Learning Control Policies of Hodgkin-Huxley Neuronal Dynamics

Malvern Madondo MALVERN.MADONDO@EMORY.EDU
Department of Computer Science

Deepanshu Verma DEEPANSHU.VERMA@EMORY.EDU
Lars Ruthotto LRUTHOTTO@EMORY.EDU
Department of Mathematics

Nicholas Au Yong NICHOLAS.AU.YONG@EMORY.EDU
Department of Neurosurgery

Emory University, Atlanta, GA, USA

arXiv:2311.07563v1 [math.OC] 13 Nov 2023

Abstract

We present a neural network approach for closed-loop deep brain stimulation (DBS). We cast the problem of finding an optimal neurostimulation strategy as a control problem. In this setting, control policies aim to optimize therapeutic outcomes by tailoring the parameters of a DBS system, typically via electrical stimulation, in real time based on the patient's ongoing neuronal activity. We approximate the value function offline using a neural network to enable generating controls (stimuli) in real time via the feedback form. The neuronal activity is characterized by a nonlinear, stiff system of differential equations as dictated by the Hodgkin-Huxley model. Our training process leverages the relationship between Pontryagin's maximum principle and Hamilton-Jacobi-Bellman equations to update the value function estimates simultaneously. Our numerical experiments illustrate the accuracy of our approach for out-of-distribution samples and the robustness to moderate shocks and disturbances in the system.

Keywords: Neuronal dynamics, Hodgkin-Huxley model, Nonlinear dynamics, Optimal Control, Machine Learning, Neural Networks, Pontryagin's Maximum Principle, Hamilton-Jacobi-Bellman equation, Semi-global solution

1. Introduction

Disruptions in the normal functioning of neurons are an early symptom of many neurological conditions such as Parkinson's Disease (Brown, 2003; Wichmann and DeLong, 2016). Controlling neuronal dynamics has potentially high practical importance in devel-

oping neurotechnology and improving health-related quality of life. In neuromodulatory interventions such as deep brain stimulation, physicians aim to alter neuronal activity by delivering electrical pulses to target locations in the brain via a battery-powered pulse generator or neurostimulator (Arlotti et al., 2016; Koeglsperger et al., 2019).

A key clinical objective is to learn an effective stimulus input with optimal therapeutic benefits for the patient while reducing unwanted side effects. Traditionally, physicians have relied on ad hoc approaches to learn the optimal stimulation strategy over multiple clinical visits (Mohammed et al., 2018; Parastarfeizabadi and Kouzani, 2017; Yu et al., 2020). However, this approach has limitations as the neurostimulator continuously delivers a fixed pattern of stimulation, reducing the battery life in the long run and potentially being unnecessary during normal neuronal function (Shirvalkar et al., 2018; Meidahl et al., 2017). Despite recent advancements in neurotechnology, the development of adaptive stimulation strategies remains a challenging research area that requires real-time control algorithms based on the brain's current clinical state (Carron et al., 2013; Parastarfeizabadi and Kouzani, 2017; Wang et al., 2007).

In this work, we tackle the biomedical problem of learning a stimulation strategy by formulating it as an optimal control problem. This framework paves the way for developing adaptive closed-loop neuromodulation strategies and is promising for advancing neurostimulation therapies. We characterize neuronal activity with the Hodgkin-Huxley model (Hodgkin and Huxley, 1952), a biophysically accurate nonlinear system of differential equations that captures the essential dynamics of neuronal behavior. By com-

bining Pontryagin's maximum principle (PMP) and Hamilton-Jacobi-Bellman (HJB) equation, we establish a solid mathematical foundation that allows us to obtain approximately optimal controls in real time. While the PMP is a local solution method and is sensitive to the initial states and perturbations in the optimal trajectories, it provides first-order necessary conditions for optimality. In contrast, the HJB equation yields controls in feedback form and provides sufficient conditions for optimality. However, solving the HJB equations in higher dimensions is notoriously challenging due to the curse of dimensionality (CoD). To overcome these limitations, we adopt a semi-global neural network (NN) approach that approximates the value function and learns a policy across a diverse set of initial conditions and the state space likely to be encountered when following optimal trajectories. This approach has been successfully demonstrated in various control problems in economics (Han et al., 2018), trajectory planning (Onken et al., 2021a), stochastic optimal control (Pereira et al., 2020; Li et al., 2022; Exarchos and Theodorou, 2018), and other domains.

Contributions Our main contributions are as follows:

- We provide a rigorous and comprehensive control formulation for designing an optimal neurostimulation strategy as a control problem. This approach enables the application of control theory and machine learning to address complex biological systems, leading to innovative approaches in both scientific research and medical practice.

- We establish a concrete link between the learning problem and optimal control theory, specified by the relation between PMP and HJB equations. We approximate the value function using a neural network satisfying the HJB, from which the optimal (stimulation) control can be recovered in real time via the feedback form. This approach is useful because it is transferable to closed-loop control problems in domains beyond healthcare.

2. Neuronal Dynamics

Developed in the 1950s, the Nobel-prize-winning Hodgkin-Huxley (HH) equations are widely considered the gold standard model of neuronal dynamics in computational neuroscience (Hodgkin and Huxley, 1952). The HH model ascribes the action potential

(spike) generation in a neuron, i.e., the moment of maximum membrane potential/voltage, to the movement of fast depolarizing and slow hyperpolarizing ionic currents.

We define the state dynamics using the HH model, which describes the electrophysiological activity of neurons and accounts for currents flowing through the neuronal membrane across different ion channels, including sodium, potassium, and leakage channels. Leak channels comprise all other ions with slower dynamics, such as chloride ions. The HH model is characterized by the following first-order dynamical system, with state variables comprising of a neuron's membrane potential (Figure $1(a)$), V_{m}, and three gating variables (Figure $1(b)$) representing probabilities of sodium activation, m, potassium activation n, and sodium inactivation, h

$$
\begin{aligned}
\frac{d\boldsymbol{z}}{dt}(t) &= f\big(t, \boldsymbol{z}(t)\big) + \boldsymbol{e}_1 u(t) \\
\boldsymbol{z}(0) &= \boldsymbol{x}, \quad \boldsymbol{e}_1 = [1,0,0,0]^\top, \quad 0 \le t \le T
\end{aligned}
\tag{1}
$$

where $\boldsymbol{z}(t) = [V_{\mathrm{m}}, m, n, h]^\top \in \mathbb{R}^4$ denotes the state variable, with $\boldsymbol{x}$ being the initial state of the system. Here, T denotes the fixed final time horizon, and the control variable $u(t) : [0,T] \to \mathbb{R}$ represents the external current/stimulus provided as input by a controller at time t. The function $f : [0,T] \times \mathbb{R}^4 \to \mathbb{R}^4$ describes the evolution of state dynamics of the HH model and can be written as

$$
\begin{aligned}
&f\left(t, \boldsymbol{z}(t)\right) = \\
&\begin{pmatrix}
f_0(t, \boldsymbol{z}) \\
\alpha_m(z_0)(1 - z_1) - \beta_m(z_0)z_1 \\
\alpha_n(z_0)(1 - z_2) - \beta_n(z_0)z_2 \\
\alpha_h(z_0)(1 - z_3) - \beta_h(z_0)z_3
\end{pmatrix},
\end{aligned}
\tag{2}
$$

where

$$
f_0\left(t, \boldsymbol{z}\right) = -\frac{1}{C_{\mathrm{m}}}
\begin{pmatrix}
\bar{g}_{\mathrm{Na}} z_1^3 z_3 (z_0 - E_{\mathrm{Na}}) \\
+ \bar{g}_{\mathrm{K}} z_2^4 (z_0 - E_{\mathrm{K}}) \\
+ \bar{g}_{\mathrm{l}}(z_0 - E_{\mathrm{l}})
\end{pmatrix}.
$$

Here, C_{m} is the membrane capacitance and $\bar{g}_{\mathrm{Na}}, \bar{g}_{\mathrm{K}}, \bar{g}_{\mathrm{l}}$ represent the maximum conductance of the sodium and potassium and the leak currents, respectively. The variables $E_{\mathrm{Na}}, E_{\mathrm{K}}$, and E_{l} represent the equilibrium potentials of the corresponding ions. The functions α_j and β_j, $j \in \{1,2,3\}$ defined in Table 1, are voltage-dependent rate functions for each of the gating variables. The default values and units of all the

2

parameters are obtained from Hodgkin and Huxley (1952) and listed under the "normal" neuronal activity category in Table 2. Figure 1 shows the state trajectory of typical HH neuronal activity.
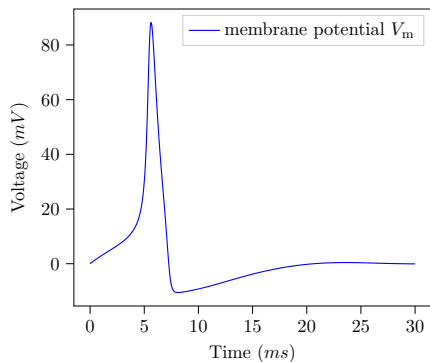
(*a*) Membrane Potential
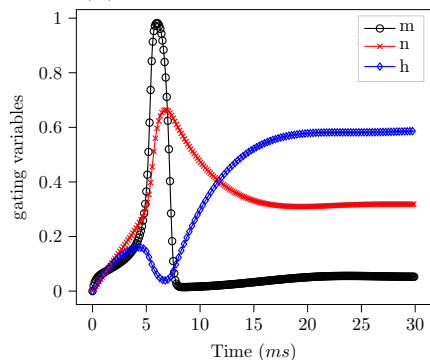
(*b*) Gating Variables

Figure 1: Evolution of the membrane potential and gating variables of **normal** HH equations, with no controls/stimuli, i.e., $u = 0$, $\forall t > 0$, and initial state $\boldsymbol{x} = [0, 0, 0, 0]^\top$.

The HH model (Equation (1)) is known to be stiff since V_m and m change rapidly relative to n and h. The rapid depolarization (membrane potential becomes more positive) and subsequent repolarization (membrane potential becomes more negative) of the neuron's membrane potential results in the spike/peak (formally known as the action potential) in Figure 1. On the one hand, the depolarization phase of the spike is largely driven by the influx of sodium ions due to the activation of the sodium channels. On the other hand, the repolarization phase is driven by the subsequent activation of the potassium channels and inactivation of the sodium chan-

nels. The general flow of ions, such as sodium and potassium, across the neuron's membrane, plays a crucial role in transmitting electrical signals in neurons. We refer to Gerstner and Kistler (2002); Koch (2004); Miller (2018) for detailed explanations of neuronal dynamics.

Table 1: Parameters of the gating variables in the HH equations. The parameters α and β depend only on the voltage, z_0.

j	α_j	β_j
z_1	$\dfrac{2.5 - 0.1z_0}{\exp(2.5 - 0.1z_0) - 1}$	$4\exp(-\frac{z_0}{18})$
z_2	$\dfrac{0.1 - 0.01z_0}{\exp(1 - 0.1z_0) - 1}$	$0.125\exp(-\frac{z_0}{80})$
z_3	$0.07\exp(-\frac{z_0}{20})$	$\dfrac{1}{\exp(3 - 0.1z_0) + 1}$

Table 2: Nominal parameter values of the HH model under normal and pathological conditions.

PARAMETER	NORMAL	PATHOLOGICAL
Cm	1.0	1.0
$\bar{g}_\mathrm{Na}$	120.0	380.0
$\bar{g}_\mathrm{K}$	36	36.0
$\bar{g}_\mathrm{l}$	0.3	0.3
E_Na	115.0	115
E_K	-12.0	-12.0
E_l	10.613	10.613

2.1. Computational Approaches to Neurostimulation

Interest in applying machine learning and optimal control to neuroscience has soared recently. While the machine learning community generally favors a data-driven approach (Boutet et al., 2021; Lu et al., 2020; Narayanan and Subbian, 2023; Peralta et al., 2021), a model-based control approach offers opportunities to not only study the effect of external stimuli on neuronal dynamics but to also understand the computational properties of neural systems further
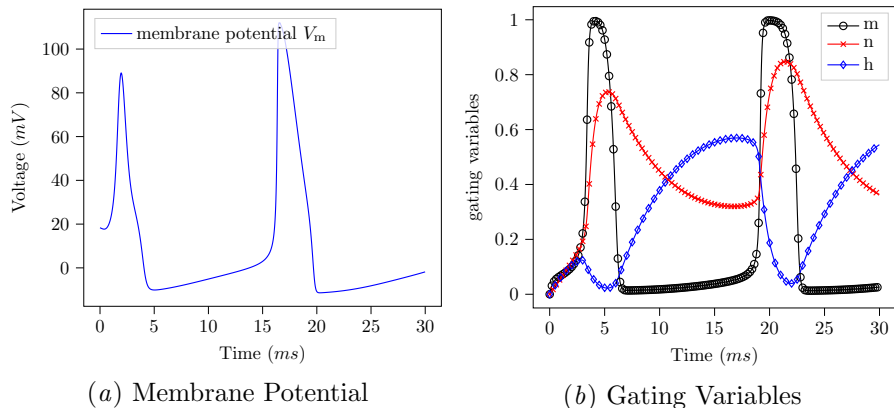
(a) Membrane Potential

(b) Gating Variables

Figure 2: Evolution of the membrane potential and gating variables of **pathological** HH equations, with no controls/stimuli, i.e., $u = 0$, $\forall\, t > 0$, and initial state $\boldsymbol{x} = [0, 0, 0, 0]^\top$.

(Breakspear, 2017; Catterall et al., 2012; Salfenmoser and Obermayer, 2022).

Recent studies on developing computational models for neurostimulation generally focus on the dynamics of a single-compartment neuron model, or its extension, the multi-compartment model, which accounts for the spatial dimensionality of neurons and provides a more detailed representation of neuronal behavior (Deco et al., 2008; Fröhlich and Jezernik, 2005; So et al., 2012; van Albada and Robinson, 2009). The underlying dynamics in both compartment models are often described by HH-type equations (Equation (2)). In this work, we focus on the single-compartment HH model describing the neuronal activity of a single neuron. We utilize the standard parameters of the HH model to represent normal neuronal dynamics.

Pathological Activity Additionally, we introduce a pathological state by randomly distorting one or more parameters, such as sodium conductance $\bar{g}_{\mathrm{Na}}$, as outlined in Table 2. This has the effect of generating abnormal spikes in the membrane potential (see Figure 2(a)) and gating variables (Figure 2(b)). This approach draws inspiration from a recent study by Narayanan and Subbian (2023) that proposed a machine learning-based deep brain stimulator for controlling epileptic seizures. While our application differs from that study, the idea of simulating pathological conditions by manipulating the parameters of the HH model aligns with our objectives. Figure 1

shows the evolution of the HH model under pathological conditions.

3. Optimal Control of Neuronal Dynamics

In this section, we target the control of pathological neuronal activity (Figure 2) by determining a control policy that can effectively restore normal functioning. We adopt an optimal control formulation of the neurostimulation problem that aims at restoring neuronal function (Figure 1) while expending less energy. Recent research enables this by combining function-approximating powers of NN and OC theory to solve high-dimensional control problems using neural networks (Onken et al., 2021b; Kunisch and Walter, 2021).

3.1. Problem Setting

For a fixed initial state x and a finite time-horizon T, consider the optimal control problem constrained by the HH-based nonlinear system dynamics in Equation (1).

We aim to find an optimal control (stimulus) u^* : $[0, T] \to \mathbb{R}$ that drives the system toward a reference/target state, $\boldsymbol{z}^*$, while incurring the minimum possible cost. To this end, we define the control objective functional (cost) J as

$$J(t, \boldsymbol{z}, u) = G\big(\boldsymbol{z}(T)\big) + \int_t^T L\big(s, \boldsymbol{z}(s), u(s)\big)ds, \quad (3)$$
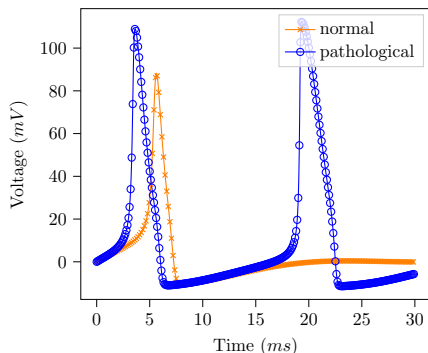
Figure 3: Membrane potentials, V_{m}, for normal condition ($\bar{g}_{\mathrm{Na}} = 120$) vs a pathological condition ($\bar{g}_{\mathrm{Na}} = 380$).

with the terminal cost $G : \mathbb{R}^4 \to \mathbb{R}$ defined by

$$G(z) = \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{z}^*(T)\|^2, \qquad (4)$$

and the Lagrangian, or running cost, $L : [0, T] \times \mathbb{R}^4 \times \mathbb{R} \to \mathbb{R}$ given by

$$L(t, \boldsymbol{z}, u) = \lambda\|u\|^2 + Q\frac{1}{2}\|\boldsymbol{z} - \boldsymbol{z}^*(t)\|^2. \qquad (5)$$

While the terminal cost, G, penalizes the distance between the final state $\boldsymbol{z}(T)$ and the given target terminal state of the system $\boldsymbol{z}^*(T)$, L accumulates the cost of controlling the system and expending energy at each time step. Here, Q penalizes the tracking term and is set to 200 in our experiments. λ is a problem-specific constant for electrode impedance used to limit the energy of the control signal. It is typically set to 0.5 (Fleming et al., 2020).

Overall, we seek to minimize J over all admissible controls $u \in \mathcal{U}$, and set

$$\Phi\big(t, \boldsymbol{z}(t)\big) = \inf_u J(t, \boldsymbol{z}, u), \quad \text{s.t. (1)}. \qquad (6)$$

Here, Φ is called the value function or the optimal cost-to-go. A solution u^* incurring this minimal cost is called an optimal control, and the corresponding state variable $\boldsymbol{z}^*$ is called an optimal trajectory.

3.2. Optimal Control Solution Approaches

Controls problems like (6) can be solved using local or global optimization techniques.

Local solution methods aim to find an optimal control policy based on a fixed initial state. These methods are typically used for *open-loop* control, which involves determining the control inputs (i.e., the optimal control trajectory) in advance, often through optimization techniques, and then applying these predetermined control inputs to the system without real-time feedback. However, recomputation or adaptation of the control strategy may be necessary when unexpected changes or perturbations occur in the system. This approach mirrors the conventional DBS practice of clinicians manually adjusting stimulation parameters to provide effective therapy for a patient. As the patient's condition evolves, additional appointments become necessary to fine-tune the treatment for optimal outcomes (Parastarfeizabadi and Kouzani, 2017; Yu et al., 2020).

A representative example of open-loop control is the *all-at-once* Interior Point Method (IPM) (Nocedal and Wright, 2006), a numerical optimization technique that determines the control inputs for the entire trajectory in advance, without incorporating real-time feedback from the system. We employ this approach as our baseline controller and iteratively solve the optimization problem (6) by considering the entire control trajectory as a single optimization variable, simultaneously optimizing both the control input and the system's state trajectory.

We also consider Pontryagin's Maximum Principle (PMP) (Pontryagin, 2018), a *local solution* method that provides first-order necessary conditions for optimality and yields an open-loop optimal control law (Onken et al., 2021b; Li et al., 2022; Fleming and Soner, 2006) as follows,

$$u^*(s) \in \arg\max_u \mathcal{H}\big(s, \boldsymbol{z}^*(s), \boldsymbol{p}(s), u(s)\big), \qquad (7)$$

for every $s \in [0, T]$. Here, for an adjoint state $\boldsymbol{p}$, the Hamiltonian $\mathcal{H}$ of the system is defined as

$$\mathcal{H}(t, \boldsymbol{z}, \boldsymbol{p}, u) = -L(t, \boldsymbol{z}, u) - \boldsymbol{p}^\top\big[f(t, \boldsymbol{z}) + \boldsymbol{e}_1 u\big]. \quad (8)$$

Solving this system yields a solution that is limited to a specific initial state, necessitating re-computation for different initial states or trajectory perturbations.

Global solution methods solve the control problem for any given initial state, often yielding controls in *closed-loop* or *feedback* form. The control inputs are adjusted in real-time based on the system's current state and its response to previous control actions, enabling closed-loop control. This is relevant
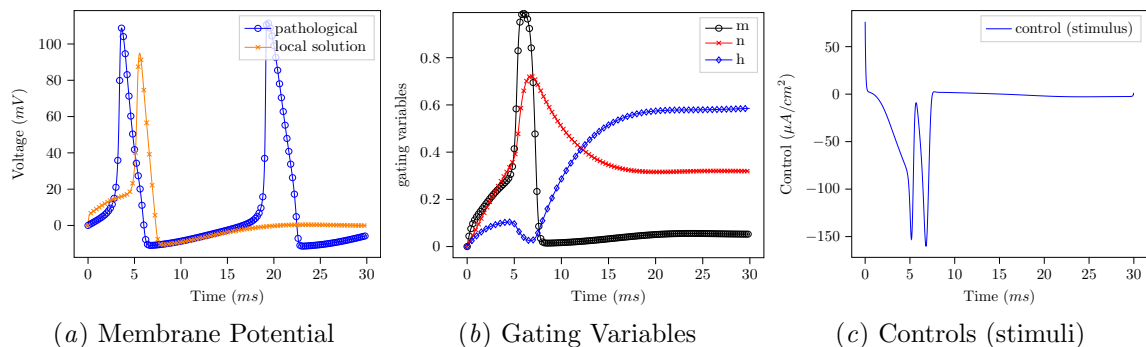
Figure 4: **Local solution** approach drives initial **pathological** HH system, with $\bar{g}_{\text{Na}} = 380$, toward **normal** behavior by injecting control stimuli.

for critical applications such as DBS where real-time adaption to a patient's ongoing symptoms is crucial.

The Hamilton-Jacobi-Bellman (HJB) equation (Kirk, 2004) (also known as the Dynamic Programming (DP) equation (Bellman, 1966)) is a global solution method used to derive the optimal control policy over a larger state space. Given any initial state, we can solve (6) and produce an optimal control that leads to global convergence of the closed-loop system. DP states that the value function Φ satisfies the HJB equation (Fleming and Soner, 2006),

$$\begin{cases} -\partial_t \Phi\big(t, \boldsymbol{z}(t)\big) + \mathcal{H}\big(t, \boldsymbol{z}, \nabla_{\boldsymbol{z}} \Phi\big(t, \boldsymbol{z}(t)\big), u\big) = 0, \\ \qquad\qquad \Phi\big(T, \boldsymbol{z}(T)\big) = G\big(\boldsymbol{z}(T)\big). \end{cases} \tag{9}$$

In the context of DBS, the HJB equation guides the development of broad control policies that account for the entire state space. It provides a more comprehensive approach to optimizing stimulation across a wide range of potential scenarios.

3.3. Semi-Global Approach

While solving the HJB equation is generally fast and feasible when the state dimension ≤ 3, it is prone to the CoD for higher dimensional problems. To mitigate this, we empoly a semi-global approach that uses neural networks to approximate the value function defined by the HJB equation, owing to their universal approximation properties (Han et al., 2018; Kunisch and Walter, 2021; Onken et al., 2021b; Exarchos and Theodorou, 2018; Pereira et al., 2020).

This approach leverages the connection between the PMP and the HJB equation, which has long

been established in control theory (Cernea and Frankowska, 2005; Fleming and Soner, 2006) as

$$\boldsymbol{p}(t) = -\nabla_{\boldsymbol{z}} \Phi\big(t, \boldsymbol{z}(t)\big).$$

This relation, at optimality, helps obtain the optimal control $u^*(s)$ from the value function Φ at any given time via the feedback form,

$$u^*(s) \in \arg\max_u \mathcal{H}\big(s, \boldsymbol{z}^*(s), \nabla_{\boldsymbol{z}} \Phi\big(s, \boldsymbol{z}^*(s)\big), u\big). \tag{10}$$

Assuming that we have a close-form solution for u^* in Equation (10) (which is true for our case), then the optimal controls can be recovered in real time using Φ and its gradient, $\nabla_{\boldsymbol{z}} \Phi$. This is ideal for clinical applications such as neurostimulation where swiftly computing controls for different times or states in real time is highly desirable.

Ties to Reinforcement Learning It is also important to note that the value function, Φ, in the HJB equation is closely related to the value function estimated by methods such as actor-critic in Reinforcement Learning (RL) (Konda and Tsitsiklis, 1999). Ties between control theory and RL have been cemented by seminal works including (Bertsekas, 2019; Recht, 2018; Sutton et al., 1992). In RL, the optimal policy maximizes Φ, whereas in OC, we seek a control policy that minimizes Φ. While scarce, existing applications of RL to neuroscience and deep brain stimulation include (Botvinick et al., 2020; Gao et al., 2020; Krylov et al., 2020; Lu et al., 2020). These works demonstrate the potential of control-based techniques in developing innovative solutions for long-standing problems in healthcare and beyond.

4. Learning Problem

Following Section 3.3, we approximate the value function, Φ, in Equation (6) using a neural network, $\mathrm{N}(\boldsymbol{y}; \boldsymbol{\theta}_{\mathrm{N}})$, with parameters $\boldsymbol{\theta}$ as follows

$$\Phi_{\boldsymbol{\theta}}(\boldsymbol{y}) = \boldsymbol{w}^{\top} \mathrm{N}(\boldsymbol{y}; \boldsymbol{\theta}_{\mathrm{N}}) + \frac{1}{2} \boldsymbol{y}^{\top} (\boldsymbol{A}^{\top} \boldsymbol{A}) \boldsymbol{y} + \boldsymbol{b}^{\top} \boldsymbol{y} + c,$$
$$\text{where } \boldsymbol{\theta} = (\boldsymbol{w}, \boldsymbol{\theta}_{\mathrm{N}}, \boldsymbol{A}, \boldsymbol{b}, c), \quad (11)$$

with space-time inputs $\boldsymbol{y} = (s, \boldsymbol{z}(s)) \in \mathbb{R}^{d+1}$. Here, d is the state dimension $(= 4)$ and $\boldsymbol{\theta}$ consists of all trainable weights: $\boldsymbol{w} \in \mathbb{R}^m, \boldsymbol{\theta}_{\mathrm{N}} \in \mathbb{R}^{p_{\mathrm{N}}}, \boldsymbol{A} \in \mathbb{R}^{d \times (d+1)}, \boldsymbol{b} \in \mathbb{R}^{d+1}$, and $c \in \mathbb{R}$, where $\boldsymbol{A}, \boldsymbol{b}$ and c model linear dynamics and neural network $\mathrm{N}(\boldsymbol{y}; \boldsymbol{\theta}_{\mathrm{N}}) : \mathbb{R}^{d+1} \to \mathbb{R}^m$ models nonlinear dynamics. p_{N} denotes the number of parameters of the NN.

To learn the parameters $\boldsymbol{\theta}$ of the NN, we first sample some initial states $\boldsymbol{x}$ from the Gaussian distribution, ρ with mean 0 and a variance of 10, and approximately solve the following minimization problem

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{z_0 \sim \rho} \left\{ \begin{array}{c} \ell(T) + g(\boldsymbol{z}(T)) + \gamma_1 c_{\mathrm{HJB}}(T) \\ + \gamma_2 |\Phi_{\boldsymbol{\theta}}(T, \boldsymbol{z}(T)) - g(\boldsymbol{z}(T))| \end{array} \right\}, \quad (12)$$

subject to

$$\partial_s \begin{pmatrix} \boldsymbol{z}(s) \\ \ell(s) \\ c_{\mathrm{HJB}}(s) \end{pmatrix} = \\ \begin{pmatrix} -\nabla_{\boldsymbol{p}} \mathcal{H}(s, \boldsymbol{z}(s), \nabla_{\boldsymbol{z}} \Phi_{\boldsymbol{\theta}}(s, \boldsymbol{z}(s)), u^*) \\ L_{\boldsymbol{z}}(s) \\ R_{\boldsymbol{z}}(s) \end{pmatrix}, \quad (13)$$

for $s \in (0, T)$, initialized with $\ell(0) = c_{\mathrm{HJB}}(0) = 0$ and $\boldsymbol{z}(0) = \boldsymbol{x}$. Here,

$$L_{\boldsymbol{z}}(s) = \\ \nabla \Phi_{\boldsymbol{\theta}}(s, \boldsymbol{z}(s)) \cdot \nabla_{\boldsymbol{p}} \mathcal{H}(s, \boldsymbol{z}(s), \nabla \Phi_{\boldsymbol{\theta}}(s, \boldsymbol{z}(s)), u^*),$$

and

$$R_{\boldsymbol{z}}(s) = \big| -\partial_t \Phi_{\boldsymbol{\theta}}(s, \boldsymbol{z}(s)) + \\ \mathcal{H}(s, \boldsymbol{z}(s), \nabla_{\boldsymbol{z}} \Phi_{\boldsymbol{\theta}}(s, \boldsymbol{z}(s)), u^*) \big|,$$

with ℓ denoting the accumulated running cost.

The hyperparameters $\gamma_1, \gamma_2 \geq 0$ balance minimization of the control objective and HJB penalization.

5. Numerical Experiments

In this section, we examine two stimulation strategies: a baseline controller that applies the all-at-once IPM to solve the control problem in an open-loop manner, i.e., with no system feedback, and a controller that leverages neural networks for optimized controls over a broad state space (semi-global method). Our NN-based controller architecture has depth 2, width 64, a learning rate of 0.005, and uses the ADAM optimizer. We evaluate robustness by introducing minor perturbations or shocks to the HH system, Equation (1).

5.1. Neuromodulatory Effects

We simulate the HH model for both normal and pathological conditions, based on parameters in Table 2. The former is achieved by modeling with the original parameters defined in (Hodgkin and Huxley, 1952), and the latter is obtained by varying the parameters, specifically targeting $\bar{g}_{\mathrm{Na}}$. This distorts the ion flow across the neural membrane, which in turn disrupts the action potential generation, see Figure 3. In this scenario, the objective is to develop control strategies that restore the disrupted system to its normal condition while minimizing the energy required for input current injection. This is valuable in countering pathological neural activity in conditions like Parkinson's disease or chronic pain (Little et al., 2013; Shirvalkar et al., 2018).

For both approaches, we initialize the simulation by perturbing the neuron's resting state (adding Gaussian noise scaled by a factor of 10 to $V_{\mathrm{m}} = 0$ while setting the other state variables to 0). This captures the effect of the aforementioned pathological activity. Figure 4 shows the state trajectories and controls learned by the baseline while driving the pathological states toward the normal states. Likewise, Figure 5 shows the state trajectories and controls learned by the neural network approach on the same problem. Both approaches restore the pathological HH system to normal conditions, albeit with varying stimuli.

5.2. Suboptimality

The baseline IPM controller modulates neuronal activity for a fixed initial state x at a time. We consider the solution obtained using this approach as the ground-truth optimal solution. To compare the performance of the semi-global NN approach, we evalu-
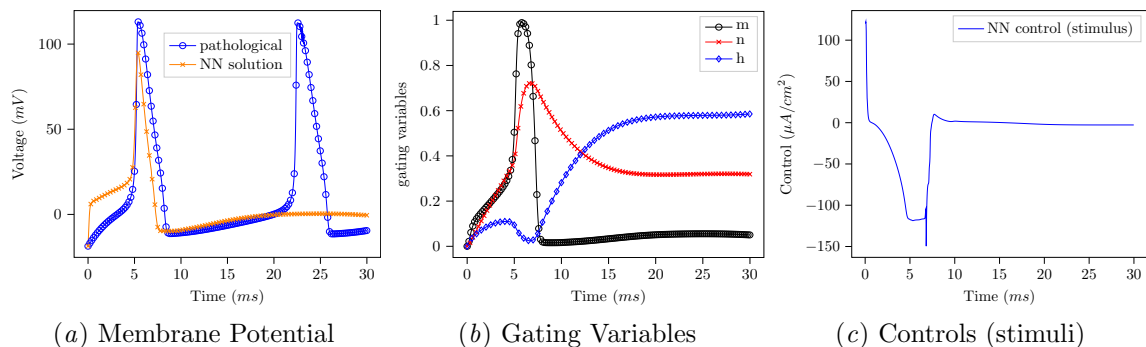
(a) Membrane Potential   (b) Gating Variables   (c) Controls (stimuli)

Figure 5: **Semi-global solution** approach leveraging neural networks successfully drives initial **pathological** HH system, with $\bar{g}_{\mathrm{Na}} = 380$, toward **normal** behavior by injecting control stimuli.

Table 3: Running and terminal costs for single instance shown in Figures 4 and 5.

	RUNNING (ℓ)	TERMINAL (G)	TOTAL
IPM	45725.6861	0.0279	45725.7140
NN	46890.1119	0.0088	46890.1207

ate its solution for the initial state and compute its suboptimality relative to the baseline solution. Table 3 compares the objective values of both methods on $\boldsymbol{x} = [0, 0, 0, 0]$. The NN-based controller achieves near optimal performance, considering it solves the control problem for a larger state space compared to the baseline.
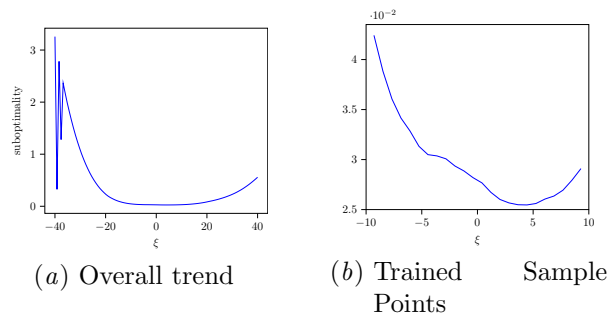
We further evaluate the performance of the NN solution by measuring its suboptimality across various initial conditions $(\xi, 0, 0, 0)$. To achieve this, we uniformly sample 100 values of ξ between -40.0 and 40.0, representing the membrane potential V_{m}. For each $(\xi, 0, 0, 0)$ combination, where the local solution is considered optimal for every initial state, we calculate the suboptimality of the NN solution relative to the local solution. It is worth noting that our network is specifically trained for ξ values within the range of $[-10, 10]$. Therefore, evaluating the performance across a broader range of ξ values also provides insights into how our model performs on out-of-distribution samples. The results of this evaluation are depicted in Figure 6.

(a) Overall trend   (b) Trained Sample Points

Figure 6: Suboptimality comparison between NN approach and local solution method for ξ in $[-40, 40]$. **Left**: Overall trend. **Right**: Zoomed-in view on trained sample points.

5.3. Robustness to shocks

We investigate the robustness of the NN control policy to shocks or disturbances in the system dynamics. In Figure 7, we illustrate two trajectories: a normal trajectory and one with an added shock, both starting from the initial state $\boldsymbol{x} = [0, 0, 0, 0]$. Remarkably, the NN is capable of recovering the optimal trajectory even after the shock. This can be attributed to the semi-global approximation of the value function employed by the NN. Additionally, since the NN is trained offline, it can handle real-time disturbances without the need for re-computation, unlike the baseline approach.
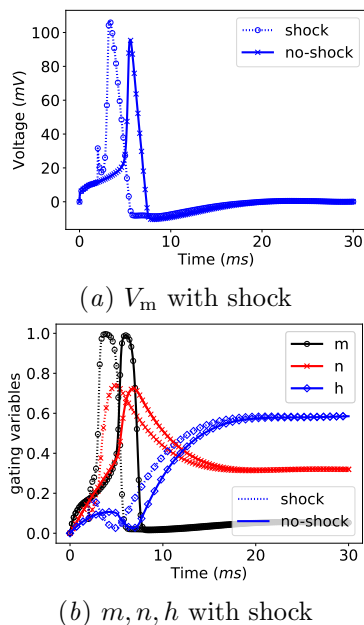
8

(a) V_{m} with shock

(b) m, n, h with shock

Figure 7: The NN handles a shock to the system dynamics and recovers the optimal trajectory.

6. Conclusion

This work explores the intersection of neuronal dynamics, optimal control theory, and machine learning. We focus on controlling neuronal dynamics characterized by the Hodgkin-Huxley model, a nonlinear system of differential equations that provides a detailed and complex representation of the spiking behavior of neurons. This has direct implications for biomedical applications like deep brain stimulation, where modulating neuronal dynamics to achieve therapeutic benefit with minimal energy expenditure is desired. Following existing approaches (Han et al., 2018; Kunisch and Walter, 2021; Onken et al., 2021b), we approximate the value function with a neural network and combine the PMP and HJB approaches to obtain feedback control policies. This allows for real-time computations of controls based on the current state and leads to overall improved robustness to disturbances, as demonstrated in our numerical experiments. This signifies a promising research direction in closed-loop DBS, for enabling a dynamic and personalized approach to treatment. By combining optimal control theory and machine learning, we highlighted new possibilities for interdisciplinary

research, particularly toward the design of adaptive neurotechnology and closed-loop systems. In future work, we expect to extend it to multi-compartment models featuring populations of neurons. These models comprise hundreds of neurons interacting across various neural structures, which pose exciting computational challenges that remain to be fully explored in these applications.

Acknowledgments

References

Mattia Arlotti, Manuela Rosa, Sara Marceglia, Sergio Barbieri, and Alberto Priori. The adaptive deep brain stimulation challenge. *Parkinsonism & related disorders*, 28:12–17, 2016.

Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

Dimitri P Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific Belmont, MA, 2019.

Matthew Botvinick, Jane X Wang, Will Dabney, Kevin J Miller, and Zeb Kurth-Nelson. Deep reinforcement learning and its neuroscientific implications. *Neuron*, 107(4):603–616, 2020.

Alexandre Boutet, Radhika Madhavan, Gavin JB Elias, Suresh E Joel, Robert Gramer, Manish Ranjan, Vijayashankar Paramanandam, David Xu, Jurgen Germann, Aaron Loh, et al. Predicting optimal deep brain stimulation parameters for parkinson's disease using functional mri and machine learning. *Nature communications*, 12(1):1–13, 2021.

Michael Breakspear. Dynamic models of large-scale brain activity. *Nature neuroscience*, 20(3):340–352, 2017.

Peter Brown. Oscillatory nature of human basal ganglia activity: relationship to the pathophysiology

of parkinson's disease. *Movement disorders: official journal of the Movement Disorder Society*, 18 (4):357–363, 2003.

Romain Carron, Antoine Chaillet, Anton Filipchuk, William Pasillas-Lépine, and Constance Hammond. Closing the loop of deep brain stimulation. *Frontiers in systems neuroscience*, 7:112, 2013.

William A Catterall, Indira M Raman, Hugh PC Robinson, Terrence J Sejnowski, and Ole Paulsen. The hodgkin-huxley heritage: from channels to circuits. *Journal of Neuroscience*, 32(41):14064–14073, 2012.

Aurelian Cernea and Hélène Frankowska. A connection between the maximum principle and dynamic programming for constrained control problems. *SIAM journal on control and optimization*, 44(2):673–703, 2005.

Gustavo Deco, Viktor K Jirsa, Peter A Robinson, Michael Breakspear, and Karl Friston. The dynamic brain: from spiking neurons to neural masses and cortical fields. *PLoS computational biology*, 4 (8):e1000092, 2008.

Lawrence C Evans. An introduction to mathematical optimal control theory version 0.2. *Lecture notes available at https://math.berkeley.edu/~evans/control.course.pdf*, 1983.

Ioannis Exarchos and Evangelos A. Theodorou. Stochastic optimal control via forward and backward stochastic differential equations and importance sampling. *Automatica J. IFAC*, 87:159–165, 2018. ISSN 0005-1098. doi: 10.1016/j.automatica.2017.09.004.

John E Fleming, Eleanor Dunn, and Madeleine M Lowery. Simulation of closed-loop deep brain stimulation control schemes for suppression of pathological beta oscillations in parkinson's disease. *Frontiers in neuroscience*, 14:166, 2020.

Wendell H. Fleming and H. Mete Soner. *Controlled Markov Processes and Viscosity Solutions*, volume 25 of *Stochastic Modelling and Applied Probability*. Springer, New York, second edition, 2006. ISBN 978-0387-260457; 0-387-26045-5.

Flavio Fröhlich and Sašo Jezernik. Feedback control of hodgkin–huxley nerve cell dynamics. *Control engineering practice*, 13(9):1195–1206, 2005.

Qitong Gao, Michael Naumann, Ilija Jovanov, Vuk Lesi, Karthik Kamaravelu, Warren M Grill, and Miroslav Pajic. Model-based design of closed loop deep brain stimulation controller using reinforcement learning. In *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (IC-CPS)*, pages 108–118. IEEE, 2020.

Wulfram Gerstner and Werner M Kistler. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

Jiequn Han, Arnulf Jentzen, and Weinan E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018.

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

Donald E Kirk. *Optimal control theory: an introduction*. Courier Corporation, 2004.

Christof Koch. *Biophysics of computation: information processing in single neurons*. Oxford university press, 2004.

Thomas Koeglsperger, Carla Palleis, Franz Hell, Jan H Mehrkens, and Kai Bötzel. Deep brain stimulation programming for movement disorders: current concepts and evidence-based strategies. *Frontiers in neurology*, 10:410, 2019.

Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

Dmitrii Krylov, Remi Tachet des Combes, Romain Laroche, Michael Rosenblum, and Dmitry V. Dylov. Reinforcement learning framework for deep brain stimulation study. *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, Jul 2020. doi: 10.24963/ijcai.2020/394.

Karl Kunisch and Daniel Walter. Semiglobal optimal feedback stabilization of autonomous systems via deep neural network approximation. *ESAIM: Control, Optimisation and Calculus of Variations*, 27:16, 2021.

Xingjian Li, Deepanshu Verma, and Lars Ruthotto. A neural network approach for stochastic optimal control. *arXiv preprint arXiv:2209.13104*, 2022.

Simon Little, Alex Pogosyan, Spencer Neal, Baltazar Zavala, Ludvic Zrinzo, Marwan Hariz, Thomas Foltynie, Patricia Limousin, Keyoumars Ashkan, James FitzGerald, et al. Adaptive deep brain stimulation in advanced parkinson disease. *Annals of neurology*, 74(3):449–457, 2013.

Meili Lu, Xile Wei, Yanqiu Che, Jiang Wang, and Kenneth A. Loparo. Application of reinforcement learning to deep brain stimulation in a computational model of parkinson's disease. *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society*, 28:339–349, 1 2020. ISSN 1558-0210. doi: 10.1109/TNSRE.2019. 2952637.

Anders Christian Meidahl, Gerd Tinkhauser, Damian Marc Herz, Hayriye Cagnan, Jean Debarros, and Peter Brown. Adaptive deep brain stimulation for movement disorders: the long road to clinical therapy. *Movement disorders*, 32(6): 810–819, 2017.

Paul Miller. *An introductory course in computational neuroscience*. MIT Press, 2018.

Ameer Mohammed, Richard Bayford, and Andreas Demosthenous. Toward adaptive deep brain stimulation in parkinson's disease: a review. *Neurodegenerative disease management*, 8(2):115–136, 2018.

S Nambi Narayanan and Sutha Subbian. Hh model based smart deep brain stimulator to detect, predict and control epilepsy using machine learning algorithm. *Journal of Neuroscience Methods*, 389: 109825, 2023.

Jorge Nocedal and Stephen J Wright. Interior-point methods for nonlinear programming. *Numerical Optimization*, pages 563–597, 2006.

Derek Onken, Levon Nurbekyan, Xingjian Li, Samy Wu Fung, Stanley Osher, and Lars Ruthotto. A neural network approach applied to multi-agent optimal control, 2021a.

Derek Onken, Levon Nurbekyan, Xingjian Li, Samy Wu Fung, Stanley Osher, and Lars Ruthotto. A neural network approach for real-time high-dimensional optimal control, 2021b.

Mahboubeh Parastarfeizabadi and Abbas Z Kouzani. Advances in closed-loop deep brain stimulation devices. *Journal of neuroengineering and rehabilitation*, 14(1):1–20, 2017.

Maxime Peralta, Pierre Jannin, and John SH Baxter. Machine learning in deep brain stimulation: A systematic review. *Artificial Intelligence in Medicine*, 122:102198, 2021.

Marcus Pereira, Ziyi Wang, Tianrong Chen, Emily Reed, and Evangelos Theodorou. Feynman-kac neural network architectures for stochastic control using second-order fbsde theory. In Alexandre M. Bayen, Ali Jadbabaie, George Pappas, Pablo A. Parrilo, Benjamin Recht, Claire Tomlin, and Melanie Zeilinger, editors, *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pages 728–738. PMLR, 10–11 Jun 2020.

Lev Semenovich Pontryagin. *Mathematical theory of optimal processes*. Routledge, 2018.

Benjamin Recht. A tour of reinforcement learning: The view from continuous control, 2018.

Lena Salfenmoser and Klaus Obermayer. Nonlinear optimal control of a mean-field model of neural population dynamics. *Frontiers in Computational Neuroscience*, 16, 2022.

Prasad Shirvalkar, Tess L Veuthey, Heather E Dawes, and Edward F Chang. Closed-loop deep brain stimulation for refractory chronic pain. *Frontiers in computational neuroscience*, 12:18, 2018.

Rosa Q So, Alexander R Kent, and Warren M Grill. Relative contributions of local cell and passing fiber activation and silencing to changes in thalamic fidelity during deep brain stimulation and lesioning: a computational modeling study. *Journal of computational neuroscience*, 32(3):499–519, 2012.

Richard S Sutton, Andrew G Barto, and Ronald J Williams. Reinforcement learning is direct adaptive optimal control. *IEEE control systems magazine*, 12(2):19–22, 1992.

Sacha Jennifer van Albada and Peter A Robinson. Mean-field modeling of the basal ganglia-thalamocortical system. i: Firing rates in healthy and parkinsonian states. *Journal of theoretical biology*, 257(4):642–663, 2009.

Jiang Wang, Liangquan Chen, and Xianyang Fei. Bi-furcation control of the hodgkin–huxley equations. *Chaos, Solitons & Fractals*, 33(1):217–224, 2007.

Thomas Wichmann and Mahlon R DeLong. Deep brain stimulation for movement disorders of basal ganglia origin: restoring function or functionality? *Neurotherapeutics*, 13:264–283, 2016.

Jiongmin Yong and Xun Yu Zhou. *Stochastic controls: Hamiltonian systems and HJB equations*, volume 43. Springer Science & Business Media, 1999.

Ying Yu, Xiaomin Wang, Qishao Wang, and Qingyun Wang. A review of computational modeling and deep brain stimulation: applications to parkinson's disease. *Applied mathematics and mechanics*, pages 1–22, 2020.

Appendix A. Background: Optimal Control theory

We present a brief background to Optimal Control (OC) theory relevant to this work. For a complete introduction to the topic, please refer to canonical texts such as Evans (1983); Kirk (2004); Yong and Zhou (1999).

OC is branch of applied mathematics that aims to find a control strategy or policy able to manipulate a dynamical system optimally over time in order to achieve a desired target state(s) of the system. OC theory has applications in various fields, including engineering, economics, and healthcare. Solving a control problem generally requires **minimizing a cost function** (performance criterion) associated with the dynamical system. Mathematically, given a dynamical system, defined over a fixed finite time-horizon T,

$$\frac{d\boldsymbol{z}}{dt}(t) = f\big(t, \boldsymbol{z}(t), \boldsymbol{u}(t)\big), \quad 0 \le t \le T$$
$$\boldsymbol{z}(0) = \boldsymbol{x}, \tag{14}$$

where $\boldsymbol{z}(t) \in \mathbb{R}^d$ represents the **state** of the system with dimension d at time t, with $\boldsymbol{x}$ being the initial state of the system. $\boldsymbol{u}(t) : [0,T] \to \mathbb{R}$ represents the external **control** provided as input by a controller (agent) at time t. The function $f : [0,T] \times \mathbb{R}^d \to \mathbb{R}^d$ governs the evolution of the state dynamics and is assumed to be *known*.

The primary objective of the OC problem is to find the control that drives the dynamical system toward a reference/target state, $\boldsymbol{z}^*$, while incurring the minimum possible cost. This involves specifying an **objective functional** (cost/utility function), J, as follows

$$J(t, \boldsymbol{z}, \boldsymbol{u}) = G\big(\boldsymbol{z}(T)\big) + \int_t^T L\big(s, \boldsymbol{z}(s), \boldsymbol{u}(s)\big)ds, \tag{15}$$

where $L : [0,T] \times \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}$ is the *Lagrangian* or *running* cost and $G : \mathbb{R}^d \to \mathbb{R}$ is the *terminal* cost. L accumulates the cost of controlling the system and expending energy at each time step. G penalizes the distance between the final state $\boldsymbol{z}(T)$ and the given target state of the system, $\boldsymbol{z}^*(T)$, at final time.

We are interested in finding a control(s) that incurs the minimal cost over all admissible controls $\boldsymbol{u} \in \mathcal{U}$,

$$\Phi\big(t, \boldsymbol{z}(t)\big) = \inf_{\boldsymbol{u}} J(t, \boldsymbol{z}, \boldsymbol{u}), \quad \text{s.t. (14)}. \tag{16}$$

A solution $\boldsymbol{u}^*$ incurring this minimal cost is called an optimal control, and the corresponding state variable $\boldsymbol{z}^*$ is called an optimal trajectory.

A.1. All-at-once Interior Point Method

The All-at-Once Interior Point Method (IPM) (Nocedal and Wright, 2006) aims to find the optimal control $u^*(t)$ by solving the following optimization problem:

$$\min_{\boldsymbol{u}(t), \boldsymbol{z}(t)} J(t, \boldsymbol{z}, \boldsymbol{u})$$
$$\text{s.t.}$$
$$\frac{d\boldsymbol{z}}{dt}(t) = f\big(t, \boldsymbol{z}(t), \boldsymbol{u}(t)\big) \tag{17}$$
$$g(\boldsymbol{z}(t), \boldsymbol{u}(t)) + s = 0$$
$$h(\boldsymbol{z}(T)) = 0$$
$$s \ge 0$$

This formulation introduces slack variables s to handle inequality constraints, and the barrier function is often used to incorporate them into the objective function. The IPM is solved iteratively, where at each iteration, the first-order necessary conditions are considered and barrier parameters are updated (Nocedal and Wright, 2006).