Distribution Agreement

In presenting this dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this dissertation. I retain all ownership rights to the copyright of the dissertation. I also retain the right to use in future works (such as articles or books) all or part of this dissertation.

_____ _____

Malvern Madondo Date

Applications of Closed-loop Control in Biomedical Interventions: From Neural Modulation to Diabetes Management

By

Malvern Madondo
Doctor of Philosophy

Computer Science and Informatics

Lars Ruthotto, PhD.
Advisor

Nicholas Au Yong, MD, Ph.D.
Committee Member

Razieh Nabi, Ph.D.
Committee Member

Carl Yang, Ph.D.
Committee Member

Accepted:

Kimberly Jacob Arriola, Ph.D.
Dean of the James T. Laney School of Graduate Studies

May 2, 2024

Date

Applications of Closed-loop Control in Biomedical Interventions: From Neural Modulation to Diabetes Management

By

Malvern Madondo
M.Sc., Emory University, GA, 2023
B.Sc. Honors, The College of St. Scholastica, MN, 2019
B.A., The College of St. Scholastica, MN, 2019

Advisor: Lars Ruthotto, PhD.

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Informatics
2024

Abstract

Applications of Closed-loop Control in Biomedical Interventions: From Neural Modulation to Diabetes Management

By Malvern Madondo

Dynamic adaptation of interventions to an individual's real-time medical condition has enormous potential in healthcare. Existing biomedical interventions, such as deep brain stimulation (DBS) for neurodegenerative disorders and insulin therapy for Type 1 diabetes (T1D), typically rely on predetermined treatment plans known as open-loop control. However, these static approaches often overlook the continuous changes in an individual's physiological state. This lack of real-time adaptation can lead to suboptimal treatment outcomes and an inability to manage fluctuating patient conditions effectively.

To address these limitations, we develop a closed-loop control framework that continuously adjusts treatment based on real-time physiological signals. We characterize neuromodulation and blood glucose regulation as control problems, leveraging a combination of machine learning (ML) and optimal control (OC) theory. Our approach integrates neural networks with classic OC techniques like Pontryagin's Maximum Principle and Hamilton-Jacobi-Bellman equations, enabling adaptive control that is robust to physiological variations – a highly desirable outcome in clinical settings. We utilize established models (Hodgkin-Huxley for neurons, Bergman's Minimal model for glucose-insulin) to simulate and optimize these closed-loop control strategies in a virtual environment, i.e., *in silico*. Additionally, we define relevant cost functions that quantify clinical objectives to guide the optimization process. We explore various control strategies, including using neural networks to learn optimal treatment adjustments in real-time, overcoming the limitations of open-loop approaches and potentially leading to improved clinical outcomes.

Applications of Closed-loop Control in Biomedical Interventions: From Neural Modulation to Diabetes Management

By

Malvern Madondo
M.Sc., Emory University, GA, 2023
B.Sc. Honors, The College of St. Scholastica, MN, 2019
B.A., The College of St. Scholastica, MN, 2019

Advisor: Lars Ruthotto, PhD.

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Computer Science and Informatics
2024

Acknowledgments

Completing this dissertation, and getting through graduate school, would not have been possible without the support and guidance of many people. I am deeply grateful to my advisor Lars Ruthotto for his mentorship throughout my graduate journey. Thanks for supporting my career interests and pursuits while giving me the freedom to explore my research interests. I would also like to thank Nick Au Yong, Razieh Nabi, and Carl Yang for participating in my dissertation committee and for providing insightful comments and valuable suggestions.

I am grateful for learning, working, and friendships with exceptional peers in Lars' group: Deepanshu Verma, Xingjian Li, Abbey Julian, Kelvin Kan, Nicole Yang, and Haley Rosso. Special thanks to Deepanshu for sharing not only his Costco benefits but also his expertise and assistance, and to Derek for his early guidance and for hosting me during my internship with Eli Lilly. I also extend my gratitude to the Emory CODES and DISC groups for creating a great research environment. Special shoutout to Elizabeth Newman, Yuanzhe Xi, and Jim Nagy. Many thanks to my friends and colleagues at Emory: Ramraj Chandradevan, Vishwanath Seshagiri, Tomilola Obadiya, Lakshmi Pullasseri, CSI cohortmates, and peers in the Math department.

Next, I would like to thank my brilliant collaborators and friends at Lilly and IBM Research – David Buchta, Stacy Hobson, Raya Horesh, Fearghal O'Donncha, Muneeza Azmat, Pablo Meyer Rojas, Mohamed Ghalwash, Zach Shahn, and Eli Sherman. Gratitude to my undergrad mentors Thomas Gibbons, Luther Qson, Kris Glesener, David Vosen, and Guanshen Ren for igniting my passion for applied research. Thanks to Lori Luing and Kevin Vaughan, among others, for enriching my college experience.

I would also like to acknowledge the financial support for this work offered by the 2021 Google PhD Fellowship in Computational and Cognitive Sciences, including mentorship from Marc G. Bellemare. Thanks to several programs that enabled my academic progression and professional development activities throughout the years,

including Philanthropic Ventures Foundation and CRA IDEALS.

I am also grateful to Raphael Mumba, Petra Ann Gumbs, Basia Walenkiewicz, Yeabsira Kefeni, Mera Petros, Laura Lastra, Daniela Moreno, Nicole Ciernia, Denisa Lalikova, Isaac Wakiro, Leul Solomon, Rick Troy, and the Finnemann family, for their continued support and friendship. Immense gratitude to my Zimbabwean village and USAP community for their unwavering support: Allen and Diana Chaparadza, Tino and Paida Chikate, Tafadzwa Mashamba, Nicole Moyo, TaRu Makuvatsine, Clive Matsika, Simosenkosi Nkomboni, Joel Tshite, Maakwe Cumanzala, Learnmore Jeremiah, Tariro Mwendamberi, Orwell Madovi, Glander Madondo, Yevedzo Chipangura, Angel Sibanda, Margaret Gurumani, Challe and Tanya, Takudzwa Munjanja, Takunda and Violet Chazovachii, Munashe Mugonda, Edward and Vimbai Chikwana, Gibson Nene, Emmanuel Chigutsa, Chipo Kambarami, Natasha Tsandukwa, Rufaro Chirongoma, Prosper Feya, Valentine Chiwome, Stephen Tsvaki, Claudius Mundoma, and Rebecca Zeigler Mano, among others.

Finally, my deepest gratitude to my family, especially my siblings Pascal, Priscilla, Vimbai, Tino, and Waraidzo, and my partner Chiza Mwinde, for their unwavering love and encouragement. Special thanks to Tawanda Chibaya, Lydia Chindaro, Julius Madondo, and the Murashi, Murefu, Kanda, and Marembo families.

This dissertation is dedicated to my late parents, Kennedy and Emiliah. Their unwavering love and exemplary lives built the foundation for this and many other achievements. I wish they could be here to witness them all. To every individual who played a part in shaping this dissertation and enriching my academic journey, thank you.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Machine Learning (ML) and Optimal Control (OC) theory play pivotal roles across diverse disciplines, shaping advancements in healthcare, robotics, finance, aerospace, manufacturing, and energy management. ML, a subfield of Artificial Intelligence (AI), involves the development of algorithms and models capable of learning from data and making data-driven predictions or decisions, without requiring explicit programming for each specific task [16, 43]. OC involves determining the sequence of control actions that optimizes a predefined objective function for a given dynamical system. These controls achieve the desired system behavior while adhering to system dynamics and constraints [35, 54].

In biomedical applications like deep brain stimulation (DBS) and blood glucose regulation, the synergy between ML and OC presents opportunities for adaptive solutions in managing neurodegenerative disorders and metabolic conditions. Individuals grappling with medication-resistant neurodegenerative disorders, such as Parkinson's disease (PD), and those managing Type 1 diabetes (T1D) often rely on predetermined or fixed treatment regimens (*open-loop* control). However, these static approaches may not adequately adapt to their changing physiological needs or responses to therapy [23, 30, 45, 66].

Closed-loop systems, where interventions dynamically adapt to real-time changes in a patient's clinical state, become pivotal in addressing the central challenge of guiding individuals toward wellness [2, 23]. This intricate process targets optimal health outcomes, weighing factors such as minimal side effects, time, cost, and other relevant criteria [11, 38, 44, 97]. This dissertation, grounded in well-established ML and OC theory ideas, seeks to develop closed-loop biomedical systems for real-time optimization of *neuromodulation* (used interchangeably with DBS) and *glycemic control*. This enables adaptive interventions/therapy (also called dynamic treatment regimens), potentially improving efficacy, reducing side effects, and significantly enhancing patients' quality of life.

1.1 Problem Statement

Neurological disorders, such as PD, and metabolic disorders like T1D present substantial challenges for both patients and healthcare providers. Existing *open-loop* treatments often yield suboptimal outcomes [23, 66]. In DBS, a common open-loop approach continuously delivers fixed stimulation, relying on manual determination of settings. This approach lacks adaptability and real-time responsiveness, leading to suboptimal outcomes and necessitating multiple clinical visits [21]. In the context of blood glucose control for T1D, the intricacies of insulin administration require closed-loop systems like artificial pancreas systems for effective glycemic control [11, 45]. Utilizing ML and OC theory, this research seeks to develop closed-loop solutions for both DBS and glycemic control. By addressing the limitations observed in current clinical interventions, we strive to advance computational approaches in biomedical interventions.

1.2 Research Overview

We leverage ML and OC theory to *in silico* biomedical interventions, specifically emphasizing neuromodulation for neurodegenerative disorders and closed-loop blood glucose control.

As a computational exploration devoid of direct patient involvement, our research employs computational models such as the Hodgkin-Huxley (HH) model [49] for neuronal dynamics and Bergman's minimal (BM) model [12] for glucose-insulin dynamics. This *in silico* approach not only ensures efficiency and cost-effectiveness but also minimizes risks associated with real-world interventions. While datasets, prevalent in many ML applications, provide valuable insights, our model-based approach allows for a more controlled and customizable simulation, capturing intricate dynamics. This approach facilitates the development of patient-specific insights and enables us to explore a wide range of scenarios, optimizing closed-loop systems for neurodegenerative disorders and glycemic control.

1.2.1 Limitations

We aim to highlight the potential of ML and OC theory for improving current interventions across diverse biomedical applications. However, it is important to clarify that our research excludes direct involvement in clinical trials, patient intervention data collection, or the use of medical devices such as pulse generators or insulin pumps. Instead, we rely on simulations based on established models. While this approach allows for controlled experiments, it may not fully capture the variability inherent in real-world biological systems or patient responses. Additionally, certain computational model parameters are adopted or estimated from existing literature, introducing a factor that might influence the broader applicability of our findings. Therefore, our research primarily serves as a proof-of-concept for future work integrating these

advanced techniques into clinical applications.

1.2.2 Contributions

Addressing the optimization challenges in biomedical interventions offers exciting research opportunities at the intersection of computational health, ML, and control systems. Our exploration of numerical approaches integrating ML and OC aims to develop adaptive closed-loop policies capable of delivering real-time therapeutic remedies in response to dynamic changes in a patient's clinical state. Through this endeavor, we hope to unravel the mutual relationship between dynamical systems, learning, and control.

Our main contributions include the following:

- We seek to improve neuromodulation and glycemic control with closed-loop systems that adapt in real time. This ensures optimal responses to changing physiological states and robustness against unexpected shocks or disturbances in system dynamics.

- We address the limitations of conventional open-loop control methods, which lack adaptability to changing conditions. By framing neuromodulation and glycemic control as control problems, we simulate their dynamics *in silico* using established computational models. We also define relevant cost functions that quantify clinical objectives to guide the optimization process.

- We explore various control strategies, including neural network-based value function approximation, to enhance adaptability and responsiveness, overcoming the limitations of open-loop approaches.

Overall, building on well-established ML and OC theory ideas, we develop closed-loop solutions for real-time interventions within these biomedical applications. This dissertation incorporates content derived from the author's published work [69] and

nonproprietary projects conducted during research internships at IBM Research and Eli Lily. This includes applications of reinforcement learning, a subclass of ML, to T1D management, and a neural ordinary differential equation (neural ODE [28]) approach to glycemic control, respectively.

1.3 Dissertation Outline

We now summarize the contents of the central chapters of this dissertation:

Chapter 2 provides a mathematical foundation for solving control problems in neuromodulation and glycemic control. We explore closed-loop system formulation, review OC theory, and discuss value function approximation using neural networks. We conclude with a brief commentary on choosing computational models, suitable performance criteria, and method(s) to address the identified control problem.

Chapter 3 introduces a neural network-based approach for closed-loop neuromodulation. We simulate neuronal dynamics with the HH model and develop a learning-based control scheme based on Pontryagin's Maximum Principle and Hamilton-Jacobi-Bellman equations.

Chapter 4 extends our exploration to glucose-insulin dynamics, applying Bergman's minimal model. We demonstrate control schemes on a virtual patient model, ranging from simplistic Proportional-Integral-Derivative (PID) controllers to advanced schemes utilizing neural ODEs, such as reinforcement learning.

We conclude with a summary of this dissertation and a discussion of potential research directions in **Chapter 5**.

Chapter 2

Hitchhiker's Guide to Solving Control Problems

2.1 Introduction

Control problems are widespread in various fields such as biomedicine, economics, engineering, and manufacturing, where system dynamics must be regulated for improved performance or outcomes. In biomedicine, control problems encapsulate a wide range of applications. In this work, we consider numerical approaches for deterministic, finite-dimensional optimal control problems in two medical applications: neuromodulation for deep brain stimulation (DBS) [31, 34, 41, 120] and blood glucose control for Type 1 diabetes (T1D) management [2, 45, 91]. We can tackle such problems using a variety of arsenal from Machine Learning (ML) and Optimal Control (OC) theory.

Our starting point will be to translate neuromodulation and glycemic control (blood glucose regulation) into mathematical terms, formulating them into control problems defined over a finite time-horizon $T > 0$ and characterized by a standard

objective functional,

$$J(t, \boldsymbol{z}, \boldsymbol{u}; \boldsymbol{\psi}) = \int_t^T L\left(s, \boldsymbol{z}(s), \boldsymbol{u}(s)\right) ds + G\left(\boldsymbol{z}(T)\right), \tag{2.1}$$

and constrained by a nonlinear dynamical system,

$$\frac{d\boldsymbol{z}}{dt}(t) = f(t, \boldsymbol{z}(t), \boldsymbol{u}; \boldsymbol{\psi}), \quad \boldsymbol{z}(t) = \boldsymbol{x}, \quad t \in [0, T]. \tag{2.2}$$

Here, the function $f : [0, T] \times \mathbb{R}^d \times \mathbb{R}^a \to \mathbb{R}^d$ models the evolution of the states $\boldsymbol{z}(t) \in \mathbb{R}^d$ in response to the control inputs $\boldsymbol{u} : [0, T] \to U \subset \mathbb{R}^a$. $\boldsymbol{x} \in \mathbb{R}^d$ denotes the initial state of the system. The parameter $\boldsymbol{\psi} \in \eta$ represents problem-specific physiological properties. The function $L : [0, T] \times \mathbb{R}^d \times \mathbb{R}^a \to \mathbb{R}$ represents the intermediate or *running cost*, and $G : \mathbb{R}^d \to \mathbb{R}$ represents the *terminal cost* in the control objective in Equation (2.1). These costs represent penalties specific to each application. For instance, the running cost may include penalties for deviations of the current blood glucose from a target glucose value or the energy consumed when delivering electrical stimulation for DBS. The terminal cost often encodes the desired behavior on termination, e.g., reaching a specific state.

2.1.1 Control Formulation

In clinical settings, when an individual with a health condition (characterized by pathological dynamics) seeks treatment, their current state serves as the starting point $\boldsymbol{x}$, and the physician endeavors to transition it, via a set of controls $\boldsymbol{u}$, to a more satisfactory or "healthy" state [2, 59, 72, 99]. The primary goal is to determine the optimal control strategy for the biological system, guided by the cost function J, which incorporates penalties for excessive control inputs or undesired states at specific times.

In mathematical terms, we aim to find an optimal control/policy that minimizes

the overall cost, i.e.,

$$\begin{aligned} \Phi(t, \boldsymbol{z}(t); \boldsymbol{\psi}) &= \inf_{u} J\left(t, \boldsymbol{z}, \boldsymbol{u}; \boldsymbol{\psi}\right), \\ \text{s.t. } & \frac{d\boldsymbol{z}}{dt}(t) = f\left(t, \boldsymbol{z}(t), \boldsymbol{u}(t); \boldsymbol{\psi}\right), \quad t \in [0, T] \\ & \boldsymbol{z}(t) = \boldsymbol{x}. \end{aligned} \tag{2.3}$$

This is widely known as the *value function*. It contains all the information about the solution and represents the minimum cost-to-go for any initial state $(t, \boldsymbol{x})$. The control achieving this minimum value is referred to as the optimal control, u_t^*, and the corresponding state trajectory is known as the optimal trajectory, $\boldsymbol{z}^*$. Solving Equation (2.3) globally remains a challenging task, especially when the state dimension $d \geq 4$. This phenomenon is commonly referred to as the *Curse of Dimensionality* (CoD), where computational costs grow exponentially with the state dimension [60].

We outline the dynamics and cost function considered for neuromodulation in Chapter 3 and for glycemic control in Chapter 4, presenting a general framework integrating ML and OC. This framework aims to determine control inputs that optimize problem-specific objectives, accounting for system dynamics and constraints. For brevity, we only consider aspects of ML and OC theory relevant to formulating closed-loop solutions integrating state feedback to these applications. We refer to [35, 54] for a detailed introduction to control theory. This text covers various aspects of control problems and different strategies for solving them. We recommend [98, 126] for a detailed overview of ML techniques and neural networks. [42, 56, 73] provide a detailed introduction to computational neuroscience, including mathematical models of neuronal dynamics, and [30] provides a comprehensive commentary on *artificial pancreas* or closed-loop systems in glucose-insulin control.

2.1.2 Ties to Biomedical Applications

Biomedical applications like neuromodulation and glycemic control target complex biological systems often modeled using differential equations (Equation (2.2)). While substantial effort goes into creating mathematical models (f) for these systems, our research leverages existing models and modifies them as needed to gain insights specific to the application (details in corresponding chapters, see an overview in Table 2.1).

Translating control theory to biomedicine comes with several challenges [6, 11, 67]. First, mathematical models representing underlying pathologies in neuromodulatory and glucose-insulin systems are often approximations, and their parameters can vary significantly between patients. Additionally, defining a universally applicable cost function is difficult due to the diverse responses individual patients may exhibit to controls.

Furthermore, control inputs must adhere to problem-specific constraints to ensure safety and feasibility. For instance, pulse generators in DBS systems may limit the maximum stimulation delivered to prevent adverse effects. Similarly, insulin pumps incorporate safety features to prevent excessive dosing and mitigate the risk of hypoglycemia. The set of *admissible controls*, $\mathcal{U}$, is typically treated as constant and independent of the state $\boldsymbol{z}$ and time t. This assumption is crucial for simplifying the modeling process and facilitating the application of control strategies within these systems.

In both DBS and glycemic control, external inputs (such as electrical current and insulin) evolve the dynamical system towards a terminal state at a given time. These controls can be *open-loop*, involving predetermined and independent of the system's state, or *closed-loop*, adjusted based on real-time feedback from the system. Closed-loop control is particularly significant as it adapts to individual needs and changes in system dynamics. This motivates our work in extending its application to biomedicine.

	$\boldsymbol{z}$	$\boldsymbol{u}$	f
DBS	membrane potential, gating variables	stimuli/electrical current	Hodgkin-Huxley model [49]
T1D	BG level (optionally, insulin-on-board)	insulin dose	Bergman's Minimal model [12]

Table 2.1: Overview for each biomedical application – neuromodulation in DBS (Chapter 3) and glycemic control in T1D management (Chapter 4)

2.2 Closed-loop Control in Biomedical Systems

Extensive research efforts have focused on developing closed-loop algorithms that continuously monitor system states, such as neuronal activity in DBS or blood glucose levels in T1D management, and adjust control inputs to maintain desired physiological conditions or achieve therapeutic goals. We refer to [38, 48, 97] for approaches in DBS and to [11, 30, 45] for a detailed exposition on glycemic control.

We will now highlight a few select methods for solving control problems like Equation (2.3). Control strategies can generally be categorized based on their scope, with prominent categories including the following:

1. **Local solution methods**. These methods compute solutions based on a fixed initial state $\boldsymbol{x}$ or within specific regions of the state space, ensuring computational efficiency. However, when faced with a different initial state or unexpected changes in the system's state due to shocks or disturbances, these methods may require re-computation or adaptation of the control policy. In clinical applications, local methods may be suitable for *open-loop* control, where clinicians predetermine interventions based on the individual's current condition.

2. **Global solution methods**. In contrast, approaches in this category aim for control solutions applicable across the entire space and yield them in *closed-loop* or *feedback form*. This is ideal in many real-world applications as it enables real-time adaptation to changes in system dynamics, eliminating the need for any re-computations. However, these methods are prone to the CoD, limiting their utility in clinical scenarios.

3. **Semi-global solution methods**. These methods combine aspects of both local and global solution methods. They tailor solutions to specific regions of the state space while maintaining broader applicability. This is crucial for clinical interventions, providing a balance between computational efficiency and the ability to find near-optimal solutions in varying clinical scenarios. Chapter 3 illustrates the merits of the semi-global solution approach in neuromodulation, also extended to glycemic control in Chapter 4.

2.2.1 Classical Control Approaches

Traditional control theory offers a systematic framework for designing algorithms that can improve the performance of dynamical systems in real-time. Two closely related approaches to solving Equation (2.3) involve: applying *Pontryagin's Maximum Principle* [84] and solving the *Hamilton-Jacobi-Bellman equation* [54, 124]. A critical component of both approaches is the Hamiltonian function, denoted by $\mathcal{H}$. This function combines the system dynamics in Equation (2.2), the cost function in Equation (2.1), and the control input, $\boldsymbol{u}$, as follows,

$$\mathcal{H}(t, \boldsymbol{z}, \boldsymbol{p}, \boldsymbol{u}; \boldsymbol{\psi}) = -L\left(t, \boldsymbol{z}, \boldsymbol{u}\right) - \boldsymbol{p}^{\top}\left[f\left(t, \boldsymbol{z}, \boldsymbol{u}; \boldsymbol{\psi}\right)\right], \tag{2.4}$$

where $\boldsymbol{p}$ is an *adjoint* variable associated with the state $\boldsymbol{z}$ at time t. For a wide variety of control problems, Equation (2.4) admits a closed-form solution [81], making it ideal for the biomedical applications under consideration.

Pontryagin's Maximum Principle

The Pontryagin's Maximum Principle (PMP) is a *local solution* method that provides first-order necessary conditions for the optimal control and state trajectory [26, 39,

62, 81, 84]. The optimality conditions for every $s \in [0, T]$ are given by the following:

$$\begin{cases} \partial_s \boldsymbol{p}(s) = \nabla_{\boldsymbol{z}} \mathcal{H}\big(s, \boldsymbol{z}^*(s), \boldsymbol{p}(s), \boldsymbol{u}^*(s); \boldsymbol{\psi}\big), \\ \partial_s \boldsymbol{z}^*(s) = -\nabla_{\boldsymbol{p}} \mathcal{H}\big(s, \boldsymbol{z}^*(s), \boldsymbol{p}(s), \boldsymbol{u}^*(s); \boldsymbol{\psi}\big) \\ \boldsymbol{z}^*(t) = \boldsymbol{x}, \ \boldsymbol{p}(T) = \nabla_{\boldsymbol{z}} G\big(\boldsymbol{z}^*(T)\big), \end{cases} \tag{2.5}$$

while the optimal control can be recovered in closed-form as $\boldsymbol{u}^*(s) = \boldsymbol{u}^*\left(s, \boldsymbol{z}^*(s), \boldsymbol{p}(s); \boldsymbol{\psi}\right)$ [81, 84].

A major appeal of the PMP is that it is not affected by the Curse of Dimensionality, remaining computationally efficient even as the dimensionality of the system increases. As such, it can be practically applied to problems with large state spaces. However, solving Equation (2.5) yields a solution based on the local behavior of the system, i.e., constrained to specific initial data, $(s, \boldsymbol{x})$. If this data changes or external shocks disrupt the optimized trajectory, recomputing the solution with the new data becomes necessary, potentially hindering its adaptability in dynamic environments.

Hamilton-Jacobi-Bellman equation

Global solution methods like the Hamilton-Jacobi-Bellman (HJB) equation consider the entire system dynamics, offering robustness and adaptability to varying conditions [124]. This broader applicability makes HJB and similar approaches particularly suitable for biomedical applications where system behaviors may vary widely and require holistic control strategies.

The HJB equation provides necessary and sufficient conditions for optimality in control theory [54, 124]. It is typically formulated as a nonlinear, second-order partial differential equation, involving the value function Φ, Hamiltonian $\mathcal{H}$, and terminal

cost $G(T)$:

$$\begin{cases} -\partial_s \Phi\big(s, \boldsymbol{z}; \boldsymbol{\psi}\big) + \sup_{\boldsymbol{u}} \mathcal{H}\big(s, \boldsymbol{z}, \nabla_{\boldsymbol{z}} \Phi\big(s, \boldsymbol{z}; \boldsymbol{\psi}\big), \boldsymbol{u}; \boldsymbol{\psi}\big) = 0, \quad \forall s \in [0, T) \\ \Phi\big(T, \boldsymbol{z}(T); \boldsymbol{\psi}\big) = G\left(\boldsymbol{z}(T)\right). \end{cases} \tag{2.6}$$

With ready access to the value function, Φ, and its gradient, $\nabla_{\boldsymbol{z}} \Phi$, the optimal control, $\boldsymbol{u}^*$, can be efficiently computed and recovered online feedback form as

$$\boldsymbol{u}^*\left(s, \boldsymbol{z}^*(s), \nabla_{\boldsymbol{z}} \Phi\big(s, \boldsymbol{z}^*(s); \boldsymbol{\psi}\big); \boldsymbol{\psi}\right) \in \arg\max_{\boldsymbol{u}} \mathcal{H}\big(s, \boldsymbol{z}^*(s), \nabla_{\boldsymbol{z}} \Phi\big(s, \boldsymbol{z}^*(s); \boldsymbol{\psi}\big), \boldsymbol{u}(s); \boldsymbol{\psi}\big). \tag{2.7}$$

The HJB equation is closely related to Dynamic Programming or Bellman's Principle of Optimality in Reinforcement Learning (RL) [13]. This connection underscores the foundation of various RL techniques, including value iteration [10], policy iteration [51], policy gradient [111], and Q-learning [109, 119]. Indeed, OC theory and RL have a lot in common [14, 89, 110] and we leverage these ties in the next chapters. We provide a concise review of RL in Subsection 2.3.

2.2.2 Learning-based Control

A key assumption is that a closed-form solution (eqn. (2.7)) to the optimal control problem exists, which is true for many control problems including our specific applications in neuromodulation and glycemic control. This provides a framework for deriving optimal control from the value function Φ, assuming smoothness and differentiability [81, 115]. However, solving the HJB equation (2.6) for high-dimensional systems ($d \geq 4$) becomes computationally infeasible, necessitating function approximation [60, 81].

Neural networks have achieved wide success in different applications, largely due to their universal function approximation capabilities and general advances in ML [36, 46, 60, 62, 79, 81, 83]. We consider a two-stage approach that involves offline approximation of the value function Φ with a neural network, followed by an online

computation of the optimal control using the feedback form (2.7) [60, 81].

Value Function Approximation with Neural Networks

We parameterize the value function as

$$\Phi_{\boldsymbol{\theta}}(\boldsymbol{y}) = \boldsymbol{w}^{\top}\mathrm{NN}(\boldsymbol{y};\boldsymbol{\theta}_{\mathrm{NN}}), \tag{2.8}$$

for space-time inputs $\boldsymbol{y} = (s, \boldsymbol{z}(s); \boldsymbol{\psi}) \in \mathbb{R}^{d+p_{\boldsymbol{\psi}}+1}$, where d is the dimension of state vector $\boldsymbol{z}$, and $p_{\boldsymbol{\psi}}$ represents the dimension of the problem-specific parameters, $\boldsymbol{\psi}$, influencing the dynamics (e.g., conductances in the Hodgkin-Huxley model [49]). $\mathrm{NN}(\boldsymbol{y};\boldsymbol{\theta}_{\mathrm{NN}}) : \mathbb{R}^{d+p_{\boldsymbol{\psi}}+1} \to \mathbb{R}^m$ can be any standard neural network architecture, such as a feedforward or residual neural network, with $(d + p_{\boldsymbol{\psi}} + 1)$ input features and m output features. $\boldsymbol{\theta}$ contains all trainable weights: $\boldsymbol{w} \in \mathbb{R}^m$ and $\boldsymbol{\theta}_{\mathrm{NN}} \in \mathbb{R}^p$, where p denotes the number of network parameters.

Training a neural network is equivalent to learning the values of $\boldsymbol{\theta}$ that approximately solve a control problem of the form:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{\psi}\sim\eta,\boldsymbol{x}\sim\rho}\Big\{ J\left(t, \boldsymbol{z}, \boldsymbol{u}; \boldsymbol{\psi}\right) + \gamma_1 P_{\mathrm{HJB},\boldsymbol{\psi}}(\boldsymbol{z}) + \gamma_2 c_{\mathrm{HJB}}(T)\Big\}, \\ \text{s.t. } \frac{d\boldsymbol{z}}{dt}(t) = f\big(t, \boldsymbol{z}(t), \boldsymbol{u}(t); \boldsymbol{\psi}\big), \;\; \boldsymbol{z}(t) = \boldsymbol{x}, \end{aligned} \tag{2.9}$$

where the precise definition of the cost function J depends on the task (neuromodulation or glycemic control). This objective functional penalizes deviations from the HJB equation along the state trajectories with the penalty term,

$$P_{\mathrm{HJB},\boldsymbol{\psi}}(\boldsymbol{z}) = \int_t^T \Big| H\big(s, \boldsymbol{z}, \nabla_{\boldsymbol{z}}\Phi_{\boldsymbol{\theta}}\big(s, \boldsymbol{z}(s); \boldsymbol{\psi}\big); \boldsymbol{\psi}\big) - \partial_s \Phi_{\boldsymbol{\theta}}\big(s, \boldsymbol{z}(s); \boldsymbol{\psi}\big)\Big| ds$$

and an additional penalty at terminal time, $c_{\text{HJB}}(T)$, defined as

$$c_{\text{HJB}} = \left|\Phi_{\boldsymbol{\theta}}\big(T, \boldsymbol{z}(T); \boldsymbol{\psi}\big) - G\big(\boldsymbol{z}(T)\big)\right|.$$

The hyperparameters $\gamma_1, \gamma_2 \geq 0$ balance minimization of the HJB penalization along the state trajectories and at the final time, respectively. In the learning process, the optimal control is computed through the feedback form provided in Equation (2.7), hence, incorporating the model information [60, 81, 115].

2.3 Reinforcement Learning and Optimal Control

We note that the control formulation and strategies discussed so far share significant common ground with Reinforcement Learning (RL) methods, particularly in their aim to find optimal policies for maximizing long-term rewards or achieving specific goals [89, 110]. RL is a sub-field of ML that involves a decision-making *agent* interacting over time with an *environment* to achieve a goal [109]. At each timestep, t, the agent chooses an action, u_t, given the current state of the environment, $\boldsymbol{z}_t$, and receives an evaluative numerical reward r. The environment then transitions to a new state, $\boldsymbol{z}'$, and this process repeats until the end of the time-horizon T or some termination criteria are specified. Throughout, the agent aims to maximize its long-term cumulative reward (expected return) over the trajectory rather than simply choosing actions that yield immediate rewards.

Both OC and RL rely on policies to dictate system behavior based on its current state [89]. In OC, the control policy finds the optimal control input for a given state. Similarly, the RL agent's policy, often denoted as π, selects actions based on the observed environment. This policy can be deterministic (mapping states directly to actions) or stochastic (mapping states to action probabilities). Central to learning in RL is the trade-off between *exploitation* and *exploration*. An agent learns to exploit

actions that were most effective in maximizing rewards in past experiences while balancing the exploration of new options to make better long-term strategies [109].

2.3.1 Value Functions in RL

The value function serves as a critical foundation in RL. It estimates the expected cumulative cost when starting in state $\boldsymbol{z}$ under policy π:

$$\begin{aligned}\Phi_\pi(\boldsymbol{z}) &= \mathbb{E}_{\pi_t(\boldsymbol{z}_t;\boldsymbol{\theta}),\boldsymbol{\psi}\sim\eta}\left[\sum_{k=0}^{N} r_k \mid \boldsymbol{z}_k = \boldsymbol{z}\right] \\ &= \mathbb{E}_{\pi_t(\boldsymbol{z}_t;\boldsymbol{\theta}),\boldsymbol{\psi}\sim\eta}\Big[J\left(s_k, \boldsymbol{z}, \boldsymbol{u}; \boldsymbol{\psi}\right) \coloneqq \Delta s \sum_{i=k}^{N} L(s_i, \boldsymbol{z}_i, \boldsymbol{u}_i) + G(\boldsymbol{z}_N)\Big],\end{aligned} \tag{2.10}$$

where the immediate reward, r_t, is carefully designed to match the negated running cost L in Equation (2.1) as follows

$$r_k = \begin{cases} -\Delta s\, L(s_k, \boldsymbol{z}_k, \boldsymbol{u}_k), & k < N \\ -\Delta s\, L(s_N, \boldsymbol{z}_N, \boldsymbol{u}_N) - G(\boldsymbol{z}_N), & k = N \end{cases}.$$

with step size $\Delta s = (T - t)/N$.

Equation (2.10) is known as the **state-value function** for policy π and it is analogous to the OC value function Φ in Equation (2.3). Both functions represent the long-term value of being in a particular state under a specific control policy. To align with the OC problem setup outlined in Section 2.1.1, our objective shifts from maximizing the expected return, as typically done in standard RL settings, to minimizing the cost function J. Thus, the state-value function Φ_π estimates the expected negated long-term cost achievable from a specific state $\boldsymbol{z}$ under policy π. In this setting, the policy π aims to minimize the total long-term cost J by accumulating negated immediate rewards, denoted by r_t.

By itself, the state-value function may lack sufficient information for decision-

making as it only evaluates states without considering actions [109]. This potentially creates uncertainty regarding the optimal action to take in a given state to maximize long-term rewards. In light of this, the **state-action value function**, or **Q-function**, denoted $Q_\pi(u, \boldsymbol{z})$, is more informative as it evaluates the expected return of taking an action $\boldsymbol{u}$ in state $\boldsymbol{z}$, and then following policy π [109, 119]:

$$Q_\pi(\boldsymbol{z}, \boldsymbol{u}) = \mathbb{E}_{\pi_t(\boldsymbol{z}_t;\boldsymbol{\theta})}\left[\sum_{k=0}^{N} r_k \mid \boldsymbol{z}_k = \boldsymbol{z}, \boldsymbol{u}_k = \boldsymbol{u}\right]. \tag{2.11}$$

The optimal value function for a policy π is given by

$$Q_\pi^*(\boldsymbol{z}, \boldsymbol{u}) = \max_\pi Q_\pi(\boldsymbol{z}, \boldsymbol{u}). \tag{2.12}$$

Given Q_π^*, the optimal action (control) can be obtained via

$$u^* \in \arg\max_{\boldsymbol{u}} Q_\pi^*(\boldsymbol{z}, \boldsymbol{u}). \tag{2.13}$$

The goal of an RL agent is to find a near-optimal policy π, i.e., $Q_\pi(\boldsymbol{z}, \boldsymbol{u}) \approx Q_\pi^*(\boldsymbol{z}, \boldsymbol{u})$.

Interestingly, both RL and OC involve approximating the value function, often utilizing neural networks for function approximation. This overlap underscores the potential for cross-pollination of ideas and techniques between the two fields, paving the way for advancements in both domains [14, 89, 98, 110]. In RL, the value function traditionally assumes a tabular representation, known as the *lookup table*, where each state or state-action pair has an entry [111, 109]. However, as many real-world problems have enormous state and/or action spaces, this approach becomes inadequate as the state dimension increases–a characteristic of the Curse of Dimensionality. Consequently, differentiable function approximators like neural networks have become popular in the deep RL community due to their ability to obtain compact representations that generalize across state or state-action pairs [64]. In these settings, the policy π is

parameterized with a neural network and the main objective is to learn the network weights $\boldsymbol{\theta}$ such that $u_t \sim \pi(\boldsymbol{z}_t; \boldsymbol{\theta})$ [75].

2.3.2 Strategies for solving RL problems

The landscape of algorithms in RL is vast. RL algorithms generally fall into the following broad categories:

1. *model-based* approaches – require explicit knowledge of the underlying dynamics f, Equation (2.2). They can either (a) learn a model of the environment offline from historical data, and then simulate state trajectories and compute a control policy $u \sim \pi$ online without further interactions (commonly seen in control algorithms, 2.2), or (b) learn the system dynamics online through continuous interactions with the environment, extracting valuable information about the underlying dynamics from the collected data (occurs prominently in some RL algorithms).

2. *model-free* approaches – learn directly from data or interactions with the environment without explicitly modeling the system dynamics f. These approaches typically learn online and update their policy based on experiences gained during interactions.

A comprehensive taxonomy of these methods exceeds our scope, with each category presenting numerous challenges that researchers strive to address. Importantly, implementing model-based RL becomes increasingly challenging and computationally expensive for complex systems, rendering it less practical. Conversely, while model-free RL methods alleviate some computational burdens, they often face difficulties in effectively learning from data or interactions, especially in environments with high dimensionality or sparse rewards. Overcoming these challenges remains a focal point in

RL research. We refer to the following reviews for an in-depth discussion of challenges and approaches in RL [7, 63].

We now discuss some of the popular RL methods, particularly of interest to our applications. Two primary approaches exist for solving RL problems: those based on value functions and those based on policy search. Additionally, a hybrid method known as the actor-critic approach combines elements of both value functions and policy search.

Value Function-based Methods – directly estimate the value-function and include methods such as Q-learning [119]. Q-learning seeks to learn the value function using temporal difference [109, 119], an update rule based on the Bellman Equation [10]. However, Q-learning relies on tabular representations, making it impractical for real-world problems with large state spaces. A modern improvement, Deep Q-Learning [75], addresses this issue by using neural networks to approximate the Q-function, $Q_\pi(\boldsymbol{z}, \boldsymbol{u})$, enabling it to handle problems with high-dimensional state spaces efficiently.

Policy Search Methods – directly aim to find an optimal policy without explicitly estimating the value function. One prominent example is the REINFORCE algorithm [122], a type of policy gradient method. These methods adjust the parameters $\boldsymbol{\theta}$ of the policy $\pi(\boldsymbol{z}_t; \boldsymbol{\theta})$ directly using optimization algorithms, such as stochastic gradient descent, to improve the expected return. Policy gradient methods are particularly advantageous in scenarios with high-dimensional or continuous state spaces where traditional value-based methods like Q-learning struggle due to the Curse of Dimensionality.

Actor-Critic Methods – offer a hybrid approach combining elements of the value function and policy search methods. They maintain and update both a value function (the critic) and a policy (the actor) iteratively. The actor learns by using feedback

from the critic to improve its policy, while the critic estimates the value of selected actions. This approach has been shown to optimize policies efficiently in complex environments. Popular policy gradient methods include Deep Deterministic Policy Gradient (DDPG)[105, 64] and Proximal Policy Optimization (PPO) [101].

2.4 Conclusion

By bridging the gap between mathematical theory and practical applications in biomedicine, this chapter sets the stage for the subsequent exploration of ML and OC methodologies in specific biomedical contexts. We framed neuromodulation (deep brain stimulation) and glycemic control (blood glucose regulation) as control problems, identifying key elements like system dynamics and cost functions. We emphasized the transition from traditional open-loop control, where interventions (stimulation or insulin dosing) are predetermined or fixed, to closed-loop control methods. This approach allows for adaptive interventions that can dynamically respond to changing physiological states in real-time, which is highly desirable in these biomedical applications.

We provided an overview of various control methods, including local, global, and semi-global solution approaches, showcasing their applicability and potential benefits in addressing specific challenges within neuromodulation and glycemic control. Additionally, we gave a succinct overview of some of the relevant concepts in Machine Learning (ML), including value function approximation using neural networks. We also introduced Reinforcement Learning (RL), highlighting its shared elements with Optimal Control(OC) such as value functions and the use of neural networks for function approximation. Both ML and RL offer valuable tools for tackling biomedical control problems.

In subsequent chapters, we will explore each biomedical application separately,

drawing on a range of techniques from both ML/RL and OC theory. By combining insights from these disciplines, we aim to develop effective control strategies tailored to the unique challenges posed by neuromodulation and glycemic control.

Chapter 3

Closed-loop Neuromodulation via Machine Learning and Optimal Control

In this chapter, we frame the biomedical challenge of finding optimal stimulation parameters as a control problem, consequently setting the stage for developing adaptive feedback control systems using Machine Learning (ML) and Optimal Control (OC). We simulate neuronal activity with the biophysically-accurate Hodgkin-Huxley (HH) model [49]. We consider a model-based approach that leverages neural networks in combination with classical control approaches such as the Pontryagin's Maximum Principle (PMP) [84] and Hamilton-Jacobi-Bellman (HJB) equations [10, 54, 124] to learn optimal stimulation strategies/policies and achieve closed-loop feedback control.

3.1 Introduction

The dynamic modulation of neuronal dynamics through ML and OC is a thriving area of research as exemplified by multiple studies including [17, 31, 68, 69, 76, 77, 90, 117]. It strives to neutralize pathological neural behavior, with the ultimate goal of alleviating

motor symptoms associated with neurological conditions such as Parkinson's disease, epilepsy, and chronic pain [57, 72, 48, 121].

In clinical settings, Deep Brain Stimulation (DBS) remains the gold standard procedure for modulating pathological neural activity [21, 40, 57, 72, 121, 102, 125]. Currently, DBS operates on an open-loop paradigm, where highly trained clinicians iteratively select and adjust stimulation parameters based on observed patient symptoms through a trial-and-error process [17, 44, 65, 86]. However, this approach is not responsive to changes in a patient's clinical state and rarely yields optimal parameter settings, which necessitates follow-up visits for parameter adjustment [4, 48, 67, 116].

A key clinical objective is to maximize the therapeutic benefit of stimulation i.e., alleviating motor symptoms, while reducing stimulation side effects. Closed-loop DBS seeks to achieve this by learning optimal stimulation parameters and adjusting them in real-time according to the patient's electrophysiological state [6, 15, 23, 48, 66, 97]. Central to closed-loop DBS are algorithms that can be deployed in implantable DBS devices to precisely modulate neural activity in response to physiological signals and restore function [58, 82, 107]. These algorithms act in response to neuronal signals, such as local field potentials, recorded from specific brain regions.

3.2 Problem Formulation

We seek to find an optimal control (policy) that minimizes a cost functional J in Equation (2.1) and satisfies the HH-based nonlinear system dynamics,

$$\begin{aligned} \frac{d\boldsymbol{z}}{dt}(t) &= f\big(t, \boldsymbol{z}(t); \boldsymbol{\psi}\big) + \boldsymbol{e}_1 \boldsymbol{u}(t), \\ \boldsymbol{z}(0) &= \boldsymbol{x}, \quad \boldsymbol{e}_1 = [1,0,0,0]^\top, \quad t \in [0,T], \end{aligned} \tag{3.1}$$

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. $\boldsymbol{z}$ comprises the neuron's membrane potential (Figure 3.1a), V_{m}, and three gating variables (Figure 3.1b) representing probabilities of sodium activation (m), potassium activation (n), and sodium inactivation (h),

- the control variable $\boldsymbol{u}(t) : [0, T] \to \mathcal{U} \subset \mathbb{R}$ represents the external electrical current/stimulus provided as input by a controller at any given time t,
- the function $f : [0, T] \times \mathbb{R}^4 \to \mathbb{R}^4$ describes the evolution of the state $\boldsymbol{z}$ in response to controls $\boldsymbol{u}$, as per the HH model (3.2),
- $\boldsymbol{\psi} = [\bar{g}_{\mathrm{Na}}, \bar{g}_{\mathrm{K}}, \bar{g}_{\mathrm{l}}] \sim \eta$ consists of three terms representing the maximum conductance of the sodium, $\bar{g}_{\mathrm{Na}}$, potassium, $\bar{g}_{\mathrm{K}}$, and leak currents, $\bar{g}_{\mathrm{l}}$, respectively. These are selected randomly throughout the parameter space, η, following Equation (3.8), to simulate varying pathological conditions,
- the cost functional J is the sum total of the running cost $L : [0, T] \times \mathbb{R}^4 \times \mathbb{R} \to \mathbb{R}$ and the terminal cost $G : \mathbb{R}^4 \to \mathbb{R}$. Details for both are provided in 3.2.2.

The goal is to approximate the value function, Φ, such that solutions can be computed readily for new contexts.

3.2.1 Neuronal Dynamics

The state dynamics, f, evolve according to the HH model, which considers currents flowing through the neuronal membrane via various ion channels, including sodium (Na), potassium (K), and leak channels (denoted l, encompassing all other ions with slower dynamics like chloride ions) [41, 55, 71, 118]. The HH model characterizes the electrophysiological activity of neurons through a system of first-order differential

equations, expressed as follows:

$$f\big(t, \boldsymbol{z}(t); \boldsymbol{\psi}\big) = \begin{bmatrix} -\frac{1}{C_{\mathrm{m}}}\left(\psi_0 z_1^3 z_3 (z_0 - E_{\mathrm{Na}}) + \psi_1 z_2^4 (z_0 - E_{\mathrm{K}}) + \psi_2 (z_0 - E_{\mathrm{l}})\right) \\ \alpha_m\big(z_0\big)(1 - z_1) - \beta_m\big(z_0\big) z_1 \\ \alpha_n\big(z_0\big)(1 - z_2) - \beta_n\big(z_0\big) z_2 \\ \alpha_h\big(z_0\big)(1 - z_3) - \beta_h\big(z_0\big) z_3 \end{bmatrix}. \tag{3.2}$$

Here, C_{m} is the membrane capacitance, and the variables $E_{\mathrm{Na}}, E_{\mathrm{K}},$ and E_{l} represent the equilibrium potentials of the corresponding ion channels. The functions α_j and β_j, $j \in \{1, 2, 3\}$ defined in Table 3.1, are voltage-dependent rate functions for each of the gating variables z_1, z_2, and z_3 (often labeled as m, n, and h, respectively.)

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

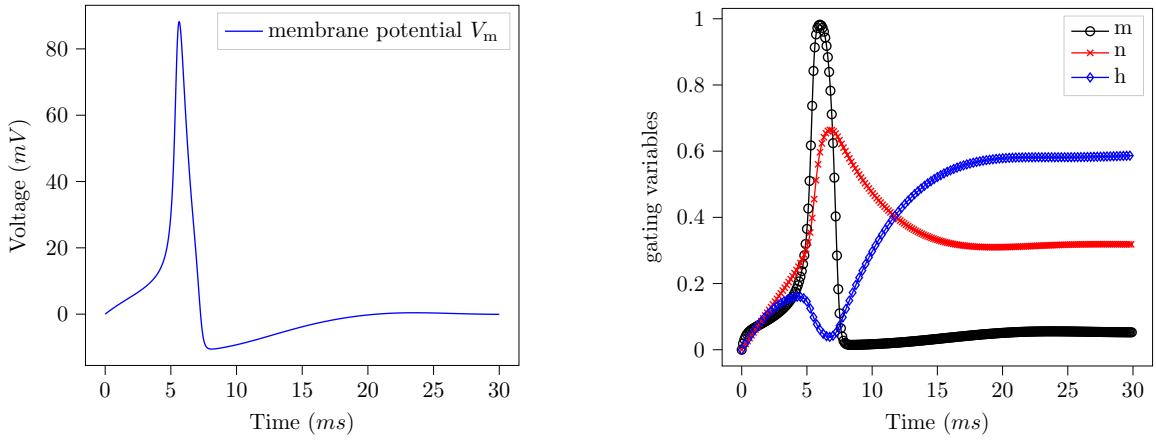
Table 3.1: Parameters of the gating variables z_1, z_2, and z_3 in the HH equations. The functions α and β depend only on the voltage, z_0.

Pathological Activity

The HH model realistically captures the electrical activity of neurons, including spiking behavior, through a system of nonlinear differential equations [24, 49]. The real intrigue of the HH model arises when we consider pathological activity, which stems from deviations from normative neuronal behavior. 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 propagation of a spike (known as the action potential) in Figure 3.1. Depolarization is driven by Na

ion influx via activated sodium channels, while repolarization is driven by potassium channel activation and sodium channel inactivation [25, 49, 55].

The general flow of ions like Na and K across the neuron's membrane plays a crucial role in transmitting electrical signals. On the other hand, pathological activity contributes to abnormal patterns of electrical signaling in neurons, marked by rapid and repetitive firing of action potentials in Figure 3.2. This behavior can manifest in various neurological disorders such as Parkinson's disease, epilepsy, and chronic pain. We refer to [42, 56, 73] for detailed explanations of neuronal dynamics.



(a) Normal action potential

(b) Normal gating variables

Figure 3.1: Evolution of the membrane potential V_m and gating variables m, n, and h of normal HH equations, with no controls/stimuli, i.e., $\boldsymbol{u}(t) = 0$, $\forall\, t > 0$, and initial state $\boldsymbol{x} = [0, 0, 0, 0]^\top$.

Neuronal dynamics governed by the HH equations exhibit complex behavior influenced by a range of parameters, $\boldsymbol{\psi}$. Controlling these dynamics, especially in pathological states, is challenging due to inherent electrophysiological variability and parameter uncertainty. To overcome these challenges, one approach leverages amortization [3, 103, 114], solving Equation (2.1) across a broad parameter distribution ($\boldsymbol{\psi} \sim \eta$), aiming to achieve robust control despite parameter uncertainty. However, as this approach is under development, we will primarily present results for the standard approach with fixed parameters.

3.2.2 Neuromodulation Cost Function

We now turn to the design choice of what loss to optimize so that the controller learns an optimal intervention. For the biomedical application under consideration, we seek to control pathological neuronal activity in order to effectively restore normal functioning. In mathematical terms, 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. This characterization of the cost function is prevalent in many DBS control applications [8, 34, 38, 41, 77]. Figure 3.1 shows the state trajectory of typical HH neuronal activity. Figure 3.2 shows the evolution of the HH model under pathological conditions.

For a given parameter vector $\boldsymbol{\psi} \sim \eta$, the cost function for the control problem (2.1) comprises the running cost, $L : [0, T] \times \mathbb{R}^4 \times \mathbb{R} \to \mathbb{R}$, given by

$$L(s, \boldsymbol{z}(s), \boldsymbol{u}(s)) = \frac{\lambda_1}{2}\|\boldsymbol{u}(s)\|^2 + \frac{\lambda_2}{2}\|\boldsymbol{z}(s) - \boldsymbol{z}^*(s)\|^2, \quad t \leq s \leq T, \tag{3.3}$$

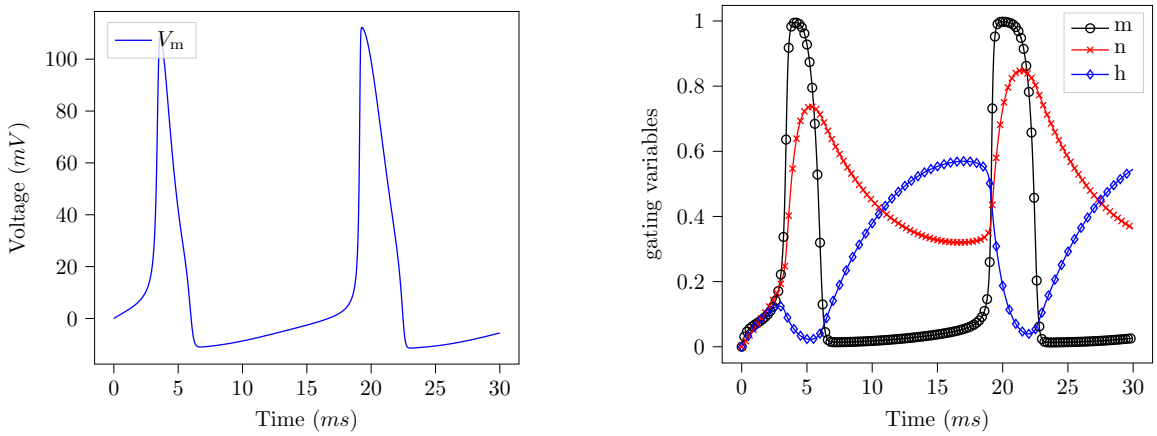
with coefficients $\lambda_1 = 1$ and $\lambda_2 = 200$ weighting the importance of the energy and tracking term, respectively. The terminal cost, $G : \mathbb{R}^4 \to \mathbb{R}$, is defined as

$$G(\boldsymbol{z}) = \frac{1}{2}\|\boldsymbol{z}(T) - \boldsymbol{z}^*(T)\|^2. \tag{3.4}$$

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. Both these costs can be used to specify desired clinical objectives or target solutions.

Closed-Form Solution With this choice of the objective functional, we can now derive the feedback form for the optimal control, based on (2.7).

The explicit feedback form for the neuromodulation problem can be obtained by



(a) Pathological action potential

(b) Pathological gating variables

Figure 3.2: 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 $\boldsymbol{x} = [0, 0, 0, 0]^\top$.

taking the gradient of the Hamiltonian with respect to the control variable $\boldsymbol{u}$ and setting the resulting expression to zero as follows,

$$\begin{aligned} \frac{\partial \mathcal{H}}{\partial u} &= \lambda_1 \boldsymbol{u}(s)^\top + \boldsymbol{p}(s)^\top \boldsymbol{e}_1 = 0, \quad \forall s \in [t, T] \\ \Rightarrow \boldsymbol{u}(s) &= -\frac{1}{\lambda_1} \boldsymbol{e}_1^\top \nabla_{\boldsymbol{z}} \Phi\big(s, \boldsymbol{z}^*(s); \boldsymbol{\psi}\big), \end{aligned} \tag{3.5}$$

based on Equations (2.4) and (3.3), with the adjoint state $\boldsymbol{p}(s)$ given by

$$\boldsymbol{p}(s) = \nabla_{\boldsymbol{z}} \Phi\big(s, \boldsymbol{z}^*(s); \boldsymbol{\psi}\big). \tag{3.6}$$

The adjoint equation plays a crucial role in establishing the connection between the PMP approach (Section 2.2.1) and the HJB equation (Section 2.2.1) [26, 39]. Provided that the value function Φ and its gradient $\nabla_{\boldsymbol{z}} \Phi$ are available, the optimal controls can be easily recovered in real-time for any space-time inputs $(s, \boldsymbol{z})$ and a given parameter $\boldsymbol{\psi}$. This is ideal for clinical applications such as DBS where swiftly computing controls for different times or states in real time is highly desirable.

3.3 Model-based Approach to Neuromodulation

In this section, we explore computational methods to determine controls aligning with our objective function (2.1). We begin by considering a local solution method resembling the standard open-loop DBS approach. This involves an iterative All-at-once Interior Point Method where system dynamics, the cost function, and all constraints across the entire time horizon are solved simultaneously. However, as an open-loop approach akin to the local methods discussed in Section 2.2, this approach necessitates recomputation of the solution whenever $\boldsymbol{\psi}$ and/or $\boldsymbol{x}$ changes, rendering prior solutions obsolete. Although global solution methods eliminate the need for recomputing controls and provide real-time solutions, they are susceptible to the Curse of Dimensionality, posing challenges in high-dimensional spaces.

To address these challenges, we adopt a semi-global solution approach that combines the function-approximating abilities of neural networks with well-established control theory techniques. This hybrid approach, stemming from [81], is tailored to effectively handle high-dimensional control problems in this setting. Several research efforts, including [46, 60, 62, 81, 94, 115], have demonstrated the utility of this approach in various applications.

3.3.1 Value Function Approximation with Neural Networks

The main idea involves approximating the solution to the parametric control problem in (2.1) offline using a neural network, followed by computing the control online using the feedback form given by Equation (2.7). We build on Section 2.2.2 formulation, parameterizing the value function based on Equation (2.8).

We represent $\mathrm{NN}(\boldsymbol{y}; \boldsymbol{\theta}_{\mathrm{NN}}) : \mathbb{R}^{d+p_{\boldsymbol{\psi}}+1} \to \mathbb{R}^m$ with a residual neural network (ResNet)

architecture [47]:

$$\begin{aligned} \boldsymbol{a}_0 &= \sigma\left(\boldsymbol{K}_0 \boldsymbol{y} + \boldsymbol{b}_0\right) \\ \boldsymbol{a}_{i+1} &= \boldsymbol{a}_i + h\sigma\left(\boldsymbol{K}_{i+1} \boldsymbol{a}_i + \boldsymbol{b}_{i+1}\right) \\ \mathrm{NN}(\boldsymbol{y}; \boldsymbol{\theta}_{\mathrm{NN}}) &= \boldsymbol{a}_{\mathrm{M}-1} + h\sigma\left(\boldsymbol{K}_{\mathrm{M}} \boldsymbol{a}_{\mathrm{M}-1} + \boldsymbol{b}_{\mathrm{M}}\right). \end{aligned} \tag{3.7}$$

where $0 \leq i \leq \mathrm{M} - 2$, with M being the depth of the network. The parameter $\boldsymbol{\theta}_{\mathrm{NN}} = (\boldsymbol{K}_0, \ldots, \boldsymbol{K}_{\mathrm{M}}, \boldsymbol{b}_0, \ldots, \boldsymbol{b}_{\mathrm{M}})$ comprises the weight matrix $\boldsymbol{K}_0 \in \mathbb{R}^{m \times (d + p_{\boldsymbol{\psi}} + 1)}$, and $\{\boldsymbol{K}_0, \ldots, \boldsymbol{K}_{\mathrm{M}}\} \in \mathbb{R}^{m \times m}$ and the bias vector, $\boldsymbol{b}_i \in \mathbb{R}^m \; \forall i$.

The activation function $\sigma : \mathbb{R} \to \mathbb{R}$ is applied element-wise and adds non-linearity to the network, allowing it to represent complex features in the data. Common activation functions include hyperbolic tangent, sigmoid, or rectified linear units (ReLU). We opt for $\sigma(\boldsymbol{x}) = \log\left(\exp(\boldsymbol{x}) + \exp(-\boldsymbol{x})\right)$, the anti-derivative of the hyperbolic tangent. $h > 0$ is a fixed step size. While we can approximate the value function offline using a neural network, finding the appropriate network architecture and weights, $\boldsymbol{\theta}$, can be quite challenging. The choice of these hyperparameters largely depends on the problem and approach.

Learning Problem

We aim to find network weights $\boldsymbol{\theta}$ such that the parameterized value function $\Phi_{\boldsymbol{\theta}}$ globally represents the value function for every given space-time input $\boldsymbol{y}$. However, achieving this for reasonable problem sizes ($d \geq 4$) becomes impractical due to the Curse of Dimensionality. Therefore, we adopt a model-based, semi-global solution approach, enforcing this property in a subset of the space-time domain [62, 115].

To learn the parameters $\boldsymbol{\theta}$ of the NN, we begin by sampling parameters $\boldsymbol{\psi}$ from the parameter space η and initial states $\boldsymbol{x}$ from a distribution ρ. Subsequently, we approximately solve the minimization problem defined in Equation (2.9).

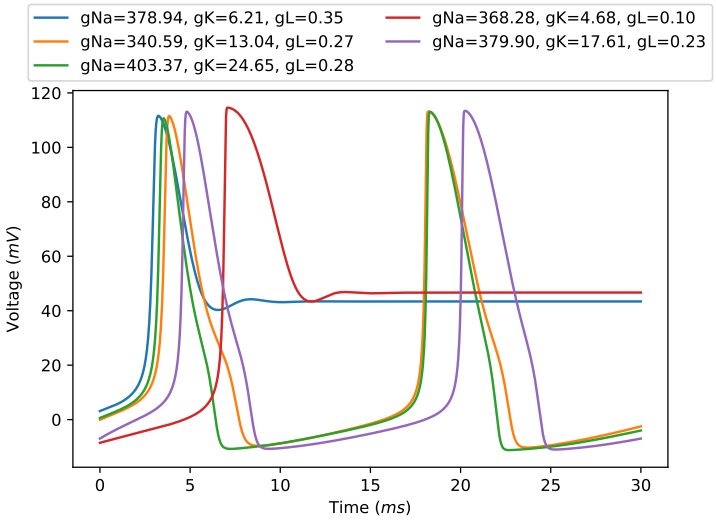


Figure 3.3: Perturbed variables in different pathological settings of the HH model.

3.4 Numerical Experiments

In this section, we examine two stimulation strategies: a baseline controller that applies the all-at-once Interior Point Method (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).

3.4.1 Neuromodulatory Effects

The HH model, Equation (3.1), is known to be stiff which may lead to increased computational costs as small step sizes would be required to accurately capture the system dynamics [20, 42, 55]. We simulate the HH model for both normal and pathological conditions, based on parameters in Table 3.2. We source the default values for simulating normative neuronal behavior from Hodgkin and Huxley [49], listed under the “normal" neuronal activity category in Table 3.2. To characterize pathological activity, we also obtain the corresponding default values and randomly distort them by sampling from a distribution η as specified in Table 3.2 and outlined

below:

$$\boldsymbol{\psi} = \left[\bar{g}_{\mathrm{Na}} \sim \mathcal{N}(\mu_{\mathrm{Na}}, \sigma^2_{\mathrm{Na}}),\ \bar{g}_{\mathrm{K}} \sim \mathcal{N}(\mu_{\mathrm{K}}, \sigma^2_{\mathrm{K}}),\ \bar{g}_{\mathrm{l}} \sim \mathcal{N}(\mu_{\mathrm{l}}, \sigma^2_{\mathrm{l}})\right], \tag{3.8}$$

where μ_q and σ^2_q, for ion channel $q \in [\mathrm{Na}, \mathrm{K}, \mathrm{l}]$, are the mean and variance of the corresponding parameter distribution.

Perturbing the HH parameters $\boldsymbol{\psi}$ has the effect of distorting the ion flow across the neural membrane, generating abnormal spikes or causing the rapid and repetitive firing of action potentials (Figure 3.3). This approach draws inspiration from a study by [77] 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.

Parameter	Normal	Pathological
C_{m}	1.0	1.0
$\bar{g}_{\mathrm{Na}}$	120.0	$\mathcal{N}(380.0, 10^2)$
$\bar{g}_{\mathrm{K}}$	36.0	$\mathcal{N}(36.0, 1^2)$
$\bar{g}_{\mathrm{l}}$	0.3	$\mathcal{N}(0.3, 0.32^2)$
E_{Na}	115.0	115
E_{K}	-12.0	-12.0
E_{l}	10.613	10.613

Table 3.2: Nominal parameter values of the HH model under normal and pathological conditions.

3.4.2 Optimal Control of Neuronal Dynamics

We aim to develop control strategies for disrupted neuronal systems. These strategies should achieve two primary goals: 1) restore the system's normal function, and 2) minimize the energy needed for control while mitigating the risk of the system entering pathological states. Such strategies would be invaluable in countering pathological neural activity in closed-loop DBS devices [66, 74, 82, 85, 97, 102].

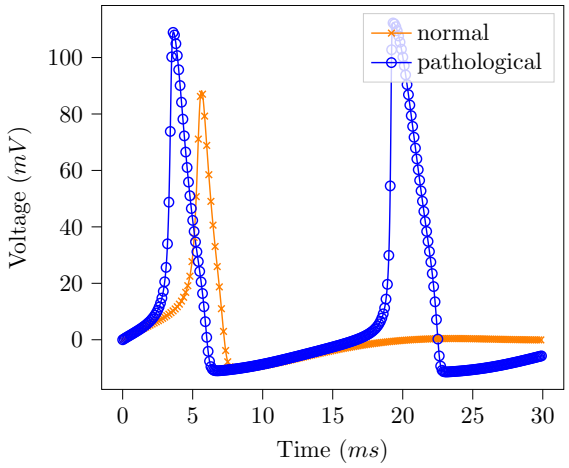


Figure 3.4: This figure compares action potentials generated by a neuron in normal and pathological states. Control strategies aim to regulate the pathological dynamics and restore the action potential profile towards a more normative pattern.

All-at-once Optimization for a Single Initial State

We provide a comparable local solution method that solves the OC problem for a fixed parameter $\boldsymbol{\psi}$ and initial state $\boldsymbol{z}(0) = \boldsymbol{x}$ and serves as our baseline. The All-at-Once Interior Point Method (IPM) [80] aims to find the optimal control $u^*(t)$ by solving the following optimization problem. $\forall\, t \in [0, T]$:

$$\min_{\boldsymbol{u}(t), \boldsymbol{z}(t)} J\left(t, \boldsymbol{z}, \boldsymbol{u}; \boldsymbol{\psi}\right)$$

s.t.

$$\frac{d\boldsymbol{z}}{dt}(t) = f\big(t, \boldsymbol{z}(t), \boldsymbol{u}(t); \boldsymbol{\psi}\big), \quad \boldsymbol{z}(t) = \boldsymbol{x}$$

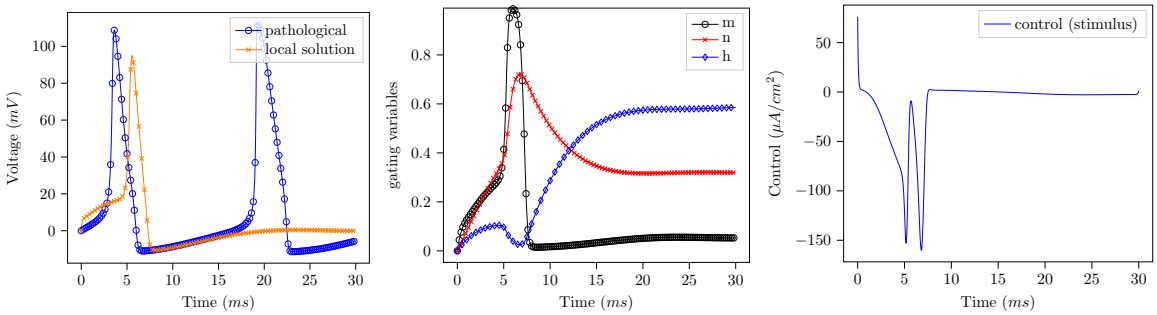
$$g\left(\boldsymbol{z}(t), \boldsymbol{u}(t)\right) + s = 0, \quad s \geq 0,$$

$$h\left(\boldsymbol{z}(T)\right) = 0,$$

This formulation introduces slack variables s and equality constraints, $g\left(\boldsymbol{z}(t), \boldsymbol{u}(t)\right)$ and $h\left(\boldsymbol{z}(T)\right)$. The IPM is solved iteratively, where at each iteration, the first-order necessary conditions are considered, and barrier parameters are updated [80].

Figure 3.5 demonstrates how the all-at-once Interior Point method can restore

normal behavior to a pathological HH system. The method applies control stimuli (3.5c) to drive the action potentials (3.5a) and gating variables (3.5b) towards their normal operating range.



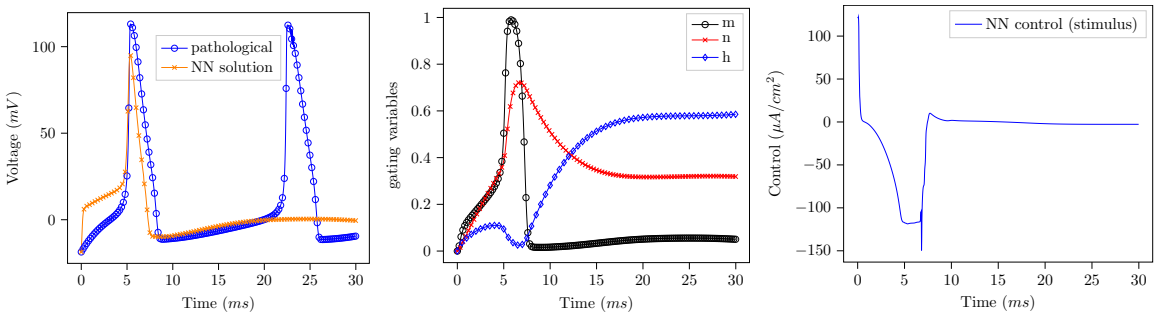
(a) Membrane potential (b) Gating variables (c) Controls

Figure 3.5: Local solution all-at-once Interior Point Method restores normal behavior to a pathological HH system.

NeuralHJB: Neural Network-based Control

We implement a simple recurrent neural network (NN) controller architecture with 2 hidden layers of 64 neurons each. This network is trained using the ADAM optimizer and a learning rate of 0.005. This design choice balances expressive power with computational efficiency, making it suitable for real-world applications. During training, the optimal control is computed through the feedback form provided in Equation (2.7). This approach leverages crucial information about the system dynamics and derivatives guided by control theory, leading to a more informed and effective control strategy.

The offline step is perhaps the most computationally expensive as it requires learning the parameters or weights of the neural network that effectively approximate Φ. After offline training, the online step uses the neural network to generate real-time controls in feedback form. The online step is usually less computationally expensive than offline training, making it suitable for real-time applications. Figure 3.6 showcases how the semi-global approach applies the learned controls or stimuli to nudge the pathological dynamics back to a healthy state.



(a) Membrane potential (b) Gating variables (c) Controls

Figure 3.6: NeuralHJB, a semi-global solution approach combining neural networks and HJB equations, effectively steers pathological HH system towards normal behavior.

Suboptimality The local solution method directly finds the control strategy that minimizes the cost function for a fixed initial state $\boldsymbol{x} = [0, 0, 0, 0]$. We establish the solution obtained using this baseline approach as a benchmark for optimality. To assess the performance of the NeuralHJB approach, we evaluate its solution for the given initial state and quantify its suboptimality compared to the baseline solution. Ideally, we want suboptimality to be close to zero, indicating that the approach is nearly optimal or that its performance is very close to the best achievable by the local solution method.

Figure 3.7 compares the suboptimality (normalized cost difference) of the NeuralHJB approach in two settings: standard (fixed HH parameters $\boldsymbol{\psi}$) and amortized ($\boldsymbol{\psi} \sim \eta$), for various initial conditions (represented by $\boldsymbol{x} + \xi$, where ξ in $[-40, 40]$. We note that the NeuralHJB approach in the standard setting achieves lower suboptimality than the amortized setting. While our initial experiments with the amortized NeuralHJB solver did not yield optimal results, its inherent adaptability – stemming from both universal approximation properties and flexibility of learning across diverse parameters ($\boldsymbol{\psi} \sim \eta$) – suggest the amortized approach as a promising avenue for further research and performance improvement.

Table 3.3 compares the performance of both methods based on the cost function. The local method, considered the optimal solution or ground truth, achieves the

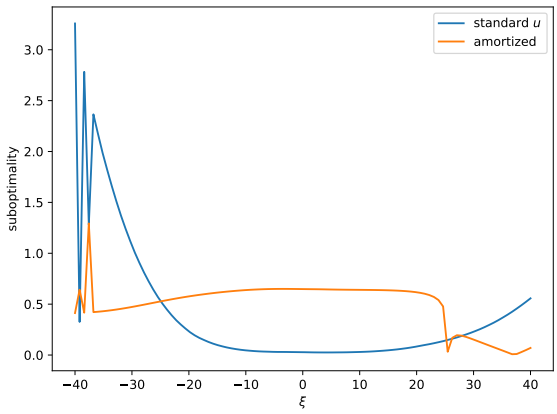


Figure 3.7: Suboptimality comparison between NeuralHJB approach and local solution method across various initial conditions $\boldsymbol{x} + \xi$, where ξ in $[-40, 40]$, for standard (fixed $\boldsymbol{\psi}$) and amortized ($\boldsymbol{\psi} \sim \eta$) controls.

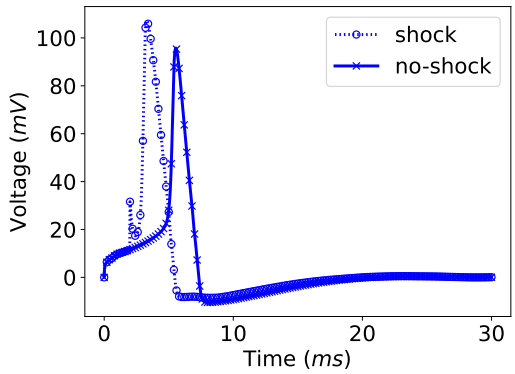
minimum cost by definition. We observe that the NN-based controller achieves near-optimal performance, which is particularly impressive considering it tackles a control problem for a larger state space compared to the baseline method.

	running cost (L)	terminal cost (G)	total
IPM	45725.6861	0.0279	45725.7140
NN	46890.1119	0.0088	46890.1207

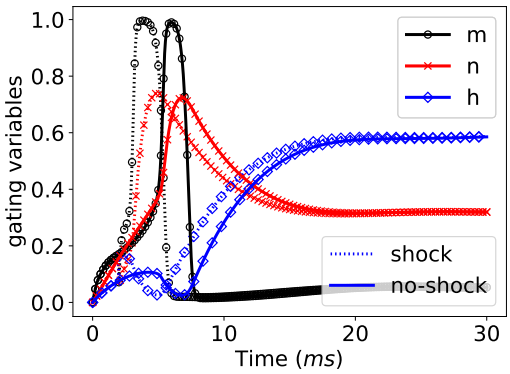
Table 3.3: Running and terminal costs for single instance shown in figures 3.5 and 3.6.

Robustness to Shocks Beyond performance metrics, a critical factor for controlling pathological neural behavior is the control strategy's robustness to unexpected changes or disturbances in the system dynamics. This is especially important for real-world applications where unforeseen states may arise and require immediate response. Since the local solution method employs non-adaptive open-loop DBS strategies, limiting its ability to respond to shocks, we will only evaluate the robustness of the semi-global NN approach to assess its suitability for real-time applications.

We introduce minor perturbations or shocks to the HH system (Equation (3.1)) and observe in Figure 3.8 that the semi-global NN method maintains optimal behavior despite shocks/perturbations in the system dynamics, demonstrating its robustness.



(a) Membrane potential V_{m} of HH state dynamics with shock added.



(b) Gating variables m, n, h of HH dynamics with shock added]

Figure 3.8: The semi-global NeuralHJB approach effectively recovers the optimal trajectory after the system is disrupted.

3.5 Summary

This chapter bridges the gap between neuronal dynamics, optimal control theory, and machine learning to address a critical challenge in deep brain stimulation (DBS). Building on elements from the Chapter 2, we cast neuromodulation as a control problem, aiming to normalize pathological neuronal activity. Leveraging the biophysically-accurate Hodgkin-Huxley (HH) model, we present a model-based approach for learning control policies. We define a cost function for DBS that balances the energy consumption of the control strategy with maintaining the desired neuronal states.

We highlighted current limitations in conventional open-loop DBS control strategies, exemplified by an all-at-once optimization scheme that solves the control problem for a fixed initial state $\boldsymbol{x}$. While offering optimal solutions for a specific initial state, this approach requires recomputing the solution for any changes in state or parameters. In contrast, global solution methods yield solutions in real-time but require sampling the entire state space, becoming computationally expensive for high-dimensional problems. To leverage the strengths of both approaches, we highlight NeuralHJB, a semi-global control method that combines the theoretical foundation of the Hamilton-Jacobi-Bellman (HJB) equation with the flexibility and learning

capabilities of neural networks [46, 60, 81]. The HJB equation provides crucial insights into the value function, while neural networks effectively learn and optimize this function for real-time control in high dimensions.

Our findings demonstrate the effectiveness of the NeuralHJB approach. It exhibits near-optimal performance for fixed parameters and remains effective across a broad range of parameter values $\boldsymbol{\psi} \in \eta$. Additionally, it shows robustness to unexpected shocks or disturbances in the system dynamics. Overall, the application of closed-loop control using the NeuralHJB approach suggests the potential for achieving real-time therapeutic outcomes through DBS with minimal energy consumption.

Chapter 4

Glucose-Insulin Control

4.1 Introduction

Glycemic control is a critical aspect of diabetes therapy. It involves regulating blood glucose (BG) levels to prevent undesirable complications associated with hypoglycemia (BG levels below 70 mg/dL) and hyperglycemia (BG levels above 180 mg/dL) [37, 78]. For individuals with Type 1 Diabetes (T1D), the inability to produce insulin necessitates exogenous administration to maintain normal BG concentration. Traditional approaches (open-loop control) rely on predetermined insulin dosing regimens based on factors like estimated meal carbohydrates (commonly known as ”carb" counting) and anticipated physical exercise. This static approach generally requires consistent meal intake, restricting dietary flexibility [1, 112]. Additionally, it cannot adjust insulin dosing in real-time based on fluctuations in BG levels [2]. This inflexibility can lead to suboptimal glycemic control, increasing the risk of hypoglycemia or hyperglycemia.

This chapter focuses on closed-loop control strategies within *artificial pancreas* (AP) systems, aiming to overcome the limitations of traditional open-loop control for T1D management. AP systems integrate three key components: 1) a continuous glucose monitor (CGM) sensor that measures BG levels every few minutes, 2) an

insulin pump that delivers insulin dose subcutaneously, and 3) a control algorithm that computes the appropriate insulin dose to be delivered by the pump based on BG data. We focus on the control algorithm, seeking strategies that achieve tighter glycemic control and reduce the risk of long-term complications associated with low BG (hypoglycemia) and high BG (hyperglycemia).

We explore two distinct approaches for closed-loop glycemic control within AP systems: 1) classical control methods, focusing on the widely used Proportional-Integral-Derivative (PID) controllers [5, 61, 87], and 2) neural network-based controllers, encompassing a model-based approach from Section 3.3 and a data-driven approach inspired by Reinforcement Learning (RL). To evaluate these algorithms, we simulate virtual patients whose glucose-insulin dynamics are modeled by established models like Bergman's Minimal Model [12].

4.2 Problem Setup

Similar to the neuromodulation control problem (Chapter 3.2), we seek to find an optimal control (insulin dosing) that minimizes a cost functional J in Equation (2.1). The control problem is constrained by some glucose-insulin dynamics, based on Bergman's Minimal Model [12], as

$$\begin{aligned} \frac{d\boldsymbol{z}}{dt}(t) &= f\big(t, \boldsymbol{z}(t); \boldsymbol{\psi}\big) + \boldsymbol{e}_1 p_5 \boldsymbol{u}(t), \\ \boldsymbol{z}(t) &= \boldsymbol{x}, \quad \boldsymbol{e}_1 = [1,0,0]^\top, \quad t \in [0,T], \end{aligned} \tag{4.1}$$

where $\boldsymbol{z}(t) = [I(t), X(t), BG(t)]^\top \in \mathbb{R}^3$ denotes the state variable, with $I(t)$, $X(t)$, and $BG(t)$ representing plasma insulin, remote insulin, and plasma glucose concentration, respectively. $\boldsymbol{x}$ denotes the initial state of the system. $\boldsymbol{u}(t) : [0,T] \to \mathbb{R}$ is the control variable (external insulin dose) provided as input by a controller at time t. p_5 is a parameter relating the rate of insulin delivery to the resulting increase in insulin

concentration. $\boldsymbol{\psi} \in \mathcal{E}$ represents the environment and includes extraneous information relevant to simulating the dynamics such as meal size and the amount appearing in the glucose compartment at the given time. The function $f : [0, T] \times \mathbb{R}^3 \to \mathbb{R}^3$ governs the evolution of the glucose-insulin dynamics.

4.2.1 Glucose-Insulin Dynamics

The glucoregulatory system is nonlinear. The complexity of the system is further exacerbated by varying physiological, pathological, or pharmacokinetic processes in diabetic individuals. Various mathematical models have been constructed to characterize glucose-insulin dynamics and aid in the understanding of BG regulation. These models range from structurally simplistic models like Bergman's Minimal Model [12] to complex semi-mechanistic Integrated Glucose Insulin models [53, 100, 104]. Additionally, physiologically-based models like the FDA-approved UVA/Padova T1D simulator [70] have also been developed.

To capture the essential dynamics of glucose-insulin interaction, we will leverage Bergman's Minimal Model as a foundation for the function f. This model is given by the following:

$$f\big(t, \boldsymbol{z}(t); \boldsymbol{\psi}\big) = \begin{bmatrix} -p_6 z_0(t) \\ -p_2 z_1(t) + p_3 \cdot [z_0(t) - I_b] \\ -p_1 z_2(t) - p_4 z_1(t) z_2(t) + p_1 G_b + R_a(t) \end{bmatrix}. \tag{4.2}$$

The physiological meaning and corresponding values of the parameters in the above equations is summarized in **Table** 4.1. $R_a(t)$ represents the rate of appearance of glucose and is computed as follows:

$$R_a(t) = \frac{C(t)}{V_G \tau_G^2} t \mathrm{e}^{-\frac{t}{\tau_G}},$$

where $C(t), V_G$ and τ represent the amount of carbohydrates consumed, distribution volume, and the peak time of meal absorption, respectively [113].

Parameter	Significance
p_1 (min^{-1})	rate of glucose clearance from plasma
p_2 (min^{-1})	rate constant for the disappearance of the insulin effect.
p_3 ($\text{min}^{-1}\text{mU}^{-1}$)	sensitivity of insulin effect on active insulin
p_4 ($\text{mL}^{-1}\text{min}^{-1}\text{mU}^{-1}$)	insulin-dependent glucose uptake
p_5 (mL^{-1})	inverse of insulin distribution volume
p_6 (min^{-1})	rate of plasma insulin clearance
C (g)	meal size/carbs consumed
I_b (mU)	basal plasma insulin concentration
G_b (mg/dL)	basal plasma glucose concentration
V_G (dL)	glucose distribution volume
τ_G (min)	peak time of meal absorption

Table 4.1: Physiological Significance of Parameters in the Bergman's Minimal Model.

4.2.2 Cost Function Formulation

Given a mathematical model for the interaction between BG and insulin, we seek to determine a control policy that achieves optimal glycemic control. To accomplish this, we define a cost function J which penalizes deviations from target BG levels. This will encourage the policy to maintain glucose levels within a safe and healthy range. J can consider other factors such as penalizing excessive insulin doses and risk for hypo- and hyperglycemia. Often the magnitude of the insulin dosage is of key interest. Too large a dose may not be desirable due to the increased risk of hypoglycemia, whereas inadequate doses may not be effective in curbing hyperglycemia.

To this end, for a given parameter vector $\boldsymbol{\psi} \in \eta$, the glycemic control cost function J comprises the running cost,

$$L\big(t, \boldsymbol{z}, \boldsymbol{u}\big) = \frac{1}{2}\|\boldsymbol{u}(t)\|^2 + \ell_{\text{risk}}\left(\boldsymbol{z}(t)\right). \tag{4.3}$$

Here, $\ell_{\text{risk}}(\boldsymbol{z})$ is a numerical measure of the overall quality of glycemia based on the

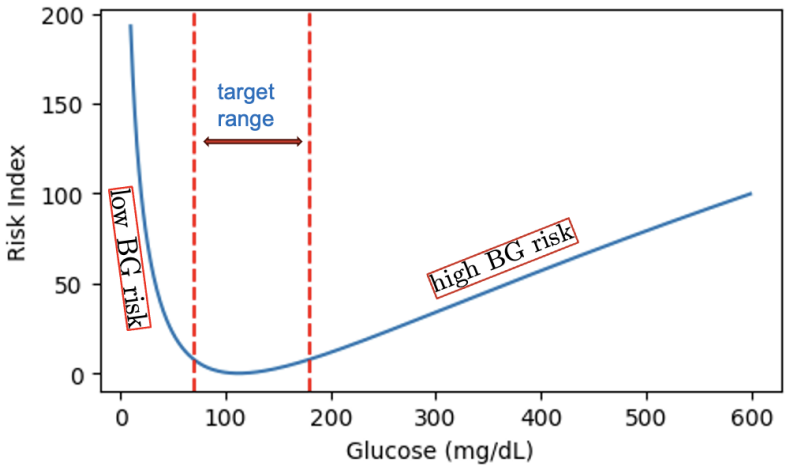


Figure 4.1: Blood Glucose Risk Index, adapted from [29]

Blood Glucose Risk Index (BGRI) metric provided by [29] as follows

$$\ell_{\text{risk}}\big(\boldsymbol{z}(t)\big) = \frac{1}{n}\sum_{i=1}^{n} rl(z_i) + \frac{1}{n}\sum_{i=1}^{n} rh(z_i), \tag{4.4}$$

where

$$rl(z) = 10Q(z) \text{ if } Q(z) < 0 \text{ and } 0 \text{ otherwise}, \tag{4.5}$$

$$rh(z) = 10Q(z) \text{ if } Q(z) > 0 \text{ and } 0 \text{ otherwise}, \tag{4.6}$$

with $Q(z) = 1.509 \times [(\ln(z))^{1.084} - 5.381]$. $\ell_{\text{risk}}(z)$ penalizes glucose levels that progressively deviate from the target range (associated with increased risk for hypo- and hyperglycemia), i.e., a BG reading of 40 mg/dL carries a higher risk compared to a BG reading of 65 mg/dL, despite both being below the lower bound of the normoglycemic target, 70 mg/dL. This categorization also applies to BG values above 180 mg/dL [29]. Figure 4.1 captures the risk index associated with BG values. We also define a terminal cost, $G : \mathbb{R}^3 \to \mathbb{R}$,

$$G\big(\boldsymbol{z}(T)\big) = \frac{1}{2}\|\boldsymbol{z}(T) - \boldsymbol{z}^*(T)\|^2, \tag{4.7}$$

Overall, we seek to minimize J over all admissible controls $\boldsymbol{u} \in U$ by finding a solution $u^*(t)$ that incurs the minimal cost, $0 \leq t \in \leq T$

$$J(t, \boldsymbol{z}, \boldsymbol{u}; \boldsymbol{\psi}) = \frac{1}{2}\|\boldsymbol{u}(t)\|^2 + \ell_{\text{risk}}\left(\boldsymbol{z}(t)\right) + \frac{1}{2}\|\boldsymbol{z}(T) - \boldsymbol{z}^*(T)\|^2, \tag{4.8}$$

as defined by the value function Φ in Equation (2.1).

4.3 Classical PID Control

Closed-loop glycemic control systems such as AP, often rely on Proportional-Integral-Derivative (PID) controllers to compute insulin dosing, given CGM measurements of BG. These controllers adjust insulin delivery based on deviations from a target BG level. They have widely been used to regulate BG since the early AP systems [2, 11, 30, 45, 52] and in various control systems including neuromodulation [38]. We briefly describe the idea behind PID controllers, including applications and limitations in glycemic control.

A PID controller comprises a weighted proportional, integral, and derivative term. The controller measures any deviations between the current state $\boldsymbol{z}$ and a target state $\boldsymbol{z}^*$. It then adjusts the control (insulin) based on this difference (proportional term), cumulative deviation of BG values from the target (integral term), and the rate of changes in BG levels over time (derivative term). Overall, the control input $\boldsymbol{u}(t)$ is computed as follows,

$$\boldsymbol{u}(t) = K_p e(t) + K_i \int_0^t e(s)ds + K_d \frac{de(t)}{dt}, \tag{4.9}$$

where $e(t) = \boldsymbol{z}(t) - \boldsymbol{z}^*(t)$ is the error term. K_p, K_i, and K_d are hyperparameters weighting the proportional, integral, and derivative terms, respectively. These parameters need to be tuned beforehand and optimizing them can be a complex and

time-consuming process [87].

While PID controllers are relatively straightforward and generalize easily to various control systems, they face limitations in managing BG [11]. The nonlinear and time-varying nature of glucose-insulin dynamics poses a great challenge. This is further complicated by variations in how individuals respond to insulin and external factors like meal intake and physical activity. As a result, PID controllers can struggle to adapt to rapid changes in BG levels, potentially leading to suboptimal control and an increased risk of hypoglycemia or hyperglycemia [87]. Nonetheless, some improvements to PID controllers, incorporating additional information such as meals and insulin-on-board, have been widely applied to commercial glucose-insulin control products such as the Medtronic 670G [108].

4.4 Neural Network-based Glycemic Control

As an alternative to traditional control schemes such as those based on PID control, we consider deep learning approaches leveraging trainable neural networks to approximately solve the glycemic control problem. We consider two approaches: 1) a model-based, semi-global solution approach, termed NeuralHJB, with details outlined in Section 3.3, and 2) a data-driven approach inspired by Reinforcement Learning (RL) and utilizing neural ODEs [28].

4.4.1 Model-based Control via NeuralHJB

As in the previous chapter 3, we approximate the value function Φ derived from the HJB equations using a neural network, extending the NeuralHJB approach to glycemic control. As before, we approximately solve the minimization problem detailed in Equation (2.9) to learn the weights $\boldsymbol{\theta}$ of the neural network. We also leverage the closed-form solution in Equation (2.7), which incorporates the system dynamics and

objective function of the glycemic control problem (Section 4.2.2).

4.4.2 Data-driven Control

Beyond model-based control, we explore a data-driven approach inspired by RL. In this framework, we utilize a neural network to directly approximate the value function Φ, incorporating dynamics based on Bergman's Minimal model (4.2). This approach does not rely explicitly on control theory. Instead, it adopts a data-driven scheme where the network learns through experience [33].

Vanilla NeuralODE Control Policy

We leverage neural Ordinary Differential Equations, NeuralODEs [28], trained on data containing current and historical states. Providing state history adds context and helps the controller learn the prevailing glycemic trends and the influence of controls on BG levels. The control(insulin dose) is computed by a network π as

$$\boldsymbol{u}_t = \pi_{\boldsymbol{\theta}}(t, \boldsymbol{z}(t); \boldsymbol{H}), \tag{4.10}$$

where $\boldsymbol{z}(t)$ is the state at time t, and $\boldsymbol{H} : [\boldsymbol{z}_i, \boldsymbol{m}_i, \boldsymbol{u}_i]\ \forall i \in [t-n-1, \ldots, t-1, 0]$ is the state and meal history of the individual n time steps before simulation at time $t = 0$. $\boldsymbol{H}$ can alternatively track cumulative experiences gained by the network during training. θ are the learnable parameters of the NeuralODE controller. NeuralODEs are ideal in this setting as they can provide continuous insulin delivery adjustments in real-time for more precise control, enabling the implementation of closed-loop control strategies.

The learning objective for the NeuralODE controller is to find the optimal policy π^* that minimizes the loss defined by Equation (2.10). We note that while the conventional approach in RL is to maximize rewards, these rewards can be typically

Algorithm 1 Glycemic Control using a NeuralODE policy

Initialize training dataset $\boldsymbol{H} : [\boldsymbol{z}, \boldsymbol{m}, \boldsymbol{u}]$
Randomly initialize NeuralODE controller $\pi_{\boldsymbol{\theta}}$
Sample random patient parameters $\boldsymbol{\psi}$
for $i \leftarrow 1$ **to** N **do**
    Sample random meal schedule M for simulation period T
    Initialize random patient state $\boldsymbol{z}_0 = \boldsymbol{x}$
    Initialize running cost $R_0 \leftarrow 0$
    for $t \leftarrow 1$ **to** T **do**
        Compute CGM measurement/observation $\boldsymbol{z}_t \leftarrow \boldsymbol{z}_t + \epsilon$
        Compute insulin dose $\boldsymbol{u}_t = \pi_{\boldsymbol{\theta}}(t, \boldsymbol{z}_t; \boldsymbol{H})$
        Update patient state: $\boldsymbol{z}_{t+1} = f(t, \boldsymbol{z}_t; \boldsymbol{\psi})$ using $\boldsymbol{u}_t$
        Compute reward: $R_t \leftarrow R_t + L(t, \boldsymbol{z}, \boldsymbol{u})$
        Append $(\boldsymbol{z}_t, \boldsymbol{M}_t, \boldsymbol{u}_t)$ to training dataset $\boldsymbol{H}$
    end for
    Train NeuralODE controller $\pi_{\boldsymbol{\theta}}$
    Compute loss $J = \text{mean}(R) + G(\boldsymbol{z}_T)$
    Update weights $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha J$
    if $i \bmod n_{val} \equiv 0$ **then**
        Perform validation
        Save parameters $\boldsymbol{\theta}$ for lowest J
    end if
end for
Return: best policy parameters $\pi_{\boldsymbol{\theta}}$

designed such that maximizing them leads to minimizing the loss (or negative reward) J in Equation (4.8). As discussed in Chapter 2.3.2, there are several options for minimizing J using RL methods (see Chapter 2.3.2). The NeuralODE controller closely resembles policy search methods that directly learn a policy mapping a current state $\boldsymbol{z}$ to an insulin dose $\boldsymbol{u}$ at time t. Algorithm 1 outlines the overall learning scheme for the NeuralODE controller.

4.5 Numerical Experiments

This section involves simulating glucose-insulin dynamics using the structurally simplistic Bergman model to capture the essential characteristics of glucose-insulin interaction. Our experimental setup includes comparing the performance of traditional PID

controllers against more advanced techniques, such as the model-based NeuralHJB controller, and the data-driven NeuralODE controller.

For each simulation, we generate meal schedules with random amounts of carbohydrates. Various scenarios can be considered, e.g., glycemic control in reduced meal schedules or sparse meal settings, or over longer simulation horizons ($6, 12, 72$ hours, or even days). For demonstrative purposes, we will conduct our experiments on a 24-hr, 3 meal setting. Another challenge lies in how long insulin stays active after injection. In the real-world, there are many types of insulin, including rapid-acting insulin with an onset of action of 15 minutes after injection and a duration of action of 5 hours. This can pose a challenge when devising control strategies. Myopic controllers would keep applying insulin not accounting for its long-term delayed effect. This solidifies the case for data-driven approaches that take treatment history into account when determining how much insulin to apply at any given time [33].

4.5.1 Pathological Glucose-Insulin Dynamics

Focusing on the Bergman model highlighted in Equation (4.2), several key parameters differentiate between diabetic and non-diabetic settings. These parameters reflect the physiological processes of glucose-insulin interaction, and their values are typically altered in diabetic individuals.

Overall, the following parameters are perturbed to influence BG levels and simulate pathological dynamics, inspired by similar approaches in [4, 92]:

- p_1 is the rate at which glucose is removed for energy production. We decrease this rate to simulate a scenario where the body struggles to utilize glucose effectively due to impaired metabolic processes. This means glucose stays in the body longer, leading to higher BG levels linked to hyperglycemia.

- p_2 indicates the rate constant for the disappearance of the insulin effect. Lowering

Parameter	Non-diabetic	Diabetic
p_1	**0.068**	**0.024**
p_2	**0.24**	**0.037**
p_3	$\mathbf{1.2 \times 10^{-5}}$	$\mathbf{1.2 \times 10^{-4}}$
p_4	1.0	1.0
p_5	0.098	0.098
p_6	0.142	0.142
I_b	**11.0**	**0**
G_b	82.0	82.0
V_G	117.0	117.0
τ_G	55.0	55.0

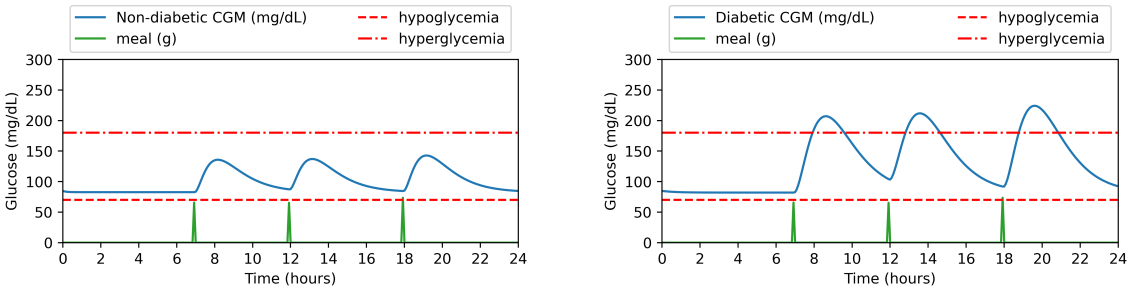
Table 4.2: Nominal values for simulating a non-diabetic and diabetic virtual patient using the Bergman's Minimal Model.

this rate signifies a faster decline in the effectiveness of insulin on glucose uptake over time. This indirectly leads to less efficient glucose removal and potentially higher BG levels (hyperglycemia).

- p_3 reflects the sensitivity of the effect of insulin on active insulin concentrations. We increase this to promote a stronger effect of insulin on glucose uptake, leading to a decrease in BG levels and potentially, hypoglycemia.

- I_b, the basal insulin level, indicating insulin secretion by the pancreas. We set this to 0 in the diabetic setting as the pancreas for individuals with T1D produces little or no insulin. This lack of basal insulin secretion contributes significantly to the hyperglycemia problem. Hence, individuals with T1D rely on external insulin administration to compensate for the missing basal insulin and manage BG levels.

With these modifications, we perturb normal dynamics (Figure 4.2a) and obtain the pathological activity we associated with diabetic conditions (Figure 4.2b). While the Bergman Minimal Model is an invaluable tool for studying glucose-insulin interactions, it has limitations. The model primarily focuses on the impact of insulin on glucose dynamics and assumes a steady equilibrium in glucose-insulin interactions in the



(a) non-diabetic BG dynamics

(b) diabetic BG dynamics

Figure 4.2: Simulations of normal and pathological BG dynamics in a non-diabetic and diabetic virtual patient during a 24-hr, 3-meal scenario.

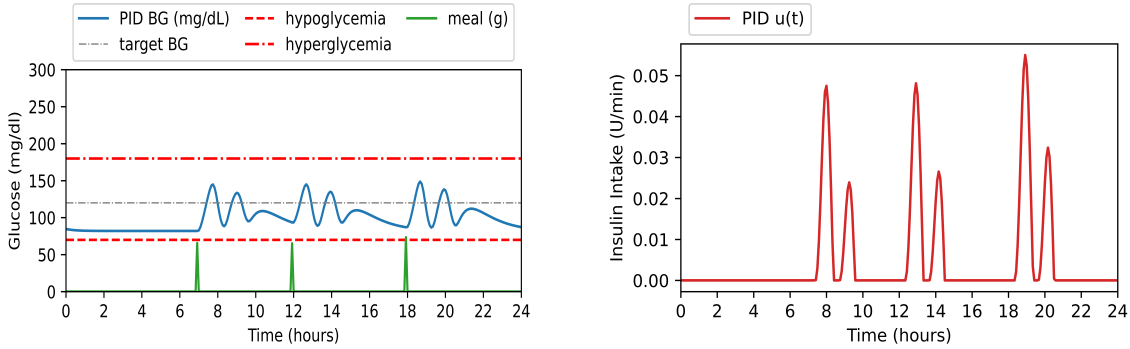
absence of external glucose., neglecting other aspects that might influence BG levels. However, the body's glucose demands are influenced by various factors such as physical activity and stress, which are not accounted for in this model. Consequently, the Bergman model's scope is restricted as it primarily considers only glucose and insulin interactions concerning meal intake. Despite these limitations, this simplified framework aligns well with our research objectives.

4.5.2 Glycemic Control

Each controller's performance is assessed using the cost function J in Equation (4.8), which comprises the control's energy expenditure (e.g., insulin amounts applied) and the risk of hypoglycemic and hyperglycemic events measured by the BGRI metric in Equation (4.4). We seek controllers that maintain BG within the normal glycemic range ($70 - 180$ mg/dL). To this end, we prioritize the time-in-range (TIR) metric, which measures the portion of time the controller keeps BG within the normal healthy range. TIR serves as a critical metric for evaluating the effectiveness of diabetes management strategies, with higher percentages indicating better glycemic control and reduced risk of hypo- and hyperglycemic events.

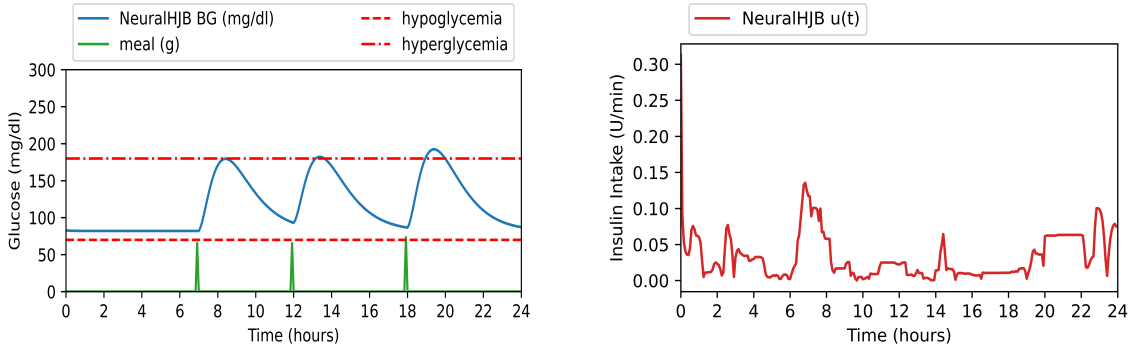
Each controller is evaluated on the pathological setting with parameters defined in Table 4.2, illustrated in Figure 4.2b. While the PID controller in Figure 4.3 effectively

regulates BG within the normal range and near the set target of 120 mg/dL, this approach may not perfectly translate to real-world scenarios. BG levels naturally fluctuate within a healthy range, and overly tight control might not be necessary or even desirable. For this reason, the NeuralHJB and NeuralODE controllers, with their more comprehensive cost functions (refer to Equation (4.8)), emerge as strong candidates for real-world applications. As shown in Figures 4.4 and 4.5 respectively, these controllers effectively manage the pathological BG events observed in Figure 4.2b.



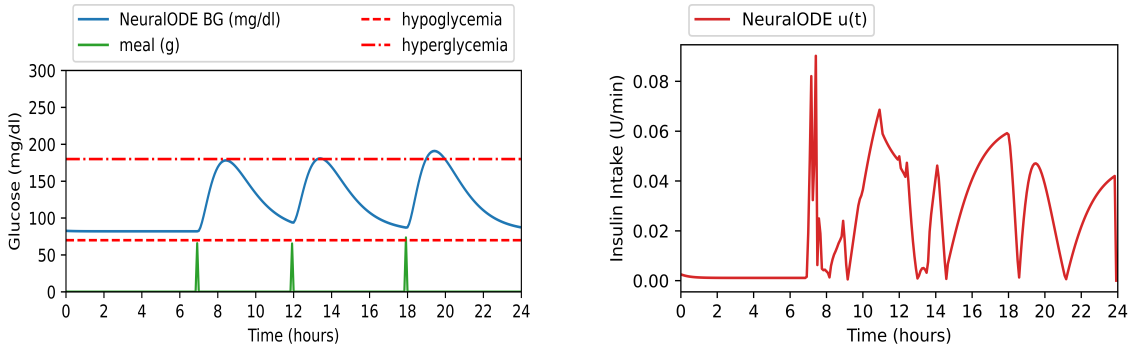
(a) Blood Glucose (b) Insulin

Figure 4.3: PID controls pathological BG dynamics within a healthy range close to a target BG.



(a) Blood Glucose (b) Insulin

Figure 4.4: NeuralHJB controller regulates pathological BG dynamics.



(a) Blood Glucose (b) Insulin

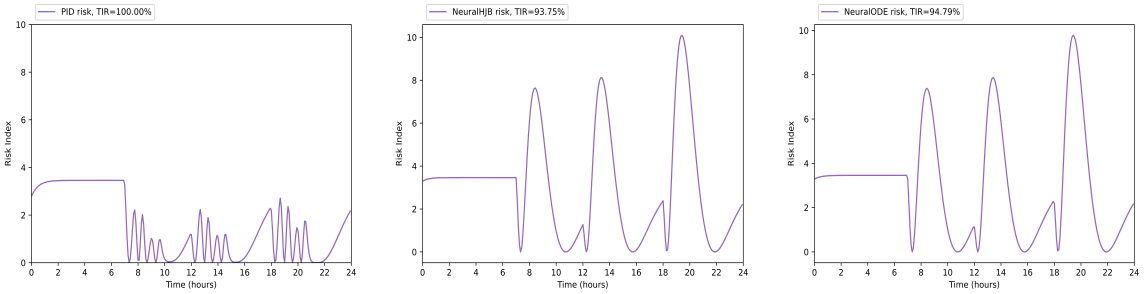
Figure 4.5: NeuralODE controls pathological BG dynamics, taking into account state history from the past hour.

Time-In-Range and Glycemic Risk Index

Our analysis of TIR revealed a clear advantage for the PID controller, which maintained BG within the target range 100% of the simulation period. While the NeuralODE controller (94.79% TIR) and the NeuralHJB controller (93.75% TIR) achieved slightly lower TIR, they offer a key benefit. These controllers utilize a more complex cost function that goes beyond simply keeping BG close to a target of 120 mg/dL (like the PID controller). This allows them to manage BG fluctuations within a healthy range more naturally, reducing the risk of hypo- and hyperglycemic events. Figure 4.6 illustrates the risk associated with each controller. The PID controller exhibits a lower BGRI compared to the neural network-based controllers, indicating tighter glycemic control (see Figure 4.3a).

Approach	TIR(%)
PID	100
NeuralHJB	93.75
NeuralODE	94.79

Table 4.3: TIR for all strategies.



(a) PID (b) NeuralHJB (c) NeuralODE

Figure 4.6: Blood Glucose Risk Index and Time-In-Range (TIR) for each control strategy.

4.6 Summary

Managing blood glucose (BG) is central in Type 1 diabetes (T1D) management. This chapter explored various closed-loop control methods to enhance glycemic control for individuals with T1D through artificial pancreas (AP) systems. These systems represent a substantial advancement over traditional open-loop BG management approaches, which rely on predetermined insulin dosing and lack adaptability to the dynamic fluctuations of BG levels.

Leveraging the Bergman's Minimal Model [12], we simulated the pathological BG activity associated with diabetic conditions. We considered various control strategies, including the widely used proportional-integral-derivative (PID) controller that adjusts the insulin applied based on deviations of BG measurements from a predetermined target [87]. Additionally, we implemented two controllers leveraging the function approximation capabilities of neural networks:

- a model-based NeuralHJB controller inspired by [69, 81, 115] pairing neural networks with control theory to optimize insulin delivery,
- a data-driven NeuralODE controller based on [28, 33] that aims to learn the underlying dynamics of the glucose-insulin system to generate optimal controls,

We evaluated the performance of each controller on a cost function incorporating such

metrics as energy utilized in applying insulin, the subsequent risk for hypoglycemia and hyperglycemia (computed using the BG Risk Index [29]), as well as deviations from a preset target BG value. We also computed the percentage of time spent by each method within a specified target range for BG levels (time-in-range). Our simulation results provide insights into the efficacy and limitations of each approach in optimizing BG levels and minimizing the risk of hypo- and hyperglycemic events.

Chapter 5

Conclusion and Research Outlook

This work focuses on applying Optimal Control(OC) theory and Machine Learning(ML) to control problems in two biomedical applications: neuromodulation in Deep Brain Stimulation (DBS) and glucose-insulin control in Type 1 Diabetes (T1D) management. In these settings, the integration of ML and OC facilitates the development of practical closed-loop systems that continuously analyze system dynamics, dynamically adjusting interventions–whether neurostimulation or insulin delivery–in real time. These closed-loop systems hold the potential to not only improve treatment outcomes for individuals living with neurological and metabolic conditions like T1D but also enhance the overall clinical experience in these applications.

In tackling the control problems in these applications, we leverage well-established models to simulate normal and pathological activity. We define cost functions that employ solution-specific knowledge such as penalties on the control signal or deviations from desired or target states. We then develop control strategies seeking to optimize these cost functions for each application. We summarize each application below and conclude with insights into potential research directions.

5.1 Dissertation Summary

Chapter 2 lays the groundwork for formulating deterministic finite-horizon optimal control problems. We frame two biomedical applications – neuromodulation in DBS and glycemic control in T1D management – as control problems. This chapter identifies key elements for control design, such as system dynamics and cost functions. Recognizing the limitations of predetermined or fixed interventions (stimulation or insulin dosing) in traditional open-loop control, the chapter motivates closed-loop control, which allows for dynamic and adaptive interventions. We then explore various control methods categorized as local, global, and semi-global approaches. Additionally, the chapter introduces established techniques like Pontryagin's Maximum Principle and Hamilton-Jacobi-Bellman (HJB) equations for finding optimal control solutions. Finally, the chapter introduces reinforcement learning (RL), highlighting its connection to OC concepts like value functions and the use of neural networks for function approximation. Overall, this chapter sets the stage for applying these control methods to the biomedical applications covered in subsequent chapters.

Chapter 3 explores a model-based approach for closed-loop control of DBS. This approach, called NeuralHJB, seeks to improve upon the limitations of predetermined open-loop DBS by dynamically adjusting stimulation based on real-time neuronal activity. We leverage a biophysically-accurate model of neuronal dynamics, the Hodgkin-Huxley (HH) model [49]. We specify a cost function for DBS including penalties on energy consumption by the control and deviations from target states. Building on established works [81], the NeuralHJB approach combines the function approximation capabilities of neural networks with HJB equations characterizing the value function. It involves pre-computing the control solution offline for a broad range of parameters characterizing diverse neuronal activity and online adaptation of the control policy to specific settings. Numerical experiments demonstrate NeuralHJB's effectiveness in restoring normal neuronal function in pathological conditions and

its ability to handle unexpected shocks or changes in neuronal dynamics. Although the NeuralHJB approach did not achieve optimal results in the amortized setting, its universal approximation properties suggest the potential for broader adaptability across a wide range of HH parameter configurations. Improving its performance would be interesting for further research.

Chapter 4 builds on the closed-loop control framework from Chapter 3 to address glycemic control problems in T1D management. In these settings, traditional open-loop control typically relies on pre-determined insulin dosing and lacks adaptability to fluctuating BG levels. We utilize Bergman's Minimal Model, a well-established mathematical model, to simulate the complex dynamics between glucose levels and insulin action [12]. We consider various control algorithms for closed-loop systems (often referred to as artificial pancreas systems in T1D management), ranging from the straightforward Proportional-Integral-Derivative controller to more advanced learning-based approaches. These approaches leverage neural networks to optimize insulin dosing and include the model-based NeuralHJB approach and a data-driven NeuralODE controller [28, 33]. Numerical experiments assess controller performance using clinically relevant Blood Glucose Risk Index and Time-in-Range metrics, evaluating accuracy in maintaining the desired BG range and percentage of time spent within it, respectively.

5.2 Research Outlook

Limitations of the research presented in this dissertation can be dealt with in many ways. We now highlight a few possible directions to expand upon our work.

5.2.1 Large-scale Dynamics

This research introduces learning-based control methods, such as NeuralHJB and NeuralODE controllers, for closed-loop neuromodulation and glucoregulatory systems, showing promising results in simulated environments to improve treatment efficacy. However, the simplified biophysical models used (Hodgkin-Huxley [49] for neuromodulation and Bergman's Minimal Model [12] for glycemic control) may not capture the full range of physiological responses, suggesting the need for more complex models [19, 70, 93, 100].

Moreover, controlling large-scale dynamics faces computational challenges due to the Curse of Dimensionality. Our semi-global NeuralHJB approach addresses this by combining HJB equations with neural networks, enabling efficient approximation of the value function in high-dimensional state spaces for effective closed-loop control. An extension involves using mean-field approximation techniques, like neural mass models [96, 106, 117], to reduce the dimensionality of the system, potentially enhancing the NeuralHJB approach's efficiency in handling high-dimensional systems.

5.2.2 Integrating External Factors

Future advancements in closed-loop neuromodulation and glycemic control may extend beyond the core physiological measures addressed in this study, such as membrane potential and blood glucose levels. For instance, in DBS, incorporating simulated local field potentials derived from multi-contact electrodes could offer a higher-resolution perspective on neural activity, facilitating the development of more precise control strategies compared to single-neuron membrane potentials [4, 65, 67, 88]. Similarly, in glycemic control, the integration of sensor data from activity trackers, dietary intake monitors, and advanced continuous glucose monitors may enable more comprehensive insulin dosing strategies [32, 50, 92]. By incorporating these external factors, future closed-loop systems have the potential to enhance adaptability and efficiency, thereby

improving treatment outcomes in both neuromodulation and glycemic control.

5.2.3 Other Biomedical Applications

In addition to neuromodulation and glycemic control, our research has the potential to benefit other applications, including mechanical ventilation in respiratory therapy and cardiac management in cardiovascular medicine. In mechanical ventilation, closed-loop control systems can optimize ventilator settings based on real-time physiological data, such as respiratory rate and oxygen saturation [22, 27]. This allows for dynamic adjustments to ventilator settings, ensuring adequate gas exchange while minimizing ventilator-induced lung injury [18]. Similarly, in cardiac management, closed-loop algorithms can be applied to implantable devices such as pacemakers and cardioverter-defibrillators, ensuring timely interventions based on real-time cardiac activity monitoring to regulate heart rhythm effectively [9, 95, 123].

Overall, this research applies closed-loop control to neuromodulation and glycemic control. We address the limitations of open-loop control (predetermined, non-responsive interventions) by integrating principles from OC theory and ML techniques. We frame these applications as control problems, simulating relevant dynamics in silico using established models. This allows us to define cost functions that optimize clinically relevant objectives. A pivotal aspect of this research is the incorporation of value function approximation with neural networks. This enables real-time, adaptive control strategies, overcoming the limitations of open-loop approaches and paving the way for improved therapeutic outcomes. These findings lay the groundwork for broader application of closed-loop control in biomedicine.

Bibliography

[1] Mary D Adu, Usman H Malabu, Aduli EO Malau-Aduli, and Bunmi S Malau-Aduli. Enablers and barriers to effective diabetes self-management: A multi-national investigation. *PloS one*, 14(6):e0217771, 2019.

[2] A Ml Albisser, BS Leibel, TG Ewart, Z Davidovac, CK Botz, W Zingg, H Schipper, and R Gander. Clinical control of diabetes by the artificial pancreas. *Diabetes*, 23(5):397–404, 1974. URL `https://doi.org/10.2337/diab.23.5.397`.

[3] Brandon Amos et al. Tutorial on amortized optimization. *Foundations and Trends® in Machine Learning*, 16(5):592–732, 2023.

[4] Daria Nesterovich Anderson, Braxton Osting, Johannes Vorwerk, Alan D Dorval, and Christopher R Butson. Optimized programming algorithm for cylindrical and directional deep brain stimulation electrodes. *Journal of neural engineering*, 15(2):026005, 2018.

[5] Kiam Heong Ang, Gregory Chong, and Yun Li. Pid control system analysis, design, and technology. *IEEE transactions on control systems technology*, 13(4): 559–576, 2005.

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

[7] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

[8] Vivek R Athalye, Jose M Carmena, and Rui M Costa. Neural reinforcement: re-entering and refining neural dynamics leading to desirable outcomes. *Current opinion in neurobiology*, 60:145–154, 2020.

[9] Temur Baykuziyev, Muhammad Jaffar Khan, Arunabha Karmakar, Muhammad Arif Baloch, and Jafar Khan. Closed-loop pharmacologic control of blood pressure: A review of existing systems. *Cureus*, 15(9), 2023.

[10] Richard Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[11] B Wayne Bequette. A critical assessment of algorithms and challenges in the development of a closed-loop artificial pancreas. *Diabetes technology & therapeutics*, 7(1):28–47, 2005.

[12] Richard N Bergman, Lawrence S Phillips, Claudio Cobelli, et al. Physiologic evaluation of factors controlling glucose tolerance in man: measurement of insulin sensitivity and beta-cell glucose sensitivity from the response to intravenous glucose. *The Journal of clinical investigation*, 68(6):1456–1467, 1981.

[13] Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*, volume 1. Athena scientific, 2012.

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

[15] M Beudel and P Brown. Adaptive deep brain stimulation in parkinson's disease. *Parkinsonism & related disorders*, 22:S123–S126, 2016. URL `https://pubmed.ncbi.nlm.nih.gov/26411502/`.

[16] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[17] 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. URL `https://www.nature.com/articles/s41467-021-23311-9`.

[18] Richard D Branson, Jay A Johannigman, Robert S Campbell, and Kenneth Davis Jr. Closed-loop mechanical ventilation. *Respiratory Care*, 47(4):427–51, 2002.

[19] Michael Breakspear. Dynamic models of large-scale brain activity. *Nature neuroscience*, 20(3):340–352, 2017. URL `https://www.nature.com/articles/nn.4497`.

[20] Romain Brette and Wulfram Gerstner. Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *Journal of neurophysiology*, 94(5):3637–3642, 2005.

[21] Jeff M Bronstein, Michele Tagliati, Ron L Alterman, Andres M Lozano, Jens Volkmann, Alessandro Stefani, Fay B Horak, Michael S Okun, Kelly D Foote, Paul Krack, et al. Deep brain stimulation for parkinson disease: an expert consensus and review of key issues. *Archives of neurology*, 68(2):165–165, 2011.

[22] JX Brunner. History and principles of closed-loop control applied to mechanical ventilation. *Nederlandse Vereniging voor Intensive Care*, 6(4):6–9, 2002.

[23] 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.

[24] William A Catterall. Structure and function of voltage-gated ion channels. *Annual review of biochemistry*, 64(1):493–531, 1995.

[25] 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.

[26] 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.

[27] Robert L Chatburn and Eduardo Mireles-Cabodevila. Closed-loop control of mechanical ventilation: description and classification of targeting schemes. *Respiratory care*, 56(1):85–102, 2011.

[28] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[29] William Clarke and Boris Kovatchev. Statistical tools to analyze continuous glucose monitor data. *Diabetes technology & therapeutics*, 11(S1):S–45, 2009.

[30] Claudio Cobelli, Eric Renard, and Boris Kovatchev. Artificial pancreas: past, present, future. *Diabetes*, 60(11):2672–2682, 2011.

[31] Isuru Dasanayake and Jr-Shin Li. Constrained minimum-power control of spiking neuron oscillators. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 3694–3699.

IEEE, 2011. URL https://folk.ntnu.no/skoge/prost/proceedings/cdc-ecc-2011/data/papers/1392.pdf.

[32] Eyal Dassau, Howard Zisser, Rebecca A Harvey, Matthew W Percival, Benyamin Grosman, Wendy Bevier, Eran Atlas, Shahar Miller, Revital Nimri, Lois Jovanovič, et al. Clinical evaluation of a personalized artificial pancreas. *Diabetes care*, 36(4):801–809, 2013.

[33] Souradeep Dutta, Taisa Kushner, and Sriram Sankaranarayanan. Robust data-driven control of artificial pancreas systems using neural networks. In *Computational Methods in Systems Biology: 16th International Conference, CMSB 2018, Brno, Czech Republic, September 12-14, 2018, Proceedings 16*, pages 183–202. Springer, 2018.

[34] M Ellinger, Melinda Evrithiki Koelling, Damon A Miller, Frank L Severance, and John Stahl. Exploring optimal current stimuli that provide membrane voltage tracking in a neuron model. *Biological cybernetics*, 104:185–195, 2011.

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

[36] 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. URL https://doi.org/10.1016/j.automatica.2017.09.004.

[37] International Diabetes Federation. Diabetes atlas. *IDF Diabetes Atlas, 7th edn. Brussels, Belgium: International Diabetes Federation*, 33(2), 2015.

[38] 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. URL `https://www.frontiersin.org/articles/10.3389/fnins.2020.00166/full`.

[39] 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.

[40] Eliana Della Flora, Caryn L Perera, Alun L Cameron, and Guy J Maddern. Deep brain stimulation for essential tremor: a systematic review. *Movement disorders*, 25(11):1550–1559, 2010.

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

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

[43] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[44] P Gorzelic, SJ Schiff, and Alok Sinha. Model-based rational feedback controller design for closed-loop deep brain stimulation of parkinson's disease. *Journal of neural engineering*, 10(2):026016, 2013. URL `https://iopscience.iop.org/article/10.1088/1741-2560/10/2/026016/meta`.

[45] Ahmad Haidar. The artificial pancreas: How closed-loop control is revolutionizing diabetes. *IEEE Control Systems Magazine*, 36(5):28–47, 2016.

[46] 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.

[47] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[48] Franz Hell, Carla Palleis, Jan H Mehrkens, Thomas Koeglsperger, and Kai Bötzel. Deep brain stimulation programming 2.0: future perspectives for target identification and adaptive closed loop stimulation. *Frontiers in neurology*, 10: 314, 2019.

[49] 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. URL `https://physoc.onlinelibrary.wiley.com/doi/pdfdirect/10.1113/jphysiol.1952.sp004764`.

[50] Roman Hovorka. Continuous glucose monitoring and closed-loop systems. *Diabetic medicine*, 23(1):1–12, 2006.

[51] Ronald A Howard. *Dynamic Programming and Markov Processes*. John Wiley, 1960.

[52] Mingzhan Huang, Jiaxu Li, Xinyu Song, and Hongjian Guo. Modeling impulsive injections of insulin: towards artificial pancreas. *SIAM Journal on Applied Mathematics*, 72(5):1524–1548, 2012.

[53] Petra M Jauslin, Hanna E Silber, Nicolas Frey, Ronald Gieschke, Ulrika SH Simonsson, Karin Jorga, and Mats O Karlsson. An integrated glucose-insulin model to describe oral glucose tolerance test data in type 2 diabetics. *The Journal of Clinical Pharmacology*, 47(10):1244–1255, 2007.

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

[55] Werner M Kistler, Wulfram Gerstner, and J Leo van Hemmen. Reduction of the hodgkin-huxley equations to a single-variable threshold model. *Neural computation*, 9(5):1015–1045, 1997. URL `https://infoscience.epfl.ch/record/97776/files/Gerstner97.pdf`.

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

[57] 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.

[58] Joachim K Krauss, Nir Lipsman, Tipu Aziz, Alexandre Boutet, Peter Brown, Jin Woo Chang, Benjamin Davidson, Warren M Grill, Marwan I Hariz, Andreas Horn, et al. Technology of deep brain stimulation: current status and future directions. *Nature Reviews Neurology*, 17(2):75–87, 2021.

[59] Alexis M Kuncel and Warren M Grill. Selection of stimulus parameters for deep brain stimulation. *Clinical neurophysiology*, 115(11):2431–2441, 2004.

[60] 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.

[61] Yongho Lee, Sunwon Park, Moonyong Lee, and Coleman Brosilow. Pid controller tuning for desired closed-loop responses for si/so systems. *Aiche journal*, 44(1): 106–115, 1998.

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

[63] Yuxi Li. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*, 2017.

[64] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[65] Simon Little and Peter Brown. What brain signals are suitable for feedback control of deep brain stimulation in parkinson's disease? *Annals of the New York Academy of Sciences*, 1265:9–24, 2012. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3495297/`.

[66] 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.

[67] Andres M Lozano, Nir Lipsman, Hagai Bergman, Peter Brown, Stephan Chabardes, Jin Woo Chang, Keith Matthews, Cameron C McIntyre, Thomas E Schlaepfer, Michael Schulder, et al. Deep brain stimulation: current challenges and future directions. *Nature Reviews Neurology*, 15(3):148–160, 2019.

[68] 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. URL `http://www.ncbi.nlm.nih.gov/pubmed/31715567`.

[69] Malvern Madondo, Deepanshu Verma, Lars Ruthotto, and Nicholas Au Yong. Learning control policies of hodgkin-huxley neuronal dynamics. *Machine Learning for Health (ML4H) Findings Collection*, 2023.

[70] Chiara Dalla Man, Francesco Micheletto, Dayu Lv, Marc Breton, Boris Kovatchev, and Claudio Cobelli. The uva/padova type 1 diabetes simulator: new features. *Journal of diabetes science and technology*, 8(1):26–34, 2014. URL `https://doi.org/10.1177/1932296813514502`.

[71] Michael Mascagni. The backward euler method for numerical solution of the hodgkin–huxley equations of nerve conduction. *SIAM journal on numerical analysis*, 27(4):941–962, 1990.

[72] 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.

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

[74] Khalid B Mirza, Caroline T Golden, Konstantin Nikolic, and Christofer Toumazou. Closed-loop implantable therapeutic neuromodulation systems based on neurochemical monitoring. *Frontiers in neuroscience*, 13:808, 2019.

[75] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[76] Ali Nabi and Jeff Moehlis. Single input optimal control for globally coupled neuron networks. *Journal of neural engineering*, 8(6):065008, 2011.

[77] 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.

[78] David M Nathan. Long-term complications of diabetes mellitus. *New England journal of medicine*, 328(23):1676–1685, 1993.

[79] T Nguyen-Thien and T Tran-Cong. Approximation of functions and their derivatives: A neural network implementation with applications. *Applied Mathematical Modelling*, 23(9):687–704, 1999.

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

[81] 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, 2021.

[82] Mahboubeh Parastarfeizabadi and Abbas Z Kouzani. Advances in closed-loop deep brain stimulation devices. *Journal of neuroengineering and rehabilitation*, 14(1):1–20, 2017. URL `https://jneuroengrehab.biomedcentral.com/articles/10.1186/s12984-017-0295-1`.

[83] 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 *Learning for Dynamics and Control*, pages 728–738. PMLR, 2020. URL `https://proceedings.mlr.press/v120/pereira20a.html`.

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

[85] J Blair Price, Aaron E Rusheen, Abhijeet S Barath, Juan M Rojas Cabrera, Hojin Shin, Su-Youne Chang, Christopher J Kimble, Kevin E Bennet, Charles D Blaha, Kendall H Lee, et al. Clinical applications of neurochemical and electrophysiological measurements for closed-loop neurostimulation. *Neurosurgical focus*, 49(1):E6, 2020.

[86] Rajamannar Ramasubbu, Stefan Lang, and Zelma HT Kiss. Dosing of electrical parameters in deep brain stimulation (dbs) for intractable depression: a review of clinical studies. *Frontiers in Psychiatry*, 9:302, 2018.

[87] Y Ramprasad, Gade Pandu Rangaiah, and Samavedham Lakshminarayanan. Robust pid controller for blood glucose regulation in type i diabetics. *Industrial & engineering chemistry research*, 43(26):8257–8268, 2004. URL `https://pubs.acs.org/doi/pdf/10.1021/ie049546a`.

[88] NJ Ray, N Jenkinson, S Wang, P Holland, JS Brittain, C Joint, JF Stein, and T Aziz. Local field potential beta activity in the subthalamic nucleus of patients with parkinson's disease is associated with improvements in bradykinesia after dopamine and deep brain stimulation. *Experimental neurology*, 213(1):108–113, 2008. URL `https://pubmed.ncbi.nlm.nih.gov/18619592/`.

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

[90] Vincent Renault, Michèle Thieullen, and Emmanuel Trélat. Optimal control of infinite-dimensional piecewise deterministic markov processes and application to the control of neuronal dynamics via optogenetics. *arXiv preprint arXiv:1607.05574*, 2016.

[91] David Rodbard. Continuous glucose monitoring: a review of successes, challenges, and opportunities. *Diabetes technology & therapeutics*, 18(S2):S2–3, 2016.

[92] Anirban Roy. *Dynamic modeling of free fatty acid, glucose, and insulin during rest and exercise in insulin dependent diabetes mellitus patients*. PhD thesis, University of Pittsburgh, 2008.

[93] Jonathan E Rubin and David Terman. High frequency stimulation of the subthalamic nucleus eliminates pathological thalamic rhythmicity in a computational model. *Journal of computational neuroscience*, 16(3):211–235, 2004. URL `https://link.springer.com/article/10.1023/B:JCNS.0000025686.47117.67`.

[94] Lars Ruthotto, Stanley J Osher, Wuchen Li, Levon Nurbekyan, and Samy Wu Fung. A machine learning framework for solving high-dimensional mean field game and mean field control problems. *Proceedings of the National Academy of Sciences*, 117(17):9183–9193, 2020.

[95] Kiichi Sagawa. Closed-loop physiological control of the heart. *Annals of Biomedical Engineering*, 8:415–429, 1980.

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

[97] Sabato Santaniello, Giovanni Fiengo, Luigi Glielmo, and Warren M Grill. Closed-loop control of deep brain stimulation: a simulation study. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 19(1):15–24, 2010.

[98] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[99] Karen Schneck, Lai San Tham, Ali Ertekin, and Jesus Reviriego. Toward better understanding of insulin therapy by translation of a pk-pd model to visualize insulin and glucose action profiles. *The Journal of Clinical Pharmacology*, 59(2): 258–270, 2019.

[100] Karen B Schneck, Xin Zhang, Robert Bauer, Mats O Karlsson, and Vikram P Sinha. Assessment of glycemic response to an oral glucokinase activator in a proof of concept study: application of a semi-mechanistic, integrated glucose-insulin-glucagon model. *Journal of pharmacokinetics and pharmacodynamics*, 40:67–80, 2013. URL `https://doi.org/10.1007/s10928-012-9287-8`.

[101] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[102] 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.

[103] Rui Shu. Amortized optimization, 2017. URL `https://ruishu.io/2017/11/07/amortized-optimization/`.

[104] Hanna E Silber, Petra M Jauslin, Nicolas Frey, and Mats O Karlsson. An integrated model for the glucose-insulin system. *Basic & clinical pharmacology & toxicology*, 106(3):189–194, 2010.

[105] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. Pmlr, 2014.

[106] 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. URL `https://link.springer.com/article/10.1007/s10827-011-0366-4`.

[107] Scott Stanslaski, Jeffrey Herron, Tom Chouinard, Duane Bourget, Ben Isaacson, Vaclav Kremen, Enrico Opri, William Drew, Benjamin H Brinkmann, Aysegul

Gunduz, et al. A chronically implantable neural coprocessor for investigating the treatment of neurological disorders. *IEEE transactions on biomedical circuits and systems*, 12(6):1230–1245, 2018.

[108] Michael P Stone, Pratik Agrawal, Xiaoxiao Chen, Margaret Liu, John Shin, Toni L Cordero, and Francine R Kaufman. Retrospective analysis of 3-month real-world glucose data after the minimed 670g system commercial launch. *Diabetes technology & therapeutics*, 20(10):689–692, 2018.

[109] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[110] 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.

[111] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

[112] Paula M Trief, Donald Cibula, Elaine Rodriguez, Bridget Akel, and Ruth S Weinstock. Incorrect insulin administration: a problem that warrants attention. *Clinical Diabetes*, 34(1):25–33, 2016.

[113] Kamuran Turksoy, Sediqeh Samadi, Jianyuan Feng, Elizabeth Littlejohn, Laurie Quinn, and Ali Cinar. Meal detection in patients with type 1 diabetes: a new module for the multivariable adaptive artificial pancreas control system. *IEEE journal of biomedical and health informatics*, 20(1):47–54, 2015.

[114] Deepanshu Verma, Xingjian Li, Nickolas Winovich, Lars Ruthotto, and Bart Van Bloemen Waanders. Advances and challenges in solving high-dimensional

hjb equations arising in optimal control. In *2023 Joint Mathematics Meetings (JMM 2023)*. AMS, 2023.

[115] Deepanshu Verma, Nick Winovich, Lars Ruthotto, and Bart van Bloemen Waanders. Neural network approaches for parameterized optimal control. *arXiv preprint arXiv:2402.10033*, 2024.

[116] Jens Volkmann, Jan Herzog, Florian Kopper, and Güntner Deuschl. Introduction to the programming of deep brain stimulators. *Movement disorders: official journal of the Movement Disorder Society*, 17(S3):S181–S187, 2002.

[117] Minh Vu, Bharat Singhal, Shen Zeng, and Jr-Shin Li. Data-driven control of neuronal networks with population-level measurement. *Research Square (Preprint)*, 2023.

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

[119] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8: 279–292, 1992.

[120] Gihan Weerasinghe, Benoit Duchet, Christian Bick, and Rafal Bogacz. Optimal closed-loop deep brain stimulation using multiple independently controlled contacts. *PLoS Computational Biology*, 17(8):e1009281, 2021.

[121] 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.

[122] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992.

[123] Eileen A Woodruff, James F Martin, and Madonna Omens. A model for the design and evaluation of algorithms for closed-loop cardiovascular therapy. *IEEE transactions on biomedical engineering*, 44(8):694–705, 1997.

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

[125] 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. URL `https://doi.org/10.1007/s10483-020-2689-9`.

[126] Aston Zhang, Zachary C Lipton, Mu Li, and Alexander J Smola. Dive into deep learning. *arXiv preprint arXiv:2106.11342*, 2021.

Index