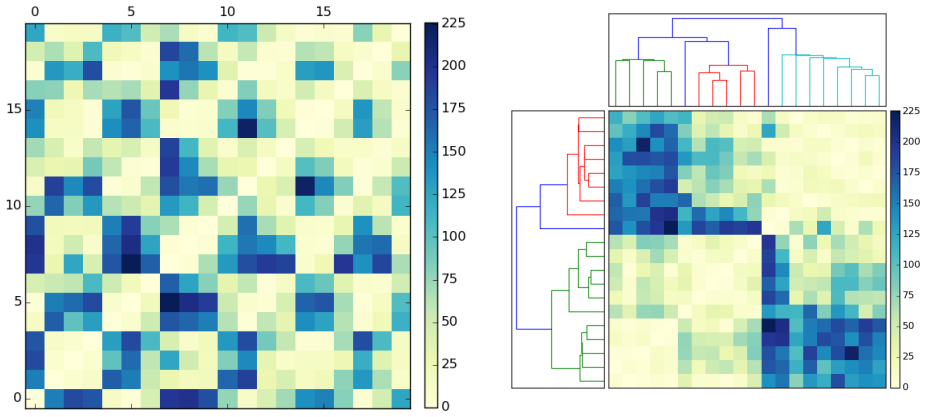


Figure 9.5: Similarity between frames measured according to mean similarity metric $\langle f_{ij}(x) \rangle$. Left panel shows similarity with regularly spaced set of frames (every 250) from Hydra video sequence. Right panel shows this similarity matrix hierarchically clustered. Clustering reveals two dominant clusters. Examining the frames within these clusters shows an extended and contracted Hydra poses. The method can thus better map contracted frames to one another, and elongated frames to one another, as would be expected.

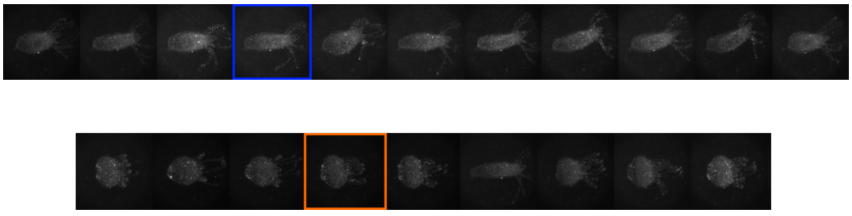


Figure 9.6: Interframes from the two dominant clusters. Generally, the clusters correspond to elongated poses, and to contracted poses.

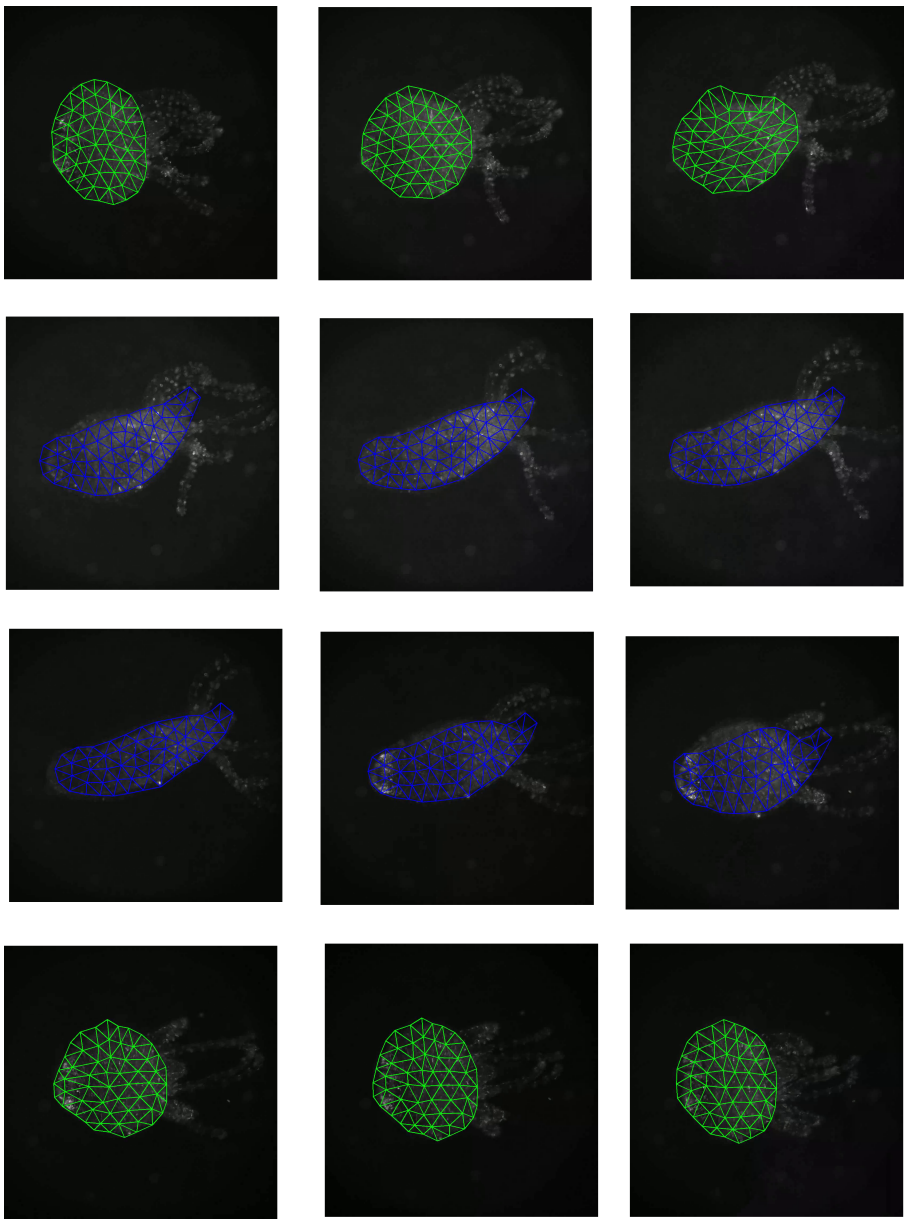


Figure 9.7: Tracking Hydra with proposed optic flow/registration method. Two reference frames are chosen and for each a contour is identified based on intensity thresholding. From this contour a mesh is initialized. The two meshes are propagated through each of the blocks associated with their respective reference frames. The result is tracking in a contracted state (top row; green mesh), before the elongation means the multi-frame optic flow fails (top right frame), and the method switches to an elongated reference frame (middle two rows; blue mesh). When the Hydra contracts again, the method detects it can switch back to the contracted reference frame (bottom row). Note the tracking is dense, for every pixel, the non-dense meshes are used here just to aid visualization.

9.5 Discussion

As in the previous chapter, a number of things need to to be performed before this approach can be demonstrated as a viable solution to Hydra body tracking. First, a ground truth dataset needs to be used to validate the model. This can be obtained from Garg et al 2013 or Li et al 2016[96, 159]. Second, the effect of different non-linearities in the composition cost function needs to be investigated. Third, the method needs to be tried on other datasets besides Hydra if it is to be valuable as a general tracking and registration method. As the method has the potential to handle occlusions it should be tested on datasets in which parts of the object of interest are not always visible. An interesting application would be tracking different an object as it rotates and different sides become visible at different times. In principle the method should be able to recognize these different sides and track them over an extended sequence.

It is important to note that the method provides a way for tracking the body of the Hydra, however unless the registration/tracking methods used are extremely accurate, then this method alone is not sufficient to solve the neuron registration problem. For this dataset the accuracy requires is beyond state-of-the-art tracking methods, so there will still be small movements in a motion compensated video that may make association of neurons very close to one another ambiguous. It is better thought of as a motion correction preprocessing step to be used in a neuron registration method. Once motion has been approximately accounted for using this method, it can then be input into a neuron registration method, similar to the registration method in Nguyen et al [191].

One issue is that, in the data presented here, neurons may not be visible for the entire recording, since it is their calcium activity that is imaged, and this is only visible when the neurons are active. This means that Nguyen's method cannot be applied out of the box: even on motion corrected videos, there remains an ambiguity between a previously tracked neuron becoming visible (trackable) again, and a previously unobserved neuron becoming visible for the first time in a region close to previously observed neurons (possibly owing to fluctuations or errors in motion

correction). Nonetheless, candidate approaches here include adopting the clustering method of Nguyen et al to this case, or adopting some other form of point-set registration (e.g. [189]) on the motion corrected frames and a set of reference frames. Since the motion has already been corrected for using the above method, the permissible deformations in the point-set registration could be made more rigid, which may help resolve some of the above ambiguity. Additional information about the detections (e.g. their radius, which should be fairly static) could also be added to the registration method as a way of handling further ambiguities.

Another issue with the Hydra data that makes it particularly challenging is that, due to the transparency of the organism, both a front and a back side may be visible at the same time. This means that there may be relative motion between the front and back layer that will manifest as a violation of the motion continuity assumed by all optic flow and point-set registration methods. Thus at the moment any relative motion between front and back layers creates errors in registration. This would not be an issue if particle tracking methods were applied, however particle tracking methods are not viable in cases where a neuron is visible, then due to inactivity becomes invisible. Reassociating the neuron with its original particle track when it becomes visible again requires a very accurate measure of where the neuron is in the body and the position of the body at all times. When front and back layers move relative to one another, these layers need therefore be tracked separately. A possible way of achieving this is through recently published dual-layer optic flow estimation methods (e.g. Yang et al 2016 [283]).

Regardless, the method presented here is an important step in solving the neuron tracking problem in this challenging case, and may find application in a number of tracking problems under occlusion and large deformation.

10

Conclusions

"*In its ideas about itself... as in all of its other endeavors, the mind goes from mastery to enslavement. By an irresistible movement, which imitates the attraction death exercises over life, thought again and again uses the instruments of its own freedom to bind itself in chains.*"

—Roberto Unger, *Knowledge and Politics*

This thesis has explored, in a number of guises, how neural activity may come to represent features of the external world and behavior, and how this understanding may inform an application of brain-computer interfaces. The last section of the thesis focused on developing methods that allow for tracking and extraction of neural activity from a significant proportion of an organism's neurons. In general, such studies have potential to change our view of nervous system function,

and may change how we view some aspects of neural encoding: if more complete views of neural activity in, say, motor cortex can be obtained then will encoding relationships between reach direction and primary motor units come to be seen as epiphenomenal? If more satisfactory mechanistic models can be created instead, does the role of representation and computation remain the same?

Computational neuroscience is a multi-faceted discipline, comprising many different methods and analyses from computer science, theoretical physics, applied mathematics, biophysical modeling/simulation, etc. This work has provided a limited cross-section: the retinal waves work utilized theoretical models and analysis, while the dual-control BCI work obviously incorporated experimental analysis and computational/statistical analysis, and the Hydra project presented computational methods that will one day aid in analyzing optical imaging experiments. Software engineer Jim Gray would describe these analyses as lying somewhere within three 'traditional' paradigms of science – theory, experiment, computation. Gray also identifies a fourth, nascent paradigm that could be loosely called data science/machine learning, separate from the other three [123]. In this paradigm more weight may be given to identifying patterns and structure from large quantities of data with less explicit incorporation of domain specific knowledge – "it's the data, stupid." Large datasets, and accompanying methods to deal with the quantity of data, are thus necessary. Often questions of prediction benefit most from these approaches, while questions of inference may benefit less so. Many domains in science, neuroscience being no exception, are undergoing a transition from data-poor to data-rich, with an accompanying shift towards data-intensive approaches. But there is more to science than prediction, and each field must grapple with how to incorporate vast quantities and novel qualities of data in order to also provide scientific insight.

For its part, neuroscience is changing rapidly. Neuroscience last century made tremendous progress towards understanding individual neurons as units of computation, perception and motor control. However, as described in the introductory chapter in the context of primary motor cortex, theories of nervous system function are necessarily constrained by the availability of experimental

data to test them. Advances in electrophysiological recording, optical imaging and stimulation techniques now allow for the simultaneous recording and stimulation of large populations of neurons. This permits the consideration of theories of nervous system function on an entirely different scale and, perhaps, of a completely different nature, to those previously considered. It is an open problem, in cases where comprehensive measures of neural activity are available, how best to utilize this information to learn principles of nervous system function. Projects like the Hydra imaging work presented here are important first steps in this direction.

Returning to the computational metaphor described in Chapter 1. While new experimental data and methods may soon provide unprecedented detail about network connectivity, neural dynamics, learning mechanisms, gene expression profiles, etc, determining how these details fit into neural computational frameworks that tell us how the brain solves particular problems will remain a challenge. Indeed such data amount to a more detailed physical picture of processes occurring within the brain. This may allow more accurate models which facilitate greater predictive ability. Yet just as machine learning methods heralding dramatic advances in prediction do not guarantee that advances in scientific insight follow, so too does a more detailed physical picture of the brain not guarantee advances in understanding how such processes relate to neural computation. The cognitive and computing revolutions share an intertwined history [14], with computational neuroscience in a sense being a natural expression of this intersection, but how this relation will adapt in the age of large-scale, data-driven neuroscience remains to be seen. Thus amidst the changes it is pertinent to keep in mind the question: what might cognition be, if not computation?

A

Proof of Results

A.1 Radially symmetric stimuli and auto-correlation

Let x be a stationary process then let $\mathbf{s}$ be a length n sequence of that process ($\mathbf{s} = (x_1, \ldots, x_n)$), having distribution $f_{\mathbf{S}}(\mathbf{s})$. Then the auto-correlation matrix Γ is a scalar multiple of the identify matrix ($\Gamma_{ij} = \sigma_x^2 \delta_{ij}$) if $f_{\mathbf{S}}(\mathbf{s})$ is radially symetric. The converse is not necessarily true, since the second order moments being spherical (a scalar multiple of the identity matrix) says nothing about higher order moments.

Proof: Assume that $f_{\mathbf{S}}(\mathbf{s})$ is radially symmetric: $f_{\mathbf{S}}(\mathbf{s}) = f_r(|\mathbf{s}|)$. Then for $i \neq j$ the function $g_{ij}(s_1, \ldots, s_n) = s_i s_j f_{\mathbf{S}}(\mathbf{s})$ is an odd function in s_j (or s_i) since

$$\begin{aligned} g_{ij}(s_1, \ldots, -s_j, \ldots s_n) &= - s_i s_j f_{\mathbf{S}}(s_1, \ldots, -s_j, \ldots s_n) \\ &= - s_i s_j f_r(|s_1, \ldots, -s_j, \ldots s_n|) = -s_i s_j f_r(|s_1, \ldots, s_j, \ldots s_n|) \\ &= - g_{ij}(s_1, \ldots, s_j, \ldots, s_n) \end{aligned}$$

meaning that

$$\begin{aligned} \langle s_i s_j \rangle &= \int_{\mathbb{R}^2} st f_{S_i S_j}(s,t)\, ds dt \\ &= 0. \end{aligned}$$

For $i = j$:

$$\langle s_i^2 \rangle = \sigma_x^2$$

by second-order stationarity of x. □

A.2 Equivalence between standard Kalman filter formulation and presented filter

Recall the Woodbury lemma:

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$$

for conformable matrices A, U, C and V. Thus from Equation (7.6) we obtain

$$W_{k+1} = W_{k+1}^- - W_{k+1}^- H^T (\sigma_\eta^2 + H W_{k+1}^- H^T)^{-1} H W_{k+1}^-,$$

which gives the update equation:

$$\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}^-_{k+1} + W^-_{k+1}\left(I - H^T(\sigma_\eta^2 + HW^-_{k+1}H^T)^{-1}HW^-_{k+1}\right)H^T\left(\sigma_\eta^2\right)^{-1}\left(\mathbf{y}_{k+1} - \hat{\mathbf{y}}^-_{k+1}\right).$$

Denoting the gain matrix by K gives:

$$\begin{aligned}
K &= W^-_{k+1}\left(I - H^T(\sigma_\eta^2 + HW^-_{k+1}H^T)^{-1}HW^-_{k+1}\right)H^T\left(\sigma_\eta^2\right)^{-1} \\
&= W^-_{k+1}H^T\left(\left(\sigma_\eta^2\right)^{-1} - (\sigma_\eta^2 + HW^-_{k+1}H^T)^{-1}HW^-_{k+1}H^T\left(\sigma_\eta^2\right)^{-1}\right) \\
&= W^-_{k+1}H^T\left(\left(\sigma_\eta^2\right)^{-1} - \left(\sigma_\eta^2\right)^{-1}\left(I + \left(\sigma_\eta^2\right)^{-1}HW^-_{k+1}H^T\right)^{-1}\left(HW^-_{k+1}H^T\right)\left(\sigma_\eta^2\right)^{-1}\right) \\
&= W^-_{k+1}H^T\left(\left(\sigma_\eta^2\right)^{-1} - \left(\sigma_\eta^2\right)^{-1}\left(\left(HW^-_{k+1}H^T\right)^{-1} + \left(\sigma_\eta^2\right)^{-1}\right)^{-1}\left(\sigma_\eta^2\right)^{-1}\right) \\
\text{(Woodbury lemma)} \qquad &= W^-_{k+1}H^T\left(HW^-_{k+1}H^T + \sigma_\eta^2\right)^{-1}.
\end{aligned}$$

Where the ultimate line is the standard definition of the gain (7.10). The covariance update:

$$\begin{aligned}
W_{k+1} =& W^-_{k+1} - W^-_{k+1}H^T(\sigma_\eta^2 + HW^-_{k+1}H^T)^{-1}HW^-_{k+1} \\
=& \left(I - KH\right)W^-_{k+1},
\end{aligned}$$

can similarly be expressed in standard form. □

Bibliography

[1] J. B. Ackman, T. J. Burbridge, and M. C. Crair, *Retinal waves coordinate patterned activity throughout the developing visual system*, Nature, 490 (2012), pp. 219–225.

[2] H. Adelsberger, O. Garaschuk, and A. Konnerth, *Cortical calcium waves in resting newborn mice.*, Nature neuroscience, 8 (2005), pp. 988–90.

[3] T. Aflalo, S. Kellis, C. Klaes, B. Lee, Y. Shi, K. Pejsa, K. Shanfield, S. Hayes-Jackson, M. Aisen, C. Heck, C. Liu, and R. A. Andersen, *Decoding motor imagery from the posterior parietal cortex of a tetraplegic human*, Science, 348 (2015), pp. 906–910.

[4] M. B. Ahrens, L. Paninski, and M. Sahani, *Inferring input nonlinearities in neural encoding models.*, Network (Bristol, England), 19 (2008), pp. 35–67.

[5] H. Akaike, *A new look at the statistical model identification*, IEEE Transactionson Automatic Control, 19 (1974), pp. 716–723.

[6] A. Akrouh and D. Kerschensteiner, *Intersecting circuits generate precisely patterned retinal waves.*, Neuron, 79 (2013), pp. 322–34.

[7] J. Aljadeff, B. Lansdell, A. Fairhall, and D. Kleinfeld, *Analysis of Neuronal Spike Trains, Deconstructed*, Neuron, 91 (2016), pp. 221–259.

[8] C. Allene, A. Cattani, J. B. Ackman, P. Bonifazi, L. Aniksztejn, Y. Ben-Ari, and R. Cossart, *Sequential generation of two distinct synapse-driven network patterns in developing neocortex.*, The Journal of neuroscience, 28 (2008), pp. 12851–63.

[9] C. Allene and R. Cossart, *Early NMDA receptor-driven waves of activity in the developing neocortex: physiological or pathological network oscillations?*, The Journal of physiology, 588 (2010), pp. 83–91.

[10] P.-O. Amblard and O. Michel, *The Relation between Granger Causality and Directed Information Theory: A Review*, Entropy, 15 (2012), pp. 113–143.

[11] E. W. Archer, U. Koster, J. W. Pillow, J. H. Macke, K. Urs, J. W. Pillow, and J. H. Macke, *Low-dimensional models of neural population activity in sensory cortical circuits*, in Advances in Neural Information Processing Systems, vol. 27, 2014, pp. 1–9.

[12] A. Arnold, Y. Liu, and N. Abe, *Temporal causal modeling with graphical granger methods*, Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, (2007).

[13] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, *A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking*, IEEE Transactions on Signal Processing, 50 (2002), pp. 174–188.

[14] P. M. Asaro, *Computers as Models of the Mind : On Simulations , Brains and the Design of Early Computers*, in The Search for a Theory of Cognition: Early Mechanisms and New Ideas, 2011, pp. 89–114.

[15] P. Bak, K. Chen, and C. Tang, *A forest-fire model and some thoughts on turbulence*, Physics Letters A, 147 (1990), pp. 297–300.

[16] P. Bak, C. Tang, and K. Wiesenfeld, *Self-organized criticality*, Physical review A, 38 (1988), pp. 364–374.

[17] C. Ballester, L. Garrido, V. Lazcano, and V. Caselles, *A TV-L1 optical flow method with occlusion detection*, Lecture Notes in Computer Science, 7476 (2012), pp. 31–40.

[18] A. Bansal, J. H. Singer, B. J. Hwang, W. Xu, A. Beaudet, and M. B. Feller, *Mice lacking specific nicotinic acetylcholine receptor subunits exhibit dramatically altered spontaneous activity patterns and reveal a limited role for retinal waves in forming ON and OFF circuits in the inner retina.*, The Journal of neuroscience, 20 (2000), pp. 7672–7681.

[19] R. Barbieri, L. M. Frank, M. C. Quirk, M. A. Wilson, and E. N. Brown, *Dynamic Analyses of Information Encoding in Neural*, Neural computation, 16 (2003), pp. 277–307.

[20] L. Barnett, A. B. Barrett, and A. K. Seth, *Granger Causality and Transfer Entropy Are Equivalent for Gaussian Variables*, Physical Review Letters, 103 (2009), pp. 238701–238701.

[21] L. Barnett and T. Bossomaier, *Transfer Entropy as a Log-likelihood Ratio*, arXiv preprint:1205.6339v3, (2012), pp. 1–10.

[22] L. Barnett and A. K. Seth, *The MVGC multivariate Granger causality toolbox: a new approach to Granger-causal inference.*, Journal of neuroscience methods, 223 (2014), pp. 50–68.

[23] L. Bashford, J. Wu, D. Sarma, K. Collins, J. Ojemann, and C. Mehring, *Natural movement with concurrent brain-computer interface control induces persistent dissociation of neural activity*, in BCI Society Meeting, CA, USA, 2016.

[24] T. Beeler, F. Hahn, D. Bradley, B. Bickel, P. Beardsley, C. Gotsman, R. W. Sumner, and M. Gross, *High-quality passive facial performance capture using anchor frames*, ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2011, 30 (2011), p. 1.

[25] J. M. Beggs, *The criticality hypothesis: how local cortical networks might optimize information processing.*, Philosophical transactions. Series A, Mathematical, physical, and engineering sciences, 366 (2008), pp. 329–43.

[26] J. M. Beggs and D. Plenz, *Neuronal avalanches in neocortical circuits.*, The Journal of neuroscience, 23 (2003), pp. 11167–11177.

[27] J. M. Beggs and N. Timme, *Being critical of criticality in the brain.*, Frontiers in physiology, 3 (2012), pp. 1–14.

[28] B. M. Bell and F. W. Cathey, *The Iterated Kalman Filter Update as a Gauss-Newton Method*, IEEE Transactions on Automatic Control, 38 (1993), pp. 294–297.

[29] M. Berman and T. Turner, *Approximating point process likelihoods with GLIM*, Journal of the Royal Statistical Society. Series C (Applied Statistics), 41 (1992), pp. 31–38.

[30] A. Blankenship and M. Feller, *Mechanisms underlying spontaneous patterned activity in developing neural circuits.*, Nature Reviews Neuroscience, 11 (2010), pp. 18–29.

[31] A. G. Blankenship, K. J. Ford, J. Johnson, R. P. Seal, R. H. Edwards, D. R. Copenhagen, and M. B. Feller, *Synaptic and extrasynaptic factors governing glutamatergic retinal waves.*, Neuron, 62 (2009), pp. 230–41.

[32] A. Bolstad, B. D. Van Veen, and R. Nowak, *Causal network inference via group sparse regularization*, IEEE Transactions on Signal Processing, 59 (2011), pp. 2628–2641.

[33] G. Borisyuk, R. Borisyuk, A. Kirillov, E. Kovalenko, and V. Kryukov, *A new statistical method for identifying interconnections between neuronal network elements*, Biological cybernetics, 306 (1985), pp. 301–306.

[34] E. R. Boykin, P. P. Khargonekar, P. R. Carney, W. O. Ogle, and S. S. Talathi, *Detecting effective connectivity in networks of coupled neuronal oscillators.*, Journal of computational neuroscience, 32 (2012), pp. 521–38.

[35] P. Brémaud and L. Massoulié, *Stability of nonlinear Hawkes processes*, The Annals of Probability, 24 (1996), pp. 1563–1588.

[36] T. Brox, N. Papenberg, and J. Weickert, *High Accuracy Optical Flow Estimation Based on a Theory for Warping*, in 8th European Conference on Computer Vision, vol. 4, 2004, pp. 25–36.

[37] T. Brox, B. Rosenhahn, and D. Cremers, *Contours, Optic Flow, and Prior Knowledge: Cues for Capturing 3D Human Motion in Videos*, Human Motion - Understanding, Modelling, Capture, and Animation, 36 (2008), pp. 265–293.

[38] M. A. Buice and J. D. Cowan, *Statistical mechanics of the neocortex.*, Progress in biophysics and molecular biology, 99 (2009), pp. 53–86.

[39] P. Burgi and N. M. Grzywacz, *Model based on extracellular potassium for spontaneous synchronous activity in developing retinas*, Neural computation, 6 (1994), pp. 983–1004.

[40] J. J. Bussgang, *Crosscorrelation functions of amplitude distorted Gaussian signals*, MIT Res Lab Elec Tech Rep, 216 (1952).

[41] D. A. Butts, M. B. Feller, C. J. Shatz, and D. S. Rokhsar, *Retinal waves are governed by collective network properties.*, The Journal of neuroscience, 19 (1999), pp. 3580–93.

[42] J. M. Carmena, M. a. Lebedev, R. E. Crist, J. E. O'Doherty, D. M. Santucci, D. F. Dimitrov, P. G. Patil, C. S. Henriquez, and M. a. L. Nicolelis, *Learning to Control a Brain-Machine Interface for Reaching and Grasping by Primates*, PLoS Biology, 1 (2003), p. e2.

[43] L. Carstensen, A. Sandelin, O. Winther, and N. R. Hansen, *Multivariate Hawkes process models of the occurrence of regulatory elements.*, Bmc Bioinformatics, 11 (2010), p. 456.

[44] M. L. Cascia, S. Sclaroff, and V. Athitsos, *Fast, Reliable Head Tracking Under Varying Illumination: An Approach Based on Registration of Texture-Mapped 3D Models*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22 (2000), pp. 322–336.

[45] A. A. Cattani, V. D. Bonfardin, A. Represa, Y. Ben-Ari, and L. Aniksztejn, *Generation of slow network oscillations in the developing rat hippocampus after blockade of glutamate uptake.*, Journal of neurophysiology, 98 (2007), pp. 2324–36.

[46] A. Chambolle, *An algorithm for total variation minimizations and applications*, Journal of Mathematical Imaging and Vision, 20 (2004), pp. 89–97.

[47] A. Chambolle, D. Cremers, and T. Pock, *A Convex Approach to Minimal Partitions*, SIAM Journal on Imaging Sciences, 5 (2012), pp. 1113–1158.

[48] A. Chambolle and T. Pock, *A first-order primal-dual algorithm for convex problems with applications to imaging*, Journal of Mathematical Imaging and Vision, 40 (2011), pp. 120–145.

[49] A. R. Chandrasekaran, R. D. Shah, and M. C. Crair, *Developmental homeostasis of mouse retinocollicular synapses.*, The Journal of neuroscience, 27 (2007), pp. 1746–55.

[50] S. M. Chase, R. E. Kass, and A. B. Schwartz, *Behavioral and neural correlates of visuomotor adaptation observed through a brain-computer interface in primary motor cortex.*, Journal of neurophysiology, 108 (2012), pp. 624–44.

[51] Z. Chen, D. F. Putrino, D. Ba, S. Ghosh, and R. Barbieri, *A regularized point process generalized linear model for assessing the connectivity in the cat motor cortex*, Annual International Conference of the IEEE, (2009), pp. 5006–5009.

[52] P. D. Cheney, D. M. Griffin, and G. M. V. Acker, *Neural Hijacking: Action of High Frequency Electrical Stimulation on Cortical Circuits*, Neuroscientist, 19 (2013), pp. 434–441.

[53] D. Chicharro and A. Ledberg, *When two become one: the limits of causality analysis of brain dynamics.*, PloS one, 7 (2012), p. e32466.

[54] E. Chichilnisky, *A simple white noise analysis of neuronal light responses*, Network: Computation in Neural Systems, 12 (2001), pp. 199–213.

[55] E. Chornoboy, L. Schramm, and A. Karr, *Maximum likelihood identification of neural point process systems*, Biological cybernetics, 275 (1988), pp. 265–275.

[56] S. H. Chung, L. Sun, and C. V. Gabel, *In vivo neuronal calcium imaging in C. elegans.*, 2013.

[57] M. M. Churchland, J. P. Cunningham, M. T. Kaufman, J. D. Foster, P. Nuyujukian, S. I. Ryu, and K. V. Shenoy, *Neural population dynamics during reaching.*, Nature, 487 (2012), pp. 51–6.

[58] M. M. Churchland, J. P. Cunningham, M. T. Kaufman, S. I. Ryu, and K. V. Shenoy, *Cortical preparatory activity: representation of movement or first cog in a dynamical machine?*, Neuron, 68 (2010), pp. 387–400.

[59] P. S. Churchland, C. Koch, and T. J. Sejnowski, *What is computational neuroscience?*, in Computational Neuroscience, 1988, p. 1.

[60] S. Clar, B. Drossel, and F. Schwabl, *Forest fires and other examples of self-organized criticality*, Journal of Physics: Condensed Matter, 8 (1996), pp. 1–37.

[61] J. L. Collinger, B. Wodlinger, J. E. Downey, W. Wang, E. C. Tyler-Kabara, D. J. Weber, A. J. C. McMorland, M. Velliste, M. L. Boninger, and A. B. Schwartz, *High-performance neuroprosthetic control by an individual with tetraplegia*, The Lancet, 381 (2013), pp. 557–564.

[62] D. Daley and D. Vere-Jones, *An Introduction to the Theoery of Point Processes: Volume I: Elementary Theory and Methods, Second Edition*, vol. I, Springer, 2003.

[63] J. G. Daugman, *Brain metaphor and brain theory*, in Philosophy and the Neurosciences: A Reader, 1993.

[64] D. DeCarlo and D. Metaxas, *Optical Flow Constraints on Deformable Models with Applications to Face Tracking*, International Journal of Computer Vision, 38 (2000), pp. 99–127.

[65] F. Dellaert, S. Thrun, and C. Thorpe, *Jacobian images of super-resolved texture maps for model-based motion estimation and tracking*, in Applications of Computer Vision, 1998. WACV '98. Proceedings., Fourth IEEE Workshop on, no. 1, 1998, pp. 2–7.

[66] M. Dhamala, G. Rangarajan, and M. Ding, *Estimating Granger Causality from Fourier and Wavelet Transforms of Time Series Data*, Physical Review Letters, 100 (2008), p. 018701.

[67] V. Didelez, *Graphical models for marked point processes based on local independence*, Journal of the Royal Statistical Society. Series B. Statistical Methodology, 70 (2005), pp. 245–264.

[68] E. J. Doedel, *Auto 2007p: Continuation and bifurcation software for ordinary differential equations (with homcont).*, 2007.

[69] D. Donoho and M. Elad, *Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization*, Proceedings of the National Academy of Sciences, 100 (2003), pp. 2197–2202.

[70] A. Doucet and A. Johansen, *A tutorial on particle filtering and smoothing: fifteen years later*, Handbook of Nonlinear Filtering, (2011), pp. 656–704.

[71] N. R. Draper and H. Smith, *Applied regression analysis*, John Wiley & Sons, 2014.

[72] B. Drossel and F. Schwabl, *Self-organized critical forest fire model*, Physical review letters, 69 (1992), pp. 1629–1632.

[73] C. Dupre and R. Yuste, *Calcium imaging reveals multiple conduction systems in hydra*, Annual Meeting of the Society for Neuroscience, 94 (2015).

[74] U. T. Eden, L. M. Frank, R. Barbieri, V. Solo, and E. N. Brown, *Dynamic analysis of neural encoding by point process adaptive filtering.*, Neural computation, 16 (2004), pp. 971–98.

[75] G. F. Elsayed, A. H. Lara, M. T. Kaufman, M. M. Churchland, and J. P. Cunningham, *Reorganization between preparatory and movement population responses in motor cortex*, Nature Communications, (2016), p. 13239.

[76] B. Engelhard, N. Ozeri, Z. Israel, H. Bergman, and E. Vaadia, *Inducing γ oscillations and precise spike synchrony by operant conditioning via brain-machine interface.*, Neuron, 77 (2013), pp. 361–75.

[77] G. B. Ermentrout and D. H. Terman, *Mathematical foundations of neuroscience*, Springer, 2010.

[78] I. Essa and A. Pentland, *A vision system for observing and extracting facial action parameters*, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, no. 247, 1994, pp. 76–83.

[79] I. a. Essa and A. P. Pentland, *Facial expression recognition using a dynamic model and motion energy*, in Proceedings of IEEE International Conference on Computer Vision, no. 307, 1995, pp. 360–367.

[80] E. V. Evarts, *Relation of pyramidal tract activity to force exerted during voluntary movement.*, Journal of Neurophysiology, 31 (1968), pp. 14–27.

[81] A. Fairhall, *The receptive field is dead. Long live the receptive field?*, Current opinion in neurobiology, 25 (2014), pp. ix–xii.

[82] M. B. Feller, D. a. Butts, H. L. Aaron, D. S. Rokhsar, and C. J. Shatz, *Dynamic processes shape spatiotemporal properties of retinal waves.*, Neuron, 19 (1997), pp. 293–306.

[83] E. Fetz and D. Finocchio, *Correlations between activity of motor cortex cells and arm muscles during operantly conditioned response patterns*, Experimental Brain Research, 23 (1975), pp. 217–240.

[84] E. E. Fetz, *Operant conditioning of cortical unit activity.*, Science, 163 (1969), pp. 955–8.

[85] E. E. Fetz, *Volitional control of neural activity: implications for brain-computer interfaces*, The Journal of Physiology, 579 (2007), pp. 571–579.

[86] E. E. Fetz and M. a. Baker, *Operantly conditioned patterns on precentral unit activity and correlated responses in adjacent cells and contralateral muscles.*, Journal of neurophysiology, 36 (1973), pp. 179–204.

[87] A. Firl, G. S. Sack, Z. L. Newman, H. Tani, and M. B. Feller, *Extrasynaptic glutamate and inhibitory neurotransmission modulate ganglion cell participation during glutamatergic retinal waves.*, Journal of neurophysiology, 109 (2013), pp. 1969–78.

[88] J. Florens and D. Fougere, *Noncausality in continuous time*, Econometrica: Journal of the Econometric Society, 64 (1996), pp. 1195–1212.

[89] J. A. Fodor, *The Language of Thought*, Harvard University Press., Cambridge, Massachusetts, 1975.

[90] K. Ford, D. Arroyo, J. Kay, E. E. Lloyd, R. M. Bryan, J. R. Sanes, and M. B. Feller, *A role for TREK1 in generating the slow afterhyperpolarization in developing starburst amacrine cells*, Journal of neurophysiology, 109 (2013), pp. 2250–2259.

[91] K. Ford and M. Feller, *Formation of Early Retinal Circuits in the Inner Plexiform Layer*, in Webvision, 2012.

[92] K. J. Ford, A. L. Félix, and M. B. Feller, *Cellular mechanisms underlying spatiotemporal features of cholinergic retinal waves.*, The Journal of neuroscience, 32 (2012), pp. 850–63.

[93] K. J. Friston, A. M. Bastos, A. Oswal, B. van Wijk, C. Richter, V. Litvak, and B. V. Wijk, *Granger causality revisited.*, NeuroImage, 101 (2014), pp. 796–808.

[94] K. Ganguly and J. M. Carmena, *Emergence of a Stable Cortical Map for Neuroprosthetic Control*, PLoS Biology, 7 (2009), p. e1000153.

[95] K. Ganguly, D. F. Dimitrov, J. D. Wallis, and J. M. Carmena, *Reversible large-scale modification of cortical networks during neuroprosthetic control.*, Nature neuroscience, 14 (2011), pp. 662–7.

[96] R. Garg, A. Roussos, and L. Agapito, *A variational approach to video registration with subspace constraints*, International Journal of Computer Vision, 104 (2013), pp. 286–314.

[97] A. Gégout-Petit, D. Commenges, and A. G.-p. D. Commenges, *A general definition of influence between stochastic processes.*, Lifetime data analysis, 16 (2010), pp. 33–44.

[98] A. Gelman, J. Hwang, and A. Vehtari, *Understanding predictive information criteria for Bayesian models*, Statistics and Computing, 24 (2014), pp. 997–1016.

[99] A. P. Georgopoulos, J. F. Kalaska, R. Caminiti, and J. T. Massey, *On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex*, J.Neurosci., 2 (1982), pp. 1527–1537.

[100] F. Gerhard, T. Kispersky, G. J. Gutierrez, E. Marder, M. Kramer, and U. Eden, *Successful reconstruction of a physiological circuit with known connectivity from spiking activity alone.*, PLoS computational biology, 9 (2013), p. e1003138.

[101] J. Geweke, *Measurement of Linear Dependence and Feedback between Multiple Time Series*, Journal of the American Statistical Association, 77 (1982), pp. 304–313.

[102] J. F. Geweke, *Measures of Conditional Linear Dependence and Feedback between Time Series*, Journal of the American Statistical Association, 79 (1984), pp. 907–915.

[103] V. Gilja, C. Pandarinath, C. H. Blabe, P. Nuyujukian, J. D. Simeral, A. a. Sarma, B. L. Sorice, J. a. Perge, B. Jarosiewicz, L. R. Hochberg, K. V. Shenoy, and J. M. Henderson, *Clinical translation of a high-performance neural prosthesis.*, Nature medicine, 21 (2015), pp. 6–8.

[104] J. Gjorgjieva and S. Eglen, *Modeling developmental patterns of spontaneous activity.*, Current opinion in neurobiology, 21 (2011), pp. 679–84.

[105] K. B. Godfrey and S. J. Eglen, *Theoretical models of spontaneous activity generation and propagation in the developing retina.*, Molecular BioSystems, 5 (2009), pp. 1527–1535.

[106] K. B. Godfrey, S. J. Eglen, and N. V. Swindale, *A multi-component model of the developing retinocollicular pathway incorporating axonal and synaptic growth.*, PLoS computational biology, 5 (2009), p. e1000600.

[107] K. B. Godfrey and N. V. Swindale, *Retinal wave behavior through activity-dependent refractory periods.*, PLoS computational biology, 3 (2007), p. e245.

[108] M. D. Golub, S. M. Chase, A. P. Batista, and B. M. Yu, *Brain-computer interfaces for dissecting cognitive processes underlying sensorimotor control*, Current Opinion in Neurobiology, 37 (2016), pp. 53–58.

[109] M. D. Golub, B. M. Yu, and S. M. Chase, *Internal models engaged by brain-computer interface control.*, IEEE Engineering in Medicine and Biology Society, 2012 (2012), pp. 1327–1330.

[110] C. W. J. Granger, *Investigating causal relations by econometric models and cross-spectral methods*, Econometrica: Journal of the Econometric Society, (1969), pp. 424–438.

[111] P. Grassberger, *Critical behaviour of the Drossel-Schwabl forest fire model*, New Journal of Physics, (2002), pp. 1–8.

[112] M. Graziano, *The organization of behavioral repertoire in motor cortex.*, Annual review of neuroscience, 29 (2006), pp. 105–34.

[113] M. S. A. Graziano, C. S. R. Taylor, and T. Moore, *Complex movements evoked by microstimulation of precentral cortex*, Neuron, 34 (2002), pp. 841–851.

[114] Y. Guo, J. Gao, and X. Hong, *Constrained grouped sparsity*, Lecture Notes in Computer Science, 7691 LNAI (2012), pp. 433–444.

[115] N. Hansen, *Penalized maximum likelihood estimation for generalized linear point processes*, arXiv preprint arXiv:1003.0848, (2010).

[116] N. R. Hansen, P. Reynaud-Bouret, and V. Rivoirard, *Lasso and probabilistic inequalities for multivariate point processes*, Bernoulli, 21 (2015), pp. 83–143.

[117] M. Harris, G. E. Blelloch, B. M. Maggs, N. K. Govindaraju, B. Lloyd, W. Wang, M. Lin, D. Manocha, P. K. Smolarkiewicz, and L. G. Margolin, *Optimizing parallel reduction in CUDA*, Proc. of ACM SIGMOD, 21, 13 (2007), pp. 104–110.

[118] S. Haufe, G. Nolte, K. Mueller, and N. Krämer, *Sparse causal discovery in multivariate time series*, arXiv preprint arXiv:0901.2234, (2009), pp. 1–10.

[119] J. Havlík and O. Straka, *Performance evaluation of iterated extended Kalman filter with variable step-length*, Journal of Physics: Conference Series, 659 (2015), p. 012022.

[120] A. Hawkes, *Spectra of some self-exciting and mutually exciting point processes*, Biometrika, 58 (1971), pp. 83–90.

[121] M. H. Hennig, C. Adams, D. Willshaw, and E. Sernagor, *Early-stage waves in the retinal network emerge close to a critical state transition between local and global functional connectivity.*, The Journal of neuroscience, 29 (2009), pp. 1077–86.

[122] S. Hergarten and R. Krenn, *A semi-phenomenological approach to explain the event-size distribution of the Drossel-Schwabl forest-fire model*, Nonlinear Processes in Geophysics, 18 (2011), pp. 381–388.

[123] K. Hey, T., Tansley, S., Tolle, *The Fourth Paradigm*, Microsoft Research, 2009.

[124] T. Hirabayashi, D. Takeuchi, K. Tamura, and Y. Miyashita, *Functional microcircuit recruited during retrieval of object association memory in monkey perirhinal cortex.*, Neuron, 77 (2013), pp. 192–203.

[125] L. R. Hochberg, D. Bacher, B. Jarosiewicz, N. Y. Masse, J. D. Simeral, J. Vogel, S. Haddadin, J. Liu, S. S. Cash, P. van der Smagt, and J. P. Donoghue, *Reach and grasp by people with tetraplegia using a neurally controlled robotic arm*, Nature, 485 (2012), pp. 372–375.

[126] A. Hoerl and R. Kennard, *Ridge regression: Biased estimation for nonorthogonal problems*, Technometrics, 12 (1970), pp. 55–67.

[127] B. K. B. Horn and B. G. Schunck, *Determining Optical Flow*, Artificial Intelligence Memo 572, 319 (1980), pp. 185–203.

[128] M. Y. Huang, S. C. Wei, B. Huang, and Y. L. Chang, *Accelerating the Kalman filter on a GPU*, in Proceedings of the International Conference on Parallel and Distributed Systems - ICPADS, 2011, pp. 1016–1020.

[129] D. Hubel and T. Wiesel, *Receptive fields of single neurones in the cat's striate cortex*, Journal of Physiology, 148 (1959), pp. 574–591.

[130] A. D. Huberman, M. Feller, and B. Chapman, *Mechanisms underlying development of visual maps and receptive fields*, Annual review of neuroscience, (2008), pp. 479–509.

[131] I. Hunter and M. Korenberg, *The identification of nonlinear biological systems: Wiener and Hammerstein cascade models*, Biological cybernetics, 144 (1986), pp. 135–144.

[132] E. J. Hwang, P. M. Bailey, and R. a. Andersen, *Volitional control of neural activity relies on the natural motor repertoire*, Current Biology, 23 (2013), pp. 353–361.

[133] S. Ito, M. E. Hansen, R. Heiland, A. Lumsdaine, A. M. Litke, and J. M. Beggs, *Extending transfer entropy improves identification of effective connectivity in a spiking cortical network model.*, PloS one, 6 (2011), p. e27431.

[134] B. Jarosiewicz, S. M. Chase, G. W. Fraser, M. Velliste, R. E. Kass, and A. B. Schwartz, *Functional network reorganization during learning in a brain-computer interface paradigm.*, Proceedings of the National Academy of Sciences of the United States of America, 105 (2008), pp. 19486–19491.

[135] T. Jebara and A. Pentland, *Parametrized structure from motion for 3D adaptive feedback tracking of faces*, Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (1997), pp. 144–150.

[136] M. Kass, A. Witkin, and D. Terzopoulos, *Snakes: Active contour models*, International Journal of Computer Vision, 1 (1988), pp. 321–331.

[137] R. Kass and V. Ventura, *A spike-train probability model*, Neural computation, 1720 (2001), pp. 1713–1720.

[138] M. T. Kaufman, M. M. Churchland, S. I. Ryu, and K. V. Shenoy, *Cortical activity in the null space: permitting preparation without movement*, Nature Neuroscience, 17 (2014), pp. 440–448.

[139] J. Keat, P. Reinagel, R. C. Reid, and M. Meister, *Predicting Every Spike: A Model for the Responses of Visual Neurons*, Neuron, 30 (2001), pp. 803–817.

[140] J. Keener and J. Sneyd, *Mathematical Physiology*, Springer, 2001.

[141] R. Kennedy and C. J. Taylor, *Optical Flow with Geometric Occlusion Estimation and Fusion of Multiple Frames*, International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition, (2015), pp. 364–377.

[142] D. Kerschensteiner, *Spontaneous Network Activity and Synaptic Development.*, The Neuroscientist, XX (2013), pp. 1–19.

[143] S. Kim, D. Putrino, S. Ghosh, and E. N. Brown, *A Granger causality measure for point process models of ensemble neural spiking activity.*, PLoS computational biology, 7 (2011), p. e1001110.

[144] T. Kispersky, G. Gutierrez, and E. Marder, *Functional connectivity in a rhythmic inhibitory circuit using Granger causality*, Neural Systems and Circuits, 1 (2011), p. 9.

[145] C. Koch, *Biophysics of computation: information processing in single neurons*, Oxford university press, 1998.

[146] C. Kondermann, R. Mester, and C. Garbe, *A statistical confidence measure for optical flows*, in Lecture Notes in Computer Science, vol. 5304 LNCS, 2008, pp. 290–301.

[147] L. Koralov and Y. G. Sinai, *Theory of probability and random processes*, Springer Science & Business Media, 2007.

[148] M. Krumin, I. Reutsky, and S. Shoham, *Correlation-based analysis and generation of multiple spike trains using hawkes models with an exogenous input.*, Frontiers in computational neuroscience, 4 (2010), p. 147.

[149] M. Krumin and S. Shoham, *Multivariate autoregressive modeling and granger causality analysis of multiple spike trains.*, Computational intelligence and neuroscience, 2010 (2010), p. 752428.

[150] J. E. Kulkarni and L. Paninski, *Common-input models for multiple neural spike-train data*, Network: Computation in Neural Systems, 18 (2007), pp. 375–407.

[151] J. Kybic and C. Nieuwenhuis, *Bootstrap optical flow confidence and uncertainty measure*, Computer Vision and Image Understanding, 115 (2011), pp. 1449–1462.

[152] G. Lajoie, N. Krouchev, J. F. Kalaska, A. Fairhall, and E. E. Fetz, *Correlation-based model of artificially induced plasticity in motor cortex by a bilateral Brain-Machine Interface*, PLoS Computational Biology, 13 (2017), pp. 1–35.

[153] B. Lansdell, K. Ford, and J. N. Kutz, *A reaction-diffusion model of cholinergic retinal waves*, PLoS computational biology, 10 (2014), p. e1003953.

[154] B. Lansdell, I. Milovanovic, C. Mellema, E. E. Fetz, A. L. Fairhall, and C. T. Moritz, *Reconfiguring motor circuits for a joint manual and BCI task*, arXiv, 1702.07368 (2017).

[155] A. J. Law, G. Rivlis, and M. H. Schieber, *Rapid acquisition of novel interface control by small ensembles of arbitrarily selected primary motor cortex neurons.*, Journal of neurophysiology, 112 (2014), pp. 1528–1548.

[156] M. A. Lebedev, J. M. Carmena, J. E. O'Doherty, M. Zacksenhouse, C. S. Henriquez, J. C. Principe, and M. A. L. Nicolelis, *Cortical Ensemble Adaptation to Represent Velocity of an Artificial Actuator Controlled by a Brain-Machine Interface*, Journal of Neuroscience, 25 (2005), pp. 4681–4693.

[157] H. Li, S. Member, P. Roivainen, and R. Forchheimer, *3-D Motion Estimation in Model-Based Facial Image Coding*, IEEE transactions on pattern analysis and machine intelligence, 15 (1993), pp. 545–555.

[158] K. Li and N. Duan, *Regression analysis under link violation*, The Annals of Statistics, (1989).

[159] W. Li, D. Cosker, and M. Brown, *Drift robust non-rigid optical flow enhancement for long sequences*, Journal of Intelligent and Fuzzy Systems, 31 (2016), pp. 2583–2595.

[160] Z. Lin, D. Moore, and S. Russell, *GPU-Based Parallel Kalman Filter*, (2015), pp. 1–6.

[161] S. Linderman and R. Adams, *Discovering latent network structure in point process data*, arXiv preprint arXiv:1402.0914, (2014).

[162] T. Liu, A. W. Bargteil, J. F. O'Brien, and L. Kavan, *Fast simulation of mass-spring systems*, ACM Transactions on Graphics, 32 (2013), pp. 1–7.

[163] Y. Liu, J. Moser, and S. Aviyente, *Directional Neural Networks Inferred From Multichannel Multisubject EEG Data*, IEEE Transactions on bio-medical engineering, 61 (2014), pp. 1919–1930.

[164] D. Lloyd, *Simple Minds*, The MIT Press, 1989.

[165] V. Loreto, L. Pietronero, a. Vespignani, and S. Zapperi, *Renormalization group approach to the critical behavior of the forest-fire model.*, Physical review letters, 75 (1995), pp. 465–468.

[166] A. C. Lozano, N. Abe, Y. Liu, and S. Rosset, *Grouped graphical Granger modeling for gene expression regulatory networks discovery.*, Bioinformatics, 25 (2009), pp. i110–8.

[167] M. Lungarella, K. Ishiguro, Y. Kuniyoshi, and N. Otsu, *Methods for quantifying the causal structure of bivariate time series*, International journal of Bifurcation and Chaos, (2007), pp. 1–44.

[168] B. Lusch, P. D. Maia, and J. N. Kutz, *Inferring connectivity in networked dynamical systems: Challenges using Granger causality*, Physical Review E, 94 (2016), p. 032220.

[169] H. Lütkepohl, *New introduction to multiple time series analysis*, Springer Science & Business Media, 2007.

[170] C. Machens, M. Wehr, and A. Zador, *Spectro-temporal receptive fields of subthreshold responses in auditory cortex*, Advances in Neural Information Processing Systems, 15 (2003), pp. 149–156.

[171] J. H. Macke, L. Buesing, J. P. Cunningham, B. M. Yu, K. V. Shenoy, and M. Sahani, *Empirical models of spiking in neuronal populations*, Advances in Neural Information Processing Systems, 24 (2011), pp. 1–9.

[172] M. Maravall and M. E. Diamond, *Algorithms of whisker-mediated touch perception*, Current Opinion in Neurobiology, 25 (2014), pp. 176–186.

[173] D. Marinazzo, M. Pellicoro, and S. Stramaglia, *Kernel method for nonlinear Granger causality*, Physical Review Letters, 100 (2008), pp. 1–5.

[174] T. Marks, J. Hershey, J. Roddey, J. Movellan, and J. H. Tim K. Marks J. Cooper Roddey and Javier R. Movellan, *3D Tracking of Morphable Objects Using Conditionally Gaussian Nonlinear Filters*, 2004 Conference on Computer Vision and Pattern Recognition Workshop, 00 (2004), pp. 190–190.

[175] T. K. Marks, J. R. Hershey, and J. R. Movellan, *Tracking motion, deformation, and texture using conditionally gaussian processes*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 32 (2010), pp. 348–363.

[176] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*, Henry Holt and Co Inc., New York, 1982.

[177] K. Martin, *A brief history of the "feature detector"*., Cerebral cortex, 4 (1994), pp. 1–7.

[178] R. K. Mehra, *On the Identification of Variances and Adaptive Kalman Filtering*, IEEE Transactions on Automatic Control, AC-15 (1970), pp. 175–184.

[179] D. Metaxas and D. Terzopoulos, *Shape and nonrigid motion estimation through physics-based synthesis*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 15 (1993), pp. 580–591.

[180] J. A. Michaels, B. Dann, and H. Scherberger, *Neural Population Dynamics during Reaching Are Better Explained by a Dynamical System than Representational Tuning*, PLoS Computational Biology, 12 (2016), pp. 1–22.

[181] I. Milovanovic, R. Robinson, E. E. Fetz, and C. T. Moritz, *Simultaneous and independent control of a brain-computer interface and contralateral limb movement*, Brain-Computer Interfaces, 2621 (2015), pp. 1–12.

[182] W. Moody and M. Bosma, *Ion channel development, spontaneous activity, and activity-dependent development in nerve and muscle cells*, Physiological reviews, 85 (2005), pp. 883–941.

[183] C. T. Moritz and E. E. Fetz, *Volitional control of single cortical neurons in a brain-machine interface.*, Journal of neural engineering, 8 (2011), p. 025017.

[184] C. T. Moritz, S. I. Perlmutter, and E. E. Fetz, *Direct control of paralysed muscles by cortical neurons*, 456 (2008), pp. 639–643.

[185] C. Morris and H. Lecar, *Voltage oscillations in the barnacle giant muscle fiber.*, Biophysical journal, 35 (1981), pp. 193–213.

[186] J. R. Movellan, J. Hershey, T. K. Marks, C. Roddey, M. Tr, C. San, and D. La, *G-Flow: A Generative Model for Fast Tracking Using 3D Deformable Models*, Computer Vision and Pattern Recognition, (2004).

[187] M. Müller, J. Dorsey, and L. McMillan, *Stable Real-time Deformations*, Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation, (2002), pp. 49 – 54.

[188] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff, *Position based dynamics*, Journal of Visual Communication and Image Representation, 18 (2007), pp. 109–118.

[189] A. Myronenko, A. Myronenko, X. Song, X. Song, M. Á. Carreira-Perpiñán, and M. Á. Carreira-Perpiñán, *Non-rigid point set registration: Coherent Point Drift*, Advances in Neural Information Processing Systems 19, (2007), pp. 1009–1016.

[190] A. G. Nedungadi, G. Rangarajan, N. Jain, and M. Ding, *Analyzing multiple spike trains with nonparametric Granger causality.*, Journal of computational neuroscience, 27 (2009), pp. 55–64.

[191] J. P. Nguyen, A. N. Linder, G. S. Plummer, J. W. Shaevitz, and A. M. Leifer, *Automatically tracking neurons in a moving and deforming brain*, PLoS computational biology, 13 (2017).

[192] J. P. Nguyen, F. B. Shipley, A. N. Linder, G. S. Plummer, M. Liu, S. U. Setru, J. W. Shaevitz, and A. M. Leifer, *Whole-brain calcium imaging with cellular resolution in freely behaving Caenorhabditis elegans.*, Proceedings of the National Academy of Sciences of the United States of America, 3 (2015), p. 33.

[193] S. Nigam, M. Shimono, S. Ito, F.-C. Yeh, N. Timme, M. Myroshnychenko, C. C. Lapish, Z. Tosi, P. Hottowy, W. C. Smith, S. C. Masmanidis, A. M. Litke, O. Sporns, and J. M. Beggs, *Rich-Club Organization in Effective Connectivity among Cortical Neurons*, Journal of Neuroscience, 36 (2016), pp. 670–684.

[194] M. Nikolova, S. Esedoglu, T. F. Chan, S. Esedoglu, M. Nikolova, S. Esedo, Â. Glu, and M. Nikolova, *Algorithms for Finding Global Minimizers of Image Segmentation and Denoising Models*, SIAM Journal on Applied Mathematics, 66 (2006), pp. 1632–1648.

[195] P. Ochs, J. Malik, and T. Brox, *Segmentation of moving objects by long term video analysis*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 36 (2014), pp. 1187–1200.

[196] B. A. Olshausen and D. J. Field, *Sparse coding of sensory inputs*, Current Opinion in Neurobiology, 14 (2004), pp. 481–487.

[197] A. Orsborn, H. Moorman, S. Overduin, M. Shanechi, D. Dimitrov, and J. Carmena, *Closed-Loop Decoder Adaptation Shapes Neural Plasticity for Skillful Neuroprosthetic Control*, Neuron, 82 (2014), pp. 1380–1393.

[198] L. Paninski, *Convergence properties of three spike-triggered analysis techniques.*, Network: Computation in Neural Systems, 14 (2003), pp. 437–64.

[199] ——, *Maximum likelihood estimation of cascade point-process neural encoding models*, Network: Computation in Neural Systems, 15 (2004), pp. 243–262.

[200] F. Papangelou, *Integrability of expected increments of point processes and a related random change of scale*, Transactions of the American Mathematical Society, 165 (1972), pp. 483–506.

[201] I. Park and J. Pillow, *Bayesian spike-triggered covariance analysis*, Advances in Neural Information Processing Systems, (2011), pp. 1–9.

[202] S. Pasha and V. Solo, *Topology identification of dynamic point process networks*, in IEEE International Conference on Acoustic, Speech and Signal Processing, 2014, pp. 375–378.

[203] A. Pentland and B. Horowitz, *Recovery of nonrigid motion and structure*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 13 (1991), pp. 325–330.

[204] P.-o. Persson and G. Strang, *A Simple Mesh Generator in MATLAB*, SIAM Review, 46 (2004), pp. 329–345.

[205] T. Petermann, T. C. Thiagarajan, M. a. Lebedev, M. a. L. Nicolelis, D. R. Chialvo, and D. Plenz, *Spontaneous cortical activity in awake monkeys composed of neuronal avalanches.*, Proceedings of the National Academy of Sciences of the United States of America, 106 (2009), pp. 15921–6.

[206] D. Pfau, E. Pnevmatikakis, and L. Paninski, *Robust learning of low-dimensional dynamics from large neural ensembles*, Advances in neural information processing systems, (2013), pp. 1–15.

[207] G. Piccinini and O. Shagrir, *Foundations of computational neuroscience*, Current opinion in neurobiology, 25 (2014), pp. 25–30.

[208] J. W. J. Pillow, J. Shlens, L. Paninski, A. Sher, A. M. Litke, E. J. Chichilnisky, and E. P. Simoncelli, *Spatio-temporal correlations and visual signaling in a complete neuronal population*, Nature, 454 (2008), pp. 995–9.

[209] V. Priesemann, M. H. J. Munk, and M. Wibral, *Subsampling effects in neuronal avalanche distributions recorded in vivo.*, BMC neuroscience, 10 (2009), p. 40.

[210] G. Pruessner, *Self-organised criticality: theory, models and characterisation*, Cambridge University Press, 2012.

[211] B. G. Quinn and E. Hannan, *The determination of the order of an autoregression*, Journal of the Royal Statistical Society. Series B, 41 (1979), pp. 190–195.

[212] C. Quinn, T. Coleman, N. Kiyavash, and N. G. Hatsopoulos, *Estimating the directed information to infer causal relationships in ensemble neural spike train recordings*, Journal of computational neuroscience, 30 (2011), pp. 17–44.

[213] K. R. Rad and L. Paninski, *Efficient, adaptive estimation of two-dimensional firing rate surfaces via Gaussian process methods.*, Network: Computation in Neural Systems, 21 (2010), pp. 142–68.

[214] R. P. N. Rao and D. H. Ballard, *Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects.*, Nature Neuroscience, 2 (1999), pp. 79–87.

[215] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, *DeepMatching: Hierarchical Deformable Dense Matching*, International Journal of Computer Vision, 120 (2016), pp. 1–24.

[216] P. Reynaud-Bouret, V. Rivoirard, and C. Tuleau-malot, *Inference of functional connectivity in neurosciences via Hawkes processes*, Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE, (2013), pp. 0–3.

[217] F. Rieke, W. David, R. de Ruyter van Steveninck, and W. Bialek, *Spikes: exploring the neural code*, MIT Press, 1999.

[218] J. M. Rodriguez, F. Gomez Fernandez, M. E. Buemi, and J. Jacobo-Berlles, *Dynamic textures segmentation with GPU*, in Lecture Notes in Computer Science, vol. 7441 LNCS, 2012, pp. 607–614.

[219] P. T. Sadtler, K. M. Quick, M. D. Golub, S. M. Chase, S. I. Ryu, E. C. Tyler-Kabara, B. M. Yu, and A. P. Batista, *Neural constraints on learning*, Nature, 512 (2014), pp. 423–426.

[220] M. Sahani and J. Linden, *Evidence optimization techniques for estimating stimulus-response functions*, Advances in Neural Information Processing Systems, 15 (2003).

[221] M. Saleh, K. Takahashi, and N. G. Hatsopoulos, *Encoding of coordinated reach and grasp trajectories in primary motor cortex*, The Journal of neuroscience, 32 (2012), pp. 1220–1232.

[222] T. D. Sanger, *Theoretical considerations for the analysis of population coding in motor cortex*, Neural Computation, 6 (1994), pp. 29–37.

[223] A. Schödl, A. Haro, and I. A. Essa, *Head Tracking Using a Textured Polygonal Model*, in PUI98, no. August, 1998, pp. 43–48.

[224] T. Schreiber, *Measuring information transfer*, Physical review letters, 85 (2000), pp. 461–464.

[225] J. Schulman, A. Lee, J. Ho, and P. Abbeel, *Tracking Deformable Objects with Point Clouds*, in IEEE International Conference on Robotics and Automation (ICRA), 2013, 2013.

[226] A. B. Schwartz, *Perspective Movement : How the Brain Communicates with the World*, Cell, 164 (2016), pp. 1122–1135.

[227] O. Schwartz, J. W. Pillow, N. C. Rust, and E. P. Simoncelli, *Spike-triggered neural characterization.*, Journal of vision, 6 (2006), pp. 484–507.

[228] G. Schwarz, *Estimating the dimension of a model*, The annals of statistics, 6 (1978), pp. 461–464.

[229] S. S. H. Scott, *Optimal feedback control and the neural basis of volitional motor control*, Nature Reviews Neuroscience, 5 (2004), pp. 534–546.

[230] J. Semedo, A. Zandvakili, A. Kohn, C. K. Machens, and B. M. Yu, *Extracting Latent Structure From Multiple Interacting Neural Populations*, Advances in neural information processing systems, (2014), pp. 1–9.

[231] L. E. L. Sergio, C. Hamel-Pâquet, and J. F. Kalaska, *Motor cortex neural correlates of output kinematics and kinetics during isometric-force and arm-reaching tasks*, Journal of neurophysiology, 94 (2005), pp. 2353–2378.

[232] E. Sernagor, S. J. Eglen, and M. J. O'Donovan, *Differential effects of acetylcholine and glutamate blockade on the spatiotemporal dynamics of retinal waves.*, The Journal of neuroscience, 20 (2000), p. RC56.

[233] E. Sernagor, C. Young, and S. J. Eglen, *Developmental modulation of retinal wave dynamics: shedding light on the GABA saga.*, The Journal of neuroscience, 23 (2003), pp. 7621–9.

[234] T. Serre, G. Kreiman, M. Kouh, C. Cadieu, U. Knoblich, and T. Poggio, *A quantitative theory of immediate visual recognition*, Progress in Brain Research, 165 (2007), pp. 33–56.

[235] S. Shamma and J. Fritz, *Adaptive auditory computations*, Current Opinion in Neurobiology, 25 (2014), pp. 164–168.

[236] P.-C. Shao, J.-J. Huang, W.-C. Shann, C.-T. Yen, M.-L. Tsai, and C.-C. Yen, *Granger causality-based synaptic weights estimation for analyzing neuronal networks.*, Journal of computational neuroscience, 38 (2015), pp. 483–497.

[237] M. Sharma, C. Gaona, J. Roland, N. Anderson, Z. Freudenberg, and E. Leuthardt, *Ipsilateral Directional Encoding of Joystick Movements in Human Cortex*, in IEEE Eng Med Biol Soc, 2009, pp. 5502–5505.

[238] T. Sharpee, W. Bialek, and N. Rust, *Analyzing neural responses to natural signals*, 250 (2002), pp. 223–250.

[239] W. L. Shew and D. Plenz, *The functional benefits of criticality in the cortex.*, The Neuroscientist, 19 (2013), pp. 88–100.

[240] W. L. Shew, H. Yang, T. Petermann, R. Roy, and D. Plenz, *Neuronal avalanches imply maximum dynamic range in cortical networks at criticality.*, The Journal of neuroscience : the official journal of the Society for Neuroscience, 29 (2009), pp. 15595–600.

[241] S. Shoham, L. M. Paninski, M. R. Fellows, N. G. Hatsopoulos, J. P. Donoghue, and R. a. Normann, *Statistical encoding model for a primary motor cortical brain-machine interface.*, IEEE transactions on bio-medical engineering, 52 (2005), pp. 1312–22.

[242] A. Shojaie and G. Michailidis, *Discovering graphical Granger causality using the truncating lasso penalty*, Bioinformatics, 26 (2010), pp. i517–i523.

[243] E. Simoncelli and L. Paninski, *Characterization of neural responses with stochastic stimuli*, in The new cognitive neurosciences, MIT Press, 3rd ed., 2004.

[244] W. S. Smith and E. E. Fetz, *Synaptic Linkages Between Corticomotoneuronal Cells Affecting Forelimb Muscles in Behaving Primates*, Journal of Neurophysiology, 102 (2009).

[245] D. Smyth, B. Willmore, G. E. Baker, I. D. Thompson, and D. J. Tolhurst, *The Receptive-Field Organization of Simple Cells in Primary Visual Cortex of Ferrets under Natural Scene Stimulation*, The Journal of Neuroscience, 23 (2003), pp. 4746–4759.

[246] D. Soudry, S. Keshri, P. Stinson, M. H. Oh, G. Iyengar, and L. Paninski, *Efficient "Shotgun" Inference of Neural Connectivity from Highly Sub-sampled Activity Data*, PLoS Computational Biology, 11 (2015), pp. 1–30.

[247] O. Sporns, *Brain connectivity*, Scholarpedia, 2 (2007), p. 4695.

[248] B. Stafford, A. Sher, A. M. Litke, and D. A. Feldheim, *Spatial-Temporal Patterns of Retinal Waves Underlying Activity- Dependent Refinement of Retinofugal Projections*, Neuron, 64 (2009), pp. 200–212.

[249] D. Stellwagen, C. J. Shatz, and M. B. Feller, *Dynamics of retinal waves are controlled by cyclic AMP*, Neuron, 24 (1999), pp. 673–685.

[250] I. Stevenson, J. Rebesco, L. Miller, and K. Kording, *Inferring functional connections between neurons*, Current opinion in neurobiology, 18 (2008), pp. 582–588.

[251] I. Stevenson, J. M. Rebesco, N. G. Hatsopoulos, Z. Haga, L. E. Miller, and K. P. Kording, *Bayesian inference of functional connectivity and network structure from spikes*, IEEE Transactions on Neural Systems and Rehabilitation Engineering, 17 (2009), pp. 203–213.

[252] I. H. Stevenson and K. P. Kording, *How advances in neural recording affect data analysis*, Nature Neuroscience, 14 (2011), pp. 139–142.

[253] I. H. Stevenson, B. M. London, E. R. Oby, N. A. Sachs, J. Reimer, B. Englitz, S. V. David, S. A. Shamma, T. J. Blanche, K. Mizuseki, A. Zandvakili, N. G. Hatsopoulos, L. E. Miller, and K. P. Kording, *Functional connectivity and tuning curves in populations of simultaneously recorded neurons.*, PLoS computational biology, 8 (2012), p. e1002775.

[254] M. Stone, *An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion*, Journal of the Royal Statistical Society. Series B, 39 (1977), pp. 44–47.

[255] A. J. Suminski, D. C. Tkach, A. H. Fagg, and N. G. Hatsopoulos, *Incorporating Feedback from Multiple Sensory Modalities Enhances Brain-Machine Interface Control*, Journal of Neuroscience, 30 (2010), pp. 16777–16787.

[256] M. M. Syed, S. Lee, J. Zheng, and Z. J. Zhou, *Stage-dependent dynamics and modulation of spontaneous waves in the developing rabbit retina.*, The Journal of physiology, 560 (2004), pp. 533–49.

[257] K. Takahashi, L. Pesce, M. Best, S. Kim, T. P. Coleman, N. G. Hatsopoulos, and C. F. Ross, *Granger causality analysis of state dependent functional connectivity of neurons in orofacial motor cortex during chewing and swallowing*, in SCIS-ISIS, 2012, pp. 1067–1071.

[258] D. M. Taylor, S. I. H. Tillery, and A. B. Schwartz, *Direct cortical control of 3D neuroprosthetic devices.*, Science, 296 (2002), pp. 1829–32.

[259] U. Technau and R. E. Steele, *Evolutionary crossroads in developmental biology : Cnidaria*, Development, 138 (2011), pp. 1447–1458.

[260] D. Terzopoulos and K. Waters, *Analysis and synthesis of facial image sequences using physical and anatomical models*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 15 (1993), pp. 569 – 579.

[261] D. Terzopoulos, A. Witkin, and M. Kass, *Symmetry-Seeking Models and 3D Object Reconstruction*, International Journal of Computer Vision, 221 (1987), pp. 211–221.

[262] E. Theunissen, S. V. David, N. C. Singh, A. Hsu, W. E. Vinje, and J. L. Gallant, *Estimating spatio-temporal receptive fields of auditory and visual neurons from their responses to natural*, Network: Computation in Neural Systems, 12 (2001), pp. 289–316.

[263] R. Tibshirani, *Regression shrinkage and selection via the lasso*, Journal of the Royal Statistical Society. Series B, 58 (1996), pp. 267–288.

[264] H. Toda and T. Yamamoto, *Statistical inference in vector autoregressions with possibly integrated processes*, Journal of econometrics, 66 (1995), pp. 225–250.

[265] W. Truccolo, U. T. Eden, M. R. Fellows, J. P. Donoghue, and E. N. Brown, *A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects.*, Journal of neurophysiology, 93 (2005), pp. 1074–89.

[266] M. Turk and A. Pentland, *Face Recognition Using Eigenfaces*, Journal of Cognitive Neuroscience, 3 (1991), pp. 72 – 86.

[267] V. Venkatachalam, N. Ji, X. Wang, C. Clark, J. K. Mitchell, M. Klein, C. J. Tabone, J. Florman, H. Ji, J. Greenwood, A. D. Chisholm, J. Srinivasan, M. Alkema, M. Zhen, and A. D. T. Samuel, *Pan-neuronal imaging in roaming Caenorhabditis elegans.*, Proceedings of the National Academy of Sciences of the United States of America, 113 (2016), pp. E1082–8.

[268] J. D. Victor, F. Mechler, M. a. Repucci, K. P. Purpura, T. Sharpee, A. M. Schmid, D. Jonathan, and P. Keith, *Responses of V1 neurons to two-dimensional hermite functions.*, Journal of neurophysiology, 95 (2006), pp. 379–400.

[269] H. White, *Maximum likelihood estimation of misspecified models*, Econometrica: Journal of the Econometric Society, 50 (1982).

[270] P. Whittle, *The analysis of multiple stationary time series*, Journal of the Royal Statistical Society. Series B, 15 (1953), pp. 125–139.

[271] M. Wibral, R. Vicente, and J. Lizier, *Directed information measures in neuroscience*, Springer, 2014.

[272] N. Wiener, *The theory of prediction*, Modern mathematics for engineers, 1 (1956), pp. 165–183.

[273] R. S. Williamson, M. Sahani, and J. W. Pillow, *The equivalence of information-theoretic and likelihood-based methods for neural dimensionality reduction.*, PLoS computational biology, 11 (2015), p. e1004141.

[274] R. P. Wishner, J. a. Tabaczynski, and M. Athans, *A Comparison of Three Non-Linear Filters*, Automatica, 5 (1969), pp. 478–496.

[275] D. Wolpert and Z. Ghahramani, *Computational principles of movement neuroscience*, Nature Neuroscience, 3 (2000), pp. 1212–1217.

[276] R. Wong, *Retinal Waves and Visual System Development*, Annual review of neuroscience, 22 (1999), pp. 29–47.

[277] R. Wong, A. Chernjavsky, S. Smith, and C. Shatz, *Early functional neural networks in the developing retina*, Nature, 374 (1995), pp. 716–718.

[278] R. O. Wong and D. M. Oakley, *Changing patterns of spontaneous bursting activity of on and off retinal ganglion cells during development.*, Neuron, 16 (1996), pp. 1087–95.

[279] S. Wood, *Generalized additive models: an introduction with R*, CRC press, 2006.

[280] J. Xiao, S. Baker, I. Matthews, and T. Kanade, *Real-time combined 2D+3D active appearance models*, in IEEE Conference on Computer Vision and Pattern Recognition, 2004, pp. 535–542.

[281] H. Xu, M. Furman, Y. S. Mineur, and H. Chen, *An instructive role for patterned spontaneous retinal activity in mouse visual map development*, Neuron, 70 (2011), pp. 1115–1127.

[282] L. Xu, J. Chen, and J. Jia, *A Segmentation Based Variational Model for Accurate Optical Flow Estimation*, Proceedings of the 10th European Conference on Computer Vision - ECCV '08, (2008), pp. 671–684.

[283] J. Yang, H. Li, Y. Dai, and R. Tan, *Robust Optical Flow Estimation of Double-Layer Images under Transparenflection*, in IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1–10.

[284] B. M. Yu, J. P. Cunningham, G. Santhanam, S. I. Ryu, K. V. Shenoy, and M. Sahani, *Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity*, Journal of Neurophysiology, 102 (2009), p. 614.

[285] B. M. Yu, J. P. Cunningham, G. Santhanam, S. I. Ryu, K. V. Shenoy, and M. Sahani, *Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity*, Journal of Neurophysiology, 102 (2009), p. 614.

[286] M. Yuan and Y. Lin, *Model selection and estimation in regression with grouped variables*, Journal of the Royal Statistical Society: Series, 1 (2006), pp. 49–67.

[287] J. Zheng, S. Lee, and Z. J. Zhou, *A developmental switch in the excitability and function of the starburst network in the mammalian retina*, Neuron, 44 (2004), pp. 851–864.

[288] ——, *A transient network of intrinsically bursting starburst cells underlies the generation of retinal waves.*, Nature neuroscience, 9 (2006), pp. 363–371.