J Comput Neurosci (2014) 37:569–582
DOI 10.1007/s10827-014-0525-5

Switching neuronal state: optimal stimuli revealed using a stochastically-seeded gradient algorithm

Joshua Chang · David Paydarfar

Received: 26 February 2014 / Revised: 13 August 2014 / Accepted: 15 August 2014 / Published online: 22 August 2014
© Springer Science+Business Media New York 2014

Abstract Inducing a switch in neuronal state using energy optimal stimuli is relevant to a variety of problems in neuroscience. Analytical techniques from optimal control theory can identify such stimuli; however, solutions to the optimization problem using indirect variational approaches can be elusive in models that describe neuronal behavior. Here we develop and apply a direct gradient-based optimization algorithm to find stimulus waveforms that elicit a change in neuronal state while minimizing energy usage. We analyze standard models of neuronal behavior, the Hodgkin-Huxley and FitzHugh-Nagumo models, to show that the gradient-based algorithm: 1) enables automated exploration of a wide solution space, using stochastically generated initial waveforms that converge to multiple locally optimal solutions; and 2) finds optimal stimulus waveforms that achieve a physiological outcome condition, without a priori knowledge of the optimal terminal condition of all state variables. Analysis of biological systems using stochastically-seeded gradient methods can reveal salient dynamical mechanisms underlying the optimal control of system behavior. The gradient algorithm may also have practical applications in future work, for example, finding energy optimal waveforms for therapeutic neural stimulation that minimizes power usage and diminishes off-target effects and damage to neighboring tissue.

Action Editor: J. Rinzel

Electronic supplementary material The online version of this article (doi:10.1007/s10827-014-0525-5) contains supplementary material, which is available to authorized users.

J. Chang (✉) · D. Paydarfar
Department of Neurology, University of Massachusetts Medical School, Worcester, MA, USA
e-mail: joshua.chang@umassmed.edu

D. Paydarfar
Wyss Institute for Biologically Inspired Engineering, Harvard University, Boston, MA, USA

Keywords Computational model · Gradient algorithm · Optimal stimulation · Single-cell neuron

1 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 (Lozano 2010) and epileptic seizures (Loddenkemper and Pan 2001). 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 (Gelfand and Fomin 2000), in which an optimal functional is derived and solved as a boundary-value problem, using the shooting method (Osborne 1969) or the Newton–Raphson method (Ypma 1995). These techniques, which have been applied to neuronal models (Forger et al. 2011; Forger and Paydarfar 2004), 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 (Forger et al. 2011).

Gradient-based optimization methods (Bryson and Ho 1975; Kelley 1962) 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 (Aghababa et al. 2012; Golfetto et al. 2012; Raivo 2000), epidemiology (Gupta and Rink 1973), game theory (Doležal 1978), kinetics (Lee 1964) and immunology (Joshi 2002; Kepler and Perelson 1993; Kirschner et al. 1997).

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 (Hodgkin and Huxley 1952) or induce or suppress repetitive firing in a bistable neuron (FitzHugh 1961; Nagumo et al. 1962). 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 paper proceeds as follows. In Section 2, we outline the gradient-based algorithm as well as its implementation for both the Hodgkin-Huxley and the Fitzhugh-Nagumo models. Online Resources 1 and 2 provide a detailed mathematical basis for the algorithm. Section 3 describes the results, highlighting the importance of broad exploration of the solution space using stochastically generated seeds. Section 4 discusses the advantages and limitations of the gradient algorithm and compares it to other previously used techniques.

2 Methods

2.1 First-order gradient algorithm

The derivation and theoretical framework of the first-order gradient optimization problem is detailed in Online Resource 1. Here we provide a step-by-step outline of how the algorithm is implemented. 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. This is done 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 (1975).

Given a nonlinear system of equations:

$$\dot{x} = f[x(t), u(t), t] \text{ for } t_o \le t \le t_f$$

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:

$$J = \int_{t_o}^{t_f} L[x(t), u(t), t]\, dt,$$

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:

$$L[x(t), u(t), t] = u^2$$

This metric can be replaced with any other 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. The performance metric could be rewritten to accommodate this.

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

$$x(t_f) = x_f.$$

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 *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

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

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

using the values from the state calculated from Step 2. Here, *p (t)* is an *n-by-1* dimensional matrix, while *R (t)* is an *n-by-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

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

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

Q and g are intermediate variables used to simplify the equations in step 5. Here, Q is a *q-by-q* dimensional matrix while g is a *q-by-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

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

$$\delta x(t_f) = \varepsilon\left[x(t_f) - x_f\right].$$

Here, v, an *n-by*-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 δx 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 ε 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

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

This entire process continues for either a predefined number of iterations, or until the standard deviation of δJ and δu 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 ε. The predicted decrease in the performance index, δJ, 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).

2.2 Hodgkin-Huxley model

One of the classic computational models regarding excitable systems is the Hodgkin-Huxley neuron model (Hodgkin and Huxley 1952). 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:

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

$$\dot{m} = -m(\alpha_m(V) + \beta_m(V)) + \alpha_m(V)$$

$$\dot{n} = -n(\alpha_n(V) + \beta_n(V)) + \alpha_n(V)$$

$$\dot{h} = -h(\alpha_h(V) + \beta_h(V)) + \alpha_h(V)$$

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

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

$$\alpha_h(V) = 0.07\phi e^{-V/20},\ \beta_h(V) = \frac{\phi}{e^{0.1(30-V)+1}}$$

where V is the membrane voltage (mV), 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 ϕ 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:

$$V(t_f) = 12 \text{ mV}.$$

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 endpoint. 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 understepping 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.

2.3 Implementing the gradient algorithm for the Hodgkin-Huxley model

Detailed derivations of equations for the gradient algorithm are included in Online Resource 2. 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).

2.4 FitzHugh-Nagumo 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 et al. (2006) 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 (Clay et al. 2008). 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 (Alon 2006; FitzHugh 1961; Forger and Paydarfar 2004; Glass 2001; Nagumo et al. 1962; Paydarfar and Buerkel 1995; Winfree 2001).

The FitzHugh-Nagumo model (FitzHugh 1961; Nagumo et al. 1962) is defined as follows:

$$\dot{x}_1 = c\left(x_2 + x_1 - \frac{{x_1}^3}{3} - r\right) + u$$

$$\dot{x}_2 = -\frac{1}{c}(x_1 - a + bx_2)$$

This model is unitless, but one can show that the FitzHugh-Nagumo model is a two-dimensional reduction of the Hodgkin-Huxley equations (FitzHugh 1961). 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 (FitzHugh 1961). 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 (Paydarfar and Buerkel 1995), 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_o 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_o=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 ϕ=0 and RF_{68} occurs at phase ϕ=1. Figure 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.

2.5 Implementing the gradient algorithm for the FitzHugh-Nagumo model from quiescence to repetitive firing

A detailed derivation of the equations for the gradient algorithm is included in Online Resource 2. We set our scaling factors to be k=0.5 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 30 milliseconds. Because

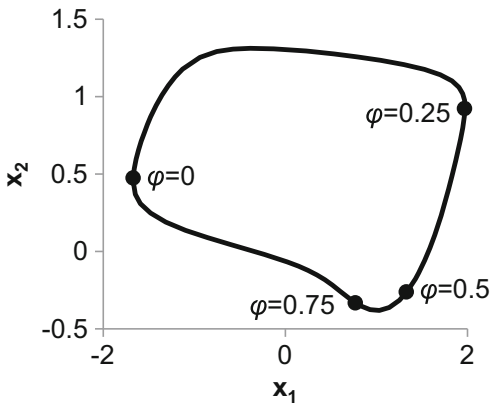

Fig. 1 Plot of the 68 points chosen to represent the limit cycle in the FitzHugh-Nagumo model. The phase values are normalized by time so that 0 and 1 are both at the "peak of the action potential" which occurs when x_1 is at its minimal value. A few other representative phase values are shown to show the time progression around the limit cycle

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 a total of 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 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 δJ was less than 0.0001 and δu was less than 0.1. For the sake of time, we terminated the gradient algorithm after 1,000 iterations if it had not yet converged.

2.6 Implementing the gradient algorithm for the FitzHugh-Nagumo model from repetitive firing to quiescence

A detailed derivation of the equations for the gradient algorithm is included in Online Resource 2. We set our scaling factors to be k=0.5 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 8 milliseconds. In order to develop results comparable to previous literature (Forger and Paydarfar 2004), 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 back to the repetitive firing state. As such, we observed the system's response for a total of 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 units 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.

Like 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 δJ was less than 0.0001 and δu was less than 0.1, with a maximum number of iterations set at 1,000.

3 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 the most 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 (Aliev and Panfilov 1996; Kawato and Suzuki 1980).

3.1 Hodgkin-Huxley model of neuronal excitation

As we can see from Fig. 2, the gradient algorithm begins with a randomly generated stimulus, and within a few generations, the rough shape of the optimal stimulus is seen. Within 30 iterations, the optimal stimulus is revealed, 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 due to the fact that because we are randomly generating the stimulus, it 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 shows the stimulus waveform of gradient algorithm at 100 iterations and the action potential it caused. Note that the membrane potential V in Fig. 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 potential. The L2-norm of this barely supra-threshold rectangular pulse was 49 $\mu J/cm^2$. Thus, the waveform generated using this technique showed a large

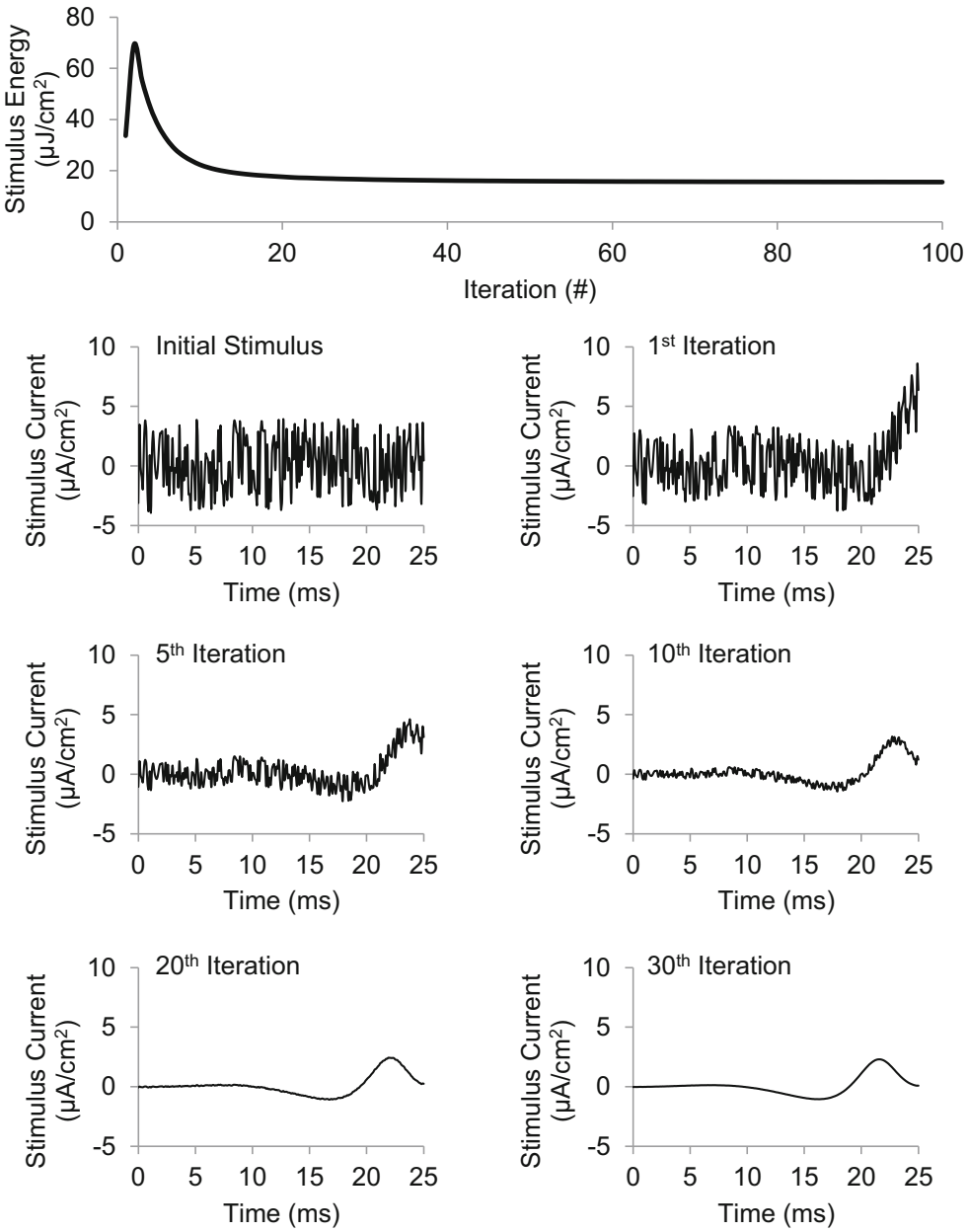

Fig. 2 Gradient algorithm shapes random stimuli towards an optimal waveform. Here we show the progression of solutions as the gradient algorithm begins with a white-noise stimulus and finds the most energetically efficient solution over 100 iterations. The top panel shows the L^2-norm trajectory over 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

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 the constant. This meant that not only did the stimulus decrease in duration, the action potential occurred sooner as well. Figure 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 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 et al. (2012) 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 shorter, the

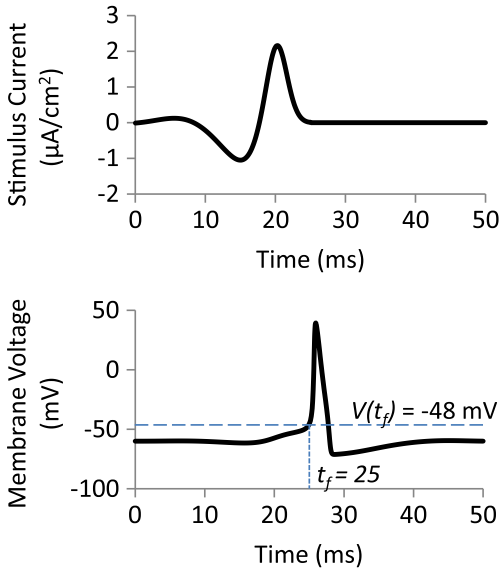

Fig. 3 The optimal stimulus derived from the gradient algorithm (*top*) triggers a single action potential in the Hodgkin-Huxley model (*bottom*)

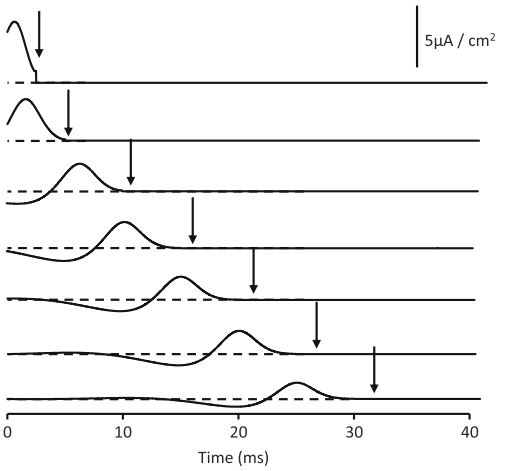

Fig. 5 The optimal waveforms change from a monophasic stimulus to a biphasic stimulus as the amount of time to action potential increases

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 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.

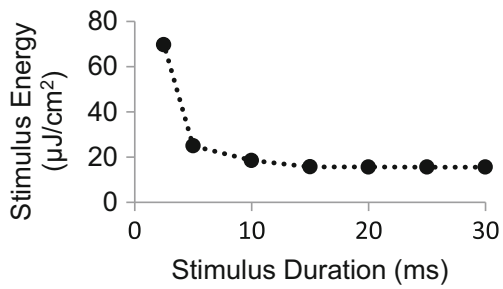

Fig. 4 Longer stimulus duration provides for more energetically efficient stimulus. Once the stimulus duration extends past a certain point, there is no further improvement in energy efficiency

3.2 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 Fig. 7. For each of these runs, the system starts at the quiescent fixed point and terminates at a specified phase, ϕ, as defined previously. The gradient algorithm failed to converge within the set limit of 1,000 iterations for some of the runs, and we have marked them accordingly. This plot reveals a clustering of low L^2-norm values around ϕ=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 ϕ=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 5,000, 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 ϕ=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

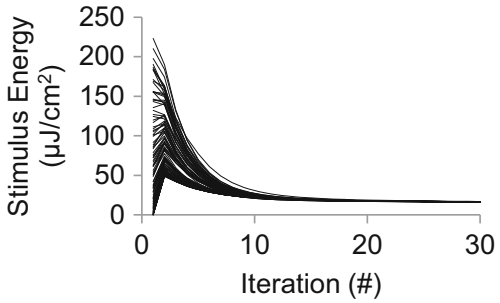

Fig. 6 Gradient algorithm is robust to the initial stimulus. The stimulus energy trajectories of 100 different seeds with different amplitudes are shown

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 $\phi>0.9$). Fig. 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

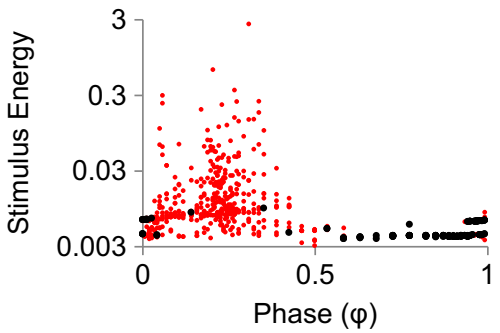

Fig. 7 Only certain phase regions converged when transitioning from quiescence to repetitive firing. The points that converged are marked in *black*, while those that failed are marked in red. For a definition of the phase, see Methods 2.4 and 2.5

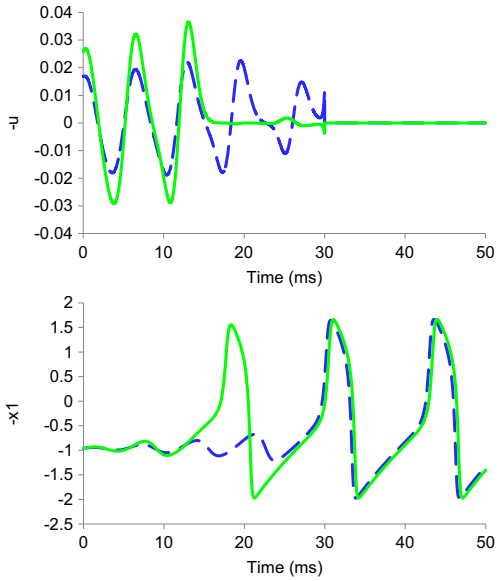

Fig. 8 Gradient algorithm finds multiple optimal solutions that induce a transition from quiescence to repetitive firing. The initial condition (quiescent fixed point) and the terminal condition ($\phi=0$) of the gradient algorithm are the same for both trials. The only difference is the initial randomly 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*). For a definition of the phase, see Methods 2.4 and 2.5

had an amplitude of 0.11 and a duration of 21.102 resulting in an L^2-norm 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, Fig. 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

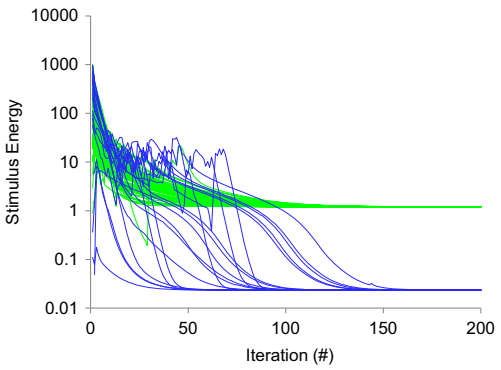

Fig. 9 Trajectories of stimulus energy through 200 iterations of gradient algorithm show convergence to two waveforms. We ran 100 different randomly generated seeds with different scaling factors. The green and blue colors match the respective green and blue waveforms seen in Fig. 8

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 Fig. 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 due to the fact that 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.

3.3 The FitzHugh-Nagumo model: repetitive firing to quiescence

Figure 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 ϕ is between 0.4 and 0.6 at which to begin transitioning 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 ϕ=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 ϕ=0.678. Figure 11 shows the convergence of a set of randomly generated initial conditions towards the two different stimuli. Figure 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 Fig. 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 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

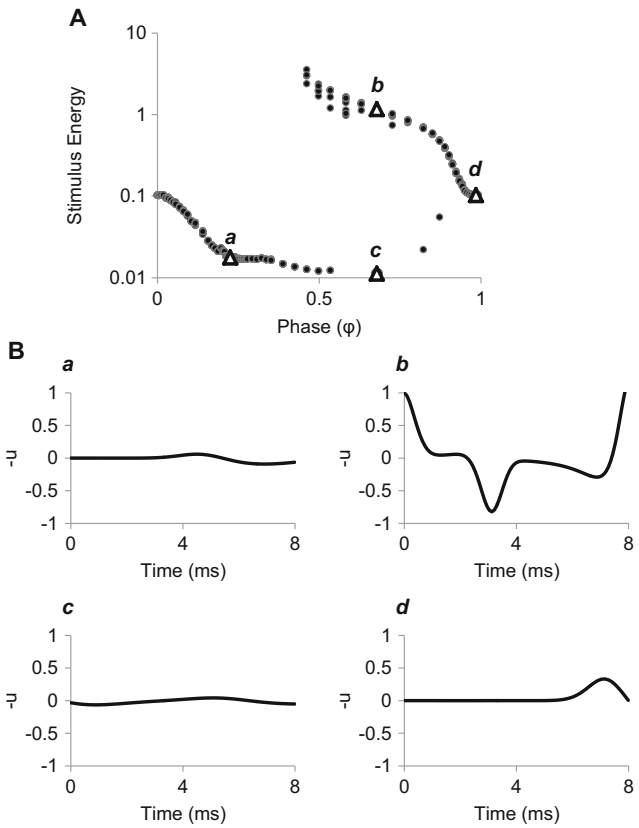

Fig. 10 Gradient algorithm reveals multiplicity in transitioning from repetitive firing to quiescence across different phases. The top panel shows the stimulus energy of the different optimized stimuli (8 ms duration) that induce transitions from different phases of repetitive firing to the quiescent 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. For a definition of the phase, see Methods 2.4 and 2.6

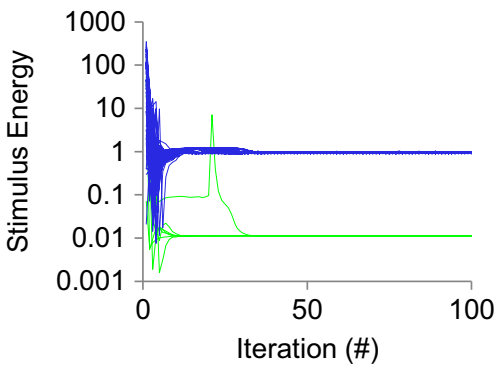

Fig. 11 Random stimuli converge towards two different optimal stimuli that induce a transition from repetitive firing to quiescence. The trajectories of stimulus energy through 100 iterations of gradient algorithm are shown here. All of these stimuli start from the same starting phase (ϕ= 0.678). For a definition of the phase, see Methods 2.4 and 2.6

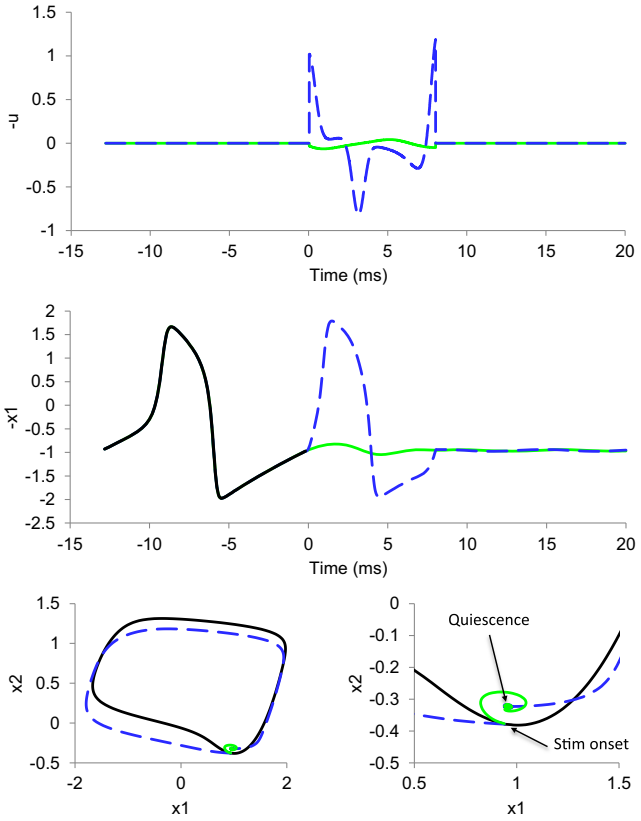

Fig. 12 Gradient algorithm reveals different mechanisms of suppressing repetitive firing. The top panel shows the stimuli, the middle panel the x_1 response to the stimuli. The bottom panel shows the entire state space response to both stimuli. As we can see, one stimulus (*green*) suppresses the system 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. For a definition of the phase, see Methods 2.4 and 2.6

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 (Dean et al. 2009; Forger and Paydarfar 2004; Serkh and Forger 2014).

4 Discussion

Optimal control theory is rooted in calculus of variations, developed by Bernoulli, de l'Hôpital, and Euler (Gelfand and Fomin 2000). 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 were developed. These numerical methods fall into roughly one of two broad categories: indirect methods and direct methods (Betts 1998; Rao 2010).

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 (Danzl et al. 2010; Moehlis et al. 2006; Nabi and Moehlis 2012), parameterized the stimulus (Tahayori and Dokos 2012), or used simpler "integrate-and-fire" models (Jezernik and Morari 2005; Offner 1946). To our knowledge only Forger et al (2011) have applied the indirect method to solve the original Hodgkin-Huxley model in its complete form.

The direct method, first proposed by Kelley (1962) and further developed by Bryson and Ho (1975), 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 only include the voltage threshold. In the study of Forger et al (2011), 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 (Fig. 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 (Bryson and Ho 1975) 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 (Golfetto et al. 2012).

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 showcased 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 (Barnett et al. 2013; Butera et al. 1999; Izhikevich 2000, 2007; Rinzel and Ermentrout 1998), 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 (Chen et al. 2011; Feng et al. 2007; Hauptmann et al. 2005; Howalski et al. 2007; Rubin and Terman 2004; Schiff 2010) and epileptic seizures (Durand and Warman 1994; Iasemidis 2003; Lian et al. 2003; Sunderam et al. 2010; Tass 2003).

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 what design principles may be in play in 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 (Koelling and Nykamp 2012; Machens et al. 2005; Watson et al. 1983), as well as to populations of neurons within functioning networks to better understand how information is transmitted from neuron to neuron (Alle et al. 2009; Attwell and Laughlin 2001; Sengupta et al. 2010; Torrealdea et al. 2006). If one hypothesized that metabolic energy, or ATP consumption, was what the endogenous stimulation optimized, one could construct an equation relating the number of ATPs consumed to the generation of the action potential itself, incorporate that into the optimization metric in the algorithm, and then determine a theoretical result that could be verified or refuted through experimental techniques.

Acknowledgments We thank Daniel Forger and Kirill Serkh for discussions on boundary value problems and also introducing us to the gradient algorithm. We thank John Clay for discussions regarding the ionic basis for excitability in squid axons, and Premananda Indic for discussions and guidance regarding gradient analysis. We also thank William Schwartz and anonymous reviewers for their suggestions and feedback on our manuscript. This work was supported by NIH R01 GM104987 and the Wyss Institute of Biologically Inspired Engineering. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Conflict of interest The authors declare that they have no conflict of interest.

References

Aghababa, M. P., Amrollahi, M. H., & Borjkhani, M. (2012). Application of GA, PSO, and ACO algorithms to path planning of autonomous underwater vehicles. *Journal of Marine Science and Application, 11*(3), 378–386.

Aliev, R. R., & Panfilov, A. V. (1996). A simple two-variable model of cardiac excitation. *Chaos, Solitons & Fractals, 7*(3), 293–301.

Alle, H., Roth, A., & Geiger, J. R. P. (2009). Energy-efficient action potentials in hippocampal mossy fibers. *Science, 325*, 1405–1408.

Alon, U. (2006). *An introduction to systems biology - design principles of biological circuits*. Boca Raton: CRC Press, Taylor & Francis Group, LLC.

Attwell, D., & Laughlin, S. B. (2001). An energy budget for signaling in the grey matter of the brain. *Journal of Cerebral Blood Flow and Metabolism, 21*(10), 1133–45.

Barnett, W., O'Brien, G., & Cymbalyuk, G. (2013). Bistability of silence and seizure-like bursting. *Journal of Neuroscience Methods, 220*(2), 179–89.

Betts, J. J. T. (1998). Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics, 21*(2), 193–207.

Bryson, A. E., & Ho, Y.-C. (1975). Applied Optimal Control (Revised Pr.). Hemisphere Publishing Corporation.

Butera, R. J., Rinzel, J., & Smith, J. C. (1999). Models of respiratory rhythm generation in the pre-Bötzinger complex. I. Bursting pacemaker neurons. *Journal of Neurophysiology, 82*(1), 382–97.

Chen, Y., Wang, J., Wei, X., Deng, B., & Che, Y. (2011). Particle swarm optimization of periodic deep brain stimulation waveforms. Proceedings of the 30th Chinese Control Conference, 754–757.

Clay, J. R., Paydarfar, D., & Forger, D. B. (2008). A simple modification of the Hodgkin and Huxley equations explains type 3 excitability in squid giant axons. *Journal of the Royal Society Interface, 5*, 1421–1428.

Clay, J. R., Forger, D., & Paydarfar, D. (2012). Ionic mechanism underlying optimal stimuli for neuronal excitation: role of Na+channel inactivation. *PLoS One, 7*(9), e45983.

Danzl, P., Nabi, A., & Moehlis, J. (2010). Charge-balanced spike timing control for phase models of spiking neurons. *Discrete and Continuous Dynamical Systems, 28*(4), 1413–1435.

Dean, D. A., Forger, D. B., & Klerman, E. B. (2009). Taking the lag out of jet lag through model-based schedule design. *PLoS Computational Biology, 5*(6), e1000418.

Doležal, J. (1978). A gradient-type algorithm for the numerical solution of two-player zero-sum differential game problems. *Kybernetika, 14*(6), 429–446.

Durand, D. M., & Warman, E. N. (1994). Desynchronization of epileptiform activity by extracellular current pulses in rat hippocampal slices. *Journal of Physiology, 480*(3), 527–537.

Feng, X., Greenwald, B., & Rabitz, H. (2007). Toward closed-loop optimization of deep brain stimulation for Parkinson's disease: concepts and lessons from a computational model. *Journal of Neural Engineering, 4*(2), L14–L21.

FitzHugh, R. (1961). Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal, 1*, 445–466.

Forger, D. B., & Paydarfar, D. (2004). Starting, stopping, and resetting biological oscillators: in search of optimum perturbations. *Journal of Theoretical Biology, 230*, 521–532.

Forger, D. B., Paydarfar, D., & Clay, J. R. (2011). Optimal stimulus shapes for neuronal excitation. *PLoS Computational Biology, 7*(7), e1002089.

Gelfand IM & Fomin SV (2000). Calculus of Variations. Courier Dover Publications

Glass, L. (2001). Synchronization and rhythmic processes in physiology. *Nature, 410*(6825), 277–284.

Golfetto, W. A., Fernandes, S., & da, S. (2012). A review of gradient algorithms for numerical computation of optimal trajectories. *Journal of Aerospace Technology and Management, 4*(2), 131–143.

Gupta, N., & Rink, R. (1973). Optimum control of epidemics. *Mathematical Biosciences, 18*, 383–396.

Hauptmann, C., Popovych, O., & Tass, P. A. (2005). Effectively desynchronizing deep brain stimulation based on a coordinated delayed feedback stimulation via several sites: a computational study. *Biological Cybernetics, 93*(6), 463–470.

Hodgkin, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology, 117*(4), 500–544.

Howalski, C. H., da Silva, G. A., Poppi, R. J., Godoy, H. T., & Augusto, F. (2007). 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. *Analytica Chimica Acta, 585*, 66–75.

Iasemidis, L. D. (2003). Epileptic seizure prediction and control. *IEEE Transactions on Biomedical Engineering, 50*(5), 549–558.

Izhikevich, E. M. (2000). Neural excitability, spiking and bursting. *International Journal of Bifurcation and Chaos, 10*(6), 1171–1266.

Izhikevich EM (2007). Dynamical systems in neuroscience: the geometry of excitability and bursting. (TJ Sejnowski & TA Poggio, Eds.). Cambridge, Massachusetts: MIT Press

Jezernik, S., & Morari, M. (2005). Energy-optimal electrical excitation of nerve fibers. *IEEE Transactions on Biomedical Engineering, 52*(4), 740–743.

Joshi, H. R. (2002). Optimal control of an HIV immunology model. *Optimal Control Applications and Methods, 23*(4), 199–213.

Kawato, M., & Suzuki, R. (1980). Two coupled neural oscillators as a model of the circadian pacemaker. *Journal of Theoretical Biology, 86*(3), 547–575.

Kelley, H. J. (1962). Methods of Gradients. In G. Leitmann (Ed.), *Optimization Techniques* (5th ed., pp. 206–254). New York: Academic Press, Inc.

Kepler, T. B., & Perelson, a. S. (1993). Somatic hypermutation in B cells: an optimal control treatment. *Journal of Theoretical Biology, 164*(1), 37–64.

Kirschner, D., Lenhart, S., & Serbin, S. (1997). Optimal control of the chemotherapy of HIV. *Journal of Mathematical Biology, 35*(7), 775–792.

Koelling, M. E., & Nykamp, D. Q. (2012). Searching for optimal stimuli: ascending a neuron's response function. *Journal of Computational Neuroscience, 33*(3), 449–473.

Lee, E. S. (1964). Optimization by a gradient technique. *Industrial & Engineering Chemistry Fundamentals, 3*(4), 373–380.

Lian, J., Bikson, M., Sciortino, C., Stacey, W. C., & Durand, D. M. (2003). Local suppression of epileptiform activity by electrical stimulation in rat hippocampus *in vitro*. *Journal of Physiology, 547*(2), 427–434.

Loddenkemper, T., & Pan, A. (2001). Deep brain stimulation in epilepsy. *Journal of Clinical Neurophysiology, 116*(6), 217–34.

Lozano, A. M. (2010). Deep brain stimulation for Parkinson's disease. *Journal of Neurosurgery, 112*(3), 199–203.

Machens, C. K., Gollisch, T., Kolesnikova, O., & Herz, A. V. M. (2005). Testing the efficiency of sensory coding with optimal stimulus ensembles. *Neuron, 47*(3), 447–456.

Moehlis, J., Shea-Brown, E., & Rabitz, H. (2006). Optimal inputs for phase models of spiking neurons. *Journal of Computational and Nonlinear Dynamics, 1*(4), 358–367.

Nabi, A., & Moehlis, J. (2012). Time optimal control of spiking neurons. *Journal of Mathematical Biology, 64*(6), 981–1004.

Nagumo, J., Arimoto, S., & Yoshizawa, S. (1962). An active pulse transmission line simulating nerve axon. *Proceedings of the IRE, 2061–2070*.

Offner, F. (1946). Stimulation with minimum power. *Journal of Neurophysiology, 9*(5), 387–390.

Osborne, M. R. (1969). On shooting methods for boundary value problems. *Journal of Mathematical Analysis and Applications, 27*, 417–433.

Paydarfar, D., & Buerkel, D. D. (1995). Dysrhythmias of the respiratory oscillator. *Chaos, 5*(1), 18–29.

Paydarfar, D., Forger, D. B., & Clay, J. R. (2006). Noisy inputs and the induction of on-off switching behavior in a neuronal pacemaker. *Journal of Neurophysiology, 96*(6), 3338–48.

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

Rao, A. (2010). A survey of numerical methods for optimal control. *Advances in the Astronautical Sciences, 135*, 497–528.

Rinzel, J., & Ermentrout, G. B. (1998). Analysis of neural excitability and oscillations. In Methods in Neuronal Modeling (2nd ed., pp. 251–292).

Rubin, J. E., & Terman, D. (2004). High frequency stimulation of the subthalamic nucleus eliminates pathological thalamic rhythmicity in a computational model. *Journal of Computational Neuroscience, 16*(3), 211–235.

Schiff, S. J. (2010). Towards model-based control of Parkinson's disease. *Philosophical Transactions of the Royal Society A, 368*, 2269–2308.

Sengupta, B., Stemmler, M., Laughlin, S. B., & Niven, J. E. (2010). Action potential energy efficiency varies among neuron types in vertebrates and invertebrates. *PLoS Computational Biology, 6*(7), e1000840.

Serkh, K., & Forger, D. B. (2014). Optimal schedules of light exposure for rapidly correcting circadian misalignment. *PLoS Computational Biology, 10*(4), e1003523.

Sunderam, S., Gluckman, B., Reato, D., & Bikson, M. (2010). Toward rational design of electrical stimulation strategies for epilepsy control. *Epilepsy & Behavior, 17*(1), 6–22.

Tahayori, B., & Dokos, S. (2012). Optimal stimulus current waveshape for a Hodgkin-Huxley model neuron. 34th Annual International Conference of the IEEE EBS, 4627–4630.

Tass, P. A. (2003). A model of desynchronizing deep brain stimulation with a demand-controlled coordinated reset of neural subpopulations. *Biological Cybernetics, 89*(2), 81–88.

Torrealdea, F. F., D'Anjou, A., Graña, M., Sarasola, C., & D'Anjou, A. (2006). Energy aspects of the synchronization of model neurons. *Physical Review E, 74*(1), 011905.

Watson, A., Barlow, H., & Robson, J. (1983). What does the eye see best? *Nature, 302*(5907), 419–422.

Winfree, A. T. (2001). *The Geometry of Biological Time. (J. E. Marsden, L. Sirovich, & S. Wiggins, Eds.)* (2nd ed.). New York: Springer.

Ypma, T. J. (1995). Historical development of the Newton–Raphson method. *SIAM Review, 37*(4), 531–551.