Parameter Estimation, Optimal Control and Optimal Design in Stochastic Neural Models

Alexandre V. Iolov

Thesis submitted to the Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Mathematics [1]

Department of Mathematics and Statistics
Faculty of Science
University of Ottawa

[1] The Ph.D. program is a joint program with Carleton University, administered by the Ottawa-Carleton Institute of Mathematics and Statistics

Abstract

This thesis solves estimation and control problems in computational neuroscience, mathematically dealing with the first-passage times of diffusion stochastic processes. We first derive estimation algorithms for model parameters from first-passage time observations, and then we derive algorithms for the control of first-passage times. Finally, we solve an optimal design problem which combines elements of the first two: we ask how to elicit first-passage times such as to facilitate model estimation based on said first-passage observations.

The main mathematical tools used are the Fokker-Planck partial differential equation for evolution of probability densities, the Hamilton-Jacobi-Bellman equation of optimal control and the adjoint optimization principle from optimal control theory.

The focus is on developing computational schemes for the solution of the problems. The schemes are implemented and are tested for a wide range of parameters.

Acknowledgements

I would like to thank my supervisor, Andre Longtin, my co-supervisor Professor Susanne Ditlevsen at Copenhagen University, my colleagues at Ottawa and Copenhagen, my parents, and Kirsten, Daniel and Eva.

Also I would like to thank NSERC Canada for funding my studies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis deals with problems in parameter estimation and stochastic optimal control arising in computational neuroscience. Neuroscience is the study of how information is processed by the nervous system of living organisms. Its main building block is the single neuron cell. A neural cell maintains a certain membrane potential, $v(t)$, an electrochemical gradient between its interior and exterior. Fluctuations in the membrane potential, v, form the key mechanism by which information is processed and transmitted. For our purposes we will assume that v is essentially uniform throughout the neuron - that is we will idealize the neuron as a single point in space. In all neural systems, information is processed and transmitted by the sharp, transient changes in the membrane potential called spikes. Currently, the dominant theory is that the exact shape of the potential excursion is irrelevant, but that all information is contained entirely in the time occurrence of the spike or equivalently the information is in the duration of the interval between two subsequent spikes.

There is a rich history on modelling the dynamics of the membrane potential starting from the Nobel-prize winning work of Hodgkin and Huxley in the 1950's. Their model, still widely considered as a benchmark for the dynamics of the membrane potential, is a 4-dimensional ordinary differential equation (ODE), where one

of the states is the voltage itself and the other three are phenomenological equations describing the behaviour of ion channels responsible for the generation of the non-linear excursion - the spike itself. There have been many reductions of the Hodgkin and Huxley model to a smaller dimension, including the Fitzhugh-Nagumo model and the Morris-Lecar model, which still retain the basic excursion non-linearity of the spiking mechanism, [1]. An even-more drastic simplification is to linearize the dynamics and then impose the non-linear voltage excursion of the neural spike artificially by detecting positive-going crossings of a threshold. This leads to the leaky integrate-and-fire (LIF) model, which forms the foundation of all the models used in our work.

The response of real neurons to identical stimuli, especially *in vivo*, is stochastic. Given the same stimulus the resulting inter-spike intervals will not be the same. There are several reasons and explanations for why that is. The simplest one is that, in addition to the stimulus delivered by the experimentalist, the neuron is bombarded by extraneous, random stimulation, for example from other neurons. Some of this stochasticity also arises from random openings and closings of the ion channels in the membrane, which causes conductance fluctuations, see [2]. To account for this random behaviour the dynamics of the membrane potential, $v(t)$, are modelled as following a stochastic differential equation (SDE). SDEs are a generalization of ODEs, which allow for stochastic terms.

This thesis deals with two main topics: 1) estimation of single-cell neural models and 2) optimal control of neural dynamics. Mathematically, this amounts to estimation of parameters in SDEs and optimizing functionals of SDEs. A complication arising from practical limitations of experimental neuroscience is that often the exact value of the voltage is difficult to observe, and instead it is only the spikes that are distinctly observable. This is because the voltage measurement across the membrane requires that an electrode to be stably inserted in the cell for a certain time; otherwise, one can infer spike times from relatively easier (especially for small or hard-to-reach

cells) extracellular recordings. Mathematically, this implies that our estimation and control algorithms will often be based on observation of *first passage* times of the system rather than detailed voltage trajectories. This makes the problem significantly more challenging and requires some non-standard algorithms. At the same time, the study of First-Passage Times (FPTs) is a classic part of Stochastic Analysis and there are many known techniques and results that we can build on.

The three problems that are discussed in this thesis can be summarized as such: in the first problem, we estimate parameters from spike observations in the context of a periodically driven system; in the second, we discuss how to control a neuron to spike at some pre-selected time; and in the third problem, we combine both approaches, i.e. we control the system in order to best estimate its parameters. The core of the work revolves around estimating and controlling the classic Ornstein-Uhlenbeck process with an absorbing barrier. Apart from advances on the mathematical front, involving extensions of estimation and optimal control theory to new settings - in particular the low information setting given by spike time observations - the work is relevant to a growing body of research that aims to monitor and control brain processes, e.g. for alleviating epilepsy, see [3], or the book [4], for example.

Much of the work in the thesis requires numerical implementation in order to ascertain that it actually works, i.e. that reliable estimates can be obtained or that a system can be controlled. This might explain why some of the problems tackled herein have not been tackled to the same extent previously. Also, while estimation and control share similar mathematical concepts, they are also related on a practical level - a control algorithm requires a reliable description of the controlled system and an estimation algorithm is used precisely to obtain such a description. Conversely, we show in the third chapter that control techniques can be fruitfully used to improve estimation. The combination of elegant mathematics and seemingly immediate practical applications makes the subjects of control and estimation quite interesting and I have found it to be even more interesting when considered as a joint subject rather

than as two separate ones.

The thesis is structured as follows: First, in chapter 2 we introduce the mathematical background from stochastic processes, parameter estimation and optimal control used in the rest of the thesis; in section 2.2 we discuss the standard stochastic models of a single neuron. The next three chapters, chapters 3 to 5 form the novel portion of this thesis where we discuss each of the three problems mentioned above. These three chapters have been adapted from journal papers either accepted for publication or currently in review. Chapter 3 has been published in the Journal of Mathematical Neuroscience (as [5]); chapter 4 has been published in Journal of Neural Engineering (as [6]), it also forms the basis of an invited chapter in [7]; chapter 5 has been submitted to SIAM's Journal of Uncertainty Quantification (as [8]). Each chapter has been stylistically revised to achieve a uniform notation and to fit into the overall flow of the thesis, however the bulk of the scientific content in the chapters is common to the original articles.

In the conclusion, chapter 6, we summarize the thesis' main findings and provide a brief outlook.

Chapter 2

Mathematical Background

Here we collect a list of mathematical tools that are used in the thesis. We first point the reader to the general references by Oksendal for SDEs [9] and by Fleming and Rishel for Optimal Control [10]. Another very readable introduction to the field of both SDEs and Optimal Control are the online notes of Professor L. Evans [11, 12]. We have also used Jacobs as our main tutorial on first-passage times for SDEs [13].

2.1 Stochastic Differential Equations

In view of our ultimate goals, we will restrict ourselves to stochastic processes whose sample paths are continuous, and more specifically to SDEs driven by Brownian motion. We will not provide proofs of results in this section, but only state the results with a view towards establishing the notation for the rest of the thesis. We will assume that the reader is familiar with the following concepts:

- a probability space $\{\Omega, \mathcal{A}, P\}$, where Ω denotes the probability space, $\mathcal{A}$ a sigma-algebra, and P a probability measure
- a continuous-time stochastic process, X_t
- a filtration $\mathcal{F}_t$, in particular the filtration generated by a stochastic process.

2.1.1 The Wiener Process and the Itô Integral

The Wiener Process is the fundamental building block of the stochastic calculus. It is often called Brownian Motion and we will denote it $\{W_t\}_{t\geq 0}$.

Definition 2.1.1. *Wiener Process, W_t:*

1. $W_0 = 0$.
2. $W_t - W_s \sim N(0, |t-s|)$, *i.e. the process increments are normally distributed with mean 0 and variance* $|t-s|$.
3. $\forall\{t_i\}_1^N, \quad \{W_{t_i} - W_{t_{i-1}}\}_2^N \sim$ *independent, i.e. the process has independent increments.*

The fact that the finite incremental distributions suffice to specify a unique continuous-time stochastic process is known as Kolmogorov's extension theorem. Now, we collect a few more relevant properties of the process W_t and its sample paths:

1. The sample paths of W_t are almost surely (a.s.) continuous and are in fact Holder continuous for any exponent $\gamma < 1/2$
2. The sample paths of W_t are nowhere differentiable
3. W_t is a Markov process: $\mathbb{P}[W_t \in B|\sigma(W_{s'\leq s})] = \mathbb{P}[W_t \in B \,|\, W_s]$, for any Borel set, $B \subset \mathbb{R}$, where $\sigma(W_{s'\leq s})$ is the filtration generated by the W_t.

We now turn to defining stochastic integrals based on the Wiener Process.

Definition 2.1.2. *Progressively Measurable Functions:*

Let $\mathcal{F}_t$ be the filtration generated by the Wiener Process.

Let X_t be a stochastic process which is $\mathcal{F}_t$-measurable $\forall t$ and which is jointly measurable in (t, ω). We call such an X_t progressively measurable.

Definition 2.1.3. $\mathbb{L}^2, \mathbb{L}^1$

We define $\mathbb{L}^2[0,T]$ *as the space of all progressively measurable* X *such that*

$$\mathbb{E}\left[\int_{[0,T]} X_t^2 \,\mathrm{d}t\right] < \infty$$

Similarly, we define $\mathbb{L}^1[0,T]$ *as the space of all progressively measurable* X *such that*

$$\mathbb{E}\left[\int_{[0,T]} |X_t| \,\mathrm{d}t\right] < \infty$$

$\mathbb{L}^2$ will be the class of functions for which the Itô integral is well-defined.

Definition 2.1.4. *Itô Integral:*

Let $P^n := a = t_1^n \dots t_{m_n}^n = b$ *be a partition of the interval* $[a,b] \subset [0,\infty)$. *Let* $|P_n| = \sup_i |t_i - t_{i-1}|$. *Let* $\lim_{n\to\infty} |P_n| \to 0$ *and consider an* $X_t \in \mathbb{L}^2[a,b]$,

then

$$\int_{[a,b]} X_t \,\mathrm{d}W := \lim_{n\to\infty} \sum_{i=1}^{m_n} X_{t_i} \left(W(t_{i+1}) - W(t_i)\right) \tag{2.1.1}$$

To be precise, our definition is actually a theorem, and the real definition is one that uses step functions and passes to the limit. Also we will write the limits of integration $\int_0^T \cdot \,\mathrm{d}W$ or $\int_{[0,T]} \cdot \,\mathrm{d}W$ interchangeably.

Again, we state without proof a few interesting properties of $\int X_t \,\mathrm{d}W$.

Theorem 2.1.1. *Itô Integral Properties*

1. $\mathbb{E}[\int_0^T X_t \,\mathrm{d}W] = 0$
2. $\mathbb{E}[\left(\int_0^T X_t \,\mathrm{d}W\right)^2] = \mathbb{E}[\int_0^T X_t^2 \,\mathrm{d}t]$
3. $I(t) = \int_0^t X_t \,\mathrm{d}W$ *is a martingale*
4. $I(t) = \int_0^t X_t \,\mathrm{d}W$ *has continuous sample paths.*

2.1.2 Itô SDEs and Itô's Lemma

Definition 2.1.5 (Itô SDE)**.** *Let $B(x,t) : \mathbb{R} \times [0,T] \to \mathbb{R}$, $G(x,t) : \mathbb{R} \times [0,T] \to \mathbb{R}$, be given functions. We say that a stochastic process X satisfies:*

$$dX = Bdt + GdW \tag{2.1.2}$$

over $[0,T]$ if

1. *X is progressively measurable wrt. $\mathcal{F}_t$*
2. *$B(X,t) \in \mathbb{L}^1[0,T]$*
3. *$G(X,t) \in \mathbb{L}^2[0,T]$*
4. *$X_t = X_0 + \int_0^t B(X_s,s)\,\mathrm{d}s + \int_0^t G(X_s,s)\,\mathrm{d}W_s$.*

We are now ready to present the celebrated Itô Lemma which is the chain-rule of stochastic calculus.

Theorem 2.1.2 (Itô Lemma)**.** *Suppose X satisfies the stochastic differential $dX_t = B(X,t)dt + G(X,t)dW$ as above and take $v(x,t) \in C^{2,1}[\mathbb{R} \times [0,T]]$, that is v is twice-continuously differentiable wrt. its first argument and once wrt. its second argument.*

Set $Y(t) = v(X,t)$

then

$$dY = \left(\partial_t v + \partial_x v \cdot B + \partial_x^2 v \cdot \frac{G^2}{2}\right)\,\mathrm{d}t + (\partial_x v \cdot G)\,\mathrm{d}W$$

As an example we will discuss the Ornstein-Uhlenbeck process, which is the basis for the models we will face later on.

Example 2.1.2.1 (O-U Process)**.** *Let X_t follow:*

$$dX = \left(\frac{\mu - X_t}{\tau}\right)dt + \beta dW \tag{2.1.3}$$

with an initial condition, X_0, which may be an arbitrary distribution independent of the Wiener Process, W. We can solve this as follows:

$$\begin{aligned} dX &= \left(\frac{\mu - X_t}{\tau}\right) dt + \beta dW \\ dX + \frac{X_t}{\tau} dt &= \frac{\mu}{\tau} dt + \beta dW \\ e^{t/\tau} dX + e^{t/\tau} \frac{X_t}{\tau} dt &= e^{t/\tau} \frac{\mu}{\tau} dt + \beta e^{t/\tau} dW \\ X e^{t/\tau} - X_0 &= \int e^{t/\tau} \frac{\mu}{\tau} dt + \int \beta e^{t/\tau} dW \\ X_t &= e^{-t/\tau} X_0 + \mu(1 - e^{-t/\tau}) + \frac{\sqrt{\tau}\beta e^{-t/\tau}}{\sqrt{2}} W(e^{2t/\tau} - 1) \end{aligned}$$

which means that X forgets its initial conditions exponentially fast and converges to a normal random variable with mean μ and variance $\frac{\tau\beta^2}{2}$.

At the end of this sub-section, we state Itô's Lemma for a multidimensional state.

Theorem 2.1.3 (Itô Lemma for $X \in R^n$, $dW \in \mathbb{R}^m$)**.** *Suppose X satisfies the stochastic differential $dX_t = B(X,t)dt + G(X,t)dW$, where $B : [R^n \times [0,T] \to \mathbb{R}^n]$, $G : [R^n \times [0,T] \in \mathbb{R}^{n \times m}$ and $dW = \left(dW^{(i)}\right) \in \mathbb{R}^m$ is a vector of independent Wiener Processes. Take $u(x,t) \in C^{2,1}[\mathbb{R}^n \times [0,T]]$. Let $D_{ij} := \frac{1}{2}\left[\sum_k G_{ik} G_{jk}\right]_{ij}$. Set $Y(t) = u(X(t),t)$. Then*

$$dY = \left(\partial_t u + \sum_i \partial_{x_i} u \cdot B_i + \sum_{i,j} \partial^2_{x_i x_j} u \cdot D_{ij}\right) \mathrm{d}t + \left(\sum_{ij} \partial_{x_i} u \cdot G_{ij} dW^{(j)}\right)$$

2.1.3 Fokker-Planck and Kolmogorov's Backward Equations

The Fokker-Planck equation and Kolmogorov's Backward equation describe the forward (resp. backward) evolution of the probability density of X_t.

For the rest of this section we will work in $\mathbb{R}^n$, i.e. X will be an $n-$dimensional vector that satisfies the stochastic differential

$$dX_t = B(X,t)dt + G(X,t)dW, \tag{2.1.4}$$

using the notation defined in Theorem 2.1.3.

Let

$$f(x,t|y,s)\,\mathrm{d}x = \mathbb{P}[X_t \in \mathrm{d}x|X_s = y] \tag{2.1.5}$$

be the transition probability density. Then f satisfies:

$$\partial_t f(x,t|y,s) = -\sum_i \partial_{x_i}\left[B_i(x,t)f(x,t|y,s)\right] + \sum_{i,j}\partial^2_{x_i x_j}\left[D_{ij}(x,t)f(x,t|y,s)\right]. \tag{2.1.6}$$

This is called the *Fokker-Planck* or *Forward Kolmogorov* equation. It can be seen as an equation describing the conservation of probability. To this end define the probability current $\phi \in \mathbb{R}^n$ and its ith component $\phi_i \in \mathbb{R}$ as:

$$\phi_i = B_i(x,t)f(x,t|y,s) - \sum_j \partial_{x_j}\left[D_{ij}(x,t)f(x,t|y,s)\right]. \tag{2.1.7}$$

Then the Fokker-Planck equation, eq. (2.1.6), can be written as:

$$\partial_t f = -\nabla\cdot\phi, \tag{2.1.8}$$

which just says that the change in probability is the difference between the flow in and the flow out.

Initial conditions for f are given by the distribution of X_s. Boundary conditions (BCs) depend on the domain of X and what happens to X once it hits its boundary. If the domain is all of $\mathbb{R}^n$ then we only insist that $\lim_{|x|\to\infty} f = 0$. If the domain has boundaries, then there are two common scenarios which we will also encounter in the sequel:

1. absorbing BCs
2. reflecting BCs.

At an absorbing boundary, the particle X is removed and $f = 0$ there. In this situation, we will not have conservation of probability and the integral of f over X's domain will monotonically decrease.

At a reflecting boundary, the particle X bounces back into its domain and we will have that the probability current $\phi \cdot n = 0$ where n is the outward normal at the reflecting boundary.

Now consider f as a function of the initial values y, s, holding the terminal values x, t fixed. Then

$$-\partial_s f(x,t|y,s) = \sum_i \left[B_i(y,s) \cdot \partial_{y_i} f(x,t|y,s)\right] + \sum_{i,j} \left[D_{ij}(y,s)\partial^2_{y_i y_j} f(x,t|y,s)\right] \quad (2.1.9)$$

This is the *Backward Kolmogorov* equation for f. Note the minus sign in front $\partial_s f$. A mnemonic for the signs of the Backward vs. Forward equation is that as t increases in the forward case, f diffuses and so ∂_t and ∂_x^2 have the same sign; in the backward case, as s increases, that is as s approaches t, f anti-diffuses and the two partial derivatives have opposite signs.

The differential operator on the right-hand side of eq. (2.1.9) occurs often in the study of SDEs and has its own name.

Definition 2.1.6 (Generator of an SDE)**.** *The Generator A of X is defined by:*

$$A[\psi(x)] = \lim_{t \searrow 0^+} \frac{\mathbb{E}[\psi(X_t)] - \psi(x)}{t}; \quad X_0 = x \in \mathbb{R}^n$$

Theorem 2.1.3.1. *For an Itô SDE as in eq.* (2.1.4)

$$A[\psi(x)] = \sum_i B_i(x,t) \cdot \partial_{x_i}\psi + \sum_{i,j} D_{ij}(x,t)\partial^2_{x_i x_j}\psi$$

2.1.4 First-Hitting Times

Most of the problems in this thesis are related to *first-hitting times*.

Definition 2.1.7. *Let $\mathcal{F}(t)$ be some filtration, a random variable τ is called a* stopping time *if*

$$\{\omega : T(\omega) \leq t\} \in \mathcal{F}_t \, \forall t$$

The colloquial way of describing stopping times is that at any time we know whether T has occurred or not. For a counterexample, the time that a Wiener Process achieves its maximum over some interval is not a stopping time, since at any given time, we do not know if the maximum has occurred or not. The most common example of a stopping time is the first hitting-time, which is the first time X_t leaves or enters some set.

Theorem 2.1.4 (First-Hitting time)**.** *Let $E \subset \mathbb{R}^n$ be open or closed and non-empty then*

$$T := \inf\{t \geq 0 | X_t \notin E\}$$

is a stopping time.

The reason stopping times are very useful is that all the facts so far quoted for Itô calculus using integrals $\int_0^t \mathrm{d}W$ remain true if the fixed time t is replaced by a stopping time T. Also it allows us to link SDE's and PDEs using the generator, A, of the diffusion:

Theorem 2.1.5 (Dynkin's formula)**.** *Given T a stopping time, $\mathbb{E}[T] < \infty$. Then:*

$$\mathbb{E}[u(X_T, T)|X_0 = x] = u(x, 0) + \mathbb{E}\left[\int_0^T \partial_t u(X_t, t) + A[u(X_t, t)|X_0 = 0]\,\mathrm{d}t\right].$$

This provides a link between PDEs and stochastic processes and allows us to go back and forth in that we can find probabilistic results by solving a PDE or we can approximate a PDE by simulating a stochastic process and averaging.

This thesis is primarily concerned with the analysis of first-hitting times. Thus we will often refer to the probability of a first-hitting time, T, to take the value t as

$$g(t) := \frac{1}{\mathrm{d}t}\Pr(T \in [t, t + \mathrm{d}t)|X_0 = x_0).$$

There is a simple relation between the hitting-time density, g, and the transition density of X_t. Let T be the first-hitting time of X to the boundary ∂E of the closed

domain $E \subset \mathbb{R}^n$, i.e. T is the first time that X_t leaves E. Then if f solves the Fokker-Planck equation for X, such that $f \equiv 0$ on $\mathbb{R}^n \setminus E$, and if ϕ is its probability flux defined in eq. (2.1.7), then we will have that

$$g(t) = \int_{\partial E} \phi \cdot n \, \mathrm{d}S, \tag{2.1.10}$$

where n is an outward normal to E, the integral is a surface integral.

We give a quick derivation of eq. (2.1.10). Let

$$G(t) = \Pr(T \leq t | X_0 = x_0) = \int_0^t g(s) \, \mathrm{d}s$$

be the cumulative distribution corresponding to g. Then $G(t)$ is the probability that at time t the particle has left the domain and so

$$G(t) = 1 - \int_E f(x, t|x_0, 0) \, \mathrm{d}x$$

Since $g(t) = \partial_t G(t)$, we then have that

$$g(t) = -\partial_t \int_E f(x, t|x_0, 0) \, \mathrm{d}x = \int_E \nabla \cdot \phi(x, t|x_0, 0) \, \mathrm{d}x = \int_{\partial E} \phi \cdot n \, \mathrm{d}S.$$

where the second equality comes from the probability form of the Fokker-Planck equation, eq. (2.1.8), and the third equality is the divergence (Gauss') theorem.

In the simplest case, of working in 1-D, such that $E = (-\infty, x_{th})$ and $x_{th} > x_0$, we will have

$$g(t) = \phi(x_{th}, t) - \phi(-\infty, t) = \phi(x_{th}, t),$$

since, at the lower boundary the probability flux $\phi(-\infty, t) = 0$. Using the expression for ϕ from eq. (2.1.7), we arrive at

$$g(t) = B(x_{th}, t) f(x_{th}, t) - \partial_x [D(x_{th}, t) \cdot f(x_{th}, t)] = -\partial_x [D(x_{th}, t) \cdot f(x_{th}, t)].$$

Since, $f(x_{th}, t) = 0$. In neural applications this is the form of g, that we will encounter.

2.1.5 Numerical Simulation of SDEs

In general SDEs, such as eq. (2.1.4), cannot be solved analytically and if one wants to obtain approximate paths from their solution, one needs numerical methods. There are many numerical methods for approximating an SDE as in eq. (2.1.4), see e.g. [14] for a popular introduction. Here we will only describe the most basic method, the Euler-Maruyama scheme, which will suffice for our purposes. In the Euler-Maruyama scheme, the approximate solution to eq. (2.1.4) is computed at predetermined time-nodes, $\{t_k\}_{k=0}^N$ with time intervals $\Delta t_k = t_{k+1} - t_k$, which are often assumed to be constant, i.e. $\Delta t_k = \Delta t \ \forall k$. Then given an initial condition $X_0 = X(t_0)$, the approximate path, $\{X_k\}_{k=0}^N = \{X(t_k)\}_{k=0}^N$ is obtained iteratively via:

$$X_{k+1} = B(X_k, t_k)\Delta t_k + G(X_k, t_k)\xi_k\sqrt{\Delta t_k} \tag{2.1.11}$$

where ξ_k is an independent draw from the standard normal distribution.

2.2 Mathematical Models in Neuroscience

We now describe in detail the basic mathematical model of a neuron that will be used throughout most of this thesis. It is just a basic linear SDE together with an associated first-hitting time specification.

An introduction to mathematical models in neuroscience is given in Gerstner et al., [15], also available online. The book Stochastic Methods in Neuroscience [16] provides a nice overview of several current research applications of stochastic techniques to neuroscience.

A neuron's most important property is its membrane potential - the electric potential difference between the cell's interior and its surroundings. Neurons relay information by means of voltage spikes - sudden sharp increases in their membrane potential, which are then transmitted to other neurons that are chemically or elec-

trically connected to the spiking neuron. The information content is thought to be contained in the timing of the spike, or equivalently in the length of the time-interval between subsequent spikes. In the simplest case, this can be thought of as a *firing rate*, i.e. the average number of spikes over some time interval, but more complicated coding schemes have been hypothesized to exist. A spike in a given neuron can be triggered as a result of spikes coming from neurons connected to it. Special neurons called sensory neurons interface with external physical stimuli such as mechanical vibrations in the case of auditory neurons and light photons in the case of visual neurons in the retina.

Given a small positive stimulus, the voltage of a neuron increases linearly, but then relaxes back down to its equilibrium, pre-stimulus, level. However, if the stimulus is sufficiently strong or if there are many small stimuli in a sufficiently short time interval, the voltage will then go through a stereotypical large non-linear excursion before resetting back down to its equilibrium - this is a spike, which can propagate along the axon to other neurons. The physical underpinnings of this event involves ion channels with different time-scales, with an initial fast excitation due to sodium ions (known as a depolarization) followed by a slightly slower counter-acting inhibition by potassium ions (repolarization). The full spike unfolds on the order of 1-5 milliseconds.

The first successful mathematical model describing this phenomena is the famous Hodgkin-Huxley (HH) model, which forms a system of 4 ODEs, one for the membrane voltage and three for the dynamics of the ion channels. From a practical point of view however, the HH model of neuron dynamics is quite complicated if one is only interested in spike times, rather than the detailed contributions of ion channels to the voltage. Thus there have been several reductions to this model. The standard approach is to keep the voltage equation, as well as one more equation as a 'recovery' variable which works on a slower scale than the voltage; together these two variables produce the very sharp up-and-down voltage excursions. Two popular examples of

such reductions are the Fitzhugh-Nagumo model and the Morris-Lecar model. The Morris-Lecar model is a 2-dimensional ODE for, respectively, the voltage and recovery variables $v(t), w(t)$. It reads:

$$\begin{cases} \dot{v}(t) &= \frac{1}{C}\Big(- g_{Ca} m_\infty(v)(v - V_{C_a}) - g_K w(v - V_K) \\ & \quad -g_L(v - V_L) + I(t)\Big) \\ \dot{w}(t) &= \alpha(v)(1 - w) - \beta(w)w \end{cases} \tag{2.2.1}$$

where the auxiliary channel gating functions, m_∞, α, β are given by:

$$\begin{aligned} m_\infty(v) &= \frac{1}{2}\left(1 + \tanh\left(\frac{v - V_1}{V_2}\right)\right), \\ \alpha(v) &= \frac{1}{2}\phi\cosh\left(\frac{v - V_3}{2V_4}\right)\left(1 + \tanh\left(\frac{v - V_3}{V_4}\right)\right), \\ \beta(v) &= \frac{1}{2}\phi\cosh\left(\frac{v - V_3}{2V_4}\right)\left(1 - \tanh\left(\frac{v - V_3}{V_4}\right)\right). \end{aligned}$$

The term $I(t)$ represents current applied on the neuron, either from natural external stimulation, an experimental control, or synaptic or electrical inputs from other neurons. In section 4.6.3, we describe how the Morris-Lecar deterministic ODE of eq. (2.2.1) is extended to an SDE, and we demonstrate how its spikes times can be optimally controlled in spite of the noise.

The 2-dimensional models like Morris-Lecar are much easier to work with mathematically and experimentally, for example for parameter estimation, but they still require a lot of effort for describing the details of a neural spike. If one is only interested in the timing of the spike then yet another simplification is to keep only the voltage dynamics, and then declare that a spike has occurred whenever the voltage crosses some appropriate threshold, v_{thresh}, which can be related to the sodium activation threshold in real neurons. For analytical purposes this turns out to be both convenient and satisfactory. The basic integrate-and-fire model then is just

$$\begin{aligned} \dot{v} &= \left(I(t) - \frac{(v - \mu)}{\tau}\right) \mathrm{d}t \\ v(T_{\mathrm{sp}}) = v_{thresh} \implies & \Big\{ v({T_{\mathrm{sp}}}^+) = 0 \end{aligned} \tag{2.2.2}$$

In many areas of the brain, due to the large number of connections, the external stimulation to a neuron is highly erratic and unpredictable. When the randomness is mostly in this input, I, one can approximate its effect by adding to the input I a Gaussian white noise. In the terminology of this thesis, this is just a term proportional to a Wiener process increment, βdW_t. Thus the final model of the basic spike-generation mechanism that accounts for its most important aspects yet retains analytical tractability is the noisy leaky-integrate-and-fire model:

$$\begin{aligned} dX_t &= \left(\alpha(t) - \frac{(X_t - \mu)}{\tau}\right) \mathrm{d}t + \beta \, \mathrm{d}W_t, \\ X(0) &= 0, \\ X(T_{\mathrm{sp}}) = x_{th} \implies &\left\{ X(T_{\mathrm{sp}}{}^{+}) = 0 \right. \end{aligned} \tag{2.2.3}$$

That is, X_t follows an Ornstein-Uhlenbeck (OU) process, but upon reaching a predetermined threshold, x_{th}, a 'spike' is deemed to have occurred and the process is 'reset' to an initial value, here 0.

Alternatives to the hard-threshold integrate-and-fire model are so-called 'soft-threshold' models which use a *hazard* function, which is akin to the intensity of a Poisson Process to determine the spike time. The hazard function increases as the voltage increases, thus higher voltages imply higher likelihood of spiking. The hard-threshold model can be seen as a special case of the soft-threshold model, with the hazard function equal to 0 below the threshold and infinity above the threshold. We will not address hazard-function-based spiking models in the rest of the thesis.

2.3 Parameter Estimation for SDEs

Let us rewrite the SDE in eq. (2.1.4) so that the functions B, G are explicitly parametrized by some parameter set, θ;

$$dX_t = B(X, t; \theta)dt + G(X, t; \theta)dW, \tag{2.3.1}$$

For example, in the case of the OU process, eq. (2.1.3), $B(X,t;\theta) = (\mu - X_t)/\tau$ and $G(X,t;\theta) = \beta$, and the parameter set is $\theta = \{\mu, \tau, \beta\}$. In the standard formulation of the SDE estimation problem, one has exact observations $\{x_n\}_{n=0}^N$ at times $\{t_n\}_{n=0}^N$ from a process X_t satisfying eq. (2.3.1), and one seeks to find the values of the parameter set θ.

2.3.1 Maximum Likelihood Estimation

A fundamental method, both practically and theoretically, for estimating parameters in an SDE is the *Maximum Likelihoood* (ML) method, which proceeds by seeking those parameters which maximize the likelihood of the observed data, $\{x_n\}_{n=0}^N$. In particular let

$$L(\{x_n\};\theta) = \mathbb{P}[X_0 = x_0 \dots X_n = x_n \dots X_N = x_N|\theta]$$

be the joint probability of observing the data $\{x_n\}$ given the parameter set θ, also known as the likelihood. Then the ML method seeks to maximize L.

In the case of independent observations, the likelihood is just the product of the individual probabilities of each observation. In the case of SDEs, the problem is only slightly more complicated, due to the Markov nature of the stochastic process. In particular, the likelihood becomes the product of the transition probabilities. Recall that earlier we defined the transition probability $f(x,t|y,s)$ in eq. (2.1.5). We shall also write this as $f_\theta(x,t|y,s)$ if we need to be reminded of f's dependence on the parameter set, θ. The likelihood of the observed x_n then becomes

$$L(\{x_n\};\theta) = \prod_{n=1}^{N} f_\theta(x_n, t_n|x_{n-1}, t_{n-1}) \tag{2.3.2}$$

Here, we assume that x_0 is fixed, otherwise we would have to add a term specifying the probability distribution of X_0.

In general the transition density for a generic SDE is impossible to find analytically. There are several ways to approximate it numerically. The most generic

way relies on the numeric solution of the Fokker-Planck PDE in eq. (2.1.6), but that is computationally quite expensive and suffers from the 'curse-of-dimensionality' for SDEs of higher dimension.

In some simple cases, the transition density *can* be calculated. The OU example described above is one such case, where the transition density is

$$\begin{aligned} f(x_n, t_n|x_{n-1}, t_{n-1}) &= f(x_n, \Delta|x_{n-1}, 0) \\ &= \frac{1}{\beta\sqrt{\tau 2\pi(1-e^{-2\Delta/\tau})}} \cdot \exp\left(\frac{\left((x-\mu)-(x_0-\mu)\cdot e^{-\Delta/\tau}\right)^2}{\tau\sigma^2(1-e^{-2\Delta/\tau})}\right) \end{aligned}$$

assuming that $\Delta_n = t_n - t_{n-1} = \Delta$ is constant for all n. With this expression, one can form the likelihood and solve analytically for the maximizers $\{\hat{\mu}, \hat{\tau}, \hat{\beta}\}$:

$$\hat{\mu} = \frac{\sum_{n=1}^{N}\left(X_n - e^{-\frac{\Delta}{\hat{\tau}}}X_{n-1}\right)}{N(1-e^{-\frac{\Delta}{\hat{\tau}}})} \tag{2.3.3}$$

$$e^{-\frac{\Delta}{\hat{\tau}}} = \frac{\sum_{n=1}^{N}(X_n-\hat{\mu})(X_{n-1}-\hat{\mu})}{\sum_{n=1}^{N}(X_{n-1}-\hat{\mu})^2} \tag{2.3.4}$$

$$\hat{\beta}^2 = \frac{2\sum_{n=1}^{N}\left(X_n - \hat{\mu} - (X_{n-1}-\hat{\mu})e^{-\frac{\Delta}{\hat{\tau}}}\right)^2}{N(1-e^{-2\frac{\Delta}{\hat{\tau}}})\hat{\tau}} \tag{2.3.5}$$

The solution for the ML estimates is almost explicit. It requires one numerical single-dimensional root-finding, which is an easy numerical task. Note that there is no guarantee that the estimate for $\hat{\beta}$ will be positive in general.

2.3.2 Fortet Equation

In the context of eq. (2.2.3), that is for a 1-dimensional SDE with a boundary above the initial value of the process, $x_{th} > x_0$, there is another relation between the hitting-time density, g and the unconditional transition density, f, that is f for which no boundary conditions are applied at x_{th}.

Consider the space integral of f

$$\Phi(x, t|x_0, t_0) = \int_{\xi<x} f(\xi, t|x_0, t_0))\,\mathrm{d}\xi,$$

which is just the probability that $X_t \leq x$, conditional on the initial state, x_0. The *Fortet equation* [17] states that

$$1 - \Phi(x_{th}, t|0, 0) = \int_0^t g(s)[1 - \Phi(x_{th}, t|x_{th}, s)]\,\mathrm{d}s. \tag{2.3.6}$$

The left hand side is simply the probability of exceeding x_{th} at time t starting at 0 at time 0. This can also be written as the probability of hitting x_{th} for the first time at time $s < t$ and then exceeding x_{th} at time t starting at x_{th} at time s, integrated over all s.

The Fortet equation is particularly appealing to use in parameter estimation contexts when we have an analytical expression for the unconditional Φ as is the case for example for the Ornstein-Uhlenbeck process. We will see in chapter 3 that eq. (2.3.6) can be extended to the case when the threshold boundary or the dynamics of X_t are not time-homogeneous.

2.3.3 Particle Filtering

In *Bayesian* approaches to parameter estimation, the unknown parameter, θ is also treated as a Random Variable, Θ, with some belief distribution:

$$\rho(\theta) = \mathbb{P}(\Theta = \theta) \tag{2.3.7}$$

which is also called a prior. Via applications of the Bayes formula, observations from the system are used to update the belief distribution.

Let us say that the Random Variable T depends on the Random Variable, Θ, which we denote by writing the probability density of T as

$$g(t|\theta) = \Pr(T = t|\Theta = \theta). \tag{2.3.8}$$

Suppose we observe $T = t$. Then Bayes' formula states that

$$\rho(\theta|t) = \frac{\rho(\theta) \cdot g(t|\theta; \alpha)}{\int_\Theta \rho(\theta) \cdot g(t|\theta; \alpha)\,\mathrm{d}\theta} \tag{2.3.9}$$

In practice, exact calculation of $\rho(\theta|t)$ would not be possible in our context, so an approximation approach needs to be made.

A standard technique is to approximate ρ by a set of weighted particles. To avoid repetition, we refer to a quick description of how this is done in chapter 5, section 5.7.1, where it is used.

2.3.4 Optimal Design

There are estimation problems in which the experimenter has some control over some of the parameters in the model and may choose to set them in order to facilitate the estimation task. *Optimal design* is the design approach for statistical experiments that is guided by optimizing some formal measure of the parameter estimates of the experiments, [18]. For example, perhaps one would like to perform linear regression with a polynomial model and one has some latitude over the points at which to evaluate the polynomial. Common criteria when selecting the design, e.g the polynomial points, are minimizing the determinant or the trace of the covariance matrix of the parameter estimators. However, the covariance matrix, directly related to the so-called Fisher Information, often depends on the very parameters one seeks to estimate.

An alternative to the Fisher Information as an objective for the formal design of experiments is to use concepts from Information Theory, [19]. A related topic in the Machine Learning literature is called 'Active Learning', e.g. see [20, 21, 22]. In the third part of the thesis, chapter 5, we will use the *Mutual Information* criteria as a guideline for choosing the stimulation that best allows parameter estimation. Thus we now define and discuss the Mutual Information between two random variables X, Θ. For reference, we follow [19].

Definition 2.3.1. *Mutual Information: Given two random variables T, Θ with joint probability density $p(x, \theta)$ and marginal densities $p(x), p(\theta)$, the Mutual Information*

between T and Θ is given by

$$I(T,\Theta) = \int_{\Theta}\int_{T} p(x,\theta)\cdot\log\left(\frac{p(x,\theta)}{p(x)p(\theta)}\right)\,\mathrm{d}x\,\mathrm{d}\theta \tag{2.3.10}$$

It is obvious that if T,Θ are independent, then $I(T,\Theta)=0$. It can be verified that $I\geq 0$ and that it is maximized if Θ is a function of T; that is, if the entropy of Θ conditional on T is zero. The mutual information, $I(T,\Theta)$ represents the 'average reduction in uncertainty about Θ that results from learning the value of T' [19]. This statement is formally correct if one takes 'uncertainty' to mean the entropy of a random variable.

In order to make use of the Mutual Information in a parameter estimation context, we again recall the prior and likelihood, $\rho(\theta), g(t)$ as defined in eqs. (2.3.7) and (2.3.8).

If we consider Θ as a parameter and T as an observation, it is then natural to seek an experiment which maximizes the Mutual Information between Θ and the observation T.

The joint distribution can be written as $p(t,\theta)=g(t|\theta)\rho(\theta)$; the marginal distribution of Θ is simply the prior, $p(\theta)=\rho(\theta)$; and the marginal of T is $p(t)=\int_{\Theta} g(t|\theta)\rho(\theta)\,\mathrm{d}\theta$. Plugging the three expressions into the definition in eq. (2.3.10) and cancelling common terms yields

$$I(T,\Theta) = \int_{\Theta}\int_{0}^{\infty} g(t|\theta)\log\left(\frac{g(t|\theta)}{\rho(\theta)\int_{\Theta} g(t|\theta)\rho(\theta)\,\mathrm{d}\theta}\right)\,\mathrm{d}t\,\mathrm{d}\theta. \tag{2.3.11}$$

Equation (2.3.11) is used in chapter 5.

2.4 Stochastic Optimal Control

Optimal Control Theory has three main components - a state, x, a control, α, and an objective J which is a functional of $\{x,\alpha\}$ and which we try to either minimize or maximize. In *Stochastic* Optimal Control, x follows a stochastic process, thus the

objective, J is most often expressed in terms of some average or expectation over the random realizations of x. The general theory of Optimal Control, as well as the subset dealing explicitly with random systems, has relied on two main analytical techniques - *the Maximum Principle* and *Dynamic Programming*, (a classic reference is [10]). The Maximum Principle uses a variational approach to characterize the optimal pair, x_{opt}, α_{opt} optimizing J, while Dynamic Programming recursively builds up the optimal solution with a backwards induction from the terminal conditions. Both techniques have their advantages and disadvantages and we use both in the thesis. It turns out that there is a close relation between the two approaches, which is well known in the deterministic finite-dimensional deterministic case, [10, 12] and less so in the stochastic case, [23].

To set the notation right, we consider the following functional

$$J[\alpha] = \mathbb{E}_X \left[\int_0^{t_f} L(X_t, \alpha_t)\, \mathrm{d}t + M(X_{t_f}) \right] \tag{2.4.1}$$

over the realizations of X_t governed by an Itô SDE as in eq. (2.1.4), such that the drift B and/or the diffusion coefficient, G are parametrized by the control $\alpha(t)$. Here L is the running cost function, which depends on the trajectory and the applied control, while M is the terminal cost function, which just depends on the value of the trajectory at the terminal time, t_f. Here we assume that the terminal time is a priori known, although this is not necessary in general.

Given J in eq. (2.4.1), we then seek α, which maximizes it (for example):

$$\alpha^* = \arg\max_{\alpha \in \mathcal{U}} [J[\alpha]] \tag{2.4.2}$$

To be mathematically correct, we should specify that the optimization in eq. (2.4.2) is done over the space $\mathcal{U}$ of stochastic processes that are measurable with respect to the filtration generated by the underlying Wiener process, W_t. Further constraints on $\mathcal{U}$ may be imposed by a specific problem.

The simple-looking eq. (2.4.1) can give rise to different variations. For one, the final-time t_f may be variable. Or we may face path or terminal constraints for X_t. In the stochastic context such constraints are only enforceable in a probabilistic sense and they can easily make the problem much more difficult. However we will not deal with either of these complications and in the sequel we will assume that t_f is fixed and that X_t faces no constraints other than that it satisfies its SDE.

Given the objective and the optimization equations, eqs. (2.4.1) and (2.4.2), the Maximum Principle relies on the forward Kolmogorov (Fokker-Planck) equation-based definition of the SDE expectation, while Dynamic Programming relies on the relation between an SDE expectation and the backward Kolmogorov equation.

2.4.1 The Maximum Principle for Transition Probability Densities

The Maximum Principle uses a variational principle to characterize the optimal control, α and the optimal *transition* density. Note that the expectation in the objective, eq. (2.4.1) can be written in terms of the forward probability density of the state X_t:

$$\begin{aligned} J[\alpha] &= \mathbb{E}_X \left[\int_0^{t_f} L(X_s, \alpha_s)\, \mathrm{d}s + M(X_{t_f}) \right] \\ &= \int_0^{t_f} \int_X L(x, \alpha_s) \cdot f(x, s|x_0, 0)\, \mathrm{d}s\, \mathrm{d}x + \int_X M(x) \cdot f(x, t_f|x_0, 0)\, \mathrm{d}x. \end{aligned} \quad (2.4.3)$$

As such, the stochastic problem is reduced to a deterministic optimization problem but for PDEs, given that f follows the forward PDE in eq. (2.1.6).

Thus, for the purpose of this thesis, the Maximum Principle for Stochastic Optimal Control is really a Maximum Principle for PDEs. Its variational argument is similar in spirit to the Euler-Lagrange equations from the Calculus of Variations; one can think of it as a generalization of the zero-tangent rule (Fermat's Rule) for finding optima in single-variable calculus. In fact, like both the Euler-Lagrange equations

and Fermat's Rule, the Maximum Principle provides necessary, but not sufficient conditions for an optimum.

Originally, the Maximum Principle was developed for finite-dimensional deterministic systems, i.e. systems described by ODEs. In that context it is known as the *Pontryagin* Maximum Principle and for ODEs, the theoretical results of existence and uniqueness of optimal controls are strongest. As is often the case, theoretical results in the infinite-dimensional context, i.e. for PDEs are more difficult to obtain, but the general technique carries over analogously. Recently, there has been a series of publications on PDE control of the Fokker-Planck equation, see [24, 25, 23]. In particular, [23] discusses the relation between the Dynamic Programming approach to Stochastic Control and the approach based on PDE optimization of the Fokker-Planck equation.

The basic idea of the Maximum Principle is that one 'adjoins' the dynamics' PDE to the objective and introduces a Lagrange multiplier, which in this case is called *the adjoint state*.

Let us rewrite the governing SDE, eq. (2.1.4), to explicitly take into account the control variable $\alpha(t)$:

$$dX_t = B(X, t; \alpha)dt + G(X, t)dW. \tag{2.4.4}$$

Given the problems considered in this thesis, we only illustrate a one-dimensional SDE. Its corresponding forward density is governed by the following Fokker-Planck equation:

$$\partial_t f = -\partial_x \left[B(x, t; \alpha) f(x, t)\right] + \partial_x^2 \left[D(x, t) f(x, t)\right], \tag{2.4.5}$$

where $D(x, t) = G^2(x, t)/2$. We assume that for all $\alpha(t)$, both the SDE and the PDE have unique solutions.

For notational convenience, we will write eq. (2.4.5) as

$$\dot{f} = \mathcal{L}_\alpha[f]$$

where $\mathcal{L}_\alpha$ is the differential operator corresponding to the right-hand side of the

Fokker-Planck equation parametrized by the control α. For now we will assume that there are no BCs, and the domain of X is all of $\mathbb{R}$.

As we already alluded to, the key concept in Maximum Principle for PDEs is to adjoin the dynamics, eq. (2.4.5), multiplied by the adjoint state, p to the objective, eq. (2.4.3)

$$\begin{aligned} J[\alpha] = &\int_0^{t_f} \int_X L(x, \alpha_s) \cdot f(x, s) \, \mathrm{d}s \, \mathrm{d}x + \int_X M(x) \cdot f(x, t_f) \, \mathrm{d}x \\ &- \int_0^{t_f} \int_X p \cdot (\partial_t f(x, s) - \mathcal{L}_\alpha[f(x, s)]) \, \mathrm{d}s \, \mathrm{d}x. \end{aligned}$$

Since f satisfies the PDE, we have added a term that equals zero. However, what this allows us to do is to 'transfer' the time and space derivatives from f to p. The reason why that is productive is that we will then be able to form the 'variation' of J with respect to the control α and either set it to zero or use this as a gradient.

To illustrate the idea, let us perform this 'transfer' explicitly - it is basically an application of integration-by-parts; in larger dimension this is also commonly called *Green's identities*. We have that

$$\int_0^{t_f} p \cdot \partial_t f \, \mathrm{d}t = p \cdot f|_0^{t_f} - \int_0^{t_f} \partial_t p \cdot f \, \mathrm{d}t$$

which 'transfers' the time-derivative to p. We also have that

$$\int_X p \cdot \mathcal{L}_\alpha[f] \, \mathrm{d}x = \int_X (B \partial_x p + D \partial_x^2 p) \cdot f \, \mathrm{d}x = \int_X {\mathcal{L}^*}_\alpha[p] \cdot f \, \mathrm{d}x$$

which transfers' the space-derivative to p. In doing so, we have naturally introduced the differential operator $\mathcal{L}^*$, which is the adjoint, in a Banach-space sense, to $\mathcal{L}$. Actually, we have already met $\mathcal{L}^*$ before - it is the generator of the SDE in eq. (2.4.4).

In the above manipulations, we have assumed that the double integrals have the same value independent of the order of integration of the space and time variables and that f and all its partials go to zero uniformly for $|x|$ large enough. If we had boundary conditions on f, those will come up in the spatial terms.

With the above integration-by-parts done, we can write the objective as

$$J[\alpha] = \int_0^{t_f} \int_X L(x_s, \alpha_s) \cdot f(x, s)\, \mathrm{d}s\, \mathrm{d}x + \int_X M(x) \cdot f(x, t_f)\, \mathrm{d}x - \int_X \left[p \cdot f|_0^{t_f} + \int_0^{t_f} (\partial_t p + \mathcal{L}^*{}_\alpha[p]) \cdot f\, \mathrm{d}s \right] \mathrm{d}x \tag{2.4.6}$$

Now we assume that we apply a small variation around a given control α:

$$\alpha_\epsilon = \alpha + \epsilon\delta\alpha$$

$$f_\epsilon = f + \epsilon\delta f$$

The variation of the objective, J, with respect to the control variation at the current α can now be calculated as:

$$\begin{aligned} \frac{dJ}{d\epsilon}\Big|_{\epsilon=0} = &\int_0^{t_f} \int_X \nabla_\alpha L(x_s, \alpha_s) \cdot \delta\alpha \cdot f(x, s) + L(x_s, \alpha_s) \cdot \delta f(x, s)\, \mathrm{d}s\, \mathrm{d}x \\ &+ \int_X M(x) \cdot \delta f(x, t_f)\, \mathrm{d}x \\ &- \int_X p \cdot \delta f(x, t_f)\, \mathrm{d}x \\ &+ \int_X \int_0^{t_f} (\partial_t p + \mathcal{L}^*{}_\alpha[p]) \cdot \delta f + \nabla_\alpha B(x, t; \alpha) \cdot \delta\alpha \cdot \partial_x p \cdot f\, \mathrm{d}s\, \mathrm{d}x \end{aligned} \tag{2.4.7}$$

A few notes are required in order to better explain eq. (2.4.7). The initial conditions are considered fixed and as such $\delta f|_{t=0} \equiv 0$, that is the variation in the control does not change $f(x, 0)$. This is why only the term $p \cdot f|_{t=t_f}$ is retained from eq. (2.4.6).

Heuristically, we would like to infer from eq. (2.4.7) a gradient with respect to the control, however we note that there are also variations with respect to f that make it impossible to do so. However, recall that p is our free variable - the Lagrange multiplier. Thus if we choose p appropriately, we can eliminate δf from the expression. Since we have set up the problem with this in mind, this is now straight forward to do - we let p evolve (backwards) according to:

$$\begin{cases} -\partial_t p &= \mathcal{L}^*[p] + L \\ p(x, t_f) &= M(x) \end{cases} \tag{2.4.8}$$

Thus, eq. (2.4.7) simplifies to

$$\frac{dJ}{d\epsilon}\Big|_{\epsilon=0} = \int_0^{t_f} \int_X [\nabla_\alpha L(x_s, \alpha_s) \cdot f(x, s) + \nabla_\alpha B(x, t; \alpha) \cdot \partial_x p \cdot f] \cdot \delta\alpha \, \mathrm{d}s \, \mathrm{d}x \quad (2.4.9)$$

From eq. (2.4.9), we can infer that

$$\nabla_a J(x, t) = \nabla_\alpha L(x, \alpha_t) \cdot f(x, t) + \nabla_\alpha B(x, t; \alpha_t) \cdot \partial_x p(x, t) \cdot f(x, t) \quad (2.4.10)$$

can be considered as a pointwise gradient of the objective with respect to changes in the control, given the current control. It is then natural to claim that setting that equal to zero will give us a necessary condition for an optimal α. If setting eq. (2.4.10) to 0 and solving for α is possible, we would then have an expression of α in terms of the state, f and the adjoint p, which we could then input in their respective equations. Of course, this explicit representation of α would not always be possible for any running cost function L or drift fields B. Even if it were possible, we would still have to solve the pair of now-nonlinear forward/backward equations for f, p. It should be clear by now why the Maximum Principle theory for PDEs faces many practical challenges.

There are two practical approaches to obtain actual numerical results given eq. (2.4.10), both of them iterative: 1) gradient descent and 2) fixed point iteration.

The fixed point approach is advocated in [26], for example, but only for simple pedagogic examples and we will not discuss it here. Alternatively, we can apply a gradient-based optimization method. Since eq. (2.4.10) gives us a gradient, we can take the current control α and increment it in the direction of $\nabla_a J$, if we are maximizing J, or $-\nabla_a J$, if we are minimizing J.

As with all gradient-based optimization, the standard disclaimer about local minima and optimization initialization applies, since there is no guarantee - and it is usually not the case - that the objective is convex with respect to the control.

Our derivations have not been very rigorous. More rigorous arguments can be found in, e.g. [27, 28].

As a closing note, we should mention that a Maximum Principle in Stochastic Control can also be stated directly in terms of the SDE for X, in which case the corresponding 'adjoint' variable satisfies a *backwards* SDE. We do not explore this approach, but the literature on this topic is also vast, for a early introductory monograph see [29].

2.4.2 Dynamic Programming for Stochastic Optimal Control

Dynamic Programming uses backwards recursion to tabulate the optimal control starting from the terminal time. The basic object in dynamic programming is the value function, v. In order to introduce it, we first extend our definition for the objective, J, to consider starting the state, X_t at later times, with different initial conditions.

The running cost-to-go corresponding to eq. (2.4.1) is:

$$J[\alpha; x, t] = \mathbb{E}\left[\int_t^{t_f} L(x_s, \alpha_s)\,\mathrm{d}s + M(x_{t_f})\right], \quad X_t = x \tag{2.4.11}$$

so that $J[\alpha; x_0, 0] = J[\alpha]$ is our original objective from eq. (2.4.1).

We now introduce the *value function*, v, defined by:

$$v(x, t) = \inf_{\alpha \in \mathcal{U}} J[\alpha; x, t];$$

it is also called the optimal cost-to-go.

We immediately note the terminal conditions on v:

$$v(x, t_f) = M(x) \tag{2.4.12}$$

In other words, if $X_{t_f} = x$, there is no more time for a control to be applied and no more running cost to be incurred and we just incur the terminal cost corresponding to wherever X is now, i.e. $M(x)$. It turns out that the value function, v, can be characterized as the solution to the following non-linear PDE:

Theorem 2.4.1 (Hamilton-Jacobi-Bellman (HJB))**.**

$$\begin{cases} -\partial_t v(x,t) & = \max_{\alpha(t)\in\mathcal{U}(t)} \left\{ L(x,\alpha) + B(x,t,\alpha)\cdot\partial_x v \right\} + \frac{G^2(x,t)}{2}\partial_x^2 v \\ v(x,t_f) & = M(x) \end{cases} \tag{2.4.13}$$

Proof: (Heuristic Derivation adapted from [12])

Suppose we are at time t and $X_t = x$.Take a time increment $[t, t+h]$ and assume that during that time we apply a constant control α and subsequently we apply the optimal control α^*. The running-cost will then break down as:

$$J[\alpha; x, t] = \mathbb{E}\left[\int_t^{t+h} L(X_s, \alpha_s)\,\mathrm{d}s + v(X_{t+h}, t+h)\right]$$

Since $v(x,t) = \inf J(\alpha; x, t)$, we must have that:

$$v(x,t) \leq \mathbb{E}\left[\int_t^{t+h} L(X_s, \alpha_s)\,\mathrm{d}s + v(X_{t+h}, t+h)\right]$$

or, rearranging, that

$$0 \leq \mathbb{E}\left[\int_t^{t+h} L(X_s, \alpha_s)\,\mathrm{d}s\right] + \underbrace{\mathbb{E}[v(X_{t+h}, t+h) - v(x,t)]}_{\mathbb{E}[dv]}$$

Now, $\mathbb{E}[dv]$ can be expressed using Dynkin's Formula, as:

$$\mathbb{E}[dv] = \int_t^{t+h} \partial_t v + B\cdot\partial_x v + \frac{G^2}{2}\cdot\partial_x^2 v\,\mathrm{d}s$$

Plugging that back, we get

$$0 \leq \int_t^{t+h} L(X,\alpha) + \partial_t v + B\cdot\partial_x v + \frac{G^2}{2}\cdot\partial_x^2 v\,\mathrm{d}s$$

Taking $h\to 0$ we get

$$0 \leq L(x,\alpha) + \partial_t v(x,t) + B\cdot\partial_x v(x,t) + \frac{G^2}{2}\cdot\partial_x^2 v(x,t)$$

and we conjecture that for the actual optimal control, the inequality becomes an equality:

$$\begin{cases} 0 & = L(x,\alpha^*) + \partial_t v(x,t) + B(x,t,\alpha^*)\cdot\partial_x v(x,t) + \frac{G^2(x,t)}{2}\cdot\partial_x^2 v(x,t) \\ \alpha^* & = \arg\max_{\alpha\in\mathcal{U}(t)} \left\{ L(x,\alpha) + B(x,t,\alpha)\cdot\partial_x v \right\} \end{cases} \tag{2.4.14}$$

It turns out that rigorous proofs of Theorem 2.4.1 , in particular specifying the exact mathematical meaning for a solution, v, to the HJB PDE are quite complex and beyond the scope of this text. Some references include [30, 31]. We will assume that numerically solving eq. (2.4.13) is sufficient.

2.4.3 Numerical Solutions to PDEs - Finite Difference Methods

Both the variational approach of the Maximum Principle and the backwards induction of Dynamic Programming result in having to solve PDEs in order to obtain the optimal control. For practical purposes, solving these PDEs requires numerical discretization. In all cases the PDEs are *parabolic* (see [32]), which in general can be written as

$$\partial_t f(x,t) = \mathcal{L}[f(x,t)] \tag{2.4.15}$$

for some differential operator, $\mathcal{L}$. In one spatial dimension, a finite difference discretization of eq. (2.4.15) is to select a set of space- and time- nodes $\{x_i\}, \{t_k\}$ and approximate f by solving for $f(x_i, t_k)$, given the initial conditions $f(x_i, t_0)$ and possibly boundary conditions.

A classic technique for parabolic PDEs is the *Crank-Nicholson* scheme, which time-discretizes eq. (2.4.15) as

$$\frac{f(x_i, t_{k+1}) - f(x_i, t_k)}{t_{k+1} - t_k} = \frac{1}{2}\left(\mathcal{L}[f(x_i, t_{k+1})] + \mathcal{L}[f(x_i, t_k)]\right)$$

and then solves for the resulting linear system before stepping iteratively forward in time. See chapter 19 in [32] for a brief discussion of the theoretical properties of the Crank-Nicholson method.

This solution approach requires a finite spatial domain. If the spatial domain is theoretically infinite, as it may be in some cases, we need to truncate it and apply some reasonable boundary conditions at the artificial boundary which approximate the solution of the theoretically infinite space. We will show details of how this is done in the context of each specific problem discussed in the thesis.

Chapter 3

Estimation in the Sinusoidally-driven Leaky-Integrate and Fire Model

3.1 Thesis Context

In the first of three main chapters of the thesis, we focus on the problem of estimating model parameters in a sinusoidally-perturbed LIF model from the observation of spike timings only. We discuss two approaches - one based on solving an integral equation and one based on solving for the forward density. In both cases we need to make an approximation in order to account for the non-renewal property of the hitting-time process. Both estimation algorithms are iterative, therefore we propose a non-trivial method for initializing them with observation-driven initial guesses.

3.2 Problem Introduction

Information processing in the nervous system is carried out by spike timings in neurons. To study the neural code in such a complicated system, a first step is to understand signal processing and transmission in single neurons. Stochastic leaky integrate-and-fire (LIF) neuronal models are a good compromise between biophysical realism and mathematical tractability, and are commonly applied as theoretical tools to study properties of real neuronal systems. A central issue is then to perform statistical inference from experimental data and estimate model parameters. Many electrophysiological experiments on neurons, namely extra-cellular recordings, are only capable of detecting the time of the spike and not the detailed voltage trajectory leading up to the spike. Estimating the parameters of the LIF model from this type of data is equivalent to estimating the parameters of a stochastic model from the statistics of the first-passage times only. A common assumption is that the data are well described by a renewal process, thus basing the statistical inference on the interspike intervals (ISIs), assuming these are realizations of independent and identically distributed random variables. Since only partial information about the process is available, the statistical problem becomes more difficult, and no explicit expression for the likelihood is available.

Different methods have been proposed. In the seminal paper [33], a point process approach is proposed. The spike trains of a collection of neurons are represented as counting processes. Time is discretized and the point processes approximated by 0-1 time series. Then the probability of firing in the next time interval is modeled as a function of the spike history. In this way maximum likelihood estimation is feasible. External stimuli are not considered. In [34] a numerically involved moment method is developed. It uses the first two moments of the first-passage times of the Ornstein-Uhlenbeck process to a constant threshold, which are given as series expressions, and equates them to their empirical counterparts. In [35, 36] certain explicit moment

relations derived from the Laplace transform of the first-passage time distribution are applied, but these are only valid under stimulation (supra-threshold regime). In [37] inference is based on numerical inversion of the Laplace transform. In [38], a functional of a 3-dimensional Bessel bridge is applied to obtain a maximum likelihood estimator. None of these methods are feasible to extend to the non-timehomogenous case, which is of our interest. In [39, 40] an integral equation is used to derive an estimator in the time-homogenous setting. This approach is readily extended to time varying input, which we will explore in this chapter. Some of the above methods are compared in [41]. Finally, a review of estimation methods is provided in [42].

Many sensory stimuli, like sound, contain an oscillatory component [43, 44]. Such inputs will cause oscillating membrane potentials in the neuron, generating rhythmic spiking patterns. The oscillation frequency determines the basic rhythm of spiking, and is considered to be significant for neuronal information processing. The dynamics of periodically forced neuron models have been extensively studied, see [45, 46, 47, 48, 49, 50] and references therein. Even so, attempts to solve the estimation problem in these non-stationary settings have been rare. One problem is that the ISIs are no longer independent nor identically distributed. In [51] a more complicated model with linear filters is considered, allowing also for the spike history to influence the membrane potential dynamics. The estimation problem is solved through numerical solutions to the Fokker-Planck equation, and it is shown that the log-likelihood is concave, thus ensuring a global maximum, see also [52, 53]. Because their model is more involved, some approximations to the solution of the Fokker-Planck equation is applied, to ensure acceptable computing times. We will apply the full Fokker-Planck equation to solve our estimation problem, since the computing time is always lower than 2 seconds for a sample size of 1000 spikes.

In this chapter, we thus describe and discuss two methods to estimate parameters of LIF models with the added complexity of a time-varying input current. We assume that the time-varying current is a sinusoidal wave, but we believe that the

approaches generalize to an arbitrary periodic forcing with known frequency. One approach relies on the Fortet integral equation, which is readily extended to the time non-homogeneous case. An advantage of this approach is that if the transition density of the diffusion in the LIF model is known, as is the case for the Ornstein-Uhlenbeck and the Feller model, the computational burden is limited. A second approach involves numerical solution of the Fokker-Planck equation, where the time-dependence is explicitly accounted for. After a numerical differentiation, the likelihood function can be calculated providing the maximum likelihood estimator. Nevertheless, we chose an alternative loss function which seems marginally more robust, directly comparing the survival function provided by the solution of the Fokker-Planck equation with its empirical counterpart. The two approaches give similar results and they are more carefully compared in the supplementary online material.

Both methods need sensible starting values for the optimization algorithms, and we provide an easy-to-implement initializer. The estimation procedures are compared on simulated data and we find that both algorithms are able to find estimates close to the true values for several different dynamical regimes. We find that for small sample sizes the Fokker-Planck algorithm can be considered marginally preferable, whereas for larger sample sizes the Fortet algorithm becomes marginally superior. Moreover, at high frequencies of the sinusoidal forcing, the Fortet is better at identifying the parameters, though in general there is less information in the data to distinguish between a constant input and the amplitude of the periodic forcing.

3.3 Model

The time evolution of the voltage of a spiking neuron is modelled by a stochastic process, V, given as solution to the following stochastic differential equation (SDE)

$$\begin{gathered} \mathrm{d}V(t) = \left(\mu - \frac{V(t)}{\tau} + A\sin(\omega t)\right)\mathrm{d}t + \sigma\,\mathrm{d}W(t), \\ t_0 = 0; \quad V(t_0) = v_0, \\ t_n = \inf\{t > t_{n-1} : V(t) = v_{\mathrm{th}}\} \quad \text{for } n \geq 1 \\ \begin{cases} V(t_n^+) = v_0 \\ J_n = t_n - t_{n-1}. \end{cases} \end{gathered} \tag{3.3.1}$$

Here, μ is a bias current acting on the cell, τ is the decay time, A and ω are the amplitude and (angular) frequency of the sinusoidal current acting on the cell, σ is the strength of the stochastic fluctuations, $W = \{W_t\}_{t\geq 0}$ is a standard Wiener process, and t_n^+ denotes the right limit taken at t_n. A spike occurs when the membrane voltage $V(t)$ crosses a voltage threshold, v_{th}, and then $V(t)$ is instantaneously reset to the resting potential v_0. The difference between subsequent spike times, $J_n = t_n - t_{n-1}$, is called the interspike interval (ISI).

We will assume that τ is known (but see section 2.3.4 for a discussion of the alternative) and non-dimensionalize eq. (3.3.1) as follows

$$\begin{aligned} s &= \frac{t}{\tau} & X_s &= \frac{V(t) - v_0}{v_{\mathrm{th}} - v_0} & W_s &= \frac{W(t)}{\sqrt{\tau}} & x_{th} &= 1 \\ \alpha &= \frac{\mu\tau}{v_{\mathrm{th}} - v_0} & \beta &= \frac{\sigma\sqrt{\tau}}{v_{\mathrm{th}} - v_0} & \gamma &= \frac{A\tau}{v_{\mathrm{th}} - v_0} & \Omega &= \omega\tau \end{aligned}$$

to obtain

$$\begin{aligned} dX_s &= (\alpha - X_s + \gamma \sin(\Omega s))\,\mathrm{d}s + \beta\,\mathrm{d}W_s, \\ s_0 &= 0; \quad X_{s_0} = 0, \\ s_n &= \inf\{s > s_{n-1} : X_s = x_{th} = 1\} \quad \text{for } n \geq 1 \\ &\begin{cases} X_{s_n^+} = 0 \\ I_n = s_n - s_{n-1}, \end{cases} \end{aligned} \tag{3.3.2}$$

where we have defined $I_n = J_n/\tau$. We can also write the dynamics between two spike times s_n and s_{n+1} in terms of elapsed time since the last spike, $s' = s - s_n$, $s' < I_{n+1}$,

$$\begin{aligned} dX_{s'} &= (\alpha - X_{s'} + \gamma \sin(\Omega(s' + \phi_n))\,\mathrm{d}s' + \beta\,\mathrm{d}W_{s'}, \\ &\begin{cases} s' &= s - s_n \\ \phi_n &= s_n \mod \frac{2\pi}{\Omega} \end{cases} \end{aligned} \tag{3.3.3}$$

This form of the dynamics highlights that this is not a renewal process since different trajectories between spikes have different phase shifts $\phi_n = s_n$ modulo $2\pi/\Omega$. This will be important in the following discussion. The shape of the ISI distribution depends on the model parameters, and it is natural to divide the parameter space in different regimes characterized by their qualitative behaviour. Four distinct parameter regimes will be considered; supra-threshold, critical, sub-threshold and super-sinusoidal. To understand the reasoning behind the regime names, observe that in the absence of noise, $\beta = 0$, the deterministic model will produce spikes if and only if

$$\alpha + \frac{\gamma}{\sqrt{1 + \Omega^2}} > 1,$$

see the discussion in [46], which can be directly inferred from the solution in eq. (3.3.12) below. In both the supra-threshold and super-sinusoidal regimes, $\alpha + \gamma/\sqrt{1 + \Omega^2} > 1$. The difference between the two is that in the supra-threshold regime the constant bias current alone is sufficient for spikes to occur, also in absence of noise, that is, $\alpha > 1$. In the super-sinusoidal regime the sinusoidal current is necessary for spikes to occur

in absence of noise, that is, $\alpha + \gamma/\sqrt{1+\Omega^2} > 1$ and $\alpha \leq 1$. In the critical regime, the sum of the two terms is just barely enough to guarantee deterministic spiking, that is $\alpha + \gamma/\sqrt{1+\Omega^2} \approx 1$. Finally, in the sub-threshold regime, there would be no spikes without the noise, $\alpha + \gamma/\sqrt{1+\Omega^2} < 1$.

Table 4.1 tabulates examples of corresponding parameter values for each regime, while fig. 3.1 shows examples of individual voltage trajectories and their associated spike trains. Figures 3.2 and 3.3 illustrate how each regime behaves for selected ϕ's by plotting the survivor distribution, $\bar{G}_\phi(t)$, and the probability density, $g_\phi(t)$, both defined in eq. (3.3.4) below.

With regards to figs. 3.2 and 3.3, it is worth noting explicitly, that combinations of noise and sinusoidal forcing can cause firing patterns in which spikes are phase locked, but skip a certain number of cycles. This leads to multi-modal ISI densities. There are many different dynamical mechanisms that can yield such patterns, and the particular correlations between the ISIs will depend on the underlying voltage dynamics (which, in our case, we assume to be given by eq. (3.3.1)); in particular, it may be difficult to distinguish whether the dynamics are sub-threshold or supra-threshold, since both can show similar ISI densities, see [54].

Regime Name	α	β	γ
Supra-threshold	1.40	0.30	0.14
Super-sinusoidal	0.10	0.30	1.98
Critical	0.50	0.30	0.71
Sub-threshold	0.40	0.30	0.57

Table 3.1: Example of α, β, γ parameter values for the different regimes, given $\Omega = 1$.

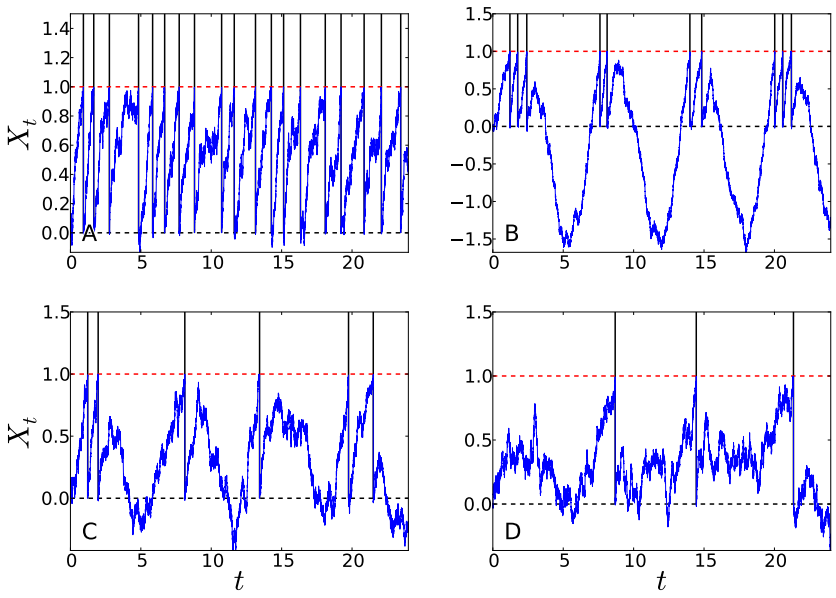


Figure 3.1: Example trajectories from eq. (3.3.2) for the four different parameter regimes using the parameter values given in table 4.1. A) supra-threshold, B) super-sinusoidal, C) critical, D) sub-threshold. In the supra-threshold regime spikes occur regularly and often; in the super-sinusoidal regime spikes cluster near the peak of the sine wave; in the critical regime they occur less often; and in the sub-threshold regime, spikes occur rarely. For all regimes, $\Omega = 1$.

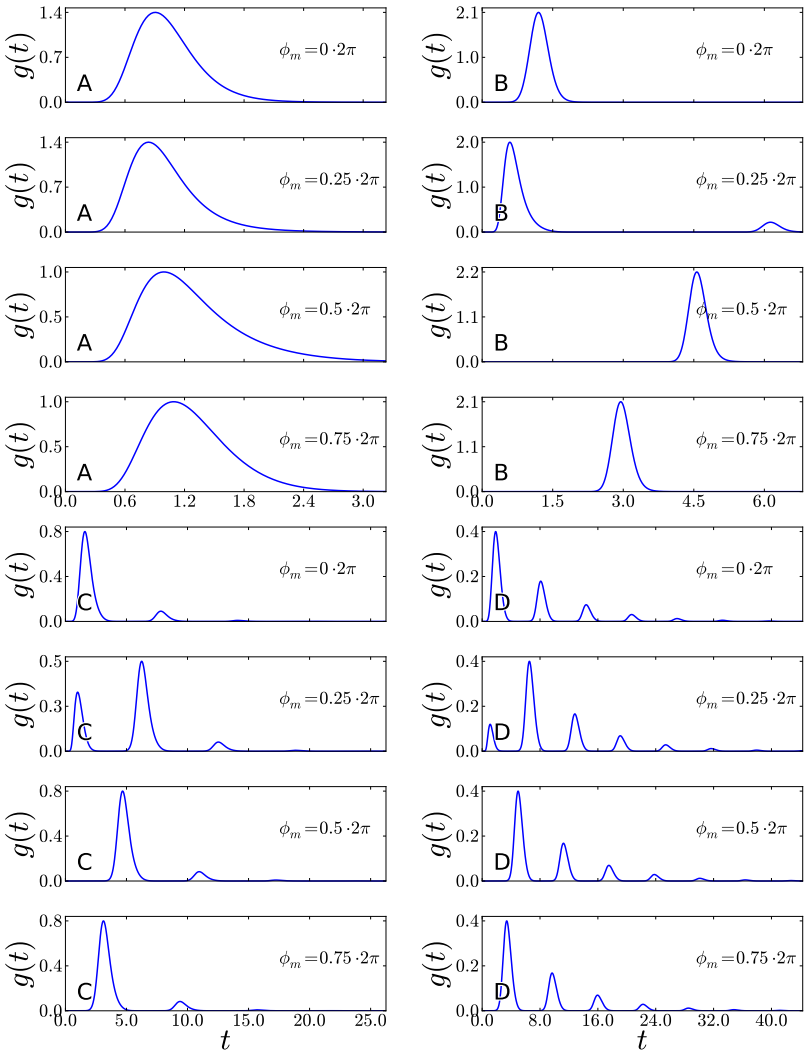


Figure 3.2: The four different parameter regimes using the parameter values given in table 4.1. Illustrated are the probability density functions, $g_{\phi_m}(t)$, for representative $\phi_m = 2\pi/\Omega \times \{0, 0.25, 0.5, 0.75\}$. Varying ϕ_m has, for the most part, the effect of shifting the curves laterally, while varying α, β, γ changes their characteristic form. For all regimes, $\Omega = 1$. A) supra-threshold, B) super-sinusoidal, C) critical, D) sub-threshold

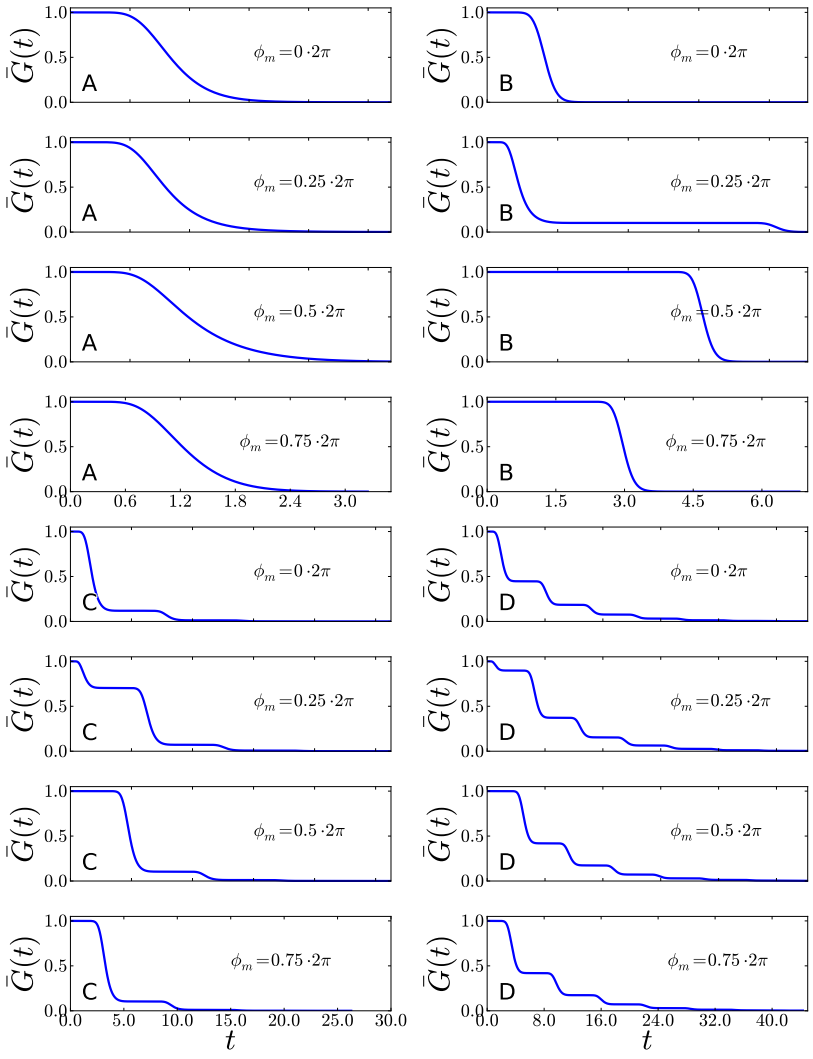


Figure 3.3: The four different parameter regimes using the parameter values given in table 4.1. Illustrated are the survivor distribution functions, $\bar{G}_{\phi_m}(t)$, for representative $\phi_m = 2\pi/\Omega \times \{0, 0.25, 0.5, 0.75\}$. Varying ϕ_m has, for the most part, the effect of shifting the curves laterally, while varying α, β, γ changes their characteristic form. A) supra-threshold, B) super-sinusoidal, C) critical, D) sub-threshold.

3.3.1 Basic ISI probability density functions

Here we introduce the notation for the probability density, distribution and survival functions of I_n, an ISI arising from a trajectory produced by eq. (3.3.3),

$$\begin{aligned} g_\phi(\tau)\,\mathrm{d}\tau &:= \mathbb{P}(I_{n+1} \in [\tau, \tau + \mathrm{d}\tau) | \phi_n = \phi) && \text{(probability density)} \\ G_\phi(t) &:= \mathbb{P}(I_{n+1} \leq t | \phi_n = \phi) = \textstyle\int_0^t g_\phi(\tau)\,\mathrm{d}\tau && \text{(cumulative distribution)} \\ \bar{G}_\phi(t) &:= \mathbb{P}(I_{n+1} > t | \phi_n = \phi) = 1 - G_\phi(t) && \text{(survivor distribution)} \end{aligned} \tag{3.3.4}$$

The subscript ϕ is to stress that g, G and $\bar{G}$ depend on the value of ϕ_n in eq. (3.3.3). This is the formal statement that in a sinusoidally-driven neuron, the interspike intervals are not identically distributed, and are only independent conditioned on the sinusoidal phase at an interval's onset. Knowing these distributions would provide the likelihood function, offering estimation by the preferred method of choice, the maximum likelihood estimator. Unfortunately, explicit expressions for the ISI distribution are not available except for the special case of $\gamma = 0$ and $\alpha = 1$, see [35]. Different representations of the likelihood function are available though, see [55], one of which we will use below.

3.3.2 Fokker-Planck Equation with Absorbing Boundaries

The Fokker-Planck equation is a partial differential equation (PDE) describing the evolution of the probability density, $f(x,t)$, of X_t. For the sinusoidally-forced Ornstein-Uhlenbeck process, eq. (3.3.3), with the threshold $x_{th} = 1$, the PDE is

$$\partial_t f^{(\phi)}(x,t) = -\partial_x[(\alpha - x + \gamma \sin(\Omega(t+\phi)) \cdot f^{(\phi)}] + \partial_x^2[\tfrac{\beta^2}{2} f^{(\phi)}], \quad x \in (-\infty, 1). \tag{3.3.5}$$

Due to the reset, we have that at time $t = 0$, $X_t = 0$ and so for the initial conditions we can write

$$f^{(\phi)}(x, t = 0) = \delta(x), \tag{3.3.6}$$

where $\delta(\cdot)$ is the Dirac delta function. The spike is represented as a zero boundary condition for f at $x = 1$

$$f(1,t) = 0.$$

The natural way of using the Fokker-Planck equation in first-hitting-times problems is as follows. Denote the integral of $f^{(\phi)}$ by $F^{(\phi)}(x,t) = \int_{\xi \le x} f^{(\phi)}(\xi,t)\,\mathrm{d}\xi$. $F^{(\phi)}(x,t)$ can be related to the ISI's survivor distribution function, $\bar{G}_\phi(t)$, by

$$\bar{G}_\phi(t) = F^{(\phi)}(1,t). \tag{3.3.7}$$

Equation (3.3.7) forms the basis of one of the methods below for estimating the structural parameters from the observed data.

Since eq. (3.3.5) has to be solved numerically, we will need to truncate its domain from below. The most natural way to do this, given the dynamics, is to impose reflecting boundary conditions at some $x = x_- \ll (\alpha - \gamma/\sqrt{1+\Omega^2})$ where the probability mass is very small. For the left (lower) limit of the computational domain, we use the formula

$$x_- = \min(\underbrace{\alpha - \gamma/\sqrt{1+\Omega^2}}_{\text{mean}} - 2\underbrace{\beta/\sqrt{2}}_{\text{std. dev}}, -0.25).$$

This choice requires some explanation. In the $t \to \infty$ limit, the distribution of X_t in eq. (3.3.3) *without* thresholding is Gaussian with mean given by eq. (3.3.12) (below) and variance equal to $\beta^2/2$. Thus to truncate the computational domain for the thresholded process from below, we take the lowest value of the asymptotic mean, $\alpha - \gamma/\sqrt{1+\Omega^2}$, then from this we subtract two standard deviations, $2\beta/\sqrt{2}$ and set the result to be the lower bound, x_-. Finally, if this value for x_- happens to be larger than $-.25$, we enforce that $x_- \le -0.25$.

Numerical considerations lead us to solve for F, instead of f, since delta functions are difficult to represent in floating point, while the initial conditions for F, the Heaviside step function, $H(x)$, faces no such difficulties [56]. The Heaviside step function is defined to be equal to 0 for $x < 0$ and to be equal to 1 for $x \ge 0$. At this

point we need to derive the PDE for the distribution F, starting from the PDE for the density, f, eq. (3.3.5).

First, at the lower boundary, it is intuitive that the distribution should be zero, $F(x_-, t) = 0$, while $f(1, t) = 0$ implies that at the upper boundary $\partial_x F(1, t) = 0$. Inside the domain, the PDE itself reformulates as

$$\partial_t f(x,t) \;=\; \partial_x \left[\frac{1}{2} \partial_x [\beta^2 f] - (\alpha - x + \gamma \sin(\Omega(t-\phi)) f \right]$$

so that

$$\partial_x \partial_t F(x,t) \;=\; \partial_x \left[\frac{\beta^2}{2} \cdot \partial_x^2 F - (\alpha - x + \gamma \sin(\Omega(t+\phi)) \cdot \partial_x F \right].$$

Integrating with respect to x then gives

$$\partial_t F(x,t) = \frac{\beta^2}{2} \cdot \partial_x^2 F - (\alpha - x + \gamma \sin(\Omega(t+\phi)) \cdot \partial_x F + C(t)$$

where $C(t)$ is a constant of integration depending on t. Now consider the lower boundary condition, $x = x_-$. Here $F(x_-, t) = 0$ implies that $\partial_t F = 0$ and so

$$C(t) = -\left[\frac{\beta^2}{2} \cdot \partial_x^2 F - (\alpha - x + \gamma \sin(\Omega(t+\phi)) \cdot \partial_x F \right]. \tag{3.3.8}$$

The right-hand side in eq. (3.3.8) is precisely the reflecting boundary condition on f once we recall that $\partial_x F = f$. Therefore $C(t) \equiv 0$.

Thus, the fully specified PDE for F, which we will be solving frequently in what follows, is

$$\partial_t F^{(\phi)}(x,t) = \frac{\beta^2}{2} \cdot \partial_x^2 F^{(\phi)} - \Big(\alpha - x + \gamma \sin(\Omega(t+\phi))\Big) \cdot \partial_x F^{(\phi)},$$

$$\begin{cases} F^{(\phi)}(x,0) & = & H(x) \\ F^{(\phi)}(x,t)|_{x=x_-} & \equiv & 0 \\ \partial_x F^{(\phi)}(x,t)|_{x=v_{\rm th}} & \equiv & 0. \end{cases} \tag{3.3.9}$$

Numerical solutions for eq. (3.3.9) are shown in fig. 3.4. We have used the standard Crank–Nicholson finite-difference algorithm (central-differences in space with equally weighted implicit-explicit terms in time, see [57]).

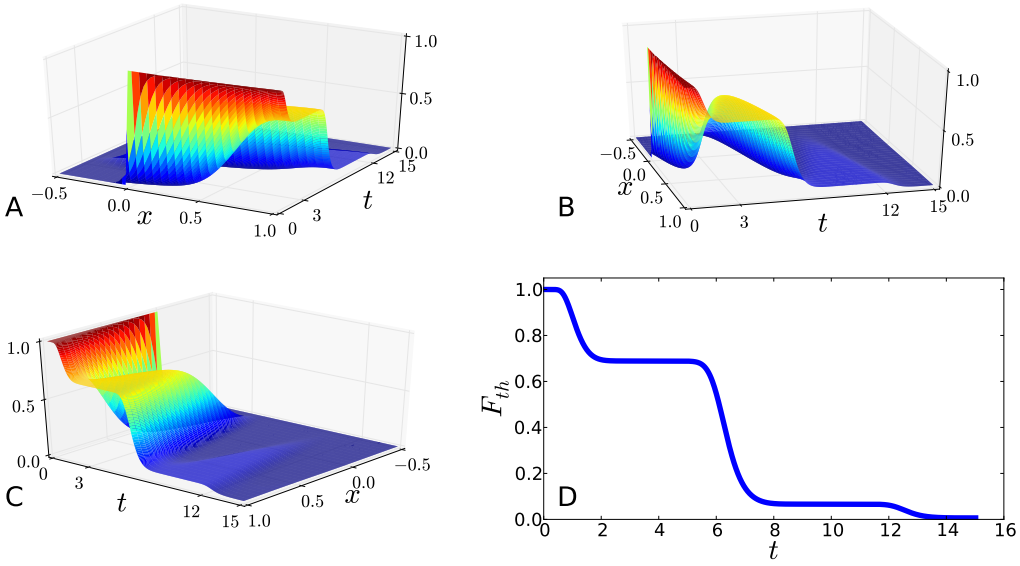


Figure 3.4: Example solution to eq. (3.3.9) for $(\alpha, \beta, \gamma) = (0.5, 0.3, 0.5\sqrt{2})$; $\Omega = 1, \phi = \pi/2$. In A,B,C, we show the full solution in space-time $F(x,t)$. In (d) we show the time solution at the upper boundary, $F(1,t)$.

3.3.3 Fortet Equation

Consider a general form of eq. (3.3.3),

$$dY_t = b(t, Y_t)dt + \sigma(t, Y_t)dW_t.$$

Let $\Phi(y, t|y_0, t_0) := \mathbb{P}[Y_t \leq y | Y_{t_0} = y_0]$ be the transition cumulative distribution of Y. Note that this is the distribution of Y_t in absence of a threshold, different from the distribution given in eq. (3.3.7), which is the distribution of the process constrained to be below the threshold. Now consider an arbitrary time-dependent threshold $v_{\text{th}}(t)$. The Fortet equation, see [17], convolves the first-hitting time probabilities, $g(t)$, with the transition density of the process. Integrating over $(-\infty, v_{\text{th}}(t))$, we obtain

$$1 - \Phi(v_{\text{th}}(t), t|v_0, 0) = \int_0^t g(\tau)[1 - \Phi(v_{\text{th}}(t), t|v_{\text{th}}(\tau), \tau)]\, \mathrm{d}\tau. \tag{3.3.10}$$

The left hand side is simply the probability of exceeding v_{th} at time t starting at v_0 at time 0. This can also be written as the probability of hitting v_{th} for the first time at time $\tau < t$ and then exceeding v_{th} at time t starting at v_{th} at time τ, integrated over all τ.

The Fortet equation is particularly appealing to use when we have an analytical expression for Φ. For the problem at hand, Φ is complicated due to the time-dependent forcing. However, the following transformation yields a time-homogeneous Y for which Φ will be tractable, along with an associated moving threshold, $v_{\text{th}}(t)$. This makes feasible the use of the Fortet equation. To obtain this transformation, cf. [58], consider the deterministic version of the SDE in eq. (3.3.3)

$$dv(t) = (\alpha - v + \gamma \sin(\Omega(t+\phi)))\ \mathrm{d}t, \tag{3.3.11}$$

$$v(0) = 0$$

with solution

$$v(t) = \alpha(1-\exp(-t)) + \frac{\gamma}{\sqrt{1+\Omega^2}}\left[\sin(\Omega(t+\phi)-\psi) - \exp(-t)\sin(\phi\Omega-\psi)\right]; \quad \psi = \arctan(\Omega). \tag{3.3.12}$$

Now take X_t, the solution to eq. (3.3.3) and $v(t)$, eq. (3.3.12), and let $Y_t = X_t - v(t)$. Then

$$dY_t = -Y_t dt + \beta\, \mathrm{d}W, \tag{3.3.13}$$

which has the time and parameter dependent threshold

$$v_{\text{th}\{\alpha,\gamma;\phi\}}(t) = v_{\text{th}} - v(t). \tag{3.3.14}$$

That is, X_t hits the constant threshold v_{th} if and only if Y_t hits the moving threshold $v_{\text{th}\{\alpha,\gamma;\phi\}}(t)$, where the subindex indicates the dependence on α, γ and ϕ. Therefore the ISIs produced by X and Y are the same and so are their distributions. Thus, $g_\phi(\tau)$ satisfies

$$1 - \Phi_{\{\beta\}}(v_{\text{th}\{\alpha,\gamma;\phi\}}(t), t|0,0) = \int_0^t g_\phi(\tau)\left[1 - \Phi_{\{\beta\}}\left(v_{\text{th}\{\alpha,\gamma;\phi\}}(t), t | v_{\text{th}\{\alpha,\gamma;\phi\}}(\tau), \tau\right)\right]\ \mathrm{d}\tau, \tag{3.3.15}$$

where

$$\Phi_{\{\beta\}}(y,t|y_0,t_0) = \frac{1}{\sqrt{\pi\beta^2(1-e^{-2(t-t_0)})}} \int_{-\infty}^{y} \exp\left(-\frac{(x-y_0e^{-(t-t_0)})^2}{\beta^2(1-e^{-2(t-t_0)})}\right) \mathrm{d}x$$

is the conditional cumulative distribution function of Y_t defined in eq. (3.3.13).

3.4 Parameter Estimation Algorithms

The unknown parameters in eq. (3.3.3) are α, β and γ, while we assume Ω known. The reason why the amplitude, γ, is often unknown while the frequency, Ω, is known is that one can usually observe the sinusoidal input and thus its frequency. Further, the encoding of the input into neuronal firing patterns often involves phase locking to the sinusoidal component. However, the actual forcing amplitude at the level of the neuron is usually modified by various synaptic and other filtering processes, unless the cell receives direct sinusoidal current injection.

Our goal is to estimate the structural parameters (α, β, γ) from a sample of spike time data, $\{i_1, \ldots, i_N\}$. There are several algorithms for estimating the parameters for the simpler and more common case of $\gamma = 0$. One such algorithm relies on the Fortet equation, see [39, 40], which we extend to the presence of a time-varying current. A more basic approach is to directly solve the Fokker-Planck equation for the probability density of X_t, [53, 51, 52], from which one can derive the survival distribution of I_n and use this to compare against the empirical survival distribution of I_n obtained from data. An approximate maximum likelihood approach is also possible by numerical differentiation. The relation between Fokker-Planck equations and the first-passage time problem is discussed in most introductory books on stochastic analysis, see, for example, [13]. A recent review of this approach for the simple $\gamma = 0$ case in neuronal modeling can be found in [53], wherein the first passage problem is discussed at great lengths in the context of spiking neurons. We will use this in section 3.3.2. A more elaborate approach using the Fokker-Planck equation to approximate the hitting time

distribution is given in [59]. The techniques in [59] avoid the need to compute the Fokker-Planck PDE numerically, instead approximating it with analytically known solutions. This approach might offer significant computational savings, but since this would at most amount to a computational speed-up of our algorithm, we have left this unexplored for now.

The immediate problem in generalizing the aforementioned approaches to the case of $\gamma \neq 0$ is that the I_n's are no longer identically distributed since the phase ϕ_{n-1} of the nth interval I_n depends on t_{n-1}, the time the previous spike occurred. The I_n's are also dependent, but conditionally independent given ϕ_{n-1}. So the trajectories in each interval are parametrized by the value of ϕ_{n-1} at the time of the last spike/reset. We overcome this obstacle by splitting the I_n's in groups, and approximating the I_n's within groups as coming from identically distributed trajectories in a sense to be specified below. This approximation which solves the challenge of dependent and non-identically distributed ISIs is the primary contribution of this chapter.

3.4.1 ϕ - binning

Before we can use eq. (3.3.9) or (3.3.15), we need to deal with the fact that ϕ is not fixed, but instead each I_n starts with a distinct ϕ_n. Our approach is to partition the interval $[0, 2\pi/\Omega]$ into M bins, where $M \ll N$, and represent each bin by the midpoint of the bin, ϕ_m. Then we approximate the N observed ϕ_n's by the closest ϕ_m and pretend that any observed I_n was not produced by a trajectory of the form in eq. (3.3.3) with $\phi = \phi_n$, but with $\phi = \phi_m$. Our hope is that for a judicious choice of M, we can balance the error of $\phi_n \neq \phi_m$ with having enough data points in each bin in order to obtain a useful estimate from eq. (3.3.9) or (3.3.15).

There is clearly much freedom in how one sets up these bins, but we will do the simplest thing and make them all of equal width, $\delta\phi = 2\pi/(\Omega M)$. Each ϕ_n will belong to one and only one of the bins $[\phi_m - \delta\phi/2, \phi_m + \delta\phi/2)_{m=1}^{M}$, with centre points

$\phi_m = \delta\phi/2 + (m-1)\delta\phi$, for $m = 1, \ldots, M$. Thus, given an empirically observed I_n with associated ϕ_n, we will pretend that it was produced by the process

$$dX_s = (\alpha - X_s)\,\mathrm{d}s + \gamma \sin(\Omega(s + \phi_m(n)))\,\mathrm{d}s + \beta\,\mathrm{d}W_s,$$

where

$$\phi_m(n) = \arg\min_{\phi_m} |\phi_n - \phi_m|.$$

This binning is illustrated in fig. 3.5.

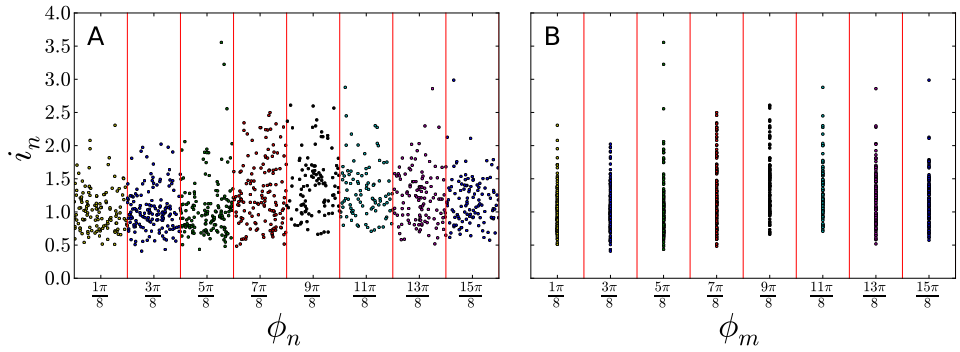


Figure 3.5: The raw (i_n, ϕ_n) pairs (left) are binned into a set of M bins with a representative ϕ_m (right) and the ISIs within each bin are treated as a renewal process. In this illustration, $M = 8$, $\Omega = 1$ while the parameters α, β, γ are taken from the supra-threshold regime.

While we have no rigorous approach to determine the value of M, our limited experience suggests that given $N = 1000$ ISIs, $M = 10$ or $M = 20$ gives satisfactory results for very different parameter regimes. In general, choosing M is a balancing act. For M too high, the resulting bins will have too few data points to approximate $\bar{G}(I)$ accurately. Therefore M is forced to be small when sample size is not large. For M too low, the approximation of the phase shifts will be poor, leading to a biased estimate of $\bar{G}(I)$. We illustrate the effect of increasing M in fig. 3.6. Generally, as long as there are sufficient data points, as M increases, the approximation of using

the survival distribution with ϕ_m instead of ϕ_n improves since $\phi_m(n) \to \phi_n$ as $M \to \infty$. In the sequel, we will use $M = 20$ for sample sizes of $N = 1000$ and $M = 8$ for sample sizes of $N = 100$.

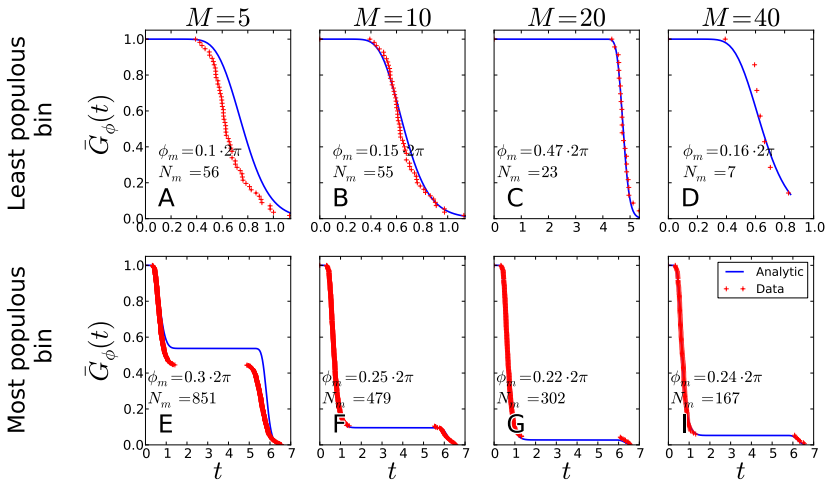


Figure 3.6: Effect of M, the number of bins, on the approximate survival distribution. The full-drawn blue curve is the true survivor distribution given in eq. (3.3.9), the red points are the approximation given in eq. (3.4.1). In the figures, the least populous (above) and most populous (below) bin for each M is shown. The width of the bins is $\delta\phi = 2\pi/(\Omega M)$. We have used A,E) $M = 5$, B,F) $M = 10$, C,G) $M = 20$, D,I) $M = 40$. As M increases, the approximation of using the survival distribution using ϕ_m instead of ϕ_n improves since $\phi_m(n) \to \phi_n$ as $M \to \infty$. The data is generated using parameter values from the super-sinusoidal regime and $N = 1000$. For this particular data set the largest generated ISI was 6.55 time units.

3.4.2 Fokker-Planck Algorithm

Within each bin it is clear how to apply eq. (3.3.7). In the mth bin, for a given ϕ_m, we approximate $\bar{G}_\phi(t)$ by

$$\hat{\bar{G}}_{\phi_m}(t) = \frac{\#[i_n > t \,\big|\, \phi_{n-1} \in [\phi_m - \delta\phi/2, \phi_m + \delta\phi/2)]}{N_m}, \tag{3.4.1}$$

where N_m is the number of ISIs in bin m. Using eq. (3.3.7) we define the loss function

$$L(\alpha, \beta, \gamma) = \sum_{\phi_m} N_m \Big\{ \sup_{t>0} \Big| \hat{\bar{G}}_{\phi_m}(t) - F^{\phi_m}_{\alpha,\beta,\gamma}(x_{th}, t) \Big| \Big\}. \tag{3.4.2}$$

The weight N_m is included so that bins with larger sample sizes have a larger influence on the estimates.

To evaluate the supremum in eq. (3.4.2), we spline interpolate the empirically discrete $\hat{\bar{G}}$ for each ϕ_m, sample at the time nodes of the PDE discretization and finally take the maximum amongst the sampled values. We then minimize L using an optimization algorithm (see below, section 3.5) and take our estimates $\hat{\alpha}, \hat{\beta}, \hat{\gamma}$ to be

$$\hat{\alpha}, \hat{\beta}, \hat{\gamma} = \arg\min_{\alpha,\beta,\gamma} L(\alpha, \beta, \gamma).$$

Note that the relation between the spike time survival density, $\bar{G}_\phi$ and the transition distribution, F_ϕ, in eq. (3.3.7) could also allow for an approximate maximum likelihood estimator (MLE), based on maximizing

$$L^{\mathrm{MLE}}(\alpha, \beta, \gamma) = \sum_n \log(g_{\phi_{n-1}}(i_n)) = \sum_n \log \big[-\partial_t F^{\phi_{n-1}}_{\alpha,\beta,\gamma}(x_{th}, t) \big] \Big|_{t=i_n},$$

where the derivative has to be approximated by finite differences. We can then again use binning to avoid having to compute the PDE separately for each (i_n, ϕ_{n-1}). Our experience with the MLE approach has been that the quality of the estimates provided are similar to those obtained by minimizing eq. (3.4.2) and that the associated computing times are on the same order.

3.4.3 Fortet Algorithm

An alternative approach is to form a loss function from eq. (3.3.15). This is similar to what is done in [39, 40] for the simpler case of a constant threshold. Noting that $\int_0^t g(\tau)[1-\Phi]\,\mathrm{d}\tau = \mathbb{E}[(1-\Phi)\mathbb{1}_{I\leq t}]$ where the expectation is taken with respect to the distribution of the random variable I, we can use the fact that the ISIs are approximately independent and invoke the law of large numbers to estimate the integral as

$$\int_0^t g_{\phi_m}(\tau)\big[1-\Phi^{(\phi)}_{\{\beta\}}(v_{\mathrm{th}\{\alpha,\gamma;\phi\}}(t),t\,|\,v_{\mathrm{th}\{\alpha,\gamma;\phi\}}(\tau),\tau)\big]\,\mathrm{d}\tau \approx 1/N_m \sum_{i_n<t}\big[1-\Phi^{(\phi)}_{\{\beta\}}(v_{\mathrm{th}\{\alpha,\gamma;\phi\}}(t),t\,|\,v_{\mathrm{th}\{\alpha,\gamma;\phi\}}(i_n),i_n)\big].$$

We then define the loss function

$$L(\alpha,\beta,\gamma) = \sum_{\phi_m} N_m \Bigg\{ \sup_{s>0} \Big| 1-\Phi^{(\phi_m)}_{\{\beta\}}(v_{\mathrm{th}\{\alpha,\gamma;\phi\}}(s),s\,|\,0,0)] - \tfrac{1}{N_m}\sum_{i_n<s}\big[1-\Phi^{(\phi_m)}_{\{\beta\}}(v_{\mathrm{th}\{\alpha,\gamma;\phi\}}(s),s\,|\,v_{\mathrm{th}\{\alpha,\gamma;\phi\}}(i_n),i_n)\big]\Big| / \omega(\phi_m;\alpha,\beta,\gamma)\Bigg\}. \tag{3.4.3}$$

We divide each inner term by $\omega(\phi_m;\alpha,\beta,\gamma) = \sup_{s>0}|1-\Phi^{(\phi_m)}_{\alpha,\beta,\gamma}(v_{\mathrm{th}}(s),s|v_0)|$, following the suggestion in [40]. This scaling ensures that eq. (3.3.15) divided by $\omega(\alpha,\beta,\gamma)$ will vary between 0 and 1 for all parameter values thus giving sense to the measure defined by the loss function. Since we can solve in closed form for Φ, we have all we need given an observed spike train of i_n's. We evaluate the sup by sampling at $K = 500$ uniformly spaced points in $(0, I_{\max}+\epsilon]$ and taking the maximum of the sampled values.

3.4.4 Initialization of the algorithms

The parameter search can be initialized in a simple way using the fact that the Fokker-Planck PDE is almost an 'advection-diffusion' equation whose solution is almost a

Gaussian. Then $\bar{G}(t)$ can be approximated by the amount of probability mass of a Gaussian to the left of the threshold at time t. The idea is as follows. Suppose we are solving the following PDE

$$\partial_t \rho = -U\partial_x[\rho] + \frac{\beta^2}{2}\partial_x^2[\rho]. \tag{3.4.4}$$

Its solution given an initial condition $\rho(x,0) = \delta(x)$ will be a Gaussian bell moving to the right with speed U and standard deviation $\sigma = \beta\sqrt{t}$.

The survivor function $\bar{G}(t)$ can be thought of as the amount of area that has passed the threshold (from the left moving to the right). We can then invert the information about $\bar{G}$ to estimate U and β. In particular, a Gaussian bell has ≈ 0.158 of its mass more than one standard deviation to the right of its mean. Thus, at time t_1 such that $\bar{G}(t_1) = 0.842$, the right tail of more than one standard deviation of the Gaussian bell has crossed the threshold. The threshold is at $x = 1$ and we obtain the following equation

$$Ut_1 + \beta\sqrt{t_1} = 1. \tag{3.4.5}$$

Similarly, at time t_2 such that $\bar{G}(t_2) = 1 - 0.842$, the Gaussian bell has crossed the threshold except for the left tail and we have

$$Ut_2 - \beta\sqrt{t_2} = 1. \tag{3.4.6}$$

If U and β were constant, then eqs. (3.4.5) and (3.4.6) provide two equations in two unknowns. However, $U = U(x,t) = (\alpha - x + \gamma\sin(\Omega(t+\phi)))$ is not constant and we approximate U as

$$U(x,t) \approx \alpha - 0.5 + \gamma\frac{1}{t}\int_0^t \sin(\Omega(\tau+\phi))\,\mathrm{d}\tau, \tag{3.4.7}$$

i.e. we approximate the space-dependent term, x, with the mid-point between the reset value, $v_0 = 0$, and the threshold, $v_{\text{th}} = 1$, and we approximate the time-dependent term, $\sin(\Omega(\tau+\phi))$, by its time-average value between 0 and t. If we use the 0^{th}, 1^{st}

and 2^{nd} standard deviation points, we can form 5 equations in 3 unknowns as follows

$$\begin{aligned}
\alpha t_1 + \gamma s(t_1) + 2\beta\sqrt{t_1} &= 1 + 0.5t_1 \\
\alpha t_2 + \gamma s(t_2) + \beta\sqrt{t_2} &= 1 + 0.5t_2 \\
\alpha t_3 + \gamma s(t_3) + 0\beta &= 1 + 0.5t_3 \\
\alpha t_4 + \gamma s(t_4) - 1\beta\sqrt{t_4} &= 1 + 0.5t_4 \\
\alpha t_5 + \gamma s(t_5) - 2\beta\sqrt{t_5} &= 1 + 0.5t_5
\end{aligned}$$

with the time-average weighting function $s(t) = (\cos(\Omega\phi) - \cos(\Omega(t+\phi)))/\Omega$. However, the approximation is best for earlier times, when the solution is closer to a Gaussian bell that is approaching the threshold, but less correct for later times, since it neglects the loss of probability mass and thus overestimates the backward probability current. Indeed, we have found it to be best to use only t_1 and t_2. In the following we use only these equations

$$\begin{aligned}
\alpha t_1 + \gamma s(t_1) + 2\beta\sqrt{t_1} &= 1 + 0.5t_1 \\
\alpha t_2 + \gamma s(t_2) + \beta\sqrt{t_2} &= 1 + 0.5t_2
\end{aligned}$$

for the initializer. We can form these equations separately for each ϕ_m bin, thus resulting in $M \times 2$ equations for the unknowns α, β and γ. Since we have more equations than unknowns, we use least-squares estimates in a regression to pick out unique α, β and γ estimates.

The proposed initialization procedure has two advantages. First, it is automatic, i.e. it requires only the data and no input or guidance from the user. Second, it is extremely fast. While the precise effect of the initializer is shown in section 3.5, it is intuitively clear that it will work best in the supra-threshold parameter regime when the bell curve is truly moving past the threshold as a whole and less so for sub-threshold regimes, when only the diffusive force serves to propel the process to reach v_{th}. The behaviour of the initializer in the different regimes is illustrated in fig. 3.7.

What we show in fig. 3.7 is the following: First we show the survival distribution for a given regime and ϕ_m fixed. Then using data generated from such a regime and with ϕ_n in the mth bin, the initializer tries to find the best approximation by the motion of a Gaussian bell which will fit this data, in the sense of solving for α, β, γ as previously described. Once this is done, we then show in red the amount of area under this Gaussian bell to the left of the threshold. Of course the interpretation of the survival distribution for an ISI as a fraction of the area under a moving bell with conserved total area is wrong, but the assumption is useful in automatically generating initial values for the more appropriate approximations to start their work.

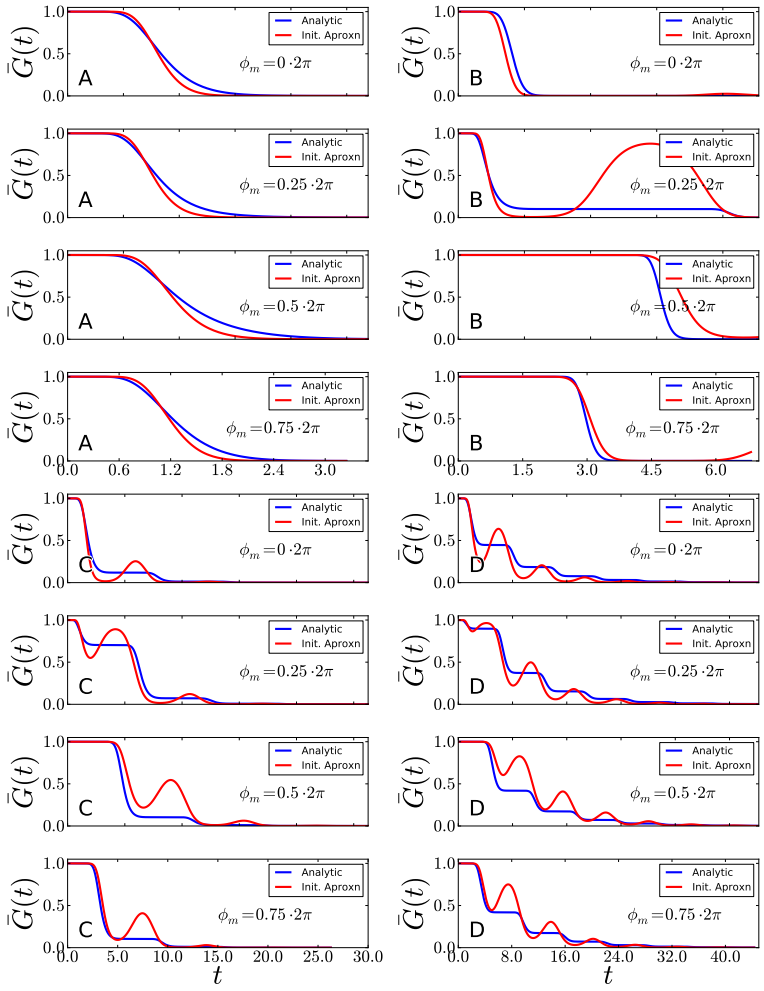


Figure 3.7: The blue curves are the numerically obtained survivor distributions $\bar{G}_\phi$ for the exact parameters in the four regimes (as in table 4.1) and $\Omega = 1$. The red curves are obtained in the following manner: Simulations using the true parameters were used to generate sample spikes. Using these samples, the initializer algorithm was used to generate estimates for α, β, γ. Using these estimates, the bell curve discussed in sec. 3.4 was formed and evolved in time. Thus, the red curve drawn in the figures measures the area under this bell that is to the left of the threshold at time t. A) supra-threshold, B) super-sinusoidal, C) critical, D) sub-threshold.

3.5 Method Comparison on Simulated Data

We will now use our algorithms on spike trains simulated from the four different regimes; the supra-threshold, the critical, the super-sinusoidal and the sub-threshold. We have used 100 sample spike trains per regime, with $N = 100$ as well as $N = 1000$ spikes per train. In order to perform the numerical minimization of eqs. (3.4.2) and (3.4.3), we have used an implementation of the Nelder-Mead algorithm from the SciPy library [60]. The Nelder-Mead algorithm is a non-linear minimization routine which uses a bounding-polygon method to zero-in on the minimum and thus avoids the need to provide the gradient of the loss function. It is the standard non-gradient minimization algorithm.

The estimation results are shown in figs. 3.8 to 3.11, where we plot box plots for the estimates, $\hat{\alpha}, \hat{\beta}, \hat{\gamma}$ in the four regimes. We also tabulate the average and the empirical 95% confidence intervals of the estimates in tables 3.2 and 3.3. Conclusions that can be drawn from these results are as follows. The initializer method is effective for the supra-threshold regime and gives reasonable ballpark estimates for all regimes, though the error can be substantial for the super-sinusoidal regime. In general, both the Fortet and Fokker-Planck algorithm estimate the parameters well in the supra-threshold, critical and super-sinusoidal regimes. The estimators variance is especially low in the supra-threshold regime, while it is higher for the critical and super-sinusoidal regimes. In the super-sinusoidal regime the two algorithms give accurate estimates even though the initializer can be quite off. On the other hand, in the sub-threshold regime the initializer has a performance comparable to that of the two more involved methods. It seems that distinguishing between the constant bias and the sinusoidal current is difficult if their sum is not sufficient to generate spikes without noise.

The Fokker-Planck method has a larger bias but a smaller spread than the Fortet method for $N = 100$, table 3.2. However for $N = 1000$, the two methods have

comparable spreads, while the Fortet method retains a smaller bias, see table 3.3. More precisely, for $N = 1000$, the Fokker-Planck method has a smaller spread in the sub-threshold regime, while the Fortet method has a smaller spread in the super-sinusoidal regime. As such, at least in the super-sinusoidal regime, the Fortet method seems superior.

The two algorithms are numerically intensive. For $N = 100$ and $N = 1000$ spikes, we show the times taken for the estimation in table 3.4. While we have done most of our numerical work in Python/SciPy[60], we have implemented the critical components of both algorithms in C. That is we solve the inner part of eq. (3.4.3) and the Fokker-Planck PDE, eq. (3.3.9), in C using the GSL libraries[61]. From table 3.4, we can verify that the computing time for the Fortet algorithm scales proportionally with the number of spikes. This is to be expected, since the Fortet equation has a term of the form $\sum_{i_n}$ which in turn has N terms and this forms the bulk of the computing time for the Fortet equation. The Fokker-Planck algorithm, on the other hand, scales less-than-linearly with N, since the dependency on N is in forming the approximation, $\hat{\bar{G}}$ to the survivor function and that is not computationally intensive (solving the PDE is).

Parameter	Initializer	Fokker-Planck	Fortet
Supra-threshold regime			
$\alpha = 1.40$	$1.43 : [1.29, 1.56]$	$1.34 : [1.24, 1.43]$	$1.41 : [1.33, 1.49]$
$\beta = 0.30$	$0.17 : [0.10, 0.24]$	$0.29 : [0.21, 0.39]$	$0.29 : [0.22, 0.36]$
$\gamma = 0.14$	$0.16 : [0.02, 0.33]$	$0.12 : [0.02, 0.23]$	$0.12 : [0.01, 0.24]$
Super-sinusoidal regime			
$\alpha = 0.10$	$0.92 : [0.83, 1.01]$	$0.28 : [0.02, 0.59]$	$0.24 : [-0.22, 0.42]$
$\beta = 0.30$	$0.15 : [0.10, 0.25]$	$0.31 : [0.14, 0.53]$	$0.32 : [0.14, 0.46]$
$\gamma = 1.98$	$1.35 : [1.13, 1.57]$	$1.67 : [1.33, 2.05]$	$1.77 : [1.44, 2.38]$
Critical regime			
$\alpha = 0.50$	$0.72 : [0.66, 0.80]$	$0.57 : [0.32, 0.73]$	$0.57 : [0.36, 0.73]$
$\beta = 0.30$	$0.19 : [0.10, 0.26]$	$0.27 : [0.17, 0.40]$	$0.25 : [0.15, 0.40]$
$\gamma = 0.71$	$0.57 : [0.44, 0.73]$	$0.55 : [0.30, 0.83]$	$0.62 : [0.38, 0.93]$
Sub-threshold regime			
$\alpha = 0.40$	$0.62 : [0.57, 0.67]$	$0.63 : [0.33, 0.84]$	$0.58 : [0.03, 1.00]$
$\beta = 0.30$	$0.17 : [0.10, 0.29]$	$0.20 : [0.10, 0.37]$	$0.19 : [0.00, 0.41]$
$\gamma = 0.57$	$0.32 : [0.00, 0.53]$	$0.29 : [0.00, 0.62]$	$0.46 : [0.00, 1.19]$

Table 3.2: Averages and empirical 95% confidence intervals of the estimates for $N = 100$ spikes per train.

Parameter	Initializer	Fokker-Planck	Fortet
Supra-threshold regime			
$\alpha = 1.40$	$1.44 : [1.40, 1.50]$	$1.36 : [1.33, 1.40]$	$1.40 : [1.37, 1.42]$
$\beta = 0.30$	$0.25 : [0.22, 0.28]$	$0.29 : [0.26, 0.32]$	$0.30 : [0.27, 0.32]$
$\gamma = 0.14$	$0.14 : [0.10, 0.19]$	$0.14 : [0.10, 0.17]$	$0.14 : [0.10, 0.18]$
Super-sinusoidal regime			
$\alpha = 0.10$	$0.90 : [0.85, 0.92]$	$0.11 : [0.03, 0.29]$	$0.10 : [0.03, 0.16]$
$\beta = 0.30$	$0.18 : [0.14, 0.23]$	$0.30 : [0.21, 0.34]$	$0.31 : [0.22, 0.34]$
$\gamma = 1.98$	$1.26 : [1.16, 1.34]$	$1.92 : [1.49, 2.05]$	$1.96 : [1.86, 2.07]$
Critical regime			
$\alpha = 0.50$	$0.73 : [0.70, 0.75]$	$0.51 : [0.43, 0.63]$	$0.53 : [0.45, 0.64]$
$\beta = 0.30$	$0.20 : [0.17, 0.24]$	$0.29 : [0.24, 0.32]$	$0.28 : [0.19, 0.33]$
$\gamma = 0.71$	$0.54 : [0.44, 0.61]$	$0.66 : [0.52, 0.76]$	$0.67 : [0.54, 0.77]$
Sub-threshold regime			
$\alpha = 0.40$	$0.62 : [0.55, 0.65]$	$0.57 : [0.45, 0.66]$	$0.56 : [0.26, 0.71]$
$\beta = 0.30$	$0.20 : [0.17, 0.26]$	$0.22 : [0.18, 0.29]$	$0.21 : [0.13, 0.35]$
$\gamma = 0.57$	$0.36 : [0.18, 0.44]$	$0.36 : [0.25, 0.50]$	$0.43 : [0.28, 0.72]$

Table 3.3: Averages and empirical 95% confidence intervals of the estimates for $N = 1000$ spikes per train.

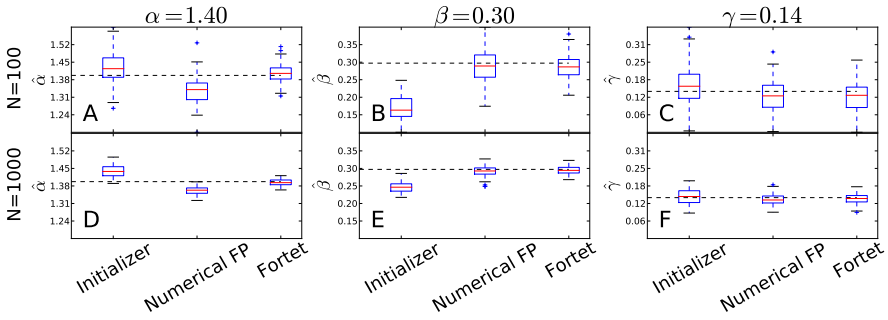


Figure 3.8: Boxplots of parameter estimates for the supra-threshold regime. The upper plots (A,B,C) show estimates using $N = 100$ sample spikes per estimation, while the lower plots (D,E,F) use $N = 1000$. The dashed line indicates the true parameter value, while the red line inside the boxes indicates the median of the estimates.

The boxes contain the central 50% of the estimates. The bars indicate the range of the estimates, except for outliers given by the points outside the bars, and defined to be more than 1.5 times the interquantile range (the height of the box) from the box.

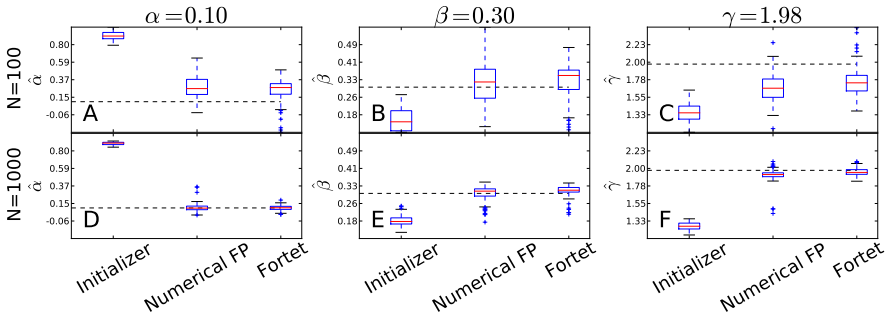


Figure 3.9: Boxplots of parameter estimates for the super-sinusoidal regime. The upper plots (A,B,C) show estimates using $N = 100$ sample spikes per estimation, while the lower plots (D,E,F) use $N = 1000$. The dashed line indicates the true parameter value, while the red line inside the boxes indicates the median of the estimates.

The boxes contain the central 50% of the estimates. The bars indicate the range of the estimates, except for outliers given by the points outside the bars, and defined to be more than 1.5 times the interquantile range (the height of the box) from the box.

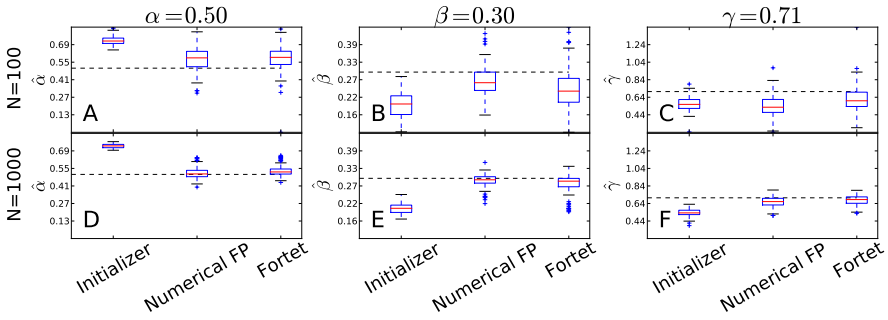


Figure 3.10: Boxplots of parameter estimates for the critical regime. The upper plots (A,B,C) show estimates using $N = 100$ sample spikes per estimation, while the lower plots (D,E,F) use $N = 1000$. The dashed line indicates the true parameter value, while the red line inside the boxes indicates the median of the estimates.

The boxes contain the central 50% of the estimates. The bars indicate the range of the estimates, except for outliers given by the points outside the bars, and defined to be more than 1.5 times the interquantile range (the height of the box) from the box.

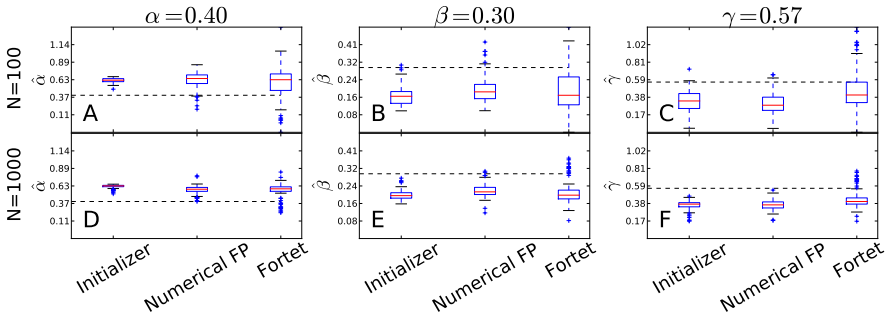


Figure 3.11: Boxplots of parameter estimates for the sub-threshold regime. The upper plots (A,B,C) show estimates using $N = 100$ sample spikes per estimation, while the lower plots (D,E,F) use $N = 1000$. The dashed line indicates the true parameter value, while the red line inside the boxes indicates the median of the estimates.

The boxes contain the central 50% of the estimates. The bars indicate the range of the estimates, except for outliers given by the points outside the bars, and defined to be more than 1.5 times the interquantile range (the height of the box) from the box.

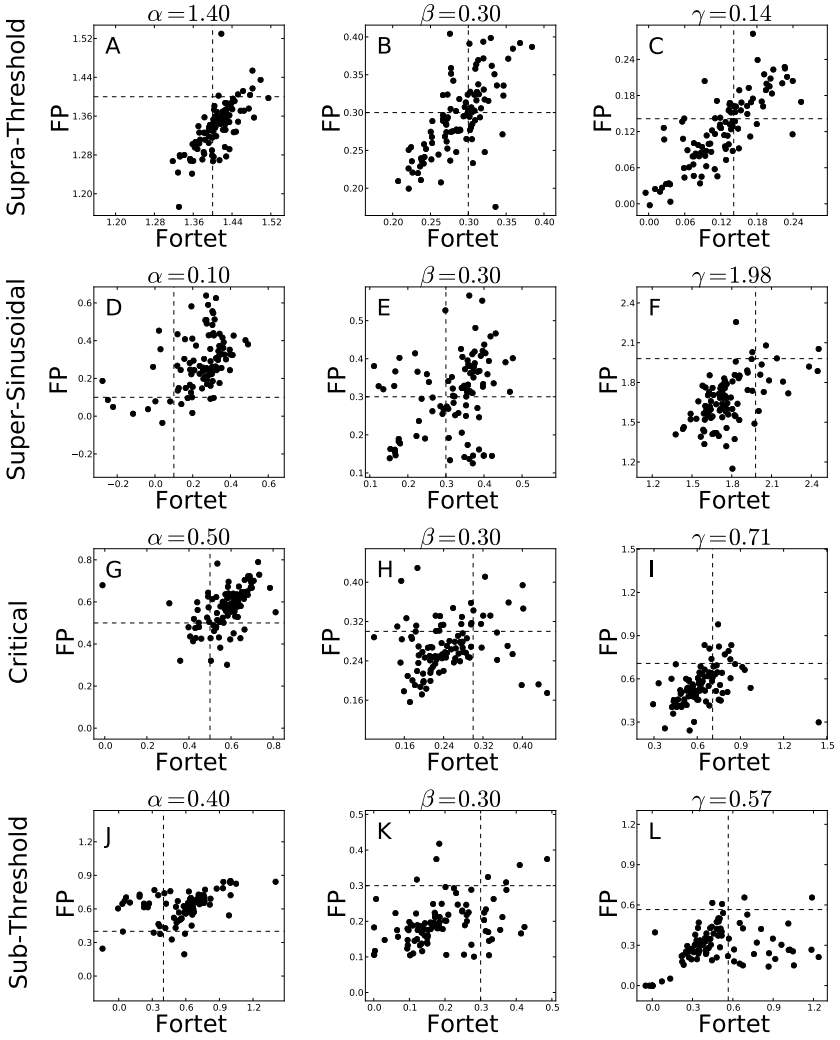


Figure 3.12: Estimates based on samples of $N = 100$ spikes obtained from the Fokker-Planck algorithm against the Fortet algorithm for the four different parameter regimes, with parameter values given in table table 4.1, fixing $\Omega = 1$. Each row corresponds to one regime and one set of simulations. Each column corresponds to a parameter, with the specific value indicated above each plot. A,B,C) Supra-threshold; D,E,F) Super-sinusoidal; G,H,I) Critical; J,K,L) Sub-threshold.

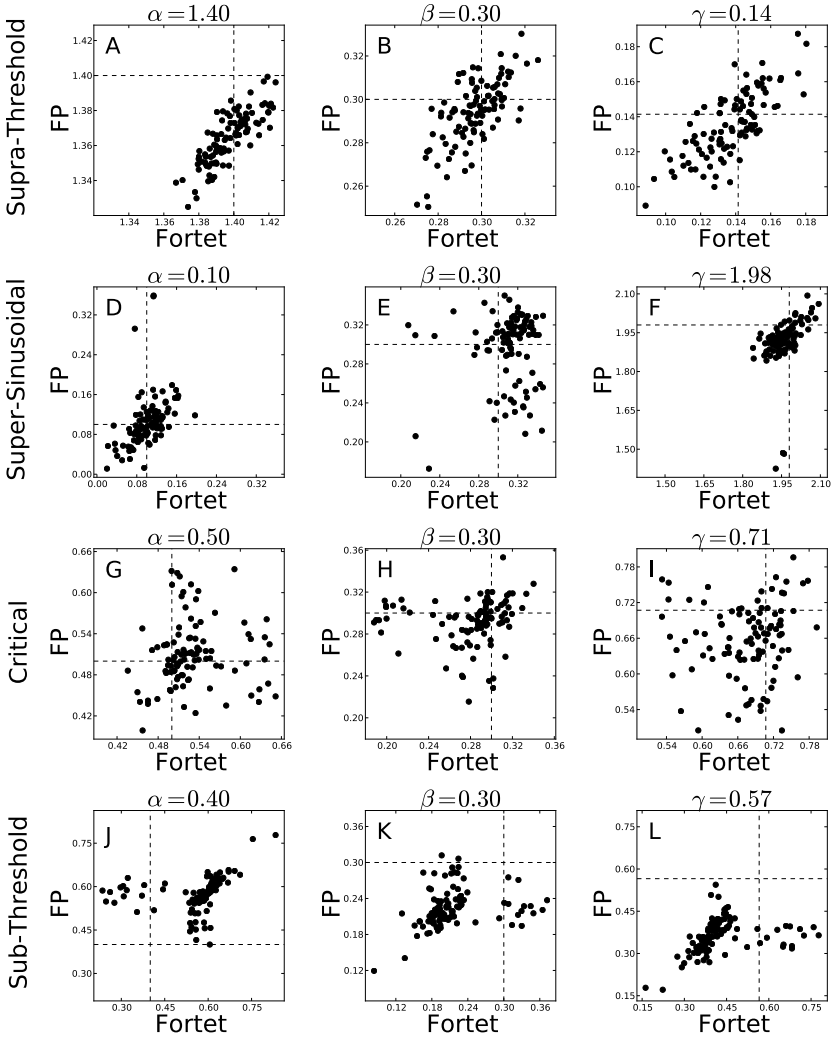


Figure 3.13: Estimates based on samples of $N = 1000$ spikes obtained from the Fokker-Planck algorithm against the Fortet algorithm for the four different parameter regimes, with parameter values given in table table 4.1, fixing $\Omega = 1$. Each row corresponds to one regime and one set of simulations. Each column corresponds to a parameter, with the specific value indicated above each plot. A,B,C) Supra-threshold; D,E,F) Super-sinusoidal; G,H,I) Critical; J,K,L) Sub-threshold.

Regime	Fortet	Fokker-Planck
Sub-threshold	1.29 ± 0.72	0.52 ± 0.21
Supra-threshold	0.83 ± 0.28	0.18 ± 0.20
Critical	0.94 ± 0.42	0.36 ± 0.16
Super-sinusoidal	1.36 ± 0.46	0.43 ± 0.17

(a) N=100

Regime	Fortet	Fokker-Planck
Sub-threshold	9.68 ± 4.98	1.69 ± 0.91
Supra-threshold	3.90 ± 1.05	0.21 ± 0.06
Critical	10.03 ± 2.88	1.28 ± 0.41
Super-sinusoidal	10.13 ± 2.24	1.06 ± 0.33

(b) N=1000

Table 3.4: Estimator Algorithm Computational Time (in seconds)

3.6 The effect of Ω

So far we have held Ω constant and equal to 1. We now investigate the effect of varying Ω on the quality of estimates. To narrow the scope, we focus on increasing Ω while keeping the parameters in the critical regime such that $\alpha + \gamma/\sqrt{1+\Omega^2} = 1$ and $\alpha = 0.5$. This amounts to increasing γ with Ω. We do the estimations for four values of $\Omega = [1, 5, 10, 20]$. Similarly to the previous section, we use 100 sample spike trains per parameter set, with each spike train consisting of $N = 1000$ ISIs.

We show box plots of the estimates for each Ω in fig. 3.14. We then directly compare the two algorithms, Fortet vs. Fokker-Planck, in fig. 3.15. The immediate observation is that the Fokker-Planck algorithm fails to keep up at the higher frequencies and consistently underestimates γ. The Fortet algorithm does better, but still underestimates γ. In general, this underestimation of γ is accompanied by an over-estimation of α. This is exacerbated at higher Ω. We illustrate the relation between estimates for α vs. γ in fig. 3.16, where it is quite clear that an underestimation of γ is proportional to the overestimation of α.

For completeness we also include the estimates' average and empirical 95% confidence intervals in table 3.5.

3.7 Maximum Likelihood Estimation Procedure

We detail the maximum likelihood (ML) approach to estimating the parameters α, β, γ in the sinusoidal, noisy LIF model.

We start by recalling the relationship between the ISI survival density, g, the survival distribution $\bar{G}$ and the Fokker-Planck transition distribution, F.

$$\bar{G}_\phi(t) = F^{(\phi)}(1, t)$$

and

$$g_\phi(t) = -\partial_t \bar{G}_\phi(t).$$

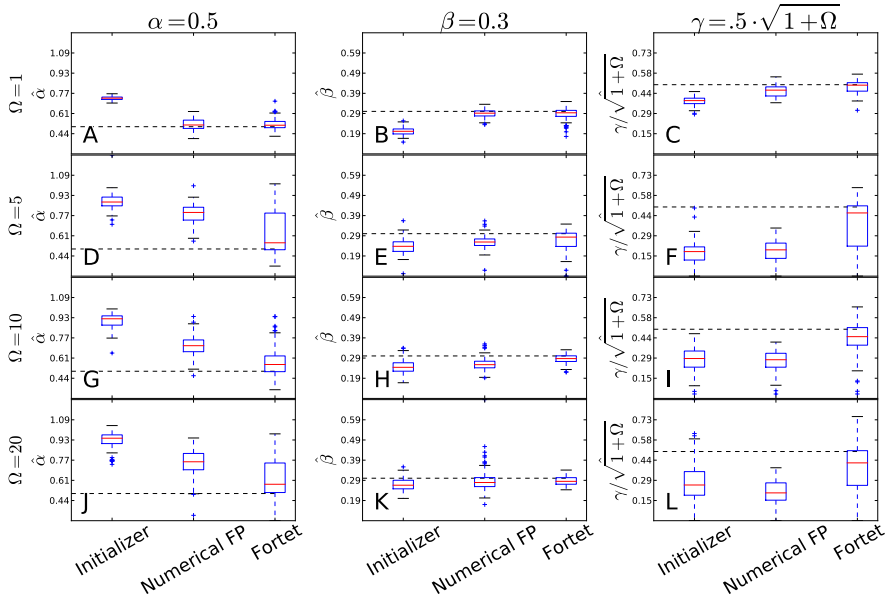


Figure 3.14: Boxplots of parameter estimates for varying Ω across $[1, 5, 10, 20]$ while holding $\gamma/\sqrt{1+\Omega^2}$ constant as to keep the parameters in the critical regime. A-C) $\Omega = 1$, D-F) $\Omega = 5$, G-I) $\Omega = 10$, J-L) $\Omega = 20$.

The boxes contain the central 50% of the estimates. The bars indicate the range of the estimates, except for outliers given by the points outside the bars, and defined to be more than 1.5 times the interquantile range (the height of the box) from the box.

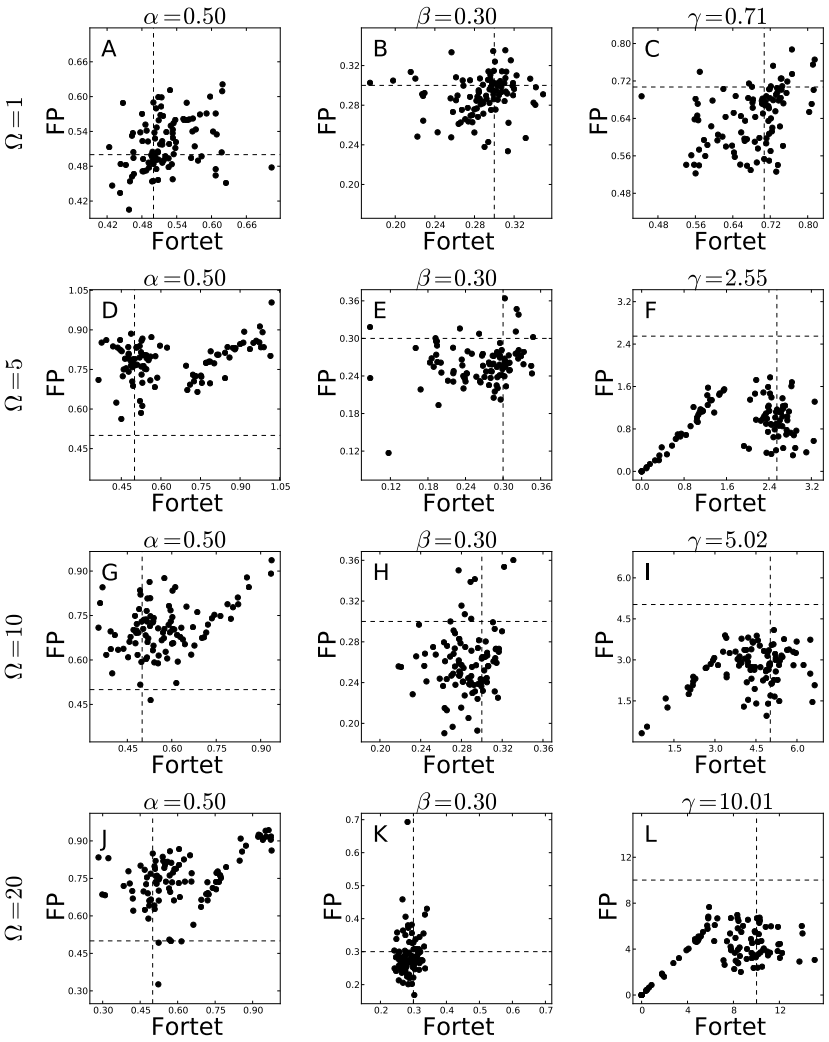


Figure 3.15: Estimates based on samples of $N = 1000$ spikes obtained from the Fokker-Planck algorithm against the Fortet algorithm for a parameter set in the critical regime, while varying Ω across $[1, 5, 10, 20]$ and holding $\gamma/\sqrt{1+\Omega^2}$ and α constant. A,B,C) $\Omega = 1$; D,E,F) $\Omega = 5$; G,H,I) $\Omega = 10$; J,K,L) $\Omega = 20$.

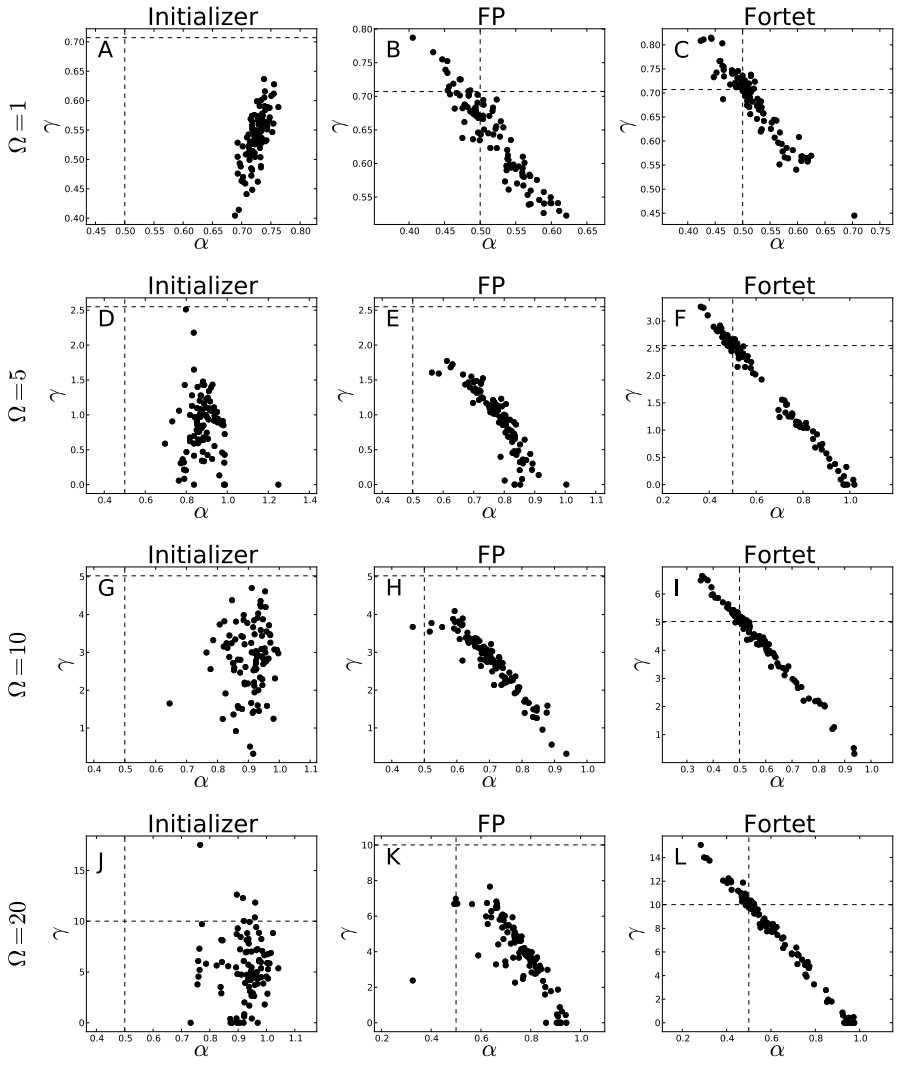


Figure 3.16: Comparison of $\hat{\alpha}$ vs. $\hat{\gamma}$ parameter estimates while varying Ω across $[1, 5, 10, 20]$, holding $\gamma/\sqrt{1+\Omega^2}$ constant as to keep the parameters in the critical regime. A,B,C) $\Omega = 1$; D,E,F) $\Omega = 5$; G,H,I) $\Omega = 10$; J,K,L) $\Omega = 20$.

Parameter	Initializer	Fokker-Planck	Fortet
	$\Omega = 1$		
$\alpha = 0.50$	$0.73 : [0.69, 0.75]$	$0.52 : [0.45, 0.61]$	$0.52 : [0.44, 0.62]$
$\beta = 0.30$	$0.20 : [0.17, 0.25]$	$0.29 : [0.24, 0.33]$	$0.29 : [0.22, 0.34]$
$\gamma = 0.71$	$0.54 : [0.44, 0.62]$	$0.64 : [0.53, 0.75]$	$0.68 : [0.55, 0.81]$
	$\Omega = 5$		
$\alpha = 0.50$	$0.88 : [0.76, 0.99]$	$0.78 : [0.61, 0.89]$	$0.64 : [0.39, 0.99]$
$\beta = 0.30$	$0.24 : [0.17, 0.31]$	$0.26 : [0.20, 0.34]$	$0.27 : [0.12, 0.34]$
$\gamma = 2.55$	$0.85 : [0.00, 1.65]$	$0.92 : [0.00, 1.68]$	$1.86 : [0.00, 3.10]$
	$\Omega = 10$		
$\alpha = 0.50$	$0.90 : [0.78, 0.99]$	$0.71 : [0.52, 0.88]$	$0.58 : [0.37, 0.86]$
$\beta = 0.30$	$0.25 : [0.18, 0.33]$	$0.26 : [0.20, 0.35]$	$0.28 : [0.23, 0.32]$
$\gamma = 5.02$	$2.82 : [0.92, 4.38]$	$2.72 : [0.95, 3.88]$	$4.32 : [1.20, 6.49]$
	$\Omega = 20$		
$\alpha = 0.50$	$0.93 : [0.76, 1.02]$	$0.75 : [0.50, 0.92]$	$0.62 : [0.31, 0.97]$
$\beta = 0.30$	$0.27 : [0.20, 0.33]$	$0.29 : [0.20, 0.43]$	$0.29 : [0.25, 0.33]$
$\gamma = 10.01$	$5.35 : [0.00, 12.29]$	$3.98 : [0.00, 6.83]$	$7.48 : [0.00, 13.96]$

Table 3.5: Averages and empirical 95% confidence intervals of estimates for $N = 1000$ spikes per train in the critical regime for varying Ω across [1,5,10,20]. Note that the upper subtable corresponds to the third subtable in table 3.3. Numbers differ slightly due to statistical fluctuations in the simulations.

Given a set of N observed spikes and phase angles, $(i_n, \phi_{n-1})_{n=1}^N$, the ML estimates are obtained by considering the log-likelihood function, L,

$$L(\alpha,\beta,\gamma) = \sum_n \log(g_{\phi_{n-1}}(i_n)) = \sum_n \log\left[-\partial_t F^{\phi_{n-1}}_{\alpha,\beta,\gamma}(x_{th},t)\right]\Big|_{t=i_n}. \tag{3.7.1}$$

and maximizing it,$\hat{\alpha},\hat{\beta},\hat{\gamma} = \arg\max L$. Theoretically, one could stop there and start the number crunching. However, for each evaluation of L we would have to solve N PDEs for F. That might become computationally burdensome. Instead, in keeping with the spirit of the chapter, we choose a set of M representative ϕ's $\{\phi_m\}$, chosen as the same phase bin midpoints as in the main text, and approximate $\bar{G}_\phi$ as a weighted average of $\{F^{\phi_m}\}$.

In particular given $\phi \in [\phi_m, \phi_{m+1}]$, we take a simple linear average:

$$\bar{G}_\phi(t) \approx \frac{\phi_{m+1}-\phi}{\phi_{m+1}-\phi_m}\cdot F^{\phi_m}(t) + \frac{\phi-\phi_m}{\phi_{m+1}-\phi_m}\cdot F^{\phi_{m+1}}(t).$$

The basic idea is that if $\phi \approx \phi_m$ we use F^{ϕ_m} and conversely if $\phi \approx \phi_{m+1}$ we use $F^{\phi_{m+1}}$. Note that this assures that $0 \le \bar{G} \le 1$.

Finally, recall that F is solved numerically, using a simple finite difference to estimate the derivative in eq. (3.7.1). If a spike time i_n falls between two time slices t_k, t_{k+1}, $i_n \in [t_k, t_{k+1})$ then we approximate $-\partial_t F(x_{th}, i_n)$ as

$$-\partial_t F(x_{th},i_n) \approx -\frac{F(x_{th},t_{k+1}) - F(x_{th},t_k)}{t_{k+1}-t_k}.$$

Putting it all together, the estimates are obtained as the maximizers of

$$L(\alpha,\beta,\gamma) = \sum_n \log\Bigg(-\frac{\phi_{m+1}-\phi}{\phi_{m+1}-\phi_m}\cdot\frac{F^{\phi_m}_{\alpha,\beta,\gamma}(x_{th},t_{k+1}) - F^{\phi_m}_{\alpha,\beta,\gamma}(x_{th},t_k)}{t_{k+1}-t_k} \\ -\frac{\phi-\phi_m}{\phi_{m+1}-\phi_m}\cdot\frac{F^{\phi_{m+1}}_{\alpha,\beta,\gamma}(x_{th},t_{k+1}) - F^{\phi_{m+1}}_{\alpha,\beta,\gamma}(x_{th},t_k)}{t_{k+1}-t_k}\Bigg)\Bigg|_{\substack{i_n\in[t_k,t_{k+1}]\\ \phi_{n-1}\in[\phi_m,\phi_{m+1}]}}. \tag{3.7.2}$$

after numerically solving the PDE for F, eq. 9 in [5].

To maximize L we again use the Nelder-Mead optimizer from the SciPy library, [60]. We also tried the gradient-based optimizers in SciPy (BFGS, Truncated Newton,

Name	Loss function L	Short description
FP	$\sum_m N_m \Big\{ \sup_t \Big\lvert \hat{\bar{G}}_{\phi_m}(t) - F^{\phi_m}_{\alpha,\beta,\gamma}(x_{th},t) \Big\rvert \Big\}$	Minimize the *sup* of the differences between a numerically calculated survivor distribution and the empirically observed survivor function
ML	$-\sum_n \log\Big(g_\phi(i_n)\Big)$	Maximize the sum of the log-likelihoods of the observed ISIs

Table 3.6: The two estimators we compare: the ML estimator and the FP estimator, which has already been discussed at length in the main text, [5]

Sequential Least Squares), but they turned out slower and provided less accurate estimates.

3.8 ML Estimates

We label estimates based on maximizing eq. (3.7.2) as ML estimates, while estimates obtained by minimizing eq. (3.4.2) are called FP estimates. Of course, both of them rely on solving the Focker-Planck equation, but we do this in order to be consistent with the earlier terminology.

To compare the FP vs. ML estimators, we draw cross-plots between the two in the same style as figs. 12,13 in the main text, [5]. As always, we use 100 spike trains to obtain 100 estimates for α, β, γ using each of the two methods. In all simulations, we set $\Omega = 1$.

We use the same regimes, super-threshold, super-sinusoidal, critical, sub-threshold, with the same parameters as in the main text,

We start by using $N = 100$ spikes per spike train - see fig. 3.17. We use 16 bins for the ML estimate and 8 for the FP. The differences between the two methods appear minor. Neither method seems to be consistently better than the other. In some cases, the ML estimators are noticeably worse - especially for the super-sinusoidal regime, where the ML method shows a bimodal distribution of the α parameter. In other cases, the ML estimator has a smaller variance - see γ in the critical regime and maybe α in the supra-threshold regime, but that is already estimated quite accurately.

Next, we try using the larger sample size, i.e. $N = 1000$ spikes per spike train - see fig. 3.18. Again the differences are minor and similar to what was seen in the smaller sample size, $N = 100$. Most notably the ML is now better at estimating α in the super-sinusoidal regime, except for one rare case.

At this point, we would conclude that the performance of the ML and FP methods is comparable. However the ML method has the potential advantage that one can use much smaller bins (more ϕ_ms) no matter how big or small the data size. This is because the FP method relies on there being a sufficient amount of spikes in each bin in order to approximate the survivor distribution, while the ML method does not have this requirement.

So as a final comparison we redo the estimation for $N = 100$, but now we use $M = 32$ bins for the ML method while still using only $M = 8$ bins for FP. The results are in fig. 3.19. It is evident that the differences between using fig. 3.17 using 16 bins for the ML method and fig. 3.19 using 32 bins are not significant.

As a final note, we mention that the computing times for both methods are roughly the same, modulo that one uses the same number of bins.

As such we conclude that the FP estimates and the ML estimates behave comparably.

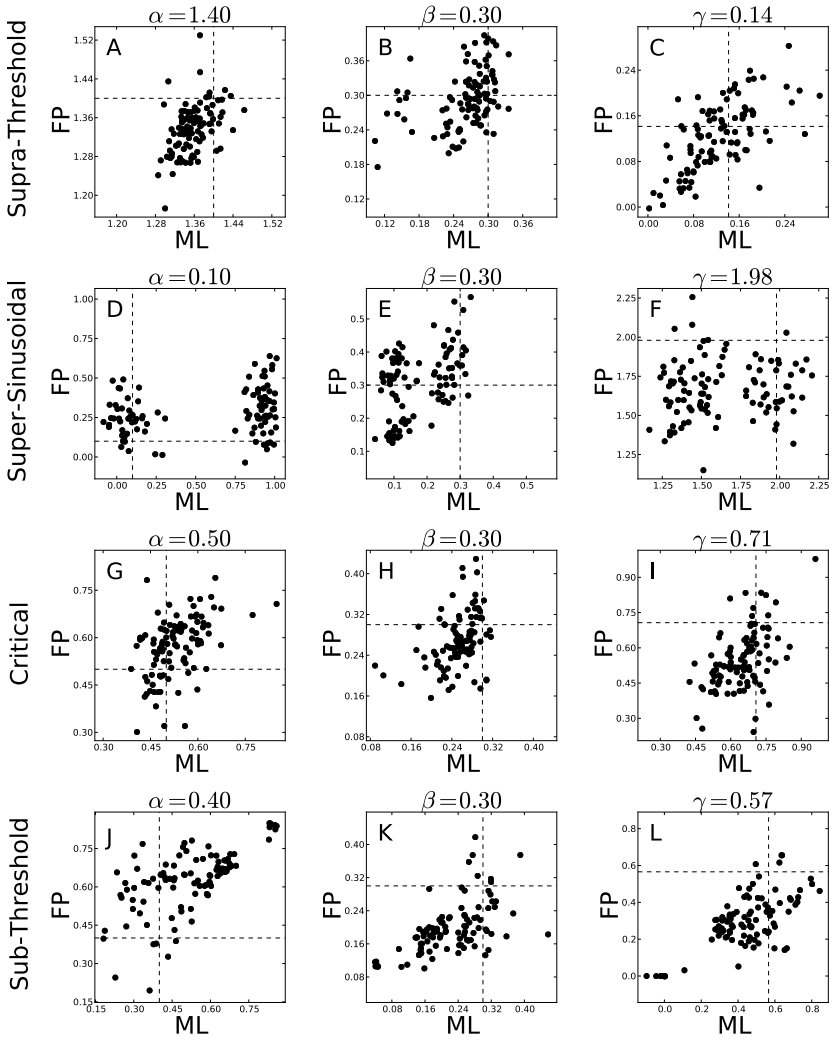


Figure 3.17: Comparison between the FP vs. ML estimators in the four regimes, using $N = 100$ spikes per sample. We used 8 bins for the FP and 16 bins for the ML. In all simulations $\Omega = 1$.

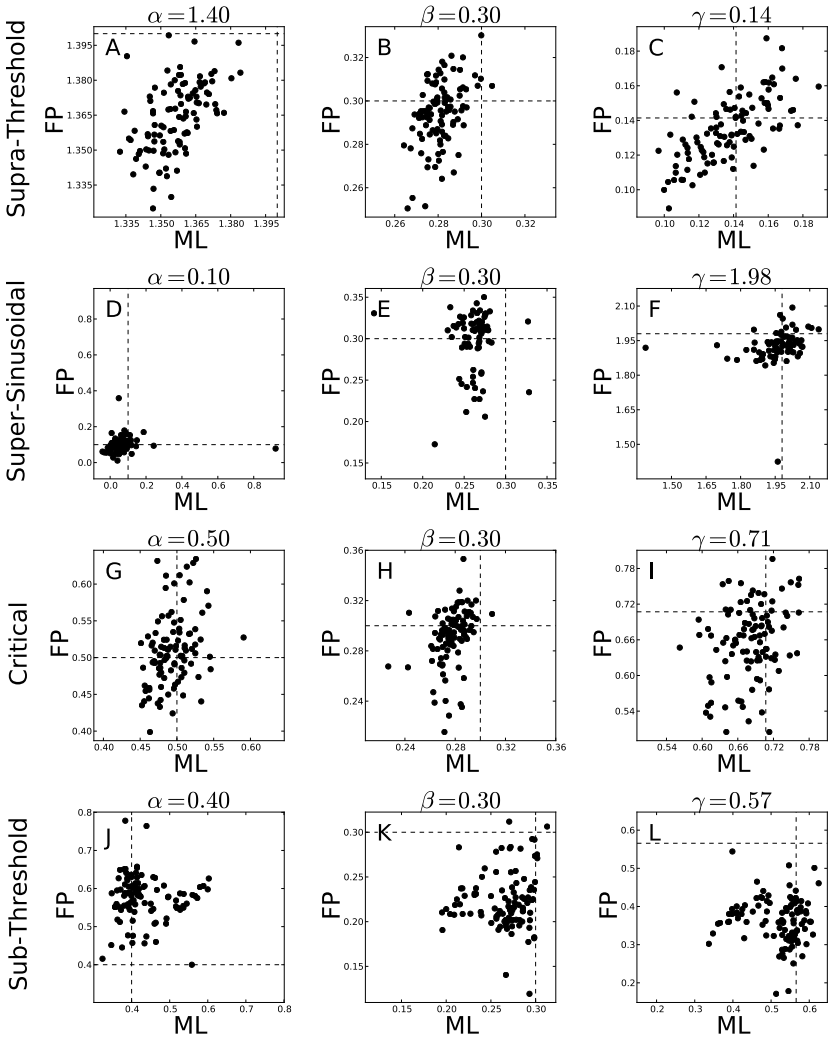


Figure 3.18: Comparison between the FP vs. ML estimators in the four regimes, using $N = 1000$ spikes per sample. We used 8 bins for the FP and 16 bins for the ML. In all simulations $\Omega = 1$.

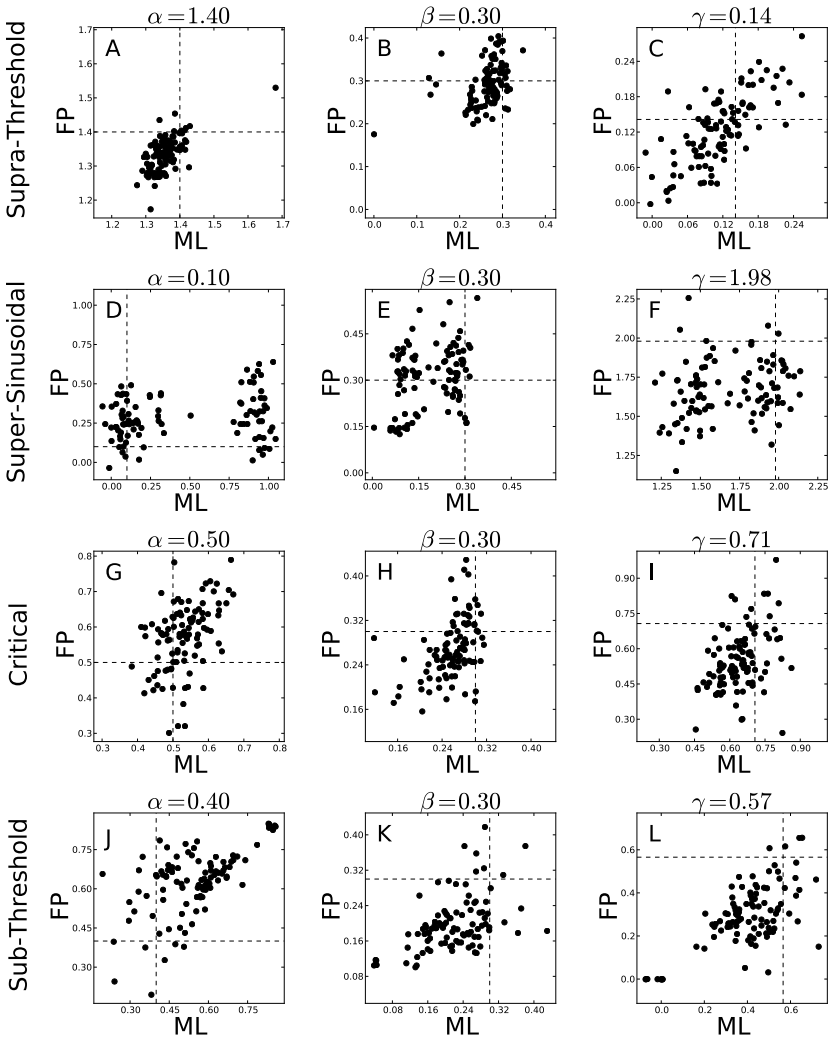


Figure 3.19: Same as fig. 3.17, but using 32 bins for ML instead of 16. $N = 100$. In all simulations $\Omega = 1$.

3.9 Discussion and Outlook

We have shown two methods to estimate parameters in eq. (3.3.2) from ISI data. Our methods are based on binning the spikes into bins with representative phase shifts. We have devised a constructive procedure to automatically initialize the methods from the data.

Our computational results suggest that for low frequencies the Fortet algorithm is superior for large sample sizes, especially in the super-sinusoidal regime, while the Fokker-Planck algorithm has a comparable accuracy and a lower variance for small sample sizes. Both algorithms find sensible estimates most of the time, although they seem less effective in the sub-threshold regime. Their performance can be partially attributed to the ability of the initializer algorithm to supply good guesses for starting the optimization iterations.

The Fokker-Planck equation allows for approximate maximum likelihood estimation. We chose an alternative loss function, though, because it marginally appeared more robust, possibly because a numerical derivation step is avoided. This is further investigated by simulations in the supplementary online material. The simulations suggest that the distribution of the maximum likelihood estimates in the super-sinusoidal regime appears bimodal, which is not occurring for the alternative loss function, eq. (3.4.2).

We have also made a preliminary exploration of the effect of Ω on the quality of the estimates. Our results show that an increase in Ω makes the parameters α and γ more difficult to estimate accurately and at high Ω, γ is underestimated, while α is over-estimated. We find that in this scenario, the Fortet algorithm does a markedly more accurate job then the Fokker-Planck algorithm.

We have assumed the time-constant τ of the leak term known. In most experiments that is not realistic, and it would be preferable to estimate τ alongside the other parameters. However, it is difficult to estimate [62]. When we tried to estimate

it together with the other parameters, we usually obtained results which were not accurate. The obtained estimates resulted in ISIs that very well matched the data, no worse than the ISIs obtained from the true parameters. This leads us to believe that the simultaneous estimation of τ along with α, β, γ using only ISI data suffers from identifiability problems. In [37], they were able to estimate τ in the simpler non-sinusoidally-driven model, but concluded that adding τ as an unknown dramatically reduced the accuracy in the estimation of the other unknown parameters. The reason is that if τ is also estimated from a single dataset alongside the other parameters, then a reasonable fit can be found to the data for various combinations of α, β, γ and τ, but the so-obtained parameter values can be far from the true values.

Our model is relatively simple and ignores neurophysiological realism, such as the fact that the spiking threshold is likely non-constant, with a time-dependent functional form that would involve further unknown parameters. A recent paper attempting the parameter estimation in such a model, but without sinusoidal forcing, is [52]. Furthermore, intra-cellular recordings suggest that a hard threshold is a rough approximation and an exponential voltage-dependent spiking intensity is more realistic [63].

While our work has used a very specific form of the periodic forcing term, namely $\gamma \sin(\Omega t)$, it is clear how to apply the approach to an arbitrary periodic function. This can be done as long as one knows where in the period of oscillation a spike has occurred. If that is the case then the binning procedure can be applied and the estimation methods proposed can be attempted.

Implementation Details

All the code used in this chapter including code to generate the figures can be found on github @ https://github.com/aviolov/SinSpikePython. Please see the README.md file for guide to the codes.

Chapter 4

Optimal Control of Single Spikes

4.1 Thesis Context

In the second of the three main chapters of the thesis, we discuss the control of spike timing in a single neuron. Depending on the ability to observe the detailed voltage or only the spike timing, we derive a Hamilton-Jacobi-Bellman equation in the first case and an adjoint-based Variational Principle for the latter. We show how in both cases the boundary conditions and the terminal conditions in the optimization PDEs are naturally obtained from the problem formulation. The optimal control is obtained by numerically solving the PDEs. In simulations, we demonstrate the effectiveness of the technique for different system regimes, i.e. parameters. We then demonstrate how to estimate system parameters of the noisy LIF model given observations from a more complicated mathematical neural model, the Morris-Lecar model. Finally, we simulate the performance on the control derived for the noisy LIF model on the more complicated model and thus demonstrate the robustness of the control scheme to model error.

4.2 Problem Introduction

Manipulation of individual neurones through electrical stimulation provides a mean of controlling their spiking activity. In applications such as brain-machine interfaces and neuroprosthetics, a common goal is to record from a neurone and interpret the firing activity. Conversely, one may wish to stimulate a cell in order that it produce a desired firing rate, either fixed or varying in time. It is known that many cells fire sequences of spikes where the spike times - rather than just the rate - matter to post-synaptic neurons. In this chapter we explore the possibility of controlling a neuron in a way that it generates a sequence of spike times close to a desired sequence. We consider this problem in the framework of Stochastic Optimal Control and give both a feedback solution (closed-loop), when the cell voltage is explicitly observable, as well as an open-loop solution when only the occurrence of spikes is observable. Importantly, we allow for an arbitrary noise intensity.

Both theoretically and in practice related problems have been addressed in the literature. One objective has been to obtain either minimum or maximum interspike intervals lengths, when the input is constrained to be between some prespecified upper and lower bounds, see e.g. [64, 65] for a mathematical treatment, or [66, 67, 68] in a neuronal context. Another objective is to break a pathological synchronous firing pattern in clusters of neurons, highly relevant for neurological disorders such as epilepsy and Parkinson's disease, [69, 70], see also [71].

Our objective, namely targeting exact spike times in single neurons, has been considered mainly in the open-loop context, and in either absence of or for small noise. In [72] they use the Spike Response Model, [73], to control output target spike trains and implement their scheme on pyramidal cells in mouse cortical slices. Their method is numerically efficient and allows for the simultaneous control of many neurons. However, it strongly relies on the assumption that the noise (the value of β in eq. (4.3.1) below) is small. They only work with open-loop control. The difference

between the objective in [72] and what we consider below is that they maximize the probability of spiking at some given time t^*, whereas we minimize the mean squared difference between the realized spike time, T_{sp} and the desired, t^*. Moehlis et al. [74] work with a Phase Response Model to obtain spikes at exact times, while keeping the root mean square of the input to a minimum. A similar approach is taken in [75]. They do not consider noise, though, and only work with open-loop control. In [76], their methods are implemented on brain slices of pyramidal neurons of rat hippocampus. While the Phase Response Curve Model is a parsimonious and effective way to describe a neuron's response to a stimulus, it is only valid in the supra-threshold regime, where the unstimulated neuron is periodically spiking. Reference [77] investigates the control of the firing times of a leaky integrate-and-fire neuron by varying the intensity of a noise process that drives the voltage (there is no other intrinsic noise in the model). To obtain a given spike time, they choose the objective of minimizing the variance of the membrane potential at the desired spike time, while forcing the mean of the membrane potential at this time point to equal the threshold. This provides exact solutions since it does not involve first-passage times, but has the drawback that there is a non-negligible probability that the obtained spike time will be far from the desired spike time.

Our objective of imposing a certain timing sequence for the spike train using an externally applied control is obtained in both the closed- and open-loop settings, and we specifically include the noise in the calculations of the controls. We restrict the controlled input to stay within pre-specified bounds, and also include a cost function to minimize intervention. Our main contributions are that we specifically allow for a non-negligible noise component in the neural activity when calculating the control, and consider both open- and closed-loop controls. The noise component is given by a Wiener process with a noise intensity $\beta \gg 0$. In particular we do not restrict our attention to the autonomously-spiking, supra-threshold regime.

The chapter is structured as follows: First we describe the neuron model and

formalize the control objective. Then we describe a feedback-based solution, which assumes that the controller has detailed access to the voltage trajectory. Then we relax the observation assumption so that the controller only has access to the spike times. Finally, we compare the two methods through simulations against a simple-minded control technique which ignores the stochastic input to the neuron.

4.3 Problem Formulation

A basic but useful model for the neural membrane potential evolution is the noisy leaky-integrate-and-fire (LIF) model:

$$\begin{aligned} dX(t) &= \left(I_{\text{ext}}(t) - \frac{X(t)}{\tau_c} \right) \mathrm{d}t + \beta \, \mathrm{d}W, \\ X(0) &= 0, \\ X(T_{\text{sp}}) = x_{th} \implies & \Big\{ X(T_{\text{sp}}{}^{+}) = 0. \end{aligned} \tag{4.3.1}$$

Here $X(t)$ represents the membrane electric potential at time t, which in absence of input decays to 0 with a time constant of τ_c, dW is a Brownian motion increment scaled by β and $I_{\text{ext}}(t)$ is the deterministic external input to the cell. Having last spiked at time 0, the potential hits x_{th} at some random time T_{sp}, the potential resets to 0 and the process starts all over again. We write $T_{\text{sp}}{}^{+}$ for the limit from the right at T_{sp}. Throughout the chapter the threshold is set to one, $x_{th} = 1$.

Suppose that we have some control over the external current such that it can be decomposed as

$$I_{\text{ext}}(t) = \mu + \alpha(t), \tag{4.3.2}$$

where μ is an uncontrollable, but constant part of the external current and $\alpha(t)$ is controllable, i.e., it can be chosen to achieve some goal. A natural goal is to attempt to control the spike time, T_{sp}. That is, how do we choose $\alpha(t)$ such that $T_{\text{sp}} \approx t^*$, where t^* is the desired spike-time. A natural optimal control objective to achieve this

is the least squares solution

$$\alpha(\cdot) = \arg\min_{\alpha(\cdot)} \{\, \mathbb{E}[(T_{\text{sp}} - t^*)^2] \,\}, \tag{4.3.3}$$

where expectation is taken with respect to the distribution of the trajectories of X.

Often the control has certain constraints. The most common are simple box constraints:

$$\alpha(t) \in [\alpha_{\min}, \alpha_{\max}] \quad \forall t. \tag{4.3.4}$$

In addition to eq. (4.3.3), we will also add to our objective a running energy cost based on the control. This regularizes the problem eliminating the subtleties of singular-control situations and it serves to avoid excessive control as well as to avoid excessive charge building up on the cell, see [72].

So we seek an optimal control, α^*, that solves

$$\begin{aligned} J[\alpha(\cdot)] &= \mathbb{E}\left[\epsilon \int_0^{T_{\text{sp}}} \alpha^2(s)\,\mathrm{d}s + \left(T_{\text{sp}} - t^*\right)^2\right] \\ \alpha^*(\cdot) &= \arg\min_{\alpha(\cdot)} J[\alpha(\cdot)], \end{aligned} \tag{4.3.5}$$

where ϵ measures how much weight we put on minimizing the energy cost. If $\epsilon = 0$, then we do not care at all about the expended energy cost.

It will often be the case that α is either a function or a value of that function at a particular time. This function could be random, i.e., a stochastic process, which is naturally the case when it is a function of the random realizations of X. We will try to make clear below when we are refering to α as a function and when we are merely referring to its particular value at some particular time. For example, in eq. (4.3.5), $\alpha(\cdot)$ refers to the function, while $\alpha(s)$ refers to that function's value, possibly random, at time s.

We will consider two control contexts – *closed-loop* and *open-loop* control. In closed-loop control, the value of X at time t is observable and can be used in determining the control. In open-loop control only the spike times are observable. In the

closed-loop context, we write the control as

$$\alpha = \alpha(x, t)$$

to indicate its dependence on $X(t)$ and to express that it will be updated based on the time-course of X. In the open-loop context, we write

$$\alpha = \alpha(t)$$

to indicate that $\alpha(\cdot)$ is decided for all times at time 0. The techniques used to obtain the optimal controls in the two scenarios will be different. For the closed-loop scenario we use Dynamic Programing, see [10], while for the open-loop scenario we use a form of the Maximum Principle applied to the transition density of the controlled process, see [28]. The transition density is the probability density function of the process being at a state y at time t, given it was at some state x at some earlier time s.

Crucially, we assume that the model parameters, μ, τ_c, β in eqs. (4.3.1) and (4.3.2) are known. This is a strong assumption and will be discussed later.

4.3.1 Parameter Regimes

Different parameter regimes can be envisioned given eq. (4.3.1), depending on whether the noise intensity, β, is relatively high or low, and whether the external, uncontrollable bias current, μ, induces spikes in the absence of noise or not. Spikes will occur in the absence of noise, if and only if $\mu > 1/\tau_c$, which is called the supra-threshold regime. When $\mu \leq 1/\tau_c$, the regime is called the sub-threshold regime. In addition we will investigate two values of β, which we will call high-noise and low-noise, respectively. Example values for each parameter regime are given in table 4.1, and we visualize a single path for each regime in fig. 4.1.

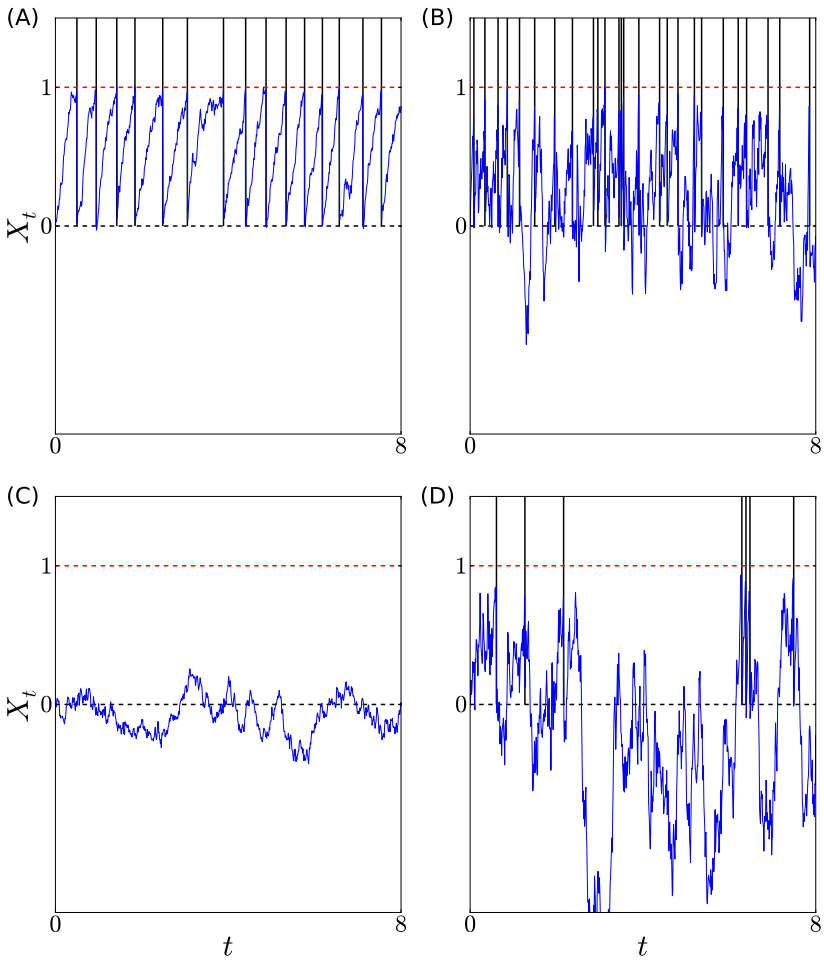


Figure 4.1: Example Trajectories from eq. (4.3.1) using the parameter values from table 4.1. A) Supra-Threshold-Low-Noise B) Supra-Threshold-High-Noise C) Sub-Threshold-Low-Noise D) Sub-Threshold-High-Noise. Note the multiple crossings very close together in the high-noise regimes in B and D.

$\mu\beta$	1.5	0.3
$1.5/\tau_c$	Supra-threshold-High-noise	Supra-threshold-Low-noise
$0.1/\tau_c$	Sub-threshold-High-noise	Sub-threshold-Low-noise

Table 4.1: Regime labels and example values. Note that for the numerical experiments below, we use $\tau_c = 0.5$

4.4 Closed-Loop Solution - Dynamic Programing

We now detail the Dynamic Programing approach to obtaining an optimal feedback control. In closed-loop, the controller can be continuously updated depending on the realization of the stochastic process, X.

Given a time t and a value $X(t) = x$ of the voltage, let $(T_{\text{sp}} - t)$ be the unknown and remaining time to spike. Note that T_{sp} is a random variable. Given arbitrary t, x, our remaining-cost objective, $J[\alpha(\cdot); x, t]$, will be

$$J[\alpha(\cdot); x, t] = \mathbb{E}\left[\left((T_{\text{sp}} - t) - (t^* - t)\right)^2 + \epsilon \int_t^{T_{\text{sp}}} \alpha^2(s)\,\mathrm{d}s \,\middle|\, X(t) = x\right]. \tag{4.4.1}$$

That is, if time t has elapsed without a spike, we now want to minimize the difference between $(T_{\text{sp}} - t)$ and $(t^* - t)$, given the current state x. The mean in eq. (4.4.1) is taken over the distribution of hitting times, T_{sp}, conditional on $X(t)$ or equivalently over the distribution of forward trajectories of X starting at x and ending at the threshold. Recall that in the closed-loop scenario, we assume that the value of $X(t)$ is known to the controller. A similar problem is discussed analytically at length in the book on optimal control by Whittle, [78], although there is no discussion there of the numerics required to solve it.

4.4.1 Hamilton-Jacobi-Bellman equation

The Hamilton-Jacobi-Bellman (HJB) equation, see [10, 78] or the articles [66, 69] in a neuroscience context, associated with the optimal control for eq. (4.4.1) is obtained as

follows. We introduce the value function, $w(x,t)$, as the minimum of the remaining-cost objective, i.e., of the cost function between the current time t and the desired spike time t^*:

$$\begin{aligned} w(x,t) &= \min_{\substack{\alpha(\cdot)_{s\geq t} \\ \alpha_{\min}\leq\alpha(\cdot)\leq\alpha_{\max}}} \{J[\alpha(\cdot);x,t]\} \\ &= \min_{\substack{\alpha(\cdot)_{s\geq t} \\ \alpha_{\min}\leq\alpha(\cdot)\leq\alpha_{\max}}} \mathbb{E}\left[\left((T_{\text{sp}}-t)-(t^*-t)\right)^2+\epsilon\int_t^{T_{\text{sp}}}\alpha^2(s)\,\mathrm{d}s\,\middle|\,X(t)=x\right]. \end{aligned} \tag{4.4.2}$$

Then w satisfies the following HJB partial differential equation (PDE):

$$\begin{gathered} \partial_t w(x,t)+\tfrac{\beta^2}{2}\partial_x^2 w(x,t)+ \\ \min_{\alpha(x,t)\in[\alpha_{\min},\alpha_{\max}]}\left\{\epsilon\alpha^2(x,t)+\left(\mu+\alpha(x,t)-\tfrac{x}{\tau_c}\right)\partial_x w(x,t)\right\}=0. \end{gathered} \tag{4.4.3}$$

The special feature of eq. (4.4.3) in contrast to a generic parabolic PDE is that it contains an embedded optimization that depends on the solution, w. For each x,t in the computational domain, α is chosen such as to minimize $\{\epsilon\alpha^2+\left(\mu+\alpha-\frac{x}{\tau_c}\right)\partial_x w\}$. Here we can solve for the optimal control, $\alpha^*(x,t)$ analytically as:

$$\begin{aligned} \alpha^*(x,t) &= \operatorname*{arg\,min}_{\alpha\in[\alpha_{\min},\alpha_{\max}]}\left\{\epsilon\alpha^2+(\mu+\alpha-\tfrac{x}{\tau_c})\partial_x w\right\} \\ &= \min\left(\alpha_{\max},\max\left(\alpha_{\min},-\frac{\partial_x w(x,t)}{2\epsilon}\right)\right). \end{aligned} \tag{4.4.4}$$

We need to consider boundary conditions (BCs) for w. If $X(t)=x_{th}$ then we have a spike now and $T_{\text{sp}}=t$. Thus

$$w(x_{th},t)=(t-t^*)^2.$$

At the threshold, the value function equals the squared difference between the desired spike time and the realized one.

For large, negative values of x, we assume that w is not significantly affected by the change in x, i.e., that

$$\partial_x w(x_-,t)=0$$

for some lower boundary x_-. Such a boundary condition will be justified if we choose x_- such that the probability for the process to take values smaller than x_- is small. For example, we can take x_- to be two standard deviations below the mean of the stationary distribution of the maximally inhibited process, i.e., setting $\alpha = \alpha_{\min}$ in eq. (4.3.2). That is, we set $x_- = \tau_c(\mu + \alpha_{\min}) - 2\beta/\sqrt{\tau_c/2}$. We further enforce that $x_- \leq -0.5$.

Note that Dynamic Programing and the HJB equation work backwards. Thus the evolution of the value function proceeds from the future to the past and we need some Terminal Condition (TCs) at some point in the future, possibly infinity, from which to start incrementing w using the dynamics and the BCs. To determine TCs for w, our idea is simple: if we reach t^* without having spiked we apply maximum control in the positive direction, i.e.,

$$t > t^* \implies \alpha(t) = \alpha_{\max}.$$

Thus:

$$w(x, t^*) = \mathbb{E}\Big[(T_{\text{sp}} - t^*)^2 \,\Big|\, X(t^*) = x, \alpha(t) = \alpha_{\max}\Big]. \tag{4.4.5}$$

Note that we are making an approximation here - we are ignoring the energy term, ϵu^2, in the objective for $t > t^*$. Naturally, this approximation is ever more accurate for $\epsilon \ll 1$. We discuss in more detail the validity of the approximation in section 4.7. Alternatively, we could impose this approximate terminal condition at some $t^+ > t^*$.

We will see the quantity on the right hand side of eq. (4.4.5) repeatedly so we will give it a special name.

$$T_{(2)}(x) := \mathbb{E}\Big[(T_{\text{sp}} - t^*)^2 \,\Big|\, X(t^*) = x, \alpha(t) = \alpha_{\max}\Big]. \tag{4.4.6}$$

$T_{(2)}$ is the second moment of the remaining time to reach the threshold starting at $X(t^*) = x$ and applying $\alpha_{\max}$ throughout. This quantity can be found easily for all x in the domain by solving a stationary backward Kolmogorov equation. This is an

extension to the calculation of the first moment of an exit-time as seen in textbooks such as [13] and we give the details in section 4.9.

Thus, we restate the HJB equation in its fully specified form:

$$\begin{aligned}
&\partial_t w(x,t) + \tfrac{\beta^2}{2}\partial_x^2 w(x,t) + \epsilon\alpha^2(x,t) + \left(\mu + \alpha(x,t) - \tfrac{x}{\tau_c}\right)\cdot \partial_x w(x,t) = 0,\\
&\alpha(x,t) = \min\left(\alpha_{\max}, \max\left(\alpha_{\min}, -\frac{\partial_x w(x,t)}{2\epsilon}\right)\right).\\
&\begin{cases} w(x_{th},t) = (t-t^*)^2 & \text{upper BC}\\ \partial_x w(x_-,t) = 0 & \text{lower BC}\\ w(x,t^*) = T_{(2)}(x) & \text{TC}\end{cases}
\end{aligned} \tag{4.4.7}$$

We are solving $w(x,t)$ over the domain $[x_-, x_{th}] \times [0, t^*]$.

4.4.2 The numerical method for the HJB equation

We now have a PDE for w and an algorithm for computing all the BCs and TCs of this PDE. It is time to discuss the numerical method for solving eq. (4.4.7). Since it is one-dimensional in space, it is straight forward to apply the standard centred finite difference using the Crank-Nicholson scheme to step in time, see ch. 19.2 in [32]. To resolve the non-linearity, $(\partial_x w)^2$, in the PDE, we treat it as a mixed implicit-explicit term

$$(\partial_x w(x,t_k))^2 = \partial_x w(x,t_k)\cdot\partial_x w(x,t_k) \approx \underbrace{\partial_x w(x,t_k)}_{\text{implicit}}\cdot\underbrace{\partial_x w(x,t_{k+1})}_{\text{explicit}}.$$

Note that the implicit term is in the previous time t_k instead of, as is conventional, the next t_{k+1}, because we are solving for w *backwards* in time from t_{k+1} to t_k down to $t_0 = 0$.

The numerical scheme is implemented in Python, using the Scipy/Numpy library, [60]. For the discretization in time and space, we choose Δx and Δt in relation to the parameter regimes. In particular, we take Δx to be less than β divided by the largest possible absolute value of $(\mu + \alpha - x/\tau_c)$ for $\alpha \in [\alpha_{\min}, \alpha_{\max}], x \in [x_-, x_{th}]$; in

our examples this always equals $\mu + \alpha_{\max} - x_-/\tau_c$. This is an attempt to ensure that numerically we are in the diffusion-dominated regime. We set Δt to be twice the value of Δx divided by the largest possible absolute value of $(\mu + \alpha - x/\tau_c)$. For example in the Supra-Threshold Low-Noise regime, we have $\Delta x = 0.0043$ and $\Delta t = 0.0013$. In all regimes, our discretization results in point-wise convergence of up to at least three significant digits for the value function. We consider this to be a sufficiently fine discretization.

We next demonstrate solutions for the value function and the associated control law for a parameter set from each regime in table 4.1.

4.4.3 Solutions of the HJB equation

We set $t^* = 1.5$ and $\epsilon = 0.001$, so that the primary focus is the accurate spiking and energy minimization considerations are secondary.

In fig. 4.2 we show the computed value function, $w(x,t)$, the solution to eq. (4.4.7), for each of the parameter regimes; while in fig. 4.3, we show the corresponding surface plots of $\alpha(x,t)$.

Let us discuss the most salient features in figs. 4.2 and 4.3. Recall that lower values for the value function, $w(x,t)$, are preferred, since we are minimizing. We see the same general shape in all four space-time surface plots of the value function in fig. 4.2. At the end of the interval, $w(x,t^*)$ is monotonically decreasing in x, which is to be expected given its terminal condition in eq. (4.4.5). That is, the lower the value of $X(t^*)$ the more on average will we have to wait for the spike to occur. As we go back in time w inverts, with a clear peak near the upper, threshold boundary. That is, for intermediate values of t we have to consider the risk of spiking too early, represented by the high value of w near the upper boundary *and* the risk of spiking too late, represented by the high values of w at the lower boundary. As we progress even further back in time to the beginning of the interval, the peak near the threshold

rises further, since we are spiking *earlier*, while the peak at the lower end flattens, since now there is enough time to reach threshold despite starting far from it. The full surface plots for $w(x,t)$ in fig. 4.2 can be thought of as the interpolation between these three basic phases.

Now focus on the controls in fig. 4.3. Naturally, at the end of the interval, the control takes its maximum value, $\alpha(x,t^*) = \alpha_{\max}$ for all x, i.e., it gives the maximum available push for the neuron to spike. This is consistent with eq. (4.4.5). As we go back in time, however, the control α decreases for $x \ll x_{th}$, and it becomes negative, i.e., inhibitory for $x \lesssim x_{th}$. That is intuitively consistent with the problem objective. For $t < t^*$, we want to bring $X(t)$ closer to the threshold, but not too close, to avoid early spiking.

Now consider the differences between the individual regimes, i.e., the effect of changing the bias and the noise intensity, μ, β. All things being equal, the effect of increasing the noise intensity, β, is to lift the value function, i.e., to make our objective worse, at the beginning of the interval and to decrease it at the end. This is to be expected since we are attempting to minimize the variance in the spike time and without noise there would be no variance at all. However, at the end of the interval, noise only helps, since now we are only interested in spiking as soon as possible, and a higher noise intensity will tend to decrease the spike time on average. Increasing β also has the effect of increasing the size of the boundary layer near x_{th}, where the value function rises steeply. That is, for small β that layer is small since the risk of spiking early is significant only close to the threshold. For larger β this layer increases, see panels A,C, with a thin layer, vs. B,D, with a thicker layer, in fig. 4.2. Similarly, increasing the bias, μ, tends to decrease the value function, especially at the end since that has the unequivocal effect of preventing late spikes.

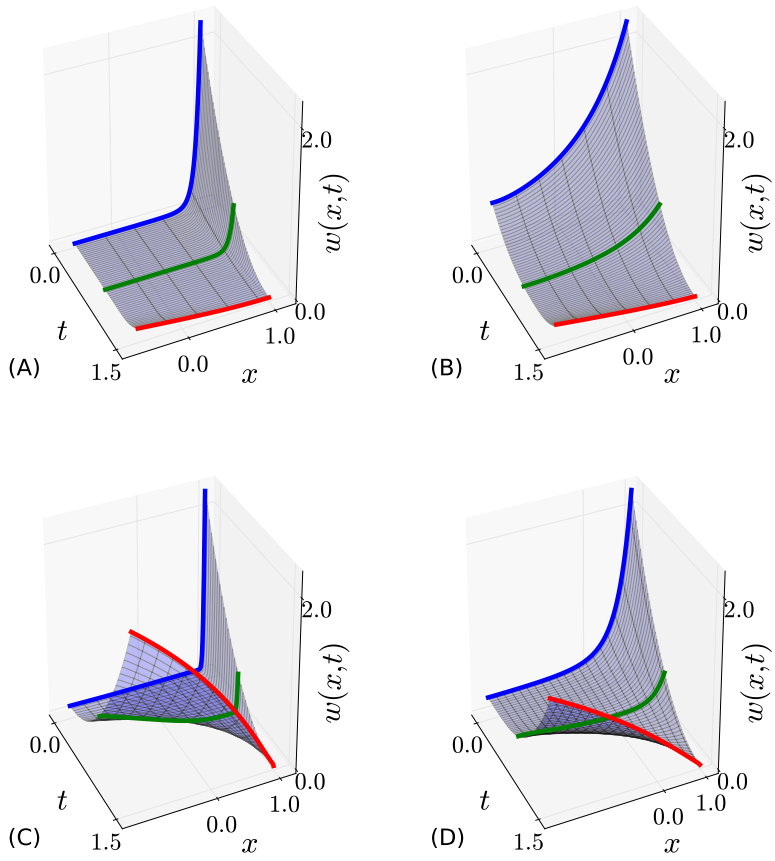


Figure 4.2: The numerical solution for the value function $w(x,t)$ HJB PDE, eq. (4.4.7), for the four different parameter regimes. The desired spike time is set to $t^* = 1.5$. The control bounds are $\alpha \in [-2, 2]$. The coloured contours for fixed t plotted in thick correspond to the beginning, middle and end times of the interval (blue: $t = 0$, green: $t = t^*/2$, red: $t = t^*$). The energy penalty is $\epsilon = .001$. Note that all red curves for the value function go to zero at the threshold to satisfy the upper BC in eq. (4.4.7).

A) Supra-Threshold-Low-Noise B) Supra-Threshold-High-Noise C) Sub-Threshold-Low-Noise D) Sub-Threshold-High-Noise

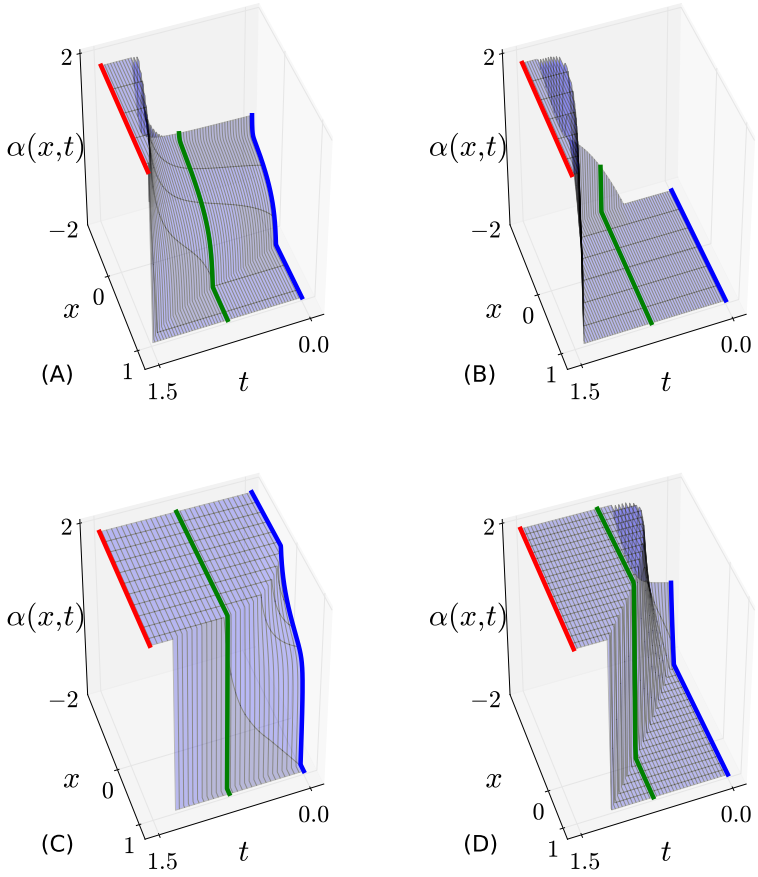


Figure 4.3: Surface plots for $\alpha(x,t)$ analogous to the surface plots for $v(x,t)$ in fig. 4.2. Note that the aspect of this figure is rotated relative to the surface plots of $w(x,t)$ in fig. 4.2 because the different surfaces are best seen from different angles.

4.5 Open-loop Stochastic Control

When the value of $X(t)$ is unobservable, the transition density can be used to perform the optimization. Since the transition density follows a deterministic PDE, we apply a Maximum Principle for PDEs as a method of obtaining the optimal control, see [28] for details on optimization with PDEs, or the article, [24], for optimization with a Fokker-Planck-type of system.

4.5.1 Fokker-Planck equation for the state density evolution

We write the transition density of X, conditional on no spikes having occurred since $t = 0$ as:

$$f(x,t)\,\mathrm{d}x = \mathbb{P}[X(t) \in \mathrm{d}x \,\big|\, X(0) = 0,\, X(s) < x_{th} \quad \forall s < t].$$

Then f satisfies a Fokker-Planck equation, see Ch. 7 in [13],

$$\partial_t f(x,t) = \frac{\beta^2}{2} \cdot \partial_x^2 f - \partial_x \Big[\left(\mu + \alpha(t) - \frac{x}{\tau_c} \right) \cdot f \Big].$$

$$\begin{cases} f(x,0) & = & \delta(x) \quad \text{delta function} \\ (\mu + \alpha(t) - \frac{x}{\tau_c}) f - \partial_x \frac{\beta^2}{2} f] \big|_{x=x_-} & \equiv & 0 \quad \text{lower BCs at some } x_- \\ f(x,t)|_{x=x_{th}} & \equiv & 0 \quad \text{upper BCs at } x_{th} \end{cases} \tag{4.5.1}$$

In theory, $x_- = -\infty$, but in the numerics below we will need to truncate it to some finite value, exactly as in the HJB equation, eq. (4.4.7). Note that we write $\alpha(t)$ here instead of $\alpha(x,t)$ since we cannot use the value of $X(t)$.

The f dynamics can also be written as

$$\partial_t f(x,t) = -\partial_x \phi(x,t)$$

for the probability flux

$$\phi(x,t) = (\mu + \alpha(t) - \tfrac{x}{\tau_c}) f - \partial_x [\tfrac{\beta^2}{2} f].$$

Then the lower BC is

$$\phi(x,t)|_{x=x_-} \equiv 0.$$

We will also need a short hand notation for the differential operator on the right side of eq. (4.5.1). Let

$$\mathcal{L}_\alpha[\cdot] := \partial_x^2\Big[\frac{\beta^2}{2}\cdot\Big] - \partial_x\Big[(\mu+\alpha(t)-\frac{x}{\tau_c})\cdot\Big],$$

where $[\cdot]$ indicates the argument of the operator $\mathcal{L}_\alpha$. We will usually omit the subscript α, but have written it now to emphasize that the differential operator is parametrized by the control, α.

4.5.2 Restating the objective in terms of the transition density

We now derive the optimality conditions for the optimal control α^* for the open-loop context. Recall our objective, eq. (4.3.3). We can write this in terms of the transition density, f, as:

$$\begin{aligned} J[\alpha(\cdot)] = &\int_{x_-}^{x_{th}} T_{(2)}(x)\cdot f(x,t^*)\,\mathrm{d}x \\ &+ \int_0^{t^*} \phi(x_{th},t)(t-t^*)^2\,\mathrm{d}t \\ &+ \epsilon\int_0^{t^*} \alpha^2(t)\,\mathrm{d}t. \end{aligned} \tag{4.5.2}$$

Let us explain in more detail each term on the right-hand side in eq. (4.5.2).

The first term,

$$\int_{x_-}^{x_{th}} T_{(2)}(x)\cdot f(x,t^*)\,\mathrm{d}x,$$

counts the cost of trajectories which spike too late. This cost is the expected squared time-to-hit starting at x, with $\alpha = \alpha_{\max}$, weighted by the probability of $X(t^*) = x$, which is just $f(x,t^*)$. Recall that $T_{(2)}$ is defined in eq. (4.4.6).

The second term,

$$\int_0^{t^*} \phi(x_{th}, t)(t - t^*)^2 \, \mathrm{d}t,$$

counts the cost of trajectories which spike too early, that is the squared difference between some realized spike time, $T_{\mathrm{sp}} = t$, and desired spike time t^*, weighted by the probability of a spike at t which is just the outward probability flux at the threshold, $\phi(x_{th}, t)$. Recall that we assume that $x_{th} = 1$ throughout. Note further that due to the homogeneous Dirichlet BC at x_{th}, $f(x_{th}) = 0$, the outward flux is simply

$$\phi(x_{th}, t) = -\frac{\beta^2}{2} \partial_x f(x_{th}, t).$$

Finally, the third term,

$$\epsilon \int_0^{t^*} \alpha^2(t) \, \mathrm{d}t,$$

is the energy cost; note that in this context the control is deterministic and thus we do not need to take its expectation.

With that our optimal control $\alpha^*(\cdot)$ will naturally be found via:

$$\alpha^*(\cdot) = \arg\min_{\alpha(\cdot)} J[\alpha(\cdot)].$$

4.5.3 Optimizing using a Maximum Principle

By now our stochastic optimal control problem modelled by SDEs has become a deterministic optimal control problem modelled by PDEs. Our control α influences the evolution density f and via f, the integrals in the objective, J.

The Maximum Principle for PDEs, which is an extension to the famous Pontryagin Maximum Principle from finite dimensional systems, introduces an adjoint variable, p, which solves a PDE related to the PDE satisfied by the density f and then calculates the optimal control, $\alpha(\cdot)$, as a functional of f and p.

In short, the equation for the adjoint function, p, is

$$\begin{aligned}\partial_t p &= -\mathcal{L}^*[p] \\ &= -\Big[\frac{\beta^2}{2}\cdot\partial_x^2 p + (\mu+\alpha(t)-\frac{x}{\tau_c})\cdot\partial_x p\Big]. \\ &\begin{cases} p\big|_{x=x_{th}} & = (t-t^*)^2 \\ \partial_x p\big|_{x=x_-} & = 0 \\ p(x,t^*) & = T_{(2)}(x) \end{cases}\end{aligned} \tag{4.5.3}$$

and then α can be found via

$$\Big\{\epsilon 2\alpha(t) + pf\Big|_{x_-} - \int_{x_-}^{x_{th}} p\cdot\partial_x f\,\mathrm{d}x\Big\} = 0 \quad \forall t\in[0,T]. \tag{4.5.4}$$

Roughly speaking, eq. (4.5.4) corresponds to setting the derivative of the objective J with respect to α to zero, in eq. (4.5.2) and the adjoint state p acts as a Lagrange multiplier corresponding to the constraints of the transition density dynamics.

More practically, the quantity

$$\delta_\alpha J(t) = \Big\{ -\int_{x_-}^{x_{th}} p(x,t)\cdot\partial_x f(x,t)\,\mathrm{d}x + p(x,t)f(x,t)\Big|_{x_-} + \epsilon 2\alpha(t)\Big\} \tag{4.5.5}$$

gives the direction of increase of J at $\alpha(t)$ and we can use it as a gradient in a descent algorithm, given some initial guess, $\alpha_0(t)$, for the control.

Finally, we are ready to calculate the open-loop stochastic optimal control for the four parameter regimes. We use the same numerical method for solving the PDEs as described in section 4.4.2, since the PDEs for f, p are very similar in structure to the PDE for w except they do not contain a nonlinear term. The details of the simple gradient descent algorithm for obtaining the optimal open-loop control are given in algorithm 1. See [25] for a more sophisticated descent method based on the conjugate-gradient method. For our applications, the algorithm in 1 typically converges in less than 10 iterations, although sometimes it can take longer.

The optimal controls obtained using algorithm 1 for each regime are in fig. 4.4. Recall that the optimal control, $\alpha^*(t)$, is open-loop and thus it is only a function of time.

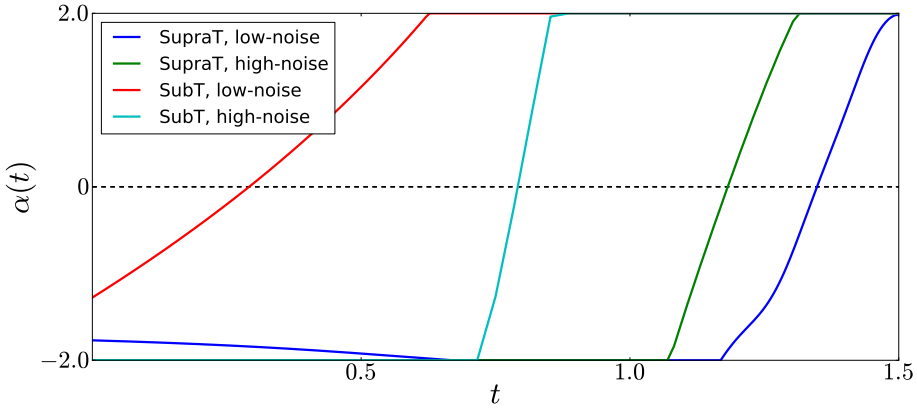


Figure 4.4: The deterministic optimal controls for each parameter regime as functions of time, $t \in [0, t^*]$. The desired spike time is set to $t^* = 1.5$, the energy penalty, $\epsilon = 0.001$ and the bounds are $\alpha \in [-2, 2]$. The initial value of the process is always the reset, i.e. $X_t = 0$ at time $t = 0$, see eq. (4.3.1). From left-to-right, the control curves correspond to the Sub-Threshold-Low-Noise, Sub-Threshold-High-Noise, Supra-Threshold-High-Noise, Supra-Threshold-Low-Noise regimes

Let us discuss the most salient features of the controls calculated in fig. 4.4. In general, the strategy is to inhibit the neuron at the beginning of the interval, $\alpha(t) = \alpha_{\min}$ for $t \ll t^*$ and to excite it near the end, $\alpha(t) = \alpha_{\max}$ for $t \approx t^*$. The smooth portion that connects these two segments in the middle of the interval is due to the energy penalty, $\epsilon \int \alpha^2(s)\,\mathrm{d}s$. The behaviour of the control in different regimes is relatively simple to explain and we see what we would expect. In a sense the bias current μ can be absorbed by the control and so increasing μ amounts to decreasing α modulo its bounds. That is exactly the difference between the Supra-Threshold vs.

Sub-Threshold plots in fig. 4.4 when holding the noise intensity fixed, either $\beta = 0.3$ or $\beta = 1.5$. Both for low and high noise intensity, a reduction of μ results in an increase in $\alpha(t)$. The effect of varying the noise intensity, β, is more subtle and it is not the same in the Supra vs. the Sub-Threshold regimes. In the Supra-Threshold regime, reducing β reduces the need for excitatory control towards the end, since the bias alone should put the neuron over the threshold. Thus the main part of the control with low noise in the Supra-Threshold regime is to stop the neuron from spiking too early. In the Sub-Threshold regime, the situation is somewhat reversed - reducing β obviates the need to apply an inhibitory control in the early part of the interval and also prompts the controller to apply excitatory input earlier, since it is the controller which is now the main drive for a spike.

4.6 Simulations

4.6.1 Single-Spike Control

Having obtained the optimal controllers, both closed- and open-loop, we evaluate their performance with simulated realizations of the voltage process, eq. (4.3.1). In particular, while they both minimize the expected squared deviation of the spike-time from some desired spike time, we analyze the distribution of the squared deviation and the behaviour of the controls for different parameter regimes.

In addition to the optimal controllers, we also show the behaviour of another control law - perhaps the most naive one - which is obtained by ignoring the noise and the energy penalty. That is, we find a *constant* value of α which satisfies the desired boundary conditions, $x(0) = 0, x(t^*) = x_{th}$. This naive controller will be called 'Deterministic', since it assumes deterministic dynamics in X, i.e., $\beta = 0$.

For the comparison, we sample N realizations of the controlled system and apply in turn each of the three controls. Naturally, we reuse the same realization of the

underlying stochastic process, W_t, for each of the three different controls.

We set $\epsilon = 0.001$ and $\alpha_{\max} = 2.0$. As such the energy cost is of secondary importance and the paramount effect on the objective is the squared difference, $(T_{\text{sp}} - t^*)^2$, which we are trying to minimize.

The performance of the three controllers for each parameter regime is given in table 4.2 and fig. 4.5. Naturally, the closed-loop achieves a lower error than the open-loop, and the naive controller fares worst. The difference in performance mostly depends on the strength of the noise. Thus, for low noise, the performance of the stochastic controllers, be they open- or closed-loop is not much superior to the naive deterministic controller. On the contrary - in the high-noise regime, using a stochastic controller gives a much lower error on average between the desired t^* and the realized T_{sp}.

To give a better view of the performance of the controllers as a function of the noise intensity we plot the 'Percentage of Correct Spikes' and the 'Average Squared Spike-Time Deviation' in fig. 4.6. In fig. 4.6 we also illustrate the effect of two values of the control bounds, $\alpha_{\max} = 2$ or 4. Naturally with a higher $\alpha_{\max}$ the error is reduced and the percentage of correct spikes is increased. As expected, the performance gets worse with increasing noise intensity. When the noise is small, the Supra-Threshold regime behaves best, but when the noise increases the Sub-Threshold regime provides more flexibility to correct for large perturbations caused by the noise.

For illustration sake, we also visualize some trajectories from the Sub-Threshold-High-Noise regime in fig. 4.7, see panels A),C),E). The most notable feature of the $\alpha(\cdot)$ plots in fig. 4.7 is that especially for $\epsilon = 0.001$, there is a high level of fluctuations in the closed-loop α. That is to be expected from the optimality condition in eq. (4.4.4). Since the control is proportional to the derivative of the value function and we are minimizing, roughly speaking, the control attempts to push the process, X, to the bottom of the valley that is formed by the value function, w. However as the stochastic fluctuations of $X(t)$ push it randomly to either side of that 'valley' and in turn force

the control to change signs to counter-act. These fluctuations are then amplified by the division by the small parameter ϵ so that the control mostly bangs up or down to its extremes $\alpha_{\min}, \alpha_{\max}$.

Control Law	Average Squared Spike-Time Deviation (N=10000)	Expected Squared Spike-Time Deviation (theory)
Deterministic	0.287	–
Open-Loop	0.003	0.008
Closed-Loop	0.001	0.003

(a) Supra-Threshold Low-Noise

Control Law	Average Squared Spike-Time Deviation (N=10000)	Expected Squared Spike-Time Deviation (theory)
Deterministic	1.095	–
Open-Loop	0.796	0.852
Closed-Loop	0.795	0.843

(b) Supra-Threshold High-Noise

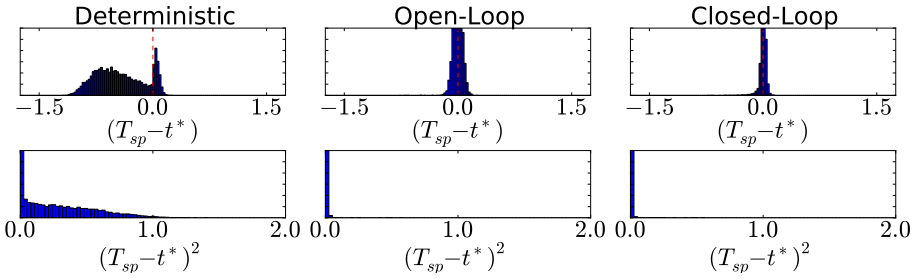
Control Law	Average Squared Spike-Time Deviation (N=10000)	Expected Squared Spike-Time Deviation (theory)
Deterministic	0.327	–
Open-Loop	0.142	0.150
Closed-Loop	0.095	0.098

(c) Sub-Threshold Low-Noise

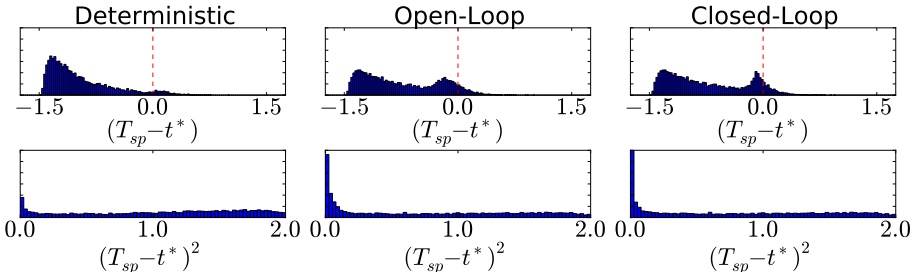
Control Law	Average Squared Spike-Time Deviation (N=10000)	Expected Squared Spike-Time Deviation (theory)
Deterministic	1.131	–
Open-Loop	0.394	0.404
Closed-Loop	0.360	0.365

(d) Sub-Threshold High-Noise

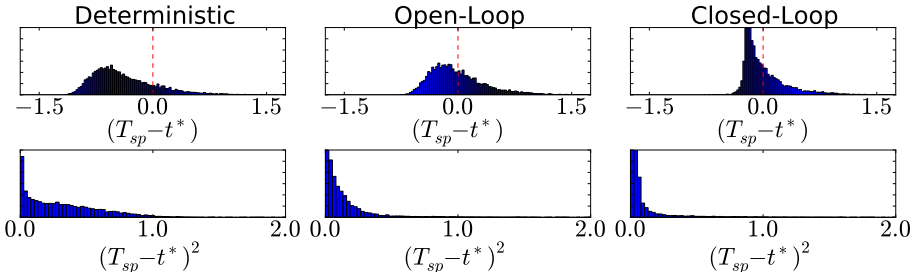
Table 4.2: Realized and theoretical performance of the different control laws. The empirical performance is obtained using $N = 10000$ sample paths. The theoretical performance is found using the optimal value for J for the open-loop stochastic control and the value function $w(x = 0, t = 0)$ for the closed-loop stochastic control.



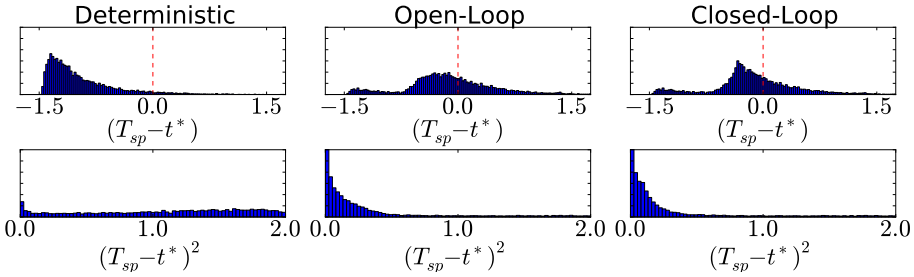
(a) Supra-Threshold Low-Noise



(b) Supra-Threshold High-Noise



(c) Sub-Threshold Low-Noise



(d) Sub-Threshold High-Noise

Figure 4.5: Histogram of the spike timing error for the deterministic (left) vs. open-loop stochastic (centre) vs. closed-loop stochastic (right) control laws. Recall that T_{sp} is the random realized spike time and t^* is the target spike time. This is for the same problem as in fig. 4.7. We have used $N = 10000$ sample paths to form the statistics.

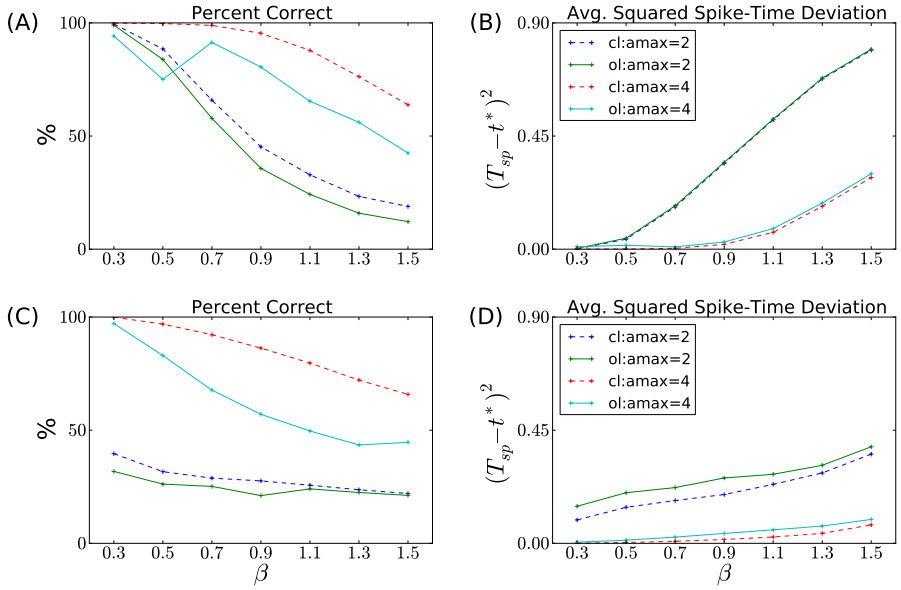


Figure 4.6: The performance of the controllers (open-loop and closed-loop) as a function of the noise intensity β. The target spike-time is again $t^* = 1.5$. On the left, panels A,C), we show the percent of correct spikes, which is defined as the number of controlled spikes which occur within 10% of t^*, the target spike time. On the right, panels B,D), we show the average squared difference between the controlled spike and the target spike. Above, in A,B), we show the Supra-Threshold regime. Below, in C,D), we show the Sub-Threshold regime

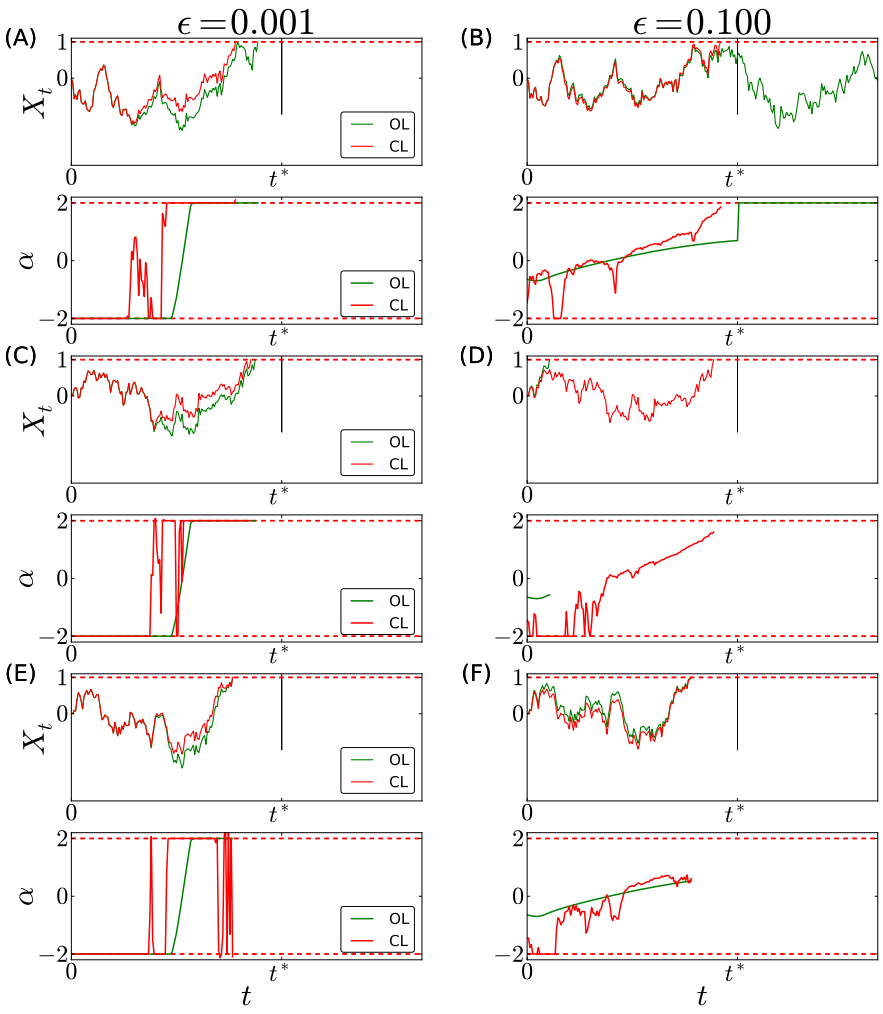


Figure 4.7: Three examples for the controlled trajectories using the deterministic, open-loop stochastic and closed-loop stochastic control approaches. The plots on the left, A,C,E, are obtained using a value of $\epsilon = 0.001$, while the plots on the right, B,D,F, use $\epsilon = 0.1$ (for a discussion on the effect of ϵ see section 4.7). The black vertical line in the plots indicates the desired spike-time, t^*. The parameter values are $\mu, \tau_c, \beta = [0.2, 0.5, 1.5]$ (Sub-Threshold High-Noise regime), with $t^* = 1.5$. The upper plots show the voltage evolution, $X(t)$, while the lower plots show the applied control, $\alpha(t)$. Note that the optimal control, $\alpha(t)$, obtained from the deterministic and from the open-loop controls are the same across all three samples.

4.6.2 Multi-Spike Control

We now turn to the ultimate goal of our analysis, the control of spike *trains*. The major challenge when working with spike trains, that is not present when considering intervals in isolation, is that the target spike train might contain several closely clustered spike times, such that if the first controlled spike time occurs with delay, there will be too little time to hit the next target spike times and they will all be delayed. However, we adapt the controller to the actual controlled spike occurrences and compensate so that it is still targeting the originally posited train. Thus there should be no accumulation of errors unless the controller is unable to catch up to an unusually high firing rate.

We proceed by generating $M = 50$ realizations of $N = 16$ spikes each in an attempt to meet a prescribed spike train. We focus on two regimes, the Supra-Threshold regime with either low or high noise. The results for parameters from the low-noise regime are shown in figs. 4.8a and 4.8b and results for the high-noise regime are shown in figs. 4.9a and 4.9b. In all cases, the target train is obtained by a simulation of the Supra-Threshold High Noise regime with no additional control, i.e., with $\alpha(\cdot) = 0$.

In each figure, figs. 4.8a, 4.8b, 4.9a and 4.9b on the top panel we show a smoothed version of the target train vs. an empirical firing rate from the simulations. The red curve is a smoothed Gaussian kernel applied at the target times where the standard deviation of the kernel is $\sigma = 0.1$. The blue curve is the empirical firing rate equal to the number of spikes in a small window divided by the number of realizations and divided by the width of the window w_b. We have used $w_b = 0.1$. In the bottom panels of figs. 4.8a, 4.8b, 4.9a and 4.9b, we show in detail 10 out of the $M = 50$ realizations.

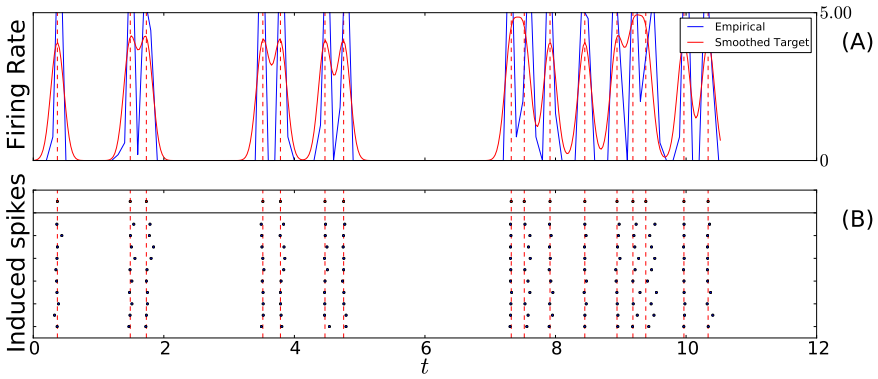
It is immediately clear that while the evoked trains closely follow the target train in the low-noise case, figs. 4.8a and 4.8b, this is a lot more difficult in the high-noise case, figs. 4.9a and 4.9b and the restricted control is not able to produce reliable

results. A stronger control is needed to obtain the same level of accuracy in the target. In fig. 4.10, we relax the bound constraints on the control $[\alpha_{\min}, \alpha_{\max}]$ from $[-2; 2]$ to $[-4; 4]$. Then a significantly better tracking of the target train is achieved. Thus, to control a system under higher noise comes at the price of allowing a more powerful control signal.

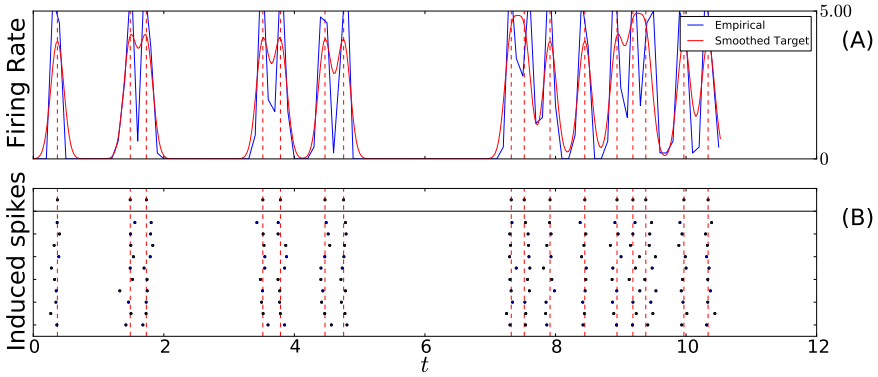
The main question in this section is whether the separate optimization of each interval independently of the others is a suitable approach when targeting a whole spike train. In particular, we do not account for the possibility that two target spikes occur closely together. Then it might make sense to attempt generating the first one a little earlier, on average, to give more time to generate the second. For example, in figs. 4.8a and 4.8b, the 13th target spike is successfully produced, but the 14th target spike occurs so soon after that in all trials it is delayed. However, if the target interspike intervals are within reasonable reach of the neuron, this is not a problem. Here, 'reasonable reach' is to be understood as any period longer than the time it takes to reach the threshold in the case of no noise and under maximum excitation, i.e. $\alpha(t) = \alpha_{max}$.

4.6.3 Controlling a Biophysical Model

To verify our method in a more biophysically realistic model, we use the established Morris-Lecar model [79], a Hodgkin-Huxley type of conductance-based model. To mimick an experimental situation we will not assume that we know the model nor the parameters, and simply use the LIF model (4.3.1)–(4.3.2) where parameters are estimated from data sampled before the control is initiated. First we estimate parameters for the LIF model from observations of the Morris-Lecar model. Then we control the Morris-Lecar model using the scheme derived for the estimated LIF model. Thus, the control strategies will be conducted in a realistic experimental setting with no specific knowledge of the underlying mechanisms, considering data as if coming

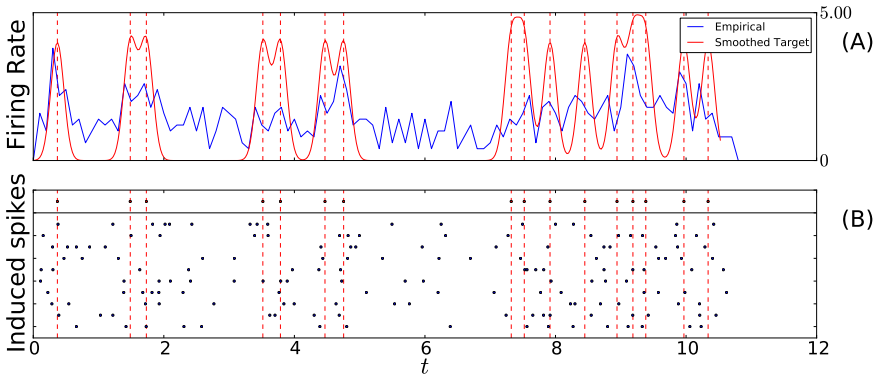


(a) Closed-Loop Controller

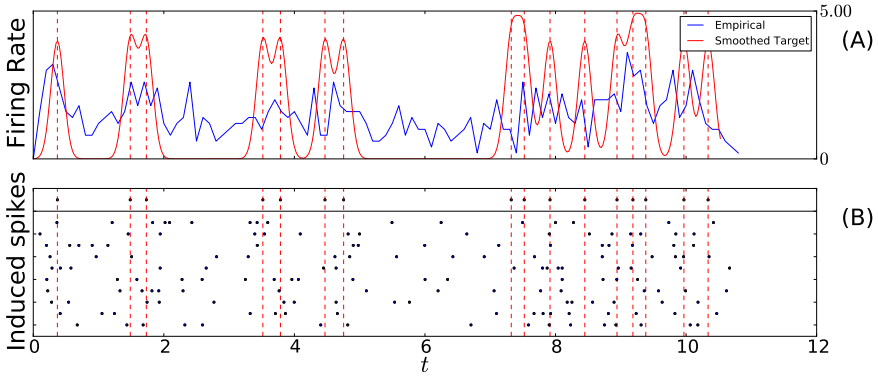


(b) Open-Loop Controller

Figure 4.8: Controller performance in the Supra-Threshold Low-Noise regime, given $\alpha \in [-2, 2]$. In (a) is depicted the closed-loop controller, in (b) the open-loop controller. In A) panels, the blue curve is the empirical firing rate of $M = 50$ controlled trains, while the red curve is a smoothed-version of the target train. (See text for details of how they are calculated). In the B) panels, the dashed, red lines indicate the target times, τ_n, while the blue dots are the realizations of the controlled trains. The upper spike train above the solid line is the target spike train. The parameters from the Supra-Threshold, Low-Noise regime are used, see table 4.1. The target spike train was generated using the model itself, with parameters from the Supra-Threshold, High-Noise regime, without an applied control, $\alpha = 0$.

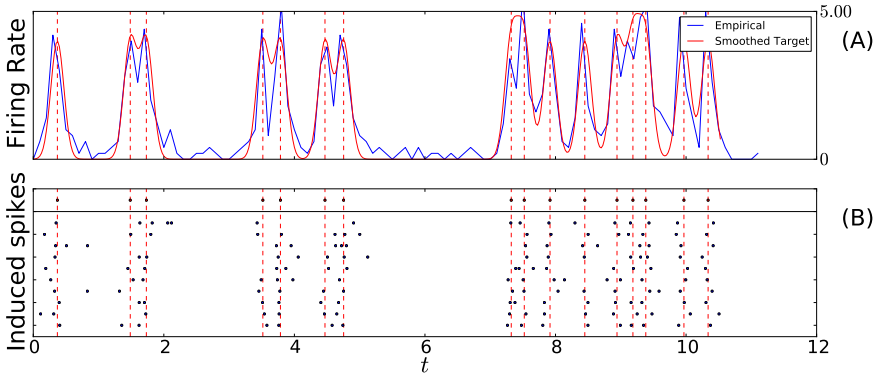


(a) Closed-Loop Controller

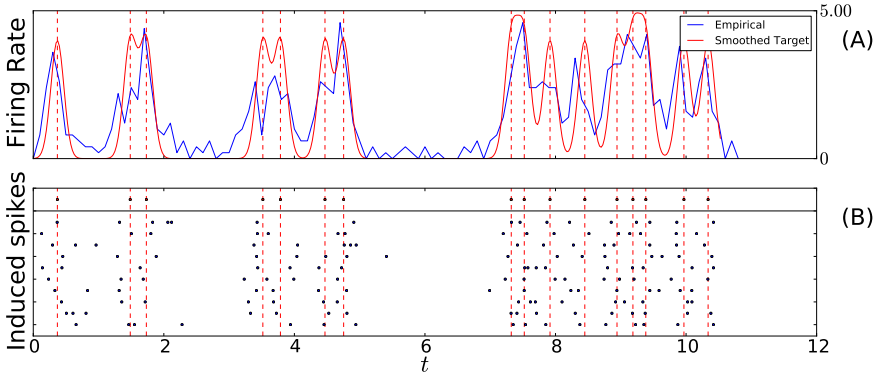


(b) Open-Loop controller

Figure 4.9: Controller performance in the Supra-Threshold High-Noise regime given $\alpha \in [-2, 2]$. In (a) is depicted the closed-loop controller, in (b) the open-loop controller. In A) panels, the blue curve is the empirical firing rate of $M = 50$ controlled trains, while the red curve is a smoothed-version of the target train. (See text for details of how they are calculated). In the B) panels below A) the dashed, red lines indicate the target times, τ_n, while the blue dots are the realizations of the controlled trains. The upper spike train above the solid line is the target spike train. The parameters from the Supra-Threshold, Low-Noise regime are used, see table 4.1. The target spike train was generated using the model itself, with parameters from the Supra-Threshold, High-Noise regime, without an applied control, $\alpha = 0$.



(a) Closed-Loop Controller



(b) Open-Loop Controller

Figure 4.10: Same as fig. 4.9, but with the control constraints, $[\alpha_{\min}, \alpha_{\max}]$ increased to $[-4, 4]$ instead of $[-2, 2]$.

from a black-box.

The Stochastic Morris-Lecar model

The stochastic Morris-Lecar model including both current and channel noise is defined as the solution to

$$\begin{cases} dV_s &= \frac{1}{C}\Big(- g_{Ca} m_\infty(V_s)(V_s - V_{C_a}) - g_K W_s (V_s - V_K) \\ & \quad -g_L(V_s - V_L) + I + A(s)\Big) dt + \gamma d\tilde{B}_s, \\ dW_s &= (\alpha(V_s)(1 - W_s) - \beta(V_s) W_s)\, dt \; + \sigma(V_s, W_s) dB_s, \end{cases} \tag{4.6.1}$$

where

$$\begin{aligned} m_\infty(v) &= \frac{1}{2}\left(1 + \tanh\left(\frac{v - V_1}{V_2}\right)\right), \\ \alpha(v) &= \frac{1}{2}\phi \cosh\left(\frac{v - V_3}{2V_4}\right)\left(1 + \tanh\left(\frac{v - V_3}{V_4}\right)\right), \\ \beta(v) &= \frac{1}{2}\phi \cosh\left(\frac{v - V_3}{2V_4}\right)\left(1 - \tanh\left(\frac{v - V_3}{V_4}\right)\right). \end{aligned}$$

The processes $\tilde{B}_s$ and B_s are independent Brownian motions. The variable V_s represents the membrane potential of the neuron at time s, and W_s represents the normalized conductance of the K^+ current. It varies between 0 and 1, and can be interpreted as the probability that a K^+ ion channel is open at time s. The equation for the dynamics of V_s contains four terms, corresponding to Ca^{2+} current, K^+ current, a general leak current, and the input current I. In addition it contains the externally applied control current $A(s)$.

The functions $\alpha(\cdot)$ and $\beta(\cdot)$ model the rates of opening and closing of the K^+ ion channels. The function $m_\infty(\cdot)$ represents the equilibrium value of the normalized Ca^{2+} conductance for a given value of the membrane potential. The parameters V_1, V_2, V_3 and V_4 are scaling parameters; g_{Ca}, g_K and g_L are conductances associated with Ca^{2+}, K^+ and leak currents; V_{Ca}, V_K and V_L are reversal potentials for Ca^{2+}, K^+ and leak currents; C is the membrane capacitance; and ϕ is a rate scaling parameter for the opening and closing of the K^+ ion channels.

The parameter γ scales the additive current noise. Conductance fluctuations caused by random opening and closing of ion channels leads to multiplicative noise on the conductance equation. Function $\sigma(V_s, W_s)$ models this channel or conductance noise. We consider the following function that ensures that W_s stays bounded in the unit interval if $\sigma \leq 1$ [80]:

$$\sigma(V_s, W_s) = \sigma\sqrt{2\frac{\alpha(V_s)\beta(V_s)}{\alpha(V_s)+\beta(V_s)}W_s(1-W_s)}.$$

Parameter values used in the simulations are the same as those of [81, 82] for a *class I* membrane: $V_K = -84$ mV, $V_L = -60$ mV, $V_{Ca} = 120$ mV, $C = 20\mu F/cm^2, g_L = 2\mu S/cm^2, g_{Ca} = 4, g_K = 8, V_1 = -1.2mV, V_2 = 18mV, V_3 = 12mV, V_4 = 17.4, \phi = 0.07, I = 40$. The noise intensities are taken from [83]: $\gamma = 1, \sigma = 0.2$. Trajectories are simulated with an Euler scheme with a time step of 0.01 ms, which is then sub-sampled every ten points to compensate for approximation errors of the simulation scheme. Finally, we obtain a data set with observations every $\Delta = 0.1$ ms. An example of a simulated trajectory is given in the top and bottom panels (V, W respectively) of Figure 4.11. The peaks correspond to spikes of the neuron.

Relating recorded voltage to a LIF model

Assume discrete observations of the membrane potential, here generated by the Morris-Lecar model as described above. These have to be related to the LIF model (4.3.1)–(4.3.2), which assumes that spikes are point events. Thus, the first problem is to partition the data into sub-threshold fluctuations and spikes. To make the method more robust we operate with two thresholds. A lower threshold, where fluctuations below can be well approximated by the LIF model, is employed to identify the data used for estimation of LIF parameters. We set this to $v_{th} = -22$ mV, see Fig. 4.11, upper and middle panel. A higher threshold is set to determine the start of the spike, a point of no return, from which the potential can only relax by going through a spike.

We set this to -14 mV, see Fig. 4.11, upper panel. When implementing the control assuming a LIF model, we attenuate the non-linear effects of being close to spiking in the biophysical model (or real data) by treating the interval between -22 and -14 mV as a grey zone, artificially setting the data to a constant value just below the threshold. Finally, we need to fix the resetting, which we set to $v_r = -44.53$ mV.

In the Morris-Lecar model the spike is not a point event, and we have to cut out the dynamics during spikes, where we will make no control and just wait for the reset. The effect of a spike for this setting of parameters is around $t_{ref} = 40$ ms, see lower panel of Fig. 4.11, at which point we restart. The time t_{ref} between the start of the spike and the reset is collapsed to a point as far as the LIF model is concerned. The value 40 is obtained from studying the voltage traces and is specific to the spike-dynamics of the Morris-Lecar model. In a real experimental setting it will probably be shorter, being the sum of the spike duration and the refractory period, maybe 10 ms.

The LIF model (4.3.1)–(4.3.2) is non-dimensional and normalized, resetting at 0 and spiking at a threshold of 1. Therefore the data have to be transformed. First we non-dimensionalize time by the transformation $t = s/\bar{T}$, where s is the time-scale of the measurements, and $\bar{T}$ is some mean interspike-interval, here we use the average of the target spike trains, $\bar{T} = 177.48$ ms. Then we transform an observation V_s to $X_t = (V_s - v_r)/(v_{th} - v_r)$. On the transformed data we estimate the parameters μ, τ_c and β. Then we are ready to start the control. The control $\alpha(t)$ is calculated on the transformed data, transformed back to a control on the original scale, $A(s) = (v_{th} - v_r)\tau_c\alpha(t)$, and fed into the Morris-Lecar model (or the cell in an experiment).

Estimating LIF parameters from Morris-Lecar data

Assume $\alpha(\cdot) = 0$ and discrete observations of model (4.3.1)–(4.3.2), $X_n^{(k)}$, for $k = 1, \ldots, K$, where K is the number of interspike intervals, and $n = 0, 1, \ldots, N_k$, where

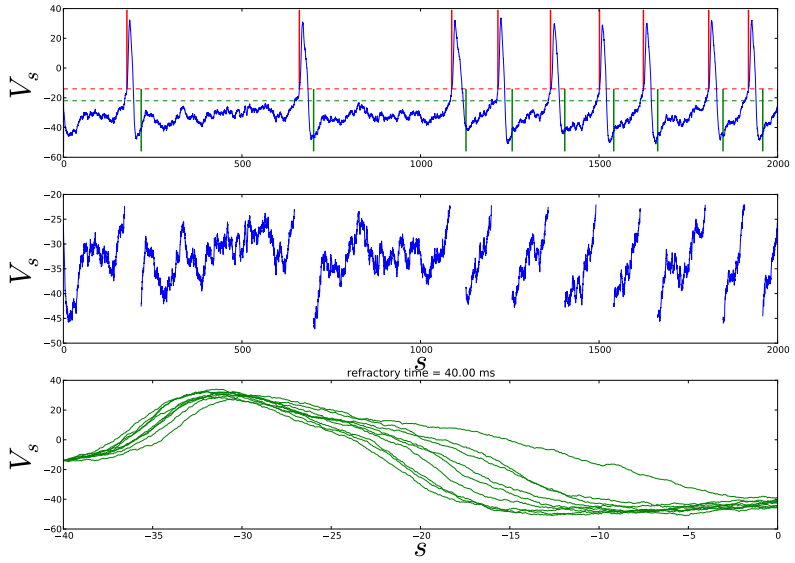


Figure 4.11: Annotating (spike-segmenting) the raw Morris-Lecar data. The top panel shows a trajectory of V_s from the Morris-Lecar model. The middle plot shows the data segments that are used to estimate the LIF parameters from Morris-Lecar data. The bottom plot shows the stereotypical spike shape and justifies why 40 ms is a reasonable recovery time

$N_k + 1$ is the number of observations in the kth interspike interval. The maximum likelihood estimators for the parameters μ, τ_c and β^2 in absence of a threshold are given by

$$\begin{aligned}
\hat{\mu} &= \frac{\sum_{k=1}^{K}\sum_{n=1}^{N_k}\left(X_n^{(k)} - e^{-\frac{\Delta}{\hat{\tau}_c}}X_{n-1}^{(k)}\right)}{N(1-e^{-\frac{\Delta}{\hat{\tau}_c}})} \\
e^{-\frac{\Delta}{\hat{\tau}_c}} &= \frac{\sum_{k=1}^{K}\sum_{n=1}^{N_k}(X_n^{(k)}-\hat{\mu})(X_{n-1}^{(k)}-\hat{\mu})}{\sum_{k=1}^{K}\sum_{n=1}^{N_k}\left(X_{n-1}^{(k)}-\hat{\mu}\right)^2} \\
\hat{\beta}^2 &= \frac{2\sum_{k=1}^{K}\sum_{n=1}^{N_k}\left(X_n^{(k)}-\hat{\mu}-(X_{n-1}^{(k)}-\hat{\mu})e^{-\frac{\Delta}{\hat{\tau}_c}}\right)^2}{N(1-e^{-2\frac{\Delta}{\hat{\tau}_c}})\hat{\tau}_c}
\end{aligned}$$

where $N = \sum_{k=1}^{K} N_k$.

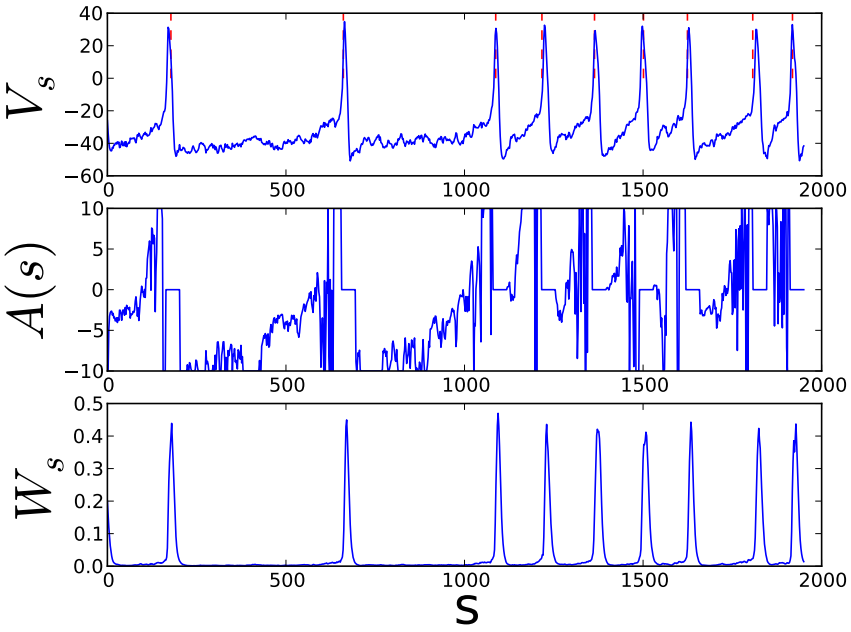
Given Morris-Lecar data points over the time interval from 0 to 2 sec, we get the following estimates: $\hat{\mu} = 3.57, \hat{\tau}_c = 0.20$ and $\hat{\beta} = 0.59$. For the bounds on our control we set $A(s) \in [\pm 10]$, which corresponds to $\alpha(t) \in [\pm 2.27]$.

Now we consider the more difficult case when only spike-times are available. Estimating τ_c from spikes is not accurate, it is (almost) un-identifiable. Therefore, we assume a fixed value of $\tau_c = 0.11$. This is quite different from the value of τ_c estimated from intra-cellular recordings, so that we investigate robustness to misspecification of this value. Thus we only estimate μ and β, using the Fortet equation, for details see [40]. The estimates are $\hat{\mu} = 6.17$ and $\hat{\beta} = 0.78$. This corresponds to $\alpha(t) \in [\pm 3.94]$ if $A(s) \in [\pm 10]$.

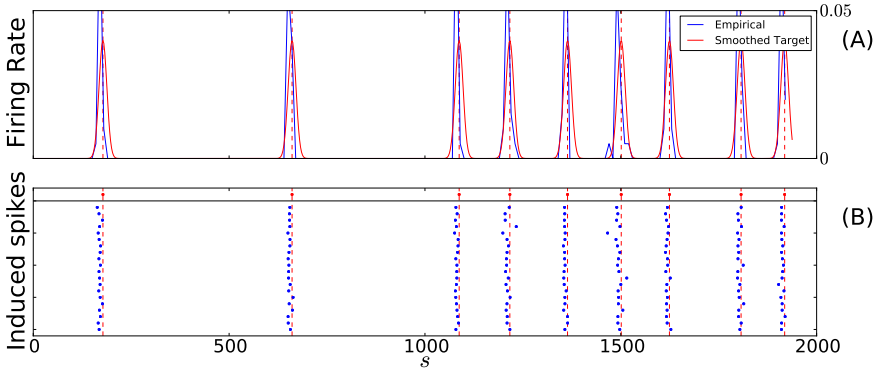
Controlling the Morris-Lecar model

We now have all the pieces to control the Morris-Lecar model. In the closed-loop control strategy we use the voltage-based estimates for the LIF model parameters, while for the open-loop strategy we use the intervals-based estimates. This is a disadvantage for the open-loop strategy, since on average it will have poorer estimates of the real system, but provides a realistic picture of an experimental situation. The

closed-loop results are shown in Fig. 4.12, and the open-loop results are shown in fig. 4.13. The results are comparable to the results of fig. 4.8, and show that for the purpose of stochastic control, it is not crucial to use a biophysical realistic model nor knowing exactly the parameters. This is partly due to a lower value of τ_c in the Morris-Lecar model, compared to what was used in the earlier simulations, and to slightly higher values of the bounds of the control. The variance is proportional to $\tau_c\beta^2$, which in the simulations are 1.125 (high noise) or 0.045 (low noise), and in the estimated Morris-Lecar is 0.07. There might also be some error cancelling, in the sense that a wrong model and wrongly estimated parameters might be the best choice for the model at hand, thus providing a robust control.

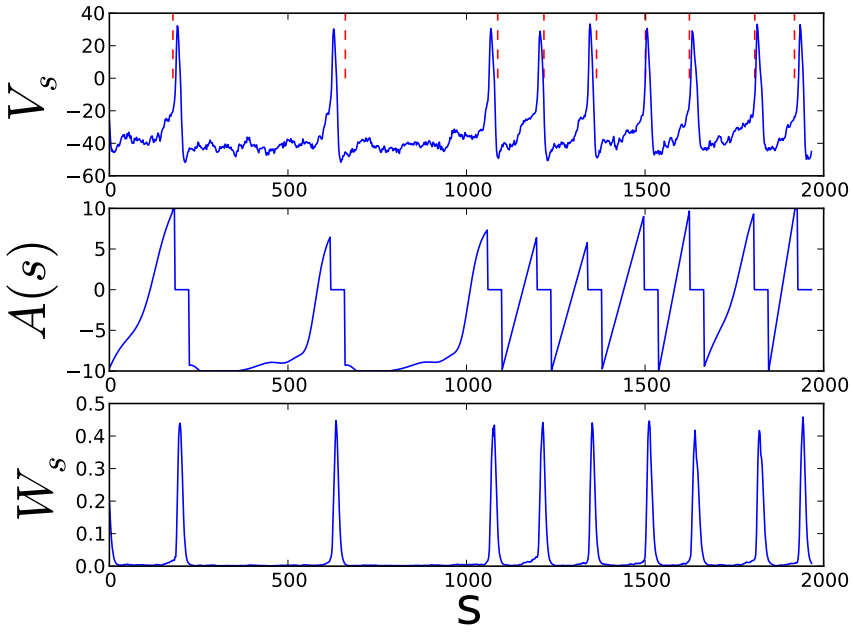


(a) Example controlled trajectory

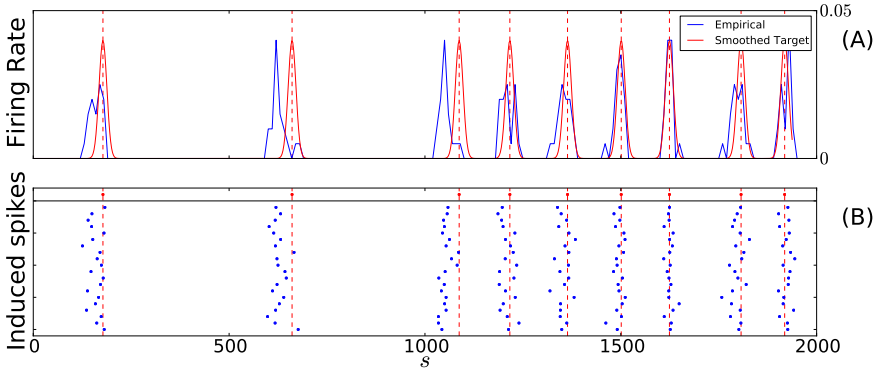


(b) Raster plot for controlled simulations

Figure 4.12: Closed-loop control of the Morris-Lecar dynamics. In (a) are plots of the trajectory and the control, in (b) is a raster plot of realized spikes. The upper spike train above the solid line is the target spike train



(a) Example controlled trajectory



(b) Raster plot for controlled simulations

Figure 4.13: Open-loop control of the Morris-Lecar dynamics. In (a) are plots of the trajectory and the control, in (b) is a raster plot of realized spikes. The upper spike train above the solid line is the target spike train

4.7 Effect of energy penalty

So far we have assumed a low value for the energy penalty parameter, ϵ, in the objective defined in eq. (4.3.5). Our approach has been to be primarily concerned with the accurate spiking and that penalizing energy is rather intended to regularize the control and avoid excessive chattering of the control between its extreme values, than as a goal in minimizing energy in its own right. Now, we take some time to explore the effect of a higher ϵ on the optimal controls. Intuitively, a higher value of ϵ will tend to bring the optimal control, α^*, closer to zero. We show the effect of setting $\epsilon = 0.1$ in each of the four regimes for both the closed-loop control and the open-loop control in figs. 4.14 and 4.15 respectively. These should be compared to figs. 4.3 and 4.4, respectively. Also, in fig. 4.7 on the right we show example trajectories for the higher energy parameter value, $\epsilon = 0.1$, in the Sub-Threshold High-Noise regime.

Indeed, for both the closed-loop and open-loop optimal controls, increasing ϵ tends to reduce the absolute value of $\alpha^*(t)$. In particular, for the closed-loop control it tends to broaden the area of transition for decreasing x when the control swings from its minimal, i.e., inhibitory value, to its maximal, i.e., excitatory value. Similarly for the open-loop control, instead of banging from its minimal bound at the beginning of the interval to its maximal bound at the end, the optimal control for $\epsilon = 0.1$ tends to have a more mild transition from a slightly inhibitory to slightly excitatory values.

Note however that increasing ϵ has an important effect on the validity of the approximation used to form the terminal conditions for the value function, w, and the adjoint variable, p. We assumed that applying $\alpha_{\max}$ is optimal for $t > t^*$. This is indeed true in the limit $\epsilon \to 0$. However, even for a finite value of ϵ, this may still be the optimal thing to do. Indeed, in the original calculations, where $\epsilon = 0.001$, see the panels on the left in figs. 4.14 and 4.15, the so-obtained value function *implied* that $\alpha(x, t^*) = \alpha_{\max}$. In other words, our guess that $\alpha(x, t) = \alpha_{\max}$ for $t > t^*$, is self-consistent with the so-obtained value function. This is akin to confirming an

ansatz. For a higher value of ϵ, like $\epsilon = 0.1$, this is no longer the case. The red curves on the right-side of fig. 4.14 are no longer pushed up at $\alpha = \alpha_{\max}$, and as such it is no longer valid, strictly speaking, to assume that the value function can be obtained by assuming $\alpha(t)|_{t>t^*} = \alpha_{\max}$ and then calculating the expected remaining time-to-spike. The exact same conclusion can be drawn from the open-loop control. There the optimal control always (almost) reaches its maximum, $\alpha_{\max}$ while choosing $\epsilon = 0.001$, this is no longer so for the higher $\epsilon = 0.1$. In fact, as we mentioned after eq. (4.4.5), the correct thing to do for higher values of ϵ is to push the calculation interval to some $t^+ > t^*$, apply the same terminal condition at t^+ instead of t^* and then solve backwards (either for the closed- or open-loop control). One will be guaranteed to find some $t^+ > t^*$ that works since eventually the quadratic term $(t - t^*)^2$ will dominate the energy term in the objective, which is linear in t. We do not explore this further here.

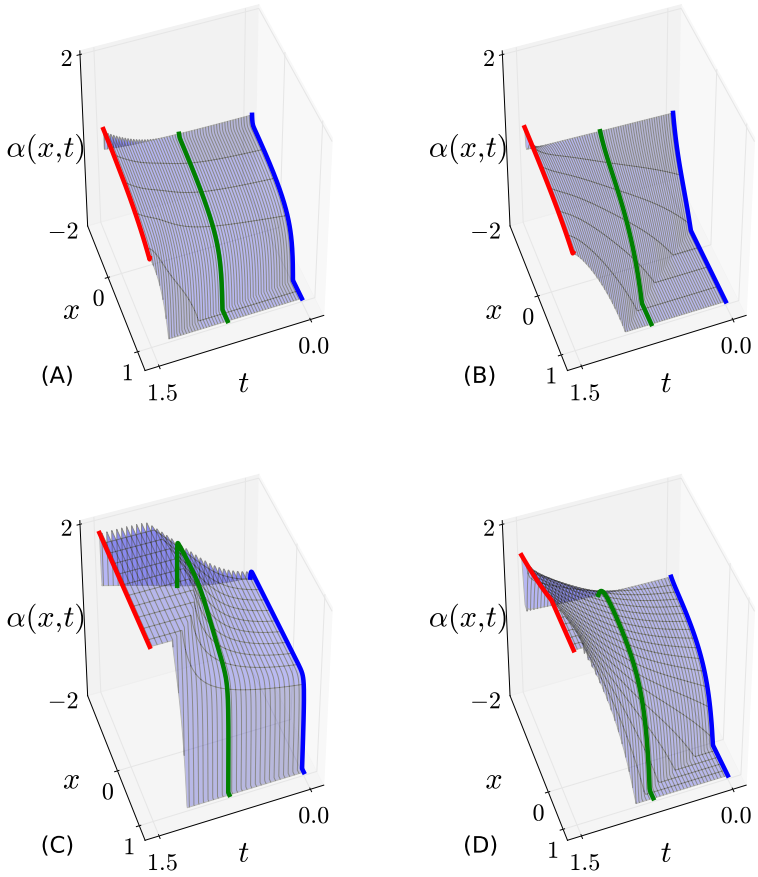


Figure 4.14: The effect of the energy parameter ϵ on the closed-loop control. This is the same as fig. 4.3, but for the higher value of $\epsilon = 0.1$. The desired spike time is set to $t^* = 1.5$ and the control bounds are $\alpha \in [-2, 2]$. A,B) Supra-Threshold-Low-Noise C,D) Supra-Threshold-High-Noise E,F) Sub-Threshold-Low-Noise G,H) Sub-Threshold-High-Noise

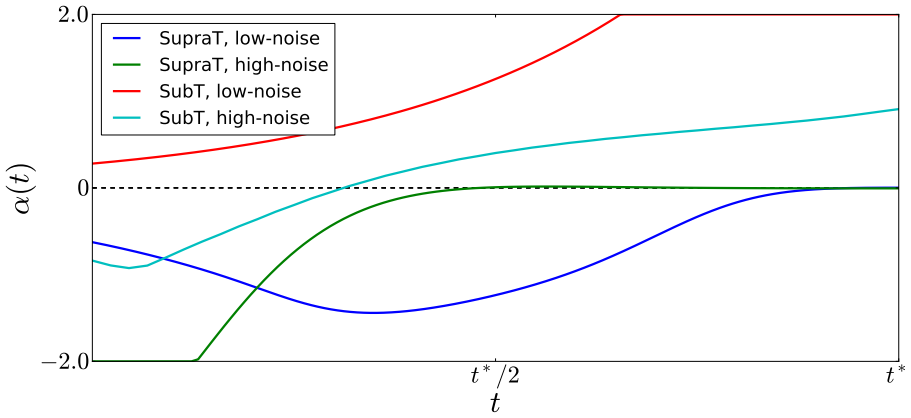


Figure 4.15: The effect of the energy parameter ϵ on the open-loop control. This is the same as fig. 4.4, but with the higher value of $\epsilon = 0.1$. The desired spike time is set to $t^* = 1.5$ and the bounds are $\alpha \in [-2, 2]$.

4.8 Discussion

We have analyzed and computed the optimal control of spikes in two scenarios - one where the underlying voltage of the neuron is observable to the controller (closed-loop control) and one where only the spikes are observable (open-loop control).

Naturally, the ability to control the system is most clearly affected by the level of the noise. In a sense, the noise acts like an adversary - in our context it has no beneficial role, except to obstruct precise spiking. When the level of the noise is high, a stronger control is needed, and the trade-off between accuracy (hitting the target spikes), possible damage of the system (the limits of the control) and energy expenditure has to be considered.

The bias current has a more helpful role at least in our examples, where a high value of the bias tends to help precise spiking. Of course, where it helps or hinders depends on the value of the desired spike time - a combination of a high positive bias and a 'distant' spike time, will tend to be difficult to control as the system will naturally tend to spike earlier than desired. In summary, we have found that systems in the Supra-Threshold, Low-Noise regime tend to be most accurately controlled, while the Supra-Threshold, High-Noise regime are the least accurate.

It should be noted that we do not require that the controlled cell is already in the Supra-Threshold regime - the control scheme applies in the same manner in the Sub-Threshold regime as well. It is only required that the maximum value of the control puts the model in Supra-Threshold regime, implying that the control can always achieve Supra-Threshold behaviour. The Supra- and Sub-Threshold distinction only applies to the intrinsic dynamics in absence of control, and ignoring the noise.

In both contexts, open-loop and closed-loop, finding the optimal control is computationally intensive as we numerically solve partial differential equations. If these algorithms were to be practical in an online setting, some more work would have to be done in order to ensure they can be computed very quickly (in the millisecond

range) or that they can be pre-computed.

Our work has stayed close to the paradigm of [72] in trying to make the neuron spike at a particular time. Similiar to them, we have also incorporated into our objective the minimization of total energy used to achieve our goal, which as discussed in [72] is sensible given the potential damage to the cell of accumulating charge. Furthermore, we have assumed that our control is constrained in magnitude, which is natural given physical limitations on equipment and safety considerations. Another possible constraint on the control is that it is charge-balanced, meaning that its time integral is zero, $\int u \, \mathrm{d}t = 0$. Such constraints are most easily posed in the context of deterministic spiking models like the Phase-Response Curve, see e.g. Danzl et al. [84]. It is possible to add them to the open-loop control, since the control is still deterministic, but it is non-trivial. It is even less trivial when one uses Dynamic Programing. In particular, insisting on charge-balance in our schemes will make it much more difficult to apply the Terminal Conditions used in deriving both the closed- and open-loop controls.

As points of outlook, we mention that our scheme can potentially be applied to produce a periodic train in the face of noise with applications to various biological pacemakers, or it can also be used to produce the opposite effect - a random behaviour at the single cell level, for example by specifying a target train with Poisson-like characteristics.

The novelty in our work is primarily in obtaining a stochastic control scheme, which is not limited by the intensity of the noise in the system. In fact, by comparison with the naive 'deterministic' controller, we have shown how wrong the control strategy can be if we assume that the noise is negligible. Given this scope, the numerical demands on our scheme are non-trivial, but our experience and experiments show that these demands can be addressed. It however remains to be seen whether they can be successfully applied in the lab and beyond.

4.9 Calculating the Terminal Conditions for the HJB equation

Here we explain how to calculate the first two moments of the remaining time-to-spike, $\mathbb{E}\left[(T_{sp}-t^*)^n \,\middle|\, X(t^*)=x, \alpha(t)=\alpha_{\max}\right]$, $n=1,2$, which is needed for the terminal conditions for $w(x,t^*)$ and $p(x,t^*)$ in, respectively, eq. (4.4.7) and eq. (4.5.3).

Let

$$\begin{aligned} T_{(1)}(x) &= \mathbb{E}\left[(T_{sp}-t^*) \,\middle|\, X(t^*)=x, \alpha(t)=\alpha_{\max}\right], \\ T_{(2)}(x) &= \mathbb{E}\left[(T_{sp}-t^*)^2 \,\middle|\, X(t^*)=x, \alpha(t)=\alpha_{\max}\right], \end{aligned} \tag{4.9.1}$$

where as before, $T_{sp}-t^*$ is the remaining time-to-spike in eq. (4.3.1), with the voltage being $X(t^*)=x$ at the desired spike-time $t=t^*$ and applying the control $\alpha(t)=\alpha_{\max}$ afterwards. Naturally, the $T_{(i)}$'s depend on x and are affected by the value of the various parameters, α, β, τ_c. We only need $T_{(2)}$ for the TCs in eq. (4.4.7) and eq. (4.5.3), but $T_{(1)}$ is needed as an auxiliary.

Analytical expressions exist for $T_{(i)}$, e.g. see [34]. However in our experience, they are awkward to work with numerically, especially as they involve infinite alternating sums which can easily overflow in numeric calculations. Moreover, they are not particularly convenient if you need $T_{(i)}(x)$ for a wide range of x's as we do here, since they calculate the value independently for each x. Instead we will follow and slightly extend Jacobs, [13], in deriving and solving a differential equation for $T_{(i)}(x)$. Jacobs only discusses how to calculate the first moment $T_{(1)}$, however it is quite clear from his discussion how to proceed recursively to obtain any higher moment $T_{(i)}$, by using previously calculated lower moments. This is what we do here. Moreover the relation between mean exit times and boundary value problems is discussed in most introductory books on stochastic processes, see [9] for example.

Basically we are calculating the mean squared time to exit an interval $[x_-, x_{th}]$. Theoretically, $x_-=-\infty$, but for our purposes we will set it to some finite value and impose reflecting boundaries there just as we did for the value function w. This is a

very good approximation for $x_- \ll 0$, as long as $\alpha = \alpha_{\max} > 0$.

Let $B(y,t|x) = \mathbb{P}[X_0 = y | X(t) = x]$ for a fixed x. Then B solves the following (backward Kolmogorov) PDE

$$-\partial_t B = (\mu + \alpha - \frac{y}{\tau_c})\partial_y B - \frac{\beta^2}{2}\partial_y^2 B. \tag{4.9.2}$$

Let $\bar{G}(t,y)$ be the survival function for T_{sp} given $X(t^*) = y$,

$$\bar{G}(t,y) = \mathbb{P}[t^* > t | X(t^*) = y] = \mathbb{P}[X(s)|_{s\le t} \in [x_-, x_{th}] \,\big|\, X(t^*) = y].$$

Because the dynamics do not depend explicitly on time, we have

$$\bar{G}(t,y) = \int_{x_-}^{x_{th}} B(y,t|x)\,\mathrm{d}x$$

and so $\bar{G}$, being a definite integral of B satisfies the same PDE:

$$-\partial_t \bar{G} = (\mu + \alpha - \frac{y}{\tau_c})\partial_y \bar{G} - \frac{\beta^2}{2}\partial_y^2 \bar{G}, \tag{4.9.3}$$

with the BCs, ICs:

$$\begin{cases} \bar{G}(t, x_{th}) = 0 & (T_{sp} = t \implies \mathbb{P}[T_{sp} > t] = 0) \\ \partial_x \bar{G}(t, x_-) = 0 & (\text{lower boundary condition}) \\ \bar{G}(t^*, y) = 1. & (T_{sp} \text{ is definitely} > \tau \text{ for } X(t^*) \in (x_-, x_{th})) \end{cases} \tag{4.9.4}$$

If we solve the PDE for $\bar{G}$ over $t \in [0,\infty)$, then we can calculate $T_{(2)}(x)$ as:

$$T_{(2)}(x) = \int_0^\infty t^2 \cdot (-\partial_t \bar{G}(t,x))\,\mathrm{d}t \tag{4.9.5}$$

$$= \int_0^\infty 2t \cdot \bar{G}(t,x)\,\mathrm{d}t, \tag{4.9.6}$$

assuming it exists.

Since $T_{(2)}$ is an integral over time, we can reduce the PDE in eq. (4.9.3) for the distribution to an ordinary differential equation (ODE) for the moments by integrating over time. Multiply both sides of eq. (4.9.3) by $2t$, and integrate with respect to time:

$$\int_0^\infty 2t(-\partial_t \bar{G})\,\mathrm{d}t = \int_0^\infty 2t \left[(\mu + \alpha_{\max} - \frac{y}{\tau_c})\partial_y \bar{G} - \frac{\beta^2}{2}\partial_y^2 \bar{G}\right]\,\mathrm{d}t. \tag{4.9.7}$$

The left-hand side turns out to be $2T_{(1)}$, so

$$2T_{(1)} = (\mu + \alpha_{\max} - \frac{y}{\tau_c})\partial_y T_{(2)} - \frac{\beta^2}{2}\partial_y^2 T_{(2)}. \tag{4.9.8}$$

Similarly we can derive an equation for $T_{(1)}$, namely:

$$1 = (\mu + \alpha_{\max} - \frac{y}{\tau_c})\partial_y T_{(1)} - \frac{\beta^2}{2}\partial_y^2 T_{(1)}. \tag{4.9.9}$$

In both cases, the boundary condtions will be:

$$\begin{cases} T_{(i)}(x_{th}) = 0 \qquad \text{(we are already spking)} \end{cases} \tag{4.9.10}$$

Equations 4.9.9 and 4.9.8 can be reduced to first order differential equations and solved analytically, at least up to some definite integral, but the analytical solution is not particularly useful or necessary for numerical calculations. Instead, we opt to calculate the solutions to 4.9.9 and 4.9.8 directly via an ODE solver, starting from $T_i(x_{th}) = 0$ and integrating down for $x \in [x_-, x_{th}]$. That is how we have obtained the terminal conditions for both $w(x, t^*)$ and $p(x, t^*)$ to apply in, respectively, eq. (4.4.7) and eq. (4.5.3).

4.10 Gradient Descent Algorithm for the Maximum Principle

For completeness we give the full pseudo-code for the gradient-descent algorithm used to compute the optimal control in the case of minimal-observation (the non-feedback case), see Algorithm 1. The algorithm continues on the next page.

Implementation Details

All the code used in this chapter including code to generate the figures can be found on github @ https://github.com/aviolov/OptSpikePython. Please see the README.md file for guide to the codes.

Algorithm 1 Gradient descent algorithm for obtaining the optimal open-loop control

Fix $t^*, \epsilon, \ldots$ the problem parameters

Fix $\{t_n\}_0^{N_t}$ a time-discretization of $[0, t^*]$

Fix g_{tol} a convergence tolerance for the gradient

Fix $K_{\max}, J_{\max}$ number of maximum iterations in outer, resp. inner descent loops

Fix s the initial step-size.

\# *we use* $g_{tol} = 1e-5, K_{\max} = 100, J_{\max} = 10, s = 10$

$\alpha_1(t) \leftarrow (\alpha_{\max} - \alpha_{\min}) \cdot t/t^* + \alpha_{\min}$

\# $\alpha_1(t) \sim$ *initial guess for the control, linear interpolate between* $\alpha_{min}, \alpha_{max}$

for $k = 1 \ldots K_{\max}$ **do**

\# *This is the outer loop where we descend down different gradients*

Calculate $f_k, J_k, p_k, \delta_\alpha J_k$ corresponding to α_k from eqs. (4.5.1) to (4.5.3) and (4.5.5)

$N_{active} \leftarrow$ Number of time nodes t_n, where either $\alpha_k(t_n) \neq \{\alpha_{\min}, \alpha_{\max}\}$ or $\delta_\alpha J_k(t_n)$ points inwards

if $||\delta_\alpha J_k||_{\mathbb{R}^{N_{active}}} \leq g_{tol} \cdot N_{active}$ **then**

\# *'Active' gradient is small enough, consider converged:*

BREAK

end if

 # Find the step size, s, of how far to move α in the direction $\delta_\alpha J_k$:

for $j = 1 \dots K_{\max}$ **do**

 # This is the inner loop where we find how much to descend down the current gradient

$\alpha_{k,j} \leftarrow \alpha_k - s_j \cdot \delta_\alpha J_k$

 # $\alpha_{k,j}$ is the new control strategy to try

Calculate $f_{k,j}, J_{k,j}$ corresponding to $\alpha_{k,j}$ from eqs. (4.5.1) and (4.5.2)

 # Recall $f_{k,j}$ is a probability density resulting from the control $a_{k,j}$ and $J_{k,j}$ is the objective value resulting from $f_{k,j}$

if $J_{k,j} < J_k$ **then**

 # We found a better (smaller) objective value

$s \leftarrow 2s_j$ *# try a more aggressive step in the next iteration*

BREAK

end if

if $j == J_{\max}$ **then**

 # We exhausted the step search without finding a smaller J, return current values

Return J_k, α_k

end if

 # Continue the inner looop, now try with a smaller step:

$s_{j+1} \leftarrow \mathrm{s}_j/2.$

end for *# single step loop*

if $k == K_{\max}$ **then**

ERROR 'Could not converge'

end if

 # Assign the new candidate for the optimal control and re-loop

$\alpha_{k+1} \leftarrow \alpha_{k,j}$

end for *# gradient descent loop*

return J_k, α_k

Chapter 5

Optimal Design for Estimation in SDEs

5.1 Thesis Context

In the last of the three main chapters of the thesis, we discuss the problem of stimulating a neuron model system in order to obtain higher fidelity model parameter estimates. We again consider the case where only hitting-times observations are available. We use the Mutual Information as the formal criterion for optimization, namely the Mutual Information between the hitting-times observations and the unknown parameters, over which we assume a prior distribution.

This chapter can be seen to combine many of the features of the previous two chapters - we are again trying to control hitting-times without observing the detailed trajectories as in chapter 4, but with the ultimate goal of estimating parameters from these observations as in chapter 3. Mathematically the greatest difference between the optimization problem in this chapter and that in chapter 4 is that we no longer assume a definite single set of model parameters, but instead must work with *distributions* over the parameters. This renders the problem more difficult and in particular makes

the adjoint derivations more complex. Instead of a single-pair of forward-backward PDEs, we now must deal with a ensemble of loosely coupled forward-backward PDEs.

We use this to find the optimal stimulation and perform several simulated estimations to demonstrate the superiority of the Mutual Information-optimal control over common-sense alternatives.

5.2 Introduction

Techniques for the estimation of equation parameters from observations of a stochastic process are an active area of research. Generally one hopes to have access to sufficient measurements of the state variables to guide the estimation of these parameters. However, in some cases the state variables are not measurable, and one has to work instead with times at which specific events occur, such as an earthquake or an epileptic seizure. Here we consider the problem of optimal design for estimation of parameters in a stochastic differential equation (SDE) given only observations of hitting times, i.e., of times at which the process crosses a fixed threshold. More specifically, since a primary motivation is the activity of neurons, we study the leaky integrate-and-fire (LIF) neuronal model, where the hitting times are identified with crossings from below of a threshold voltage. This corresponds to so-called extracellular potential measurements of spikes, which occur when the voltage inside the cell, evolving according to an Ornstein-Uhlenbeck process, crosses a fixed threshold value. While our motivation is based in neuroscience, the technique we develop is more generally applicable to dynamical models from which a point process can be extracted.

We assume that the controller has influence over some part of the SDE by causing time-varying perturbations of the process. In the case of a neuron, this would correspond to the case where one could influence the voltage evolution, e.g. through modulation of synaptic input, while one could only reliably record threshold crossing times. We further assume that the objective is to obtain crossing time observations,

known as hitting times or spike times in the neural context, that are most informative in both formal and informal senses. That is, we choose the control as the solution to a well-posed optimization problem, where the ultimate goal is to obtain more accurate and more precise estimates of the unknown parameter(s).

We espouse a Bayesian perspective and proceed by assuming a prior distribution of the unknown parameters. Then we seek to maximize the mutual information (MI) between the posterior of the unknown parameters given the observations and the distribution of the hitting times. We note that, even for a fixed and known parameter set, the hitting time observations are still random due to the inherent stochasticity of the SDE.

Using MI for the selection of maximally informative experiments has been advocated by several recent lines of research, for example, in experimental psychology, [85, 86], computational neuroscience, [87, 88, 89], and quantum physics, [90]. An alternative approach is to maximize the expected Fisher information of the experiment as in [91]. This is especially effective if the Fisher information is available as an analytical formula. That is the case when one observes trajectories from the diffusion process, as in [91], but not in the much more limited context of observations of hitting times.

The MI measures the expected reduction in uncertainty in the parameter value given an observation, under a fixed experimental stimulation. Thus, the stimulation that maximizes the MI should extract the maximum information about the model parameters, on average. Using MI for estimating statistical quantities is formally justified; see e.g. [88] where it is proven under fairly weak assumptions that using a stimulus input that maximizes the MI between future observations and the parameters of interest leads to consistent and efficient parameter estimates. However, while theoretically sound, optimizing or even merely computing the MI may be computationally prohibitive. For our problem, the primary issue is the low level of information available given that one has only access to hitting times, rather than measurements

of the underlying state variable.

We explicitly single out a paper [90] that describes a strategy to experimentally estimate the parameters of the Hamiltonian of a quantum system. The qubit state outcomes of zeroes or ones follow a Bernoulli distribution whose parameter is sought, and the time at which observations are made is under experimental control. They use a simple filtering technique originally suggested by [92] to maintain and update the prior distribution of the unknown parameters. In principle, their approach is similar to ours; the main difference is the nature of the observations, which in our case are the hitting times.

As posed, our optimization problem reduces to a slightly non-standard partial differential equation (PDE) optimization problem, in particular a Fokker-Planck optimization problem. The field of PDE optimization is vast and we just mention one of many good references, [28]. A series of papers on Fokker-Planck PDE-optimization has also appeared, e.g. [24, 23]. None of these, however, deal with the boundary-based term that arises from the objective based on hitting times. This is done in our previous work [6], where we also solve a hitting time-inspired optimization problem. The fundamental difference is that there we assume model parameters are known, whereas the uncertainty in the model parameters in the present work complicates the problem as it requires the solution of a system of (loosely-) coupled PDEs (the number depending on the form of the prior distribution), rather than just one as in [6]. That work, as well as the present chapter, are part of growing efforts to control nerve cells (see e.g. [93])

The present chapter is also part of the general literature on estimating parameters of diffusion processes from observations only of hitting times. Partly due to the applications of this problem in theoretical neuroscience a lot has been written on the subject, for example in [51, 40, 42, 37], as well as our own work [5] where we deal with the more exotic case of a sinusoidal driving force. The literature on estimation from hitting times of the class of diffusion models we consider asserts that one parameter

in particular, the characteristic time (i.e., the time constant of the LIF model), is the most difficult to estimate, and has by far the widest confidence bounds. Thus, we focus on devising an input control that will provide the most accurate estimates for this parameter, assuming that the other parameters are known or at least nominally known in a sense that we explain below. We also shortly touch upon estimation of all parameters, showing that it is in fact possible to do.

The chapter can be summarized as solving a PDE-optimization arising from maximizing the MI between a prior of the characteristic time parameter and the random (future) hitting-times. Its outline is as follows: in section 5.3 we give the mathematical formulation of the optimal design problem; in section 5.4, we describe the optimization procedure, in particular the derivation of the objective gradient and the numerical methods used; in section 5.5 we discuss the general shape of the optimal solution and its dependence on the various parameters in the model as well as the robustness of the optimization algorithm; in section 5.6, we show the performance of the optimally designed stimulation when many observations are first collected and only a single-pass, *batch* estimation is performed afterwards; in section 5.7, we turn to the *online* problem, where the estimation is performed after each observation and the stimulation is adjusted to reflect the newly updated prior of the unknown parameters; in section 5.8 we make concluding and forward-looking remarks.

5.3 Model

Consider a noisy LIF model. This is a diffusion process that has an absorbing barrier at an upper boundary and is unbounded from below. In particular, for $t \geq 0$ and

$k = 1, 2, \ldots$, it is governed by

$$\begin{aligned} dX(t) &= \left(\alpha(t) + \frac{1}{\tau}(\mu - X(t))\right) \mathrm{d}t + \sigma \, \mathrm{d}W(t), \quad X(0) = 0, \\ X(T_{\mathrm{sp}}) = x_{th} \implies &\begin{cases} X(T_{\mathrm{sp}}{}^{+}) & = 0 \\ t_k & = T_{\mathrm{sp}} \\ k & = k + 1 \end{cases} \end{aligned} \tag{5.3.1}$$

The random variables T_{sp} are the first hitting times of the process on the upper boundary, x_{th}, which is set to $x_{th} = 1$ without loss of generality. At each hitting time, the process is reset to 0 and then continues from there. Here, $T_{\mathrm{sp}}{}^{+}$ denotes the right limit at T_{sp}. The control $\alpha(t)$ is manipulated by us and can be altered at any time. The parameter set, $\theta = \{\mu, \tau, \sigma\}$, is assumed to be constant. For notational convenience, we denote the drift by $U(x, t) = -(\alpha(t) + (\mu - x)/\tau)$.

In the context of parameter estimation, we assume that the parameter set $\theta = \{\mu, \tau, \sigma\}$ or a subset thereof is unknown and only the hitting times, $\{t_k\}$, are observable. First we consider only estimating the characteristic time τ, assuming μ and σ known. The literature on estimating parameters of LIF models from hitting times, e.g. [40, 37], have at length discussed that the estimation of τ is clearly the hardest. In fact, the literature is not clear about whether τ is hard to estimate (practically unidentifiable, needing unrealistically large sample sizes) or whether it is structurally unidentifiable (hitting times contain no information on τ, which can therefore never be estimated, no matter the sample size). Since these studies were done in the context of no control, $\alpha = 0$, or at least α constant, we posit that a judicious choice for the shape of α can significantly improve the estimation of τ, so that it is in effect identifiable, also for realistic sample sizes.

The goal is to choose the control, $\alpha(t)$, such as to best estimate τ given that only the spike times $\{t_k\}$ are observed. Equivalently, setting $t_0 = 0$, we observe only the inter-spike intervals, $\{s_k\}$, defined as the differences between consecutive spike times,

$s_k = t_k - t_{k-1}$.

The probability density of the length of the kth interval, which is a random variable denoted by S_k, conditional on an applied control $\alpha(\cdot)$, is denoted by

$$g_k(s) := \frac{1}{\mathrm{d}s}\mathbb{P}[S_k \in [s, s+\mathrm{d}s) \,|\, \alpha(\cdot)]. \tag{5.3.2}$$

We drop the index k when there is no confusion, and we sometimes write $g(s|\theta, \alpha(\cdot))$ to emphasize the dependence on the parameters and the control. The likelihood of the observed inter-spike intervals, $\{s_k\}_{k=1,\dots,n}$, is then

$$L(\{s_k\}_{k=1,\dots,n}|\theta, \alpha(\cdot)) = \prod_{k=1}^{n} g(s_k|\theta, \alpha(\cdot)), \tag{5.3.3}$$

and the maximum likelihood estimator of θ is

$$\hat{\theta} = \arg\max_{\theta} L(\{s_k\}_{k=1,\dots,n}|\theta, \alpha(\cdot)) \tag{5.3.4}$$

for a given applied control and resulting spike train.

The transition density for the state variable, X_t, $t \in [s_k, s_{k+1})$, is denoted by

$$f(x,t) := \frac{1}{\mathrm{d}x}\mathbb{P}\left[X_t \in [x, x+\mathrm{d}x) \,|\, X_{s_k} = 0, X_s < 1 \text{ for } s_k < s < t\right]. \tag{5.3.5}$$

The transition density satisfies a Fokker-Planck PDE over the domain, $x \in (-\infty, x_{th}]$, which we approximate with the domain $x \in (x_-, x_{th}]$, for some lower boundary x_- such that the probability that the process takes values smaller than this is negligible. We obtain

$$\begin{aligned} \partial_t f(x,t) &= \tfrac{1}{2}\sigma^2\partial_x^2 f + \partial_x\Big(U(x,t)\, f\Big) \\ &= -\partial_x \phi(x,t) \end{aligned}$$

$$\begin{cases} f(x,0) &= \delta(x) \\ \left(\tfrac{1}{2}\sigma^2\partial_x f + Uf\right)|_{x=x_-} &= 0 \\ f|_{x=x_{th}} &= 0. \end{cases} \tag{5.3.6}$$

We have imposed *absorbing* boundary conditions at x_{th} and *reflecting* boundary conditions at x_-. The time-dependent probability flux at the threshold, $\phi(x_{th}, t)$, is

important as it is related to the density of hitting times, eq. (5.3.2), via

$$g(t) = \phi(x_{th}, t) = -\frac{1}{2}\sigma^2 \partial_x f|_{x=x_{th}}, \tag{5.3.7}$$

since $Uf = 0$ at the absorbing boundary, x_{th}.

To see why eq. (5.3.7) is true, note that the integral of $f(x,t)$ over the domain (x_-, x_{th}) equals the probability that X has not left the domain until time t. This is related to g through:

$$\Pr[T_{\text{sp}} > t] = 1 - \int_0^t g(s)\,\mathrm{d}s = \int_{x_-}^{x_{th}} f(\xi, t)\,\mathrm{d}\xi.$$

Differentiating this expression with respect to t, we obtain

$$g(t) = -\partial_t \int_{x_-}^{x_{th}} f(\xi,t)\,\mathrm{d}\xi = \int_{x_-}^{x_{th}} \partial_x \phi(\xi,t)\,\mathrm{d}\xi = \phi(x,t)\Big|_{x_-}^{x_{th}}.$$

This gives eq. (5.3.7), since $\phi(x_-, t) = 0$ is our *reflecting* boundary condition in eq. (5.3.6), see [94] for details.

Intuitively, we wish to find a measure that quantifies the information the sampled spike train carries about the parameter as a function of the control. We take a Bayesian approach and assume some *prior distribution* of the random variable Θ over possible values of θ. We take a discrete prior distribution and denote it by

$$\rho(\theta) \;:=\; \mathbb{P}(\Theta = \theta).$$

Given a single observation, T_{sp}, the parameter posterior distribution is

$$p(\theta | T_{\text{sp}}; \alpha) = \frac{g(T_{\text{sp}}|\theta;\alpha)\rho(\theta)}{\int_\Theta g(T_{\text{sp}}|\theta;\alpha)\rho(\theta)\,\mathrm{d}\theta} \tag{5.3.8}$$

where $g(T_{\text{sp}}|\theta;\alpha)$ is the likelihood of the hitting time given in eq. (5.3.2). We then choose the control, $\alpha(\cdot)$, that maximizes the MI between the two random variables, Θ and T_{sp}. Conditional on $\alpha(\cdot)$, the MI, I, is given by

$$I[\alpha] = \int_\Theta \int_0^\infty g(t|\theta)\rho(\theta) \log\left(\frac{g(t|\theta)}{\int_\Theta g(t|\theta)\rho(\theta)\,\mathrm{d}\theta}\right) \mathrm{d}t\,\mathrm{d}\theta, \tag{5.3.9}$$

see section 2.3.4 for the derivation. For different controls the MI will be different since the hitting time density depends on the shape of α, whereas the prior, ρ, does not. Note that the Bayesian approach is only used to choose the control and thus obtain a sample spike train. The estimator is still the standard maximum likelihood estimator eq. (5.3.4), and can take any value, including values which are not in the support of the prior nor in that of the posterior distribution. This will become important later, when a simple discrete prior distribution will be used and shown to be sufficient to find a control that improves the estimation over the non-controlled case. A discrete prior on only a few values of θ, in contrast to a continuous distribution over the entire parameter space, is chosen for computational reasons.

To summarize, we seek the control input $\alpha(\cdot)$ which maximizes I in eq. (5.3.9), and we then verify that observations obtained under such optimal stimulation lead to parameter estimates that are more accurate and more precise than those obtained for observations when the system is perturbed by sub-optimal stimulations, or no stimulation at all.

5.4 Maximizing the Mutual Information

Consider the optimization problem of maximizing the MI in eq. (5.3.9),

$$\alpha(\cdot) = \underset{\text{admissible } \alpha}{\arg\max}\ I[\alpha].$$

We consider as 'admissible' controls those that are continuous and bounded, $\alpha(t) \in [\alpha_{\min}, \alpha_{\max}]$ for all t. First we calculate the gradient of I with respect to α, and then use it in a standard gradient-based optimization procedure. There is the added complexity that the gradient here is an infinite-dimensional object, which raises some subtle functional analysis questions from a purely mathematical point-of-view; we refer to the references [26, 28] for rigorous justification of our manipulations.

5.4.1 Calculation of the Gradient using the Adjoint Method

A standard approach of deriving a gradient of a functional of a solution to a PDE with respect to an input is the *adjoint* method. We first rewrite eq. (5.3.9) in terms of the transition density using eq. (5.3.7),

$$I[\alpha] = -\int_\Theta \int_0^\infty \partial_x f_\theta(x_{th}, t|\theta)\rho(\theta) \log\left(\frac{\partial_x f_\theta(x_{th}, t|\theta)}{\int_\Theta \partial_x f_\theta(x_{th}, t|\theta)\rho(\theta)\,\mathrm{d}\theta}\right) \mathrm{d}t\,\mathrm{d}\theta, \tag{5.4.1}$$

where the constant term $\sigma^2/2$ is dropped as it is irrelevant for obtaining the maximizing value of α. The objective, I, is augmented with the dynamics of f by introducing the *adjoint* variable p:

$$I = -\int_\Theta \int_0^\infty \partial_x f_\theta(x_{th}, t|\theta)\rho(\theta) \log\left(\frac{\partial_x f_\theta(x_{th}, t|\theta)}{\int_\Theta \partial_x f_\theta(x_{th}, t|\theta)\rho(\theta)\,\mathrm{d}\theta}\right) \mathrm{d}t\,\mathrm{d}\theta \tag{5.4.2}$$

$$+ \int_\Theta \rho(\theta) \int_0^\infty \int_{x_-}^{x_{th}} p_\theta(\partial_t f_\theta + \partial_x \phi)\,\mathrm{d}x\,\mathrm{d}t\,\mathrm{d}\theta, \tag{5.4.3}$$

where ϕ is the probability flux, see eq. (5.3.6).

The added eq. (5.4.3) is always zero for all choices of adjoint variable p_θ since f satisfies the PDE in eq. (5.3.6). However, by adding this null term and choosing p_θ in an appropriate way, we can remove the dependence of I on first-order perturbations of f. This allows us to compute the differential of I with respect to α, without needing to additionally compute the differential of f with respect to α as the chain rule would otherwise require. This is useful, since the differential of f with respect to α is a complicated mathematical object because any value of $f(x,t)$ depends on the entire history of $\alpha(\cdot)$, i.e., on all values $\alpha(s)$ for $s \leq t$.

The elimination of I's dependence on f is done by transferring the derivatives of f to p using repeated integration-by-parts. For completeness, we demonstrate in detail how this is done, dropping θ from the notation. First, we introduce a finite

(large) final time t_f where we can approximate $f(x, t_f) = 0$. Then

$$\int_0^{t_f}\int_{x_-}^{x_{th}} p\,[\partial_t f + \partial_x \phi]\,\mathrm{d}x\,\mathrm{d}t = \int_{x_-}^{x_{th}} p\,f\Big|_0^{t_f}\mathrm{d}x - \int_0^{t_f}\int_{x_-}^{x_{th}} \partial_t p\,f\,\mathrm{d}x\,\mathrm{d}t + \int_0^{t_f} p\,\phi\Big|_{x_-}^{x_{th}}\mathrm{d}t - \int_0^{t_f}\int_{x_-}^{x_{th}} \partial_x p\,\phi\,\mathrm{d}x\,\mathrm{d}t.$$

Using that $\phi = -Uf - \frac{1}{2}\sigma^2\partial_x f$ yields

$$-\int_0^{t_f}\int_{x_-}^{x_{th}} \partial_x p\,\phi\,\mathrm{d}x\,\mathrm{d}t = \int_0^{t_f}\int_{x_-}^{x_{th}} \partial_x p\,U f\,\mathrm{d}x\,\mathrm{d}t + \int_0^{t_f}\frac{1}{2}\sigma^2\partial_x p\,f\Big|_{x_-}^{x_{th}}\mathrm{d}t - \int_0^{t_f}\int_{x_-}^{x_{th}}\frac{1}{2}\sigma^2\partial_x^2 p f\,\mathrm{d}x\,\mathrm{d}t.$$

Some simplifications can be made based on the terminal and boundary conditions for f. Since the initial conditions for f are fixed so that perturbations of f at $t = 0$ do not exist, perturbations of the control, α, do not perturb the initial conditions of f. The same is true for $t = t_f$. Thus, we disregard any terms involving $f(x, 0)$ or $f(x, t_f)$ since they are constant in α, and therefore vanish when taking derivatives. Furthermore, $\phi(x_-, t) = 0$ and $f(x_{th}, t) = 0$. Let C be a constant, then the adjoint term simplifies to

$$\int_0^{t_f}\int_{x_-}^{x_{th}} p\,[\partial_t f + \partial_x \phi]\,\mathrm{d}x\,\mathrm{d}t = C + \int_0^{t_f}\int_{x_-}^{x_{th}}\left[-\partial_t p - \frac{1}{2}\sigma^2\partial_x^2 p + U\partial_x p\right] f\,\mathrm{d}x\,\mathrm{d}t - \int_0^{t_f}\frac{1}{2}\sigma^2 p(x_{th}, t)\partial_x f(x_{th}, t)\,\mathrm{d}t + \int_0^{t_f}\frac{1}{2}\sigma^2\partial_x p(x_-, t) f(x_-, t)\,\mathrm{d}t.$$

The adjoint term thus breaks down into a constant term, a spatial term giving the (backwards) evolution for the adjoint variable, p, and two terms providing the boundary conditions for p.

Now replace the adjoint term in the original equation for I, eq. (5.4.3),

$$I = \int_\Theta \rho(\theta)\left[-\int_0^{t_f}\partial_x f_\theta(x_{th}, t)\,\log\left(\frac{\partial_x f_\theta(x_{th}, t)}{\int_\Theta \partial_x f_\theta(x_{th}, t)\rho(\theta)\,\mathrm{d}\theta}\right)\mathrm{d}t\right.$$

$$
\begin{aligned}
&+ \int_0^{t_f} \int_{x_-}^{x_{th}} \left[-\partial_t p_\theta - \frac{1}{2}\sigma^2 \partial_x^2 p_\theta + U \partial_x p_\theta \right] f_\theta \,\mathrm{d}x \,\mathrm{d}t \\
&- \int_0^{t_f} \frac{1}{2}\sigma^2 p_\theta(x_{th}, t)\, \partial_x f_\theta(x_{th}, t) \,\mathrm{d}t \\
&+ \int_0^{t_f} \frac{1}{2}\sigma^2 \partial_x p_\theta(x_-, t)\, f_\theta(x_-, t) \,\mathrm{d}t \Bigg] \,\mathrm{d}\theta.
\end{aligned}
$$

Consider the effect of small perturbations, $\delta\alpha$, of the control on our objective,

$$
\delta I_\epsilon = \frac{I(\alpha + \epsilon\delta\alpha) - I(\alpha)}{\epsilon}.
$$

The natural assumption is that given $\alpha + \epsilon\delta\alpha$, there is a corresponding solution to the PDE, $f + \epsilon\delta f$. Taking the limit of $\epsilon \to 0$, we get

$$
\begin{aligned}
\delta I = \int_\Theta \rho(\theta) \int_0^{t_f} \Bigg[& - \partial_x \delta f_\theta(x_{th}, t) \, \log\left(\frac{\partial_x f_\theta(x_{th}, t)}{\int_\Theta \partial_x f_\theta(x_{th}, t) \rho(\theta) \,\mathrm{d}\theta} \right) \\
& - \frac{\partial_x f_\theta(x_{th}, t)}{\partial_x f_\theta(x_{th}, t)} \, \partial_x \delta f_\theta(x_{th}, t) + \frac{\partial_x f_\theta(x_{th}, t) \int_\Theta \rho(\theta) \partial_x \delta f_\theta(x_{th}, t) \,\mathrm{d}\theta}{\int_\Theta \partial_x f_\theta(x_{th}, t) \rho(\theta) \,\mathrm{d}\theta} \\
& + \int_{x_-}^{x_{th}} \left[-\partial_t p_\theta - \frac{1}{2}\sigma^2 \partial_x^2 p_\theta + U \partial_x p_\theta \right] \delta f_\theta \,\mathrm{d}x - \int_{x_-}^{x_{th}} \partial_x p_\theta \, f_\theta \, \delta\alpha \,\mathrm{d}x \\
& - \frac{1}{2}\sigma^2 p_\theta(x_{th}, t) \, \partial_x \delta f_\theta(x_{th}, t) + \frac{1}{2}\sigma^2 \partial_x p_\theta(x_-, t) \, \delta f_\theta(x_-, t) \Bigg] \,\mathrm{d}t \,\mathrm{d}\theta.
\end{aligned} \tag{5.4.4}
$$

Thus, δI depends on both $\delta\alpha$ and δf_θ. In fact, it only depends on $\delta\alpha$ through the third last term, and on δf_θ through all the others. However, we can select p_θ in a judicious matter, such that the dependence on δf vanishes. To eliminate δf on the lower boundary, x_-, we need that

$$
\partial_x p(x_-, t) = 0.
$$

To eliminate the spatial dependence on δf, we enforce that p evolves (backwards) as

$$
\partial_t p + \frac{1}{2}\sigma^2 \partial_x^2 p - U \partial_x p = 0.
$$

Note that

$$
\int_\Theta \rho(\theta) \int_0^{t_f} \left[- \partial_x \delta f_\theta(x_{th}, t) + \frac{\partial_x f_\theta(x_{th}, t) \int_\Theta \rho(\theta) \partial_x \delta f_\theta(x_{th}, t) \,\mathrm{d}\theta}{\int_\Theta \rho(\theta) \partial_x f_\theta(x_{th}, t) \,\mathrm{d}\theta} \right] \mathrm{d}t \,\mathrm{d}\theta
$$

$$= \int_\Theta \rho(\theta) \int_0^{t_f} \left[-1 + \frac{\int_\Theta \rho(\theta)\partial_x f_\theta(x_{th},t)\,\mathrm{d}\theta}{\int_\Theta \rho(\theta)\partial_x f_\theta(x_{th},t)\,\mathrm{d}\theta} \right] \partial_x \delta f_\theta(x_{th},t)\,\mathrm{d}t\,\mathrm{d}\theta$$
$$= 0.$$

Finally, to eliminate the dependence of δI on $\partial_x \delta f_\theta$, we apply the simple boundary condition,

$$p_\theta(x_{th},t) = -\frac{2}{\sigma^2}\log\left(\frac{\partial_x f_\theta(x_{th},t)}{\int_\Theta \partial_x f_\theta(x_{th},t)\rho(\theta)\,\mathrm{d}\theta}\right).$$

In summary, the adjoint variables, p_θ, must satisfy the following adjoint PDE

$$\partial_t p_\theta = -\frac{1}{2}\sigma^2 \partial_x^2 p_\theta + U(x,t;\theta)\partial_x p_\theta$$

$$\begin{cases} p_\theta(x_{th},t) & = -\frac{2}{\sigma^2}\log\left(\frac{\partial_x f_\theta(x_{th},t)}{\int_\Theta \partial_x f_\theta(x_{th},t)\rho(\theta)\,\mathrm{d}\theta}\right) \\ \partial_x p_\theta\big|_{x=x_-} & = 0 \\ p_\theta(x,t_f) & = 0\,. \end{cases} \tag{5.4.5}$$

This finally provides us with a simple equation to calculate the objective gradient. Given p_θ and f_θ, the differential of I in eq. (5.4.4) with respect to the control $\alpha(t)$ is

$$\delta I = \left[-\int_\Theta \rho(\theta)\left(\int_{x_-}^{x_{th}} \partial_x p_\theta\, f_\theta\,\mathrm{d}x \right)\mathrm{d}\theta \right]\delta\alpha. \tag{5.4.6}$$

5.4.2 Gradient Ascent Procedure

Maximization of the MI, eq. (5.4.1), is approached as a gradient-based iterative method, where the infinite-dimensional gradient, δI in eq. (5.4.6), is calculated by some finite-dimensional approximation. The process involves three basic stages:

1. Given the current control iterate, $\alpha_n(t)$, numerically solve for the corresponding f, p from their respective PDEs, eqs. (5.3.6) and (5.4.5).

2. Form the adjoint differential, $\delta I/\delta\alpha$, using eq. (5.4.6).

3. Increment α_{n+1} in the direction of increasing $\delta I/\delta\alpha$

$$\alpha_{n+1}(t) = \alpha_n(t) + \frac{\delta I}{\delta\alpha}(t)\,\Delta\alpha \tag{5.4.7}$$

for some step-size $\Delta\alpha$.

In practice, we need to align the various discretization grids, and the simplest thing to do is to discretize α, f, p at the same time points, t_j. The gradient evaluated at t_j is

$$\frac{\delta I}{\delta \alpha}(t_j) = -\int_\Theta \rho(\theta)\left(\int_{x_-}^{x_{th}} \partial_x p_\theta(t_j, x) f_\theta(t_j, x)\,\mathrm{d}x\right)\mathrm{d}\theta.$$

Equation (5.4.7) is the simplest gradient ascent scheme one can devise. The literature on gradient-based optimization, [95], and the subset on PDE-based gradient optimization, [28], offers more sophisticated schemes, but for simplicity we start with this basic one. An example of a more sophisticated scheme is the nonlinear conjugate-gradient ascent, as recommended in the literature on Fokker-Planck control, [25]. A full description of the optimization algorithm is provided in the appendix, see Algorithm 2.

In general, there is no *a priori* way how to select the terminal time, t_f. For practical purposes we proceed as follows. Given a set of plausible parameters we set t_f large enough for $f \approx 0$, and we further select a sub-interval, $[0,] \subset [0, t_f]$, such that we only seek to optimize over $t \in [0,]$ and for $t >$ we let $\alpha = \alpha_{\max}$. This should ensure that

$$\int_{t>t_f} g(t)\,\mathrm{d}t \ll 1.$$

Finally, the calculation of the posterior distribution should be addressed. For computational reasons, we need to approximate the integral with respect to θ by a sum. For a given $\alpha(t)$, we solve for p, f for a few sampled values of θ from the updated prior distribution $\rho(\theta)$ and their corresponding probabilities. This is equivalent to assuming a discrete or a particle prior, i.e., a sum of weighted Dirac delta masses. In the simplest case, these are N values with equal probability,

$$\rho(\tau) = \sum_i \tfrac{1}{N}\delta(\tau - \tau_i). \tag{5.4.8}$$

5.5 Properties of the Optimal Control

We now investigate the optimization algorithm and the properties of the optimal control. In the calculations below, we set $\mu = 0$, $\sigma = 1$, $t_f = 15$, $= 10$, $\alpha_{\min} = -2$, $\alpha_{\max} = 2$ unless otherwise stated. We use the same numerical scheme, the Crank-Nicholson method, for the integration of the forward and adjoint PDEs as in our previous work, [6], as this is a well-known classic technique for parabolic PDEs.

In the computational neuroscience literature this parameter set for μ, σ corresponds to what is known as the sub-threshold and high-noise regime, [5], as long as the control α equals zero. Increasing the control to $\alpha > 1$ will move it into the qualitatively different super-threshold regime, where the process can hit the threshold even in the absence of noise. In the sub-threshold regime, a non-zero noise parameter is required if any spikes are to occur.

5.5.1 Optimization algorithm

A single control increment is shown in fig. 5.1, while an example of a full optimization ascent is given in fig. 5.2 using an assumed prior with two possible values of τ, namely 1/4 and 4. Intuitively, the control tries to separate the hitting time distributions, such that an observed hitting time can more clearly be attributed to one of the two potential parameter values, τ. For the combination of known parameters, μ, σ and prior ρ, used in fig. 5.2, the optimal control first suppresses firing for an initial time interval and then maximally stimulates firing in the remaining time. It is visually obvious that the result on $g(t|\tau)$ is to go from a case where the two hitting time distributions overlap, to one where they are clearly delineated. Thus, given an observation from the optimally stimulated system, one can more confidently estimate what was the underlying value of τ that produced the observation.

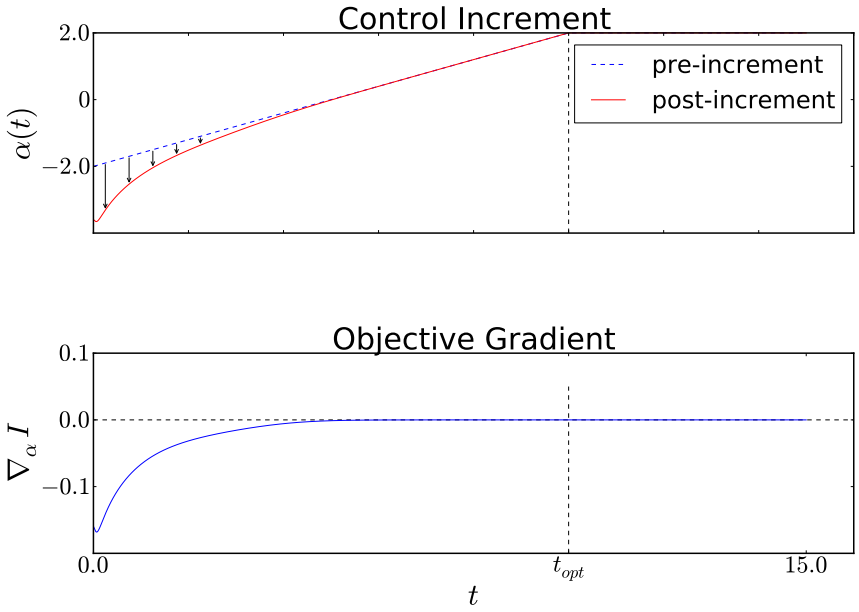


Figure 5.1: Illustration of how the control is incremented given the gradient. Top panel: the initial control in dashed and the incremented control in solid with arrows indicating the incremental change. Bottom panel: the corresponding value of $\delta I/\delta\alpha(t)$. Note that we only optimize $\alpha(t)$ for $t \in [0, t_{opt}]$ and set $\alpha = \alpha_{\max}$ for $t \in [t_{opt}, t_f]$. Also note that the post-increment α exceeds the admissible bounds (here, $[-2, 2]$) for some t. In practice, for those t, the post-increment α will be reset to the closest value within the admissible bounds.

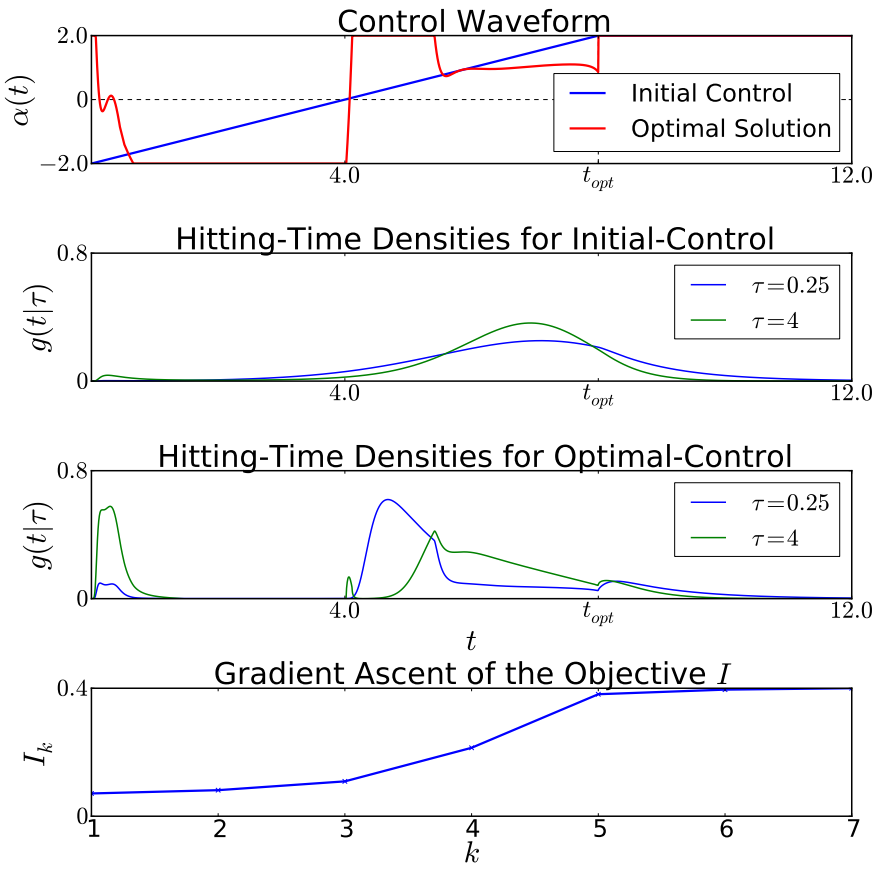


Figure 5.2: Example of the gradient ascent algorithm for the optimization of I in eq. (5.4.1) illustrating that the effect of the optimal control is to separate the hitting time distributions corresponding to the potential values of τ, here for two values of τ. Top panel: initial and optimal controls, $\alpha_0(t)$ and $\alpha_{opt}(t)$. Second panel: hitting time distributions $g(t|\tau)$ conditional on using the initial control, α_0. Third panel: $g(t|\tau)$ conditional on using the optimal control, α_{opt}. Bottom panel: progress of the mutual information, eq. (5.3.9), as a function of the gradient ascent iterations (here the algorithm converged in 9 iterations). The optimal control significantly improves the objective in comparison to the initial guess.

5.5.2 Switch point sweep

Given the optimal solution obtained in fig. 5.2, we suspect that the optimal solution is approximately of a simple switching type, in the sense of first holding everything back and then exciting maximally ('inhibit-excite'). It is interesting to check how sensitive the objective is to the exact value of the switching time, t_{switch}. If it is not very sensitive, it would be practically useful as the exact optimization of the PDE-based objective will not be necessary, and any such switching-type solution will achieve satisfactory improvement in the estimation. We therefore sweep through different points of exactly when the switch from inhibition to excitation occurs and see how it impacts the resulting objective, I. The results are in fig. 5.3. It seems that, at least for this parameter set, there is an improvement in the objective by putting the switching point past $t = 6$, but going much beyond that there is no real difference.

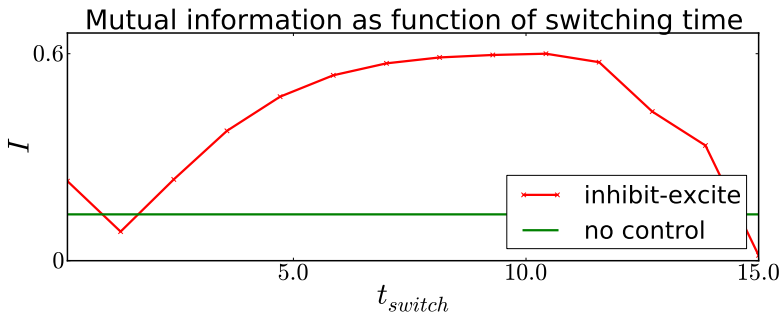


Figure 5.3: Effect of switching time in an inhibit-excite control on the mutual information. The red line is the value of the mutual information as a function of the switching time between maximally-inhibitive to maximally-excitatory control. The blue line, plotted for reference, is the value of the mutual information for no control ($\alpha = 0$ for all t). The same simulation parameters are used as in fig. 5.2.

5.5.3 Sensitivity of the Optimization Method

In this subsection, we explore the effect of the number of points in the prior, the variance of the prior, the initial guess for the control $\alpha(\cdot)$, and the values of the known parameters μ and σ.

Number of points and shape of the prior. It is of interest to know whether the exact shape of the prior matters or whether the first two moments (mean and variance) are sufficient to determine the optimal control. We make a simple experiment where we compute the optimal control for two similar priors, the first using 2 points in the prior and the second using 3 points, but the priors have the same log-mean and log-variance. Results are presented in fig. 5.4, where it is seen that the optimal control is essentially the same. This suggests that, for practical purposes, it suffices to choose a 2-point prior that matches the (log-) variance of the more detailed prior distribution. This reduces substantially the computational burden.

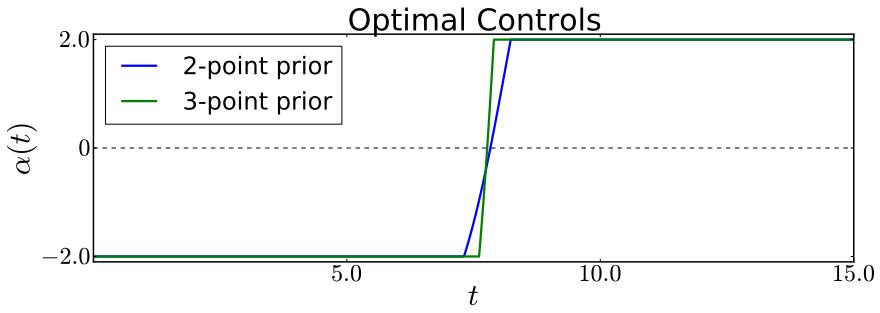


Figure 5.4: Effect of the number of points in the prior on the Optimal Control. The blue curve was calculated with a 2-point prior, the green curve with a 3-point prior. The priors had the same log-mean and log-variance.

The dispersion of the prior. It is also of interest to know how the dispersion of the prior, here quantified by the log-variance, changes the optimal control. Again, we take two 2-point priors, one with particles at $\tau \in \{1/4, 4\}$ and the other at $\tau \in \{3/4, 4/3\}$. Results are shown in fig. 5.5. It is clear that while the general shape of the optimal control is the same, regardless of the width of the prior, there is some difference. We further investigate the practical difference between the two priors in section 5.6.

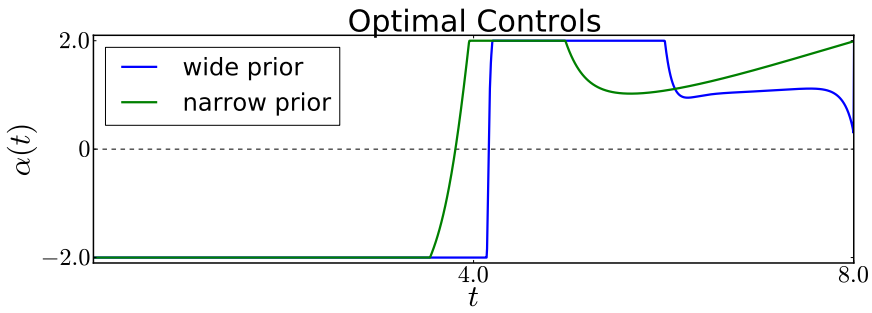


Figure 5.5: Effect of the variance of a 2-point prior on the optimal control. The wide prior has probability 0.5 for both the values $\tau = 1/4$ and 4, while the narrow prior has points positioned much closer to each other, $\tau = 3/4$ or $4/3$.

The initial guess for the control α_0. Like all gradient-based optimization procedures, the algorithm is sensitive to the choice of initial guess for the independent variable, i.e., the choice of α_0. We illustrate this in fig. 5.6. For three different priors, namely a wide, a medium and a narrow prior, we run the optimization scheme from four different initial guesses for α_0:

1. zero for the entire (optimization) time interval;
2. linearly increasing from the lower to the upper control bound over the opti-

mization interval;

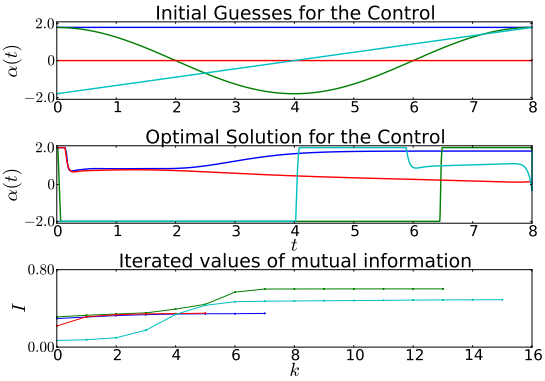
3. a sinusoidal wave from maximum to minimum and again to maximum;

4. maximum for the entire optimization time interval .

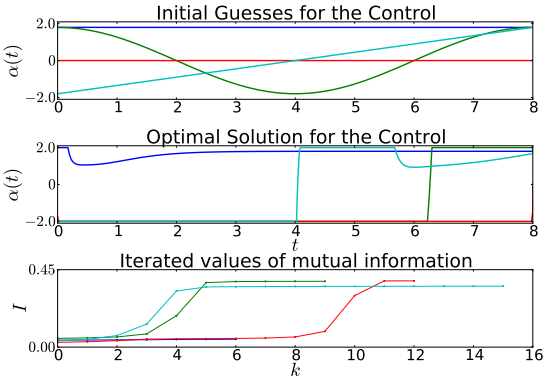
In fig. 5.6, we see that for different initial guesses the optimization routine finds different optimal controls (see middle panels). An initial guess which is non-constant, here linearly increasing or sinusoidal, is better than the constant initial guesses (the MI is larger, see lower panels). It is also seen that the shape of the prior does not have much influence on the optimal control found by the algorithm (the optimal solutions for the same initial guesses but different priors are close, as one can see by comparing the middle panels). The final value of the MI is larger for a wider prior (see lower panels), but this is a scaling issue, since of course the final parameter estimation will be the same if the control is the same. To summarize: the initial guess of the control is important, and different initial guesses should be tried to see if the objective can be improved, whereas the choice of prior distribution is less important.

Values of the known parameters μ, σ. So far we have restricted ourselves to one scenario of the assumed known parameters, $\mu = 0$ and $\sigma = 1$.

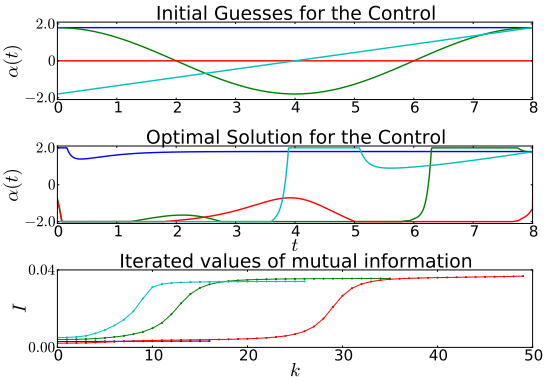
We also redid the basic analyses by varying these parameters, analyzing six qualitatively different sets of parameter values. In all cases the same 2-point prior was used, with probability one half for each of $\tau = 1/2$ and 2. The following parameter sets were investigated: $(\mu, \sigma) = (-0.5, 1), (0.1, 1), (1, 1), (0, 0.3), (0, 0.9)$ and $(0, 1.5)$. In all cases the algorithm succeeded in finding an optimal control (a different one for each pair) that significantly increased the MI relative to that of the initial control guess (results not shown). Although not an exhaustive study, this strongly suggests that our optimal design procedure is applicable and would be effective for a wide range of the values of μ, σ.



(a) Wide prior



(b) Medium prior



(c) Narrow prior

Figure 5.6: Illustration of the dependence of the optimization scheme on the initial guess for the optimal control for (a) a wide prior, (b) a medium prior, and (c) a narrow prior. In the top panels the initial guesses are shown; in the middle panels the corresponding solution obtained by the optimization routine are shown, and in the bottom panels the corresponding evolutions of the mutual information are shown. The various initial conditions considered are: maximum stimulation for all t (blue); sinusoidal stimulation (green); no control at all, $\alpha = 0$, (red) and a linearly increasing control from min to max (cyan)

5.6 Batch Estimation

We now proceed to generate a large sample of hitting times with several types of controlled stimulation and then estimate the unknown parameter after observing the whole sample. We call this batch estimation as opposed to online estimation, where estimates are updated after every single observation, and where the control is updated according to the updated prior distribution. Online estimation is performed in section 5.7.

We assume that the true parameters governing eq. (5.3.1) are as before; $\mu = 0, \tau = 1$ and $\sigma = 1$. We will assume σ, μ known and only estimate the time constant τ, so we will maximize the MI between T_{sp} and τ. To obtain the optimal control, we take a discrete uniform prior on τ, using 10 points uniformly spaced in the interval $[1/4, 4]$, i.e., each point in the prior has probability $1/10$. Note that neither the mean nor the log-mean of the prior correspond to the unknown true τ or $\log(\tau)$.

The obtained control is denoted *opt*. In addition, we consider two optimal controls obtained using a wide and a narrow 2-point prior, as shown in fig. 5.5. We call these *opt-wide* for the prior with $\tau = [1/4, 4]$, and *opt-narrow* for the prior with $\tau = [3/4, 4/3]$. We also use the control *crit* where α is set to the fixed critical value $\alpha = \alpha_{crit} = x_{th}/\tau = 1$. The critical value defines the smallest value for α such that the process converges to the spiking threshold in the absence of noise (i.e., for $\sigma = 0$). Finally, we use a control *max* set to its upper bound, $\alpha = \alpha_{\max}$.

5.6.1 The Estimation Algorithm for the Batch Problem

In this section, we only aim to estimate one parameter, τ, which amounts to single-variable optimization. The negative log-likelihood of an observed hitting time set $\{t_n\}$ is

$$l(\tau) = -\sum_n \log(g(t_n|\tau)) = -\sum_n \log(-\frac{\sigma^2}{2}\partial_x f(t_n|\tau)|_{x_{th}}). \tag{5.6.1}$$

We minimize eq. (5.6.1) using the standard single-variable optimization routine in NumPy based on Brent's method.

5.6.2 Comparison of Estimators

To summarize, we will stimulate the system using 5 stimulation waveforms:

1. *opt* - the optimal control α_{opt}, based on a 10-point uniform prior between $[1/4, 4]$
2. *opt-wide* - the optimal control using a 2-point prior on $[1/4, 4]$
3. *opt-narrow* - the optimal control using a 2-point prior on $[3/4, 4/3]$
4. *crit* - the constant control α_{crit}, $(= x_{th}/\tau)$
5. *max* - the max constant control, $\alpha_{\max}$ $(= 2)$

We simulate N_b blocks of N_s hitting times each for each of the five α's above. We then estimate τ over each set using maximum likelihood with the computed expression for the density, $g(t|\tau; \alpha(t))$. For a fair comparison between controls, we use the same Gaussian random draws per block of N_s hitting times for each control. The estimation results are tabulated in table 5.1, and scatterplots of estimates can be found in fig. 5.7.

Together, table 5.1 and fig. 5.7 confirm a clear advantage to using the optimal control, α_{opt} over the simpler, constant controls. In particular, the bias of the estimates is significantly reduced, except for $N_s = 100$, where the bias is comparable. From fig. 5.7(a) it is seen that this is caused by the presence of a few outlier estimates for the optimal control, but for most data sets, the optimal control is clearly better. These outliers disappear for larger sample sizes.

Comparing the different optimal controls based on various priors, *opt* vs. *opt-wide* or *opt-narrow*, we see that they all perform approximately the same. This confirms the findings of the previous section that the number of points in the discrete prior is

control type	mean($\hat{\tau}$)	std($\hat{\tau}$)
opt	0.887	0.27
opt-wide	0.766	0.32
opt-narrow	0.826	0.30
crit	0.913	0.53
max	0.872	0.37

(a) $N_b = 10^3, N_s = 10^2$

control type	mean($\hat{\tau}$)	std($\hat{\tau}$)
opt	1.011	0.08
opt-wide	0.946	0.20
opt-narrow	1.005	0.07
crit	0.797	0.07
max	0.787	0.06

(b) $N_b = 10^2, N_s = 10^3$

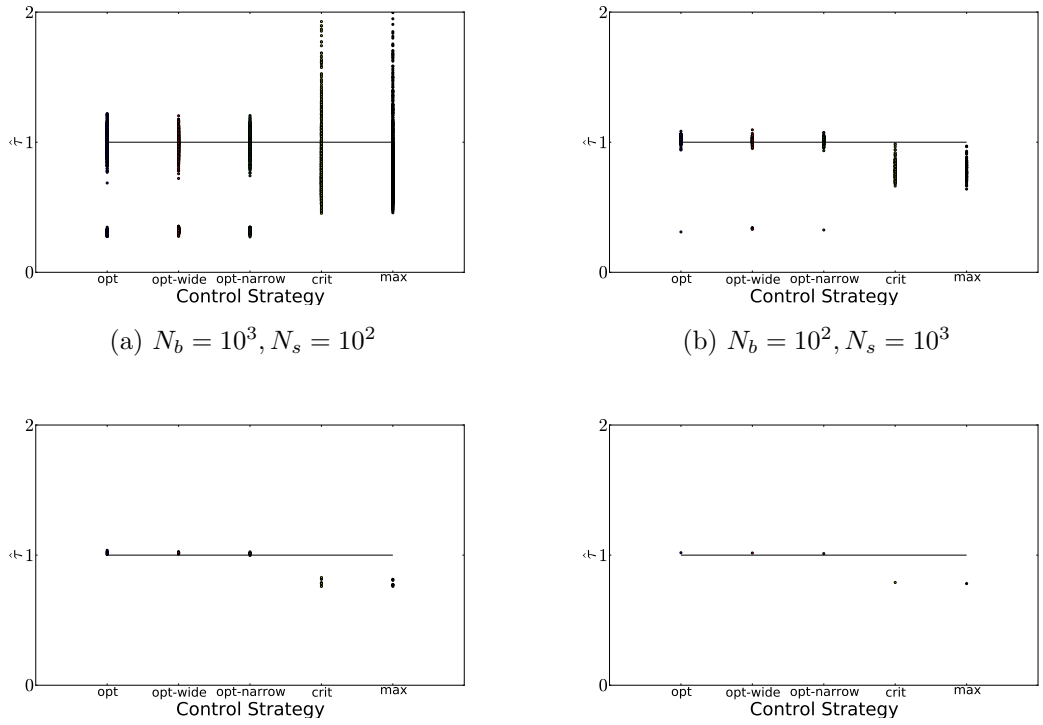
control type	mean($\hat{\tau}$)	std($\hat{\tau}$)
opt	1.018	0.01
opt-wide	1.016	0.01
opt-narrow	1.012	0.01
crit	0.791	0.02
max	0.782	0.02

(c) $N_b = 10, N_s = 10^4$

control type	mean($\hat{\tau}$)	std($\hat{\tau}$)
opt	1.018	-
opt-wide	1.016	-
opt-narrow	1.012	-
crit	0.790	-
max	0.781	-

(d) $N_b = 1, N_s = 10^5$

Table 5.1: Results for the estimates arising from simulations using various values of α (*opt, crit, max*). In each sub-table there are N_b parameter estimates for each distinct α, with N_s hitting times used to form a $\tau-$estimate. The true value of τ is $\tau = 1$. Also see fig. 5.7



(a) $N_b = 10^3, N_s = 10^2$ (b) $N_b = 10^2, N_s = 10^3$

(c) $N_b = 10, N_s = 10^4$ (d) $N_b = 1, N_s = 10^5$

Figure 5.7: Visualization of the maximum likelihood estimates of τ for the different controls. Note that in (d) there is only one estimate per control; all simulated hitting times are used in that estimation. See also table 5.1

not that essential, and computational time can be considerably reduced by choosing priors with only a few points.

5.6.3 Proof of concept for estimating the entire parameter set

So far we have assumed that the drift parameter μ and the variance parameter σ are known. We now relax that assumption and attempt to estimate also these two parameters, in addition to the characteristic time τ. We will still use the optimal control

obtained assuming μ, σ known. Why is this expected to work? First of all we found that the optimal control seems largely insensitive to the exact values of μ, σ as long as they are not too far from the nominal values ($\mu = 0, \sigma = 1$). This is in agreement with the findings of [62], where it is shown that the estimation of the time constant in the Ornstein-Uhlenbeck process from observations of the process itself can be hugely improved by perturbating the process, leaving the estimation of the other parameters largely unaffected. Moreover, as discussed in the Introduction, the τ parameter is well-known to be the hardest to estimate. Thus, a scheme designed to estimate it well is conjectured to help with estimation of the entire parameter set, since reducing fluctuations in one parameter estimate is expected to reduce (or leave unaffected) fluctuations in the others. Moreover, the computational complexity quickly increases when optimizing the MI for 3-dimensional priors, due to the potentially much larger number of PDEs required to be solved.

We use the same simulated hitting time data set as in table 5.1 and fig. 5.7, using $N_s = 10^4$ hits and $N_b = 10$ separate samples, i.e., we make 10 estimates. The results in fig. 5.8 make it clear that the optimally stimulated samples yield much more accurate and precise estimates. In particular, all stimulation schemes come up with high-fidelity estimates for σ; it is in resolving the interplay between μ and τ that the optimal stimulation excels. The constant stimulations do not allow to distinguish a high μ and a small τ from a low μ and a high τ, while the MI-optimizing dynamic stimulation makes it possible.

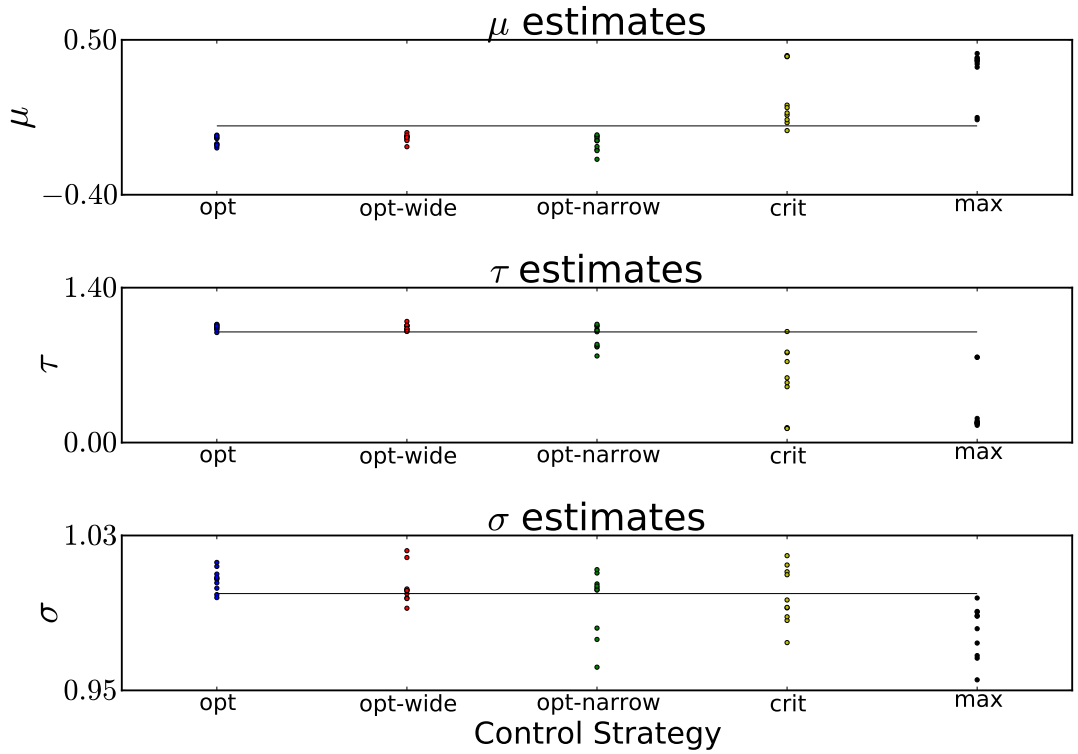


Figure 5.8: Estimation of all parameters. Optimal stimulation greatly improves estimates of all parameters. From the top to bottom, we show $N_b = 10$ estimates of μ, τ and σ, formed after observing $N_s = 10^4$ hitting times per estimate. There are three optimal stimulations (*opt*, *opt-wide* and *opt-narrow*) and two constant ones (*crit* and *max*). The scatter dots are the individual estimates per stimulation-parameter pair. The horizontal line indicates the true value of the parameters, i.e., the value used to simulate the hitting times in the data samples.

5.7 Online Estimation

In the *online* optimization, we proceed in a slightly more complicated way:

1. Find α_{opt} using the gradient ascent for the prior ρ
2. Apply α_{opt} and measure N_s hitting times t_k, $k = 1, 2, \ldots, N_s$
3. Update the ρ into a posterior conditional on the observed $\{t_k\}$
4. Recalibrate α_{opt} using the new ρ, i.e., go back to 1.

We progressively double N_s during the course of the simulation, starting from $N_s = 1$, so we re-compute α_{opt} after the 1st spike, then after the 3rd, 7th and so on. The heuristic is that at the beginning of the experiment the updated prior would be changing more rapidly than after observing several hundred spikes, where the information in data is large, and less affected by incoming new spikes.

Computational efficiency considerations aside, we have all the tools to carry out points 1,2 and 4 above; only the prior update in point 3 needs to be discussed.

5.7.1 Quick Introduction to Particle Filtering

Recall Bayes formula

$$\rho(\theta|\{t_k\}) = \frac{\rho(\theta) \cdot \prod_k g(t_k|\theta;\alpha)}{\int_\Theta \rho(\theta) \cdot \prod_k g(t_k|\theta;\alpha)\,\mathrm{d}\theta}.$$

In practice, the exact calculation of $\rho(\theta|\tau_k)$ would not be possible in our context, so an approximation needs to be made. The standard approach is to approximate the prior distribution by an ensemble of N_p points (particles). We now describe the basic aspect of how the particle ensemble is constructed, how it is updated and how it is resampled. We make use of reference [90], in particular section 4 therein.

Take a prior $\rho(\theta)$, which we represent or approximate via an ensemble of points θ_i by

$$\rho(\theta) \sim \sum_i^{N_p} w_i \delta(\theta - \theta_i).$$

This is what we have been doing up to now for the MI optimization routines. Recall that a Bayesian update, given the kth hitting time, is

$$\rho(\theta|t_k) \propto g(t_k|\theta)\rho(\theta)$$

and thus the weights are iteratively updated as

$$w_i \to w_i g(t_k|\theta_i).$$

Given the particle ensemble, $\{\theta_i, w_i\}$, we can then approximate the mean and variance of Θ by

$$\mathbb{E}[\Theta] \approx \sum_i^{N_p} w_i\theta_i \quad \text{and} \quad \mathbb{V}\text{ar}[\Theta] \approx \sum_i^{N_p} w_i\theta_i\theta_i - (\mathbb{E}[\Theta])^2.$$

In general

$$\mathbb{E}[f(\Theta)] \approx \sum_i^{N_p} w_i f(\theta_i)$$

approximates the expectation of any function, f, of the random variable Θ.

The crux of the update/resample algorithm involves the following. Given a new observation, i.e., the latest hitting time t_k, the weights are updated according to

$$w_{i,k} = w_{i,k-1} \cdot g(t_k|\theta_i, \alpha_k),$$

where $g(|\theta_i, \alpha_k)$ is the probability density given the parameter value θ_i and the chosen stimulation that was applied during the kth sample, $\alpha_k(\cdot)$. The weights are then re-normalized so that at all time

$$\sum_i^{N_p} w_i = 1.$$

The literature suggests that this procedure will tend to concentrate all the mass on one particle. Such concentration is undesirable since, eventually, the distribution has converged artificially to a point which might be the most likely point from the initial ensemble, but still be far from the true value of the parameter. This adverse effect can be mitigated by resampling the *locations* of the particle ensemble, θ_i. This resampling can be done in many ways, but a standard way is described in algorithm 3 (which is essentially the same as Algorithm 4 in Granade et. al. [90], who in turn closely follow [92]). While updating happens after every iteration, re-sampling happens only when

$$\frac{1}{\sum_i^{N_p} w_i^2} < \frac{N_p}{2}.$$

The complete filtering algorithm is provided in Algorithm 3 in the Appendix.

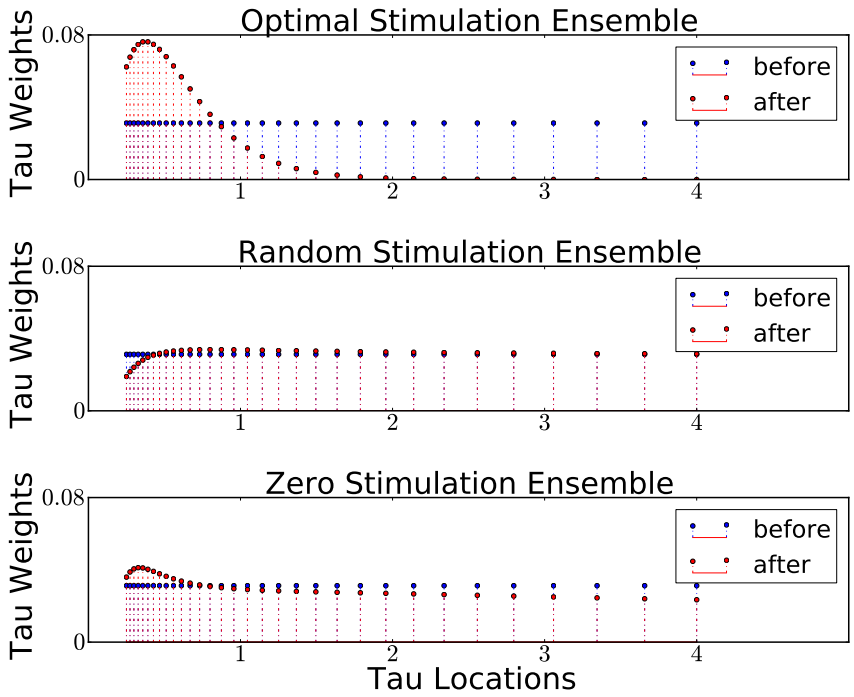
5.7.2 Simulation Results

We now illustrate how the optimal design procedure works while using the particle filtering methodology to represent and update the parameter prior.

Single Hitting time Illustration

In fig. 5.9, we illustrate one iteration of the update, that is one hitting time given a stimulation from either the MI optimal control, or from a random control or from zero control; the random control just gives a randomly chosen constant stimulation for the computation of each hitting time. Lower values of τ imply a higher restoring force of the LIF process towards the origin, and therefore longer time between hitting times. Since in this sample the observed hitting times were fairly long, especially for the MI-optimal stimulation, weights for smaller τ's grow larger, while weights for bigger τ's become smaller. However, it is immediately clear that the MI-optimal stimulation is more discerning as it has almost entirely discarded (correctly) the possibility that

$\tau > 2$, while the other two stimulations have resulted in only mild perturbations of the prior distribution.



(a) Particle Ensemble Updates

Figure 5.9: Examples of a single iteration of the Online Stimulation-Estimation scheme. The actual hitting time observations were 8.16 (optimal stimulation), 1.02 (random stimulation), 7.62 (zero-stimulation)

Full Multiple Hitting time Experiment

Finally, let us compute an entire estimation experiment. We stimulate the process and obtain a sequence of hitting times and online update our parameter prior distribution after every observation. We also online-update our MI-optimal stimulation as the prior evolves after the 1st, 3rd, 7th and so on observation.

We repeat and simulate 50 independent experiments of 500 hitting times each and then average the obtained ensembles of estimates. The aggregated evolutions for the updated prior distributions are visualized in fig. 5.10. We plot the quantiles of the running means of the updated priors for the N experiments. It is immediately clear that the MI optimal control produces more accurate and more precise estimates, and that the increase in precision is especially clear in the earlier parts of the experiment where less information is available. That is, using the MI optimal control protocol reduces the uncertainty in the parameters much faster.

The fact that the base cases of constant controls are neither accurate nor particularly precise is consistent with the general results on the estimation of the characteristic time parameter. The novel finding is that a suitably tuned stimulation can result in much more accurate estimates.

5.8 Discussion

We introduced a method for optimal design given hitting time observations. This contrasts with earlier work where the control is based on observations of the state variable of the SDE whose parameters are to be estimated. Our method is based on maximizing the MI between the observed hitting times and the posterior distribution of the parameters. The optimal control tends to separate the hitting time distributions associated with alternative values of the unknown parameter, thereby facilitating the identification of the parameter once an observation is made. The simulations show that the resulting estimates from the optimally-stimulated system have higher precision and accuracy than sensible alternatives, such as random stimulation, critical stimulation at $\alpha = x_{th}/\tau$, or no stimulation at all.

Since the MI is expressed in terms of the hitting time density and the hitting time density can be related to the boundary term of a Fokker-Planck PDE, we approach the maximization of the MI as a PDE optimization problem. The standard use of

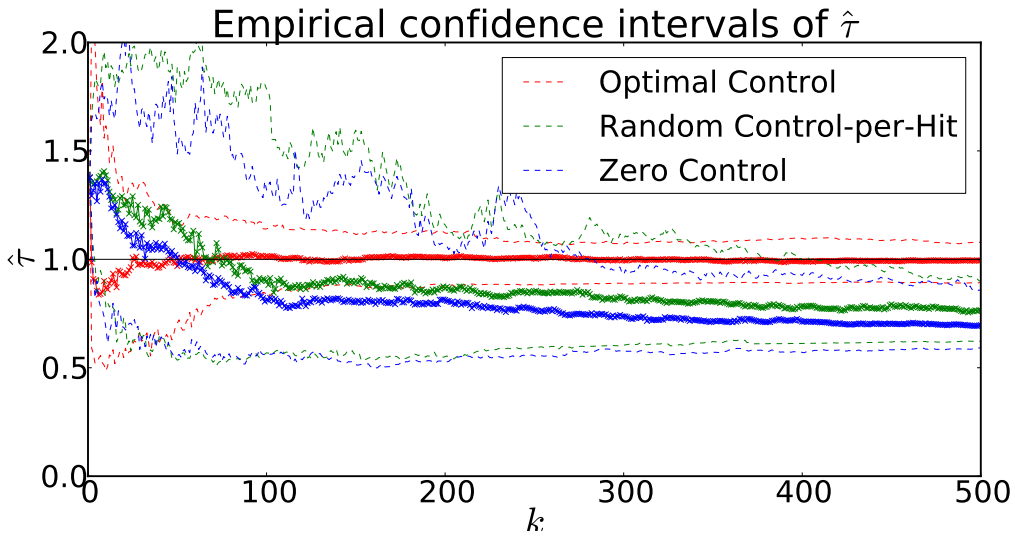


Figure 5.10: Optimal-designed stimulation produces more precise and more accurate parameter estimates. Visualized are the median and 5/95th percentiles of the mean of the updated prior distribution over the $N = 50$ experiments. I.e. for each experiment, j, and each hitting time, k, we compute the mean of the updated prior $\rho_{j,k}(\tau)$ and then we plot the median (crosses) of these N means as well as the third smallest and third-largest of them (dashes). The black line indicates the true value of τ used to generate the synthetic observations. Again we see that the median and the extreme quantiles of the optimally stimulated estimates are much closer to the true value of the parameter, τ, then the estimates obtained from the naive stimulations.

an adjoint variable to obtain the objective gradient is made more complicated by the outer integration with respect to the unknown parameter prior. However, we are still able to derive the adjoint PDEs and thus to compute the objective gradient despite this added complication.

Our PDE-based adjoint optimization methods can be expensive and sensitive to the various parameters of the optimization. The main computational cost is the numerical solution to the forward and backward PDEs, which have to be solved numerous times during the optimization search. In principle, one has to solve as many PDEs as there are particles in the prior distribution. However, we have seen that in practice it is sufficient to use very coarse approximations to the prior. For example as few as two or three particles which match the mean/variance of the more detailed higher-resolution prior seem to be sufficient to obtain the same optimal control as that obtained by using all the particles in the prior, at least for our problem.

In general, the optimally-stimulated samples give (much) more accurate estimates than the ones that are naively-stimulated. However, we see in fig. 5.7, panel b) that occasionally (once), the optimally-stimulated sample can give very wrong estimates. We have taken a closer look at what happens in those situations. It seems that in the bad cases, there are enough extreme (i.e. very long) hitting times that the estimation procedure infers a very small characteristic time (i.e. the attractive force towards $\mu = 0$ is very strong). Generally in long samples such extreme hitting times are relatively rare, and the correct parameter value is inferred.

Our results point to a novel way to estimate the characteristic time of the LIF process. Parameter estimation techniques applied to the hitting time process have reported difficulties for this particular parameter. This is unfortunate given that it is a parameter that is deemed important to understand the integrative properties of single neurons. It is also a parameter that can vary based on the amount of inputs the neuron receives from other cells; more specifically it reflects the ongoing conductance of the cell. It would be particularly interesting to see whether the batch

or online implementations of the hitting time-based optimal design method could be implemented in an experimental setting and what new challenges arise in this context. Our method has the advantage of dealing with the intrinsic stochasticity of the underlying process for designing stimulation protocols that best reveal system parameters.

Doing a batch estimation, when the control is computed only once and then applied for all observations, is the computationally simpler approach. More interestingly, we show that the optimal control can be evolved online as new observations are assimilated into the parameter prior, and the optimal control is updated accordingly. However, we note that the optimal control is fairly similar for a wide range of priors, which somewhat limits the additional value of recomputing the optimal control as the observations from the experiment are recorded.

In line with the observation that the optimal control is fairly similar for various priors of τ, we note that the slight differences in the actually obtained stimulations does not make a big difference when it comes to estimating the parameters. Independent of the prior used, the resulting estimates in all cases are much better, more accurate and precise, than the estimates obtained using the base case of no stimulation. In fact, we empirically demonstrate that the optimal stimulation scheme also results in excellent estimation results for *all* system parameters, even though it is designed only to estimate τ.

In general, we note that the control shape that consists of an initial segment that inhibits spiking, $\alpha(t) \approx \alpha_{\min}$, followed by a segment that promotes spiking, where $\alpha(t) \approx \alpha_{\max}$, is obtained for a wide variety of priors or model parameters, μ, σ. In some contexts, however we have observed a more complex optimal control, where for an initial region that is excitation, then inhibition and then excitation again. It is very much an interesting open problem to determine how exactly these switching times depend on the various model parameters.

Future work could address the case where all the parameters are treated in a

full Bayesian setting and the parameter prior is over all of them, not just over τ. In principle our derivations carry over unchanged. The only potential complication is that the prior will need to contain more points to fully describe the uncertainty in all three parameters. It would also be of interest to test the estimation scheme for other dynamical regimes than the sub-threshold high-noise one studied here. We anticipate that the estimation will improve in supra-threshold and/or lower noise cases.

Other minor tweaks involve using a more sophisticated gradient optimization method, such as a nonlinear conjugate gradient, which might be more computationally efficient. Finally our work can be viewed as an optimal design for parameter estimation of PDEs. It would be interesting to see whether in general the MI criterion can be applied successfully for the selection of perturbation in other PDE parameter estimation contexts.

5.9 Numerical Algorithms

Algorithm 2 Gradient ascent algorithm for obtaining the optimal control. The gradient of the objective with respect to the control is computed up to K_{max} times and at each iteration k, the kth control, $\alpha_k(t)$ is incremented in the direction of the kth gradient in order to achieve an improvement in the objective value $I_{k+1} = I(\alpha_{k+1}) > I_k$.

Fix $t_f, t_{opt}, \mu, \sigma$ the problem parameters

Fix $\{t_n\}_0^{N_t}$ a time-discretization of $[0, t_f]$

Fix g_{tol} a convergence tolerance for the gradient

Fix $K_{\max}, J_{\max}$ number of maximum iterations in outer, inner loops

Fix s the initial step-size for incrementing α.

\# *we use* $g_{tol} = 10^{-5}, K_{\max} = 100, I_{\max} = 10, s = 10$

$\alpha_1(t) \leftarrow (\alpha_{\max} - \alpha_{\min}) \cdot t/t_{opt} + \alpha_{\min}$

\# $\alpha_1(t) \sim$ *initial guess for the control, linear interpolate between* $\alpha_{min}, \alpha_{max}$

for $k = 1 \dots K_{\max}$ **do**

\# *This is the outer loop corresponding to the* k*th ascend up the gradient*

Calculate $f_{\theta,k}, I_k, p_{\theta,k}, \delta_\alpha I_k$ corresponding to α_k from eqs. (5.3.6), (5.4.1), (5.4.5) and (5.4.6)

$N_{active} \leftarrow$ Number of time nodes t_n, where either $\alpha_k(t_n) \neq \{\alpha_{\min}, \alpha_{\max}\}$ or $\delta_\alpha I_k(t_n)$ points inwards

if $||\delta_\alpha I_k||_{\mathbb{R}^{N_{active}}} \leq g_{tol} \cdot N_{active}$ **then**

\# *'Active' gradient is small enough, consider converged:*

BREAK

end if

 # Find the step size, s, for how far to move α in the direction $\delta_\alpha I_k$:

for $j = 1 \dots J_{\max}$ **do**

 # This is the inner loop where we find how much to ascend down the current gradient

$\alpha_{k,j} \leftarrow \alpha_k + s_j \cdot \delta_\alpha I_k$

 # $\alpha_{k,j}$ is the new control strategy to try, we threshold it so it stays within control bounds

Calculate $f_{\theta,k,j}, I_{k,j}$ corresponding to $\alpha_{k,j}$ from eqs. (5.3.6) and (5.4.1)

 # Recall $f_{\theta,k,j}$ is a probability density resulting from the control $a_{k,j}$ and $I_{k,j}$ is the objective value resulting from $f_{k,j}$

if $I_{k,j} > I_k$ **then**

 # We found a better (larger) objective value

$s \leftarrow 2s_j$ *# start from a bigger step in the next iteration*

BREAK

end if

if $j == J_{\max}$ **then**

 # We exhausted the step search without finding a smaller I, return current values

Return I_k, α_k

end if

 # Continue inner loop, try a smaller step, half the size:

$s_{j+1} \leftarrow s_j/2$

end for *# single step (inner) loop*

if $k == K_{\max}$ **then**

ERROR 'Could not converge'

end if

 # Assign the new candidate for the optimal control and re-loop

$\alpha_{k+1} \leftarrow \alpha_{k,j}$

end for *# gradient ascent (outer) loop*

return I_k, α_k

5.10 Code Repository

All the code used in the chapter including code to generate the figures can be found on github @ https://github.com/aviolov/OptEstimatePython. Please refer to the README.md file at the root of the repository.

Algorithm 3 Particle Filtering for Parameter Estimation

Given $\{w_i, \theta_i\}_1^{N_p}$ the current particle ensemble (weight, locations)

 # Observe a single hitting time, t_k from the real (or simulated) system conditional on the applied control $\alpha_k(\cdot)$

 # Update weights with likelihood given t_k and normalize:

$w_i \leftarrow w_i \cdot g(t_k|\theta_i, \alpha_k)$

$w_i \leftarrow w_i / \sum w_i$

if

$$\frac{1}{\sum_i^{N_p} w_i^2} < \frac{N_p}{2}$$

then

 # Re-sample the ensemble:

$\mu \leftarrow \mathbb{E}[\Theta] = \sum w_i \theta_i$

$a = 0.98$, see [90, 92]

$h = \sqrt{1 - a^2}$ i.e. $h \approx 0.1990$

$\Xi \leftarrow \mathrm{h}^2 \cdot \mathbb{V}\mathrm{ar}[\Theta]$

 # Re-sample each particle individually:

for $i = 1 \ldots N_p$ **do**

draw j with probability w_j

$\mu_i \leftarrow \mathrm{a}\ \ \theta_j + (1 - a)\mu$

Resample θ_i from $N(\mu_i, \Xi)$

$w_i \leftarrow 1/\mathrm{N}_p$

end for *#$1 \ldots N_p$ particle resampling*

end if *Conditional Resampling*

return w_i, θ_i the updated and possibly-resampled particle ensemble

Chapter 6

Conclusion and Outlook

The three core chapters of this thesis all explore the first-passage times of an SDE. In the first one, we use the observations of the first-passage times to estimate the parameters governing the SDE. In particular we explore the difficult problem when there is a sinusoidal driving force that renders the observations non-identically distributed. Through a basic approximation we can still apply known techniques devised in the identically-distributed case, and demonstrate that parameter identification can still be performed. We also quantify how the estimation depends on the dynamical regime of the SDE, namely low-vs-high bias and low-vs-high noise.

In the second core chapter, we aim to target the hitting times of the stochastic system to some desired sequence. Depending on whether we can observe or not the evolving trajectory of the state, we apply different optimization techniques.

In the third core chapter, we seek a perturbation (i.e. a time-dependent control) that makes the resulting hitting time observations 'most informative' about the SDE parameters. Informally, this amounts to devising a control that will best separate the 'hitting-time' densities that corresponding to different parameters.

The basic mathematical concepts in all three cases are the relations between the hitting-times density, g and the state-transition density, f. Since f is governed

by a PDE, the algorithms both for estimation and control often stem from PDE-optimization techniques.

The three new directions set forth in this thesis follow a fairly natural level of progression in complexity. In the first, already known techniques are extended to the newer time-inhomogeneous setting. In the second problem, the PDE-optimization techniques used can be seen as novel to the neuroscience context, especially due the particular boundary term that arises due to the hitting-time nature of the problem. The third direction can be seen as the most novel, as the PDE-optimization itself is not-standard due to the 'uncertainty' in the parameters. The belief distribution appearing in the objective makes for a new adjoint-derivation, and the coupled backwards system can, to our knowledge, be considered novel not just in mathematical neuroscience, but in PDE-optimization in general.

The progression of complexity in these three core chapters also means that the avenues for future research differ in scope. In the case of estimating parameters in a non-autonomous SDE from its hitting times, one can see that there remains at least one interesting problem, namely why the estimation quality degrades with higher sinusoidal frequency, in particular for one of the two methods (the Fokker-Planck / Likelihood based method).

In the second case of targeting spike-times, much more ambitious goals can be set, for example one could consider multi-state control, that is the control of spike-sequence in coupled cells. This will require a breakthrough as the computations in higher dimensions become exponentially more time-consuming.

The third case involving stimulation of an unknown system with the goal of identifying its parameters (assuming a basic underlying dynamical form) has possibly the highest potential both in the context of mathematical neuroscience and in machine learning in general. As in the spike-targeting problem, here the possibility of extending the technique to several coupled neurons also rises serious mathematical challenges in particular of computational nature. Certainly solving the problem for

multiple coupled cells would not be a trivial extension of the techniques presented herein. But one can hope that the mathematics and accompanying computational tools laid out in this thesis can form a solid basis for doing so.

Bibliography

[1] Eugene M Izhikevich: *Dynamical Systems in Neuroscience*. Cambridge, MA: The MIT Press 2010.

[2] Stein RB, Gossen ER, Jones KE: **Neuronal variability: noise or part of the signal?** *Nature reviews. Neuroscience* 2005, **6**(5):389–97.

[3] Wilson D, Moehlis J: **A Hamilton-Jacobi-Bellman approach for termination of seizure-like bursting.** *Journal of computational neuroscience* 2014, **37**(2):345–55.

[4] Schiff SJ: *Neural Control Engineering: The Emerging Intersection Between Control Theory and Neuroscience*. MIT Press 2012.

[5] Iolov A, Ditlevsen S, Longtin A: **Fokker-Planck and Fortet Equation-Based Parameter Estimation for a Leaky Integrate-and-Fire Model with Sinusoidal and Stochastic Forcing**. *Journal of mathematical neuroscience* 2014, **4**:1–34.

[6] Iolov A, Ditlevsen S, Longtin A: **Stochastic optimal control of single neuron spike trains**. *Journal of neural engineering* 2014, **11**(4):046004.

[7] Hady AE (Ed): *Closed Loop Neuroscience*. Elsevier Science 2016.

[8] Iolov A, Ditlevsen S, Longtin A: **Optimal Design for Estimation in Diffusion Processes from First-Hitting-Times** 2015.

[9] Oksendal BK: *Stochastic Differential Equations: An Introduction with Applications*. Springer-Verlag, 6 edition 2007.

[10] Fleming WH, Rishel RW: *Deterministic and stochastic optimal control*. Springer 1975.

[11] Evans LC: **An Introduction to Stochastic Differential Equations, version 1.2**. Tech. rep.

[12] Evans LC: **An Introduction to Mathematical Optimal Control Theory, version 0.2**. Tech. rep.

[13] Jacobs K: *Stochastic processes for physicists: understanding noisy systems*. Cambridge: Cambridge University Press 2010.

[14] Higham D: **An algorithmic introduction to numerical simulation of stochastic differential equations**. *SIAM review* 2001, **43**(3):525–546.

[15] Gerstner W, Kistler W, Naud R, Paninski L: *Neuronal dynamics: from single neurons to networks and models of cognition*. Cambridge: Cambridge University Press, 1 edition 2014.

[16] Laing C, Lord GJ (Eds): *Stochastic Methods in Neuroscience*. Oxford University Press, USA 2009.

[17] Fortet R: **Les fonctions aléatoires du type de Markoff associée à certaines équations linéaires aux dérivées partielles du type parabolique**. *J. Math. Pures Appl* 1943, **9**(22):177–243.

[18] Pukelsheim F: *Optimal Design of Experiments*. Society for Industrial and Applied Mathematics 2006.

[19] MacKay DJC: *Information Theory, Inference and Learning Algorithms*. Cambridge University Press 2003.

[20] Cohn DA, Ghahramani Z, Jordan MI: **Active Learning with Statistical Models** 1996, **4**:129–145.

[21] Settles B: **Active Learning Literature Survey**. Tech. rep. 2009.

[22] Seeger MW: **Bayesian Inference and Optimal Design for the Sparse Linear Model** 2008, **9**:759–813.

[23] Annunziato M, Borzì A, Nobile F, Tempone R: **On the Connection between the Hamilton-Jacobi-Bellman and the Fokker-Planck Control Frameworks**. *Applied Mathematics* 2014, (September):2476–2484.

[24] Annunziato M, Borzì A: **Optimal control of probability density functions of stochastic processes**. *Mathematical Modelling and Analysis* 2010, **15**(4):393–407.

[25] Annunziato M, Borzì A: **A Fokker-Planck control framework for multidimensional stochastic processes**. *Journal of Computational and Applied Mathematics* 2013, **237**:487–507.

[26] Lenhart S, Workman JT: *Optimal Control Applied to Biological Models (Chapman & Hall/CRC Mathematical & Computational Biology)*. Chapman and Hall/CRC 2007.

[27] Fattorini HO: *Infinite Dimensional Optimization and Control Theory (Encyclopedia of Mathematics and its Applications)*. Cambridge University Press 1999.

[28] Borzì A, Schulz V: *Computational optimization of systems governed by partial differential equations*. Philadelphia: Society for Industrial and Applied Mathematics 2012.

[29] Haussmann UG: *A stochastic maximum principle for optimal control of diffusions*. New York: Wiley 1986.

[30] Krylov NV: *Controlled Diffusion Processes*. Springer 2008.

[31] Fleming WH, Soner H: *Controlled Markov processes and viscosity solutions*, *Volume 25 of* Stochastic Modelling and Applied Probability. New York: Springer-Verlag 2006.

[32] Press WH, Flannery BP, Teukolsky SA, Vetterling WT: *Numerical Recipes in C: The Art of Scientific Computing, Second Edition*. Cambridge University Press 1992.

[33] Brillinger DR: **Maximum likelihood analysis of spike trains of interacting nerve cells**. *Biological Cybernetics* 1988, **59**:189–200.

[34] Inoue J, Sato S, Ricciardi LM: **On the parameter estimation for diffusion models of single neuron's activities**. *Biological Cybernetics* 1995, **73**:209–221.

[35] Ditlevsen S, Lánský P: **Estimation of the input parameters in the Ornstein-Uhlenbeck neuronal model**. *Physical Review E* 2005, **71**:11907.

[36] Ditlevsen S, Lánský P: **Estimation of the input parameters in the Feller neuronal model**. *Physical Review E* 2006, **73**:61910.

[37] Mullowney P, Iyengar S: **Parameter estimation for a leaky integrate-and-fire neuronal model from ISI data**. *J. Comput. Neurosci.* 2008, **24**(2):179–194.

[38] Zhang X, You G, Chen T, Feng J: **Maximum likelihood decoding of neuronal inputs from an interspike interval distribution**. *Neural Computation* 2009, **21**(11):3079–3105.

[39] Ditlevsen S, Ditlevsen O: **Parameter estimation from observations of first-passage times of the Ornstein-Uhlenbeck process and the Feller process**. *Probabilistic Engineering Mechanics* 2008, **23**:170–179.

[40] Ditlevsen S, Lánský P: **Parameters of stochastic diffusion processes estimated from observations of first-hitting times: Application to the leaky integrate-and-fire neuronal model**. *Physical Review E* 2007, **76**:41906.

[41] Ditlevsen S, Lánský P: **Comparison of statistical methods for estimation of the input parameters in the Ornstein-Uhlenbeck neuronal model from first-passage times data**. In *Collective Dynamics: Topics on Competition and Cooperation in the Biosciences*, *Volume CP1028 of* American Institute of Physics Proceedings Series. Edited by LM Ricciardi, A Buonocore, E Pirozzi, American Institute of Physics, Melville, NY 2008:171–185.

[42] Lánský P, Ditlevsen S: **A review of the methods for signal estimation in stochastic diffusion leaky integrate-and-fire neuronal models.** *Biological Cybernetics* 2008, **99**(4-5):253–62.

[43] Braun HA, Wissing H, Schafer K, Hirsch MC: **Oscillation and noise determine signal-transduction in shark multimodal sensory cells**. *Nature* 1994, **367**(6460):270–273.

[44] Chacron MJ, Longtin A, St-Hilaire M, Maler L: **Suprathreshold Stochastic Firing Dynamics with Memory in P-Type Electroreceptors**. *Physical Review Letters* 2000, **85**(7):1576–1579.

[45] Bulsara A, Elston TC, Doering CR, Lowen SB, Lindenberg K: **Cooperative behavior in periodically driven noisy integrate-fire models of neuronal dynamics**. *Physical Review E* 1996, **53**(4, Part b):3958–3969.

[46] Burkitt A: **A review of the integrate-and-fire neuron model: II. Inhomogeneous synaptic input and network properties.** *Biological Cybernetics* 2006, **95**(2):97–112.

[47] Lánský P: **Sources of periodical force in noisy integrate-and-fire models of neuronal dynamics**. *Physical Review E* 1997, **55**(2):2040–2043.

[48] Longtin A, Bulsara A, Pierson D, Moss F: **Bistability and the dynamics of periodically forced sensory neurons**. *Biological Cybernetics* 1994, **70**(6):569–578.

[49] Sacerdote L, Giraudo MT: **Stochastic Integrate and Fire Models: a review on mathematical methods and their applications**. In *Stochastic Biomathematical Models with Applications to Neuronal Modeling*, *Volume 2058 of* Lecture Notes in Mathematics. Edited by Bachar M, Batzel J, Ditlevsen S, Springer-Verlag Berlin Heidelberg 2013.

[50] Shimokawa T, Pakdaman K, Takahata T, Tanabe S, Sato S: **A first-passage-time analysis of the periodically forced noisy leaky integrate-and-fire model.** *Biological Cybernetics* 2000, **83**(4):327–40.

[51] Paninski L, Pillow J, Simoncelli E: **Maximum likelihood estimation of a stochastic integrate-and-fire neural encoding model**. *Neural computation* 2004, **16**(12):2533–61.

[52] Dong Y, Mihalas S, Russell A, Etienne-Cummings R, Niebur E: **Estimating parameters of generalized integrate-and-fire neurons from the maximum likelihood of spike trains.** *Neural Computation* 2011, **23**(11):2833–67.

[53] Sirovich L, Knight B: **Spiking neurons and the first passage problem.** *Neural Computation* 2011, **23**(7):1675–703.

[54] Longtin A: **Mechanisms of stochastic phase locking**. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 1995, **1**(5):209–215.

[55] Alili L, Patie P, Pedersen J: **Representations of the First Hitting Time Density of an Ornstein-Uhlenbeck Process**. *Stochastic Models* 2005, (21):967–980.

[56] Hurn A, Jeisman J, Lindsay K: **ML estimation of the parameters of SDE's by numerical solution of the Fokker-Planck equation**. In *MODSIM 2005: International Concgress on Modelling and Simulation: Advances and Applications for Management and Decision Making* 2005:849–855.

[57] Karniadakis GE, Kirby RM: *Parallel Scientific Computing in C++ and MPI*. Cambridge: Cambridge University Press 2003.

[58] Shimokawa T, Pakdaman K, Sato S: **Time-scale matching in the response of a leaky integrate-and-fire neuron model to periodic stimulus with additive noise**. *Physical Review E* 1999, **59**(3):3427–3443.

[59] Lo C, Chung TK: **First passage time problem for the Ornstein-Uhlenbeck neuronal model**. In *13th International Conference on Neural Information Processing* 2006:324–333.

[60] Jones E, Oliphant T, Peterson P, et al.: **SciPy: Open source scientific tools for Python** 2001–.

[61] Galasi M: *GNU Scientific Library Reference Manual - Third Edition*. Network Theory Ltd., 3 edition 2009.

[62] Ditlevsen S, Lánský P: **Only through perturbation can relaxation times be estimated**. *Physical Review E, Rapid Communications* 2012, **86**(5):050102(R).

[63] Jahn P, Berg RW, Hounsgaard J, Ditlevsen S: **Motoneuron membrane potentials follow a time inhomogeneous jump diffusion process**. *Journal of Computational Neuroscience* 2011, **31**:563–579.

[64] Lee S: **Optimal Drift on [0, 1]**. *Transactions of the American Mathematical Society* 1994, **346**:159–175.

[65] Lefebvre M: **Optimal control of an Ornstein-Uhlenbeck process**. *Stochastic Processes and their Applications* 1987, **24**:89–97.

[66] Danzl P, Hespanha J, Moehlis J: **Event-based minimum-time control of oscillatory neuron models: phase randomization, maximal spike rate increase, and desynchronization.** *Biological cybernetics* 2009, **101**(5-6):387–99.

[67] Nabi A, Moehlis J: **Time optimal control of spiking neurons.** *Journal of mathematical biology* 2012, **64**(6):981–1004.

[68] Wu Y, Peng J, Luo M: **First spiking dynamics of stochastic neuronal model with optimal control**. In *Advances in Neuro-Information Processing 15th International Conference, ICONIP 2008, Auckland, New Zealand, November 25-28, 2008, Revised Selected Papers*, Springer Berlin Heidelberg 2009:129–136.

[69] Nabi A, Mirzadeh M, Gibou F, Moehlis J: **Minimum energy desynchronizing control for coupled neurons.** *Journal of computational neuroscience* 2013, **34**(2):259–71.

[70] Nabi A, Moehlis J: **Single input optimal control for globally coupled neuron networks**. *Journal of neural engineering* 2011, **8**(6):065008.

[71] Feng XJ, Greenwald B, Rabitz H, Shea-Brown E, Kosut R: **Toward closed-loop optimization of deep brain stimulation for Parkinson's disease:**

concepts and lessons from a computational model. *Journal of neural engineering* 2007, **4**(2):L14–21.

[72] Ahmadian Y, Packer AM, Yuste R, Paninski L: **Designing optimal stimuli to control neuronal spike timing.** *Journal of neurophysiology* 2011, **106**(2):1038–53.

[73] Gerstner W, Kistler WM: *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press 2002.

[74] Moehlis J, Shea-Brown E, Rabitz H: **Optimal Inputs for Phase Models of Spiking Neurons**. *Journal of Computational and Nonlinear Dynamics* 2006, **1**(4):358–367.

[75] Dasanayake I, Li JS: **Constrained minimum-power control of spiking neuron oscillators**. In *IEEE Conference on Decision and Control and European Control Conference*, IEEE 2011:3694–3699.

[76] Nabi A, Stigen T, Moehlis J, Netoff T: **Minimum energy control for in vitro neurons.** *Journal of neural engineering* 2013, **10**(3):036005.

[77] Feng J, Tuckwell H: **Optimal Control of Neuronal Activity**. *Physical Review Letters* 2003, **91**:1–4.

[78] Whittle P: *Optimal Control: Basics and Beyond*. Wiley 1996.

[79] Morris C, Lecar H: **Voltage oscillations in the barnacle giant muscle fiber**. *Biophys. J.* 1981, **35**:193–213.

[80] Ditlevsen S, Greenwood P: **The Morris-Lecar neuron model embeds a leaky integrate-and-fire model**. *J. Math. Biol.* 2013, **67**(2):239–259.

[81] Rinzel J, Ermentrout GB: *Methods in Neural Modeling*, The MIT Press, Cambridge 1989 chap. Analysis of neural excitability and oscillations.

[82] Tateno T, Pakdaman K: **Random dynamics of the Morris-Lecar neural model**. *Chaos* 2004, **14**:511–530.

[83] Ditlevsen S, Samson A: **Estimation in the partially observed stochastic Morris-Lecar neuronal model with particle filter and stochastic approximation methods**. *Annals of Applied Statistics* 2014. [To appear].

[84] Danzl P, Nabi A, Moehlis J: **Charge-balanced spike timing control for phase models of spiking neurons**. *Discrete and Continuous Dynamical Systems* 2010, **28**(4):1413–1435.

[85] Cavagnaro DR, Myung JI, Pitt MA, Kujala JV: **Adaptive design optimization: a mutual information-based approach to model discrimination in cognitive science.** *Neural computation* 2010, **22**(4):887–905.

[86] Myung JI, Cavagnaro DR, Pitt Ma: **A Tutorial on Adaptive Design Optimization.** *Journal of mathematical psychology* 2013, **57**(3-4):53–67.

[87] Paninski L, Pillow J, Lewi J: **Statistical Models for Neural Encoding, Decoding, and Optimal Stimulus Design**. *Progress in Brain Research* 2007, **165**:493–507.

[88] Paninski L: **Asymptotic Theory of Information-Theoretic Experimental**. *Neural Computation* 2005, **17**:1480–1507.

[89] Lewi J, Butera R, Paninski L: **Sequential optimal design of neurophysiology experiments.** *Neural computation* 2009, **21**(3):619–87.

[90] Granade CE, Ferrie C, Wiebe N, Cory DG: **Robust online Hamiltonian learning**. *New Journal of Physics* 2012, **14**(10):103013.

[91] Hooker G, Lin KK, Rogers B: **Control Theory and Experimental Design in Diffusion Processes**. *SIAM Journal of Uncertainty Quantification* 2015, **3**:234–264.

[92] Liu J, West M: **Combined parameter and state estimation in simulation-based filtering**. *Sequential Monte Carlo methods in practice* 2001.

[93] Ullah G, Schiff SJ: **Tracking and control of neuronal Hodgkin-Huxley dynamics**. *Physical review. E, Statistical, nonlinear, and soft matter physics* 2009, **79**(4 Pt 1):040901.

[94] Tuckwell HC: *Introduction to Theoretical Neurobiology: Volume 2, Nonlinear and Stochastic Theories* 2005.

[95] Nocedal J, Wright SJ: *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering, New York: Springer-Verlag 1999.