3-30-2015

# Flipping Biological Switches: Solving for Optimal Control: A Dissertation

Joshua TsuKang Chang
*University of Massachusetts Medical School Worcester*

# FLIPPING BIOLOGICAL SWITCHES: SOLVING FOR OPTIMAL CONTROL

A Dissertation Presented

By

JOSHUA TSUKANG CHANG

Submitted to the Faculty of the

University of Massachusetts Graduate School of Biomedical Sciences, Worcester

In partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

March 30th, 2015

M.D./Ph.D. PROGRAM

CLINICAL AND POPULATION HEALTH RESEARCH

FLIPPING BIOLOGICAL SWITCHES: SOLVING FOR OPTIMAL CONTROL

A Dissertation Presented
By

Joshua TsuKang Chang

The signatures of the Dissertation Defense Committee signify
completion and approval as to style and content of the Dissertation

_____
David Paydarfar, M.D., Thesis Advisor

_____
Premananda Indic, Ph.D., Member of Committee

_____
Zhiping Weng, Ph.D., Member of Committee

_____
Eli Shlizerman, Ph.D., Member of Committee

The signature of the Chair of the Committee signifies that the written dissertation meets
the requirements of the Dissertation Committee

_____
Peter Grigg, Ph.D., Chair of Committee

The signature of the Dean of the Graduate School of Biomedical Sciences signifies
that the student has met all graduation requirements of the school.

_____
Anthony Carruthers, Ph.D.,
Dean of the Graduate School of Biomedical Sciences

Clinical and Population Health Research

March 30th, 2015

*"To Him, who is able to do immeasurably more than all we ask or imagine, according to His power that is at work within us, to Him be glory in the church and in Christ Jesus throughout all generations, for ever and ever! Amen."*

*Ephesians 3:20-21*

# Acknowledgements

I praise and thank **God**, for providing me with the opportunity as well as the stamina and strength to undertake this journey. As the apostle Paul quotes, "For in Him we live and move and have our being" (Acts 17:28).

Indeed, this experience has been a reminder that He is able to do immeasurably more than I can ask for or even imagine. However, not only has He brought me to the end of my thesis, He has blessed me every step of the way with the best people possible. I thank Him for:

- My professor, advisor, mentor, and friend, **Dr. David Paydarfar**. I am truly amazed at how providential it had been to come across your lab. I do not think I had anticipated ever being able to do research at UMass that not only allows me to grow and develop in the medical profession, but also allows me to leverage my engineering background. Thank you for being such a great adviser, keeping me balanced between my research and medicine, between getting lost in the fun of solving derivatives and programming MATLAB scripts and remembering the importance of looking at the big picture. It has been a true joy and honor to have worked with you and learn much from you. I hope that we will have many more conversations and discussions ahead even after I reenter the medical school.
- My thesis committee, **Dr. Peter Grigg** as chair, **Dr. Premananda Indic** and **Dr. Zhiping Weng**. Thank you very much for taking a chance on this graduate student who wanted to ambitiously integrate artificial intelligence, signal processing, and

medicine all together. I could not have asked for a better thesis committee who has supported me every step of the way. Your insights along the way have provided me with much needed guidance.

- **Dr. Eli Shlizerman**, who has joined my thesis committee as the external member of my dissertation examination committee. Thank you for being willing to participate in this endeavor even though you are on the other side of the country.
- For the Paydarfar Lab, **Dr. David Paydarfar**, **Dr. Elizabeth Salisbury**, **Dr. Premananda Indic**, **Courtney Dunn Temple**, **Ian Zuzarte**, and **Alan Gee**. Thank you for welcoming me to the lab and bringing joy to my family and me. This lab knows how to celebrate events, and I thank you so much for allowing me to be a part of it.
- For **Dr. John Clay**. Thank you for allowing me the opportunity to work, and stay, with you at Woods Hole Oceanographic Institute. Our discussions on the ionic basis for excitability in squid axons were very informative.
- For Dr. Daniel Forger and Kirill Serkh. Thank you for the discussions on boundary value problems and your introduction to the gradient algorithm.
- For **Dr. William Schwartz**. Thank you for your advice on navigating the MD/PhD program, on how to construct language to be more understandable, and on how to celebrate birthdays and milestones properly.
- My church community and my small group in particular who have provided moral and prayer support through all the ups and downs. While there are too many to name,

allow me to thank the men in my small group (**Andrew Tsang**, **Daniel Dai**, **David Chiu**, **William Leung**, and **Eric Mui**) for your regular intercession before our Father.

- For friends who have taken the time to read my horrible writing, and provided insights and guidance on how to clarify and crystallize my thoughts onto paper.
- My wife, **Jessica Chang**. Words are unfortunately insufficient to express adequately my appreciation for all that you have done. Thank you for being willing to marry somebody who is a "life-long student." Someday, Lord willing, I will be done with school. Thank you for brightening each of my days with your grace and love. May God grant us many more years together.

*Lord God, I thank you for each of these individuals that you have so carefully placed into my life. Would You bless them as richly as You have blessed me through them.*

*Soli Deo Gloria.*

# Abstract

Switches play an important regulatory role at all levels of biology, from molecular switches triggering signaling cascades to cellular switches regulating cell maturation and apoptosis. Medical therapies are often designed to toggle a system from one state to another, achieving a specified health outcome. For instance, small doses of subpathologic viruses activate the immune system's production of antibodies. Electrical stimulation revert cardiac arrhythmias back to normal sinus rhythm. In all of these examples, a major challenge is finding the optimal stimulus waveform necessary to cause the switch to flip. This thesis develops, validates, and applies a novel model-independent stochastic algorithm, the Extrema Distortion Algorithm (EDA), towards finding the optimal stimulus. We validate the EDA's performance for the Hodgkin-Huxley model (an empirically validated ionic model of neuronal excitability), the FitzHugh-Nagumo model (an abstract model applied to a wide range of biological systems that that exhibit an oscillatory state and a quiescent state), and the genetic toggle switch (a model of bistable gene expression). We show that the EDA is able to not only find the optimal solution, but also in some cases excel beyond the traditional analytic approaches. Finally, we have computed novel optimal stimulus waveforms for aborting epileptic seizures using the EDA in cellular and network models of epilepsy. This work represents a first step in developing a new class of adaptive algorithms and devices that flip biological switches, revealing basic mechanistic insights and therapeutic applications for a broad range of disorders.

# Contents

Cover Page ..... i
Acknowledgements ..... iv
Abstract ..... vii
Contents ..... viii
List of Figures ..... xi
CHAPTER 1: Introduction ..... 15
Background ..... 16
Contribution ..... 17
CHAPTER 2: Gradient Algorithm: Mathematical Foundations ..... 22
CHAPTER 3: Gradient Algorithm: Revealing Optimal Stimuli for Switching Neuronal State ..... 30
Introduction ..... 30
Methods ..... 32
First-order gradient algorithm ..... 32
Hodgkin-Huxley model ..... 37
Implementing the gradient algorithm for the Hodgkin-Huxley model ..... 40
FitzHugh-Nagumo (FHN) model ..... 43
Implementing the algorithm for the FHN model from quiescence to repetitive firing ..... 45
Implementing the algorithm for the FHN model from repetitive firing to quiescence ..... 47
Results ..... 48
Hodgkin-Huxley model of neuronal excitation ..... 48
The FitzHugh-Nagumo model: quiescence to repetitive firing ..... 55
The FitzHugh-Nagumo model: repetitive firing to quiescence ..... 61
Discussion ..... 67
CHAPTER 4: Stochastic Search Algorithms: Review of Applications in Medicine ..... 73
Introduction ..... 73
Review of methods ..... 74
Spike-triggered averaging ..... 76
Stochastic hill-climbing ..... 78
Simulated annealing ..... 82

Genetic algorithms ..... 84
Discussion ..... 89
CHAPTER 5: Spike-Triggered Averaging: Limitations as an Optimization Tool ..... 91
Introduction ..... 91
Methods ..... 92
Hodgkin-Huxley model ..... 92
Determining optimality using a stochastically-seeded gradient algorithm ..... 92
Finding the spike triggered average using different noise levels ..... 93
Results ..... 94
Gradient-based calculation of stimulus optimality for eliciting an action potential ..... 94
Comparing STA waveforms to optimal waveforms ..... 94
Discussion ..... 102
CHAPTER 6: Extrema Distortion Algorithm: Harnessing Noise to Optimally Flip a Switch ..... 106
Introduction ..... 106
Methods ..... 107
Formalization of the problem ..... 107
Stochastic Extrema Distortion Algorithm ..... 109
Induction of an action potential: The Hodgkin Huxley model ..... 112
Switching off oscillations: The Fitzhugh-Nagumo model ..... 113
Flipping a genetic toggle switch ..... 115
Results ..... 118
Induction of an action potential: The Hodgkin Huxley model ..... 118
Switching off oscillations: The Fitzhugh-Nagumo model ..... 126
Flipping a genetic toggle switch ..... 130
Evaluating the Use of Pre-Processing Noise Filter ..... 134
Discussion ..... 138
CHAPTER 7: Extrema Distortion Algorithm: Optimally Suppressing Epileptic Seizures ..... 145
Introduction ..... 145
Methods ..... 147
Extrema Distortion Algorithm ..... 147

Single cell repetitive firing....148
Single cell bursting....149
Coupled oscillators network....151
Population-based systemic bursting....152
Results....155
Suppression of single cell repetitive firing....155
Suppression of single cell bursting....155
Desynchronization of coupled oscillators....157
Suppression of systemic bursting....158
Discussion....161
Electrical stimulation for epileptic treatment....161
Rectangular pulses are not optimal....162
Optimal stimulus waveforms are short discrete bursts in antiphase to the system....162
Understanding the single cell burster....165
CHAPTER 8: Conclusions and Future Directions....167
Application to Real-World Clinical Problems....167
Modeling Complex and Sensitive Systems....170
Optimality as a Design Principle in Nature....172
References....173

# List of Figures

Figure 1.1: Example of a switch being stimulated by a variety of stimulus waveform shapes, some successful while others are not. ..... 18

Figure 3.1: Plot of the limit cycle in the FitzHugh-Nagumo model. The phase values are normalized by time so that 0 and 1 are both at the "peak of the action potential" which occurs when $x_1$ is at its minimal value. Four representative phase values are shown to show the time progression around the limit cycle ..... 46

Figure 3.2: Gradient algorithm shapes random stimuli towards an optimal waveform. Here we show the progression of solutions as the gradient algorithm begins with a white-noise stimulus and finds the most energetically efficient solution over 100 iterations. The top panel shows the $L^2$-norm trajectory over 100 iterations. The six panels show the evolving waveform at the 1st, 5th, 10th, 20th, and 30th iteration of the algorithm. There is very little improvement between the 30th and the 100th iteration as seen in the $L^2$-norm trajectory (top panel). By convention, positive current is depolarizing ..... 49

Figure 3.3: The optimal stimulus derived from the gradient algorithm (top) triggers a single action potential in the Hodgkin-Huxley model (bottom) ..... 51

Figure 3.4: Longer stimulus duration provides for more energetically efficient stimulus. Once the stimulus duration extends past a certain point, there is no further improvement in energy efficiency ..... 53

Figure 3.5: The optimal waveforms change from a monophasic stimulus to a biphasic stimulus as the amount of time to action potential increases ..... 54

Figure 3.6: Gradient algorithm is robust to the initial stimulus. The stimulus energy trajectories of 100 different seeds with different amplitudes are shown ..... 56

Figure 3.7: Only certain phase regions converged when transitioning from quiescence to repetitive firing. The points that converged are marked in black, while those that failed are marked in red. ..... 57

Figure 3.8: Gradient algorithm finds multiple optimal solutions that induce a transition from quiescence to repetitive firing. The initial condition (quiescent fixed point) and the terminal condition ($\phi = 0$) of the gradient algorithm are the same for both trials. The only difference is the initial randomly generated stimulus that is given to the gradient algorithm. The top panel shows the two optimal stimuli (green and blue), while the bottom panel shows the response from the $x_1$ variable in the FitzHugh-Nagumo model to the stimuli (matched green and blue). ..... 60

Figure 3.9: Trajectories of stimulus energy through 200 iterations of gradient algorithm show convergence to two waveforms. We ran 100 different randomly generated seeds with different scaling factors. The green and blue colors match the respective green and blue waveforms seen in Figure 8 ..... 62

Figure 3.10: Gradient algorithm reveals multiplicity in transitioning from repetitive firing to quiescence across different phases. The top panel shows the stimulus energy of the different optimized stimuli (8 ms duration) that induce transitions from different phases of repetitive firing to the quiescent fixed point. Specific examples of different solutions are shown in the bottom pane, labeled *a* through *d*. Note that *b* and *c* are solutions with the same starting phases. ..... 63

Figure 3.11: Random stimuli converge towards two different optimal stimuli that induce a transition from repetitive firing to quiescence. The trajectories of stimulus energy through 100 iterations of gradient algorithm are shown here. All of these stimuli start from the same starting phase ($\phi = 0.678$). ..... 65

Figure 3.12: Gradient algorithm reveals different mechanisms of suppressing repetitive firing. The top panel shows the stimuli, the middle panel the $x_1$ response to the stimuli. The bottom panel shows the entire state space response to both stimuli. As we can see, one stimulus (green) suppresses the system quickly, while the other stimulus (blue dashed) causes the system to run more quickly around the oscillatory state before entering into the quiescent state. The black line in the figure is marked to show the first course of the system without any stimulation. This marks the limit cycle of the repetitive firing state. ....66
Figure 4.1: Fundamental concept of stochastic search algorithms ....75
Figure 4.2: Illustration of stochastic hill-climbing ....80
Figure 4.3: Flaw in stochastic hill-climbing algorithm ....81
Figure 4.4: Encoding signals as chromosomes. In this example, the amplitude of the signal at each time step becomes the value of genes. Once the method of encoding has been decided, the genetic algorithm is initiated with a generation, or set, of stochastic seeds. ....85
Figure 4.5: Each iteration of the genetic algorithm goes through a set of recombination, mutation, and selection in order to form the next generation. Here we use the generation initialized from Figure 4.4 as an example of how recombination, mutation, and selection would work to generate the next set of solutions. ....87
Figure 5.1: Energy-optimal stimulus waveform for the near-bistable condition displays more resonance than the optimal waveform for the standard condition for the Hodgkin-Huxley model ....95
Figure 5.2: Comparison of spike triggered averaging results (orange) and the gradient algorithm (blue). We note two main discordances: a phase reversal (*) and a right-shift (#). ....97
Figure 5.3: Optimal waveform shapes given different stimulus durations as determined by the gradient algorithm. ....98
Figure 5.4: The near-bistable solution has at least two locally optimal solutions: one that fires a single action potential (red), another which fires two action potentials in rapid succession (blue). ....100
Figure 5.5: When averaging stimuli causing only one spike, the STA (red) phase reversal disappears, and STA aligns more closely with the true optimal (blue). ....101
Figure 5.6: Whether the spike-triggered average is close to the energy-optimal is related to the shape of the intersection (Region C) between the set of all possible stochastic signal (Region B) and the set of all possible stimuli that successfully trigger an event (Region A). Sometimes when the center of mass (y) is scaled (y'), it may move closer to the optimal stimulus (x) as seen in the top panel, while other times it may move further away as seen in the bottom panel. ....105
Figure 6.1: Diagram of a single iteration of the Extrema Distortion Algorithm. Each panel shows the system response (top), the stimulus (middle) and a cartoon representation of the algorithm at work. The extrema are determined, and the intervals as well as the amplitudes are then distorted using multiplicative and addictive noise respectively to generate a new signal. ....111
Figure 6.2: Diagram of the genetic toggle switch. Figure adapted from Gardner TS, Cantor CR, Collins JJ (2000) Construction of a genetic toggle switch in Escherichia coli. Nature 403: 339-342. ....116
Figure 6.3: The Extrema Distortion Algorithm shapes stochastically generated stimuli towards an optimal waveform. Here we show the progression of solutions from the original white noise stimulus in the 1st

iteration to the 2nd, 5th, 10th, 20th, 50th, 1000th and 5000th iteration. By convention, positive current is depolarizing.....119
Figure 6.4: The extrema in the signal evolves over the course of the algorithm. Both the timing (top) and the amplitude (bottom) of the extrema begin randomly scattered and over the course of a few iterations, it rapidly converges towards a few key locations. Over the course of a few hundred iterations, one can see the algorithm prune away at unnecessary extrema.....121
Figure 6.5: The algorithm takes randomly generated stimuli and evolves them towards a single global optimal solution for the Hodgkin-Huxley model. As seen from the time course of the stimulus energy (A), the algorithm begins with very energetic stochastically generated stimuli and rapidly finds more energetically efficient stimuli. Panel (B) shows all of the resulting stimuli overlaid on top of each other, aligned by where they cause an action potential to occur. The stimulus' baseline is at 0 μA / $cm^2$.....122
Figure 6.6: The Extrema Distortion Algorithm finds solutions that match very closely with that found using the gradient algorithm. The result from the best of the 35 snippets using distortion of the extrema (red) is shown in panel A compared to the best of the 35 snippets using distortion of all the points (blue). While the fundamental shape is very similar, the result using distortion of all the points retains more noise than the result using distortion of only the extrema. Panel (B) shows the range of stimulus energy in 35 runs of both the Extrema Distortion Algorithm as well as the gradient algorithm. The stimulus' baseline is at 0 μA / $cm^2$.....124
Figure 6.7: The Extrema Distortion Algorithm (red) finds solution more efficiently than an all-points distortion algorithm (blue). We show both the $L^2$-norm progress of 5,000 iterations across both algorithms for the same initial 35 snippets in panel A. Panel B shows the best solution from both algorithms. The stimulus' baseline is at 0 μA / $cm^2$.....125
Figure 6.8: The Extrema Distortion Algorithm converges towards the two locally optimal solutions in the FitzHugh-Nagumo model. The 49 snippets used on the FitzHugh-Nagumo model revealed two separate locally optimal waveforms. The top panels show the model response to the stimuli shown in the bottom panels. Note that the left stimulus is much smaller in scale than the right stimulus. The snippets are aligned such that the first valley after repetitive firing has ended occurs at 0 ms. The stimulus' baseline is at zero. .....127
Figure 6.9: The Extrema Distortion Algorithm allows for more flexibility in terminal conditions leading to more optimal results. Panel A shows the result from the gradient algorithm (blue) as well as two versions of the Extrema Distortion Algorithm. The first version of the Extrema Distortion Algorithm (red) limits the terminal condition of success to be extremely close to the fixed point similar to the gradient algorithm. The second version of the Extrema Distortion Algorithm (green) specifies the condition of success to just be a suppression of action potentials ($-x_1 < 1.5$). Panel B shows the state space of the system's response to the various stimuli (left) with a zoomed in version to what is occurring next the fixed point (right). The stimulus's baseline is at 0. .....129
Figure 6.10: The Extrema Distortion Algorithm can be used to gain insight into the phase-dependency and duration of the optimal stimulus. The Extrema Distortion Algorithm used on 17 unique 100-ms snippets shows converge towards a phase-specific stimulus waveform. These snippets were all aligned such that the first peak after the action potential occurred at 100 ms. The top panel shows the

FitzHugh-Nagumo's response to the stimulus displayed in the bottom panel. The stimulus's baseline is at 0. ..... 131
Figure 6.11: The Extrema Distortion Algorithm can be used to find optimality under constrained conditions. In panel A, success is defined by seeing a transition occur at 20 minutes. The stimulus in terms of concentration of [IPTG] is shown above the concentration of repressor 2. As seen, the Extrema Distortion Algorithm (red) produces a very similar waveform to that of the gradient algorithm (blue). In panel B, success defines the transition occurring at 100 minutes, while the stimulus duration is still kept at 20 minutes. The gradient algorithm calculates the optimal stimulus using knowledge of where the separatrix, while the Extrema Distortion Algorithm does not. The stimulus starts at 0 M [IPTG]. ..... 132
Figure 6.12: The Extrema Distortion Algorithm can discover optimal stimuli based off of different performance metrics. Here we show the resulting optimal stimuli discovered by the Extrema Distortion Algorithm using the $L^2$-norm performance metric (blue) and the maximum amplitude performance metric (red). The stimulus starts at 0 M [IPTG]. ..... 135
Figure 6.13: Comparison of the ranges of $L^2$-norm results between the gradient algorithm, the Extrema Distortion Algorithm, and the Empirical Mode Decomposition combined with the Extrema Distortion Algorithm. ..... 137
Figure 6.14: Comparison of the best result from Extrema Distortion Algorithm (red) with the best result from the Extrema Distortion Algorithm combined with an initial Empirical Mode Decomposition denoising. The stimulus is aligned such that the action potential occurs at 0ms. ..... 139
Figure 7.1: Diagram of the Suffczynski et al model. Four populations of neurons are modeled: pyramidal neurons (PY), interneurons (IN), reticulothalamic neurons (RE), and thalamocortical neurons (TC). 154
Figure 7.2: Optimal stimulus waveform for suppressing repetitive firing in a single cell. ..... 156
Figure 7.3: Optimal stimulus waveform of the suppression of a single cell burster. ..... 157
Figure 7.4: Optimal stimulus waveform for desynchronizing a network of coupled oscillators ..... 159
Figure 7.5: Optimal stimulus waveform for suppressing seizure activity in a systemic population-based model. We show the stimulus, the system's response, and the control response with no stimulus. The bottom set shows a magnification of the transition due to the stimulus. ..... 160

# CHAPTER I
# Introduction

Switches play an important regulatory role at all levels of biology. At a molecular level, we see examples of switches that can activate or deactivate specific biochemical pathways[1,2] or trigger signaling cascades[3]. At a cellular level, we know about switches that play key roles as gatekeepers regulating cell maturation[4–7] and cell death[8]. At an intercellular level, postsynaptic potentials from one neuron can lead to opening or closing of voltage-dependent gates in other neurons allowing or inhibiting propagation of action potentials[9–11]. At an organism level, neural clusters in the hypothalamus act as a toggle switch that governs whether the body is in a sleep or wake state[12,13]. In all of these examples, biological switches flip between distinct states.

Medical therapies often seek to trigger, or stimulate, biological switches that toggle a system from one state to another, achieving a specified health outcome. For example, clinicians use small doses of subpathologic viruses to activate the immune system's production of antibodies for the purpose of vaccination[14]. Electricity has been used to induce a switch in clinical state, as seen in cardiac defibrillation where short electrical pulses are delivered to revert arrhythmias back to normal sinus rhythm[15–19]. More

recently, electrical stimuli are applied to deep brain structures in order to combat Parkinsonian tremors[20–23], epileptic seizures[24–26], and even major depressive disorders[27–30]. We have studied cases where mechanical stimulation has been used to stimulate normal breathing patterns in neonates suffering from life-threatening apneas[31].

In all these instances, one of the major challenges has been finding optimal stimulus waveforms, the ideal shape of the stimulus intensity over time. Figure 1.1 shows examples of different stimulus waveforms, some successfully causing a change in state. While different waveforms may be able to trigger a switch in the system, they may not all be optimal. It is important to note at this point that different applications may define optimality uniquely. For electrical stimulation, clinicians may want to reduce the maximum stimulus intensity at any point in time for fear of causing tissue damage or energy leakage into neighboring regions leading to detrimental side effects. Other clinicians may be concerned about reducing total energy consumption, in order to maximize battery life for implantable devices. In vaccines, one may seek to minimize the total injected dose to prevent toxicity. Whatever the criterion, the fundamental question remains, how does one go about finding the optimal stimulus waveform?

# Background

Traditional methods have involved systematically testing a large number of different waveforms and examining which stimuli performed optimally[15]. Unfortunately, as one can imagine, there are an infinite number of possible waveforms, and thus systematically

searching through all of them is impossible. Some researchers have narrowed the search field by assuming a specific waveform shape, and systematically exploring the parameters that make up the size of the shape[16]. For example, one may seek to find the optimal pulse width and amplitude of a rectangular stimulus. These constraints lead to a biased search, missing potentially more optimal solutions.

As an alternative to an exhaustive search, some researchers have sought to apply mathematical techniques from the field of calculus of variations[32] to solve for the optimal stimulus waveform[33–38]. In these instances, researchers use a mathematical model of the behavior in question, applying various formulas and techniques from calculus of variations to derive the optimal stimulus waveform. While this method does indeed result in the optimal solution, it is very time-intensive and difficult to use on complex systems. Furthermore, it requires a mathematical model describing the behavior accurately. With biological systems as complex as they are, often times it is very hard to model a specific behavior much less an entire system.

# Contribution

Thus, the challenge we have set for ourselves in this thesis is:

*To develop an algorithm that can find an optimal stimulus waveform without any knowledge of the mechanisms or models underlying the physiological system.*

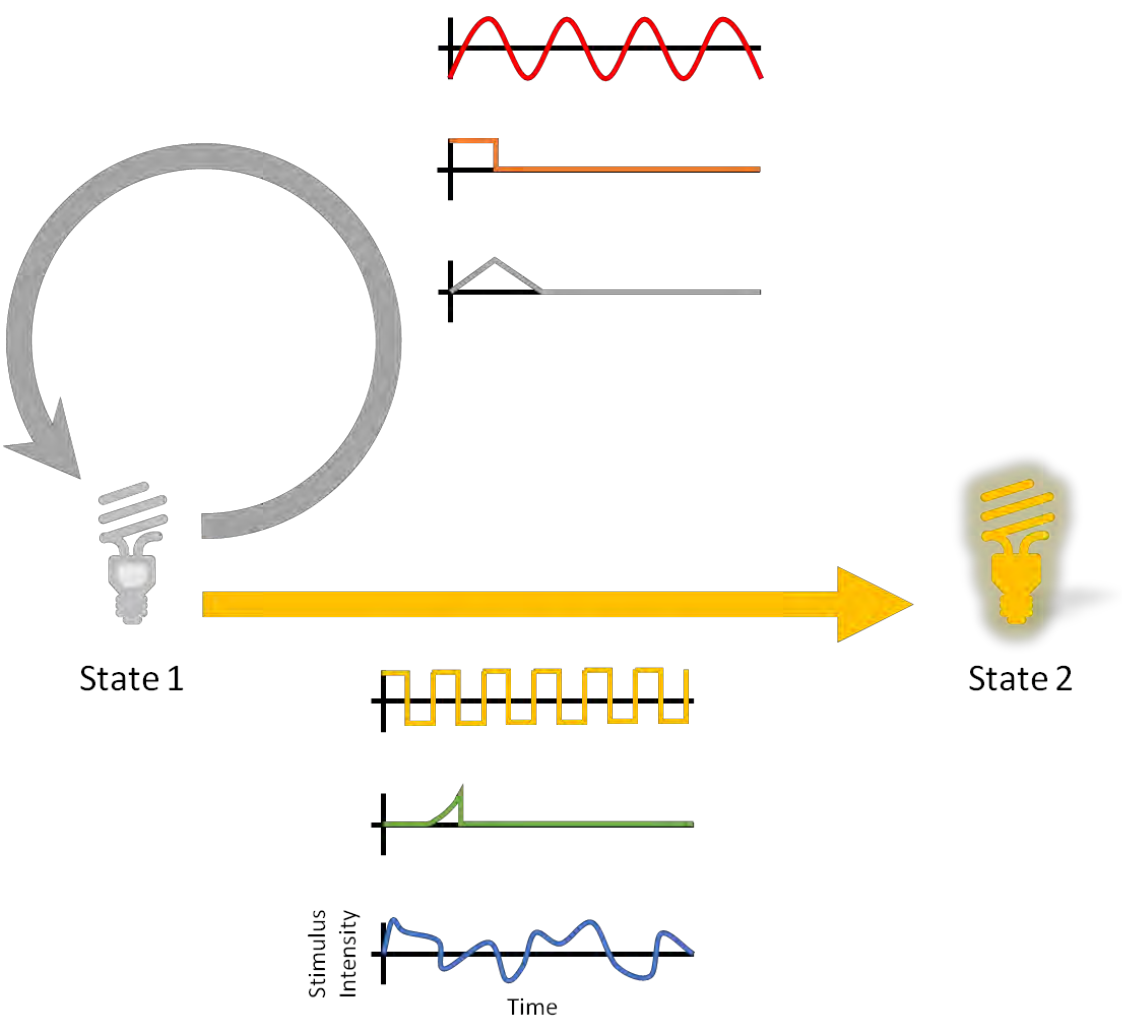


**Figure 1.1: Example of a switch being stimulated by a variety of stimulus waveform shapes, some successful while others are not.**

To answer this challenge, the thesis can be broken into three aims:

*Aim 1: Expansion of a gradient-based algorithm by incorporating stochastic seeding to develop standards by which to measure the validity of any new algorithm*

The validation of any novel algorithm requires the definition of an analytical standard. We defined our standard based on analytical approaches using calculus of variations. As we had stated earlier, applications of calculus of variations in biology have been limited due to the complexity and difficulty of convergence required for implementation of this technique. Furthermore, traditional variational approaches compute local optima whereas there is often a need to explore the solution space more broadly to find the global optimum. To address these limitations, we have devised a stochastically-seeded gradient algorithm; a series of random stimuli is the starting points for the algorithm. The algorithm derives how the system will respond to minute changes for each of these initial starting points. Using first-order gradients, or slopes, the algorithm proceeds to calculate a more optimal solution. This algorithm iterates this process until it converges to a single solution. The inclusion of stochastic seeding, or the randomly generated initial stimuli, enables the algorithm to explore a large solution space. We demonstrate and discuss the ease of using the gradient-based algorithm in searching for optimality.

Chapter 2 presents an overview of the mathematical foundations and derivations underlying calculus of variations and more specifically the gradient algorithm we used. We detail our incorporation of stochastic seeding and its application to finding optimal stimulus waveforms for two separate neuronal models in Chapter 3.

*Aim 2: Development and Validation (using the methods of Aim 1) of a novel model-independent stochastic algorithm for finding optimal stimulus waveforms, which we call the Extrema Distortion Algorithm.*

Stochastic algorithms are computational tools that engineers have applied to optimization problems. These iterative methods leverage random processes to sample the solution space broadly surrounding a set of initial solutions. After exploring the solution space, a pruning process chooses a few of the sampled solutions, which then become the starting points of the next iteration. The simplest stochastic algorithm that we have found used in experimental neuroscience has been what is known as spike-triggered averaging[33,39]. While the literature has demonstrated that the spike-triggered average produces results that qualitatively look very similar to what analytical techniques have produced, we explore some quantitative discrepancies further in Chapter 4.

Because of the discrepancies that lie with spike-triggered averaging, we developed a novel Extrema Distortion Algorithm to find optimal stimulus waveforms to flip a switch, which we call the Extrema Distortion Algorithm. Chapter 5 describes and validates the Extrema Distortion Algorithm using the results from our first aim.

*Aim 3: Application of the Extrema Distortion Algorithm to find optimal stimulus waveforms for suppressing epileptic seizures*

While simple neuronal models are interesting, their clinical applications are limited. Thus, we wanted to apply the Extrema Distortion Algorithm to a more complicated

problem, one that is too complex for analytical techniques to solve, in order to glean therapeutic and clinical insights by finding the optimal stimulus waveform. In Chapter 6, we apply our algorithm to the problem of suppressing epileptic seizures using a range of computational models based on different mechanisms hypothesized in the literature.

Chapter 7 closes the thesis with a discussion of a few interesting directions for the future of this research.

# CHAPTER II

# Gradient Algorithm: Mathematical Foundations

Before we apply the gradient-based algorithm as developed by Bryson and Ho [40] to our optimization problem, it would be beneficial to first understand the mathematical construct and foundations by which various analytical techniques are used. Many of the algorithms used for finding optimal waveforms rely on a branch of mathematics called calculus of variations.

Most people are familiar with calculus as a field of mathematics used to find the minimum or maximum points for a given function. What calculus of variations does is find the function with a minimum or maximum value based on a function of functions, or, in mathematics, a functional. In the context of finding optimal waveforms, calculus of variations helps us find the waveform, or function, that uses the least amount of energy, which is a function of the waveform, or functional.

---

Portions of this chapter have been reprinted with permission from: Chang J, Paydarfar D. Switching neuronal state: optimal stimuli revealed using a stochastically-seeded gradient algorithm. J Comput Neurosci. 2014 37(3):569-82.

In order to progress into the mathematical foundations, let us begin by first defining the system in question as a set of nonlinear differential equations:

$$\dot{x} = f[x(t), u(t), t],\ x(0) \text{ is given},\ t_0 \leq t \leq t_f \tag{2-1}$$

where $x(t)$ is an *n*-dimensional vector function describing the *n* states in the nonlinear system, and $u(t)$ is an *m*-dimensional vector function describing the external stimulus? As an example, in the Hodgkin-Huxley equations, $x(t)$ is a 4-dimensional vector function, $[V(t), m(t), n(t), h(t)]$ and $u(t)$ is a 1-dimensional vector function describing the current intensity at any given time.

In order to find an optimum, we need to define a metric to measure "optimality." We call this a performance index. In most mathematical abstractions, the performance index follows the form:

$$J = \varphi\left[x(t_f), t_f\right] + \int_{t_0}^{t_f} L[x(t), u(t), t]dt \tag{2-2}$$

where $\varphi\left[x(t_f), t_f\right]$ is a *q*-dimensional vector describing the *q* terminal conditions, and $L[x(t), u(t), t]$ is an equation that uses both the states and the stimulus across all time to measure some level of performance. The convention in most of the literature is to minimize $J$. If one desires to maximize a performance metric, the negative of the metric is used, and thus the problem becomes one of minimization. Using equations (2-1) and (2-2), the optimization problem is now mathematically formalized as finding the function $u(t)$ that minimizes the functional $J$ given the system equations as defined by $\dot{x}(t)$.

At this point, we have system constraints as a unique entity from the performance index. In calculus of variations, we can combine these distinct entities into a single functional using Lagrange multipliers. Lagrange multiplier, also termed influence functions in some literature, is a technique that allows us to combine, or adjoin, multiple functions or constraints into one function. In order to combine the system constraints, (2-1), and the performance index, (2-2), into one unified equation, we adjoin the two functions using a set of Lagrange multipliers that we define here as $\lambda(t)$:

$$\bar{J} = \varphi\left[x(t_f), t_f\right] + \int_{t_0}^{t_f} [L[x(t), u(t), t] + \lambda^T(t)\{f[x(t), u(t), t] - \dot{x}\}]dt \qquad (2\text{-}3)$$

One of the foundational terms in calculus of variations is the Hamiltonian which is a scalar function, $H$, used to simplify such equations:

$$H[x(t), u(t), \lambda(t), t] = L[x(t), u(t), t] + \lambda^T(t) f[x(t), u(t), t] \qquad (2\text{-}4)$$

In mechanics, the Hamilton refers to the total energy in the system. We use it now to simplify the mathematics a little. Substituting (2-4) into (2-3), as well as integrating by parts the last component of the new performance index, we can rewrite $\bar{J}$:

$$\begin{aligned} \bar{J} = {} & \varphi\left[x(t_f), t_f\right] \\ & + \int_{t_0}^{t_f} [H[x(t), u(t), t] + \lambda^T(t)x]dt - \lambda^T(t_f)x(t_f) \\ & + \lambda^T(t_0)x(t_0) \end{aligned} \qquad (2\text{-}5)$$

In order to optimize this modified performance index, we want to find the variation in $\bar{J}$ due to variations in $x(t)$ and $u(t)$ at $t = t_f$ and $t = t_0$, the temporal boundary conditions of the problem:

$$\delta\bar{J} = \left[\left(\frac{\partial\varphi}{\partial x} - \lambda^T\right)\delta x\right]_{t=t_f} + [\lambda^T \delta x]_{t=t_0} + \int_{t_0}^{t_f}\left[\left(\frac{\partial H}{\partial x} + \dot{\lambda}\right)\dot{x} + \frac{\partial H}{\partial u}\delta u\right]dt \qquad (2\text{-}6)$$

In order to simplify the equation, we remove the $\delta x$ variable by defining the Lagrange multiplier, $\lambda(t)$, as follows:

$$\dot{\lambda} = -\frac{\partial H}{\partial x} = -\frac{\partial L}{\partial x} - \lambda^T\frac{\partial f}{\partial x}, \text{ with boundary conditions } \lambda^T(t_f) = \frac{\partial\varphi}{\partial x(t_f)} \qquad (2\text{-}7)$$

This choice of the Lagrange multiplier in (2-26) simplifies (2-6):

$$\delta J = \lambda^T(t_0)\delta x(t_0) + \int_{t_0}^{t_f}\frac{\partial H}{\partial u}\delta u dt \qquad (2\text{-}8)$$

For an optimum solution, $\delta J = 0$ for any arbitrarily defined $\delta u(t)$. Examining (2-8), we see that this happens only if

$$\frac{\partial H}{\partial u} = 0, t_0 \le t \le t_f \qquad (2\text{-}9)$$

The two equations, (2-7) and (2-26), constitute the Euler-Lagrange equations in calculus of variations.

With the Euler-Lagrange equations, we can now translate an optimization problem into a set of system equations that define the optimal stimulus. However, this is a two-point boundary value problem, which can be difficult to solve where we can define some of the variables.

Up to this point, we have allowed $\delta u(t)$ to be arbitrary. However, we can restrict the stimulus further by including the terminal condition constraints:

$$x_i(t_f) = x_{i,f} \tag{2-10}$$

Because we now have constraints for both the initial conditions and the terminal conditions, we know

$$\delta x_i(t_f) = 0\ ,\ \delta x_i(t_i) = 0,\ i = 1, \dots, q \tag{2-11}$$

We can rewrite (2-6) to include the constraints from (2-26). To differentiate the Lagrange multipliers, we rename them from $\lambda$ to $p$:

$$\delta J = \int_{t_0}^{t_f} \left[ \left( \frac{\partial H}{\partial x} + \dot{p}^T \right) \delta x + \frac{\partial H}{\partial x} \delta u \right] dt \tag{2-12}$$

Again, we choose the *n*-vector multipliers $p(t)$ such that we remove the $\delta x$ component:

$$\dot{p}^T = -\frac{\partial H}{\partial x} = -\frac{\partial L}{\partial x} - p^T \frac{\partial f}{\partial x} \tag{2-13}$$

The boundary conditions are such that if the terminal conditions are known, $p^T(t_f) = 0$. If they are not known, then the terminal conditions from the Euler-Lagrange equations still hold:

$$p^T(t_f) = \begin{cases} 0 & j = 1, \dots, q \\ \left.\dfrac{\partial \varphi}{\partial x_j}\right|_{t=t_f} & j = q+1, \dots, n \end{cases} \tag{2-14}$$

The multiplier function, $p(t)$, can be understood as the influence the stimulus, $\delta u$, has on the performance index, $J$. We also want to find the influence the stimulus has on the system arriving at the terminal conditions. To do this, we rewrite the performance index such that $\varphi = x_i(t)$ and $L = 0$. We then reproduce the process in equations (2-12) to (2-26) using this new performance index, and defining the influence functions in this particular iteration as, $R(t)$, an $(n \times q)$ matrix:

$$\delta x(t_f) = \int_{t_0}^{t_f} R^T \frac{\partial f}{\partial u} \delta u dt \tag{2-15}$$

with $R(t)$ defined as:

$$\dot{R} = -\left(\frac{\partial f}{\partial x}\right)^T R \text{ where } R(t_f) = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \tag{2-16}$$

Now, we can expand our performance index to include both the effects of $\delta u$ on $\delta J$ and $\delta x_i$. We introduce, $v$, as yet another Lagrange multiplier to adjoin these two variables:

$$\delta J + v\delta x = \int_{t_0}^{t_f} \left[\frac{\partial L}{\partial u} + [p + vR]^T \frac{\partial f}{\partial u}\right] \partial u dt \tag{2-17}$$

At this point, we want to find $\delta u$ such that the combined performance index and distance to terminal condition improves over time. Thus, we choose:

$$\delta u = -k\left\{\left(\frac{\partial f}{\partial u}\right)^T [p + vR] + \left(\frac{\partial L}{\partial u}\right)^T\right\} \tag{2-18}$$

where $k$ is a scaling factor between 0 and 1.

Plugging this back into (2-17) we see:

$$\delta J + v\delta x = -k\int_{t_0}^{t_f} \left\{\left(\frac{\partial f}{\partial u}\right)^T [p + vR] + \left(\frac{\partial L}{\partial u}\right)^T\right\}^2 dt \tag{2-19}$$

The fact that this equation is always negative means that this choice of $\delta u$ guarantees an improved $\delta J + v\delta x$, which should lead to a better performance.

In order to find out what the proper multiplier, $v_i$, should be, we plug (2-18) into equation (2-15):

$$\delta x(t_f) = -k\int_{t_0}^{t_f} R^T \frac{\partial f}{\partial u}\left\{\left(\frac{\partial f}{\partial u}\right)^T [p + vR] + \left(\frac{\partial L}{\partial u}\right)^T\right\} dt \tag{2-20}$$

After rearranging the terms, we see that:

$$\delta x(t_f) = -k \int_{t_0}^{t_f} R^T \frac{\partial f}{\partial u} \left[ \left( \frac{\partial f}{\partial u} \right)^T p + \left( \frac{\partial L}{\partial u} \right)^T \right] dt - vk \int_{t_0}^{t_f} R^T \frac{\partial f}{\partial u} \left( \frac{\partial f}{\partial u} \right)^T R dt \qquad (2\text{-}21)$$

We now define variables $Q$ and $g$ as follows:

$$Q = k \int_{t_0}^{t_f} R^T \frac{\partial f}{\partial u} \left( \frac{\partial f}{\partial u} \right)^T R dt \qquad (2\text{-}22)$$

$$g = k \int_{t_0}^{t_f} R^T \frac{\partial f}{\partial u} \left\{ \left( \frac{\partial f}{\partial u} \right)^T p + \left( \frac{\partial L}{\partial u} \right)^T \right\} dt \qquad (2\text{-}23)$$

which allows us to rewrite (2-21) as follows:

$$\delta x(t_f) = -g - vQ \qquad (2\text{-}24)$$

Solving for $v$:

$$v = -Q^{-1} \left[ \delta x(t_f) + g \right] \qquad (2\text{-}25)$$

Because we want to move $x(t_f)$ towards $x_{i,f}$, we will therefore choose

$$\delta x(t_f) = \varepsilon \left[ x(t_f) - x_{i,f} \right] \qquad (2\text{-}26)$$

where $\varepsilon$ is a scaling factor between 0 and 1.

# CHAPTER III

# Gradient Algorithm: Revealing Optimal Stimuli for Switching Neuronal State

## Introduction

There has been recent interest in establishing methods that utilize external electrical stimulation for controlling pathological neuronal activities in neurological disorders, for example, Parkinsonian tremor [21] and epileptic seizures [24]. Determining the minimal effective stimulus for such control is a key practical question because energetic optimization of the stimulus will decrease power usage and prolong battery life, as well as diminish off-target effects and damage to neighboring regions.

Finding optimal stimulus waveforms to control biological systems poses interesting computational challenges. Traditionally, optimization of signals has been conducted analytically using calculus of variations [32], in which an optimal functional is derived and solved as a boundary-value problem, using the shooting method [41] or the Newton-Raphson method [42]. These techniques, which have been applied to neuronal models

---

Portions of this chapter have been reprinted with permission from: Chang J, Paydarfar D. Switching neuronal state: optimal stimuli revealed using a stochastically-seeded gradient algorithm. J Comput Neurosci. 2014 37(3):569-82.

[33,39,43], requires an initial guess that seeds an algorithm used to solve the boundary-value problem. Finding an initial guess that converges to a solution can be difficult in mathematical models with steep nonlinearities of multiple state variables. Another important limitation for solving optimal functionals as a boundary value problem is the need for *a priori* knowledge of the optimal endpoint of all state variables. Optimization problems are often defined by a single outcome measure (e.g., achieving a voltage threshold for an action potential) and the optimal endpoint for all state variables may be unknown. Finding a global optimum therefore would require solving the boundary value problem multiple times for all possible endpoints that include the desired outcome measure [33].

Gradient-based optimization methods [40,44] offer an alternative computational approach for variational analysis that retains the complete model description and circumvents the need for solving a boundary-value problem. Gradient-based algorithms solve the optimization problem directly without first deriving a functional with defined boundary conditions. Recent success using this approach has been achieved for solving complex control problems in mechanics [45–47], epidemiology [48], game theory [49], kinetics [50] and immunology [51–53].

To our knowledge, gradient algorithms have not been applied to problems in computational neuroscience, and here we ask if such an approach might be useful for identifying minimal effective stimuli for controlling neuronal activity. As an initial analysis, we focus on waveforms that induce a single action potential in a classical

monostable neuron [10] or induce or suppress repetitive firing in a bistable neuron [54,55]. Furthermore, we develop and apply a stochastic seeding approach to the gradient algorithm in order to explore more fully the solution space to determine globally optimal solutions as opposed to just locally optimal solutions. Our results provide insight into how optimal control of neuronal activity is strongly influenced by temporal constraints of the stimulus waveform and the terminal conditions that define the outcome measure.

The chapter proceeds as follows. We first outline the gradient-based algorithm as well as its implementation for both the Hodgkin-Huxley and the Fitzhugh-Nagumo models. We highlight the importance of broad exploration of the solution space using stochastically generated seeds as we examine the results. Finally, we discuss the advantages and limitations of the gradient algorithm and compare it to other previously used techniques.

# Methods

## First-order gradient algorithm

The previous chapter detailed the derivation and theoretical framework of the first-order gradient algorithm. Here we provide a systematic outline of the implementation of the algorithm. To summarize, this algorithm begins with an initial estimate of the optimal stimulus and iteratively chooses a better stimulus based on the first-order gradient, or slope, of the system's response. The algorithm accomplishes this by calculating how the changes in the stimulus will affect the performance index as well as the error in terminal conditions. To calculate the error in terminal conditions, the algorithm runs the system's

response to the stimulus and compares the actual state at the terminal point with the desired terminal state.

As we will show, this algorithm is very robust to the initial estimate of the optimal stimulus, enabling us to use randomly generated stimuli to search across a larger solution space. We describe the gradient algorithm below, which is based on the formalisms developed by Bryson and Ho [40].

Given a nonlinear system of equations:

$$\dot{x} = f[x(t), u(t), t] \text{ for } t_0 \leq t \leq t_f \tag{3-1}$$

where $x(t)$ describes an *n*-dimensional system and $u(t)$ describes an *m*-dimensional external stimulus to the system, and the system's initial conditions $x_0$, we seek an optimal stimulus, $u(t)$, that minimizes the scalar performance index, $J$, such that:

$$J = \int_{t_0}^{t_f} L[x(t), u(t), t]dt \tag{3-2}$$

where $L[x(t), u(t), t]$ is a performance metric. The performance metric can be any function of both the system and the stimulus. In our examples, we will be using $L^2$-norm as the performance metric to calculate the energy of the stimulus. In examining exogenous stimulation, this metric is relevant to the power used by the stimulus. This means that our performance metric is:

$$L[x(t), u(t), t] = u^2 \tag{3-3}$$

We can replace this metric with another mathematical expression meaningful to any other optimization parameter. For instance, in endogenous stimulation, one may be more interested in ATP consumption as opposed to the $L^2$-norm, and may change the performance metric accordingly.

Finally, the algorithm is constrained by $q$ terminal conditions of the form:

$$x(t_f) = x_f \tag{3-4}$$

The algorithm proceeds as follows:

1. Estimate the stimulus variable, $u(t)$. We used a uniform random number generator to specify the initial stimulus values for each time step from $t_0$ to $t_f$.
2. Integrate the state variables $\dot{x}(t)$ forward with the given initial conditions and the stimulus generated in Step 1.
3. Determine the influence functions $p(t)$ and $R(t)$ by backward integration of

$$\dot{p}^T = -\frac{\partial L}{\partial x} - p^T \frac{\partial f}{\partial x} \tag{3-5}$$

$$\dot{R} = -\left(\frac{\partial f}{\partial x}\right)^T R \tag{3-6}$$

using the values from the state calculated from Step 2. Here, $p(t)$ is an $n \times 1$ dimensional matrix, while $R(t)$ is an $n \times q$ dimensional matrix. In these

calculations, $p(t)$ represents the strength of influence changes to the stimulus will have on the performance index, while $R(t)$ represents the strength of influence changes to the stimulus will have on the error in terminal conditions.

These two influence functions describe how changes in the stimulus will affect the performance index and the distance from the expected terminal conditions.

4. Simultaneously with Step 3, compute:

$$Q = k \int_{t_0}^{t_f} R^T \frac{\partial f}{\partial u} \left(\frac{\partial f}{\partial u}\right)^T R dt \tag{3-7}$$

$$g = k \int_{t_0}^{t_f} R^T \frac{\partial f}{\partial u} \left\{ \left(\frac{\partial f}{\partial u}\right)^T p + \left(\frac{\partial L}{\partial u}\right)^T \right\} dt \tag{3-8}$$

$Q$ and $g$ are intermediate variables used to simplify the equations in step 5. Here, $Q$ is a $q \times q$ dimensional matrix while $g$ is a $q \times 1$ dimensional matrix. The variable $k$ is a scaling factor that describes how large of a step size should be taken. We discuss how to choose this value at the end of the algorithm.

5. Using the results from the state variables in Step 2, calculate

$$v = -Q^{-1}\left[\delta x(t_f) + g\right] \tag{3-9}$$

$$\delta x(t_f) = \varepsilon\left[x(t_f) - x_f\right] \tag{3-10}$$

Here, $v$, an $n \times 1$ dimensional matrix, becomes a multiplier that balances the two different influence functions $p(t)$ and $R(t)$. As $v$ becomes larger, the algorithm

puts more weight in the effect of the influence factor $R(t)$ as compared to the influence factor $p(t)$.

The variable $\delta x$ is proportional to the distance between the actual terminal state due to the estimated stimulus and the desired terminal conditions that have been defined. The variable $\varepsilon$ is another scaling factor that describes how large the step size is. The larger the scaling factor, the larger the step will be in the direction of the gradient towards the optimal solution.

6. Repeat steps 1 to 5 with an improved estimate of the stimulus variable using

$$\delta u = -k\left\{\left(\frac{\partial f}{\partial u}\right)^T [p + vR] + \left(\frac{\partial L}{\partial u}\right)^T\right\} \quad (3\text{-}11)$$

This entire process continues for either a predefined number of iterations, or until the standard deviation of $\delta J$ and $\delta u$ over a set number of iterations has fallen below a specified threshold, indicating that the algorithm has converged to a solution.

It is interesting to note that this algorithm weighs and manages the weights of both the performance index as well as how closely the estimated stimulus fulfills the terminal conditions. As seen in Step 5, the further away the terminal states of the system due to the estimated stimulus is from the terminal conditions, the larger $v$ becomes. Thus more weight is put on the effect of the stimulus with regards to fulfillment of the terminal

conditions, $R(t)$ as opposed to the effect of the stimulus on improving the performance index, $p(t)$.

Some trial and error may be needed with regards to how to choose scaling factors for both $k$ and $\varepsilon$. The predicted decrease in the performance index, $\delta J$, can be compared to actual decrease in performance index. If the difference is large, the scaling factor can be decreased. If the difference is small between predicted versus actual, the scaling factor can be increased.

All of the simulation and algorithmic work was carried out in Matlab (The MathWorks Inc., Natick, MA, USA). Our code is publically accessible on the Internet at PhysioNet (http://physionet.org).

## Hodgkin-Huxley model

One of the classic computational models regarding excitable systems is the Hodgkin-Huxley neuron model [10]. The model is a four dimensional system that captures the ionic mechanisms underlying the generation of an action potential. The Hodgkin-Huxley model is defined as follows:

$$C\dot{V} = -120m^3h(V-115) - 36n^4(V+12) - 0.3(V-10.613) - u \tag{3-12}$$

$$\dot{m} = -m\big(\alpha_m(V) + \beta_m(V)\big) + \alpha_m(V) \tag{3-13}$$

$$\dot{n} = -n\big(\alpha_n(V) + \beta_n(V)\big) + \alpha_n(V) \tag{3-14}$$

$$\dot{h} = -h\big(\alpha_h(V) + \beta_h(V)\big) + \alpha_h(V) \tag{3-15}$$

$$\alpha_m(V) = \frac{0.1\phi(25-V)}{e^{0.1(25-V)}-1}, \beta_m(V) = 4\phi e^{-V/80} \quad (3\text{-}16)$$

$$\alpha_n(V) = \frac{0.01\phi(10-V)}{e^{0.1(10-V)}-1}, \beta_n(V) = 0.125\phi e^{-V/80} \quad (3\text{-}17)$$

$$\alpha_h(V) = 0.07\phi e^{-V/20}, \beta_m(V) = \frac{\phi}{e^{0.1(30-V)}+1} \quad (3\text{-}18)$$

where $V$ is the membrane voltage, $m$, $n$, and $h$ represent dimensionless quantities associated with sodium channel activation, potassium channel activation, and sodium channel inactivation respectively, and $u$ represents the exogenous stimulation we are looking to input into the system ($\mu A/cm^2$). The parameter $\phi$ is based on the ambient temperature, which we have set to 1.5. The parameter $C$ is the capacitance of the membrane, which we set at (1 $\mu F/cm^2$). The Hodgkin-Huxley model that we have defined here is monostable; the membrane is quiescent, firing an action potential only when it is elicited by the input stimulus.

By setting each of the differentials as well as the stimulus to 0, we are able to find the resting state of the Hodgkin-Huxley model. We found this resting state to be $V$ = 0.0026 mV, $m$ = 0.0529, $n$ = 0.3177 and $h$ = 0.596.

Using the gradient algorithm, we can determine what the optimal stimulus should be in order to cause an action potential from the resting state using the least amount of energy as determined by the performance index mentioned earlier, $L^2$-norm, which we measured in $\mu J/cm^2$. The standard model measures current per unit area of membrane (1 $cm^2$).

We generated our initial estimate by choosing random values at intervals of 0.1 milliseconds. Each random value was chosen from a uniform distribution from -1 $\mu A/cm^2$ to 1 $\mu A/cm^2$. These initial estimates were multiplied by a scaling factor to allow for different stimulus strengths. To increase the chance that we find the global optimal, we ran the algorithm 10 times with a different randomly generated stimulus.

We define the terminal condition as the voltage above which an action potential is guaranteed:

$$V(t_f) = 12\ mV \tag{3-19}$$

The gradient algorithm allows us the flexibility to define the terminal conditions for only one of the four state variables. This is useful because it allows us to broaden the search for stimuli that achieve the outcome (an action potential) without artificially restricting ourselves to solutions that may not be optimal. For our purposes, we only require the stimulus to trigger an action potential irrespective of the values of $m$, $n$, and $h$. Placing constraints on those values could restrict the search to a solution that is not optimal.

Another consideration for setting the terminal condition is whether the first order gradient converges to desired end-point. For example, if we set the terminal condition of $V$ to be near the peak of the action potential (e.g.$V(t_f) = 95\ mV$), the algorithm has a very difficult time converging to a solution that ends at the peak of the action potential. This is due to the extreme nonlinearity of the state variables in that particular region of the action potential. A small change in the stimulus results in either a better performance

index or a lower error in terminal conditions. Because we are using a first-order gradient algorithm, we would often end up either over- or under-stepping the stimulus, thereby causing a failure in convergence. We found that by setting the terminal condition to be lower, we were both able to guarantee the desired outcome (an action potential) and convergence of the gradient algorithm towards a solution.

## Implementing the gradient algorithm for the Hodgkin-Huxley model

We defined optimization by trying to minimize the $L^2$-norm of a stimulus that takes us from the initial stationary state to the threshold for induction of an action potential. Thus our performance index is

$$J = \int_{t_0}^{t_f} u^2 dt \tag{3-20}$$

The influence functions for the Hodgkin-Huxley model are:

$$\dot{p}^T = -p^T \begin{bmatrix} \frac{d\dot{V}}{dV} & \frac{d\dot{V}}{dm} & \frac{d\dot{V}}{dn} & \frac{d\dot{V}}{dh} \\ \frac{d\dot{m}}{dV} & \frac{d\dot{m}}{dm} & 0 & 0 \\ \frac{d\dot{n}}{dV} & 0 & \frac{d\dot{n}}{dn} & 0 \\ \frac{d\dot{h}}{dV} & 0 & 0 & \frac{d\dot{h}}{dh} \end{bmatrix} \tag{3-20}$$

with the boundary condition $p^T(t_f) = [0 \quad 0 \quad 0 \quad 0]$, and where

$$\frac{d\dot{V}}{dV} = -120m^3h - 36n^4 - 0.3 \tag{3-21}$$

$$\frac{d\dot{V}}{dm} = -360m^2h(V - 115) \tag{3-22}$$

$$\frac{d\dot{V}}{dn} = -144n^3(V + 12) \tag{3-23}$$

$$\frac{d\dot{V}}{dh} = -120m^3(V - 115) \tag{3-24}$$

$$\frac{d\dot{m}}{dV} = \frac{\varphi(m-1)}{10e^{0.1(25-V)} - 1} + \frac{2m\varphi}{9e^{V/18}} + \frac{\varphi(m-1)\big(0.1(V-25)\big)e^{0.1(25-V)}}{10(e^{0.1(25-V)} - 1)^2} \tag{3-25}$$

$$\frac{d\dot{m}}{dm} = \frac{\varphi\big(0.1(V-25)\big)}{e^{0.1(25-V)} - 1} - \frac{4\varphi}{e^{V/18}} \tag{3-26}$$

$$\frac{d\dot{n}}{dV} = \frac{\varphi(n-1)}{100e^{0.1(10-V)} - 1} + \frac{\varphi n}{640e^{V/80}} + \frac{\varphi(n-1)\big(0.01(V-10)\big)e^{0.1(10-V)}}{10(e^{0.1(10-V)} - 1)^2} \tag{3-27}$$

$$\frac{d\dot{n}}{dn} = \frac{0.01\varphi(V-10)}{e^{0.1(10-V)} - 1} - \frac{\varphi}{8e^{V/80}} \tag{3-28}$$

$$\frac{d\dot{h}}{dV} = \frac{7\varphi(h-1)}{2000e^{V/20}} - \frac{\varphi e^{0.1(30-V)}}{10(e^{0.1(30-V)} + 1)^2} \tag{3-29}$$

$$\frac{d\dot{h}}{dh} = \frac{\varphi}{e^{0.1(30-V)} + 1} - \frac{7\varphi}{100e^{V/20}} \tag{3-30}$$

For this study, we only have one terminal condition, $V$ crosses a certain threshold, leading to an action potential. Thus, the rest of our equations reflect having only one terminal condition.

$$\dot{R}^T = -R^T \begin{bmatrix} \frac{d\dot{V}}{dV} & \frac{d\dot{V}}{dm} & \frac{d\dot{V}}{dn} & \frac{d\dot{V}}{dh} \\ \frac{d\dot{m}}{dV} & \frac{d\dot{m}}{dm} & 0 & 0 \\ \frac{d\dot{n}}{dV} & 0 & \frac{d\dot{n}}{dn} & 0 \\ \frac{d\dot{h}}{dV} & 0 & 0 & \frac{d\dot{h}}{dh} \end{bmatrix} \tag{3-31}$$

with boundary conditions $R^T(t_f) = [1 \quad 0 \quad 0 \quad 0]$ and where the gradients are the same as for the influence function $p$.

The variables $Q$, $g$, and $\delta u$ now become:

$$Q = k \int_{t_0}^{t_f} {R_{11}}^2 dt \tag{3-32}$$

$$g = -k \int_{t_0}^{t_f} R_{11}(-p_1 + 2u)dt \tag{3-33}$$

$$\delta u = -k(2u - p_1 - v_1 R_{11}) \tag{3-34}$$

We set the algorithm to run for 100 iterations, with k = 0.1 and ε = 0.5. To generate the initial stimulus, a uniform distribution random number generator with a range of -1 to 1 was used to generate the stimulus amplitudes at every 0.1 ms interval for a total of 25 milliseconds. Because of the stiffness of the Hodgkin-Huxley equations, we used MATLAB's differential equation solver, ode113 (MathWorks; Natick, MA).

## FitzHugh-Nagumo (FHN) model

While the transition from quiescence to a single action potential is interesting, many biological systems exist in states that are oscillatory in nature. In a recent study, Paydarfar, Forger and Clay [43] showed that small oscillatory stimuli can be used to induce a state transition in a bistable system. While, we could theoretically model the Hodgkin-Huxley model as a bistable system by adding a sufficiently large exogenous depolarizing persistent current, previous studies have shown that the squid axon, which is the basis for the Hodgkin Huxley model, fails to exhibit repetitive firing under the condition of a persistent depolarizing current clamp [56]. Furthermore, our preliminary analysis suggested that the first-order gradient algorithm does not readily converge because of the sensitivity of the Hodgkin-Huxley system, specifically the terminal point, even to small changes in the stimulus. To change the stimulus by a small amount would cause the system to overshoot the terminal condition, causing the gradient algorithm to attempt to reverse the problem, but overshooting the terminal condition again in the opposite direction.

We found the FitzHugh-Nagumo system to be much more lenient as a bistable model. The FitzHugh-Nagumo model is a two dimensional system that has been used to describe excitability in neurons and has served as a model system for bistable behavior [39,54,55,57–60].

The FitzHugh-Nagumo model [54,55] is defined as follows:

$$\dot{x_1} = c\left(x_2 + x_1 - \frac{{x_1}^3}{3} - r\right) + u \tag{3-35}$$

$$\dot{x_2} = -\frac{1}{c}(x_1 - a + bx_2) \tag{3-36}$$

This model is unitless, but one can show that the FitzHugh-Nagumo model is a two-dimensional reduction of the Hodgkin-Huxley equations[54]. With regards to neuronal excitability, $x_1$ is analogous to Hodgkin-Huxley's $V$ and $m$, while $x_2$ is analogous to Hodgkin-Huxley's $h$ and $n$ states [54]. The variable $u$ is analogous to current stimulation, which can be in the form of an endogenous persistent current or an exogenous input stimulus. In order for the system to exhibit bistability (quiescence and repetitively firing), we have chosen parameters as previously defined [57], $a$ = 0.7, $b$ = 0.8, $c$ = 3.0, and $r$ = 0.342. In this particular configuration, the system gravitates, when there is no stimulation, towards one of two states: quiescence or repetitive firing. The repetitive firing state is an oscillatory limit cycle, while the quiescent state is a fixed point. The minimum value of $x_1$ is the equivalent of the peak of an action potential in the FitzHugh-Nagumo model. We wanted to determine the optimal stimulus to both induce and suppress repetitive firing in a bistable system. Thus, we used the gradient algorithm to calculate the optimal stimulus when transitioning from the fixed point to the oscillatory limit cycle, and vice versa.

In order to systematically explore the entire limit cycle in the repetitive firing state, we captured a set of 68 points ($RF_1$, $RF_2$ … $RF_{68}$) dispersed around the limit cycle. We determined this set of 68 points by allowing the system to reach steady state in MATLAB

and recording the values of $x_1$ and $x_2$ as returned by the Runga-Kutta differential equation solver, ode45. To find the optimal stimulus from quiescence to the repetitive firing state, we set up 68 computational experiments, each starting where $x_0$ is equal to quiescent point and ending at $x_f = RF_n$ where $n$ lies between 1 and 68. Similarly to find the optimal stimulus from the repetitive firing state to quiescence state, we set up 68 computational experiments, each starting with $x_0 = RF_n$ where *n* lies between 1 and 68, and ending at $x_f$ equal to the quiescent point. In order to define the phase around the limit cycle, we normalize the time at which each of the 68 points occur so that $RF_1$ occurs at phase $\phi = 0$ and $RF_{68}$ occurs at phase $\phi = 1$. Figure 3.1 shows a graph of all the 68 points, with some of the phases marked off. We ran each computational experiment 10 times to again increase the probability of finding the global optimal for each phase.

## Implementing the algorithm for the FHN model from quiescence to repetitive firing

We set our scaling factors to be $k = 0.5$ and $\varepsilon = 0.5$. To generate the initial stimulus, a uniform distribution random number generator with a range of -1 to 1 was used to generate the stimulus amplitudes at every 0.1 ms interval for a total of 30 milliseconds. Because we were dealing with a bistable system, we wanted to ensure that the system did not revert back to the quiescent state. As such, we observed the system's response for 100 milliseconds, 70 milliseconds after the stimulus had completed. We verified in each of our results that the range of $x_1$ values remained larger than 3.5 units, which was the range of the repetitive firing steady state in the last 20 milliseconds of our 100-millisecond system response. Because the FitzHugh-Nagumo model was less stiff, we

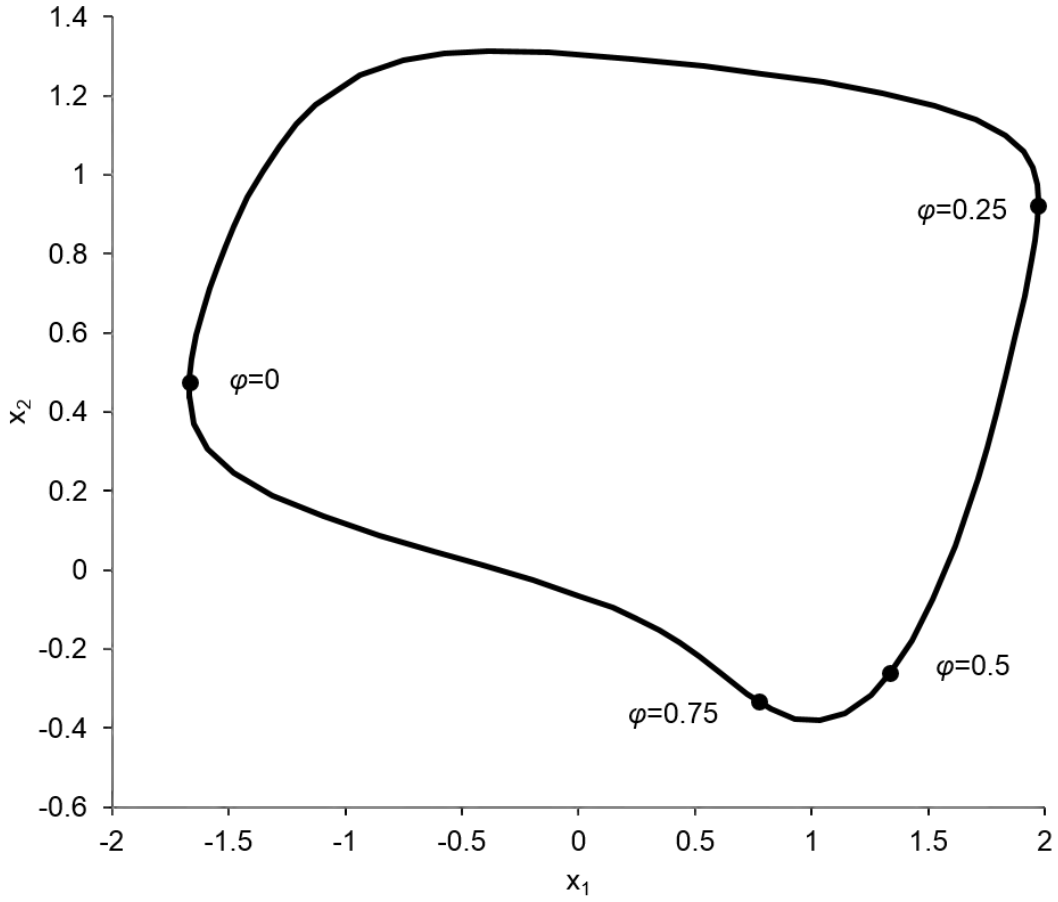


**Figure 3.1: Plot of the limit cycle in the FitzHugh-Nagumo model. The phase values are normalized by time so that 0 and 1 are both at the "peak of the action potential" which occurs when $x_1$ is at its minimal value. Four representative phase values are shown to show the time progression around the limit cycle**

were able to use MATLAB's ode45 differential equation solver. In this application of finding transitions from quiescence to repetitive firing, we are examining the phase at which the stimulus terminates on the limit cycle.

We found that depending on which of the 68 points we were using as the terminal condition, the gradient algorithm would take a variable number of iterations to converge to a solution. Thus, instead of specifying a threshold, we terminated the algorithm when the standard deviation of the last 20 iterations of $\delta J$ was less than 0.0001 and $\delta u$ was less than 0.1. For the sake of time, we terminated the gradient algorithm after 1000 iterations if it had not yet converged.

## Implementing the algorithm for the FHN model from repetitive firing to quiescence

We set our scaling factors to be $k = 0.5$ and $\varepsilon = 0.5$. To generate the initial stimulus, a uniform distribution random number generator with a range of -1 to 1 was used to generate the stimulus amplitudes at every 0.1 ms interval for a total of 8 milliseconds. In order to develop results comparable to previous literature [39], we've chosen a stimulus duration of 8 milliseconds, which is less than the limit cycle period (12.84 ms). In this application of finding transitions from repetitive firing to quiescence, we are examining the phase at which the stimulus begins from the limit cycle.

Because we were dealing with a bistable system, we wanted to ensure that the system did not revert to the repetitive firing state. As such, we observed the system's response for 100 milliseconds, 92 milliseconds after the stimulus had completed. We verified in each

of our results that the range of $x_1$ values remained smaller than 3.5, which was the range of the repetitive firing steady state in the last 20 milliseconds of our 100-millisecond system response. Again, because the FitzHugh-Nagumo model was less stiff, we were able to use MATLAB's ode45 differential equation solver.

As we did for determining convergence when finding the optimal stimulation from quiescence to repetitive firing, we terminated the algorithm when the standard deviation of the last 20 iterations of $\delta J$ was less than 0.0001 and $\delta u$ was less than 0.1, with a maximum number of iterations set at 1000.

# Results

In order to achieve a better understanding of how the gradient algorithm performs, its advantages and its limitations, we applied it to three distinct scenarios: the triggering of a single action potential in a monostable system, the initiation of repetitive firing in a bistable system and the suppression of repetitive firing in a bistable system. We proceeded with two of classic neuronal model systems: the Hodgkin-Huxley model as our monostable system and the FitzHugh-Nagumo system as the bistable system. While the FitzHugh-Nagumo system has been used mainly in neuronal systems, it also has broader applications in other biological systems [61,62].

## Hodgkin-Huxley model of neuronal excitation

Figure 3.2 shows the gradient algorithm beginning with a randomly generated stimulus, and within a few iterations, we see the rough shape of the optimal stimulus. Within 30

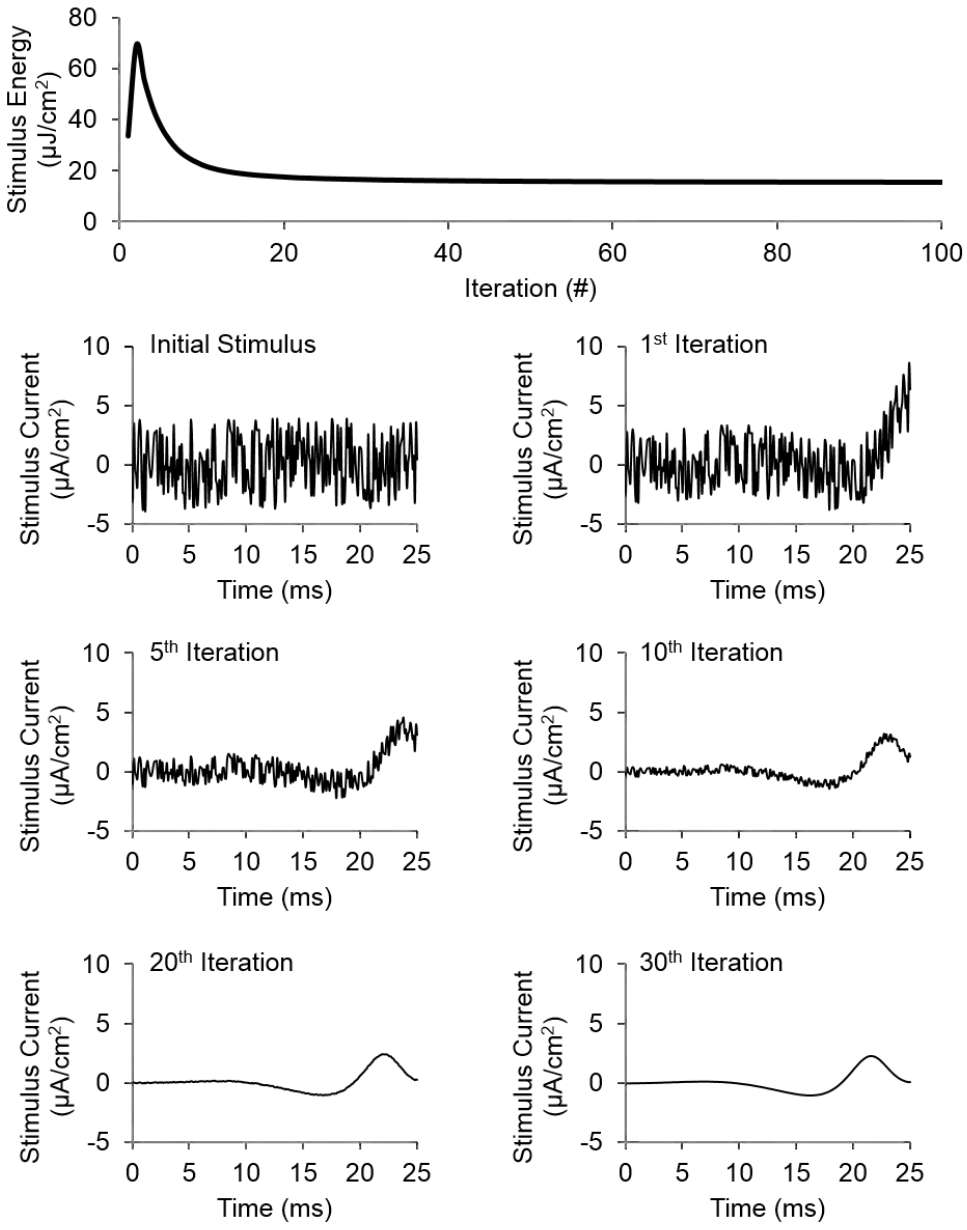


**Figure 3.2: Gradient algorithm shapes random stimuli towards an optimal waveform. Here we show the progression of solutions as the gradient algorithm begins with a white-noise stimulus and finds the most energetically efficient solution over 100 iterations. The top panel shows the $L^2$-norm trajectory over 100 iterations. The six panels show the evolving waveform at the 1st, 5th, 10th, 20th, and 30th iteration of the algorithm. There is very little improvement between the 30th and the 100th iteration as seen in the $L^2$-norm trajectory (top panel). By convention, positive current is depolarizing**

iterations, the algorithm reveals the optimal stimulus with little further improvement in $L^2$-norm following subsequent iterations. One interesting observation we noted is that the first iteration often produces a stimulus with a poorer performance index. This is because we are randomly generating the stimulus, which most likely fails to meet the terminal condition. Thus, the algorithm first changes the stimulus to be closer to meeting the terminal condition, before it begins improving the performance index.

Figure 3.3 shows the stimulus waveform of gradient algorithm at 100 iterations and the action potential it caused. Note that the membrane potential $V$ in Figure 3.3 is offset from the HH model by -60 mV, which is also consistent with modern usage of the Hodgkin-Huxley model. The original model arbitrarily set the resting potential at 0 mV, while neurons actually rest in a hyperpolarized state. The result of the gradient algorithm had an $L^2$-norm of 15.5 $\mu J/cm^2$. As a point of comparison, we used a constant amplitude 25-ms stimulus and reduced the amplitude down until it just barely created an action potential. We found that using an amplitude of 2.255 $\mu A/cm^2$, the neuron would fire an action potential at 11 ms. We then reduced the duration of the stimulus until it no longer fired an action potential, and found that when the amplitude was 2.255 $\mu A/cm^2$ and the duration was 9.7 ms, the neuron would just barely fire an action

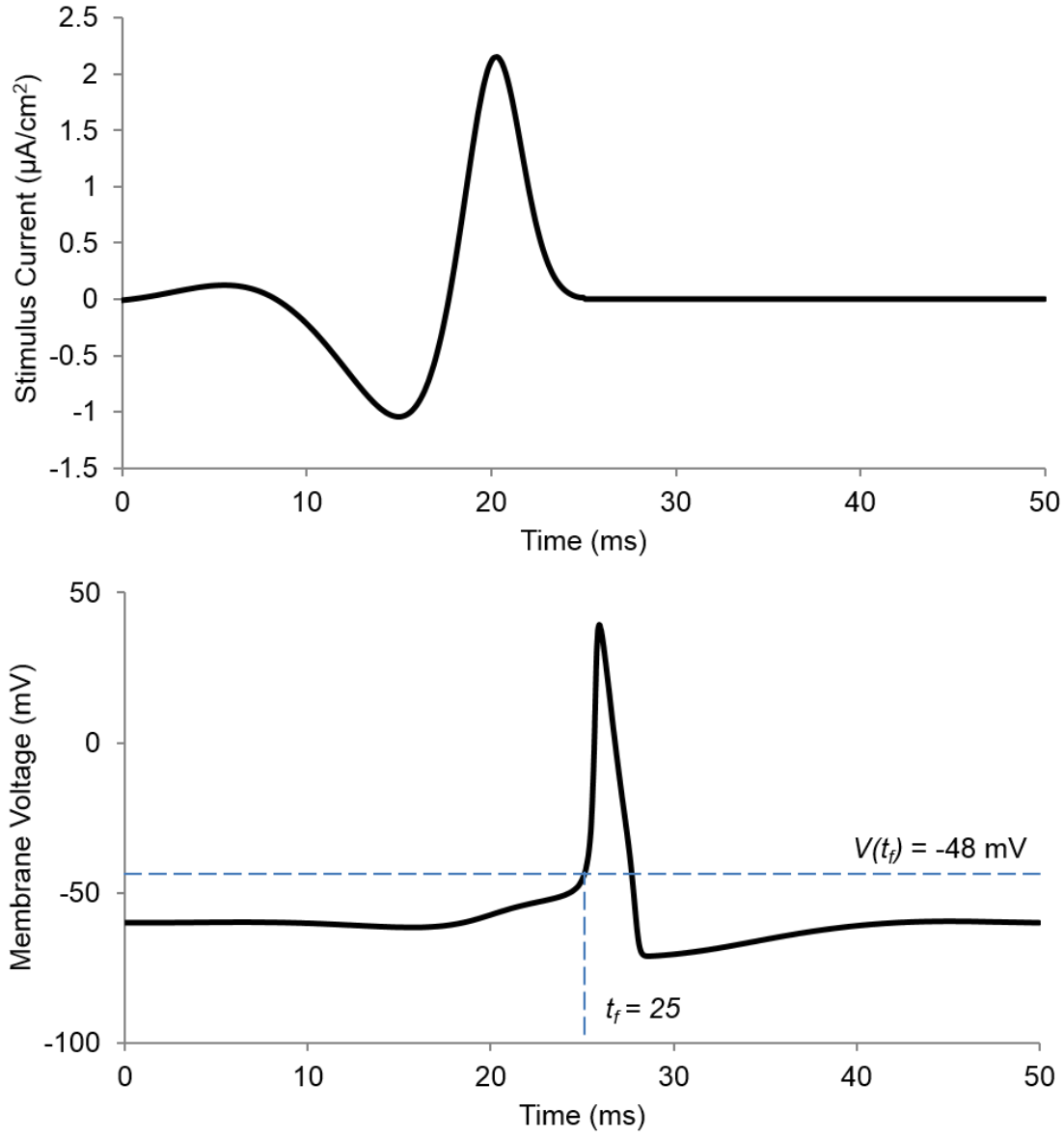


**Figure 3.3: The optimal stimulus derived from the gradient algorithm (top) triggers a single action potential in the Hodgkin-Huxley model (bottom)**

potential. The $L^2$-norm of this barely supra-threshold rectangular pulse was 49 $\mu J/cm^2$. Thus, the waveform generated using this technique showed a large reduction of energy necessary for causing the neuron to fire a single action potential.

As we sought to gain a better understanding of the algorithm's application, we examined the effect that stimulus duration would affect the optimal stimulus' $L^2$-norm. We changed $t_f$, but kept the rest of the system parameters constant. This meant that not only did the stimulus decrease in duration, the action potential occurred sooner as well. Figure 3.4 shows the plot illustrating how the stimulus duration affected the optimal stimulus' $L^2$-norm. As we can see from the plot, when the stimulus duration was increased, smaller $L^2$-norms could be achieved, up until a certain point. Beyond 25 milliseconds, the improvements in $L^2$-norm are minimal.

Figure 3.5 shows the shape of the optimal waveforms under the different conditions of when the system crosses the specified threshold. It is interesting to note that when an action potential is desired earlier (less than 7 ms), the optimal waveform is monophasic; whereas action potential timings that are later (more than 7 ms) are optimally achieved with biphasic waveforms. In a recent study, Clay, Forger and Paydarfar [63] explained that the hyperpolarization phase of the optimal stimuli is useful for removing a small amount of sodium inactivation, thus allowing for a less energetic depolarizing phase to still elicit an action potential. This figure shows that this is indeed true, but only when the desired time to action potential is long enough. If the time to action potential is

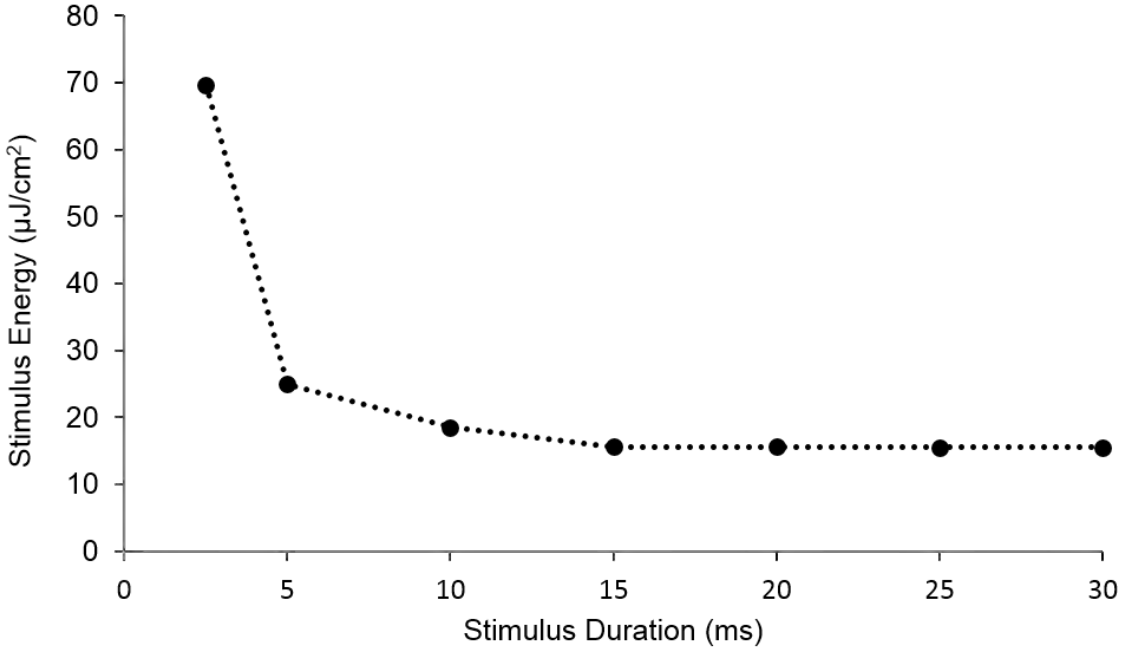


**Figure 3.4: Longer stimulus duration provides for more energetically efficient stimulus. Once the stimulus duration extends past a certain point, there is no further improvement in energy efficiency**

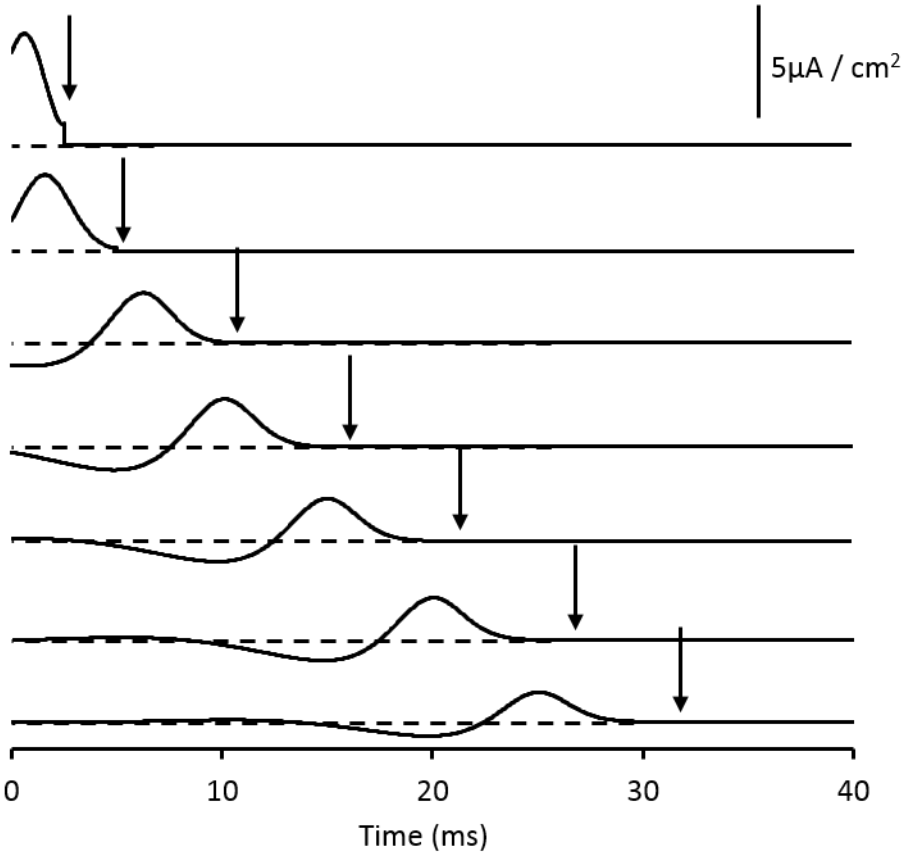


**Figure 3.5: The optimal waveforms change from a monophasic stimulus to a biphasic stimulus as the amount of time to action potential increases**

shorter, the hyperpolarization phase is reduced, and it disappears all together if the time to action potential is too short. From a design perspective, this may suggest that in neuronal systems that require rapid elicitation of an action potential, excitatory postsynaptic currents would be much more prevalent than in neuronal systems that are more amenable to delayed elicitation of action potentials.

We examined the robustness of the gradient algorithm to randomly generated initial estimates of the optimal stimulus. As such, we created a set of randomly generated seeds with varying amplitudes (at 0.1 ms resolution), and tracked for each iteration both the $L^2$-norm and the distance the calculated endpoint was from the desired terminal state. Figure 3.6 shows a plot of $L^2$-norm trajectory over the first 30 iterations. While each trajectory started at different places, they all ended at solutions with the roughly the same $L^2$-norm. There is some variation in the error due to terminal conditions, but the system is extremely sensitive, and so any small changes in the stimulus will result in small variations in the distance of the end point to the terminal conditions.

## The FitzHugh-Nagumo model: quiescence to repetitive firing

The $L^2$-norm values for all 10 runs of each of the 68 computational studies are shown in Figure 3.7. For each of these runs, the system starts at the quiescent fixed point and terminates at a specified phase, $\phi$, as defined previously. The gradient algorithm failed to converge within the set limit of 1000 iterations for some of the runs, and we have

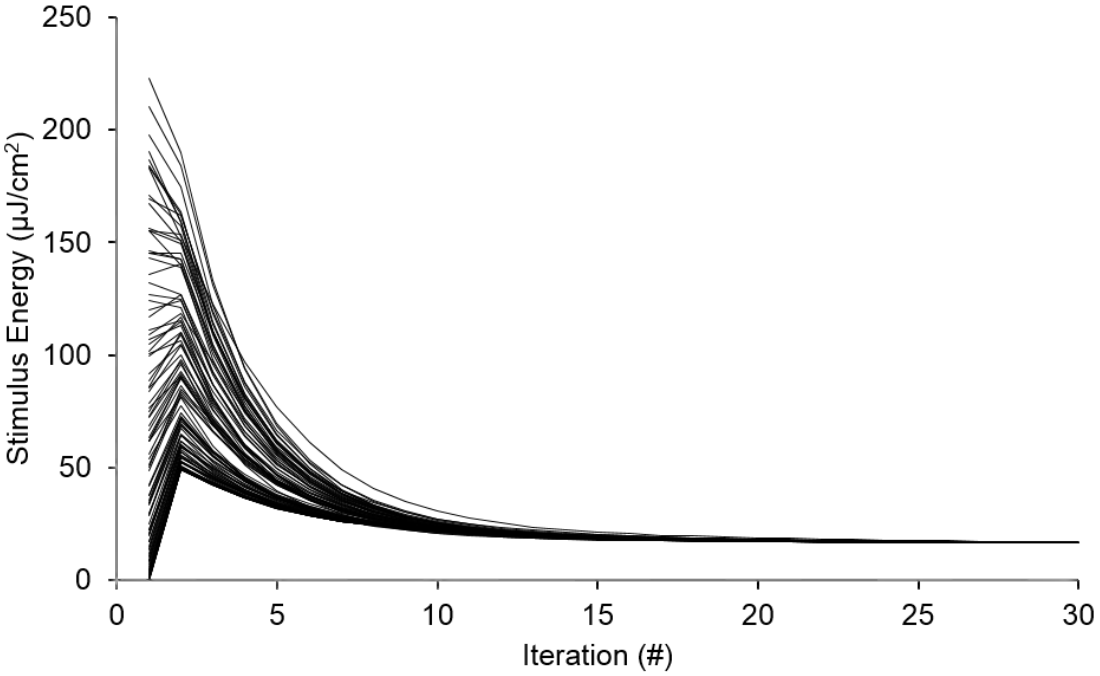


**Figure 3.6: Gradient algorithm is robust to the initial stimulus. The stimulus energy trajectories of 100 different seeds with different amplitudes are shown**

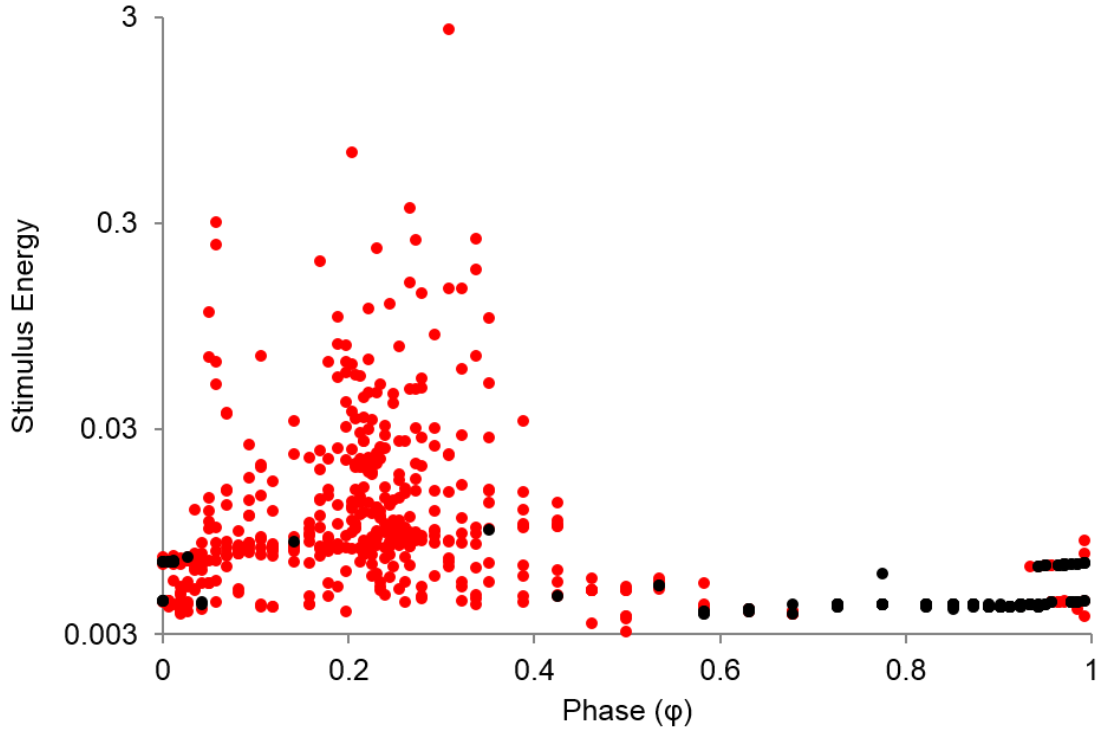


**Figure 3.7: Only certain phase regions converged when transitioning from quiescence to repetitive firing. The points that converged are marked in black, while those that failed are marked in red.**

marked them accordingly. This plot reveals a clustering of low $L^2$-norm values around $\phi = 0.58$, a lack of convergence in the range $0.05 < \phi < 0.5$, and a multiplicity of solutions where $\phi > 0.9$ or $\phi < 0.05$.

To begin, we notice that the optimal transition from the quiescent state to the repetitive firing state is when the terminal condition is around $\phi = 0.58$. In fact, there is a small range, $0.55 < \phi < 0.65$ where the gradient algorithm produces consistently low $L^2$-norm values. This shows a particular trajectory with the most ideal "entrance" into the repetitive firing state from the quiescent state. At these phases, there appears to be only one extrema in the solution space.

Secondly, there is a large range of phases, in which the gradient algorithm is unable to converge to a solution with the predetermined number of iterations. We increased the number of iterations to 5000, but the gradient algorithm still failed to converge to a solution in the range $0.05 < \phi < 0.4$, which corresponds to rapid changes in the state variables during the action potential. Because we are using a first-order gradient algorithm, as the algorithm gets closer to those solutions, small changes in the stimulus can cause the algorithm to overshoot its estimation of the optimal, leading to loss of convergence. We believe that a solution exists because the terminal condition is on a steady state limit cycle. If we took the solution when $\phi = 0.58$ and padded the end with zeroes, we should be able to find a stimulus that takes us to a phase angle between 0.05 and 0.4. This suggests that perhaps the set of initial estimates that allow for convergence to a local optima solution, also known as the region of convergence, is small compared to

the region of convergence we have seen for the mono-stable Hodgkin-Huxley or the bistable FitzHugh-Nagumo when the terminal condition is within the range of $0.55 < \phi < 0.65$.

Finally, we notice multiple local optima found when the terminal condition on the limit cycle is close to the peak of $x_1$ ($\phi < 0.05$, and $\phi > 0.9$). Figure 3.8 shows an example of two stimulus waveforms that resulted in a transition from quiescence to repetitive firing using the exact same starting and ending points. As illustrated here, this multiplicity occurs because the terminal condition is on an oscillatory limit cycle. In this example, two distinct optimal stimuli result because there can be a multiplicity of subthreshold oscillations before the trajectory jumps to the stable limit cycle. In this example, the algorithm converged to two optimal stimulus waveforms that caused either one or three subthreshold oscillations before inducing a jump to the limit cycle. The stimulus inducing a jump after one subthreshold oscillation has shorter duration and larger amplitude, compared to the optimal stimulus that transitions more gradually. This finding matched one of the results we found in the Hodgkin-Huxley model: If we want to transition states quicker, more energy is required.

To provide a comparison, we again calculated the optimal rectangular pulse to switch the FitzHugh-Nagumo model from quiescence to repetitive firing and found that the best stimulus had an amplitude of 0.11 and a duration of 21.102 resulting in an $L^2$-norm

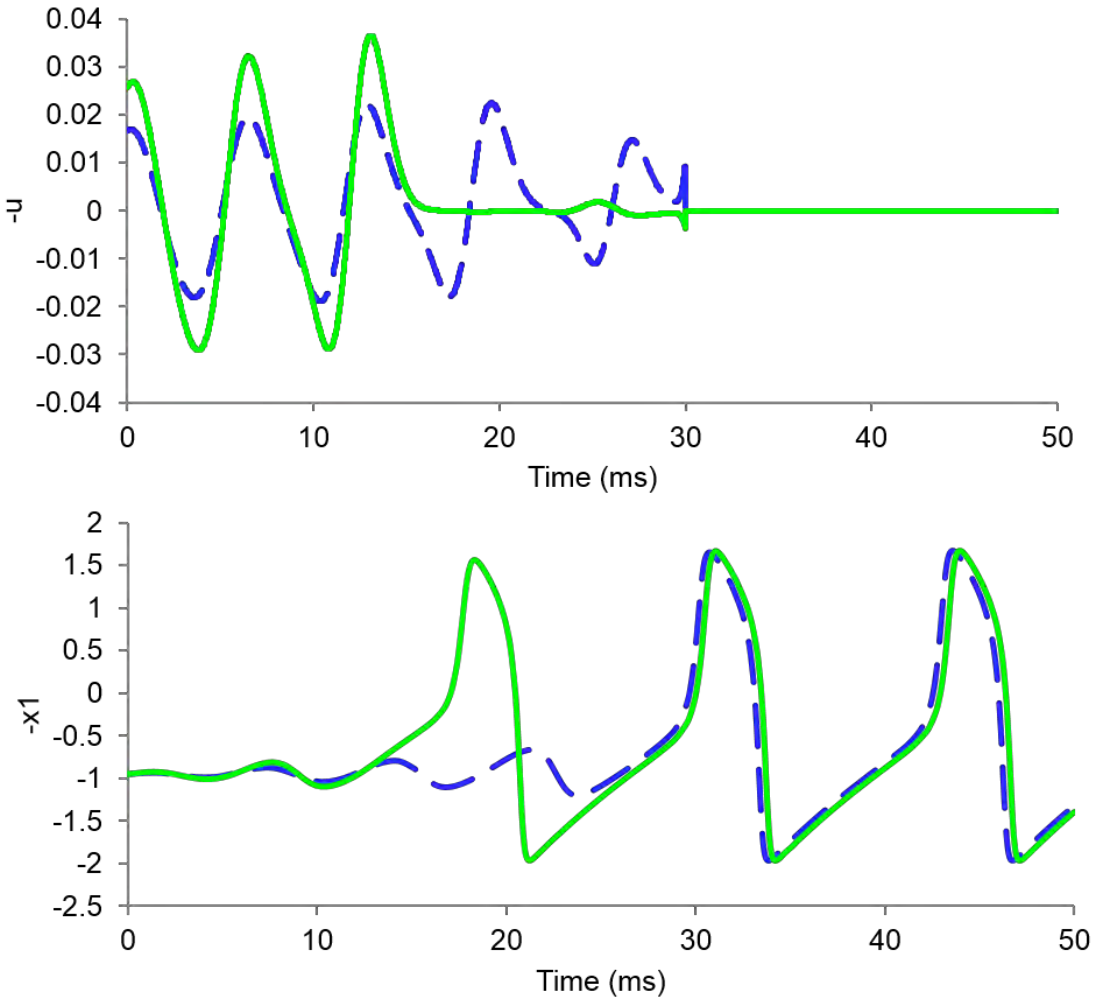


**Figure 3.8: Gradient algorithm finds multiple optimal solutions that induce a transition from quiescence to repetitive firing. The initial condition (quiescent fixed point) and the terminal condition ($\phi = 0$) of the gradient algorithm are the same for both trials. The only difference is the initial randomly generated stimulus that is given to the gradient algorithm. The top panel shows the two optimal stimuli (green and blue), while the bottom panel shows the response from the $x_1$ variable in the FitzHugh-Nagumo model to the stimuli (matched green and blue).**

of 0.26. The results from the gradient algorithm ranged from 0.0038 at the best and 0.0097 at its worst. Here we can see a substantial improvement over standard rectangular pulse stimulus.

Like our results in the Hodgkin-Huxley model, Figure 3.9 shows the paths toward optimality resulting from different initial randomly generated seeds. Although the different seeds have a broad range of $L^2$ norms and distances to the terminal condition, almost all of them converge to one of the two clusters. We found that of the solutions that converged, 78% of the randomly generated seeds converged to the larger $L^2$-norm solution and 22% converged to the smaller $L^2$-norm solution. We note in Figure 9 that there are some points that drop $L^2$-norm very sharply with successive iterations, almost as if attracted to the lower local optimum, but then bounce back up, settling into the upper local optimum. This phenomenon is because a single iteration may cause the $L^2$-norm to drop by a lot, but it may increase the error in the terminal conditions. In the next iteration, the gradient algorithm corrects the error in the terminal condition, leading to the bounce back in the $L^2$-norm value.

## The FitzHugh-Nagumo model: repetitive firing to quiescence

Figure 3.10 shows the $L^2$-norms of 10 runs each of the 68 computational studies from each of the points along the repetitive firing limit cycle to the quiescent point. In all cases, the stimulus duration was set at 8 ms. From the figure, we can see that there is again an optimal window where $\phi$ is between 0.4 and 0.6 at which to begin transitioning

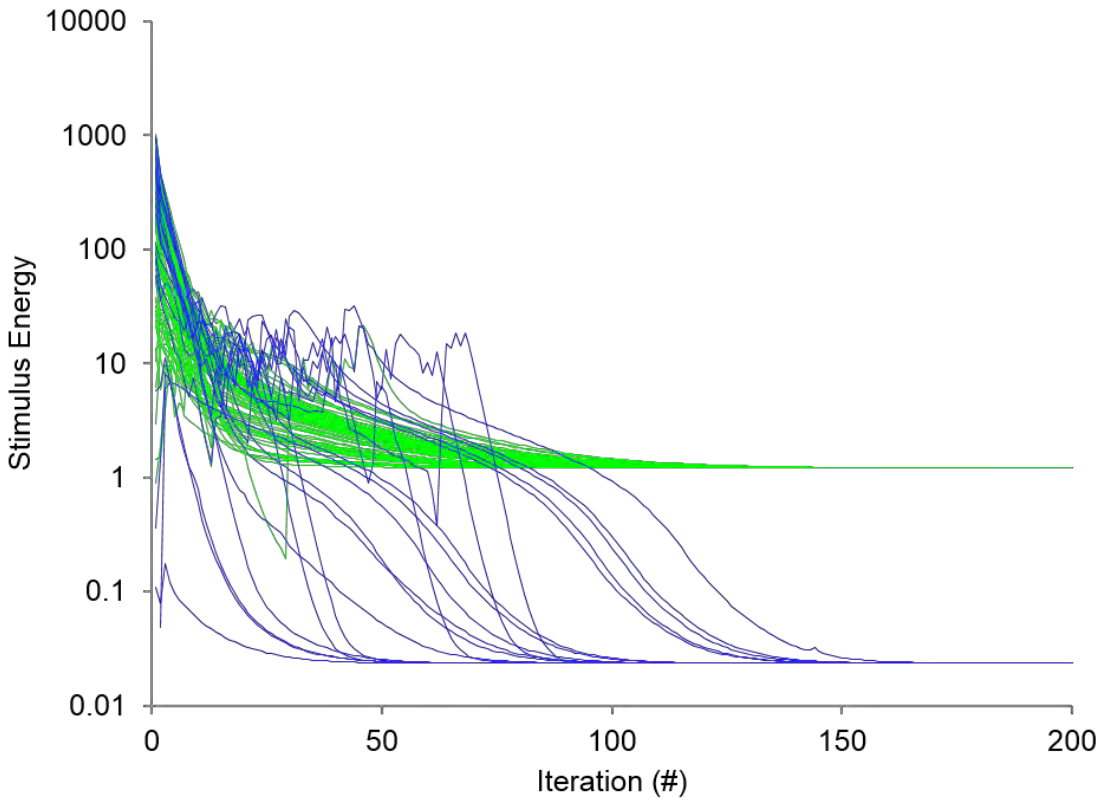


**Figure 3.9: Trajectories of stimulus energy through 200 iterations of gradient algorithm show convergence to two waveforms. We ran 100 different randomly generated seeds with different scaling factors. The green and blue colors match the respective green and blue waveforms seen in Figure 8**

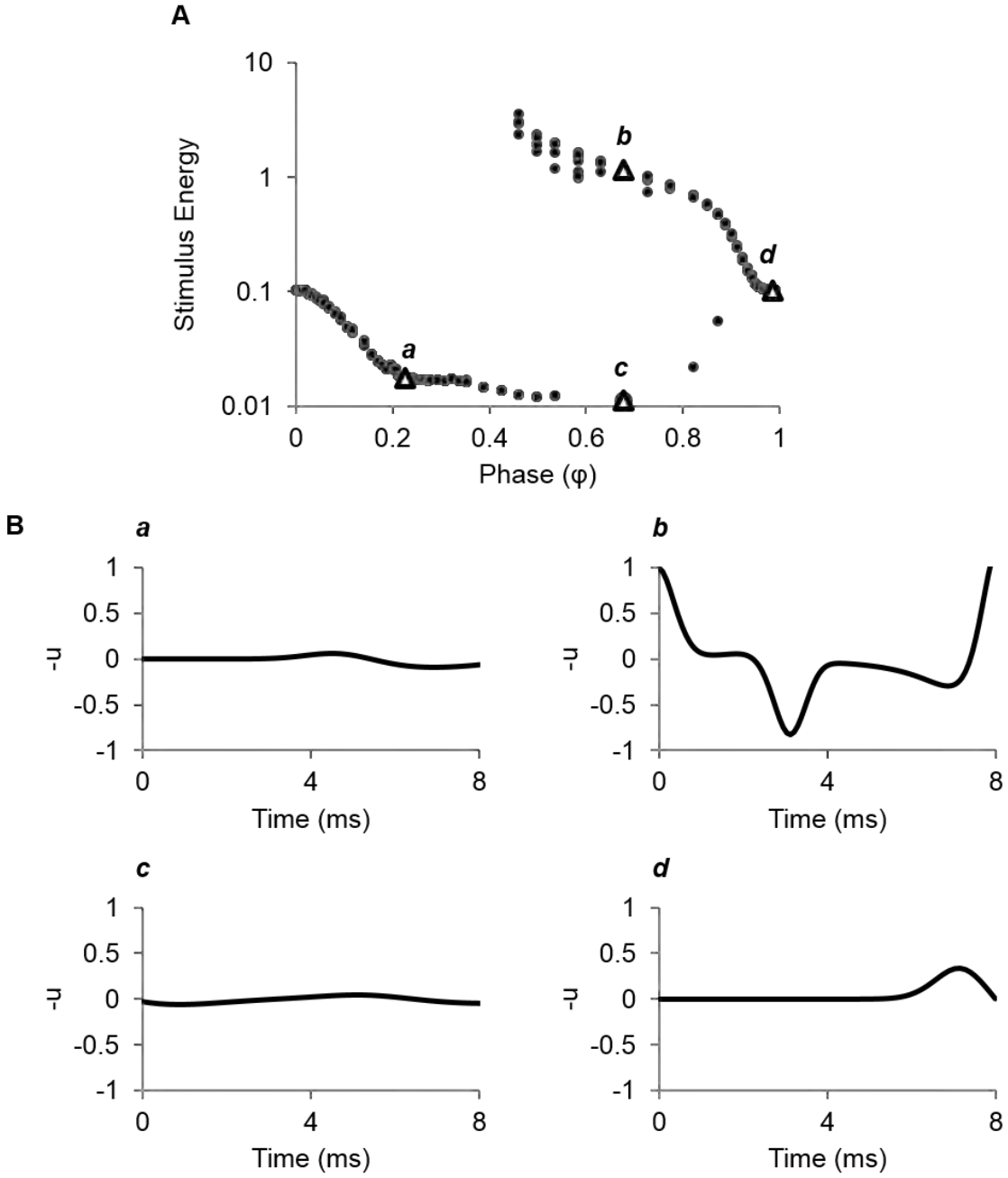


**Figure 3.10: Gradient algorithm reveals multiplicity in transitioning from repetitive firing to quiescence across different phases. The top panel shows the stimulus energy of the different optimized stimuli (8 ms duration) that induce transitions from different phases of repetitive firing to the quiescent fixed point. Specific examples of different solutions are shown in the bottom pane, labeled *a* through *d*. Note that *b* and *c* are solutions with the same starting phases.**

from the repetitive firing limit cycle to the quiescent fixed point. This corresponds to the location closest to the quiescent fixed point. Around this window, the best $L^2$-norm values sit around 0.012. In comparison, the most optimal constant stimulus waveform has an $L^2$-norm of 0.064, for a duration of 7.87 ms and an amplitude of 0.09 given at $\phi$=0.72. It is interesting to note that when using a constant stimulus waveform, the discrete stimulus fails to suppress repetitive firing across a large range of phase angles. In contrast, the gradient method enables discovery of novel waveforms that induce a transition from a broad range of phases around the limit cycle to the quiescent state.

We had discussed how multiplicity occurred in the earlier example of stimuli that induce the FitzHugh-Nagumo neuron to transition from the quiescent state to the repetitive firing state, due to the cyclical nature of the system. Here, we can see that even when the stimulus duration is smaller than one cycle, we can find multiplicity with the FitzHugh-Nagumo system. We chose the example where $\phi = 0.678$. Figure 3.11 shows the convergence of a set of randomly generated initial conditions towards the two different stimuli. Figure 3.12 shows two stimuli that start from the same point on the repetitive firing limit cycle and end up at the quiescent point.

One interesting implication about this particular result is that there is potential for applying the gradient algorithm to phase shifting an oscillatory system. What we can see here in Figure 3.12 is that one of the solutions transitions almost immediately into the quiescent point, while the other makes a loop around the limit cycle before entering

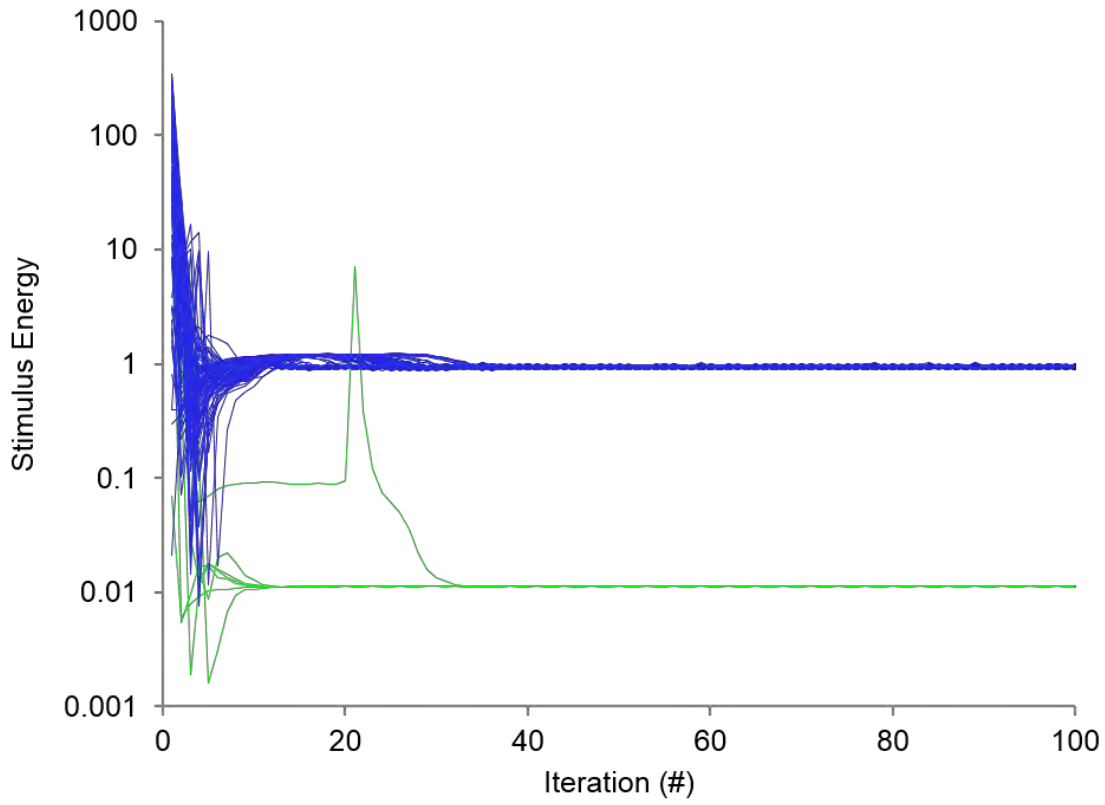


**Figure 3.11: Random stimuli converge towards two different optimal stimuli that induce a transition from repetitive firing to quiescence. The trajectories of stimulus energy through 100 iterations of gradient algorithm are shown here. All of these stimuli start from the same starting phase ($\phi$ = 0.678).**

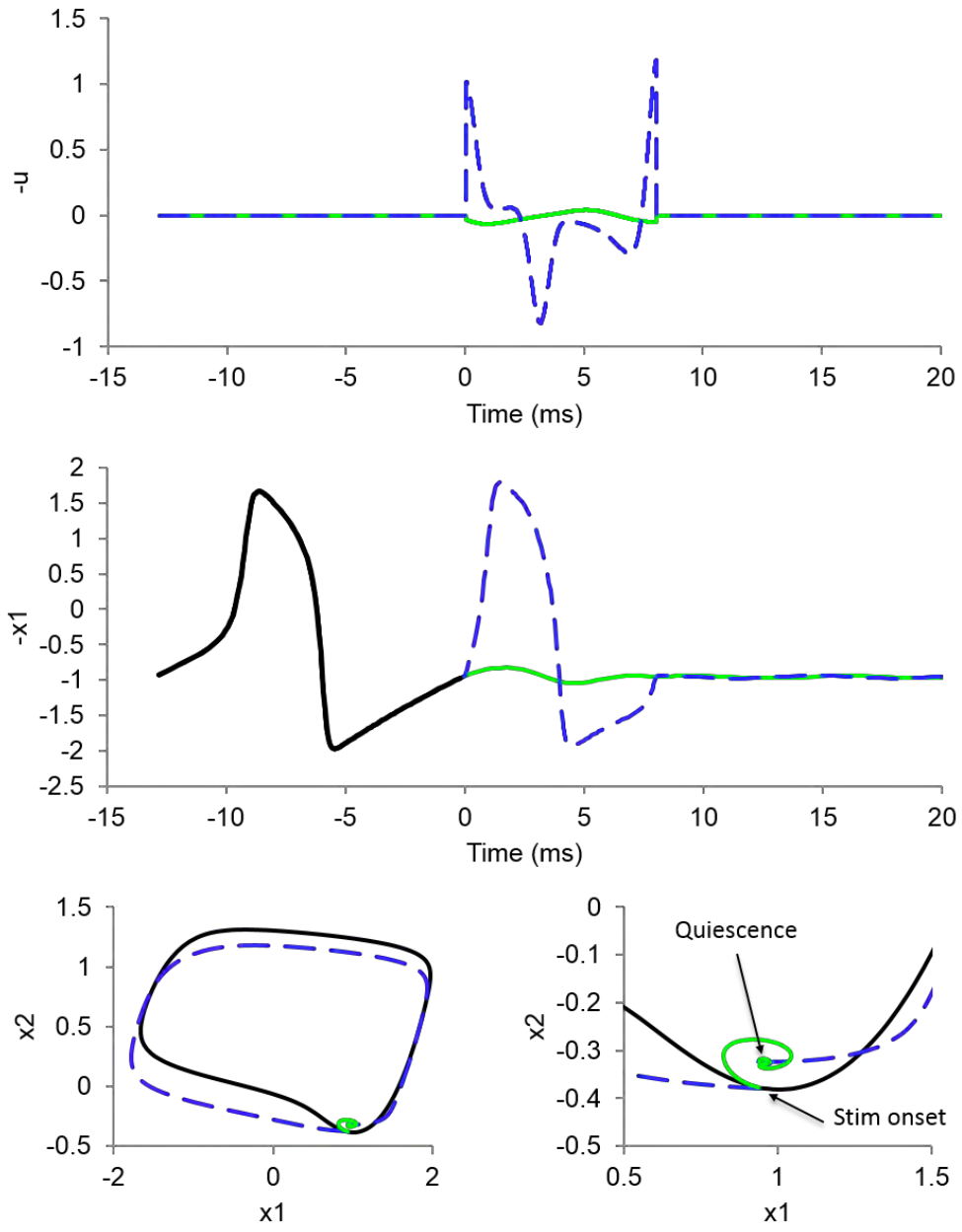


**Figure 3.12: Gradient algorithm reveals different mechanisms of suppressing repetitive firing. The top panel shows the stimuli, the middle panel the $x_1$ response to the stimuli. The bottom panel shows the entire state space response to both stimuli. As we can see, one stimulus (green) suppresses the system quickly, while the other stimulus (blue dashed) causes the system to run more quickly around the oscillatory state before entering into the quiescent state. The black line in the figure is marked to show the first course of the system without any stimulation. This marks the limit cycle of the repetitive firing state.**

into the quiescent point. This is interesting because the normal period of one cycle is 12.84 ms. We captured an instance where the stimulus in 8 ms has traveled around the entire limit cycle. We could theoretically set up the gradient algorithm to travel from one point on the limit cycle to a different point of the limit cycle, requiring this phase shift to take place within a fraction of the period of the limit cycle. In this way, the gradient algorithm can be used to find optimal phase shifting stimuli [39,64,65].

# Discussion

Optimal control theory is rooted in calculus of variations, developed by Bernoulli, de l'Hôpital, and Euler [32]. However, the real-world applications of calculus of variations did not start until more recently in the 1950s. Because problems in optimal control generally are nonlinear, they do not have simple solutions that can be analytically determined. Thus, a range of numerical methods was developed. These numerical methods fall into roughly one of two broad categories: indirect methods and direct methods [66,67].

Indirect methods use Pontryagin's maximum principle to determine a set of first-order conditions that define the optimal solution. This set of first-order conditions combines the original state variables with an extra set of adjoint variables, one for each of the original states, that measures the influence of the state variables to each other. A boundary value problem (BVP) solver like the shooting method or Newton-Raphson method is then used to solve numerically this new system of equations. The advantage of

the indirect method is that once a solution is found from the BVP solvers, it is easily verified against the first-order conditions captured by calculus of variations.

One of the disadvantages of the indirect method is that the region of convergence around the variables may be smaller with the addition of the adjoint variables. Thus, there is a higher chance of starting a BVP solver at a state that ultimately diverges from the optimal solution. With most numerical algorithms, a good initial guess can avoid starting at locations that diverge from the optimal solution. However, one usually requires some background information or understanding of the equations to be able to provide a good initial seed from which most BVP solvers will work. If a bad seed is chosen, the BVP solver will diverge away from the solution. With the adjoint variables, developing the initial seed becomes even more difficult as they do not have any physical interpretation by which to understand how they relate to the other variables, and thus it is difficult to even develop an initial estimate for these variables by which to seed the algorithm. To this point, most researchers have attempted to simplify the Hodgkin-Huxley model in order to circumvent these two disadvantages. Researchers have used a phase reduction model of the neuron [37,38,68], parameterized the stimulus [69], or used simpler "integrate-and-fire" models [34,35]. To our knowledge only Forger et al [33] have applied the indirect method to solve the original Hodgkin-Huxley model in its complete form.

The direct method, first proposed by Kelley [44] and further developed by Bryson and Ho [40], does not create a surrogate system of equations, but instead uses the original

system of equations to iteratively move towards a more optimal solution. This method does not add any new parameters and thus avoids the need to calculate another set of variables and first-order derivatives. Furthermore, by not adding a new set of parameters to the systems of equations, it allows for a larger region of convergence from the initial estimate. Indeed, we have shown that randomly generated stimulus waveforms can be used to find optimal solutions for induction of an action potential in the monostable Hodgkin-Huxley equations, as well as induction or suppression of repetitive firing in the bistable FitzHugh-Nagumo equations. We were able to show with both models that the solutions converged even though they started from very different seeds. Because of this, *a priori* knowledge is not needed to find locally optimal solutions. In order to explore the solution space further and find a global optimal solution, we have developed this algorithm to include a stochastically seeding component. By running the algorithms with different randomly generated seeds, the algorithm allows for more of the solution space to be explored, thus allowing one to have greater confidence that the best solution found is indeed the global optimal.

One of the advantages of using this gradient algorithm is its ability to find optimal solutions even when terminal conditions are not defined for all of the state variables. As we noted in finding the optimal stimulus necessary to trigger a single action potential, the gradient algorithm did not require us to have the terminal condition defined for all four state variables. We were able to construct the algorithm such that it found the optimal stimulus necessary to achieve a membrane voltage above the threshold for an action potential. Considering that our specific goal was to find the optimal stimulus necessary

to elicit an action potential, this allowed us to find the optimal without adding any extra restrictions on the terminal conditions. By adding more terminal conditions than we need, we are actually restricting our search for a global optimum and including certain assumptions into the algorithm that may lead to sub-optimal solutions.

Our computational study using the gradient algorithm has shown the importance of precisely defining the optimality problem, lest we actually find an optimal solution to a different problem. In our study, we wanted a globally optimal solution, and so we left the terminal condition to include only the voltage threshold. In the study of Forger et al [33], specific states were obtained using *a priori* knowledge of squid axons that were related to different biological mechanisms leading to action potentials. By applying this knowledge, they were able to find two unique optimal solutions specific to the two unique mechanisms. Although we imposed no restrictions on the physiological mechanism for eliciting action potentials in this study, we did find that constraints on the timing of a spike resulted in qualitatively different optimal waveforms (Figure 3.5).

While the gradient algorithm has been advantageous for our applications, it has also shown some of its limitations. Because we were using a first-order system, there were terminal conditions to which the algorithm was unable to converge, due to the highly nonlinear sensitivity of the system to the stimulus in certain regions. We have found that in these situations, the first-order gradient will overshoot the optimal solution and cause the algorithm to iterate through a worse solution. From here, the algorithm re-iterates to improve the solution again towards the optimal, but repeatedly fails near the terminal

condition by overshooting again. We note that this pattern occurs most often in rapidly changing regions in our systems (e.g. near the peak of the action potential). In areas that are less sensitive to changes in the stimulus, the gradient algorithm performs very well.

It is possible to use the second-order gradient in the algorithm as well in order to prevent overshooting. However, the second-order gradient algorithm is much more sensitive to the initial estimate, and thus has a difficult time even beginning to iterate towards an optimal solution [40] as the region of convergence is much smaller. One alternative proposal is to combine both the first-order and the second-order gradient algorithms in order to maximize the first-order's ability to converge quickly at the beginning, with the second-order's ability to converge more accurately at the end [45].

We have shown how stochastically-seeded gradient algorithm can be applied to finding energetically optimal stimuli for transitioning various biological systems from one state to another, whether it is from a quiescent state to a single action potential, a quiescent state to repetitive firing, or from repetitive firing back to quiescence. In this study, we have highlighted this algorithm and its use in gaining insight into the Hodgkin-Huxley system and the FitzHugh-Nagumo system when evaluating optimization based on $L^2$-norm. Future work may focus on applying the same techniques to neurons that exhibit a much wider repertoire of behaviors [70–74], which have been classified extensively using bifurcation theory. These techniques can also be applied to more complicated models like those that describe the impact of deep brain stimulation to treat Parkinsonian tremors [22,23,75–78] and epileptic seizures [25,26,79–81].

Furthermore, while applying this algorithm to finding optimal external electrical stimulation, we postulate that the stochastically-seeded gradient algorithm can also aid in gaining insight into the design principles governing endogenous neuronal stimulation. There has been a wealth of research recently focusing on understanding fundamental design principles that govern neuronal excitation, for instance in elucidating how sensory percepts are encoded [82–84], as well as to populations of neurons within functioning networks to better understand how information is transmitted from neuron to neuron [85–88]. In order to gain insight into how endogenous stimulation are optimized, one could use equations relating to different optimization rules, incorporate that into the performance metric in the algorithm, determining a theoretical result verified or refuted through experimental techniques.

# CHAPTER IV

# Stochastic Search Algorithms: Review of Applications in Medicine

## Introduction

While the gradient algorithm has shown to be quite robust, it requires the existence of a mathematical model. Biological systems are often so complex that mathematical models are not available. As such, one needs alternative model-independent techniques to seek out optimality. Researchers have developed various algorithms that use stochasticity, or random processes, to explore the solution space in search of these optimal solutions.

Stochastic search algorithms are a class of methodologies used heavily in the engineering and computational disciplines to find optimal solutions to problems that have extremely large solution spaces. These algorithms seek to learn over time what the important features of the solution are by analyzing the outputs of known inputs as seen in Figure 4.1. This mentality is very different to analytical techniques that seek to analyze the system itself in order to solve for the optimal stimulus. Stochastic search algorithms use stochasticity to explore broadly the solution space, while the iterative and adaptive nature of the algorithm helps focus it towards the optimal solution. While these algorithms are

still relatively new, researchers use them for a wide range of applications from developing optimal traffic light schedules for minimizing gridlock in cities or finding optimal protein structure conformations to minimize energy costs.

As we have stated previously, finding optimal stimulus waveforms is an extremely difficult task due to the infinite number of possible stimulus waveform. In this chapter, we review four prominent stochastic search algorithms used in clinical applications: spike-triggered averaging, stochastic hill-climbing, simulated annealing, and genetic algorithms. We examine their strengths and weaknesses, and we discuss their potential use in finding optimal stimulus waveforms.

# Review of Methods

Like analytical techniques, stochastic search algorithms seek to find the best set of values for a given set of parameters, $u$, in order to minimize or maximize a cost function, $L(u)$. It is important to note that stochastic search algorithms often work with discrete parameters as opposed to the continuous functions that the gradient algorithm used. As such, instead of thinking of the stimulus as $u(t)$, stochastic search algorithms would set up the stimulus as $u[n]$, *w*here $u$ is a vector of discrete parameters, each of which represent a unique time point in the continuous stimulus.

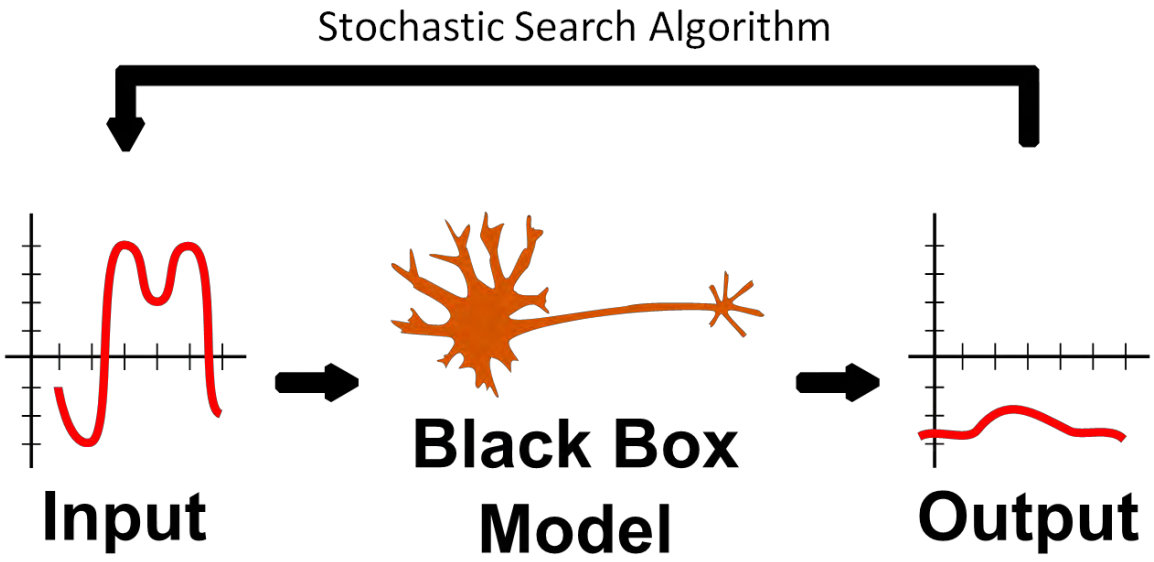


**Figure 4.1: Fundamental concept of stochastic search algorithms**

## Spike-triggered averaging

One of the simplest forms of stochastic search used extensively in experimental neuroscience has been spike-triggered averaging (STA). STA feeds broadband stochastic stimulation into the system, averaging a snippet of the input stimulus preceding each spike generated. Neuroscientists have used this method to gain insight into key stimulus features that govern neural encoding of sensory information[43,89–91]. At a very high level, the algorithm is as follows:

1. Generate a long randomly generated stimulus
2. Apply the stimulus to the system in question
3. For every spike, or event, triggered, save the preceding *T* seconds worth of stimulus.
4. Average all the saved snippets.
5. Scale the average until the resulting stimulus just barely triggers an event.

While the spike-triggered averaging has been used predominantly to gain insight into neural encoding, a few recent studies have shown the potential use of spike-triggered averaging to finding optimal stimulus waveforms[33,39,43]. In the first of these studies, Forger and Paydarfar used the bistable FitzHugh-Nagumo model to compare optimal stimuli found using calculus of variations with those from the spike-triggered average. They examined stimulus waveforms necessary to trigger repetitive firing from quiescence and vice versa. What was interesting about this study is that the general waveform shapes from both methods were qualitatively similar to each other. While quantitatively, calculus

of variations produced a more optimal solution when transitioning from quiescence to repetitive firing, they found that spike-triggered averaging found a better solution when suppressing repetitive firing.

The fact that calculus of variations produced a better answer than spike-triggered averaging was expected because the former methodology utilized analytical techniques to find the best solution. What was surprising from the study of Forger and Paydarfar was the second observation that the spike-triggered averaging produced better results when suppressing repetitive firing. They explained the phenomenon by noting that around the point representing quiescence, there is a stable basin of attraction. This meant that if the stimulus took the system to a state within that basin, it would eventually spiral towards quiescence without further stimulation. Thus, the goal may not necessarily be to get from oscillation to quiescence, but rather to get from oscillation to just past the lip of the basin of attraction. The technique using calculus of variations could not take into account the basin, and as such used more energy to go completely to the center of the basin of attraction of quiescence as opposed to just to the edge.

Furthermore, unlike calculus of variations, spike-triggered averaging did not have any concept of initial conditions. As such, when they examined the results of the two algorithms, they noticed that the timing of when the stimulus is given was different between the two methodologies.

While the research here show the potential spike-triggered averaging has on finding optimal stimulus waveforms, it is unclear why this method appears so effective. In

chapter 5, we examine more closely the relationship between STA and optimality, finding that STA has a few weaknesses that lead it towards suboptimal results.

## Stochastic hill-climbing

One of the ways of finding more optimal solutions with stochastic search algorithms is to include an adaptive or learning component to the algorithm. One of the simplest stochastic search algorithms with an adaptive component is stochastic hill-climbing. While stochastic hill-climbing has not been used to find optimal stimulus waveforms, it has been applied in other medical applications including image classification[92] and analysis[93,94], and vaccination strategies[95].

Stochastic hill-climbing starts with a randomly generated solution and explores a sample of neighboring solutions around that initial seed. It then chooses the best of the samples to be the starting seed of the next iteration. The outline of the stochastic hill-climbing algorithm is as follows:

1. Choose a random initial solution. (e.g. $u = \{x_1, x_2, x_3, \dots, x_n\}$)
2. Add a small noise component to $u$ to move to a neighboring solution, $u_1$ . (e.g. $u = \{x_1 + \alpha_1, x_2 + \alpha_2, x_3 + \alpha_3, \dots, x_n + \alpha_n\}$ where $\alpha_i$ is a small randomly generated increment)
3. Repeat step 2 a few times to generate a set of neighboring solutions: $u_2$ , $u_3$ , $u_4$ , .....
4. Evaluate $F(u_n)$ for all the neighboring solutions.

5. Select the $u_n$ with the best $F(u_n)$.
6. If the best $F(u_n)$ is better than $F(u)$, set $u$ to be $u_n$. Otherwise, keep $u$ to be the same.
7. Repeat steps 2-6 for a defined number of iterations

Figure 4.2 illustrates this algorithm. We have constructed a simple two-dimensional solution space, where each point in the x-y plane represents a solution, $u$. The z-axis represents the performance index, $F(u)$. The algorithm seeks to find the solution that maximizes the performance index. Each arrow shows one iteration through the algorithm. As seen from the illustration, each step is taken in a direction that leads upwards towards the maximum. Because the algorithm does not sample every single possible neighboring solution, the move from one solution to another may not be the most optimal, but it is always closer than the current step.

Figure 4.3 illustrates one of the major weaknesses of stochastic hill-climbing: local optima. If the algorithm reaches a local optimal solution, it will remain there, as any move away from the optimal solution will lead to a poorer performance. One technique that researchers use to get around this is to pick multiple random starting solutions, allowing for greater exploration across the solution space. Some iterations of the algorithm may end at the local optimal solution, but with enough starting solutions scattered randomly across the search space, some iterations may reach the global optimal solution.

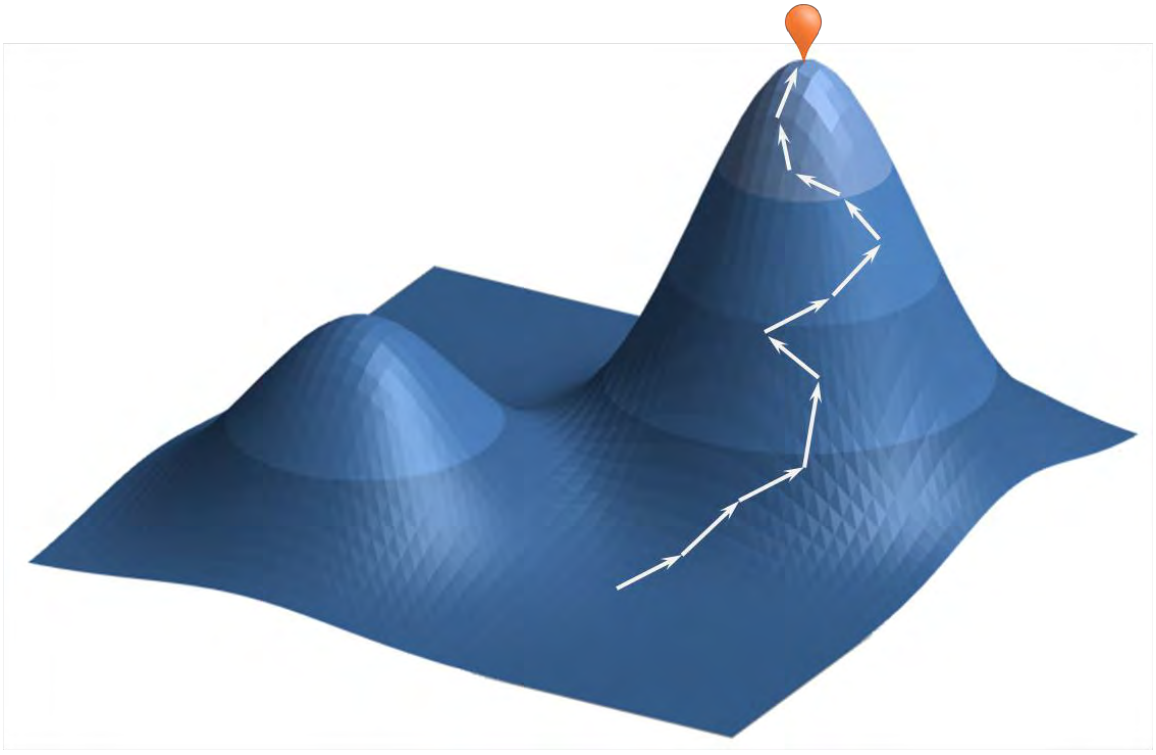

**Figure 4.2: Illustration of stochastic hill-climbing**

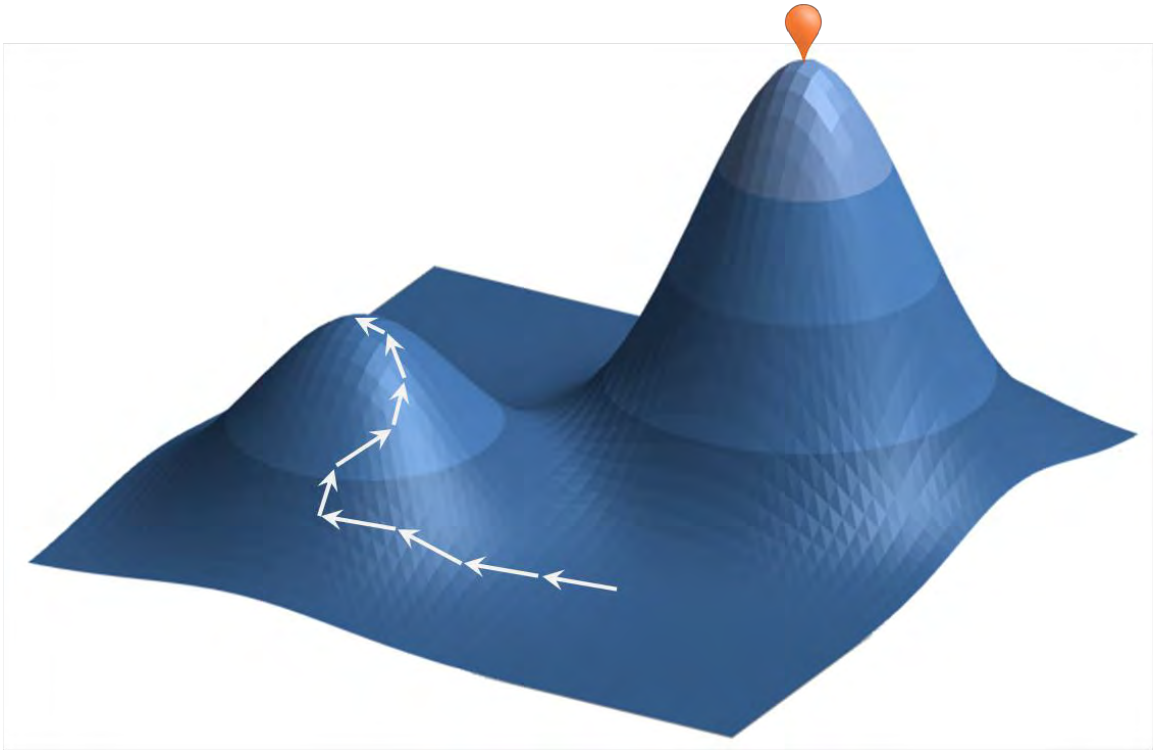

**Figure 4.3: Flaw in stochastic hill-climbing algorithm**

## Simulated annealing

Simulated annealing is similar to stochastic hill-climbing, but it makes allowances to avoid being stuck at a local optimal solution. Kirkpatrick was the first to detail this algorithm, taking inspiration from metallurgy where the heating and cooling of metals in a controlled manner guide atoms towards a more stable, and thus more optimal, state[96]. Like stochastic hill-climbing, simulated annealing has not seen applications relating to the problem of finding optimal time-varying stimulus waveform shapes, however, it has had wide use in radiology[97–101], biochemistry[102–111], image processing[112–115], genetics[116], and therapeutics[117–121].

The advantage of simulated annealing compared to stochastic hill-climbing is that under certain conditions, it allows a worse solution to become the starting seed of the next iteration. Thus, even if the algorithm has arrived at a local optimal solution, it can still potentially move towards away and end up at an even better solution later. Simulated annealing uses a parameter, called temperature, to determine of whether or not a worse solution becomes the starting seed of the next iteration. The fundamental steps of the simulated annealing algorithm are as follows:

1. Choose a random initial solution, $u$.
2. Choose an initial temperature, $T$.
3. Select a randomly generated neighboring solution, $u$ .
4. If $F(u$ ) is better than $F(u)$, select the new solution as the starting seed.

5. If $F(u\ )$ is worse than $F(u)$, decide based on comparing $F(u\ ) - F(u)$ with $T$ to determine whether to select the new solution as the starting seed for the next iteration.
6. Decrease temperature
7. Repeat steps 3-6 for a defined number of iterations

In this algorithm, the rate at which the temperature decreases over time and the exact equation that determines the relationship between temperature and the new solution are all customizable. Often temperature decreases with some relationship to the iteration number. Concerning step 5, the simplest version of simulated annealing uses temperature as a marker of how large the difference can be between the new solution and the current solution. Thus, if the difference is less than the temperature, the algorithm accepts the new solution as the starting seed. As one can imagine, as the temperature decreases over time, the margin of allowing poorer solutions narrows.

While this is the simplest, mathematicians have developed theories about how to define this relationship between temperature and the acceptance of a new solution. The Metropolis-Hastings algorithm is a specific variation of the simulated annealing algorithm, constructing a probabilistic model for the acceptance of poorer solutions based on the temperature parameter[122,123].

## Genetic algorithms

Genetic algorithms are a specific type of stochastic search algorithm, which incorporate elements of learning across a population of solutions. While stochastic hill-climbing and simulated annealing have been the progression of one solution over time, genetic algorithms work with an entire set of solutions all at once, allowing each solution to "learn" from other solutions in that set. These algorithms have gained significant traction in the scientific community in the past two decades. The initial idea for genetic algorithms came from evolutionary biologists who wanted to model evolutionary processes on the computer in the late 1950s and early 1960s[124]. While various computer scientists had incorporated concepts like population, mutation and recombination into their algorithms, it was not until 1975 when John Holland developed the first systematic understanding and application of genetic algorithms as a problem solving mechanism[125].

Genetic algorithms encode solutions as "chromosomes," one-dimensional arrays of parameters, or "genes." Figure 4.4 shows an example of how the genetic algorithm could encode a variety of signals.

A "fitness function," the performance index, is defined to determine how well a solution survives. The fundamental structure of a genetic algorithm is as follows:

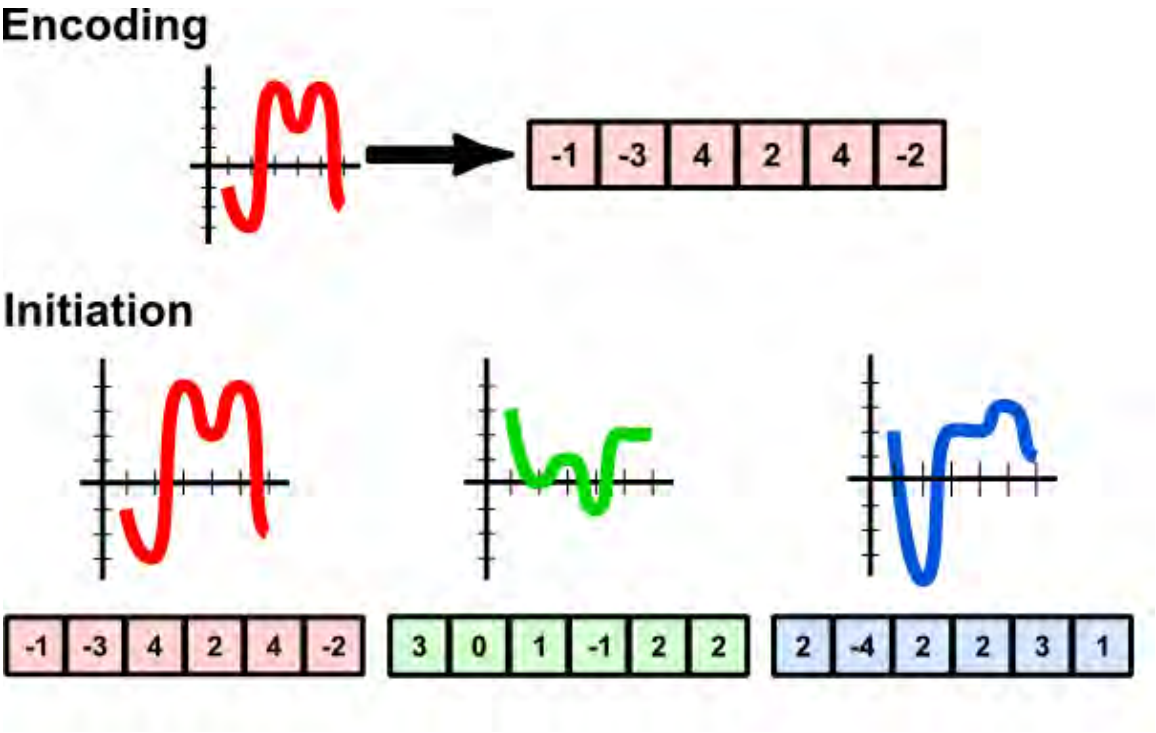


**Figure 4.4: Encoding signals as chromosomes. In this example, the amplitude of the signal at each time step becomes the value of genes. Once the method of encoding has been decided, the genetic algorithm is initiated with a generation, or set, of stochastic seeds.**

1. Choose a random initial generation, or a group of solutions.
2. Reproduce the next generation of solutions using recombination and mutation.
3. Select which of the chromosomes from the generation are fit enough to propagate based on the cost function.
4. Repeat steps 2 and 3 for a defined number of iterations

Traditionally, recombination selects a random point in the chromosome and switches everything on one side with the corresponding region on the second chromosome. The algorithm mimics mutation by adding a small perturbation to a randomly selected gene in the chromosome. Figure 4.5 illustrates how a few of the components of the genetic algorithm works.

Researchers can customize the algorithm in many different ways. Some customize the way chromosomes encode the solutions, while others may customize the order in which recombination, mutation, and selection occur. Forrest provides more information about the variations in genetic algorithms and their applications[126].

Because genetic algorithms operate on populations of chromosomes, they are less likely to fall prey to local optimal solutions. Even if certain solutions congregated around local optimal solutions, the recombination and mutation process shake up the population

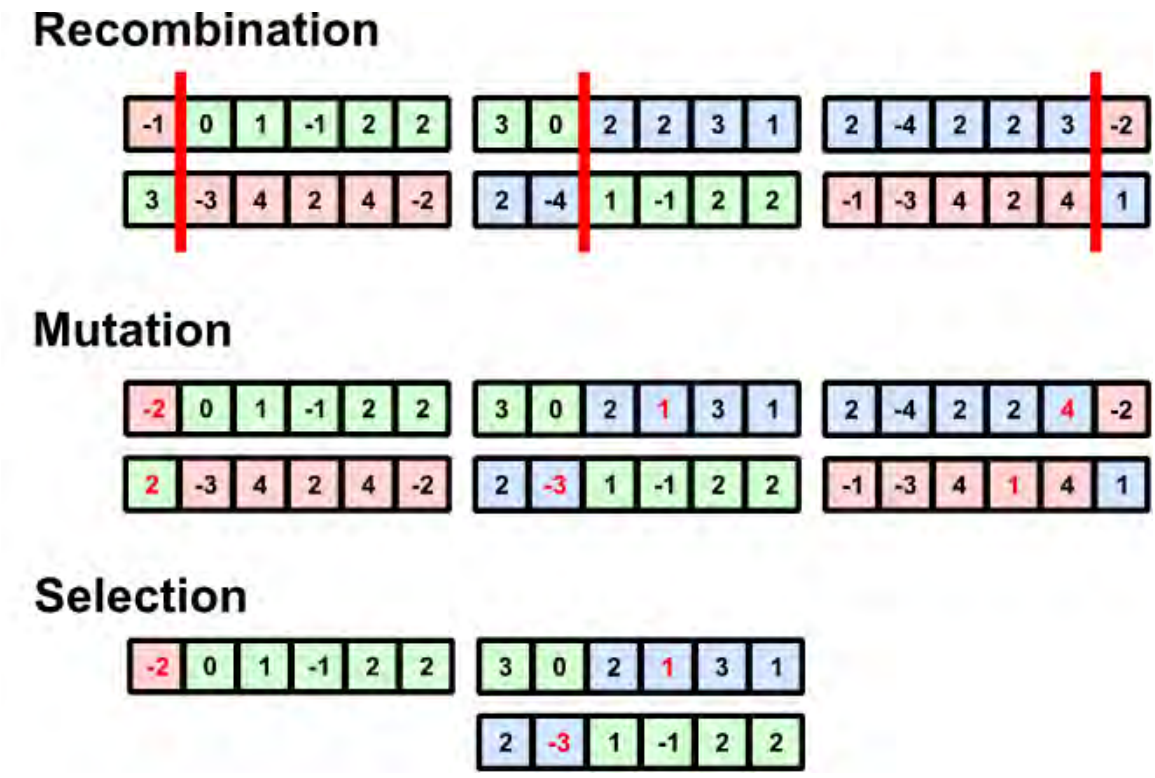


**Figure 4.5: Each iteration of the genetic algorithm goes through a set of recombination, mutation, and selection in order to form the next generation. Here we use the generation initialized from Figure 4.4 as an example of how recombination, mutation, and selection would work to generate the next set of solutions.**

sufficiently to cause wider exploration of the search space. Furthermore, because the population of solutions is in communication with each other, there is a greater chance of finding the optimal solution. Genetic algorithms have been in a wide range of applications including image processing[127–134], physiology[135–137], genetics[138,139], biochemistry[140–144], epidemiology[145–151], diagnostic analysis[152–159], radiology[160,161], and even hospital administration[162–165].

Wongsarnpigoon and Grill published a study using genetic algorithms to find optimal stimulus waveforms for a model of neural stimulation[166]. They found the optimal stimulus waveform necessary to trigger an action potential in the MRG model of a peripheral nervous system myelinated axon as described by McIntyre et al[167]. In order to accomplish this, they encoded a stimulus as a chromosome where each gene represented a single stimulus amplitude at a given time step ($dt = 2\ \mu s$) for a given pulse width. They ran the experiment with a range of pulse widths from 10 μs to 1 ms. Each generation consisted of 50 chromosomes. They defined the fitness function as the energy used in the waveform combined with a penalty if the waveform failed to elicit an action potential. In order to get convergence to an optimal solution, they ran the algorithm for a total of 10,000 generations. The result of this algorithm led to finding that truncated Gaussian waveforms were more optimal than other monophasic conventional waveforms that were tested including rectangular waveforms and exponentially decaying waveforms.

# Discussion

While we have examined a few of the more prominent stochastic search algorithms, there are still many more algorithms available including stochastic search variable selection[168–170], simultaneous perturbation stochastic approximation[171], and particle swarm optimization[46,75]. Spall's book provides a very helpful introduction to the theory behind stochastic search and optimization as well as in-depth analysis on the algorithms listed above[172].

All of the stochastic search algorithms encode the stimulus as a vector, where each parameter represents an independent dimension in the solution space. Depending on the resolution of the stimulus and the stimulus duration, the dimensionality of the solution space could become extremely large. For instance, if a 30-ms stimuli sampled every 0.1 ms, results in a 300-dimension solution space. This "curse of dimensionality" is one of the biggest problems with stochastic search algorithms.

The term "curse of dimensionality" was coined by Richard Bellman, who described how as the dimensionality of the solution increased, the solution space itself increases much more rapidly[173]. Assume, in our search for optimal stimulus waveforms, that we limit our search space to stimulus intensities of only integers between -10 to 10, meaning that for every time step, there are only 21 possible intensity values (-10, -9, -8, … 8, 9, and 10). If we wanted to find the optimal 0.1-ms stimulus waveform sampled every 0.1-ms (e.g. there was only one time interval), we would have to search 21 possible solutions. If we were searching for optimal 0.2-ms stimulus waveforms, we would have to search 21 *

21, or 441 possible solutions. Studying a 1-ms stimulus waveform gives us $1.7 * 10^{13}$ possible solutions. At these levels, getting a meaningful sample of the neighboring regions becomes very difficult. What is staggering is that this is only after we have limited our search space to only integers. If we allowed the search space to include values up to three or four decimal points, it would be impossible to produce any meaningful sampling.

In order to use stochastic search algorithms efficiently, we will need to develop a way to simplify the solution space. Spike-triggered averaging avoids this conundrum by simply averaging all the data without any structured search patterns. We explore this further in the next chapter. In Chapter 6, we develop and validate a novel approach to reduce adaptively the dimensionality of the problem as the algorithm iterates forward to reduce the impact of this curse.

# CHAPTER V

# Spike-Triggered Averaging: Limitations as an Optimization Tool

## Introduction

As we stated previously, one of the simplest stochastic algorithms is spike-triggered averaging (STA), a technique used in experimental neuroscience to gain insight into neuronal responses to stimulus features [89,90,174–177]. STA, also known as reverse correlation or white-noise analysis, computes the average input stimulus preceding each spike generated from broadband stochastic stimulation. This process reveals key stimulus features that govern how neurons inherently encode sensory information.

While there are other stochastic search algorithms, as detailed in the previous chapter, many of them fall prey to the "curse of dimensionality." Spike triggered averaging is one of the few techniques unaffected by the curse because of its simplicity. As we have discussed already, some studies have examined the similarities between STA waveforms and optimal stimuli calculated using analytical techniques [33,39,63].

---

Portions of this chapter are reprinted with permission from: Chang J, Paydarfar D. Optimal stimulus waveforms for eliciting a spike: How close is the spike-triggered average? 7th International IEEE EMBS Conference on Neural Engineering, April 22-24, 2014, The Corum, Montpellier, France.

In this chapter, we analyze how noise-based STA waveforms relates to energy-optimal waveforms for eliciting a spike in the Hodgkin-Huxley (HH) model [10]. In order to explore a range of neuronal threshold levels, we studied the HH model under two conditions: the standard condition in which there is no persistent current, and a near-bistable condition, in which a large persistent current is added causing the membrane to be at a heightened state of excitability near a sub-critical Hopf bifurcation [71].

# Methods

## Hodgkin-Huxley model

The Hodgkin-Huxley model is a four-dimensional system that captures the ionic mechanisms underlying the generation of an action potential. By adjusting the persistent current, we can raise the excitability of the membrane such that much less stimulus energy is required to fire an action potential. Using different levels of persistent current, we are able to study differences in stimulus optimality related to a range of membrane thresholds, for example between high threshold interneurons as compared to low thresholds seen in some cortical neurons. Also, raising the persistent current affects the resonance properties of the membrane such that the neuron is more responsive to oscillatory stimuli [71].

## Determining optimality using a stochastically-seeded gradient algorithm

In order to provide a gold standard for optimality, we rely on analytic methods. In this chapter, we calculate energy-optimal waveforms by minimizing the $L^2$-norm of the stimulus. Traditionally, optimization of signals has been conducted using calculus of variations [32], in which an optimal functional is derived and solved as a boundary-value

problem. This approach has been used to calculate energy-optimal stimuli for the HH model [33]. An alternative approach has solved this problem directly using a stochastically-seeded gradient algorithm [178]. We use this approach for the HH model, to determine the optimal stimulus necessary to trigger a single action potential in both high and low persistent current conditions.

The stochastically-seeded gradient algorithm generates a series of random stimuli, and using a gradient-based approach, calculates how the system will respond to minute changes in the stimulus. The algorithm iteratively uses these small gradients, or slopes, to converge towards the optimal solution. Stochastic seeding enables the algorithm to explore a large solution space. In the case of the standard HH model (with no persistent current), we previously showed convergence to the global optimal waveform [178].

## Finding the spike triggered average using different noise levels

In order to mimic experimental spike-triggered averaging, we simulated white noise current stimulation, by choosing randomly the stimulus intensity from a Gaussian distribution for each 0.1 ms sample data point. We tuned the amplitude of the noise such that the action potentials would fire at a rate of <1 Hz. We fed white noise continuously until we generated 3,000 action potentials. Each time an action potential occurred, the stimulus signal preceding the spike (up to 100 ms) was stored. The average of all 3,000 stimulus snippets (aligned to the peak of the action potential) was scaled to the minimum amplitude for triggering an action potential as previously described [33,39].

We ran the above protocol for the HH model under standard conditions (no persistent current) and the near-bistable condition (a persistent current of 6μA/cm2). The duration of the stimulus prior to each spike was 30 ms for the standard condition and 100 ms for the near-bistable condition. In order to capture the full oscillatory stimulus pattern in the neuron near its sub-critical Hopf bifurcation, we used a longer averaging period.

All of the simulations and data analysis was done in MATLAB (The Mathworks, Inc, Natick, MA) on a Dell Precision T7500 (Intel Xeon Core E5520 @ 2.27 GHz, 2.26 GHz processors and 24 GB RAM). The code for the gradient algorithm can be obtained directly from the authors or from Physionet (www.physionet.org).

# Results

## Gradient-based calculation of stimulus optimality for eliciting an action potential

As seen in Figure 5.1, the standard HH model fires an action potential most efficiently when the stimulus includes a single hyperpolarization followed by depolarization phase. On the other hand, when the neuron is more highly excitable in the near-bistable condition, the stimulus shows more prolonged sub-threshold oscillations with multiple cycles between hyperpolarization and depolarization building up to the action potential.

## Comparing STA waveforms to optimal waveforms

Figure 5.2 shows the STA waveform and the optimal waveform calculated from the gradient algorithms. The two waveforms are remarkably similar, as reported previously

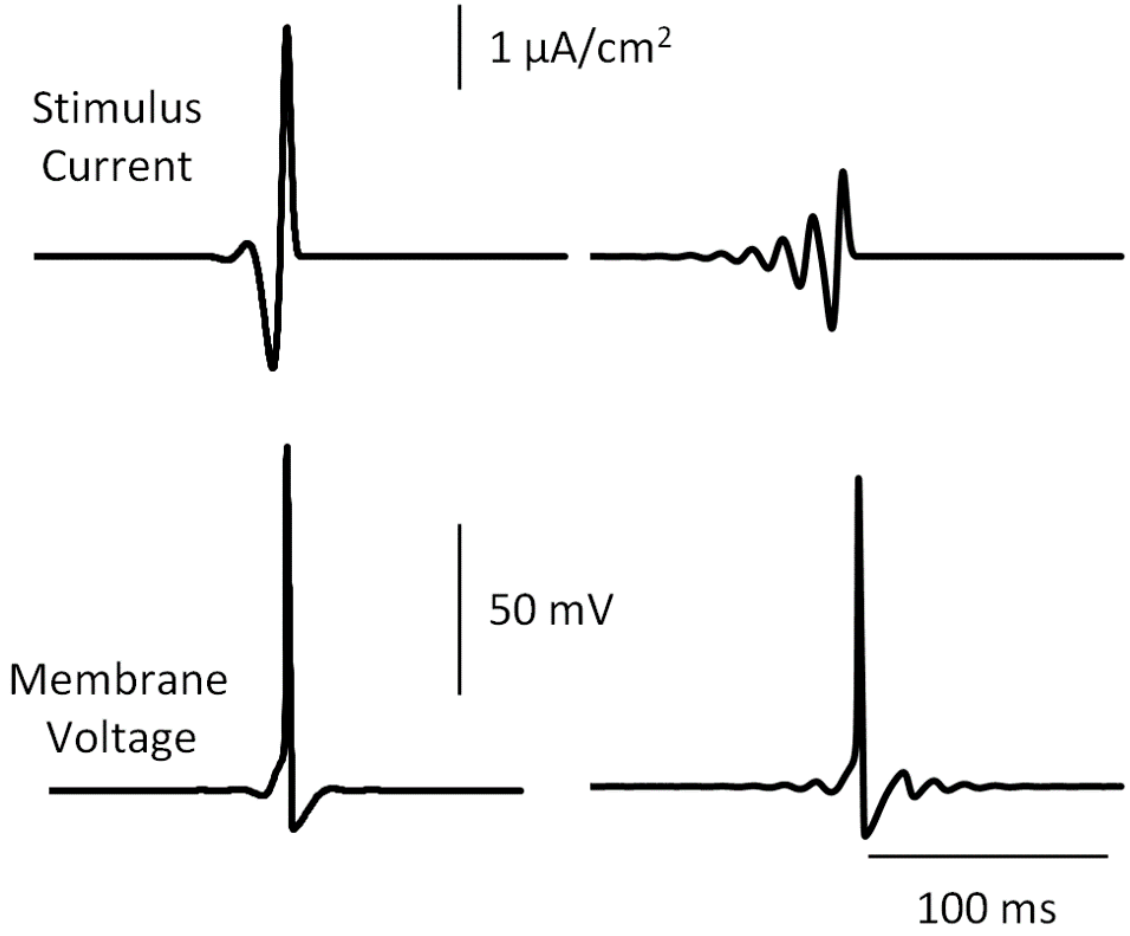


**Figure 5.1: Energy-optimal stimulus waveform for the near-bistable condition (right) displays more resonance than the optimal waveform for the standard condition for the Hodgkin-Huxley model (left)**

[7], [8], [11]. Two major differences stand out, however. First, the spike-triggered average appears to be right-shifted compared to the gradient algorithm in both the standard as well as the near-bistable conditions. Second, in the near-bistable condition, the spike-triggered average exhibits a phase reversal in the stimulus about 20-ms prior to the action potential.

We can explain the STA right-shift by considering that the waveform is an average of a library of successful stimuli without restriction on the effective portion of the stimulus that causes the spike. The algorithm would average short brief large fluctuations that induce a spike with stimuli that have longer duration with smaller amplitude fluctuations. In Figure 5.3, we show using the gradient algorithm the optimal waveforms for progressively longer stimulus durations, all lined up to the peak of the spike. Note that the shorter stimuli have higher amplitude and are right shifted, compared to the longer duration stimuli. We previously showed that the longer stimuli are more energetically optimal [178]. Because of this phenomenon, we can see that if longer and shorter stimuli are included in the library of snippets that induce a spike, averaging the briefer waveforms with the more prolonged waveforms would result in a right-shift compared to the global optimal calculated from the gradient algorithm. As such, when averaging the optima across different time scales, we get this right-shift.

Regarding the deviation of the spike-triggered averaging from the gradient algorithm in the near-bistable condition, we examined the system's response to all the individual

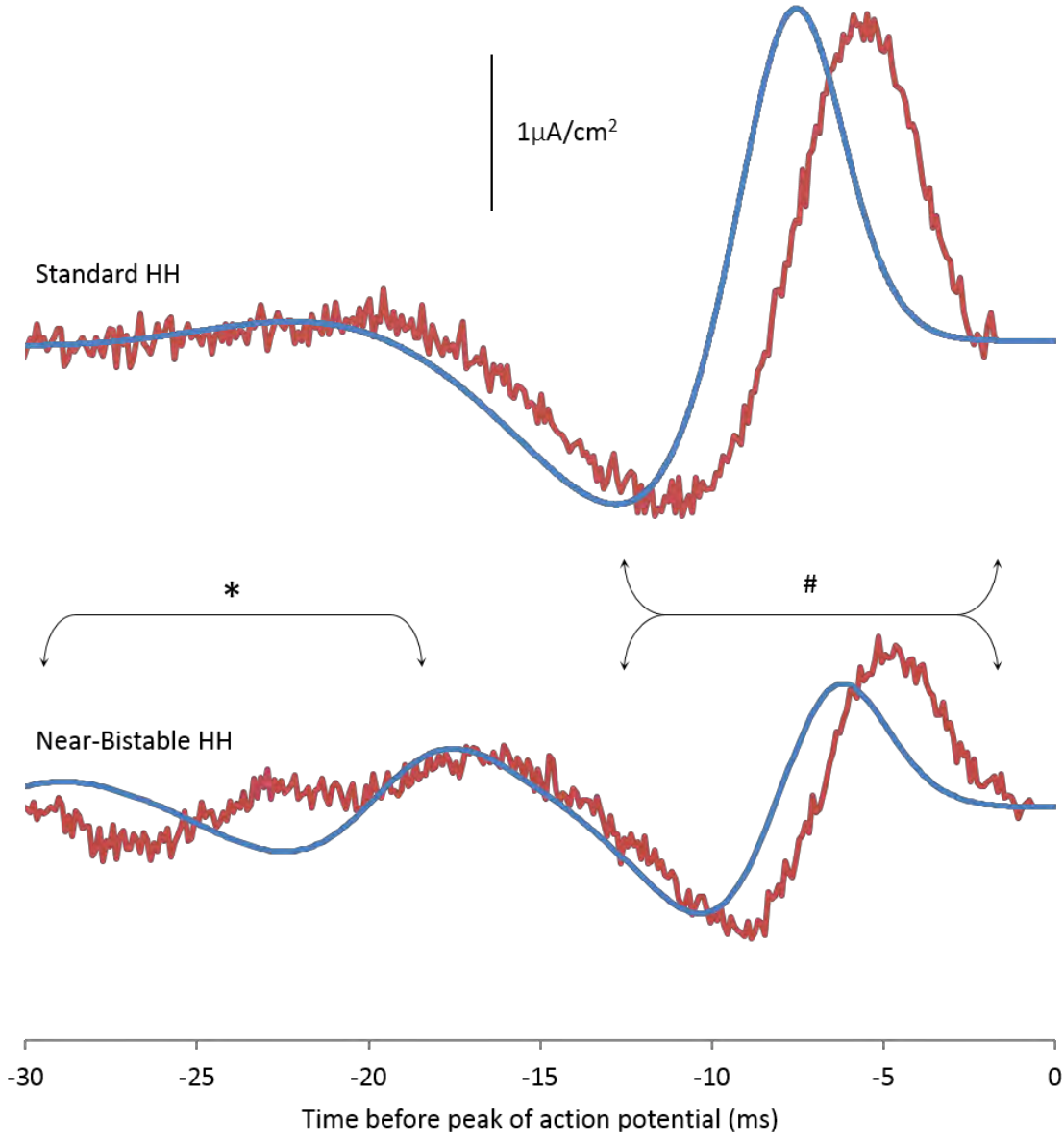


**Figure 5.2: Comparison of spike triggered averaging results (orange) and the gradient algorithm (blue). We note two main discordances: a phase reversal (*) and a right-shift (#).**

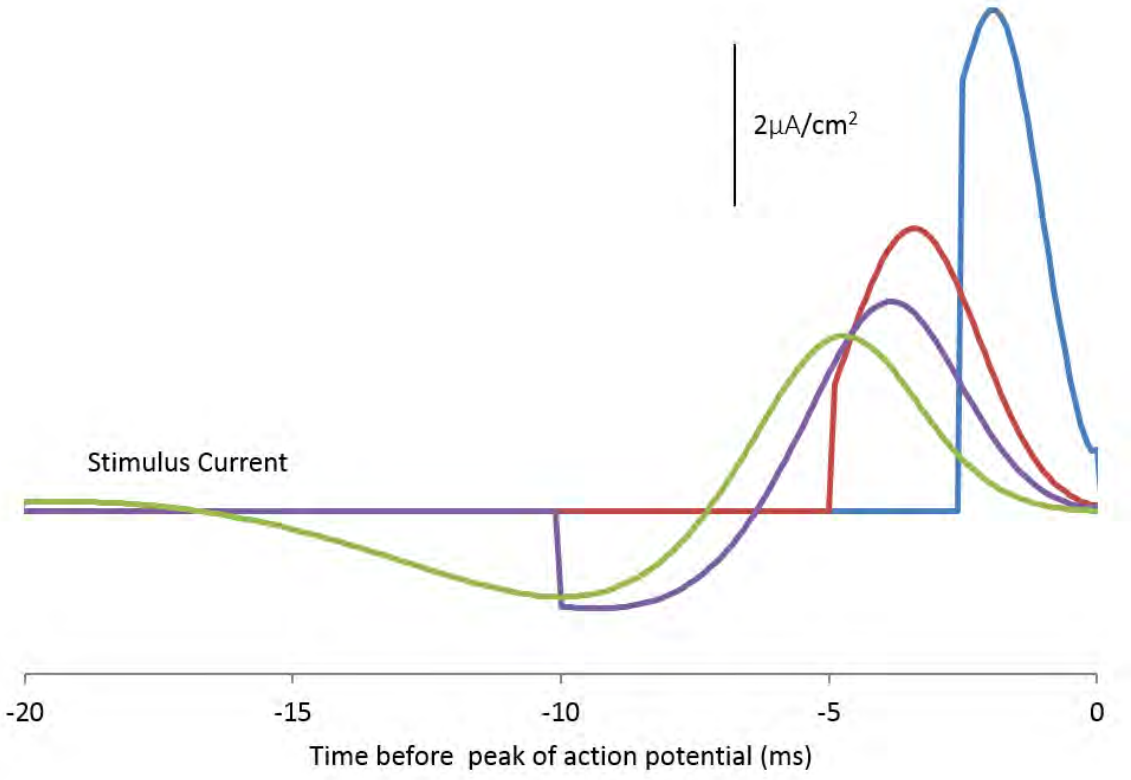


**Figure 5.3: Optimal waveform shapes given different stimulus durations as determined by the gradient algorithm.**

snippets in our library. We noticed that some of them caused single spikes, while others caused two or more spikes in rapid succession. This led us to speculate that the stimulus necessary to cause repetitive spikes might be qualitatively different from that which causes a single spike. We re-evaluated the near-bistable condition using the gradient algorithm, and we found two locally optimal solutions shown in Figure 5.4.

Informed by this result, we re-analyzed the noise induced spike trains in the near-bistable condition and calculated the STA on the snippets that contained just one action potential. The result was much closer to the energy optimal waveform computed using the gradient algorithm, as seen in Figure 5.5. The small rightward shift for the oscillations near the spike are not affected by this re-analysis, presumably because of the mechanism related to short duration stimuli (Figure 5.3). This confirms that by averaging different local optimal solutions, or different modes, we get discordance from the true optimal.

We did take into consideration the fact that we are using extremely high frequency noise stimuli, but when we calculated the spike-triggered average with white-noise filtered through a 1-kHz low-pass filter (which is more biologically realistic[174]), we found similar results.

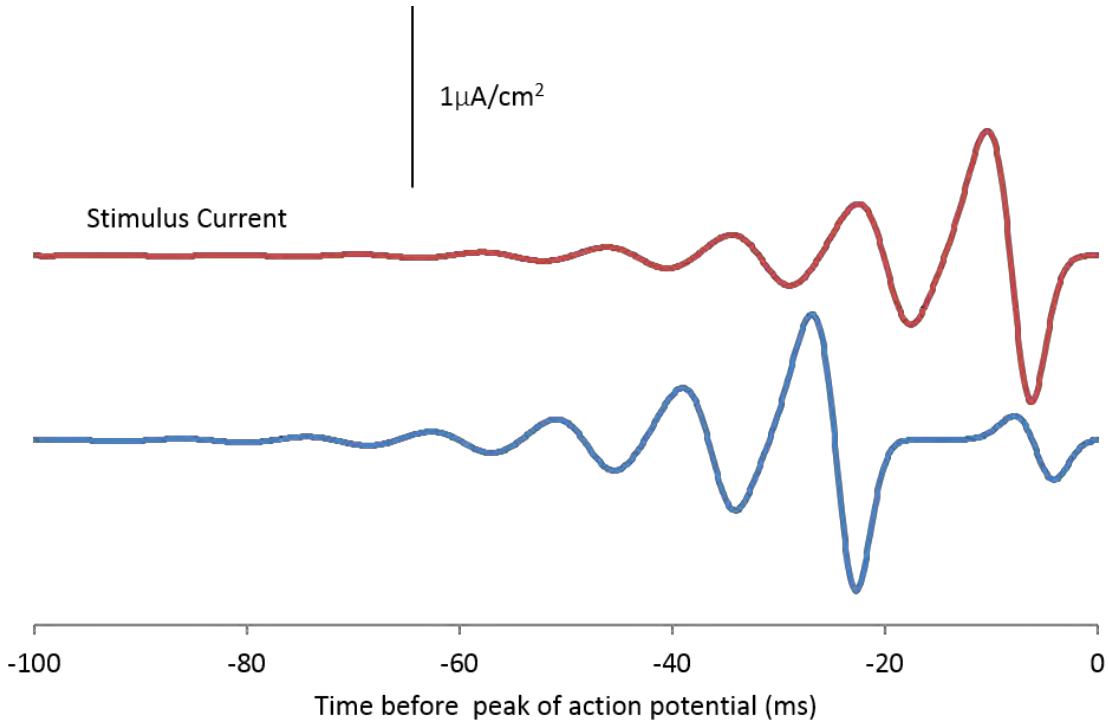


**Figure 5.4: The near-bistable solution has at least two locally optimal solutions: one that fires a single action potential (red), another which fires two action potentials in rapid succession (blue).**

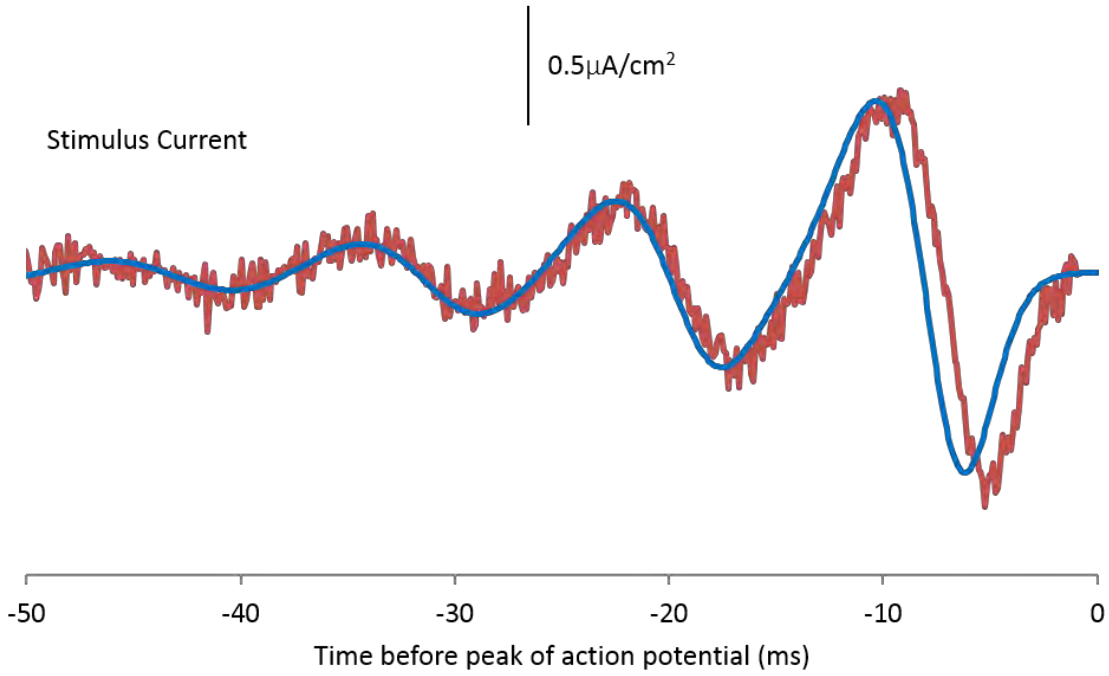


**Figure 5.5: When averaging stimuli causing only one spike, the STA (red) phase reversal disappears, and STA aligns more closely with the true optimal (blue).**

# Discussion

Our results suggest that while STA and energy optimal waveforms are remarkably similar, a number of mechanisms contribute to their discordance.

First, we explain how a rightward shift in the STA occurs when averaging stimuli having a range of time scales contributing to spike generation. The rightward shift occurs because the briefer stimuli cause a more rapid depolarization and induction of a spike compared to the longer duration stimuli. Because the space of all possible solutions for the Hodgkin-Huxley model includes solutions that are larger, which fire an action potential sooner, the spike-triggered average waveform is right-shifted from the optimal solution as determined by the gradient algorithm. We find that using larger noise amplitudes can potentially cause a larger deviation away from energetically optimal solutions. It is interesting to note that when choosing a smaller amplitude noise profile, the amplitude of the spike-triggered average also decreased, however the right-shift remained (Figure 5.3). In theory, if we choose an amplitude of the noise such that it is close to the amplitude of the optimal solution, we might be able to reduce further the right-shift in the STA because we would not be including into the average the large amplitudes that cause the action potential to fire faster. In practice, however, when we titrated the noise level further toward the amplitude of the optimal solution, spikes disappeared and we were not able to achieve this result.

Second, multiplicity of optima can skew the STA waveform. When there are multiple possible mechanisms to trigger an event, the multiplicity can cause the spike-triggered average to move away from the true optimal solution (Figure 5.5).

To gain a better appreciation of the relationship between spike triggered averaging and optimality, we imagine a stimulus with only two points. This illustration can be extrapolated to higher dimensions, but for visual purposes we will use a two point stimulus. The space created in Figure 5.6 is the set of all possible stimuli. We have marked off two regions within this space. Region A represents all the stimuli that successfully cause an event, while Region B represents all the stimuli that our white noise generator produces given a specific amplitude.

In this depiction, the spike-triggered average is the centroid of Region C, which is the intersection between Region A and Region B. In calculating the energy of the stimuli, we are using a simple $L^2$-norm metric, which is equivalent to the distance between the stimuli and the origin. Thus, the farther a point is from the origin, the larger is its energy. Using this representation, we can begin to gain some insight into the relationship between spike-triggered averaging and energy optimality.

Figure 5.6 (top) is a representation of the standard Hodgkin-Huxley neuron. As seen in the figure, the spike-triggered average alone is not close to the optimal, but when scaled to just barely causing an action potential, it is quite close to the true optimal solution.

When multiplicity of solutions exists (e.g., as seen in the near-bistable condition), one can imagine a system similar to Figure 5.6 (bottom), where there exist two groups of solutions within the stimulus space, the scaled spike-triggered average could look very different when compared to the true optimal solution.

In our experiments, we recognized the existence of two sets of solutions by observing the train of action potentials in response to some of the stimuli. As such, we filtered the data to produce only the stimuli that exist in one of these sets, and then performed spike-triggered averaging, which led us to a result much closer to the true optimal solution. As we stated earlier, we could theoretically also have decreased the noise amplitude, shrinking Region B, to eliminate the possibility of the second solution set intersecting with the noise profile we are using. This is a potential solution, but only one that works if a large enough margin separates the two solution sets. Again, it is difficult to build a library of successful stimuli if the noise profile is barely above the amplitudes of the true optimal stimuli. As such, more often than not, the noise will be larger than what one can do to separate out multiple sets of successful stimuli.

As seen, spike-triggered averaging can approximate the energetically efficient optimal solution, there are also times when it will fail to do so. We posit that whether spike-triggered averaging is successful or not depends largely on the shape of the solution space. Quantifying the solution space could enable a more accurate and consistent method to approximate the optimal signal using the spike-triggered library.

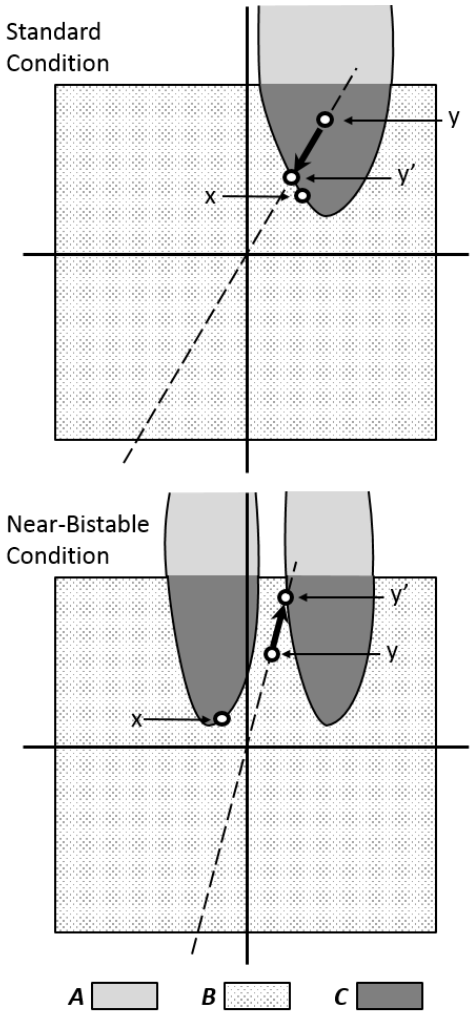


**Figure 5.6: Whether the spike-triggered average is close to the energy-optimal is related to the shape of the intersection (Region C) between the set of all possible stochastic signal (Region B) and the set of all possible stimuli that successfully trigger an event (Region A). Sometimes when the center of mass (y) is scaled (y'), it may move closer to the optimal stimulus (x) as seen in the top panel, while other times it may move further away as seen in the bottom panel.**

# CHAPTER VI

# Extrema Distortion Algorithm: Harnessing Noise to Optimally Flip a Switch

## Introduction

In Chapter 4, we examined how stochastic search algorithms find optimal solutions without knowledge of the model. We noted that one of the major problems with stochastic search algorithms as it relates to time-varying signals is the "curse of dimensionality," that as we increase the resolution or duration of the optimal solution, the search space grows at an extremely rapid rate. While spike-triggered averaging avoids this curse, we found in Chapter 5 that its relationship with optimality is more dependent on the solution space shape. Using STA, we may get close to optimality, but there would be no guarantee without knowing what the solution space looked like.

As such, we needed to develop another method to use stochastic search, while minimizing the effect of the curse. In this chapter, we address the curse of dimensionality by representing signals by their extrema points. Instead of searching neighboring solutions using every single point of the stimulus, we use the extrema as key

features points. By distorting around these extrema, we can search for more optimal solutions using fewer dimensions.

We test this algorithm on three unique models: a Hodgkin-Huxley model for triggering a single action potential, a bistable FitzHugh-Nagumo model for suppressing repetitive firing, and a genetic toggle switch model for switching states. In each of these models, we applied analytical techniques to compare the accuracy of the results from our algorithm. Finally, we discuss future directions of how this algorithm can continue to be developed.

# Methods

## Formalization of the problem

Let us define the mechanics of our physiological system as $\dot{x} = f[x(t), u(t), t]$, where $x(t)$ defines the $m$-dimensional states of the system and $u(t)$ defines the $n$-dimensional stimulus. As we have seen in previous chapters, these definitions are generally the starting point from which traditional methods like calculus of variations or gradient-based algorithms solve for optimality. Because we are developing a model-independent, stochastic algorithm, we focus our attention on the stimulus. Most implementations of stochastic algorithms treat each point in the stimulus as its own individual dimension. Thus, we can calculate the number of dimensions a stimulus requires by multiplying the duration of the stimulus by the resolution of the stimulus generator. Thus, if we have a 25-ms 1-dimensional stimulus generated at a 0.1-ms resolution, the solution space has a dimension of 250.

The fundamental concept underlying stochastic algorithms is that adaptive distortions of randomly generated initial estimates of solutions, which we designate as $u_0$, evolve towards an optimal solution, which we designate as $u_{optimal}$, based on some predefined criteria. This thesis uses the $L^2$-norm, or "energy" of the stimulus as the basis of our definition of optimality. We constrain our solution space to only solutions that successfully trigger a state change. Mathematically, our goal is to minimize $\int u^2 dt$ given that the solution exists in the solution space.

We modeled our algorithm initially off the stochastic hill-climbing approach in which a noise component, $\delta\mu$, is introduced to the original stimulus, allowing the algorithm to search locally for more optimal solutions. Once the algorithm finds a more optimal solution, that new solution, labeled $\mu_1$, becomes the new starting point of yet another local search. Thus, this algorithm operates on the premise that

$$\lim_{k\to\infty} \mu_k = \mu_{opt}$$

One of the observations we made early in our study was that with large dimensional solution spaces, the closer we were to optimality, the more dependent the dimensions became on each other. Thus, as we found solutions closer and closer to optimality, it was more important that both $\mu[t]$ and $\mu[t+1]$ moved in harmony. When these dimensional spaces are large, the probability of many different parameters moving in synchronization with each other became rarer, making it more difficult for the stochastic search to find more optimal solutions successfully.

In this study, we propose that instead of focusing on manipulating every single data point in the stimulus, we focus on manipulating the extrema of each stimulus. This method has the benefit of allowing many degrees of freedom early in the process, but as the algorithm progresses, and extraneous extrema are removed, the degrees of freedom decreases, allowing for a higher probability of finding more optimal solutions.

## Stochastic Extrema Distortion Algorithm

As stated, we propose a stochastic Extrema Distortion Algorithm based on the idea of allowing each extrema to move randomly, using a linear transformation of the original signal with the new extrema as endpoints to fill in the gaps in between extrema. The steps of the algorithm are as follows:

1. Select a randomly generated starting seed, $\mu_0$, which causes a state transition.
2. Find the extrema of $\mu_0$ and measure the time interval between them as seen in Figure 6.1A.
3. Multiply each of the time intervals by a randomly generated number with a Gaussian distribution.
4. Rescale the time of the stimulus to match the duration of the original stimulus.
5. Add to each peak a randomly generated number with a Gaussian distribution.
6. Linearly transform all the data points between the extrema with the new extrema endpoints as seen in Figure 6.1B.
7. Scale the new stimulus to the same $L^2$-norm as the most optimal stimulus seen so far

8. Scale the resulting stimulus down until it just barely achieves the desired outcome.
9. Repeat steps 2-8 ten times with the same starting seed.
10. Calculate the $L^2$-norm of each of the ten results and choose the best one. If none of the ten results is better than the original seed, choose the original seed.
11. Repeat steps 2-10 using the chosen stimulus as the starting seed for a predetermined number of iterations.

In order to capture our starting seeds, we fed white noise stimuli into the system and captured only the snippets that successfully caused a state transition. These snippets became the starting seeds for our algorithm.

In order to validate our algorithm, we used the gradient algorithm as detailed in chapter 3 to set up a benchmark by which we could compare our results.

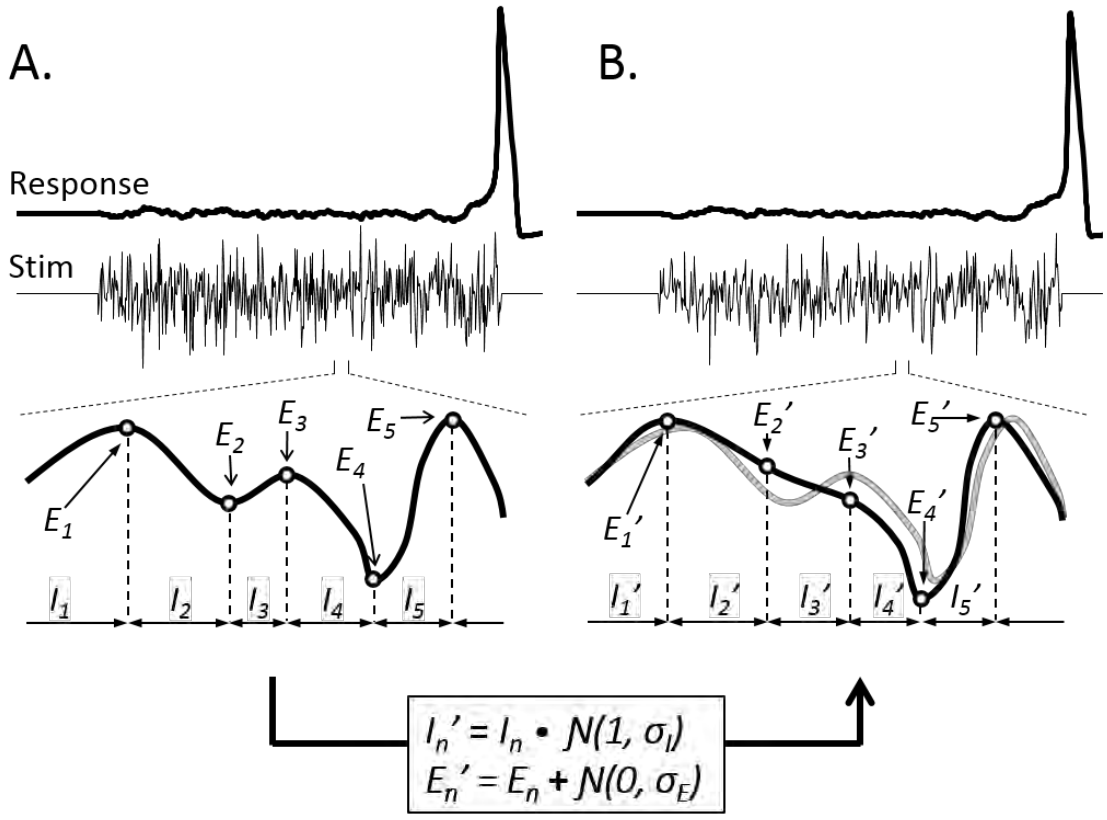


**Figure 6.1: Diagram of a single iteration of the Extrema Distortion Algorithm. Each panel shows the system response (top), the stimulus (middle) and a cartoon representation of the algorithm at work. The extrema are determined, and the intervals as well as the amplitudes are then distorted using multiplicative and addictive noise respectively to generate a new signal.**

## Induction of an action potential: The Hodgkin Huxley model

The Hodgkin-Huxley[10] model is one of the best known computational models in biology . This model is a four dimensional system that captures the ionic mechanisms underlying the firing of an action potential in a single neuron. We apply our distortion algorithm to find the most energetically efficient, as defined by $L^2$-norm, stimulus that will cause a single neuron to fire an action potential. Chapter 3 describes the mathematical model for the Hodgkin-Huxley neuron. As a reminder, our goal for this model was to find the optimal 50-ms stimulus that would successfully cause the neuron to fire an action potential within 100-ms of the start of the stimulus.

For the Hodgkin-Huxley model, we multiplied the time intervals by a randomly generated number using a Gaussian distribution with mean of 1 and standard deviation of 0.25. We added to each peak a randomly generated number (Gaussian distribution with mean of 0 and standard deviation of 10% of the amplitude of the peak). We allowed the algorithm to run for 3,000 iterations.

Because we were looking for discrete state changes, a single action potential, we gave continuous white noise with an amplitude of $5\mu A/cm^2$ to the Hodgkin-Huxley model. Whenever an action potential occurred, we cut out the 50-ms snippet of stimulus proceeding the action potential. These snippets became the starting seeds of the Extrema Distortion Algorithm. We ran 10 seconds of white noise using a uniform distribution random number with a resolution of 0.1 ms, capturing 35 50-ms snippets.

## Switching off oscillations: The Fitzhugh-Nagumo model

The FitzHugh-Nagumo[54,55] model of neuronal excitation is a two-dimensional model that has been used to abstractly describe a simple excitable neuron, as well as cardiac and circadian rhythms has been detailed in Chapter 3. As a reminder, the model is unitless, but regarding neuronal excitability, $x_1$ is analogous to Hodgkin-Huxley's $V$ and $m$, while $x_2$ is analogous to Hodgkin-Huxley's $h$ and $n$ states. The variable $u$ represents the current stimulation, which can be in the form of an endogenous persistent current or an exogenous input stimulus. Using the parameters found by Paydarfar and Buerkel[57], we defined $a = 0.7$, $b = 0.8$, $c = 3.0$ and $r = 0.342$. In this particular configuration, the system gravitates, when there is no stimulation, towards one of two states: quiescence (stable fixed point) or repetitive firing (stable oscillatory limit cycle). The minimum value of $x_1$ is the equivalent of the peak of an action potential in the FitzHugh-Nagumo model.

For our purposes, we are looking to minimize the energy needed to suppress repetitive firing, for instance in the clinical case of suppressing an epileptic neuron. We have previously determine the optimal shape and phase angle for the start of an 8-ms stimulus waveform. Thus, using the same parameters, we chose to apply the stimulus under the same starting conditions for the system ($x_1 = 0.9302$, $x_2 = -0.3760$. Our goal for this model was to give an 8-ms stimulus, such that system suppresses repetitive firing stably for 92-ms after the stimulus is given.

We defined suppression of repetitive firing as the lack of peaks less than -1.5 units tall, slightly smaller than the height of a normal action potential. For the FitzHugh-Nagumo model, we multiplied the time intervals by a randomly generated number using a Gaussian distribution with mean of 1 and standard deviation of 0.25. We added to each peak a randomly generated number with a Gaussian distribution with mean of 0 and standard deviation of 10% of the amplitude of the peak. We allowed the algorithm to run for 10,000 iterations.

The bistable system is slightly different from a monostable system in that it is a little harder to define a clear distinguishing feature that we could use to dictate what portion of the stimulus was contributing to the state change. As such, we gave discrete 8-ms snippets to the FitzHugh-Nagumo model and used successful snippets as the starting seeds for the Extrema Distortion Algorithm. We used white noise generated using a uniform distribution random number generator with an amplitude of 2 units and a resolution of 0.1 ms, capturing 49 snippets over the course of 1,000 total snippets.

As a proof of principle, we ran an instance of the Extrema Distortion Algorithm with 100-ms long stimulus. We wonder if the algorithm could guide us to knowing what the proper phase response of the stimulation should be. We were able to capture 16 100-ms white noise snippets that successfully suppressed repetitive firing.

## Flipping a genetic toggle switch

One of the more recent innovations in biotechnology has been the construction of genetic toggle switches[179]. These switches play an important role in synthetic gene-regulatory networks by allowing transient stimuli, chemical, or thermal, to trigger state changes in bistable networks. This switch is composed of two repressors and two constitutive promoters. The opposing promoter, as seen in Figure 6.2, transcribes each repressor.

This model consists of two state variables, $u$ and $v$, representing the two repressor concentrations. The equations governing this system are:

$$\frac{du}{dt} = \frac{\alpha_1}{1 + v^{\beta}} - u$$

$$\frac{dv}{dt} = \frac{\alpha_2}{1 + u^{\gamma}} - v$$

where $\alpha_1$ and $\alpha_2$ are the rate of synthesis of repressor 1 and repressor 2 respectively, $\beta$ is the cooperativity of repression of promoter 2 and $\gamma$ is the cooperativity of repression of promoter 1. In the study, they created a single plasmid that used the Lac repressor (lacI) in conjunction with the Ptrc-2 promoter as repressor 1 and a temperature-sensitive λ repressor (cIts), in conjunction with a $P_L$s1con promoter as repressor 2. Because of this particular setup, the parameters of the model are $\alpha_1 = 156.25$, $\alpha_2 = 15.6$, $\beta = 2.5$, and $\gamma = 1$.

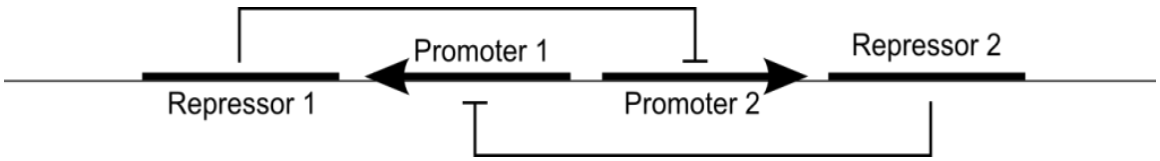


**Figure 6.2: Diagram of the genetic toggle switch. Figure adapted from Gardner TS, Cantor CR, Collins JJ (2000) Construction of a genetic toggle switch in Escherichia coli. Nature 403: 339-342.**

In this toggle switch, only one of the concentrations is elevated at any given time. They used a pulse of isopropyl-β-D-thiogalactopyranoside (IPTG) to switch the system so that the lacI concentration was elevated, and then used a thermal pulse to switch the system back. The model was expanded to include IPTG as the exogenous stimulus to the system.

$$\frac{dv}{dt} = \frac{\alpha_2}{1 + \left( \frac{u}{\left(1 + \frac{[IPTG]}{K}\right)^{\eta}} \right)^{\gamma}} - v$$

where $K$ is the dissociation constant of IPTG from LacR and $\eta$ is the cooperativity of IPTG binding. $K$ is set at 2.9618 x $10^{-5}$, and $\eta = 2.0015$. Using the model, we can calculate the stable points by setting $\frac{du}{dt}$ and $\frac{dv}{dt}$ to zero. We learn that there are two stable fixed points: (0.3319, 11.7080) and (155.5143, 0.0995). An unstable fixed point was calculated at (1.3144, 6.7342).

From the literature, we have seen that a 20-minute pulse of IPTG is used to constitutively turn on lacI protein production. Our goal was to find the optimal 20-minute pulse of IPTG that would turn off the lacI protein production at the end of 20 minutes. One of the twists regarding the genetic toggle switch model is that the stimuli need to be positive-only because the stimulus represents the concentration of IPTG. A negative concentration is impossible. Thus, when finding our starting snippets, we tested 20-minute discrete non-negative intensity snippets. We generated white noise using a

uniform distribution random number generate with an amplitude of 0.0002 and a resolution of 0.1 minutes, capturing 29 snippets out of 50 snippets. In the distortion phase, instead of adding a randomly generated number to the peak, we multiplied the peak by a randomly generated number with a Gaussian distribution with mean of 1 and standard deviation of 0.1. This way, the values would stay positive. For the genetic toggle switch model, we multiplied the time intervals by a randomly generated number with a Gaussian distribution with mean of 1 and standard deviation of 0.25. The algorithm iterations 2,000 times.

# Results

We applied the Extrema Distortion Algorithm to three unique models to display the adaptability of this algorithm to a range of different systems. We have chosen two of the classic computational models used in neuroscience: the Hodgkin-Huxley model for triggering a single action potential, and the FitzHugh-Nagumo model for suppressing repetitive firing. We have also chosen to apply this algorithm to a relatively new model in synthetic biology, the genetic toggle switch for causing a switch from one state to the other.

## Induction of an action potential: The Hodgkin Huxley model

We ran 35 instances of the Extrema Distortion Algorithm, each with a unique starting seed. Figure 6.3 illustrates the path the algorithm takes as it begins with a stochastically

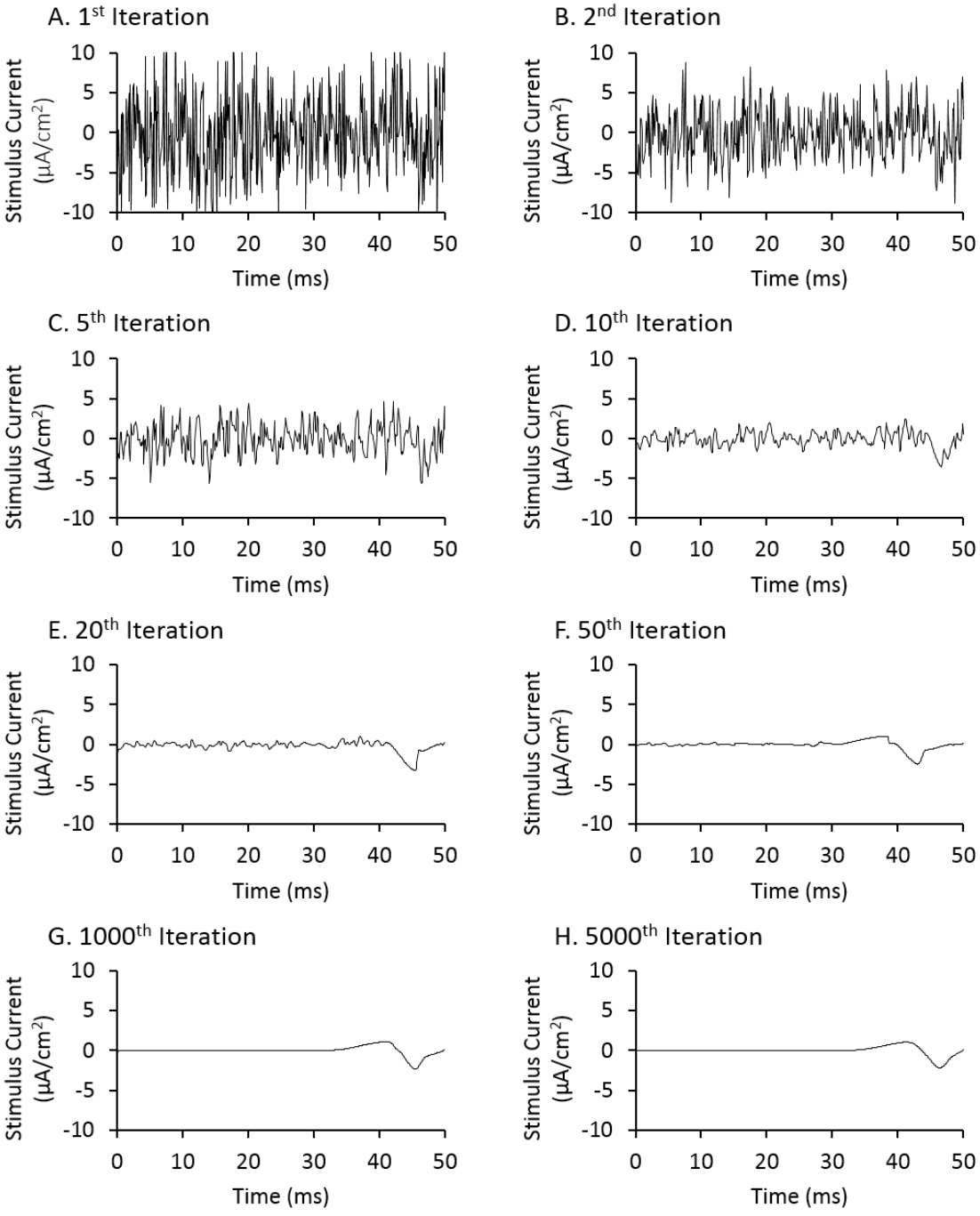


**Figure 6.3: The Extrema Distortion Algorithm shapes stochastically generated stimuli towards an optimal waveform. Here we show the progression of solutions from the original white noise stimulus in the 1st iteration to the 2nd, 5th, 10th, 20th, 50th, 1000th and 5000th iteration. By convention, positive current is depolarizing.**

generated initial stimulus, and it slowly evolves towards the optimal stimuli. It is interesting to note that the algorithm makes substantial improvements even within the first 10 iterations, when fundamental shapes become visible. By 50 iterations, most of the noise disappeared, and we saw convergence to a solution at 5,000 iterations.

Because this is an Extrema Distortion Algorithm, we can monitor the extrema time and amplitude over the course of the algorithm as well as seen in Figure 6.4. Here we can see that again, the algorithm begins with stochastically generated stimuli where the extrema are uniformly scattered out both in time and amplitude. Over the course of 100 iterations, we can see the algorithm pruning away extraneous extrema, converging the remaining extrema towards specific locations. There is a brief moment around 900 iterations where the algorithm does not find any improvement for a short period, but then it is able to find an improvement, which leads to more optimal solutions.

The resulting 35 stimuli had an average $L^2$-norm of 15.550 and a standard deviation of 0.149. Figure 6.5 shows the progression of improvement in stimulus energy over the 5,000 iterations, as well as the resulting stimuli for all 35 snippets. We lined up the resulting 35 stimuli based on where the action potential occurred. As we can see, the algorithm did a good job at converging the various stochastically generated noise samples towards the optimal solutions.

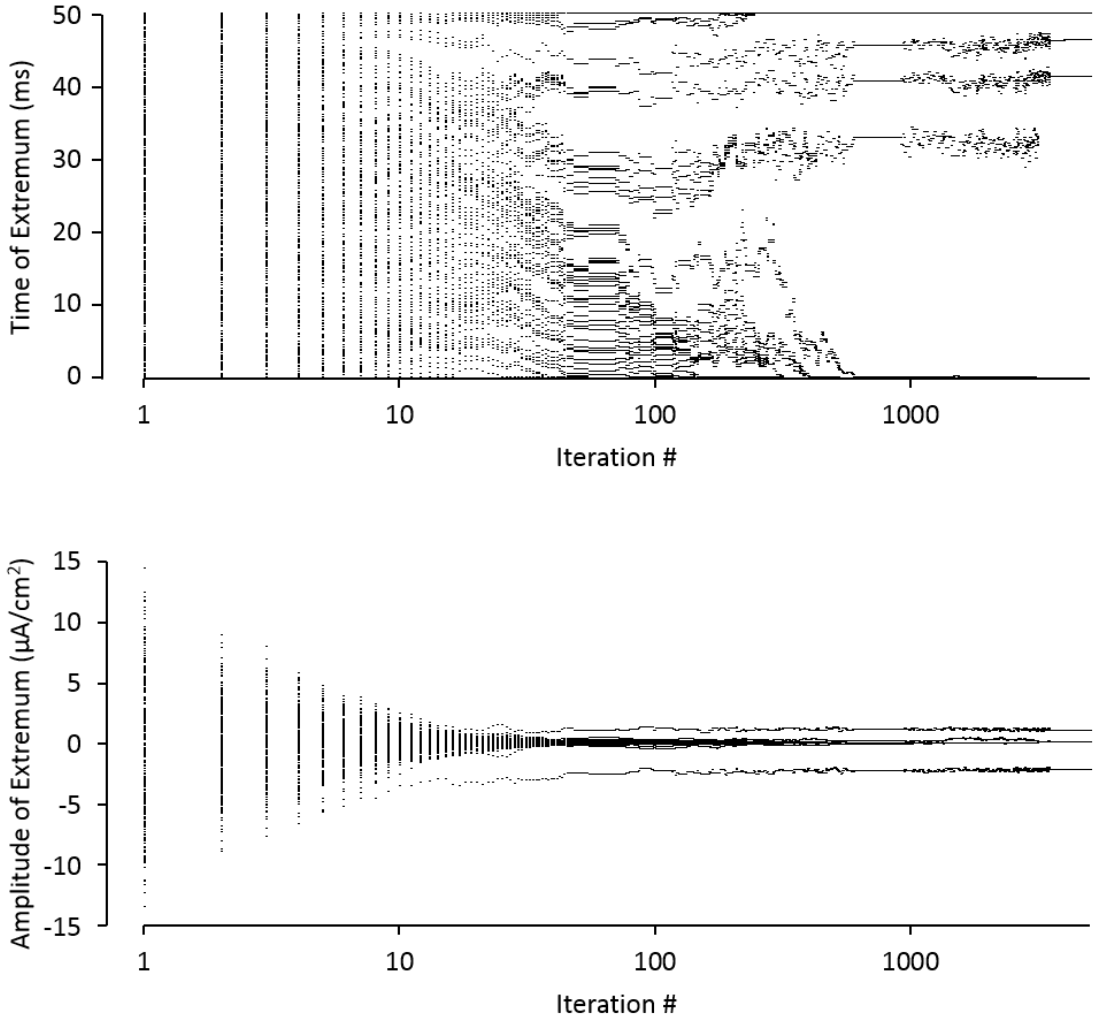


**Figure 6.4: The extrema in the signal evolves over the course of the algorithm. Both the timing (top) and the amplitude (bottom) of the extrema begin randomly scattered and over the course of a few iterations, it rapidly converges towards a few key locations. Over the course of a few hundred iterations, one can see the algorithm prune away at unnecessary extrema.**

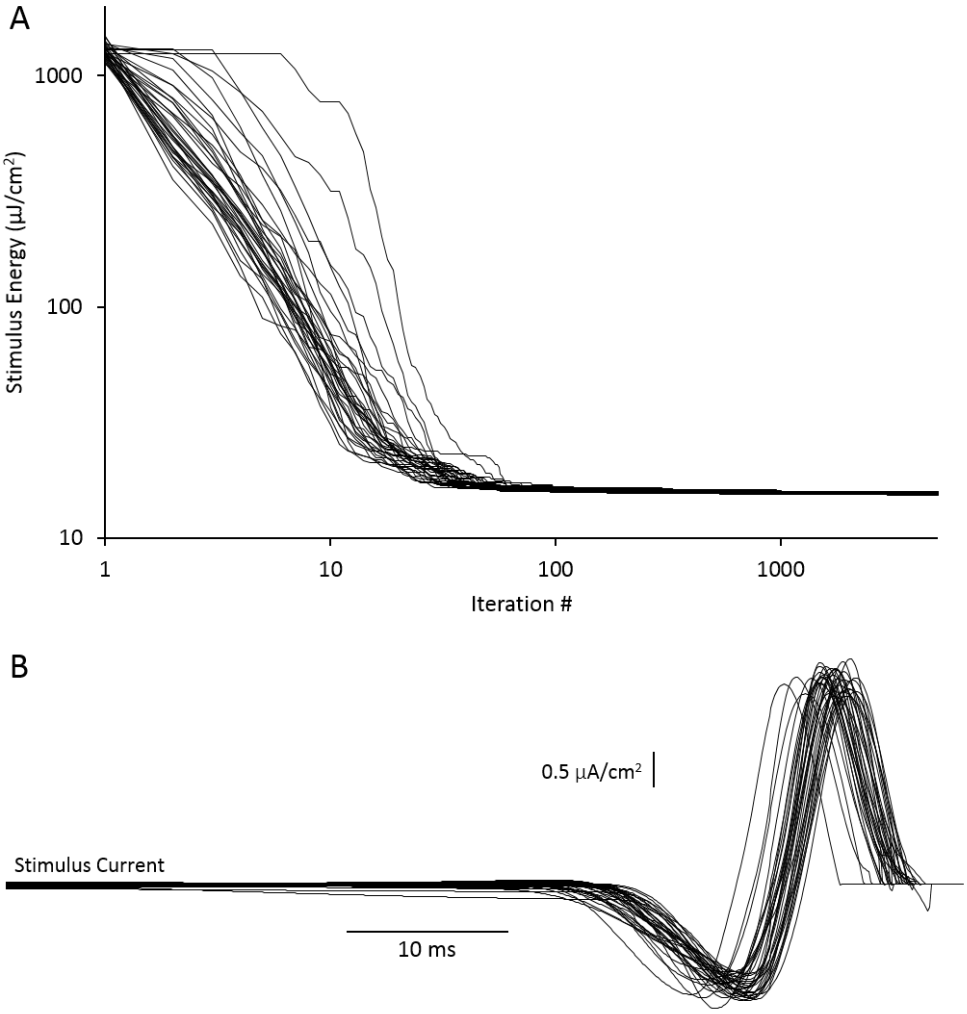


**Figure 6.5: The algorithm takes randomly generated stimuli and evolves them towards a single global optimal solution for the Hodgkin-Huxley model. As seen from the time course of the stimulus energy (A), the algorithm begins with very energetic stochastically generated stimuli and rapidly finds more energetically efficient stimuli. Panel (B) shows all of the resulting stimuli overlaid on top of each other, aligned by where they cause an action potential to occur. The stimulus' baseline is at 0 μA / $cm^2$**

The best of the 35 snippets had an $L^2$-norm of 15.270 compared to what we were able to determined using the gradient algorithm, 15.354. Figure 6.6A shows the comparison of the best result from the distortion algorithm compared to the result from the gradient algorithm. While we do see that the best of the Extrema Distortion Algorithm is better than the result from the gradient algorithm, we found that the variation among the 35 resulting snippets were much larger using the Extrema Distortion Algorithm compared to the gradient algorithm, as seen in Figure 6.6B.

The choice to use extrema instead of every single data point was to combat the "curse of dimensionality." We hypothesized that using the extrema alone would improve computational efficiency of the algorithm as well as produce results that are more accurate. In order to test this hypothesis, we reconfigured our algorithm to use every single data point in what we are calling the all-points distortion algorithm instead of the Extrema Distortion Algorithm. As we can see in Figure 6.7A, the improvement of $L^2$-norm across the 35 snippets using Extrema Distortion Algorithm converges towards an optimal solution faster than the improvement of $L^2$-norm across 35 snippets using the all-points distortion algorithm.

Quantitatively, the average $L^2$-norm of the all-points distortion algorithm is 15.902 with a standard deviation of 0.171. The best result from all-points distortion algorithm has an $L^2$-norm of 15.683. Figure 6.7B shows that the best result in all-points distortion retains a considerable level of noise as compared to the best result in extrema

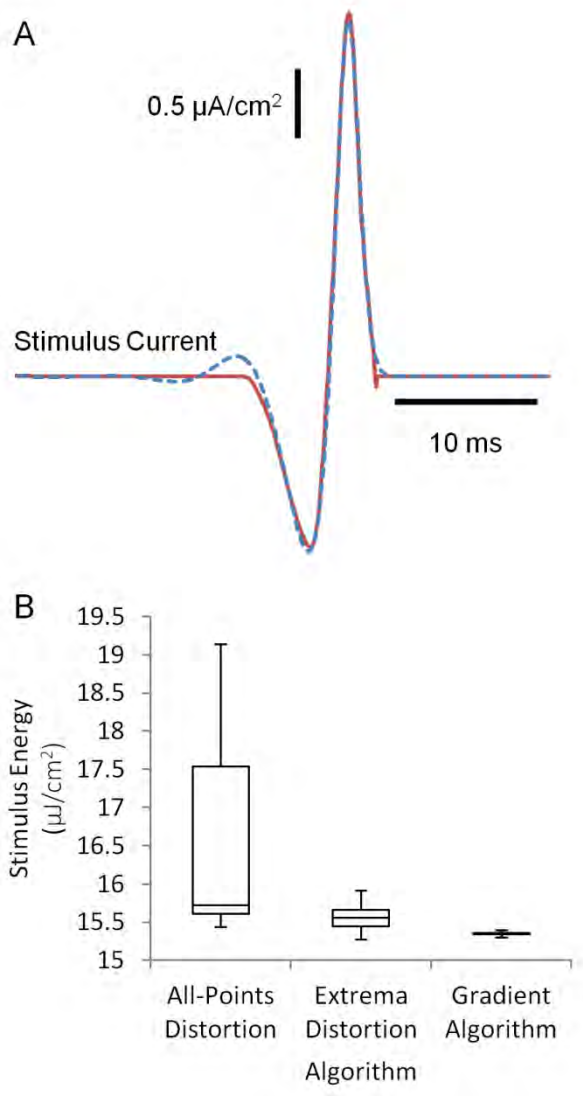


**Figure 6.6: The Extrema Distortion Algorithm finds solutions that match very closely with that found using the gradient algorithm. The result from the best of the 35 snippets using distortion of the extrema (red) is shown in panel A compared to the best of the 35 snippets using distortion of all the points (blue). While the fundamental shape is very similar, the result using distortion of all the points retains more noise than the result using distortion of only the extrema. Panel (B) shows the range of stimulus energy in 35 runs of both the Extrema Distortion Algorithm as well as the gradient algorithm. The stimulus' baseline is at 0 µA / cm$^2$**

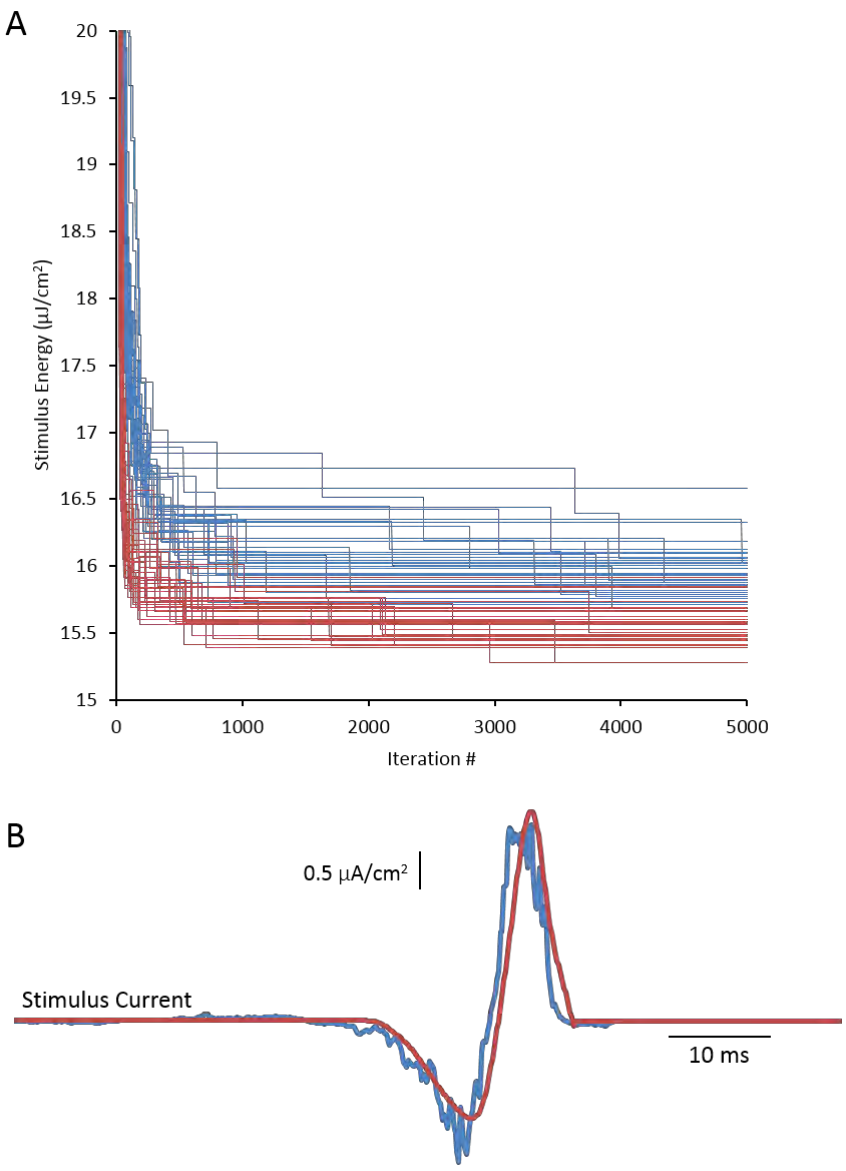


**Figure 6.7: The Extrema Distortion Algorithm (red) finds solution more efficiently than an all-points distortion algorithm (blue). We show both the $L^2$-norm progress of 5,000 iterations across both algorithms for the same initial 35 snippets in panel A. Panel B shows the best solution from both algorithms. The stimulus' baseline is at 0 µA / $cm^2$.**

distortion. Both of these results support our hypothesis that the extrema distortion is more efficient than the all-points distortion method, while maintaining a high level of accuracy.

## Switching off oscillations: The Fitzhugh-Nagumo model

To push the algorithm further, we applied the algorithm to a bistable FitzHugh-Nagumo model to see how the algorithm would perform when suppressing repetitive firing in a bistable system. We captured 49 white noise snippets, which we then ran through the Extrema Distortion Algorithm. We noted that the resulting stimuli fell into one of two modes: one mode with an average $L^2$-norm of 0.504 and a standard deviation of 0.079, while the other mode had an average $L^2$-norm of $7.207 \times 10^{-4}$ and a standard deviation of $2.778 \times 10^{-4}$. Figure 6.8 shows the final form of all 49 stimuli separated into the two modes. These waveforms correspond to results that we have obtained from the gradient algorithm, which produced two separate shapes, the smaller one with an $L^2$-norm of 0.0112 while the larger one had an $L^2$-norm of 1.1277.

We observed that the Extrema Distortion Algorithm performs better than the gradient algorithm in both shapes. Upon closer examination, we realize that this was because we had different definitions of successful transition between the two algorithms. The Extrema Distortion Algorithm was set to find an optimal solution that suppressed action potentials, while the gradient algorithm was find an optimal solution that suppressed all oscillatory activity. Because the FitzHugh-Nagumo quiescent state has a shallow basin

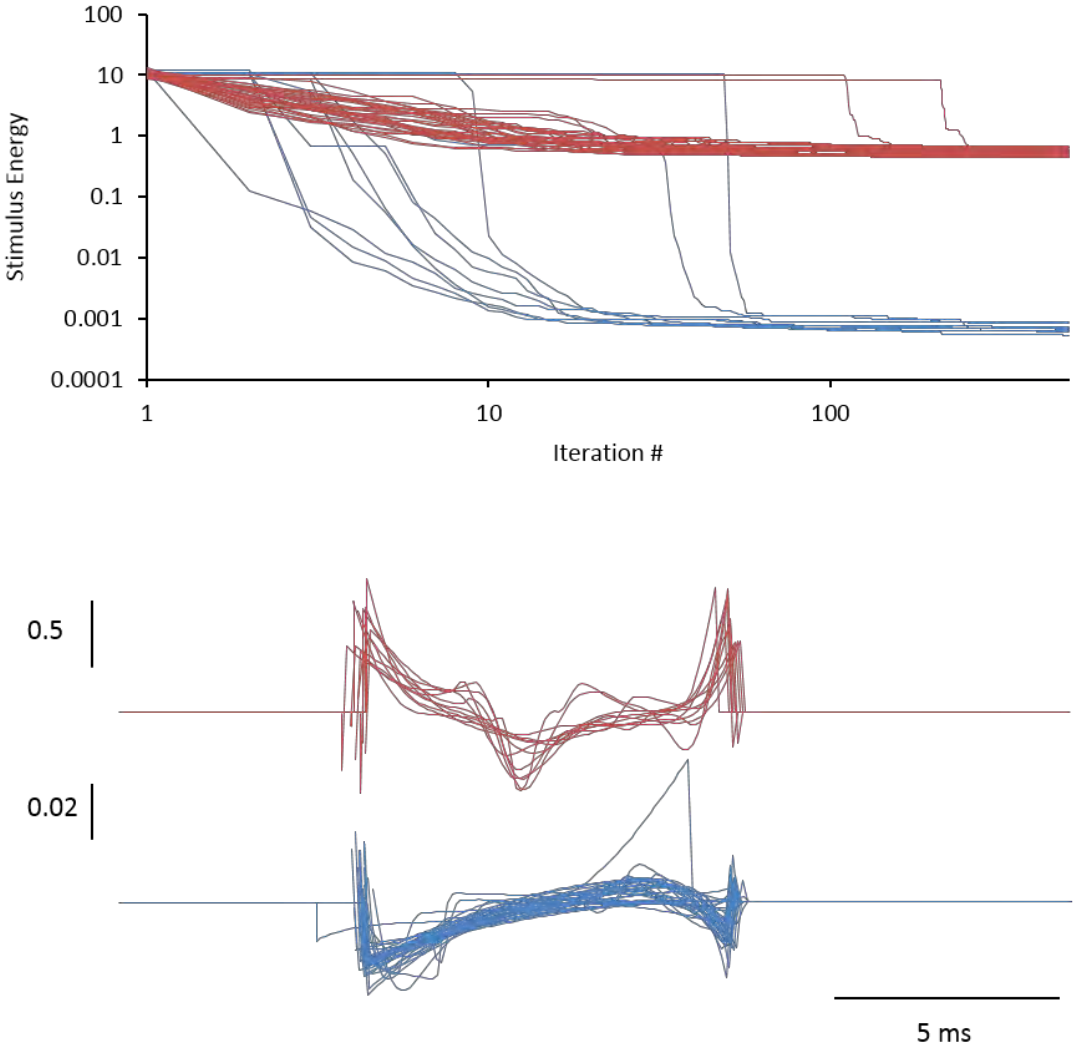


**Figure 6.8: The Extrema Distortion Algorithm converges towards the two locally optimal solutions in the FitzHugh-Nagumo model. The 49 snippets used on the FitzHugh-Nagumo model revealed two separate locally optimal waveforms. The top panels show the model response to the stimuli shown in the bottom panels. Note that the left stimulus is much smaller in scale than the right stimulus. The snippets are aligned such that the first valley after repetitive firing has ended occurs at 0 ms. The stimulus' baseline is at zero.**

of convergence, the state can oscillate at a sub-action potential level for a longer period before arriving at the fixed point. Thus, simply suppressing action potentials does not necessarily equate to arriving at the fixed point.

In order to account for this difference between the two algorithms, we applied stricter definitions of success to the Extrema Distortion Algorithm. Because we were unable to find any stimuli that took the system exactly to the fixed point of quiescence, we were unable to use that as the exact criteria. What we did instead was shrink the definition of success in different experiments until we were no longer able to find any successful white-noise solutions to use as starting seeds.

The resulting stimuli using a much tighter definition of success again was bimodal with the smaller set having an average $L^2$-norm of 0.0072 and a standard deviation of 0.0018, while the larger set had a an average of 1.094 and a standard deviation of 0.123. Figure 6.9 shows just the smaller set of solutions found and compares the result from the gradient algorithm to the results from both the loose and strict definitions of successful transitions. As seen here, the stricter definition of successful transitions does indeed find a waveform closer in shape to the solution found using the gradient algorithm.

Finally, we examine how robust the Extrema Distortion Algorithm works with long stimulus durations. The purpose of this experiment was two-fold. First, we wanted to test the robustness of the Extrema Distortion Algorithm when using large dimensional

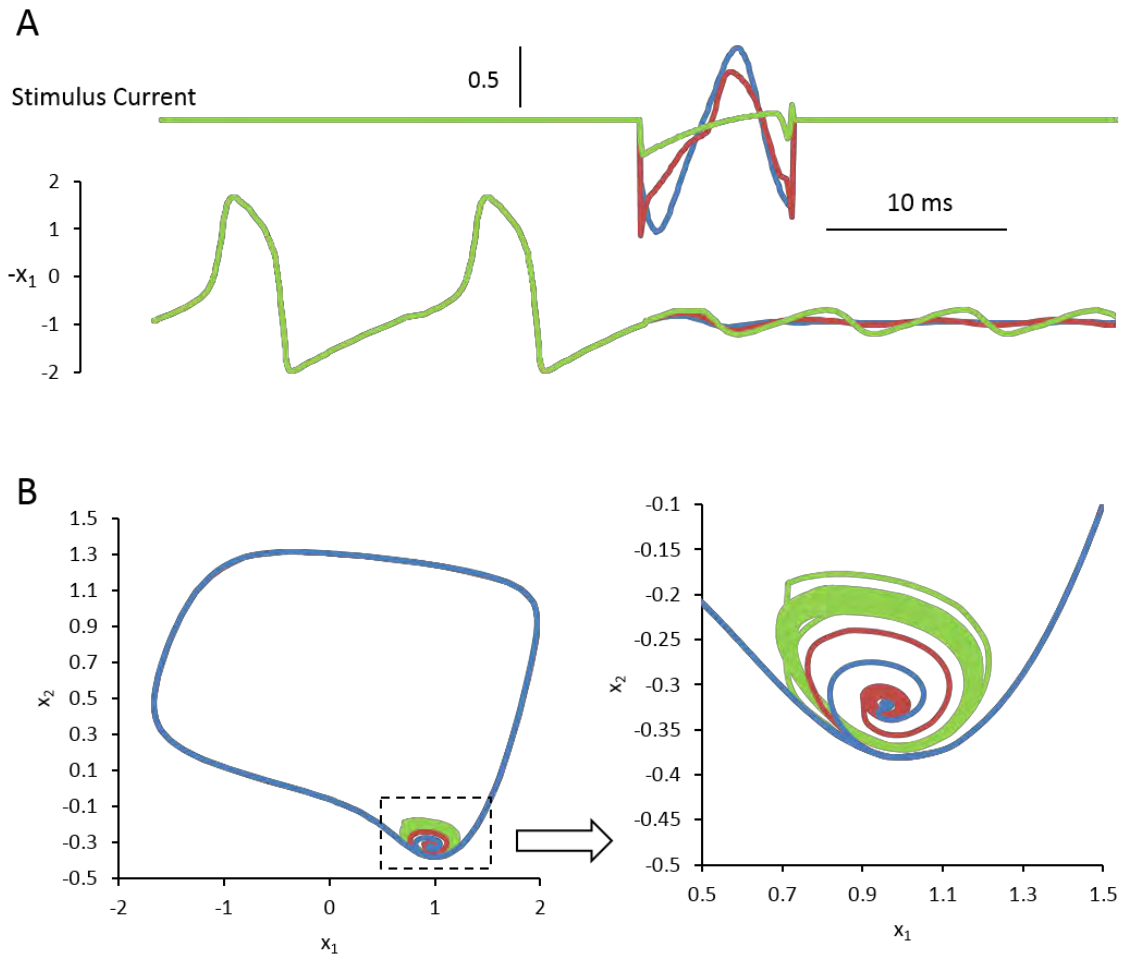


**Figure 6.9: The Extrema Distortion Algorithm allows for more flexibility in terminal conditions leading to more optimal results. Panel A shows the result from the gradient algorithm (blue) as well as two versions of the Extrema Distortion Algorithm. The first version of the Extrema Distortion Algorithm (red) limits the terminal condition of success to be extremely close to the fixed point similar to the gradient algorithm. The second version of the Extrema Distortion Algorithm (green) specifies the condition of success to just be a suppression of action potentials ($-x_1 < 1.5$). Panel B shows the state space of the system's response to the various stimuli (left) with a zoomed in version to what is occurring next the fixed point (right). The stimulus's baseline is at 0.**

stimuli. Secondly, from a clinical perspective, the optimal phase and stimulus duration are often unknowns. We wanted to find out if our algorithm could assist in the determination of these parameters by allowing it a great deal of flexibility to find optimal solutions. Figure 6.10 shows the results from 17 white-noise snippets, aligned by where the first sub-action potential peak occurs after the transition from repetitive firing to quiescence. As we can see, the algorithm does give general structures for what the shape of the optimal stimulus should be, as well as phase information. From what we can see in the figure, the optimal stimulus waveform is in anti-phase to the system response in order to cause successful transition from repetitive firing to quiescence. As seen in the figure, when we overlay all the solutions on top of each other, there is a remarkable amount of convergence between them.

## Flipping a genetic toggle switch

So far, our applications have been neurologically based, but this algorithm is completely model agnostic. Thus, we decided to expand our test cases to a non-neurological model. The genetic toggle switch, a relatively recent development in synthetic biology, is a bistable system, but instead of having one oscillatory state and one fixed state like the FitzHugh-Nagumo, both stable states are fixed. We ran 29 white-noise snippets through the Extrema Distortion Algorithm. The resulting optimal stimuli had an average $L^2$-norm of $2.968 \times 10^{-8}$ and a standard deviation of $2.25 \times 10^{-9}$. Figure 6.11A shows the best of the 29, with an $L^2$-norm of $2.563 \times 10^{-8}$, along with the results from the gradient

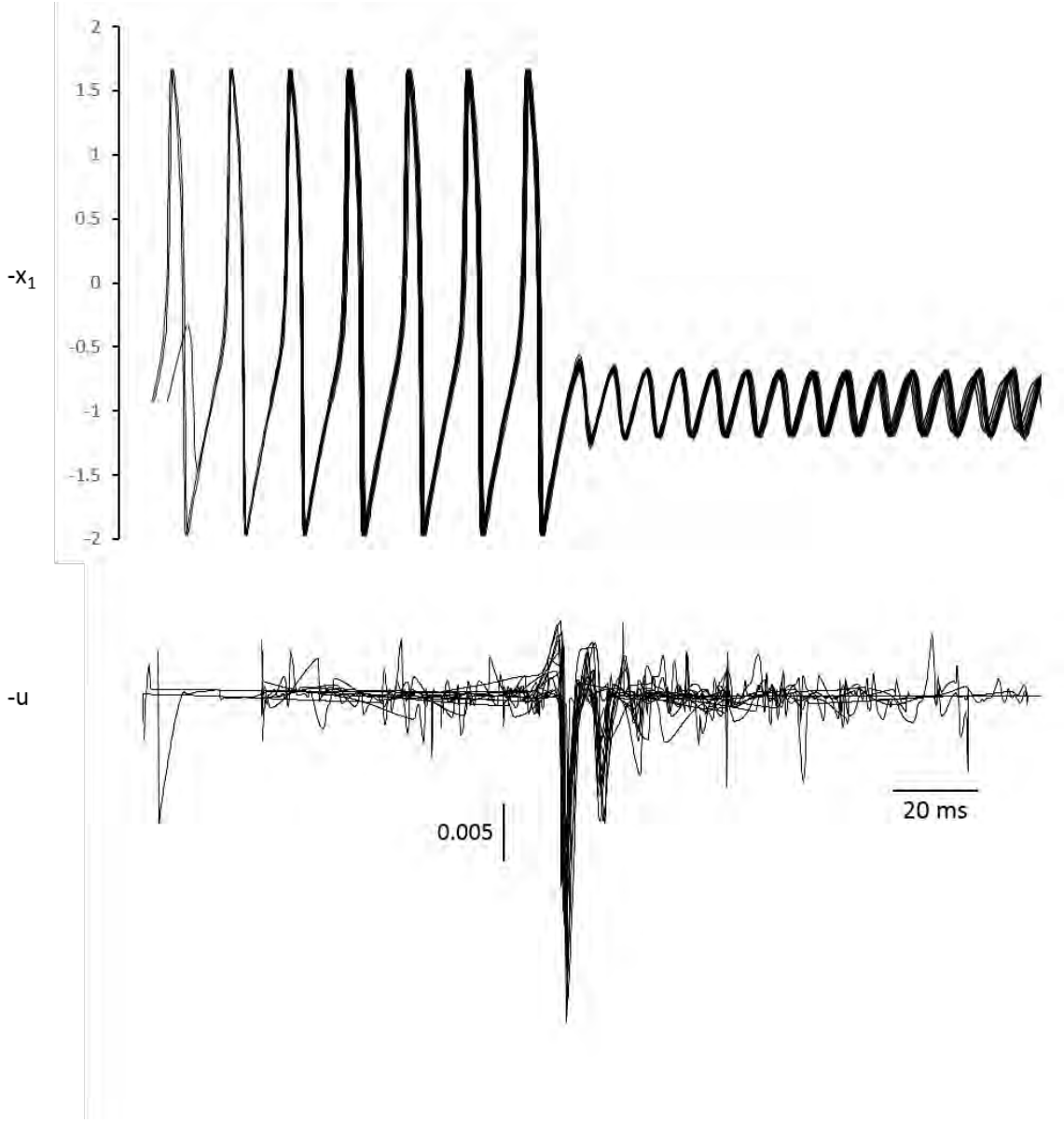


**Figure 6.10: The Extrema Distortion Algorithm can be used to gain insight into the phase-dependency and duration of the optimal stimulus. The Extrema Distortion Algorithm used on 17 unique 100-ms snippets shows converge towards a phase-specific stimulus waveform. These snippets were all aligned such that the first peak after the action potential occurred at 100 ms. The top panel shows the FitzHugh-Nagumo's response to the stimulus displayed in the bottom panel. The stimulus's baseline is at 0.**

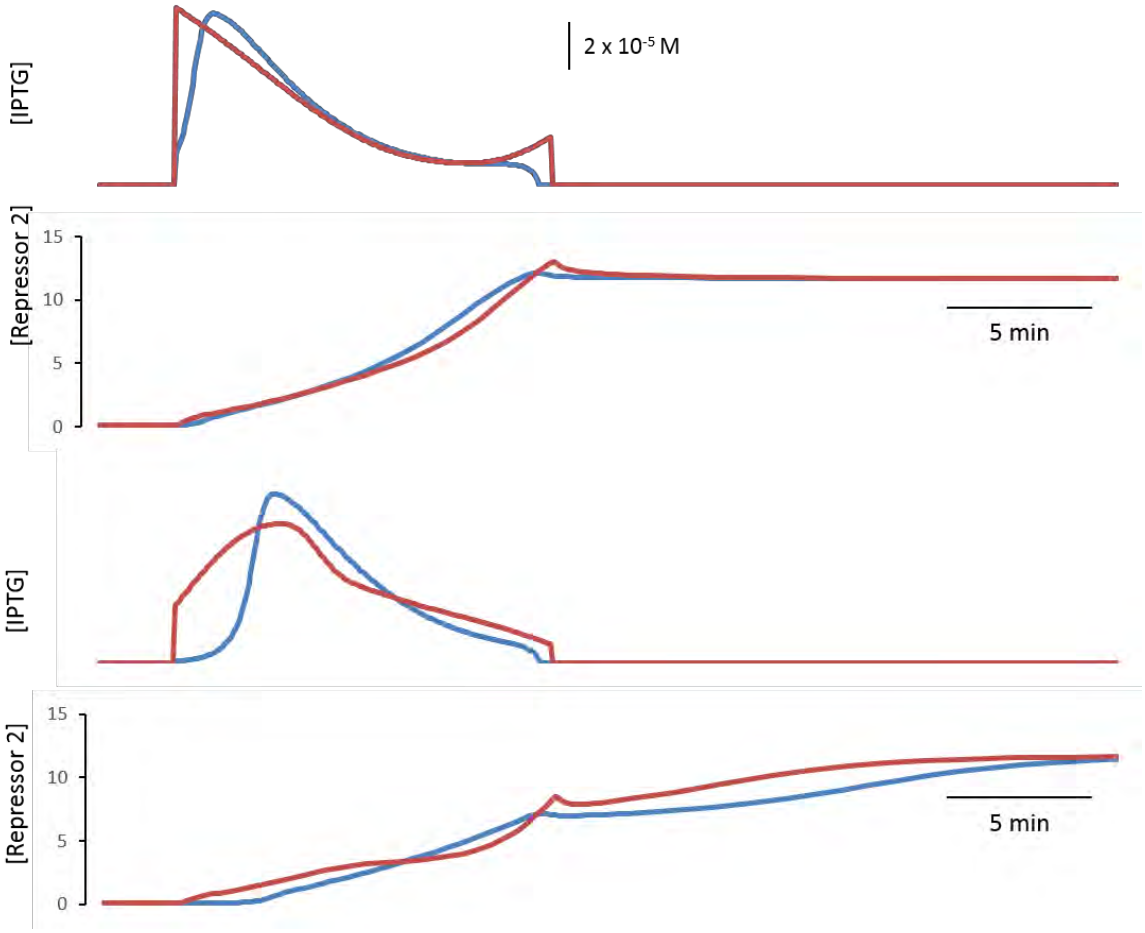


**Figure 6.11: The Extrema Distortion Algorithm can be used to find optimality under constrained conditions. In panel A, success is defined by seeing a transition occur at 20 minutes. The stimulus in terms of concentration of [IPTG] is shown above the concentration of repressor 2. As seen, the Extrema Distortion Algorithm (red) produces a very similar waveform to that of the gradient algorithm (blue). In panel B, success defines the transition occurring at 100 minutes, while the stimulus duration is still kept at 20 minutes. The gradient algorithm calculates the optimal stimulus using knowledge of where the separatrix, while the Extrema Distortion Algorithm does not. The stimulus starts at 0 M [IPTG].**

algorithm, which has an $L^2$-norm of $2.487 \times 10^{-8}$. Again, we can see that the two results are similar in both shape as well as $L^2$-norm value.

As we had seen with the FitzHugh-Nagumo model, a change in the definition of success could result in a different optimal solution. In searching for the optimal stimuli, we wanted the system to reach the terminal state at the end of the 20 minutes. For practical implementation, one may only need it to have switched from being in the basin of attraction of one state to being in the basin of attraction of the other state. In this desired implementation, one would only need to find an optimal stimulus that would cross the separatrix instead of travel all the way over to the other state. Thus, we reconfigured the gradient algorithm to find the optimal stimuli to the separatrix. We ran our Extrema Distortion Algorithm with a 20-minute stimulus, but instead of setting the condition of success as reaching the terminal state by the end of 20 minutes, we set it to reach the terminal state by 100 minutes. With the gradient algorithm, we had a mathematical model that we could use to calculate where the separatrix existed. Because we wanted to maintain our model-independent principle, we allowed the algorithm to reach the terminal condition at a more distant point in time, meaning that the stimulus must have cleared the separatrix in the first 20 minutes in order for it to transition. Figure 6.11B shows one of the results we found compared to the gradient algorithm. Again, we can see that the Extrema Distortion Algorithm was able to find a solution that had a very similar waveform shape and $L^2$-norm values as the gradient algorithm. It is interesting to see that the Extrema Distortion Algorithm was able to evolve a solution that learned where the

separatrix existed even without any knowledge of the mathematical model underlying the system.

So far, we have been doing all our analysis using the $L^2$-norm as an indicator of performance. When dealing with chemical reactions however, the $L^2$-norm has very little qualitative meaning. In most chemical reactions, it is more important to minimize the maximum dosage of the stimulus at any given time. We ran the Extrema Distortion Algorithm again using our goal as peak minimization. Figure 6.12 shows the resulting waveform compared to the optimal waveform based on minimizing $L^2$-norm.

## Evaluating the Use of Pre-Processing Noise Filter

In the course of this work, we speculated that some initial filter of the seed would be useful to increase the efficiency of the algorithm. As such, we explored this idea by using Empirical Mode Decomposition (EMD) to break down the successful white noise snippets into what we termed "distilled" snippets. We then processed these distilled snippets through the Extrema Distortion Algorithm, allowing us to evaluate the benefits of the noise filter.

Pure white noise snippets contain many extrema. We hypothesized that if we were able to reduce the number of extrema down to just the right number of extrema, each iteration of the distortion algorithm would require fewer computations, and thus over

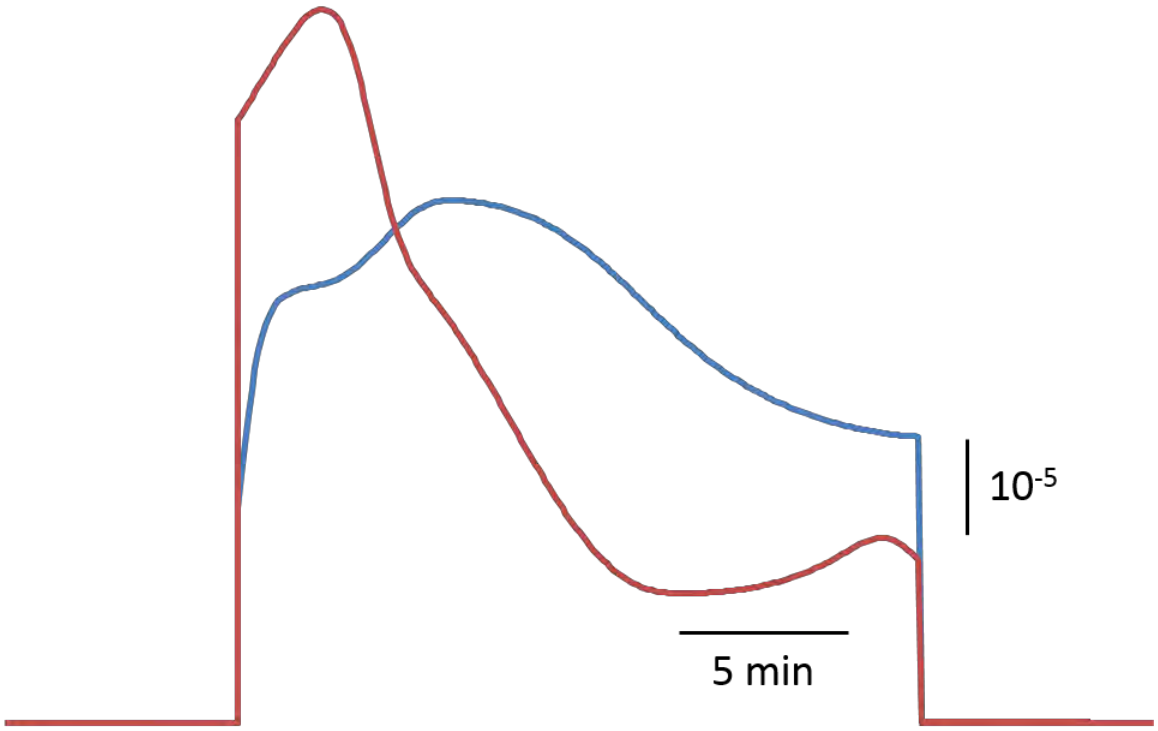


**Figure 6.12: The Extrema Distortion Algorithm can discover optimal stimuli based off of different performance metrics. Here we show the resulting optimal stimuli discovered by the Extrema Distortion Algorithm using the $L^2$-norm performance metric (blue) and the maximum amplitude performance metric (red). The stimulus starts at 0 M [IPTG].**

all the algorithm would be more efficient. In order to distill the snippets down, we considered using Fourier analysis, wavelet analysis, and EMD to break down the white noise snippets, determining the fundamental component of the snippet causing the state transition. We decided to use the Empirical Mode Decomposition. Because we were working with short discrete stimuli, we did not want to use Fourier analysis because of the known edge effects. Furthermore, we chose not to use wavelet analysis because we were concerned that the choice of wavelets may bias the results.

Thus, we chose to use EMD to separate out the white noise snippets into a set of intrinsic mode functions (IMFs). We tested every combination of consecutive IMFs, scaling each combination until it just barely caused a state transition. From these results, we chose the combination of consecutive IMFs that used the least amount of energy.

The distillation process is able to reduce many of the white noise snippets into very energy efficient snippets that successfully cause a state transition with a mean $L^2$-norm of 29.206 and standard deviation of 8.256. We used these resulting filtered snippets as the starting seeds of the Extrema Distortion Algorithm. The resulting distilled and distorted snippets have a mean $L^2$-norm of 16.695 and standard deviation of 1.3194.

Figure 6.13 shows that compared to just during extrema distortion alone with no filtering (mean = 15.550, standard deviation = 0.149), running EMD before EDA caused

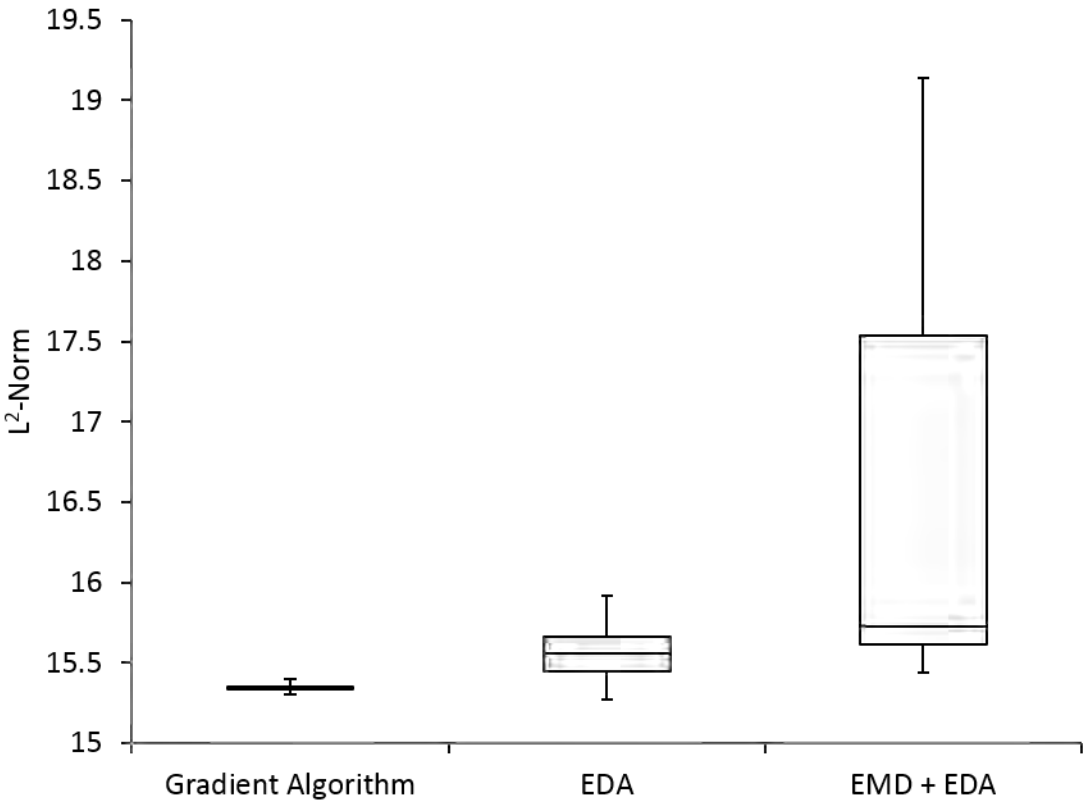


**Figure 6.13: Comparison of the ranges of $L^2$-norm results between the gradient algorithm, the Extrema Distortion Algorithm, and the Empirical Mode Decomposition combined with the Extrema Distortion Algorithm.**

increased variability. The best result from doing the pre-processing had an $L^2$-norm of 15.438, which was comparable to the 15.275 from the Extrema Distortion Algorithm alone. Figure 6.14 shows the best result from the Extrema Distortion Algorithm with and without pre-processing filter.

It is interesting to note that there is no correlation between how well a snippet did after the distillation phase as compared to the distortion phase. Furthermore, it would seem that the distillation process produced solutions that were worse than those generated from using only distortion were. Our conclusion is that it is possible for any sort of pre-processing work to oversimplify the white noise snippets, biasing the signal away from optimality. Future research and algorithm development may show how better to reduce the amount of noise to start from the white noise snippet, while not biasing the solution.

# Discussion

As can be seen in our results, the Extrema Distortion Algorithm is able to find stimulus waveforms that are very close to what we have analytically calculated to be optimal. The use of extrema in simplifying data sets is not a novel concept in the field of signal processing. In the field of pattern recognition, certain algorithms use the coordinates of robust extrema to simplify the number of points to match[180]. In computer vision, extrema are used as key feature markers for object recognition and multi-scale representation of the original images[181,182]. In many of these algorithms, extrema

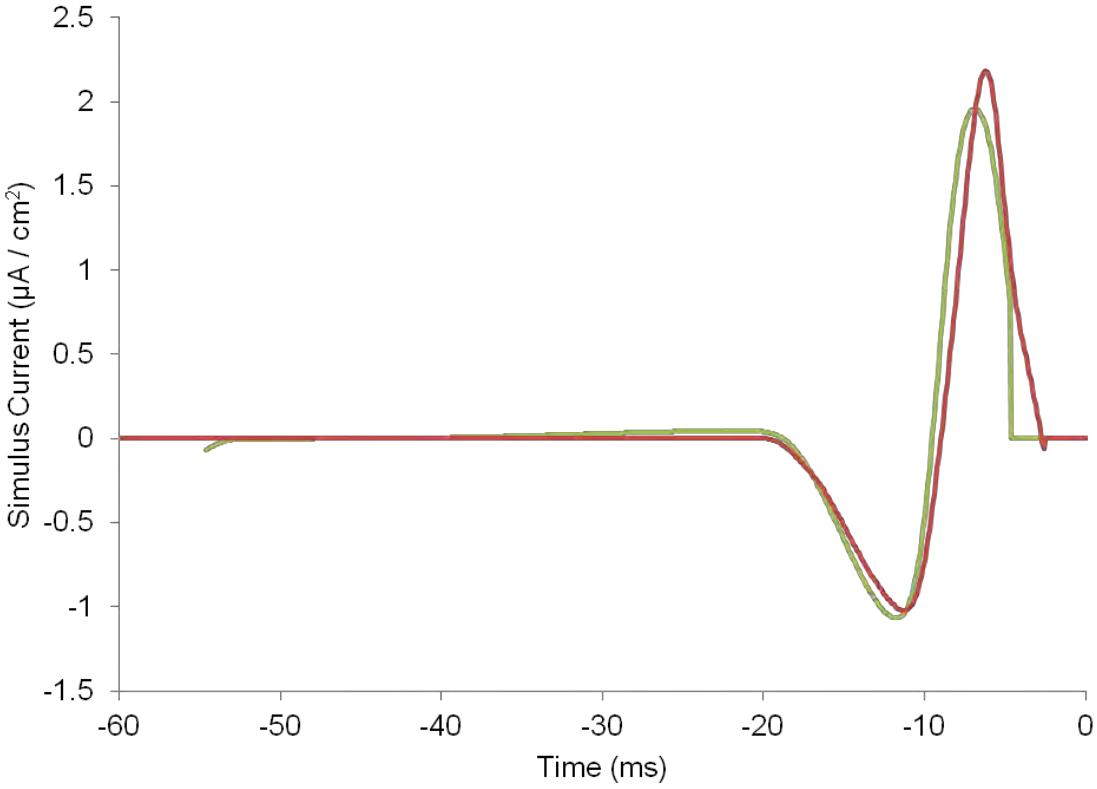


**Figure 6.14: Comparison of the best result from Extrema Distortion Algorithm (red) with the best result from the Extrema Distortion Algorithm combined with an initial Empirical Mode Decomposition denoising. The stimulus is aligned such that the action potential occurs at 0ms.**

mark key feature points, and thus, they construct a simplified representation of the original signal retaining only the most important components. This makes the manipulation and usage of the signals more efficient and simpler.

To our knowledge, algorithms that search for optimal stimulus waveforms have not used extrema. As we have stated, one of the biggest problems with searching for optimal stimulus waveforms is that they exist in a very high dimensional solution space. Some researchers approach this problem head on and apply search algorithms to the entire solution space[166], but others choose to simplify the solution space by making certain assumptions. One of the most common methods of simplifying the solution space is by assuming certain basic shapes. For instance, a number of studies search for the optimal dimensions of rectangular pulses for causing state transitions[72,76]. One study recently tried using the pivot points of cubic splines to redefine and simplify the solution space[69]. The problem with most of these studies that seek to simplify the solution space is that they bring in certain presuppositions of what the optimal stimulus waveform looks like. Studies have shown that the rectangular pulse is not energetically efficient[34,183], and thus the use of these simplifications limit the possibilities of even better optimal stimulus shapes.

One of the benefits of the Extrema Distortion Algorithm is that there are no presuppositions about optimality introduced into the algorithm. The algorithm begins with stochastically generated noise containing many extrema. This is beneficial in that it allows the stimulus to have more degrees of freedom to explore for optimality at the

beginning when the algorithm knows little about the optimal solution. As the algorithm progresses, unnecessary extrema are filtered out, increasing the efficiency of the algorithm.

One weakness with the algorithm is that we assume a linear transformation from one set of extrema to the other. Once the waveform shape has been determined between extrema, there is little chance that it will change again. So far, in the three examples we have used, we have not noticed this to be a problem. We postulate that because we are beginning with many extrema, each extrema has the opportunity to be at the proper inflection points in the waveform before they disappear. To improve this technique in the future, we may want to examine methods to aid in the search and verification of intermediate points as well as the extrema.

In the Hodgkin-Huxley model, we noticed that the best result from the Extrema Distortion Algorithm produced an even better result than that found using the gradient algorithm. While this surprised us at first, we realized that this result showed us the limitations of the gradient algorithm. We used a first-order gradient algorithm for its robustness to initial conditions. However, a first-order gradient algorithm has the weakness of not being able to fine-tune convergence when it sits very close to optimality because it overestimates or underestimates the next step due to its use of only first-order gradients. Many algorithms need to trade between the ability to converge towards optimality and robustness to initial conditions. The fact that the Extrema Distortion Algorithm can at times surpass analytical methods is thus a pleasant surprise. That being

said, it is important to realize that while the best of the Extrema Distortion Algorithm was able to successfully find a better solution, there were many solutions found that although were close, were not as good as the gradient algorithm. This variability is an inherent component of any stochastic algorithm.

While this algorithm has performed both accurately and efficiently, it is a rather simplistic algorithm in the world of model-independent optimization techniques. By examining these other techniques, we can gain ideas on how to improve both the accuracy and the efficiency of this algorithm. We highlight at least three areas that we think future research may explore using our algorithm as a first step: initial noise filtration, cross learning between different seeds, and incorporation of statistical principles to guide choice of next stimulus.

One of the first things we noticed when developing this algorithm is that there is an excessive level of noise in the starting seeds. We speculate that if we were able to remove this noise, we would be able to converge quicker towards the optimal solution. We describe later in this chapter our first attempt to filter out some of the noise. Unfortunately, we have found that while filtering can help reduce the noise, how well a stimulus does after filtering does not necessarily correlate to how well it will do after the distortion process. Furthermore, sometimes filtering the stimulus can bias it towards a specific waveform shape and thus push the solution further away from optimality as opposed to towards optimality.

Secondly, we recognize that the Extrema Distortion Algorithm utilizes at its heart a stochastic hill-climbing approach [172], which starts with a single seed and progresses its way forward without any input from how other parallel processes are performing. As we have stated before, stochastic hill-climbing is one of the simplest approaches. More recently, there has been growing interest in population-based evolutionary algorithms like genetic algorithms and particle swarm optimization. These algorithms incorporate a learning process between different runs of the algorithm that start with different seeds. One of the earlier studies that looked at model-free methods to obtaining optimal stimulus waveforms used genetic algorithms[166]. Unfortunately, most of these population-based algorithms require all the solutions share a common dimensionality in order for them to be able to recombine. Our algorithm does not meet that criterion as the number of extrema changes from one solution to the next. However, the concept of allowing some form of learning between solutions is an area for future investigation as we may be able to boost efficiency of the algorithm.

Finally, the algorithm chooses the next stimulus by examining a set of randomly chosen neighboring stimuli with no guidance as to where better solutions may come from. The algorithm samples all neighboring solutions with equal probability. One interesting approach that we came across when examining optimal stimulus waveforms was efficient global optimization (EGO)[184], which constructs a generalized least squares model of the response surface by making the assumption that solutions close together will elicit similar responses from the system in question. The algorithm tests a set of points in the solution space in such a way as to minimize the error potential of the response surface

model by choosing new points to test that have the highest level of uncertainty as determined through statistical means.

Unfortunately, this method assumes a continuous response surface. In our search for optimal stimulus waveforms, we need to take into account the fact that certain stimuli do not successfully cause the state change that we want. We break the assumption that neighboring solutions have similar responses due to solutions that sit close to the boundary between successful and unsuccessful stimuli. While EGO may not directly be applicable, the concept of incorporating some sort of statistical measurement to guide the algorithm in sampling neighboring solutions is an area of future research that could improve the efficiency of the algorithm further.

What we have done in this study is present a first step in the development of a stochastic optimization algorithm for black-box systems to find optimal stimulation for physiological systems that utilizes extrema to simplify the search space. With further development by potentially incorporating noise filters, learning mechanisms or statistical principles, we may see these algorithms applied in a clinical setting to determine efficiently optimal stimulus waveforms for therapeutic treatments.

# CHAPTER VII

# Extrema Distortion Algorithm: Optimally Suppressing Epileptic Seizures

## Introduction

Epilepsy affects 2.3 million adults and over 450,000 children under the age of 18 in the United States[185,186]. Physicians diagnose roughly 150,000 new cases each year. The total financial burden of this disease in both direct and indirect costs was estimated at $9.6 billion in 2009[187]. While the primary form of treatment for epilepsy is in the form of anti-seizure medication, nearly 30% of these patients do not respond to drug therapies[188]. While some patients can undergo surgery to treat refractory epilepsy, not all medically refractory patients are candidates for these procedures. As an alternative, it has been shown that electrical stimulation of specific regions of the brain can aid in the suppression of seizure frequency and severity[189–195].

One of the key features of epileptic seizures is dysfunctional neurological oscillatory activity. Treatment for epilepsy via electrical stimuli has focused on the rectification or suppression of these dysfunctional oscillations. To that end, three different types of treatments have been developed: vagus nerve stimulation[196–198], deep brain

stimulation of the thalamus, the subthalamic nucleus, the caudate nucleus and the cerebellum [24,189,199], and responsive neural stimulation of cortical epileptogenic regions[192–194]. In each of these treatments, a fundamental waveform is used, and physicians tweak a few basic parameters including frequency, pulse duration, and amplitude based on how the patient responds in order to improve performance.

An important computational challenge arises here not only to find stimuli that successfully suppress seizures, but also to minimize its energy usage. Each of the existing treatments has some adverse effects potentially related to energy leakage from the stimuli into neighboring areas. For instance, patients that are given vagal nerve stimuli have demonstrated coughing, voice alteration, paraesthesia, dyspnea and headache[197,198]. Patients undergoing deep brain stimulation appear to be less prone to adverse effects, but there have been reports of depression, memory impairment, anxiety, and paraesthesia[189] related to stimulation. Patients undergoing responsive neural stimulation of the cortex have shown instances of implant site pain, headaches, and dysesthesia[193].

In this chapter, we apply the Extrema Distortion Algorithm to four separate mathematical models of epilepsy based on different mechanisms examining different levels of complexity. We demonstrated how this algorithm is able to find optimal solutions relatively quickly without any bias towards one waveform over another and without needing any knowledge of the mechanisms underlying the models. By applying this

algorithm to the four separate models, we seek to gain therapeutic insights into the clinical treatment of epilepsy with electrical stimulation.

# Methods

## Extrema Distortion Algorithm

The Extrema Distortion Algorithm (EDA) finds optimal stimulus waveforms without requiring any *a priori* knowledge of the underlying mechanisms or mathematical equations defining the system or behavior. EDA works by iteratively taking a starting waveform and allowing its extrema (the peaks and valleys) of the stimulus, to move around in search for a more optimal stimulus. This more optimal stimulus becomes the starting waveform for the next iteration.

We first seed the algorithm by applying white noise into the model in search for snippets that successfully cause the desired outcome. The white noise snippets that successfully trigger the outcome turn into the seeds for EDA. In all of the experiments done in this study, we sought to minimize the energy usage of the stimulus as conventionally defined by the $L^2$-norm, which is the sum of the squares of the stimulus amplitudes over time.

Devices giving electrical charge to the body attempt to do so in such a way that the net electrical charge injected is equal to zero. This is because if excessive charge builds up in the body, irreversible damage is done to the neural tissue[200,201]. As such, we added a new constraint to our algorithm such that after every distortion, we subtract the mean of the stimulus from the stimulus. This forces each distortion to produce only zero-mean stimulus waveforms. Thus, we were able to restrict the search space for all the

experiments to return only charge-neutral solutions. We ran each experiment twenty times with different starting seeds each time.

## Single cell repetitive firing

In many instances of epilepsy, seizure activity arises from a derangement in the neuron's intrinsic properties resulting in repetitive firing[11,202]. A subset of epilepsies falls in the category of channelopathies, mutations in genes that encode various ionic channels. Among these channelopathies, forms of generalized epilepsy with febrile seizures has been linked to mutations in genes encoding voltage-gated sodium channel α, leading to persistent inward current which likely causes increased excitability of the neuron's membrane[202]. To model this, we have added a 9 μA/cm$^2$ persistent current to the Hodgkin-Huxley model, essentially increasing the excitability of the neuron's membrane. Our model consists of the following equations:

$$C\dot{V} = -120m^3h(V-115) - 36n^4(V+12) - 0.3(V-10.613) - 9 - u \quad (6\text{-}1)$$

$$\dot{m} = -m\big(\alpha_m(V) + \beta_m(V)\big) + \alpha_m(V) \quad (6\text{-}2)$$

$$\dot{n} = -n\big(\alpha_n(V) + \beta_n(V)\big) + \alpha_n(V) \quad (6\text{-}3)$$

$$\dot{h} = -h\big(\alpha_h(V) + \beta_h(V)\big) + \alpha_h(V) \quad (6\text{-}4)$$

$$\alpha_m(V) = \frac{0.1\phi(25-V)}{e^{0.1(25-V)}-1},\ \beta_m(V) = 4\phi e^{-V/80} \quad (6\text{-}5)$$

$$\alpha_n(V) = \frac{0.01\phi(10-V)}{e^{0.1(10-V)}-1},\ \beta_n(V) = 0.125\phi e^{-V/80} \quad (6\text{-}6)$$

$$\alpha_h(V) = 0.07\phi e^{-V/20},\ \beta_m(V) = \frac{\phi}{e^{0.1(30-V)}+1} \quad (6\text{-}7)$$

where $V$ is the membrane voltage, $m$, $n$, and $h$ represent dimensionless quantities associated with sodium channel activation, potassium channel activation, and sodium channel inactivation respectively, and $u$ represents the exogenous stimulation we are looking to input into the system. The addition of the persistent current causes the Hodgkin-Huxley model that we have defined here to be bistable representing the increased excitability found in cells prone to seizures.

Using EDA, we found the optimal 31-ms stimulus (the duration of two cycles of repetitive firing) that successfully suppressed the system from repetitive firing to quiescence where the membrane voltage no longer went above 50 mV.

## Single cell bursting

There has been recent work that has examined the possibility that the epileptic cell is not repetitively firing, but instead bursting (e.g. firing sets of action potentials followed by temporary quiescence). Cressman et al. have worked to develop a more accurate model of a single-cell neuron that includes the interaction of the neuron with the different ionic concentrations in its microenvironment[203] in an attempt to model this bursting behavior.

The equations for this model are as follows:

$$C\dot{V} = I_{Na} + I_K + I_{Cl} \tag{6-12}$$

$$I_{Na} = -g_{Na}[m\ \ (V)]^3 h(V - V_{Na}) - g_{NaL}(V - V_{Na}) \tag{6-13}$$

$$I_K = -\left(g_K n^4 + \frac{g_{AHP}[Ca]_i}{1+[Ca]_i}\right)(V - V_K) - g_{KL}(V - V_K) \tag{6-14}$$

$$I_{Cl} = -g_{ClL}(V - V_{Cl}) \tag{6-15}$$

$$\dot{q} = \phi\left[\alpha_q(V)(1-q) - \beta_q(V)q\right], \text{ where } q = n, h \tag{6-16}$$

$$[\dot{Ca}]_\iota = \frac{-0.002 g_{Ca}(V - V_{Ca})}{1 + e^{-\frac{V+25}{2.5}}} - \frac{[Ca]_i}{80} \tag{6-17}$$

$$[\dot{K}]_o = -0.33 I_K - 2\beta I_{pump} - I_{glia} - I_{diff} \tag{6-18}$$

$$[\dot{Na}]_\iota = 0.33\frac{I_{Na}}{\beta} - 3I_{pump} \tag{6-19}$$

Variables and accessory equations are detailed in [203].

What is interesting about this system compared to other experiments that we have done is that this system is not bistable. The system only exists in its bursting state, and it is impossible to suppress the bursting behavior indefinitely. As such, we try to accomplish something a little different. Instead of seeking to suppress the system for a long period, we seek to suppress the burst quicker than normal, while maintaining when the next burst will occur.

The bursting state cycles every 38 seconds, with action potentials firing the first 6 seconds and quiescence in the remaining 32 seconds. Using EDA we sought to give a 2-second stimulus to the system 100 milliseconds after the bursting begins, suppressing any action potentials from occurring for 36 seconds following the end of the stimulus.

## Coupled oscillators network

From a network perspective, there is a hypothesis that one of the mechanisms underlying epileptic seizures is a synchronization of various regions of the brain[204–207]. Synchronization has been observed in rat hippocampal slices perfused with high potassium saline solutions[208]. One of the alternative electrical therapies for patients has been vagal nerve stimulation (VNS). While its exact mechanism is still unknown, it is hypothesized that VNS works by desynchronizing neuronal activity, an idea supported by the EEG response to VNS[209].

As such, we have attempted to model a network of five coupled Hodgkin-Huxley neurons. We chose five for computational purposes, but we could scale up the model relatively easily. In order to model them as oscillators, we have increased the persistent current to 12 μA/cm$^2$ such that the system is monostable repetitive firing. We coupled every neuron with every other neuron by using a constant coupling factor and the voltage differences between the neurons:

$$\alpha \sum_{j=1}^{5} (V_i - V_j) \tag{6-20}$$

where $i$ represents the neuron in question, $j$ represents all neurons and $\alpha$ is the coupling constant. The new model of equations replaces Eq. (6-1) with:

$$C\dot{V}_i = -120{m_i}^3 h_i(V_i - 115) - 36{n_i}^4(V_i + 12) - 0.3(V_i - 10.613) - 12 - \alpha\sum_{j=1}^{5}(V_i - V_j) - u \quad (6\text{-}21)$$

where $V_i$, $m_i$, $n_i$, and $h_i$are the state variables of the $i^{th}$ neuron. All the other equations still applied.

We started the five neurons in a near synchronous state. We chose to use a 31-ms stimulus duration (two cycle lengths). In this model, the mean-field potential of the five neurons is representative of what an EEG reading. From our white noise snippets, we noted that a very small percentage (0.02%) successfully dropped the mean-field below 32.5 mV. As such, we set up EDA to find the optimal 31-ms stimulus waveform that would take the mean-field potential to below 32.5 mV by the end of the 31 milliseconds.

## Population-based systemic bursting

The models of epilepsy can be generally categorized as either microscopic, ionic models and macroscopic, population models[210]. The first three models we have examined fall in the first category, examining and quantifying the ionics of the system. Because epilepsy affects extremely large networks, it is difficult and time-intensive to calculate the ionic mechanism underlying each individual neuron in these large networks. As such, researchers have developed population-based models that lump together groups of neurons into their mean-field components, modeling interactions between populations of neurons as opposed to individual neurons.

For our experiment, we have examined a simple bistable system developed by Suffczynski et al[211]. This model examines epilepsy through the interactions of four populations of neurons: pyramidal neurons, interneurons, thalamocortical neurons and reticulothalamic neurons. In this model, the pyramidal neurons qualitatively mimic EEG activity. Figure 7.1 shows a diagram with the various interconnections modeled between the four populations of neurons. The researchers tailored the parameters to match experimental data collected from the Wistar albino Glaxo from Rijs-wijk (WAG/Rij) rat, which is a genetic animal model of absence epilepsy.

In this model, there exist two states, a quiescent state, and a seizure state delineated by the strength of the pyramidal neurons' mean-field voltage. The system can spontaneously transition on its own from one state to the other due to a low level of sensory noise programmed into the model. We found a set of initial conditions for the system such that the system begins in the quiescent state, but then spontaneously transitions into a seizure state a second later. For this model, we used EDA to find the optimal 100-ms waveform that was sufficient to suppress the seizing state back to quiescence within 100 milliseconds after the stimulus terminated.

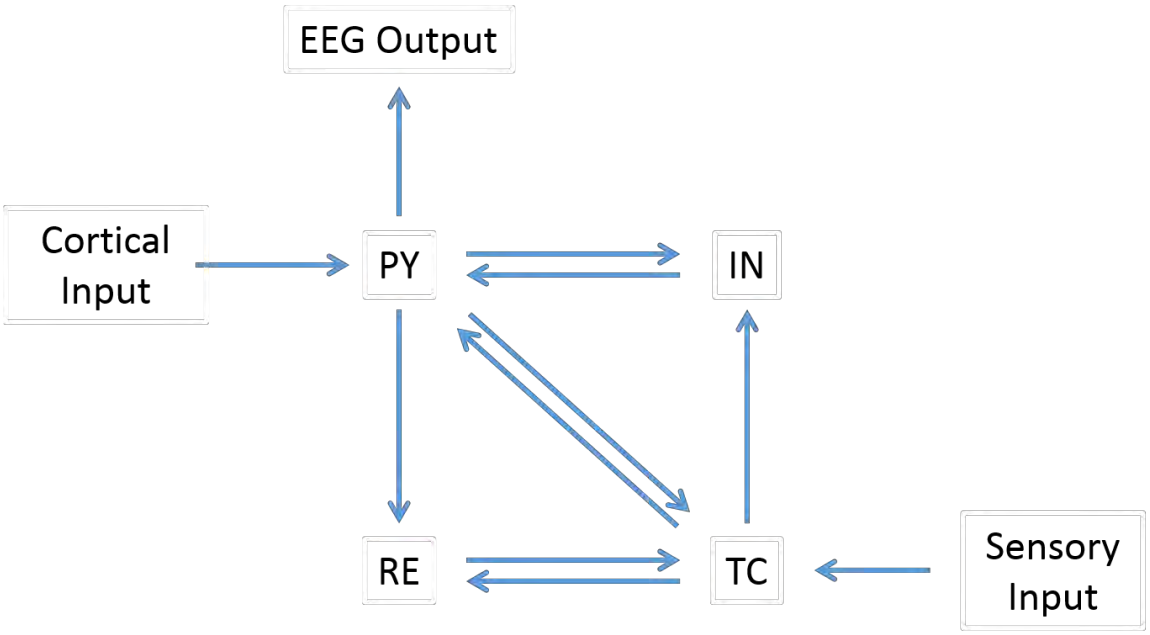


**Figure 7.1: Diagram of the Suffczynski et al model. Four populations of neurons are modeled: pyramidal neurons (PY), interneurons (IN), reticulothalamic neurons (RE), and thalamocortical neurons (TC).**

# Results

## Suppression of single cell repetitive firing

Figure 7.2 shows the optimal stimulus found through EDA. As seen in this figure, the optimal stimulus is multi-phasic with a prominent biphasic component in the center. The stimulus is also in anti-phase with the membrane voltage.

As a point of comparison, we did a grid search of biphasic rectangular pulses with varying stimulus duration (1 ms to 20 ms per phase) and stimulus amplitude (-10 $\mu A/cm^2$ to 10 $\mu A/cm^2$ for the first phase, the inverse for the second phase, in 0.1 $\mu A/cm^2$ increments). The optimal rectangular pulse that we found was 2-ms duration per phase, 1.4 $\mu A/cm^2$ and -1.4 $\mu A/cm^2$ for the first and second phase respectively and had a total energy usage of 7.84 $\mu J$. The optimal stimulus waveform found using EDA had a total energy usage of 2.63 $\mu J$.

## Suppression of single cell bursting

This model of the single cell burster is a little different from the other three models in that the cell is not in a clearly defined repetitive firing state or a quiescent state, but instead in perpetual bursting state. Figure 7.3 shows the optimal 2-second stimulus that successfully suppressed one burst prematurely while maintaining when the next burst occurs. What is difficult to see in the figures is that there are 168 action potentials occurring both when the stimulus is given and when the stimulus is not given. We discuss this result further in the discussion.

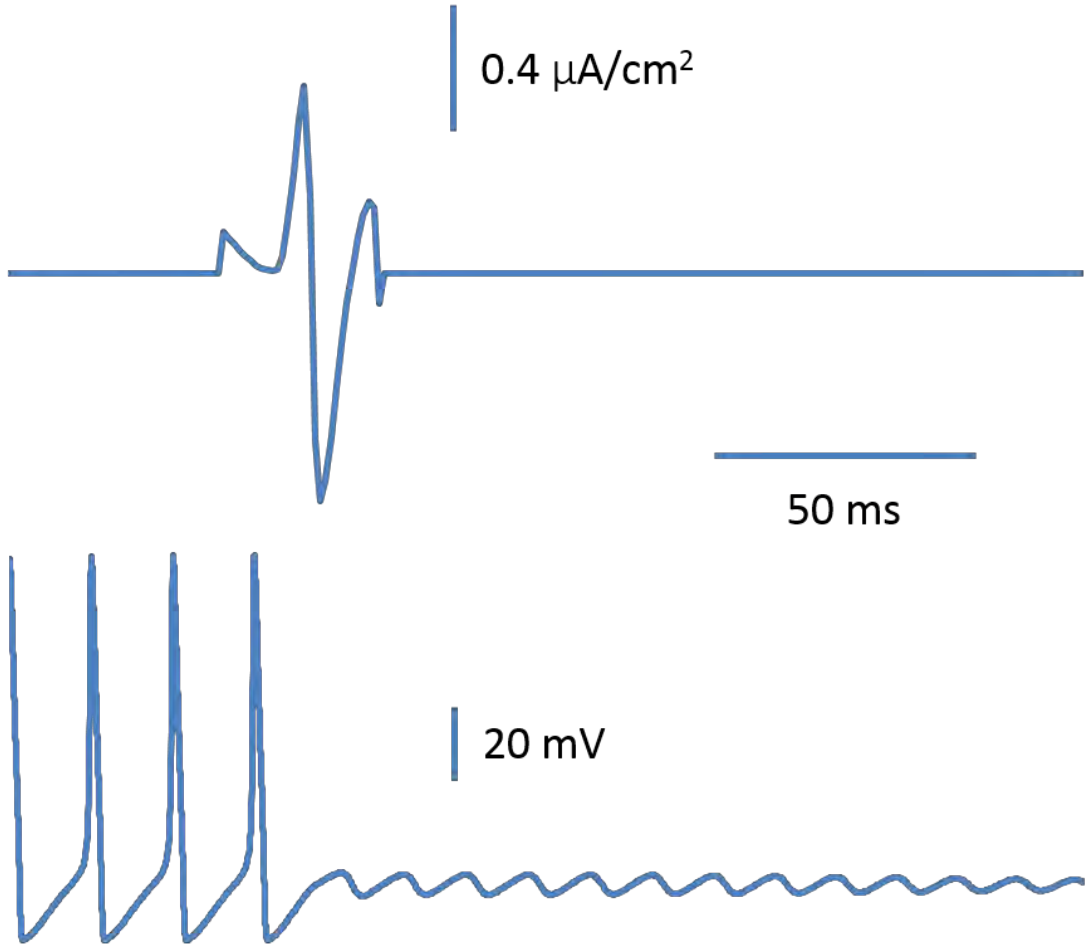


**Figure 7.2: Optimal stimulus waveform for suppressing repetitive firing in a single cell.**

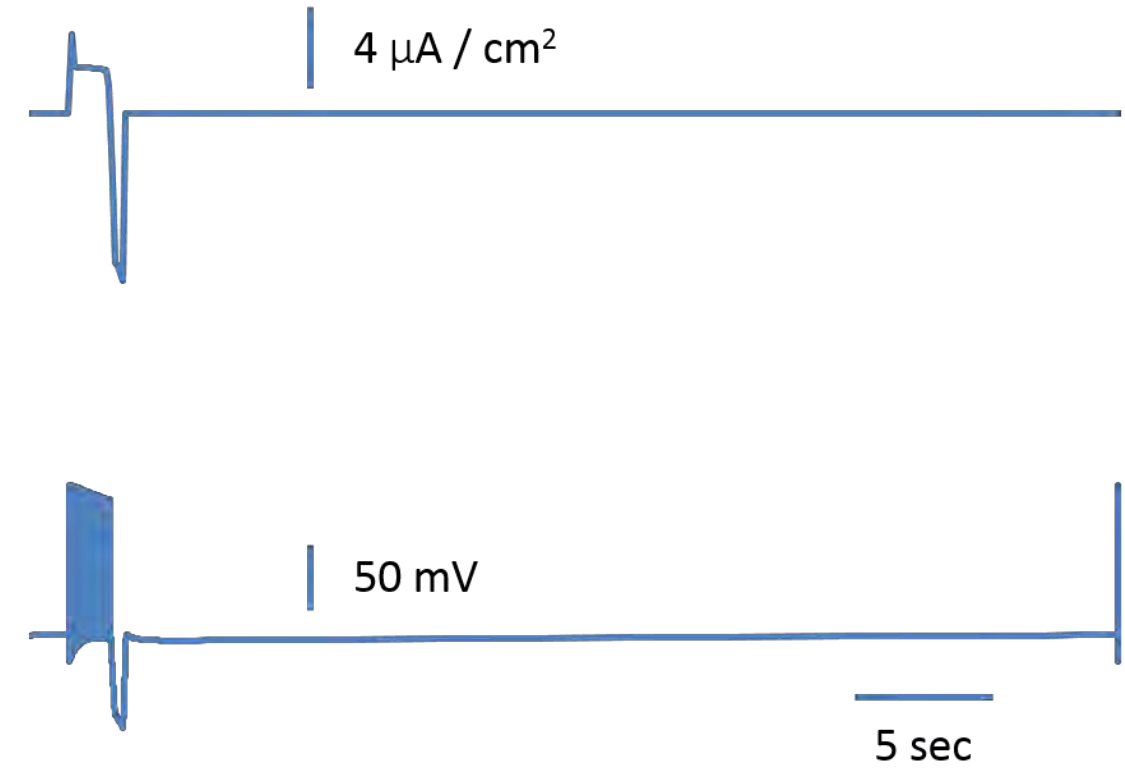


**Figure 7.3: Optimal stimulus waveform of the suppression of a single cell burster.**

## Desynchronization of coupled oscillators

From the perspective of a network of coupled oscillators, we again see a biphasic stimulus that is in antiphase to the system as seen in Figure 7.4. It is interesting to note that the timing of the stimulus as it relates to the phase of the neurons is extremely important. We see that the stimulus leverages the slight differences between the oscillators to time the stimulus just right so that each one responds slightly differently compared to the others, causing desynchronization.

For this model, we also did a rectangular pulse grid search and found that the optimal rectangular waveform had an $L^2$-norm of 154.88 $\mu J/cm^2$ as compared to the optimal determined by EDA which had an $L^2$-norm of 11.80 $\mu J/cm^2$.

## Suppression of systemic bursting

Figure 7.5 shows the optimal stimulus waveform found by EDA for the population-based model of systemic bursting. As we can see, the solution does indeed suppress the seizure state and bring it back to quiescence. Again, we see the biphasic nature of the stimulus in antiphase to the voltage membrane of the pyramidal neurons.

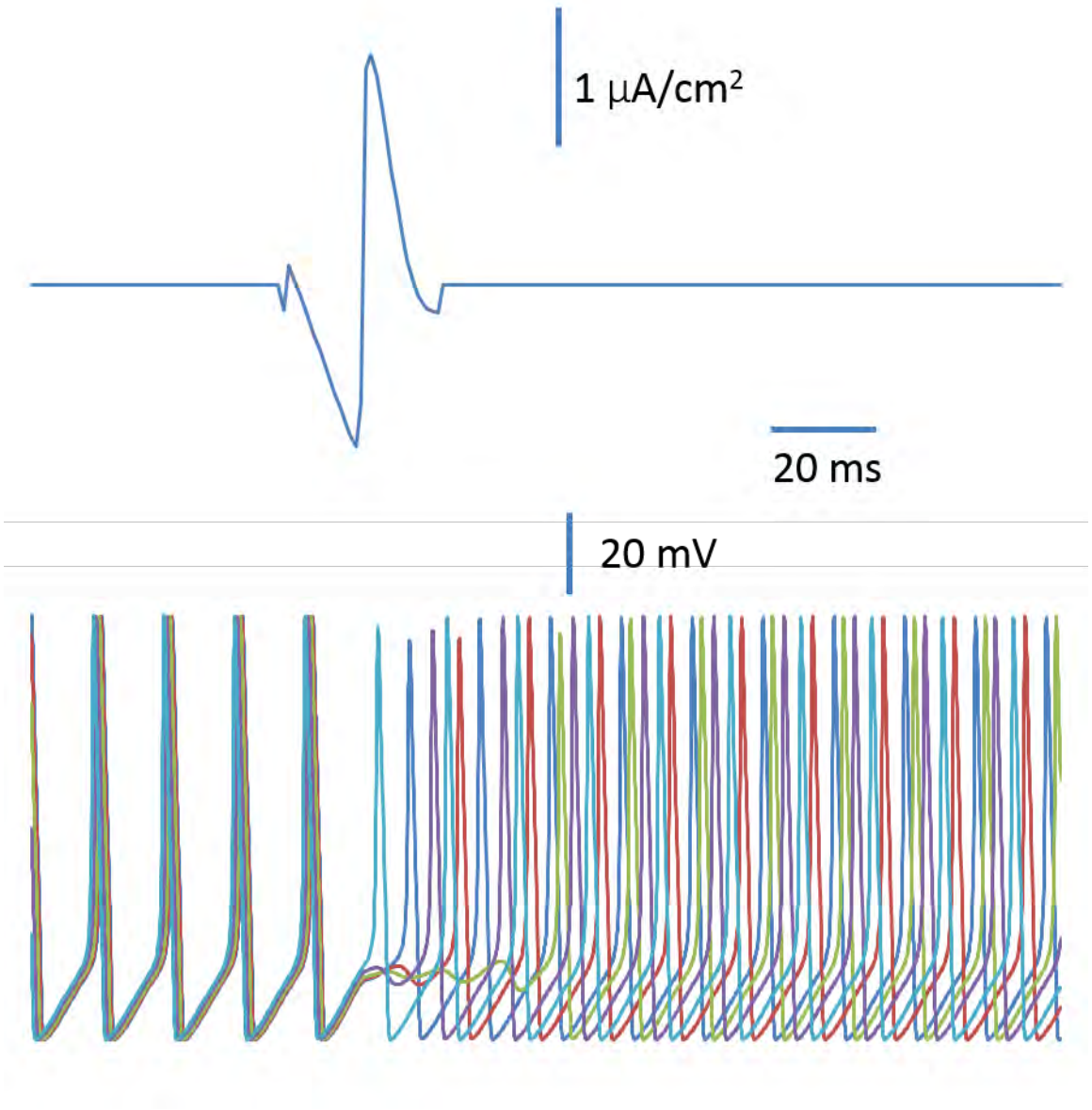


**Figure 7.4: Optimal stimulus waveform for desynchronizing a network of coupled oscillators**

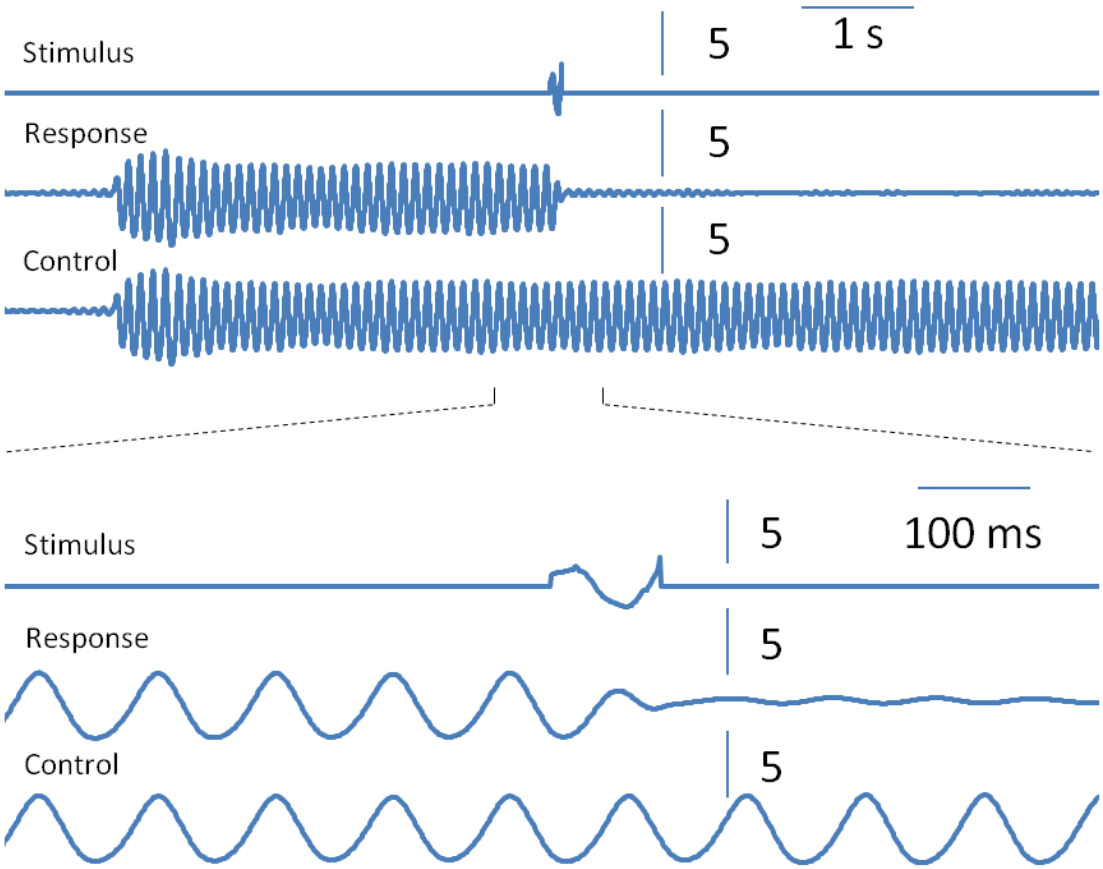


**Figure 7.5: Optimal stimulus waveform for suppressing seizure activity in a systemic population-based model. We show the stimulus, the system's response, and the control response with no stimulus. The bottom set shows a magnification of the transition due to the stimulus.**

# Discussion

## Electrical stimulation for epileptic treatment

Research has shown that 30% of all patients suffering from epilepsy do not see a reduction of seizure activity on anti-epileptic drugs[188]. While surgical procedures are an alternative form of treatment for patients, not all of them are capable of undergoing those procedures[212]. As such, researchers have looked into brain stimulation as a method to help in the suppression of epileptic seizures.

Today, there are three types of brain stimulation that clinicians use to treat epilepsy: vagus nerve stimulation, deep brain stimulation, and responsive neural stimulation. The traditional brain stimulation waveform for these therapies is rectangular. In order to tailor the stimulation for each patient, clinicians adjust a limited set of parameters including pulse duration, pulse frequency, and pulse amplitude to reduce adverse effects and maximize efficacy.

From a computational perspective, there is growing interest in applying non-standard pulsatile stimulation for the treatment of epilepsy. Wilson and Moehlis have applied mathematical techniques and principles to solving computational models of epilepsy to find energetically efficient stimuli[213] to terminate seizure-like bursting behavior to a reduced version of the Cressman model. Tass has examined the use of delayed feedback stimulation as a method to suppress seizures, but he does not examine it from the perspective of energy-efficiency[26].

## Rectangular pulses are not optimal

In this chapter, we have captured the optimal stimulus waveform using EDA on four separate computational models of epilepsy: suppression of single cell repetitive firing, desynchronization of networked coupled oscillators, suppression of systemic bursting and suppression of single cell bursting. We have shown in the first two models that the optimal solution performs much better than optimal rectangular waveforms, thus allowing for potentially large energy reductions in energy consumption. This improvement in energy consumption makes sense from the perspective that the biological world responds more often to signals bandlimited by certain frequencies. From a signals perspective, sharp edges in the stimulus require extremely high frequencies, and thus these frequencies are wasted energy when given to a system that does not respond to such extreme rates of change.

## Optimal stimulus waveforms are short discrete bursts in antiphase to the system

The field of dynamical system theory has studied effect of short discrete pulses in the context of oscillatory systems. Some of the research has focused on examining how these short discrete bursts have changed the phase settings of the system[214], while others have examined how short discrete bursts can be used to desynchronized coupled oscillator networks or completely suppress oscillations [58]. Here in our study, we have seen that in all four models, a short discrete burst of stimuli, when given at the proper time with the proper waveform, can rectify the dysfunctional oscillatory activity that we see during seizure activity.

Furthermore, we also note that for three of the models (suppression of single cell repetitive firing, desynchronization of coupled oscillators and suppression of systemic bursting), the optimal stimulus waveform is in antiphase with the system's internal oscillation, indicating that optimally effective stimuli are phase-dependent. The same stimulus waveform may not work if given at a different phase, or time in the cycle. While the idea that antiphasic stimuli has a suppressive relationship with an oscillatory system is well-known in dynamical systems theory, it is slowly gaining traction in the field of neuroscience as well. Rosenblaum and Pikovsky, as well as others later, demonstrated how synchronization can be controlled via a delayed-feedback stimulus that is slightly out of phase with the system[23,26,215]. More recently, computational studies have found optimal stimuli that are not just slightly out of phase, but in complete antiphase with the system's oscillations[33,43].

Mechanistically, if we examine the suppression of the repetitive firing of the single cell Hodgkin-Huxley, we can observe that the optimal stimulus gives a brief depolarization pulse followed by a larger hyperpolarization pulse. The hyperpolarization pulse when the system would normally be building towards the action potential. We have seen this result occur before in the suppression of repetitive firing in the FitzHugh-Nagumo model as seen in the results from Chapter 3 and 6.

We also see this mechanism of suppressing repetitive firing at the system level bursting. We see that the stimulus given is in antiphase with the system such that it counteracts the oscillations, suppressing the burst.

This concept of stimulating in antiphase is intuitive for most people. As most people will know, one of the gentlest ways to stop somebody on a swing is to apply opposing pressure on the swing to counteract the force by the child. The same principles apply here: the stimulus is a depolarizing and hyperpolarizing force gently stopping the system from continuing on its oscillatory state.

At first glance, the desynchronization of the coupled oscillator network is different from the other two models. What is important to remember here is that the neurons are monostable repetitive firing. It is impossible to suppress these neurons to complete quiescence. When we examine what is happening, we note that the stimulus is in antiphase again with the system as we have seen in the other two models. However, because the system is stable only in the repetitive firing state, the neuron will inevitably return to repetitive firing. In the attempt to suppress the neuron, the stimulus does cause a slight delay in between firing. The slight differences in phase with respect to the timing of the stimulus lead to the dispersion of phase across the different oscillators.

The fact that the stimuli are all in antiphase with the system indicates that the stimulus is phase-dependent, meaning that the stimulus needs to be given at exact times with respect to the system. If the stimulus was in phase with the system, one could hypothesize that the stimulus may actually amplify the seizures, causing further debilitation and other detrimental side effects.

With more research, we may be able to one day develop therapeutic devices that can measure the phase and the frequency of the epileptic dysfunctional oscillations and

calculate the proper parameters for the waveform to minimize detrimental side effects while maximizing efficacy. Furthermore, by using an adaptive algorithm, these devices could potentially even adapt to changes in the phase or frequency of the system due to physiologically changes in the patient.

## Understanding the single cell burster

The single-cell bursting model poses an interesting challenge, as it appears to be an exception to the theme in that the optimal stimulus waveform does not match what we have seen in the other three models. When examining the bursting system, we note that there is a fast wave and a slow wave. The fast wave dictates the firing of the action potentials during the repetitive firing phase and the slow wave dictates whether the cell is in the repetitive firing phase or the quiescent phase of the bursting cycle. From what we can see, our stimulus is on a much slower time scale and thus appears to suppress the bursting behavior by affecting the slow part of the burster.

We also note that while the last action potential in the repetitive firing phase ends much sooner than the control, the stimulus fires as many action potentials as the control. When examining the other state variables, it would seem that the stimulus pushes the system around the limit cycle quicker in order to suppress the repetitive firing portion sooner. From a clinical perspective, this may not be of much benefit to the patient, as it would appear that the stimulus seems to magnify the epileptic seizure first before suppressing it.

We did try running the same algorithm using an extremely small 20-ms stimulus in order to match the fast part of the bursting model. Unfortunately, we were unable to capture

any successful stimuli in our search. This would imply that it is easier to affect the suppression of the bursting behavior by targeting the slow component as opposed to the fast component. Because of computational limitations, we have not yet been able to run a long high-resolution stimulus targeting both fast and slow waves simultaneously in order to determine whether the optimal stimulus waveform can benefit by affecting both components.

While the antiphasic feature does not appear in the single cell burster, it is possible that through exploring higher resolution stimuli with EDA, we will find a stimulus that exists in antiphase to the fast wave and thus confirm the pattern we have seen in the other three models. For that matter, if we were to run a much longer stimulus, we may also see the antiphasic properties appear in relation to the slow wave. From a patient's perspective, a weak, but prolonged stimulus may not be desirable compared to a strong, but short stimulus. It may also be interesting to examine a long stimulus with extremely high resolution to be examine both the fast wave and the slow wave components at the same time. This would be a computational challenge, but it may give us unique insight into how the different components play a role in the suppression of epileptic seizures.

# CHAPTER VIII

# Conclusions and Future Directions

The challenge we set out for ourselves at the beginning of this endeavor was to develop an algorithm that can develop an optimal stimulus waveform for a given behavior without any knowledge of the mechanisms or mathematical models underlying the system. To that end, we have developed the Extrema Distortion Algorithm (EDA). We have validated the EDA by using standards constructed through a robust stochastically-seeded gradient algorithm. We have even demonstrated the application of EDA towards gaining insight into therapeutic design as seen in our study on epilepsy.

This work has just been the first step into this journey. With this foundation, we can branch out in many different directions. In this chapter, we highlight a few of them.

## Application to Real-World Clinical Problems

One of the driving forces behind this research has been to develop goal has been one of the driving forces behind much of the work done. We specifically focused on model-independent algorithms so that one day this work, integrated into therapeutic devices, can find meaningful patient-specific optimal stimuli at the bedside. While two patients may

have the same symptoms, their underlying pathophysiology may be different. The differences in their internal neural wiring may lead one person to respond to specific frequencies and amplitudes differently than another person. What works for one patient may not work for another patient. As such, this algorithm can potentially "learn" what the unique optimal stimulus waveform is for each patient.

Before this vision becomes reality, there are a few more research steps to take. First, all our models and algorithms do not take into account noise inherent to the system. We have treated each system as a noise-free deterministic system, always starting at the same state, where the same input given at the exact same time will always result in the same output. Unfortunately, biological systems are almost never so deterministic, and it is rare for us to be able to give a stimulus under the exact same conditions every time. In order to gauge the impact this has on EDA, we would first need to change our algorithm in such a way that we observe how a system responds to the same stimulus under multiple different noisy conditions. The performance metric will need to change, no longer reflecting if the stimulus successfully reaches the desired outcome, but how often it does.

Once we have a better idea how the algorithm performs in noisy systems, we would probably want to start to test the algorithm on a simple real-life biological system, whether in an animal model or through a non-invasive human procedure. The fundamental set up should not be too complicated as all the algorithm needs is an input and an output. As an example, an easy set up would be to hook up a surface EMG to determine what the optimal stimulus waveform would be to cause a contraction in a

muscle. A surface electrode could be set up to give a small burst of stimulus to the region, and a second electrode can then be used measure muscle. These two electrodes would serve as the input and output to the algorithm. MATLAB has a library that allows it to read from and write to a National Instruments data acquisition (DAQ) system. With this set up, minimal changes would be required to the existing code base.

During this process, it will be important to examine how many stimuli we are giving to the system, and how the system responds over the course of the iterations. For instance, in many of our studies we ran the algorithm for 1,000 iterations, each iteration containing 10 distortions. From a subject's perspective, that would be 10,000 bursts of stimulus given to the patient in one experiment. We may learn that this is too many test stimuli for a patient, and we may need to revise our EDA to be much more efficient.

We have discussed in Chapter 7 a number of different avenues by which to improve EDA's efficiency. First, we currently use a random normal distribution to search neighboring extrema. By constructing a secondary adaptive algorithm, we may be able to adjust the random distribution to more heavily take into account past extrema that we have tested. For instance, if we know that we have jittered an extrema in one direction without successful outcomes, it may be helpful to bias the future distortions away from that direction.

Another technique that we can use is to allow some form of cross learning to occur between the different starting seeds. The algorithm currently works linearly, starting from a single seed and iteratively progressing towards a single optimal solution. We run

this process twenty times on average. If the algorithm was able to collate information across all twenty processes at each iteration, sharing information from one process to another, we may be able to converge more quickly towards an optimal solution.

One of the other problems that may arise in the application of EDA to clinical problems is that we use white-noise generators to develop starting seeds for EDA. We may be unable to accomplish this in a clinical setting. As such, we may instead consider applying an extremely small white-noise process to a known stimulus waveform that works. The small white-noise process would not affect the fundamental shape of the stimulus waveform, but it would add a number of extrema points that EDA can use to begin searching locally for more optimal solutions. The advantage of this method is that it would allow for a sure starting point, removing the need to give large amounts of white-noise stimulation to find successful starting seeds. The disadvantage of this method is that fixing the starting seed may bias the search algorithm, and more optimal solutions may be lost. More research would have to be done in this area to find out what the appropriate trade-offs would be and under what circumstances would one be favorable to the other.

## Modeling Complex and Sensitive Systems

In some clinical applications, there may not be enough time, no matter how efficient our algorithm may be, to learn what the optimal stimulus waveform is. For instance, if a patient is suffering from cardiac arrhythmias, he or she may not have time to sit through a number of test stimuli to find what is optimal. In these cases, running EDA on models

may still be beneficial. First, the insights gained from EDA even on models may prove valuable in providing researchers with new hypotheses on novel stimulus waveforms. Second, in coupling the optimal stimulus waveforms obtained through EDA with real-world experimentation, we may be able to develop stronger and more accurate models. Because the optimal stimulus waveforms often exploit some mechanism to produce optimal solutions, researchers can verify experimentally if these mechanisms are true to real-world systems or if further modifications in state variables or parameters are necessary in their models.

In our exploration into computational biology, we have found many more biological models that can be studied, from cardiology (examining cardioversion or defibrillation from arrhythmias) to respiratory (stimulating respiratory rhythms in patients with apnea) even to pharmacological (studying optimal dosing patterns for drug treatments). Furthermore, it may be interesting to model not just the biological behavior, but also the electrical and physical properties of the electrode in relationship with the system. As some researchers have noted, while we may be able to cut down on the energy consumption of the signal itself, the energetic requirements of the hardware may overwhelm any energy savings on the stimulus waveform itself[183]. Searching for optimality in the entire system from electrode to system may yield further insights into how we can improve the biological response to exogenous stimuli.

# Optimality as a Design Principle in Nature

As reviewed in the introduction of this work, switches exist throughout biology. For a non-therapeutic application, biologists often think about why nature is designed a certain way. There is a hypothesis that the biological world is optimal and efficient, but the metric used to determine optimality might not be apparent. It may be interesting to test out various hypotheses regarding optimality, using EDA, to gain insight into why nature has designed to one versus another specific waveform to control a particular process. Examining biology from the perspective of optimality may yield novel insights into the mechanisms behind various biological behaviors.

# References


1. Cheng HH, Muhlrad PJ, Hoyt M a, Echols H. Cleavage of the cII protein of phage lambda by purified HflA protease: control of the switch between lysis and lysogeny. Proc Natl Acad Sci U S A. 1988;85: 7882–7886.

2. Kim JW, Tchernyshyov I, Semenza GL, Dang C V. HIF-1-mediated expression of pyruvate dehydrogenase kinase: A metabolic switch required for cellular adaptation to hypoxia. Cell Metab. 2006;3: 177–185.

3. Lai K, Robertson MJ, Schaffer D V. The sonic hedgehog signaling system as a bistable genetic switch. Biophys J. 2004;86: 2748–2757.

4. Ferrell JE, Machleder EM. The biochemical basis of an all-or-none cell fate switch in Xenopus oocytes. Science. 1998;280: 895–898.

5. Laurent M, Kellershohn N. Multistability: A major means of differentiation and evolution in biological systems. Trends Biochem Sci. 1999;24: 418–422.

6. Stacey DW. Cyclin D1 serves as a cell cycle regulatory switch in actively proliferating cells. Curr Opin Cell Biol. 2003;15: 158–163.

7. Weinberg R. The retinoblastoma protein and cell cycle control. Cell. 1995;81: 323–330.

8. Zhang D-W, Shao J, Lin J, Zhang N, Lu B-J, Lin S-C, et al. RIP3, an energy metabolism regulator that switches TNF-induced cell death from apoptosis to necrosis. Science. 2009;325: 332–336.

9. Guy HR, Seetharamulu P. Molecular model of the action potential sodium channel. Proc Natl Acad Sci U S A. 1986;83: 508–512.

10. Hodgkin AL, Huxley AF. A quantitative description of membrane current and its application to conduction and excitation in nerve. J Physiol. Blackwell Publishing; 1952;117: 500–544.

11. Hahn PJ, Durand DM. Bistability dynamics in simulations of neural activity in high-extracellular-potassium conditions. J Comput Neurosci. 2001;11: 5–18.

12. Lu J, Sherman D, Devor M, Saper CB. A putative flip-flop switch for control of REM sleep. Nature. 2006;441: 589–594.

13. Saper CB, Fuller PM, Pedersen NP, Lu J, Scammell TE. Sleep state switching. Neuron. 2010;68: 1023–1042.

14. Segel L a, Jäger E, Elias D, Cohen IR. A quantitative model of autoimmune disease and T-cell vaccination: does more mean less? Immunol Today. 1995;16: 80–84.

15. Lown B. Electrical reversion of cardiac arrhythmias. Br Heart J. 1967;29: 469–89.

16. Kouwenhoven WB. The development of the defibrillator. Ann Intern Med. 1969;71: 449–458.

17. Walcott GP, Walker RG, Cates a W, Krassowska W, Smith WM, Ideker RE. Choosing the optimal monophasic and biphasic waveforms for ventricular defibrillation. J Cardiovasc Electrophysiol. 1995;6: 737–50.

18. Qu F, Nikolski VP, Wollenzier BR, Efimov R. Comparison of three biphasic defibrillation waveforms : Gurvich waveform is more efficient. Proc Second Jt EMBS/BMES Conf. 2002; 1439–1440.

19. Keener JP, Panfilov a V. A biophysical model for defibrillation of cardiac tissue. Biophys J. Elsevier; 1996;71: 1335–45.

20. Hammond C, Bergman H, Brown P. Pathological synchronization in Parkinson's disease: networks, models and treatments. Trends Neurosci. 2007;30: 357–64.

21. Lozano AM. Deep brain stimulation for Parkinson's disease. J Neurosurg. 2010;112: 199–203.

22. Rubin JE, Terman D. High frequency stimulation of the subthalamic nucleus eliminates pathological thalamic rhythmicity in a computational model. J Comput Neurosci. 2004;16: 211–235.

23. Hauptmann C, Popovych O, Tass PA. Effectively desynchronizing deep brain stimulation based on a coordinated delayed feedback stimulation via several sites: a computational study. Biol Cybern. 2005;93: 463–470.

24. Loddenkemper T, Pan A. Deep brain stimulation in epilepsy. J Clin Neurophysiol. 2001;116: 217–34.

25. Durand DM, Warman EN. Desynchronization of epileptiform activity by extracellular current pulses in rat hippocampal slices. J Physiol. 1994;480: 527–537.

26. Tass PA. A model of desynchronizing deep brain stimulation with a demand-controlled coordinated reset of neural subpopulations. Biol Cybern. 2003;89: 81–88.

27. Jiménez F, Velasco F, Salin-Pascual R, Hernández J a., Velasco M, Criales JL, et al. A patient with a resistant major depression disorder treated with deep brain stimulation in the inferior thalamic peduncle. Neurosurgery. 2005;57: 585–592.

28. Mayberg HS, Lozano AM, Voon V, McNeely HE, Seminowicz D, Hamani C, et al. Deep brain stimulation for treatment-resistant depression. Neuron. 2005;45: 651–660. doi:10.1016/j.neuron.2005.02.014

29. Aouizerate B, Cuny E, Martin-Guehl C, Guehl D, Amieva H, Benazzouz A, et al. Deep brain stimulation of the ventral caudate nucleus in the treatment of obsessive-compulsive disorder and major depression. Case report. J Neurosurg. 2004;101: 682–686.

30. Lozano AM, Mayberg HS, Giacobbe P, Hamani C, Craddock RC, Kennedy SH. Subcallosal cingulate gyrus deep brain stimulation for treatment-resistant depression. Biol Psychiatry. 2008;64: 461–467.

31. Bloch-Salisbury E, Indic P, Bednarek F, Paydarfar D. Stabilizing immature breathing patterns of preterm infants using stochastic mechanosensory stimulation. J Appl Physiol. 2009;107: 1017–27.

32. Gelfand IM, Fomin SV. Calculus of Variations. Courier Dover Publications; 2000.

33. Forger DB, Paydarfar D, Clay JR. Optimal stimulus shapes for neuronal excitation. PLoS Comput Biol. 2011;7: e1002089.

34. Offner F. Stimulation with minimum power. J Neurophysiol. 1946;9: 387–390.

35. Jezernik S, Morari M. Energy-optimal electrical excitation of nerve fibers. IEEE Trans Biomed Eng. 2005;52: 740–743.

36. Fishler MG. Theoretical predictions of the optimal monophasic and biphasic defibrillation waveshapes. IEEE Trans Biomed Eng. 2000;47: 59–67.

37. Moehlis J, Shea-Brown E, Rabitz H. Optimal inputs for phase models of spiking neurons. J Comput Nonlinear Dyn. 2006;1: 358–367.

38. Danzl P, Nabi A, Moehlis J. Charge-balanced spike timing control for phase models of spiking neurons. Discret Contin Dyn Syst. 2010;28: 1413–1435.

39. Forger DB, Paydarfar D. Starting, stopping, and resetting biological oscillators: in search of optimum perturbations. J Theor Biol. 2004;230: 521–532.

40. Bryson AE, Ho Y-C. Applied Optimal Control. Revised Pr. Hemisphere Publishing Corporation; 1975.

41. Osborne MR. On shooting methods for boundary value problems. J Math Anal Appl. 1969;27: 417–433.

42. Ypma TJ. Historical development of the Newton-Raphson method. SIAM Rev. 1995;37: 531–551.

43. Paydarfar D, Forger DB, Clay JR. Noisy inputs and the induction of on-off switching behavior in a neuronal pacemaker. J Neurophysiol. 2006;96: 3338–48.

44. Kelley HJ. Methods of Gradients. In: Leitmann G, editor. Optimization Techniques. 5th ed. New York, New York: Academic Press, Inc; 1962. pp. 206–254.

45. Golfetto WA, Fernandes S da S. A review of gradient algorithms for numerical computation of optimal trajectories. J Aerosp Technol Manag. 2012;4: 131–143.

46. Aghababa MP, Amrollahi MH, Borjkhani M. Application of GA, PSO, and ACO algorithms to path planning of autonomous underwater vehicles. J Mar Sci Appl. 2012;11: 378–386.

47. Raivo T. Computational Methods for Dynamic Optimization and Pursuit-Evasion Games. Helsinki University of Technology. 2000.

48. Gupta N, Rink R. Optimum control of epidemics. Math Biosci. 1973;18: 383–396.

49. Doležal J. A gradient-type algorithm for the numerical solution of two-player zero-sum differential game problems. Kybernetika. 1978;14: 429–446.

50. Lee ES. Optimization by a gradient technique. Ind Eng Chem Fundam. 1964;3: 373–380.

51. Joshi HR. Optimal control of an HIV immunology model. Optim Control Appl Methods. 2002;23: 199–213.

52. Kirschner D, Lenhart S, Serbin S. Optimal control of the chemotherapy of HIV. J Math Biol. 1997;35: 775–792.

53. Kepler TB, Perelson AS. Somatic hypermutation in B cells: an optimal control treatment. J Theor Biol. 1993;164: 37–64.

54. FitzHugh R. Impulses and physiological states in theoretical models of nerve membrane. Biophys J. 1961;1: 445–466.

55. Nagumo J, Arimoto S, Yoshizawa S. An active pulse transmission line simulating nerve axon. Proc IRE. 1962; 2061–2070.

56. Clay JR, Paydarfar D, Forger DB. A simple modification of the Hodgkin and Huxley equations explains type 3 excitability in squid giant axons. J R Soc Interface. 2008;5: 1421–1428.

57. Paydarfar D, Buerkel DD. Dysrhythmias of the respiratory oscillator. Chaos. 1995;5: 18–29.

58. Winfree AT. The Geometry of Biological Time. 2nd ed. Marsden JE, Sirovich L, Wiggins S, editors. New York, NY: Springer-Verlag; 2001.

59. Glass L. Synchronization and rhythmic processes in physiology. Nature. 2001;410: 277–284.

60. Alon U. An Introduction to Systems Biology - Design Principles of Biological Circuits. Boca Raton, FL: CRC Press, Taylor & Francis Group, LLC; 2006.

61. Kawato M, Suzuki R. Two coupled neural oscillators as a model of the circadian pacemaker. J Theor Biol. 1980;86: 547–575.

62. Aliev RR, Panfilov A V. A simple two-variable model of cardiac excitation. Chaos, Solitons & Fractals. 1996;7: 293–301.

63. Clay JRJ, Forger DD, Paydarfar D. Ionic mechanism underlying optimal stimuli for neuronal excitation: role of Na+ channel inactivation. PLoS One. 2012;7: e45983.

64. Serkh K, Forger DB. Optimal schedules of light exposure for rapidly correcting circadian misalignment. PLoS Comput Biol. 2014;10: e1003523.

65. Dean DA, Forger DB, Klerman EB. Taking the lag out of jet lag through model-based schedule design. PLoS Comput Biol. 2009;5: e1000418.

66. Betts JJT. Survey of numerical methods for trajectory optimization. J Guid Control Dyn. 1998;21: 193–207.

67. Rao A. A survey of numerical methods for optimal control. Adv Astronaut Sci. 2010;135: 497–528.

68. Nabi A, Moehlis J. Time optimal control of spiking neurons. J Math Biol. 2012;64: 981–1004.

69. Tahayori B, Dokos S. Optimal stimulus current waveshape for a Hodgkin-Huxley model neuron. 34th Annu Int Conf IEEE EBS. 2012; 4627–4630.

70. Izhikevich EM. Neural excitability, spiking and bursting. Int J Bifurc Chaos. 2000;10: 1171–1266.

71. Izhikevich EM. Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting. Sejnowski TJ, Poggio TA, editors. Cambridge, Massachusetts: MIT Press; 2007.

72. Barnett W, O'Brien G, Cymbalyuk G. Bistability of silence and seizure-like bursting. J Neurosci Methods. Elsevier B.V.; 2013;220: 179–89.

73. Rinzel J, Ermentrout GB. Analysis of neural excitability and oscillations. Methods in Neuronal Modeling. 2nd ed. 1998. pp. 251–292.

74. Butera RJ, Rinzel J, Smith JC. Models of respiratory rhythm generation in the pre-Bötzinger complex. I. Bursting pacemaker neurons. J Neurophysiol. 1999;82: 382–97.

75. Chen Y, Wang J, Wei X, Deng B, Che Y. Particle swarm optimization of periodic deep brain stimulation waveforms. Proc 30th Chinese Control Conf. 2011; 754–757.

76. Feng X, Greenwald B, Rabitz H. Toward closed-loop optimization of deep brain stimulation for Parkinson's disease: concepts and lessons from a computational model. J Neural Eng. 2007;4: L14–L21.

77. Howalski CH, Silva GA da, Poppi RJ, Godoy HT, Augusto F. Neuro-genetic multioptimization of the determination of polychlorinated biphenyl congeners in human milk by headspace solid phase microextraction coupled to gas chromatography with electron capture detection. Anal Chim Acta. 2007;585: 66–75.

78. Schiff SJ. Towards model-based control of Parkinson's disease. Philos Trans R Soc A. 2010;368: 2269–2308.

79. Iasemidis LD. Epileptic seizure prediction and control. IEEE Trans Biomed Eng. 2003;50: 549–558.

80. Lian J, Bikson M, Sciortino C, Stacey WC, Durand DM. Local suppression of epileptiform activity by electrical stimulation in rat hippocampus in vitro. J Physiol. 2003;547: 427–434.

81. Sunderam S, Gluckman B, Reato D, Bikson M. Toward rational design of electrical stimulation strategies for epilepsy control. Epilepsy Behav. Elsevier Inc.; 2010;17: 6–22.

82. Machens CK, Gollisch T, Kolesnikova O, Herz AVM. Testing the efficiency of sensory coding with optimal stimulus ensembles. Neuron. 2005;47: 447–456.

83. Koelling ME, Nykamp DQ. Searching for optimal stimuli: ascending a neuron's response function. J Comput Neurosci. 2012;33: 449–473.

84. Watson A, Barlow H, Robson J. What does the eye see best? Nature. 1983;302: 419–422.

85. Attwell D, Laughlin SB. An energy budget for signaling in the grey matter of the brain. J Cereb Blood Flow Metab. 2001;21: 1133–45.

86. Sengupta B, Stemmler M, Laughlin SB, Niven JE. Action potential energy efficiency varies among neuron types in vertebrates and invertebrates. PLoS Comput Biol. 2010;6: e1000840.

87. Torrealdea F, D'Anjou A, Graña M, Sarasola C. Energy aspects of the synchronization of model neurons. Phys Rev E. 2006;74: 011905.

88. Alle H, Roth A, Geiger JRP. Energy-efficient action potentials in hippocampal mossy fibers. Science (80- ). 2009;325: 1405–1408.

89. Schwartz O, Pillow JW, Rust NC, Simoncelli EP. Spike-triggered neural characterization. J Vis. 2006;6: 484–507.

90. Cheney P. Response of rubromotoneuronal cells identified by spike-triggered averaging of EMG activity in awake monkeys. Neurosci Lett. 1980;17: 137–141.

91. Pillow J, Simoncelli E. Dimensionality reduction in neural models: an information-theoretic generalization of spike-triggered average and covariance analysis. J Vis. 2006;6: 414–428.

92. Rotger D, Radeva P, Bruining N. Automatic detection of bioabsorbable coronary stents in IVUS images using a cascade of classifiers. IEEE Trans Inf Technol Biomed. 2010;14: 535–7.

93. Klein S, Staring M, Andersson P, Pluim JPW. Preconditioned stochastic gradient descent optimisation for monomodal image registration. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics). 2011;6892 LNCS: 549–56.

94. Zommer S, Ribak E, Lipson S, Adler J. Simulated annealing in ocular adaptive optics. Opt Lett. 2006;31: 939–941.

95. Patel R, Longini IM, Halloran ME. Finding optimal vaccination strategies for pandemic influenza using genetic algorithms. J Theor Biol. 2005;234: 201–12.

96. Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. Science (80- ). 1983;220: 671–680.

97. Crozier S, Roffmann WU, Luescher K, Snape-Jenkinson C, Forbes LK, Doddrell DM. An "openable," high-strength gradient set for orthopedic MRI. J Magn Reson. 1999;139: 81–9.

98. Ma Y, Lee L, Keshet O, Keall P, Xing L. Four-dimensional inverse treatment planning with inclusion of implanted fiducials in IMRT segmented fields. Med Phys. 2009;36: 2215.

99. Li JG, Boyer a L, Xing L. Clinical implementation of wedge filter optimization in three-dimensional radiotherapy treatment planning. Radiother Oncol. 1999;53: 257–64.

100. Pugachev A, Xing L. Incorporating prior knowledge into beam orientation optimization in IMRT. Int J Radiat Oncol Biol Phys. 2002;54: 1565–74.

101. Shepard DM, Earl M a., Li X a., Naqvi S, Yu C. Direct aperture optimization: A turnkey solution for step-and-shoot IMRT. Med Phys. 2002;29: 1007.

102. Gilbert GE, Baleja JD. Membrane-binding peptide from the C2 domain of factor VIII forms an amphipathic structure as determined by NMR spectroscopy. Biochemistry. 1995;34: 3022–31.

103. Jao CC, Hegde BG, Chen J, Haworth IS, Langen R. Structure of membrane-bound alpha-synuclein from site-directed spin labeling and computational refinement. Proc Natl Acad Sci U S A. 2008;105: 19666–19671.

104. Jones MK, Gu F, Catte A, Li L, Segrest JP. “Sticky” and “promiscuous”, the yin and yang of apolipoprotein A-I termini in discoidal high-density lipoproteins: a combined computational-experimental approach. Biochemistry. 2011;50: 2249–63.

105. Katsamba PS, Bayramyan M, Haworth IS, Myszka DG, Laird-Offringa IA. Complex role of the beta 2-beta 3 loop in the interaction of U1A with U1 hairpin II RNA. J Biol Chem. 2002;277: 33267–74.

106. Lu J, Hall KB. Tertiary structure of RBD2 and backbone dynamics of RBD1 and RBD2 of the human U1A protein determined by NMR spectroscopy. Biochemistry. 1997;36: 10393–405.

107. Metcalf DG, Law PB, DeGrado WF. Mutagenesis data in the automated prediction of transmembrane helix dimers. Proteins Struct Funct Genet. 2007;67: 375–384.

108. Narula SS, Brouwer M, Hua Y, Armitage IM. Three-dimensional solution structure of Callinectes sapidus metallothionein-1 determined by homonuclear and heteronuclear magnetic resonance spectroscopy. Biochemistry. 1995;34: 620–631.

109. Ruan K, So S, Zheng W, Wu J, Li D, Kung J. Solution structure and topology of the N-terminal membrane anchor domain of a microsomal cytochrome P450: prostaglandin I2 synthase. Society. 2002;728: 721–728.

110. Tashiro M, Tejero R, Zimmerman DE, Celda B, Nilsson B, Montelione GT. High-resolution solution NMR structure of the Z domain of staphylococcal protein A. J Mol Biol. 1997;272: 573–90.

111. Tregear GW, Bathgate R a, Layfield S, Ferraro T, Gundlach A, Ma S, et al. The chemistry and biology of human relaxin-3. Ann N Y Acad Sci. 2005;1041: 40–6.

112. Held K, Rota Kops E, Krause BJ, Wells WM, Kikinis R, Müller-Gärtner HW. Markov random field segmentation of brain MR images. IEEE Trans Med Imaging. 1997;16: 878–86.

113. Loeckx D, Maes F, Vandermeulen D, Suetens P. Temporal subtraction of thorax CR images using a statistical deformation model. IEEE Trans Med Imaging. 2003;22: 1490–1504.

114. Yaqub M, Boellaard R, Kropholler MA, Lammertsma AA. Optimization algorithms and weighting factors for analysis of dynamic PET studies. Phys Med Biol. 2006;51: 4217–32.

115. Yue Y, Tagare H, Madsen E, Frank G. Evaluation of a cardiac ultrasound segmentation algorithm using a phantom. Med Image Comput. 2008;11: 101–109.

116. Weeks DE, Lange K. Preliminary ranking procedures for multilocus ordering. Genomics. 1987;1: 236–242.

117. Ji S, Zeng Y, Wu P, Lee EJD. Sampling schedule design towards optimal drug monitoring for individualizing therapy. Comput Methods Programs Biomed. 2005;80: 57–63.

118. Kubicky CD, Yeh BM, Lessard E, Joe BN. Inverse planning simulated annealing for magnetic resonance imaging-based intracavitary high-dose-rate brachytherapy for cervical cancer. Brachytherapy. 2008;7: 242–247.

119. Meyer J, Phillips MH, Cho PS, Kalet I, Doctor JN. Application of influence diagrams to prostate intensity-modulated radiation therapy plan selection. Phys Med Biol. 2004;49: 1637–1653.

120. Sahgal A, Jabbari S, Chen J, Pickett B, Roach M, Weinberg V, et al. Comparison of dosimetric and biologic effective dose parameters for prostate and urethra using 131 Cs and 125 I for prostate permanent implant brachytherapy. Int J Radiat Oncol Biol Phys. 2008;72: 247–54.

121. Zhang H, Meyer R, Wu J. A two-stage sequential linear programming approach to IMRT dose optimization. Phys Med Biol. 2010;55: 883–902.

122. Hastings W. Monte Carlo sampling methods using Markov chains and their applications. Biometrika. 1970;57: 97–109.

123. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. Equation of state calculations by fast computing machines. J Chem Phys. 1953;21: 1087–1092.

124. Mitchell M. An Introduction to Genetic Algorithms. Computers Mathematics with Applications. MIT Press; 1996. p. 133. doi:10.1016/S0898-1221(96)90227-8

125. Holland JH. Adaptation in Natural and Artificial Systems. MIT Press; 1992.

126. Forrest S. Genetic algorithms: Principles of natural selection applied to computation. Science (80- ). American Association for the Advancement of Science; 1993;261: 872–878.

127. Abdoli M, Ay MR, Ahmadian A, Dierckx RAJO, Zaidi H. Reduction of dental filling metallic artifacts in CT-based attenuation correction of PET data using weighted virtual sinograms optimized by a genetic algorithm. Med Phys. 2010;37: 6166.

128. Angeletti C, Harvey NR, Khomitch V, Fischer AH, Levenson RM, Rimm DL. Detection of malignancy in cytology specimens using spectral-spatial analysis. Lab Invest. 2005;85: 1555–64.

129. Karmali F, Shelhamer M. Compensating for camera translation in video eye-movement recordings by tracking a representative landmark selected automatically by a genetic algorithm. J Neurosci Methods. 2009;176: 157–65.

130. Lee Y, Hara T, Fujita H, Itoh S, Ishigaki T. Automated detection of pulmonary nodules in helical CT images based on an improved template-matching technique. IEEE Trans Med Imaging. 2001;20: 595–604. doi:10.1109/42.932744

131. Munteanu C, Morales FC, Ruiz-Alzola J. Speckle reduction through interactive evolution of a general order statistics filter for clinical ultrasound imaging. IEEE Trans Biomed Eng. 2008;55: 365–9. doi:10.1109/TBME.2007.897833

132. Qian W, Sankar R, Song X, Sun X. Standardization for image characteristics in telemammography using genetic and nonlinear algorithms. Comput Biol Med. 2005;35: 183–196.

133. Tohka J, Krestyannikov E, Dinov ID, Graham AM, Shattuck DW, Ruotsalainen U, et al. Genetic algorithms for finite mixture model based voxel classification in neuroimaging. IEEE Trans Med Imaging. 2007;26: 696–711.

134. Yang F, Jiang T. Cell image segmentation with kernel-based dynamic clustering and an ellipsoidal cell shape model. J Biomed Inform. 2001;34: 67–73.

135. Boiculese L. Genetic algorithm in the control optimization. Rev Medico-Chirurgicala A Soc Medici Si Nat Iasi. 1999;103: 176–81.

136. Shelhamer M. Use of a genetic algorithm for the analysis of eye movements from the linear vestibulo-ocular reflex. Ann Biomed Eng. 2001;29: 510–522.

137. Yu M, Manoogian S, Duma SM, Stitzel JD. Finite element modeling of human placental tissue. 53rd AAAM Annual Conference, Annals of Advances in Automotive Medicine. 2009. pp. 257–70.

138. Pardi F, Lewis CM, Whittaker JC. SNP selection for association studies: maximizing power across SNP choice and study size. Ann Hum Genet. 2005;69: 733–46.

139. Smigrodzki R, Goertzel B, Pennachin C. Genetic algorithm for analysis of mutations in Parkinson's disease. Artif Intell Med. 2005;35: 227–241.

140. Du J, Yang S, Lin X, Bu L, Nan Y, Huo S, et al. Use of anchorchip-time-of-flight spectrometry technology to screen tumor biomarker proteins in serum for small cell lung cancer. Diagn Pathol. 2010;5: 60.

141. Elofsson A, Fischer D, Rice D, Grand S Le. A study of combined structure/sequence profiles. Fold Des. 1996;1: 451–461.

142. Lin W-T, Tsai C-C, Chen C-Y, Lee W-J, Su C-C, Wu Y-J. Proteomic analysis of peritoneal dialysate fluid in patients with dialysis-related peritonitis. Ren Fail. 2008;30: 772–7.

143. Ul Haq Z, Uddin R, Wai LK, Wadood A, Lajis NH. Docking and 3D-QSAR modeling of cyclin-dependent kinase 5/p25 inhibitors. J Mol Model. 2011;17: 1149–61.

144. Yan D, White MM. Spatial orientation of the antagonist granisetron in the ligand-binding site of the 5-HT3 receptor. Mol Pharmacol. 2005;68: 365–371.

145. Dybowski R, Gant V, Weller P, Chang R. Prediction of outcome in critically ill patients using artificial neural network synthesised by genetic algorithm. Lancet. 1996; 1–10.

146. Holmes J, Durbin D. The learning classifier system: an evolutionary computation approach to knowledge discovery in epidemiologic surveillance. Artif Intell Med. 2000;

147. Hsieh C-H, Liang K-H, Hung Y-J, Huang L-C, Pei D, Liao Y-T, et al. Analysis of epistasis for diabetic nephropathy among type 2 diabetic patients. Hum Mol Genet. 2006;15: 2701–8.

148. Jaimes F, Farbiarz J, Alvarez D, Martínez C. Comparison between logistic regression and neural networks to predict death in patients with suspected sepsis in the emergency room. Crit Care. 2005;9: R150–6.

149. Lim CW, Kirikoshi T. Understanding the effects of pharmaceutical promotion: a neural network approach guided by genetic algorithm-partial least squares. Health Care Manag Sci. 2008;11: 359–372.

150. Nieto P, Malone JB, Bavia ME. Ecological niche modeling for visceral leishmaniasis in the state of Bahia, Brazil, using genetic algorithm for rule-set prediction and growing degree day-water budget analysis. Geospat Health. 2006;1: 115–26.

151. Tadesse Argaw A, Shannon EJ, Assefa A, Mikru FS, Mariam BK, Malone JB. A geospatial risk assessment model for leprosy in Ethiopia based on environmental thermal-hydrological regime analysis. Geospat Health. 2006;1: 105–113.

152. Heckerling PS, Gerber BS, Tape TG, Wigton RS. Selection of predictor variables for pneumonia using neural networks and genetic algorithms. Methods Inf Med. 2005;44: 89–97.

153. Heckerling PS, Gerber BS, Tape TG, Wigton RS. Use of genetic algorithms for neural networks to predict community-acquired pneumonia. Artificial intelligence in medicine. 2004. pp. 71–84.

154. Jefferson M, Burlinson S. Clinical features of dementia associated with apolipoprotein ϵ4: discrimination with a neural network genetic algorithm. Int J Geriatr Psychiatry. 2001;16: 77–81.

155. Jefferson MF, Pendleton N, Lucas SB, Horan MA. Comparison of a genetic algorithm neural network with logistic regression for predicting outcome after surgery for patients with nonsmall cell lung carcinoma. Cancer. 1997;79: 1338–42.

156. Kline JA, Novobilski AJ, Kabrhel C, Richman PB, Courtney DM. Derivation and validation of a Bayesian network to predict pretest probability of venous thromboembolism. Ann Emerg Med. 2005;45: 282–90.

157. Seli E, Sakkas D, Scott R, Kwok SC, Rosendahl SM, Burns DH. Noninvasive metabolomic profiling of embryo culture media using Raman and near-infrared spectroscopy correlates with reproductive potential of embryos in women undergoing in vitro fertilization. Fertil Steril. 2007;88: 1350–7.

158. Vinterbo S, Ohno-Machado L. A genetic algorithm approach to multi-disorder diagnosis. Artif Intell Med. 2000;18: 117–32.

159. Ying H, Lin F, MacArthur RD, Cohn JA, Barth-Jones DC, Ye H, et al. A self-learning fuzzy discrete event system for HIV/AIDS treatment regimen selection. IEEE Trans Syst Man Cybern - Part B Cybern. 2007;37: 966–79.

160. Cotrutz C, Xing L. Segment-based dose optimization using a genetic algorithm. Phys Med Biol. 2003;48: 2987–98.

161. Williams G, Fisher B. Design of biplanar gradient coils for magnetic resonance imaging of the human torso and limbs. Magn Reson Imaging. 1999;17: 739–754.

162. Cho PS, Taira RK, Kangarloo H. Syntactic parsing of medical reports using evolutionary optimization. AMIA 2005 Symposium Proceedings. 2005. p. 920.

163. Mao Y, Wu Z, Tian W, Jiang X, Cheung WK. Dynamic sub-ontology evolution for traditional Chinese medicine web ontology. J Biomed Inform. 2008;41: 790–805.

164. Podgorelec V, Kokol P. Genetic algorithm based system for patient scheduling in highly constrained situations. J Med Syst. 1997;21: 417–27.

165. Sasaki S, Comber AJ, Suzuki H, Brunsdon C. Using genetic algorithms to optimise current and future health planning - the example of ambulance locations. Int J Health Geogr. 2010;9: 4.

166. Wongsarnpigoon A, Grill W. Energy-efficient waveform shapes for neural stimulation revealed with a genetic algorithm. J Neural Eng. IOP Publishing; 2010;7: 1–20.

167. McIntyre CC, Richardson AG, Grill WM. Modeling the excitability of mammalian nerve fibers: influence of afterpotentials on the recovery cycle. J Neurophysiol. 2002;87: 995–1006.

168. George EI, Mcculloch RE. Approaches for bayesian variable selection. Stat Sin. 1997;7: 339–373.

169. Suh YJ, Ye KQ, Mendell NR. A method for evaluating the results of Bayesian model selection: application to linkage analyses of attributes determined by two or more genes. Hum Hered. 2003;55: 147–52.

170. Kim K-W, Suh Y-J, Park W-Y, Jhoo J-H, Lee D-Y, Youn J-C, et al. Choline acetyltransferase G +4 A polymorphism confers a risk for Alzheimer's disease in concert with Apolipoprotein E epsilon4. Neurosci Lett. 2004;366: 182–6.

171. Spall JC. An overview of the simultaneous perturbation method for efficient optimization. Johns Hopkins APL Tech Dig. 1998;19: 482–492.

172. Spall JC. Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control. John Wiley and Sons; 2003.

173. Bellman RE. Dynamic Programming. Courier Dover Publicans; 2003.

174. Mainen ZF, Sejnowski TJ. Reliability of spike timing in neocortical neurons. Science (80- ). 1995;268: 1503–1506.

175. Bryant H, Segundo J. Spike initiation by transmembrane current: a white-noise analysis. J Physiol. 1976;260: 279–314.

176. Dick T, Berger A. Axonal projections of single bulbospinal inspiratory neurons revealed by spike-triggered averaging and antidromic activation. J Neurophysiol. 1985;53: 1590–1603.

177. Lankheet MJM, Klink PC, Borghuis BG, Noest AJ. Spike-interval triggered averaging reveals a quasi-periodic spiking alternative for stochastic resonance in catfish electroreceptors. PLoS One. 2012;7: e32786.

178. Chang J, Paydarfar D. Switching neuronal state: optimal stimuli revealed using a stochastically-seeded gradient algorithm. J Comput Neurosci. 2014;

179. Gardner TS, Cantor CR, Collins JJ. Construction of a genetic toggle switch in Escherichia coli. Nature. 2000;403: 339–42.

180. Vemulapalli PK, Monga V, Brennan SN. Optimally robust extrema filters for time series data. Am Control Conf. 2012; 2189–2195.

181. Lindeberg T. Image matching using generalized scale-space interest points. Proc Int Conf Scale Sp Var Methods Comput Vis. 2013;7893: 335–367.

182. Witkin AP. Scale-space filtering: A new approach to multi-scale description. Acoust Speech Signal Process IEEE Int Conf ICASSP. 1984;9: 150–153.

183. Foutz T, McIntyre C. Evaluation of novel stimulus waveforms for deep brain stimulation. J Neural Eng. 2010;7: 1–22.

184. Jones DR, Schonlau M, Welch WJ. Efficient global optimization of expensive black-box functions. J Glob Optim. 1998;13: 455–492.

185. Kobau R, Luo Y, Zack M, Helmers S, Thurman D. Epilepsy in adults and access to care - United States, 2012. Morb Mortal Wkly Rep. 2012;61: 909–913.

186. Russ SA, Larson K, Halfon N. A National Profile of Childhood Epilepsy and Seizure Disorder. Pediatrics. 2012;129: 256–264.

187. Clark JY, Thompson IM, Optenberg SA, Yoon D, Frick KD, Carr DA, et al. Economic impact of epilepsy in the United States. Epilepsia. 2009;50: 2186–2191.

188. Kwan P, Brodie MJ. Early identification of refractory epilepsy. N Engl J Med. 2000;342: 314–9.

189. Fisher R, Salanova V, Witt T, Worth R, Henry T, Gross R, et al. Electrical stimulation of the anterior nucleus of thalamus for treatment of refractory epilepsy. Epilepsia. 2010;51: 899–908.

190. Morrell M. Brain stimulation for epilepsy: can scheduled or responsive neurostimulation stop seizures? Curr Opin Neurol. 2006;19: 164–8.

191. Salanova V, Worth R. Neurostimulators in epilepsy. Curr Neurol Neurosci Rep. 2007;7: 315–319.

192. Sun FT, Morrell MJ, Wharen RE. Responsive cortical stimulation for the treatment of epilepsy. Neurotherapeutics. 2008;5: 68–74.

193. Morrell MJ. Responsive cortical stimulation for the treatment of medically intractable partial epilepsy. Neurology. 2011;77: 1295–1304.

194. Kossoff EH, Ritzl EK, Politsky JM, Murro AM, Smith JR, Duckrow RB, et al. Effect of an external responsive neurostimulator on seizures and electrographic discharges during subdural electrode monitoring. Epilepsia. 2004;45: 1560–7.

195. Penfield W, Jasper H. Epilepsy and the Functional Anatomy of the Human Brain. Oxford, England: Little, Brown & Co; 1954.

196. Vonck K, Van Laere K, Dedeurwaerdere S, Caemaert J, De Reuck J, Boon P. The mechanism of action of vagus nerve stimulation for refractory epilepsy: the current status. J Clin Neurophysiol. 2001;18: 394–401.

197. Ben-Menachem E. Vagus-nerve stimulation for the treatment of epilepsy. Lancet Neurol. 2002;1: 477–482.

198. Morris GL, Mueller WM. Long-term treatment with vagus nerve stimulation in patients with refractory epilepsy. Neurology. 1999. pp. 1731–1731.

199. Boon P, Vonck K, De Herdt V, Van Dycke A, Goethals M, Goossens L, et al. Deep brain stimulation in patients with refractory temporal lobe epilepsy. Epilepsia. 2007;48: 1551–1560.

200. Lilly JC, Hughes JR, Alvord EC, Galkin TW. Brief, noninjurious electric waveform for stimulation of the brain. Science (80- ). 1955;121: 468–469.

201. Lilly JC, Austin GM, Chambers WW. Threshold movements produced by excitation of cerebral cortex and efferent fibers with some parametric regions of rectangular current pulses: (cats and monkeys). J Neurophysiol. 1952;15: 319–341.

202. Lossin C, Wang DW, Rhodes TH, Vanoye CG, George AL. Molecular basis of an inherited epilepsy. Neuron. 2002;34: 877–884.

203. Cressman JR, Ullah G, Ziburkus J, Schiff SJ, Barreto E. The influence of sodium and potassium dynamics on excitability, seizures, and the stability of persistent states: I. Single neuron dynamics. J Comput Neurosci. 2009;26: 159–170.

204. Jirsa VK, Stacey WC, Quilichini PP, Ivanov AI, Bernard C. On the nature of seizure dynamics. Brain. 2014; 2210–2230.

205. Truccolo W, Ahmed OJ, Harrison MT, Eskandar EN, Cosgrove GR, Madsen JR, et al. Neuronal ensemble synchrony during human focal seizures. J Neurosci. 2014;34: 9927–44.

206. Jiruska P, de Curtis M, Jefferys JGR, Schevon C a, Schiff SJ, Schindler K. Synchronization and desynchronization in epilepsy: controversies and hypotheses. J Physiol. 2013;591: 787–97.

207. Jefferys JG, Haas HL. Synchronized bursting of CA1 hippocampal pyramidal cells in the absence of synaptic transmission. Nature. 1982. pp. 448–450.

208. Jensen MS, Yaari Y. Role of intrinsic burst firing, potassium accumulation, and electrical coupling in the elevated potassium model of hippocampal epilepsy. J Neurophysiol. 1997;77: 1224–1233.

209. Jaseja H. EEG-desynchronization as the major mechanism of anti-epileptic action of vagal nerve stimulation in patients with intractable seizures: Clinical neurophysiological evidence. Med Hypotheses. Elsevier Ltd; 2010;74: 855–856.

210. Schiff SJ, Ullah G. Models of Epilepsy. Scholarpedia. 2009;4: 1409.

211. Suffczynski P, Kalitzin S, Lopes Da Silva FH. Dynamics of non-convulsive epileptic phenomena modeled by a bistable neuronal network. Neuroscience. 2004;126: 467–84.

212. Wiebe S, Blume W, Girvin J, Eliasziw M. A randomized, controlled trial of surgery for temporal-lobe epilepsy. N Engl J Med. 2001;345: 311–318.

213. Wilson D, Moehlis J. A Hamilton-Jacobi-Bellman approach for termination of seizure-like bursting. J Comput Neurosci. 2014; 345–355.

214. Glass L, Mackey MC. From Clocks to Chaos: The Rhythms of Life. Princeton University Press; 1988. p. 248.

215. Rosenblum MG, Pikovsky AS. Controlling synchronization in an ensemble of globally coupled oscillators. Phys Rev Lett. 2004;92: 114102.