

University of California  
Santa Barbara

# **Model-Based and Machine Learning-Based Control of Biological Oscillators**

A dissertation submitted in partial satisfaction  
of the requirements for the degree

Doctor of Philosophy  
in  
Mechanical Engineering

by

Bharat Monga

Committee in charge:

Professor Jeff Moehlis, Chair  
Professor Igor Mezić  
Professor Frederic Gibou  
Professor João Hespanha

March 2020

The Dissertation of Bharat Monga is approved.

---

Professor Igor Mezić

---

Professor Frederic Gibou

---

Professor João Hespanha

---

Professor Jeff Moehlis, Committee Chair

February 2020

Model-Based and Machine Learning-Based Control of Biological Oscillators

Copyright © 2020

by

Bharat Monga

Dedicated to my Mom and Dad.

## Acknowledgements

First and foremost, I would like to thank my PhD advisor Professor Jeff Moehlis. His mentorship and support shaped every aspect of my PhD including research, teaching, and job search. His cool vibe helped take pressure off when things were not working, or when we had tight deadlines, and made my PhD experience enjoyable. I am grateful for him providing financial support during my PhD.

I am also thankful to my PhD committee members Professor Igor Mezić, Professor Frederic Gibou, and Professor João Hespanha for insightful discussions during my candidacy. I acknowledge my lab mate Tim Matchen for providing insight to my research during group meetings which led to new ideas. I thank Tim, Allan, Karthik, Poorva, and Gowtham for providing feedback on my defense presentation.

I am grateful to our collaborator Professor Tay Netoff for welcoming me in his lab at University of Minnesota where I spent a summer doing brain slice experiments. The experiments gave me perspective on experimental neuroscience and shaped the direction of my later research.

I am extremely grateful to my Mom and Dad for providing love and support during my PhD. This work would not have been possible without their support. I am thankful for my friends Angela, Evan, Olga, Anant, SeongHyeon, Hector, and others. Last but not the least, I express gratitude towards the Pacific Ocean and the mountains around Santa Barbara.

# Curriculum Vitæ

## Bharat Monga

### Education

- 2020 Ph.D. in Mechanical Engineering (Expected), University of California, Santa Barbara.
- 2015 Master and Bachelor of Technology in Mechanical Engineering, Indian Institute of Technology Bombay.

### Publications

- B. Monga and J. Moehlis, Supervised Learning Algorithms for Controlling Under-actuated Dynamical Systems, arXiv preprint arXiv:1909.11119.
- B. Monga, D. Wilson, T. Matchen and J. Moehlis, Phase Reduction and Phase-Based Optimal Control for Biological Systems: A Tutorial, *Biological Cybernetics*, 113 (2019), pp. 11 - 46.
- B. Monga, G. Froyland, and J. Moehlis, Synchronizing and Desynchronizing Neural Populations through Phase Distribution Control, in 2018 American Control Conference (ACC), June 2018, pp. 2808 - 2813.
- B. Monga and J. Moehlis, Optimal Phase Control of Biological Oscillators using Augmented Phase Reduction, *Biological Cybernetics*, 113 (2019), pp. 161 - 178.
- B. Monga and J. Moehlis, Phase Distribution Control of a Population of Oscillators, *Physica D: Nonlinear Phenomena*, 398 (2019), pp. 115 - 129.
- B. Monga and J. Moehlis, Augmented Phase Reduction for Homoclinic Bifurcation and Relaxation Oscillators, (In preparation).
- D. Kuelbs, J. Dunefsky, J. Moehlis and B. Monga, Analysis of Deep Brain Stimulation for Neural Population Dynamics, (In preparation).

### Awards and Honors

- 2019 Awarded Dissertation Fellowship by University of California Santa Barbara.
- 2019 Invited to Dean's cabinet meeting to showcase the impact of my PhD research to university donors.
- 2019 Won ME Grad Slam, a department-wide graduate student public speaking research competition.
- 2019 Awarded Excellence in Teaching Award by University of California Santa Barbara.
- 2019 Semifinalist in Grad Slam, a university-wide graduate student public speaking research competition.

- 2019      Awarded SIAM Travel Award to present at the SIAM Conference on Applications of Dynamical Systems 2019.
- 2018      Awarded Doctoral Travel Grant to present research at the 2018 American Control Conference.
- 2015      Accorded Institute Silver Medal for graduating top of my IIT Bombay class.
- 2013      Awarded DAAD-WISE research grant by German Academic Exchange Service to pursue research in Germany.
- 2014      Awarded Institute Academic Prize at IIT Bombay for highest GPA in academic year 2013-14.

### **Conference Presentations**

- *Changing Collective Behavior of Oscillators by Controlling the Destiny of Their Phase Density*, SIAM Conference on Applications of Dynamical Systems, Snowbird, UT, May 2019.
- *Synchronizing and Desynchronizing Neural Populations*, American Control Conference, Milwaukee, WI, June 2018.
- *Optimal Timing Control of Biological Oscillators using Augmented Phase Reduction*, SIAM Conference on Applications of Dynamical Systems, Snowbird, UT, May 2017.

### **Peer Reviews**

- Journal of Mathematical Biology.
- Chaos: An Interdisciplinary Journal of Nonlinear Science.

### **Teaching and Mentorship**

- Lead Teaching Assistant, UCSB, September 2018 – August 2019.
- Mentor: Research Mentorship Program, UCSB, Summer 2018.
- Instructor, Nonlinear Dynamics and Chaos, UCSB, Spring 2018.
- Teaching Assistant: Introduction to Programming, UCSB, Fall 2018 & 2019.
- Teaching Assistant: Online Course on Introduction to Programming, UCSB, Summer 2016, 2018 & 2019.
- Teaching Assistant, Mechatronics, UCSB, Fall 2015.
- Teaching Assistant: Engineering Mechanics: Vibrations, UCSB, Winter 2016.
- Teaching Assistant: Solid Mechanics Lab, IIT Bombay, Spring 2015.
- Teaching Assistant: Microprocessors and Automatic Control, IIT Bombay, Autumn 2014.
- Teaching Assistant: Electricity and Magnetism, IIT Bombay, Summer 2012.
- Teaching Assistant: Differential Equations, IIT Bombay, Spring 2012.

## Abstract

Model-Based and Machine Learning-Based Control of Biological Oscillators

by

Bharat Monga

Nonlinear oscillators - dynamical systems with stable periodic orbits - arise in many systems of physical, technological, and biological interest. This dissertation investigates the dynamics of such oscillators arising in biology, and develops several control algorithms to modify their collective behavior. We demonstrate that these control algorithms have potential in devising treatments for Parkinson's disease, cardiac alternans, and jet lag.

Phase reduction, a classical reduction technique, has been instrumental in understanding such biological oscillators. In this dissertation, we investigate a new reduction technique called augmented phase reduction, and calculate its associated analytical expressions for six dynamically different planar systems: This helps us to understand the dynamical regimes for which the use of augmented phase reduction is advantageous over the standard phase reduction.

We further this study by developing a novel optimal control algorithm based on the augmented phase reduction to change the phase of a single oscillator using a minimum energy input. We show that our control algorithm is effective even when a large phase change is required or when the nontrivial Floquet multiplier of the oscillator is close to unity; in such cases, the previously proposed control algorithm based on the standard phase reduction fails.

We then devise a novel framework to control a population of biological oscillators as a whole, and change their collective behavior. Our first two control algorithms are Lyapunov-based, and our third is an optimal control algorithm which minimizes the con-

trol energy consumption while achieving the desired collective behavior of an oscillator population. We show that the developed control algorithms can synchronize, desynchronize, cluster, and phase shift the population.

We continue this investigation by developing two novel machine learning control algorithms, which have a simple and intelligent structure that makes them effective even with a sparse data set. We show that these algorithms are powerful enough to control a wide variety of dynamical systems and not just biological oscillators. We conclude this study by understanding how the developed machine learning algorithms work in terms of phase reduction.

In this dissertation, we have developed all these algorithms with the goal of ease of experimental implementation, for which the model parameters/training data can be measured experimentally. We close the loop on this dissertation by carrying out robustness analysis for the developed algorithms; demonstrating their resilience to noise, and thus their suitability for controlling living biological tissue. They truly hold great potential in devising treatments for Parkinson's disease, cardiac alternans, and jet lag.

# Contents

<b>Curriculum Vitae</b>	<b>vi</b>
<b>Abstract</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Background</b>	<b>8</b>
2.1 Introduction . . . . .	8
2.2 Standard Phase Reduction and Control . . . . .	8
2.3 Augmented Phase Reduction . . . . .	12
<b>3 On Augmented Phase Reduction: Analytical and Numerical Results</b>	<b>19</b>
3.1 Introduction . . . . .	19
3.2 Analytical and Numerical Computation of the Augmented Phase Reduction	22
3.3 Discussion and Conclusions . . . . .	49
<b>4 Optimal Phase Control using Augmented Phase Reduction</b>	<b>52</b>
4.1 Introduction . . . . .	52
4.2 Optimal Phase Control . . . . .	55
4.3 Applications . . . . .	59
4.4 Effect of Noise . . . . .	75
4.5 Discussion and Conclusion . . . . .	78
<b>5 Phase Distribution Control</b>	<b>81</b>
5.1 Introduction . . . . .	81
5.2 Control Algorithm . . . . .	83
5.3 Degenerate Set . . . . .	87
5.4 Numerical Methods . . . . .	90
5.5 Applications . . . . .	93
5.6 Addition of White Noise . . . . .	106
5.7 Optimal Control of Phase Distributions . . . . .	109
5.8 Conclusion . . . . .	113

<b>6</b>	<b>Supervised Learning Based Control</b>	<b>115</b>
6.1	Introduction . . . . .	115
6.2	Supervised Learning Algorithm . . . . .	118
6.3	Bistable Dynamical Systems . . . . .	122
6.4	Phase Control of an Oscillator . . . . .	129
6.5	Desynchronization of a Population of Coupled Oscillators . . . . .	132
6.6	Stabilizing an Unstable Fixed Point . . . . .	137
6.7	Robustness to Noise . . . . .	140
6.8	Conclusion . . . . .	143
<b>7</b>	<b>Conclusions and Experimental Implementation</b>	<b>144</b>
<b>A</b>	<b>Models</b>	<b>148</b>
A.1	Thalamic neuron model . . . . .	148
A.2	YNI model . . . . .	149
A.3	Clock gene regulation model . . . . .	150
A.4	Reduced Hodgkin-Huxley model . . . . .	150
<b>B</b>	<b>Numerical Methods</b>	<b>151</b>
B.1	Computation of PRC . . . . .	151
B.2	Computation of Floquet multipliers . . . . .	151
B.3	Two point BVP with Newton Iteration . . . . .	152
B.4	Two point BVP with modified Newton Iteration . . . . .	154
	<b>Bibliography</b>	<b>157</b>

# Chapter 1

## Introduction

Nonlinear oscillators - dynamical systems with stable periodic orbits - arise in many systems of physical, technological, and biological interest [1, 2, 3, 4]. This dissertation investigates the dynamics of such oscillators arising in biology, and develops several control algorithms to modify their behavior.

Examples of biological oscillators include the beating of pacemaker cells in the heart, the firing of action potentials in neurons, and circadian rhythms, among many others. The collective behavior of such oscillators varies, and includes synchronization, desynchronization, and clustering. For example, the beating of the heart is regulated by constant pacing of synchronized cardiac pacemaker cells [5, 6], and neural synchrony is essential in visual and odor processing [7, 8], and also in learning and memory recall [9, 10]. However, synchronization can be detrimental as well. For example, pathological neural synchronization in the thalamus and the subthalamic nucleus (STN) brain region is hypothesized to be one of the causes of motor symptoms for essential and parkinsonian tremor, respectively [11, 12]; this motivates the goal of designing a control stimulus to desynchronize a neural oscillator population. Recently there has also been focus on achieving partial synchrony through clustering instead of complete neural desynchro-

nization [13, 14, 15]. One motivation behind such clustering is that neural plasticity rewires synaptic connections between neurons and thus stabilizes clusters in the long term. Besides these control objectives, it is also advantageous to change the phase of a synchronized oscillator population to potentially help in treatment of cardiac alternans, and to treat jet lag [16].

This motivates devising algorithms to control the behavior of these biological oscillators. However, the dynamical models describing such biological oscillations tend to be highly nonlinear and high-dimensional. This hinders analysis of their dynamics and the formulation of control algorithms. “Standard” phase reduction [3, 17, 2, 18], a classical reduction technique based on isochrons [3, 17, 19], has been instrumental in understanding such biological oscillators. It works by reducing the dimensionality of a dynamical system with a periodic orbit to a single phase variable. This reduction captures the oscillator’s dynamics near the periodic orbit and the change in its phase due to an external control stimulus through the phase response curve (PRC). Not only does it make the analysis of the high dimensional biological systems more tractable, but it also has the potential to make their control [20, 21, 22, 23, 24] experimentally implementable; see e.g., [25, 26, 27, 22]. This is because although the whole state space dynamics of the biological system may not be known, PRCs can often be measured experimentally; see e.g., [28, 24]. We heavily employ standard phase reduction for analysis and control of biological oscillators in this dissertation.

Standard phase reduction is valid only in a small neighborhood of the periodic orbit. Consequently, the magnitude of the allowable perturbation is limited by the nontrivial Floquet multiplier [29] of the periodic orbit: in systems with a Floquet multiplier close to one, even a relatively small perturbation can lead to a trajectory which stays away from the periodic orbit, rendering the phase reduction inaccurate and phase reduction based control ineffective. This necessitates the use of augmented phase reduction [30],

an  $n$ -dimensional reduction based on both isochrons and isostables [31]. While the first dimension captures the phase of the oscillator along the periodic orbit, like the standard phase reduction, the other dimensions capture the oscillator's transversal approach to the periodic orbit. This reduction ascertains the effect of an external stimulus on the oscillator's phase change through the PRC, and the change in its transversal distance to the periodic orbit through the isostable response curve (IRC). We envision that IRCs can be measured experimentally just like PRCs, making the control based on the augmented phase reduction experimentally amenable as well. In this dissertation, we show that control algorithms based on the augmented phase reduction are expected to be more effective [16], as they can be designed to allow a larger stimulus without the risk of driving the oscillator too far away from the periodic orbit.

We begin this dissertation by giving background on standard and augmented phase reduction in Chapter 2. In the same chapter, we also detail how the control stimulus comes into the picture in these model reduction techniques. Finally, we describe various methods to calculate the response functions of these model reductions - PRCs and IRCs, as they are instrumental in development of the control algorithms.

In Chapter 3, we calculate analytical expressions for the augmented phase reduction for six dynamically different planar systems:  $\lambda - \omega$  systems, periodic orbits born out of four codimension-one bifurcations, and relaxation oscillators. Our contribution is the analytical calculation of IRCs and the nontrivial Floquet exponent for each of these six systems, and the PRC for a simple model undergoing SNIPER bifurcation. To validate our calculations, we simulate several models in these dynamical regimes, and compare their numerically computed augmented phase reduction with the derived analytical expressions. These analytical and numerical calculations help us understand the dynamical regimes for which the use of augmented phase reduction is advantageous over the standard phase reduction.

We continue this investigation in Chapter 4, where we control the phase of a single oscillator. We develop a novel optimal control algorithm based on augmented phase reduction to change the phase of an oscillator using a minimum energy input, which also minimizes the oscillator's transversal distance to the uncontrolled periodic orbit. In the same chapter, we develop a novel method to eliminate cardiac alternans by connecting our control algorithm with the underlying physiological problem. We also describe how the devised algorithm can be used for spike timing control, which can potentially help with motor symptoms of essential and parkinsonian tremor, and aid in treating jet lag. To demonstrate the advantages of this algorithm, we compare it with a previously proposed optimal control algorithm based on standard phase reduction for the Hopf bifurcation normal form, and models for cardiac pacemaker cells, thalamic neurons, and the circadian gene regulation cycle in the suprachiasmatic nucleus. We show that our control algorithm is effective even when a large phase change is required or when the nontrivial Floquet multiplier is close to unity; in such cases, the previously proposed control algorithm fails.

The aforementioned chapters focus on dynamics and control of a single oscillator. In Chapter 5, we formulate dynamics of a population of biological oscillators and develop control algorithms to change their collective behavior. Our formulation is based on the Fourier decomposition of the partial differential equation governing the evolution of the phase distribution of a population of identical, uncoupled oscillators. Our first two control algorithms in this chapter are Lyapunov-based, which work by decreasing a positive definite Lyapunov function towards zero. We construct a degenerate set of phase distributions and phase response curves for which the devised controls would not work. Our third control is an optimal control algorithm, which minimizes the control energy consumption while achieving the desired collective behavior of an oscillator population. This formulation results in high-dimensional Euler-Lagrange equations that we solve as a two point Boundary Value Problem (BVP) numerically. Since the BVP is high dimen-

sional, we construct a modified Newton Iteration method that is effective for our problem. Motivated by pathological neural synchrony, we apply our control to desynchronize an initially synchronized neural population. Given the proposed importance of enhancing spike time dependent plasticity to stabilize neural clusters and counteract pathological neural synchronization, we formulate the phase difference distribution in terms of the phase distribution, and prove some of its fundamental properties, and in turn apply our control to transform the neural phase distribution to form clusters. Finally, motivated by eliminating cardiac alternans, we apply our control to phase shift a synchronous cardiac pacemaker cell population. To demonstrate the effectiveness of our control for each of these applications, we show that a population of 100 phase oscillators with the applied control mimics the desired phase distribution.

The control algorithms developed in Chapters 4 and 5 are model-based, which can have limitations when the system under study is very complex and it is not possible to construct a model. Even if an accurate model could be built to describe the dynamics of such a system, developing a classical model-based control for such an underactuated system is a challenging task. If the parameters of the system change with time, or if the model doesn't describe the dynamics accurately, the theoretical control guarantees like stability and boundedness may not apply in real systems [32, 33]. This calls for data driven control methods like machine learning, which has spread to many fields in the recent years including control theory. However, the success of such algorithms has been dependent on availability of large datasets [34], which can be limited in fields like neuroscience where the cost of obtaining human/animal brain data is very high. Moreover, due to their black box nature, it is challenging to analyze how they work, which may be crucial in applications where safety is very important and failure is costly. Another limitation of such methods is their inability to take advantage of the inherent dynamics of the system to achieve the task, which limits their performance. All these limitations

call for a new machine learning control algorithm that doesn't rely on large amounts of data, is easy to understand, and can take advantage of the underlying dynamics in achieving the task. In Chapter 6, we have developed two related novel supervised learning algorithms based on these three goals. The algorithms are powerful enough to control a wide variety of complex underactuated dynamical systems, and yet have a simple and intelligent structure that allows them to work with a sparse data set even in the presence of noise. We demonstrate the versatility of our algorithms by applying them to a diverse range of applications including: switching between bistable states, changing the phase of an oscillator, desynchronizing a population of synchronized coupled oscillators, and stabilizing an unstable fixed point. For most of these applications we are able to reason why our algorithms work by using traditional dynamical systems and control theory. We also compare our learning algorithms with some traditional phase reduction based control algorithms, and reason out why our algorithms work based on phase reduction. Finally, we carry out a robustness analysis to demonstrate the effectiveness of our algorithms even in the presence of noise.

In Chapter 7, we conclude this dissertation by summarizing the work done, and discussing future research directions and experimental implementations of the developed algorithms in living biological tissue.

## **Publications**

This dissertation covers our work originally documented in the following peer reviewed publications:

- B. Monga, D. Wilson, T. Matchen and J. Moehlis, Phase Reduction and Phase-Based Optimal Control for Biological Systems: A Tutorial, *Biological Cybernetics*, 113 (2019), pp. 11 - 46.

- 
- B. Monga and J. Moehlis, Optimal Phase Control of Biological Oscillators using Augmented Phase Reduction, *Biological Cybernetics*, 113 (2019), pp. 161 - 178.
  - B. Monga, G. Froyland, and J. Moehlis, Synchronizing and Desynchronizing Neural Populations through Phase Distribution Control, in *2018 American Control Conference (ACC)*, June 2018, pp. 2808 - 2813.
  - B. Monga and J. Moehlis, Phase Distribution Control of a Population of Oscillators, *Physica D: Nonlinear Phenomena*, 398 (2019), pp. 115 - 129.
  - B. Monga and J. Moehlis, Supervised Learning Algorithms for Controlling Under-actuated Dynamical Systems, arXiv preprint arXiv:1909.11119.
  - B. Monga and J. Moehlis, Augmented Phase Reduction for Homoclinic Bifurcation and Relaxation Oscillators, (In preparation).

# Chapter 2

## Background

### 2.1 Introduction

Phase reduction, a powerful classical technique for the analysis of periodic orbits, is heavily employed in this dissertation. Thus we give background on phase reduction in this chapter, which is organized as follows. Section 2.2 describes the standard phase reduction and phase response curves for nonlinear oscillators. Section 2.3 covers an extension of standard phase reduction called augmented phase reduction, which includes the concept of isostable response curves for nonlinear oscillators.

### 2.2 Standard Phase Reduction and Control

Consider an autonomous vector field

$$\frac{d\mathbf{x}}{dt} = F(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n, \quad (n \geq 2) \tag{2.1}$$

having a stable hyperbolic periodic orbit  $\mathbf{x}^\gamma(t)$  with period  $T$ . The set of all points in the basin of attraction is defined as  $\mathcal{B}$ . For each point  $\mathbf{x}^*$  in  $\mathcal{B}$  there exists a unique  $\theta(\mathbf{x}^*)$  such that [35, 36, 17, 37, 19, 20, 38]

$$\lim_{t \rightarrow \infty} \left| \mathbf{x}(t) - \mathbf{x}^\gamma \left( t + \frac{T}{2\pi} \theta(\mathbf{x}^*) \right) \right| = 0, \quad (2.2)$$

where  $\mathbf{x}(t)$  is a trajectory starting with the initial point  $\mathbf{x}^*$ . The function  $\theta(\mathbf{x})$  is called the *asymptotic phase* of  $\mathbf{x}$ , and takes values in  $[0, 2\pi)$ . Other conventions, related to this through a simple rescaling, define the asymptotic phase to take values in  $[0, T)$  or in  $[0, 1)$ .

Let  $\mathbf{x}_0^\gamma$  be the point on the periodic orbit where the phase is zero. The typical convention is to choose  $\mathbf{x}_0^\gamma$  as corresponding to the global maximum of the first coordinate on the periodic orbit. An *isochron* is a level set of  $\theta(\mathbf{x})$ , that is, the collection of all points in the basin of attraction of  $\mathbf{x}^\gamma$  with the same asymptotic phase [3, 19]. We note that if  $\mathbf{x}(0)$  is a point on a periodic orbit, the isochron associated with that point is the set of all initial conditions  $\tilde{\mathbf{x}}(0)$  such that  $\|\mathbf{x}(t) - \tilde{\mathbf{x}}(t)\| \rightarrow 0$  as  $t \rightarrow \infty$ . Isochrons extend the notion of phase of a stable periodic orbit to the basin of attraction of the periodic orbit. It is conventional to define isochrons so that the phase of a trajectory on the periodic orbit advances linearly in time:

$$\frac{d\theta}{dt} = \frac{2\pi}{T} \equiv \omega \quad (2.3)$$

both on and off the periodic orbit.

Control theory seeks to design inputs to a dynamical system which change its behavior in a desired way. With this in mind, we consider the perturbed system

$$\frac{d\mathbf{x}}{dt} = F(\mathbf{x}) + U(t), \quad (2.4)$$

where  $U(t)$  is a small control input. The evolution of this system in terms of isochrons is [37, 18]

$$\frac{d\theta}{dt} = \frac{\partial\theta}{\partial\mathbf{x}} \cdot \frac{d\mathbf{x}}{dt} = \frac{\partial\theta}{\partial\mathbf{x}} \cdot (F(\mathbf{x}) + U(t)) = \omega + \frac{\partial\theta}{\partial\mathbf{x}} \cdot U(t).$$

Evaluating on the periodic orbit  $\mathbf{x}^\gamma$  for the unperturbed system gives, to leading order,

$$\frac{d\theta}{dt} = \omega + \mathcal{Z}(\theta) \cdot U(t), \quad \mathcal{Z}(\theta) = \left. \frac{\partial\theta}{\partial\mathbf{x}} \right|_{\mathbf{x}^\gamma(\theta)} \equiv \nabla_{\mathbf{x}^\gamma}\theta. \quad (2.5)$$

Here  $\mathcal{Z}(\theta) \in \mathbb{R}^n$  is the gradient of phase variable  $\theta$  evaluated on the periodic orbit, and is referred to as the (*infinitesimal*) *phase response curve (PRC)* [19, 39, 40, 28]. It quantifies the effect of an external perturbation on the phase of a periodic orbit. We call (2.5) the *standard phase reduction*.

## 2.2.1 Calculating Phase Response Curves

Given the importance of PRCs for phase reduction, we now describe several ways in which they can be calculated.

### Direct Method

The direct method [19, 41, 28] is the classical way to compute the PRC, which is useful especially in experimental studies. Letting  $\mathbf{x} = (x_1, x_2, \dots, x_n)$ , by definition

$$\left. \frac{\partial\theta}{\partial x_i} \right|_{\tilde{\mathbf{x}}^\gamma} = \lim_{\Delta x_i \rightarrow 0} \frac{\Delta\theta}{\Delta x_i}, \quad i = 1, \dots, n, \quad (2.6)$$

where  $\Delta\theta = \theta(\tilde{\mathbf{x}}^\gamma + \Delta x_i \hat{i}) - \theta(\tilde{\mathbf{x}}^\gamma)$  is the change in  $\theta(\mathbf{x})$  resulting from the perturbation  $\tilde{\mathbf{x}}^\gamma \rightarrow \tilde{\mathbf{x}}^\gamma + \Delta x_i \hat{i}$  from the base point  $\tilde{\mathbf{x}}^\gamma$  on the periodic orbit in the direction of the  $i^{\text{th}}$  coordinate. Since  $\dot{\theta} = 2\pi/T$  everywhere in the neighborhood of  $\mathbf{x}^\gamma$ , where the dot indicates  $\frac{d}{dt}$ , the difference  $\Delta\theta$  is preserved under the flow; thus, it may be measured in the

limit as  $t \rightarrow \infty$ , when the perturbed trajectory has collapsed back to the periodic orbit. That is,  $\left. \frac{\partial \theta}{\partial x_i} \right|_{\mathbf{x}^\gamma}$  can be found by comparing the phases of solutions in the infinite-time limit with initial conditions on and infinitesimally shifted from base points on  $\gamma$ .

### Adjoint Method

Another technique for finding the PRC involves solving an associated adjoint equation [42, 43, 18]:

$$\frac{d\nabla_{\mathbf{x}^\gamma(t)}\theta}{dt} = -DF^T(\mathbf{x}^\gamma(t))\nabla_{\mathbf{x}^\gamma(t)}\theta, \quad (2.7)$$

subject to the initial condition

$$\nabla_{\mathbf{x}^\gamma(0)}\theta \cdot \mathbf{F}(\mathbf{x}^\gamma(0)) = \omega. \quad (2.8)$$

Since  $\nabla_{\mathbf{x}^\gamma(t)}\theta$  evolves in  $\mathbb{R}^n$ , (2.8) supplies only one of  $n$  required initial conditions; the rest arise from requiring that the solution  $\nabla_{\mathbf{x}^\gamma(t)}\theta$  to (2.7) be  $T$ -periodic. This adjoint equation can be solved numerically with the program XPP [44] to find the PRC  $Q_{\text{XPP}}$ . Since XPP computes the PRC in terms of the change in time instead of the change in phase, we rescale the XPP PRC  $Q_{\text{XPP}}$  as

$$\nabla_{\mathbf{x}^\gamma}\theta = \omega Q_{\text{XPP}}.$$

### Example PRC Calculation: Thalamic Neuron Model

As an illustration, we calculate the PRC using both the direct method and the adjoint method for the thalamic neuron model [45] for the spiking behavior of neurons in the

thalamus:

$$\dot{v} = \frac{-I_L(v) - I_{Na}(v, h) - I_K(v, h) - I_T(v, r) + I_b}{C_m} + u(t), \quad (2.9)$$

$$\dot{h} = \frac{h_\infty(v) - h}{\tau_h(v)}, \quad (2.10)$$

$$\dot{r} = \frac{r_\infty(v) - r}{\tau_r(v)}. \quad (2.11)$$

In these equations  $I_b$  is the baseline current, which we take as  $5\mu A/cm^2$ ,  $v$  is the transmembrane voltage, and  $h$ ,  $r$  are the gating variables of the neuron which describe the modulation of the flow of ions across the neural membrane.  $u(t)$  represents the applied current as the control input. For details of the currents ( $I_L, I_{Na}, I_K, I_T$ ), functions  $h_\infty, \tau_h, r_\infty, \tau_r$  and the rest of the parameters, see Appendix A.1. With no control input, these parameters give a stable periodic orbit with period  $T = 8.3955$  ms.

The first (i.e., voltage) component of the PRC for this periodic orbit is shown in the right panel of Figure 2.1. In this figure, we used XPP to calculate the first component of the PRC from the adjoint method. For the direct method, a Matlab code was written where perturbations of size  $\delta v = -0.3$  were given at 20 points spread along the periodic orbit. Once the perturbed trajectories came reasonably close to the periodic orbit, spike time changes caused by the perturbations were scaled to obtain the corresponding phase changes, which when normalized by the magnitude of the perturbation gives the first component of the PRC.

## 2.3 Augmented Phase Reduction

The standard phase reduction (2.5) is valid only in a small neighborhood of the periodic orbit. Therefore, a control input derived based on the standard phase reduction can only be expected to be effective if its amplitude is small enough that it does not drive the

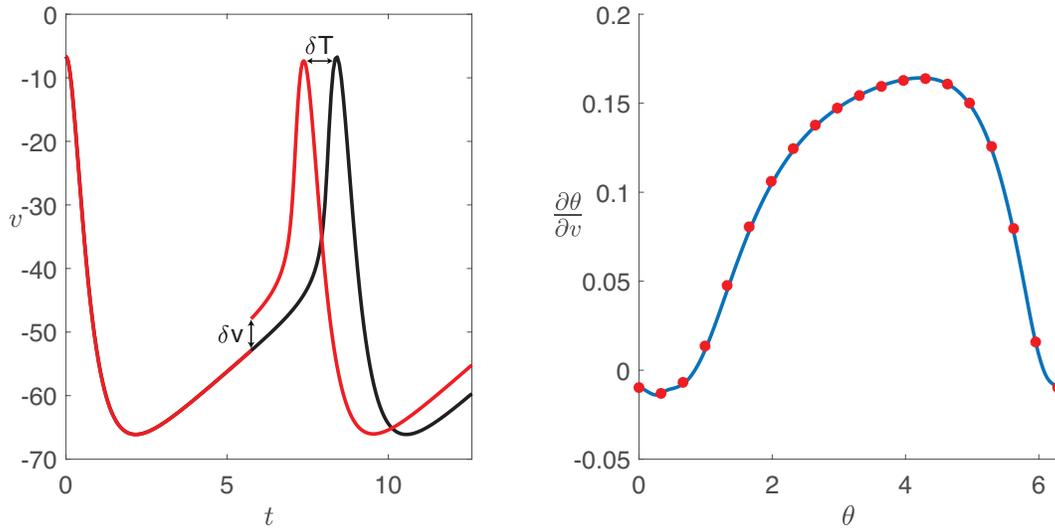


Figure 2.1: Thalamic neuron model: Left panel shows how the spike time changes by  $\delta T$  under an external perturbation  $\delta v$ . Here, black (resp., red) line shows the voltage under no (resp.,  $\delta v$ ) perturbation. In the right panel, the blue line (resp., red dots) shows the first component of the PRC computed from the adjoint (resp., the direct) method.

system far away from the periodic orbit. This limitation becomes even more important if the nontrivial Floquet multiplier, which describes the rate of decay of perturbations transverse to the periodic orbit, has magnitude close to unity [16]. This limits the ability to achieve certain control objectives and necessitates the use of *augmented phase reduction*, to be described below. Augmented phase reduction, an  $n$ -dimensional reduction, uses the concept of *isostables* for a periodic orbit [30], which are coordinates that give a sense of the distance in directions transverse to the periodic orbit. The first dimension captures the phase of the oscillator along the periodic orbit, as for the standard phase reduction, while the other  $n - 1$  dimensions give a measure of the oscillator's transversal distance from the periodic orbit along the  $n - 1$  isostable directions. The addition of these transversal coordinates allows one to design control algorithms which, while achieving the desired control objective, also keep the controlled trajectory close to the periodic

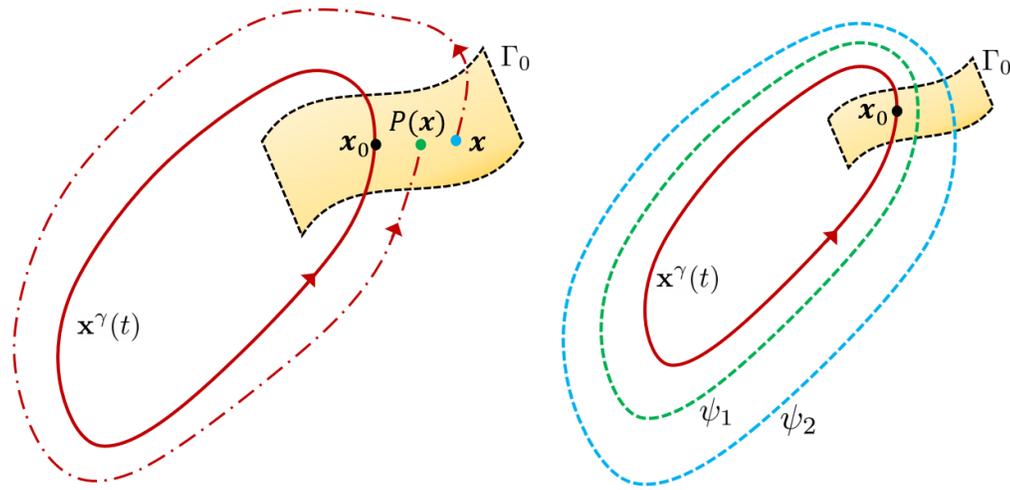


Figure 2.2: Isostables for a periodic orbit. The left panel shows the Poincaré map  $P$  on the isochron  $\Gamma_0$  of the periodic orbit  $\mathbf{x}^\gamma(t)$ . The trajectory starting from  $\mathbf{x}$  on the isochron lands back on the isochron at  $P(\mathbf{x})$  after one period. The right panel visualizes the isostables as giving a sense of transversal distance from the periodic orbit by showing two isostable level sets  $\psi_1$  and  $\psi_2$ .

orbit [16].

For systems which have a stable fixed point, it can be useful to define *isostables* [31], which are sets of points in phase space that approach the fixed point together and are analogous to isochrons for asymptotically periodic systems. Isostables are related to the eigenfunctions of the Koopman operator [31]. The notion of isostables was recently adapted for systems having a stable periodic orbit [30], where isostables were defined to be the set of points that approach a periodic orbit together. They give a sense of the distance in directions transverse to the periodic orbit, visualized in the right panel of Figure 2.2. Standard phase reduction can be augmented with these coordinates as follows.

Consider a point  $\mathbf{x}_0$  on the periodic orbit  $\mathbf{x}^\gamma(t)$  with the corresponding isochron  $\Gamma_0$ . The transient behavior of the system (2.4) near  $\mathbf{x}_0$  can be analyzed by a Poincaré map  $P$  on

$\Gamma_0$ ,

$$P : \Gamma_0 \rightarrow \Gamma_0; \quad \mathbf{x} \rightarrow P(\mathbf{x}). \quad (2.12)$$

This is shown in the left panel of Figure 2.2. Here  $\mathbf{x}_0$  is a fixed point of this map, and we can approximate  $P$  in a small neighborhood of  $\mathbf{x}_0$  as

$$P(\mathbf{x}) = \mathbf{x}_0 + DP(\mathbf{x} - \mathbf{x}_0) + O(\|\mathbf{x} - \mathbf{x}_0\|^2), \quad (2.13)$$

where  $DP = dP/dx|_{\mathbf{x}_0}$ . Suppose  $DP$  is diagonalizable with  $V \in \mathbb{R}^{n \times n}$  as a matrix with columns of unit length eigenvectors  $\{v_i | i = 1, \dots, n\}$  and the associated eigenvalues  $\{\lambda_i | i = 1, \dots, n\}$  of  $DP$ . These eigenvalues  $\lambda_i$  are the Floquet multipliers of the periodic orbit. For every nontrivial Floquet multiplier  $\lambda_i$ , with the corresponding eigenvector  $v_i$ , the set of isostable coordinates is defined as [30]

$$\psi_i(\mathbf{x}) = e_i^T V^{-1}(\mathbf{x}_\Gamma - \mathbf{x}_0) \exp(-\log(\lambda_i)t_\Gamma/T), \quad i = 1, \dots, n-1. \quad (2.14)$$

Here  $\mathbf{x}_\Gamma$  and  $t_\Gamma \in [0, T)$  are defined to be the position and the time at which the trajectory first returns to the isochron  $\Gamma_0$ , and  $e_i$  is a vector with 1 in the  $i^{\text{th}}$  position and 0 elsewhere. As shown in [30], we get the following equations for  $\psi_i$  and its gradient  $\nabla_{\mathbf{x}^\gamma(t)}\psi_i$  under the flow  $\dot{\mathbf{x}} = F(\mathbf{x})$ :

$$\dot{\psi}_i = k_i \psi_i, \quad (2.15)$$

$$\frac{d\nabla_{\mathbf{x}^\gamma(t)}\psi_i}{dt} = (k_i I - DF(\mathbf{x}^\gamma(t))^T) \nabla_{\mathbf{x}^\gamma(t)}\psi_i, \quad (2.16)$$

where  $k_i = \log(\lambda_i)/T$  is the  $i^{\text{th}}$  nontrivial Floquet exponent,  $DF$  is the Jacobian of  $F$ , and  $I$  is the identity matrix. We refer to this gradient  $\nabla_{\mathbf{x}^\gamma(t)}\psi_i \equiv \mathcal{I}_i(\theta)$  as the *isostable response curve (IRC)*. Its  $T$ -periodicity along with the normalization condition

$\nabla_{\mathbf{x}_0} \psi_i \cdot v_i = 1$  gives a unique IRC. It gives a measure of the effect of a control input in driving the trajectory away from the periodic orbit. The  $n$ -dimensional system (given by equation (2.4)) can be realized as [30]

$$\dot{\theta} = \omega + \mathcal{Z}^T(\theta) \cdot U(t), \quad (2.17)$$

$$\dot{\psi}_i = k_i \psi_i + \mathcal{I}_i^T(\theta) \cdot U(t), \quad \text{for } i = 1, \dots, n-1. \quad (2.18)$$

We refer to this reduction as the augmented phase reduction. Here, the phase variable  $\theta$  indicates the position of the trajectory along the periodic orbit, and the isostable coordinate  $\psi_i$  gives information about transversal distance from the periodic orbit along the  $i^{\text{th}}$  eigenvector  $v_i$ . It is evident from (2.17, 2.18) that an external perturbation affects the oscillator's phase through the PRC, and its transversal distance to the periodic orbit through the IRC. In practice, isostable coordinates with nontrivial Floquet multiplier close to 0 can be ignored as perturbations in those directions are nullified quickly under the evolution of the vector field. If all isostable coordinates are ignored, the augmented phase reduction reduces to the standard phase reduction.

### 2.3.1 Calculating Isostable Response Curves

Given the importance of IRCs for the augmented phase reduction, we now describe several ways in which they can be calculated.

#### Direct Method

PRCs are calculated by the direct method by giving perturbations to the oscillator at various phases, and recording the phase change caused by the perturbation as a function of the stimulation phase. IRCs can be measured in a similar way. Perturbations ( $\tilde{\mathbf{x}}^\gamma + \Delta x_i \hat{i}$ ) are applied at various phases along the periodic orbit in the direction of the  $i^{\text{th}}$

coordinate. A time series of crossings of the  $\Gamma_0$  isochron,  $t_\Gamma^j$ , as well as the crossing locations,  $\mathbf{x}_\Gamma^j$  are recorded. This information is used with the definition of isostable in (2.14) to calculate the isostable change  $\Delta\psi$  caused by the perturbation, which when scaled by the magnitude of the perturbation yields the IRC.

### Adjoint Method

Unlike solving for the PRC, backwards integration of equation (2.16) will result in positive Floquet exponents, and hence a periodic solution that is unstable. We have therefore found it useful to formulate the calculation as a boundary value problem and solve it with Newton iteration; see Appendix C. The first step is to compute and save the periodic solution  $\mathbf{x}^\gamma(t)$  using an ODE solver. For the two point boundary value formation, we take the boundary conditions as  $\mathcal{I}(0) = \mathcal{I}(T)$ . For Newton iteration, we take

$$\begin{aligned} c^\nu &= \mathcal{I}(0), \\ g(c^\nu) &= \mathcal{I}(0) - \mathcal{I}(T), \\ \left. \frac{\partial g}{\partial c} \right|_{c^\nu} &= Id - J, \end{aligned}$$

where  $Id$  is the identity matrix, and  $J$  is the Jacobian matrix

$$J = \frac{\partial \mathcal{I}(T)}{\partial \mathcal{I}(0)},$$

which is computed numerically. Once a periodic solution is obtained, the computed IRC is scaled by the normalization condition  $\nabla_{\mathbf{x}_0} \psi \cdot v_i = 1$ .

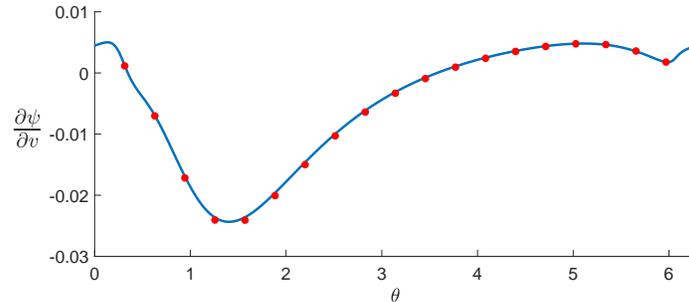


Figure 2.3: IRC for the thalamic neuron model: the blue line (resp., red dots) shows IRC in response to voltage perturbations computed from the adjoint (resp., the direct) method.

### Example IRC Calculation: Thalamic Neuron Model

As an illustration, we calculate the IRC using both the direct method and the adjoint method for the thalamic neuron model given by equations (2.9-2.11) with the same parameters as before. Those parameters give a stable periodic orbit with time period  $T = 8.3955 \text{ ms}$  and nontrivial Floquet multipliers 0.8275 and 0.0453. Since one of the nontrivial Floquet multiplier is close to 0, we only consider the isostable coordinate corresponding to the larger nontrivial Floquet multiplier in the augmented phase reduction. To calculate the IRC by the adjoint method, we solve the corresponding adjoint equation as a two-point boundary value problem. For the direct method, a Matlab code was written where perturbations of size  $\delta v = -0.3$  were given at 20 points spread along the periodic orbit. Once the perturbed trajectories came reasonably close to the periodic orbit, the corresponding isostable change was calculated, which when normalized by the magnitude of the perturbation gives the first component of the IRC. The first (i.e., voltage) component of the IRC for the periodic orbit is shown in Figure 2.3.

# Chapter 3

## On Augmented Phase Reduction: Analytical and Numerical Results

### 3.1 Introduction

As previously discussed, Standard phase reduction is valid only in a small neighborhood of the periodic orbit. Consequently, the magnitude of the allowable control stimulus is limited by the nontrivial Floquet multiplier [29] of the periodic orbit: in systems with a small-magnitude negative nontrivial Floquet exponent, even a relatively small perturbation can lead to a trajectory which stays away from the periodic orbit, rendering the phase reduction inaccurate and phase reduction based control ineffective. This necessitates the use of augmented phase reduction [30], a two-dimensional reduction based on both isochrons and isostables [31]. While the first dimension captures the phase of the oscillator along the periodic orbit, like the standard phase reduction, the second dimension captures the oscillator's transversal approach to the periodic orbit. This reduction ascertains the effect of an external stimulus on the oscillator's phase change through the PRC, and the change in its transversal distance to the periodic orbit through the

isostable response curve (IRC). An equivalent reduction based on the Koopman operator is given in [46].

This gives the same outcome as the phase-amplitude reduction devised in [47] for planar systems, but the augmented phase reduction does not require computationally intensive calculation of a coordinate system with respect to periodic orbit of dimensionality greater than 2. Moreover, the phase-amplitude description devised in [48] is not explicitly dependent on the Floquet multipliers of the system, whereas the augmented phase reduction is. This dependency on Floquet multipliers is advantageous in higher dimensional systems, where the periodic orbit is weakly stable in only a few directions, as it allows us to reduce the dimensionality of the augmented phase reduction to capture transversal dynamics only along the weakly stable directions. The use of Floquet coordinates [49] results in a similar reduction, but an additional step is required to quantify the effect of an external perturbation on the oscillator's dynamics. It also requires the knowledge of the whole state space dynamics along the periodic orbit, which might not be observable in an electrophysiological setting. On the other hand, for our algorithm, the response functions that arise for augmented phase reduction in principle can be measured in an electrophysiological setting; indeed, we envision that IRCs can be measured experimentally just like PRCs, making the control based on the augmented phase reduction experimentally amenable as well. Control algorithms based on the augmented phase reduction are expected to be more effective [30, 16], as they can be designed to allow a larger stimulus without the risk of driving the oscillator too far away from the periodic orbit.

In this chapter, we analytically calculate the augmented phase reduction for periodic orbits of planar systems having distinct dynamics. Specifically, we derive expressions for  $\lambda - \omega$  systems, relaxation oscillators, and systems in which periodic orbits are born out of four codimension one bifurcations, the last four systems being the normal form

of a supercritical Hopf bifurcation, the normal form of a Bautin bifurcation which has a saddle-node bifurcation of limit cycles, and simple two-dimensional models undergoing SNIPER and homoclinic bifurcations. Our contribution is the analytical calculation of IRCs and the nontrivial Floquet exponent for each of these six systems, and the PRC for the simple model undergoing SNIPER bifurcation. A similar analysis was done in [50, 18], where analytical expressions of PRCs were derived for  $\lambda - \omega$  systems, and for systems undergoing the stated codimension one bifurcations, respectively. While the authors in [18] considered a one dimensional model of SNIPER bifurcation for the PRC calculation, here we consider a two-dimensional model, as a minimum of two dimensions is necessary for the augmented phase reduction. My approach for the IRC calculation for a relaxation oscillator is in line with Izhikevich's analysis [51] for the calculation of the PRC for such systems. To validate our calculations, we simulate six different models in these regimes, and compare their numerically computed augmented phase reduction with the derived analytical expressions. These analytical and numerical calculations help us understand under which dynamical regimes is the use of augmented phase reduction advantageous over the standard phase reduction.

The models that we calculate the augmented phase reduction for are two-dimensional, so there is only one isostable coordinate. We thus write the adjoint equation as

$$\frac{d\nabla_{\mathbf{x}^\gamma(t)}\psi}{dt} = (kI - DF(\mathbf{x}^\gamma(t))^T) \nabla_{\mathbf{x}^\gamma(t)}\psi, \quad (3.1)$$

and the augmented phase reduction as

$$\dot{\theta} = \omega + \mathcal{Z}^T(\theta) \cdot U(t), \quad (3.2)$$

$$\dot{\psi} = k\psi + \mathcal{I}^T(\theta) \cdot U(t), \quad (3.3)$$

where  $\mathcal{I}(\theta) \equiv \nabla_{\mathbf{x}^\gamma(t)} \psi$ . We have removed the subscript for  $\psi$  and  $k$ , as we only have one isostable coordinate. The eigenvector  $v$  is then the unit vector along the one-dimensional isochron  $\Gamma_0$ . The nontrivial Floquet exponent  $k$  can then be computed from the divergence of the vector field as [52]

$$k = \frac{\int_0^T \nabla \cdot F(\mathbf{x}^\gamma(t)) dt}{T}. \quad (3.4)$$

This chapter is organized as follows. In Section 3.2, we analytically calculate the augmented phase reduction for the six systems, and simulate six different models under the appropriate regimes to validate our calculations. Section 3.3 concludes the chapter by summarizing the derived analytical expressions, and discussing their implications.

The main results of this chapter have been published in [38, 53].

## 3.2 Analytical and Numerical Computation of the Augmented Phase Reduction

Bifurcation theory [29, 54] identifies four codimension one bifurcations which give birth to a stable limit cycle for generic families of vector fields: a supercritical Hopf bifurcation, a saddle-node bifurcation of limit cycles, a SNIPER bifurcation (saddle-node bifurcation of fixed points on a periodic orbit, also called a SNIC bifurcation), and a homoclinic bifurcation. These bifurcations are illustrated in Figure 3.1. Figure 3.1 also shows a relaxation oscillator, where dynamics for one of the variables is considerably faster than the other.

In this section, we analyze planar dynamical systems which have a stable limit cycle which arises from these four codimension one bifurcations, specifically the normal form for a supercritical Hopf bifurcation, the normal form for a Bautin bifurcation which has a

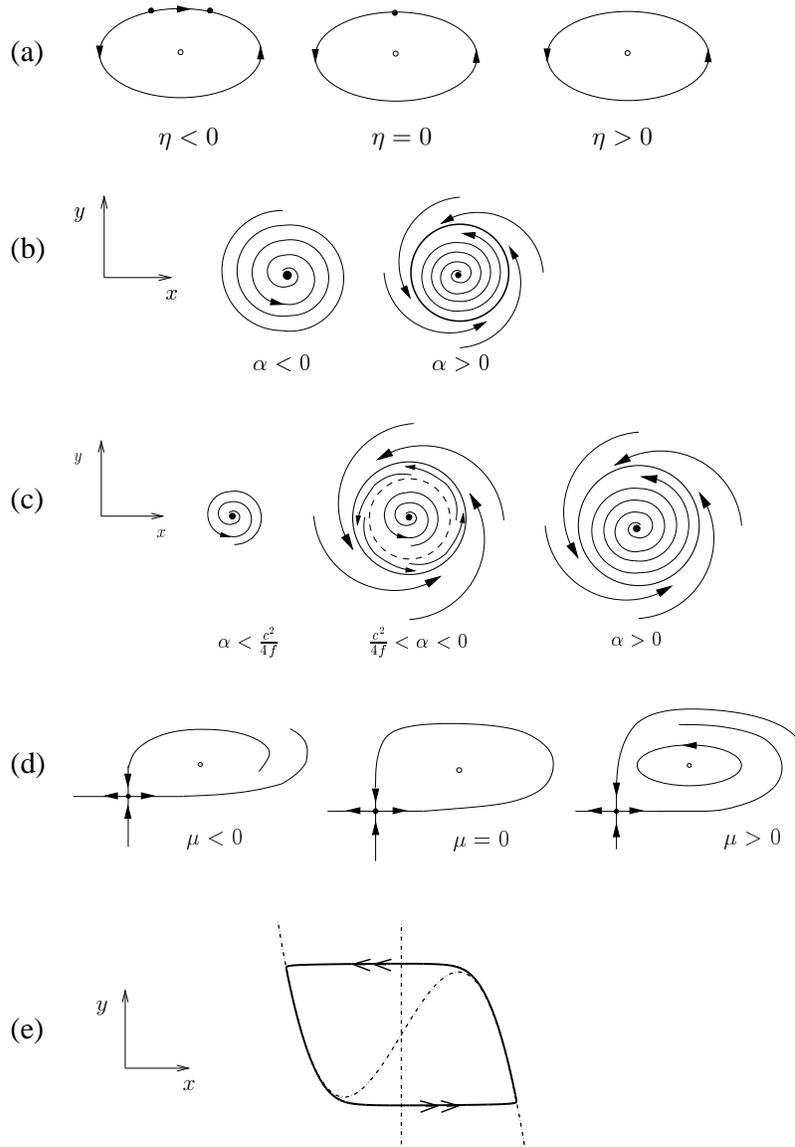


Figure 3.1: (a) SNIPER bifurcation: Two fixed points die in a saddle-node bifurcation at  $\eta = 1$ , giving a periodic orbit for  $\eta > 1$ , assumed to be stable. (b) Supercritical Hopf bifurcation: A fixed point loses stability as  $a$  increases through zero, giving a stable periodic orbit (closed curve). (c) Bautin bifurcation: There is a subcritical Hopf bifurcation at  $a = 0$ , and a saddle-node bifurcation of periodic orbits at  $a = c^2/4f$ . Both a stable (solid closed curve) and unstable (dashed closed curve) periodic orbit exist for  $c^2/4f < a < 0$ . The fixed point is stable (resp., unstable) for  $a < 0$  (resp.,  $a > 0$ ). (d) Homoclinic bifurcation: A homoclinic orbit exists at  $\mu = 0$ , giving rise to a stable periodic orbit for  $\mu > 0$ . (e) A relaxation oscillator (solid closed curve) is shown with its nullclines (dashed curves).

saddle-node bifurcation of limit cycles, and simple two-dimensional models undergoing a SNIPER and a homoclinic bifurcations. We first consider  $\lambda - \omega$  systems, of which Hopf and Bautin normal forms are specific examples. We also consider relaxation oscillators with fast-slow dynamics. We derive analytical expressions of the augmented phase reduction (3.2, 3.3) for these systems. Specifically, our contributions are the calculation of the nontrivial Floquet exponent  $k$  and IRCs for all six dynamical systems, and the PRC for the system undergoing a SNIPER bifurcation. Similar calculations for the  $\lambda - \omega$  system, and for the system undergoing a SNIPER bifurcation have been done in [55] using different methods.

To validate the calculations, we simulate six different models whose dynamics are expected to be captured by the aforementioned planar systems. We compare their numerically computed IRCs (and PRC for SNIPER case) with the derived analytical expressions. In the numerical computation of the IRCs for the planar systems, we directly calculate the nontrivial Floquet exponent  $k$  as the mean of the divergence of vector field along the periodic orbit according to (3.4). On the other hand, for higher dimensional models, we first compute the PRC using the software XPP [44], then choose an arbitrary point on the periodic orbit as  $\theta = 0$ , and approximate the isochron as a vector orthogonal to the PRC at that point. To compute the Jacobian  $DF$ , we compute  $\mathbf{x}_\mathbf{r}$  for a number of initial conditions  $\mathbf{x}_\mathbf{0}$  spread out on the isochron. Eigenvector decomposition of  $DF$  gives us the Floquet multipliers of the periodic orbit and thus  $k$ . After obtaining  $k$ , we use Newton iteration to obtain the IRC as the periodic solution to equation (3.1). Note that the higher dimensional systems we consider for numerical simulation in this section have only one negative small magnitude nontrivial Floquet exponent, so the reduction given by (3.2,3.3) still applies.

### 3.2.1 $\lambda - \omega$ systems

The normal forms for Hopf and Bautin bifurcations are special cases of  $\lambda - \omega$  systems [56, 57]. Therefore, we first analyze these general dynamical systems and calculate their IRC expressions, which we then will use to evaluate for the IRCs for the Hopf and Bautin normal forms. Some of these results are similar, but obtained using a different method, to the results in [55]. The general form of  $\lambda - \omega$  systems is most conveniently written as

$$\dot{r} = G(r), \quad (3.5)$$

$$\dot{\phi} = H(r), \quad (3.6)$$

where  $r$  and  $\phi$  are the standard polar coordinates in two dimensions. We assume there is a stable periodic orbit with radius  $r_{po}$  found from  $G(r_{po}) = 0$ , and angular frequency  $\omega = H(r_{po})$ . The PRC of the periodic orbit can be written in polar coordinates as [50]

$$\left( \frac{\partial \theta}{\partial r}, \frac{\partial \theta}{\partial \phi} \right) = \left( -\frac{H'(r_{po})}{G'(r_{po})}, 1 \right). \quad (3.7)$$

This implies that the phase coordinate  $\theta$  is equal to the azimuthal coordinate  $\phi$  on the periodic orbit. Transforming to Cartesian coordinates  $(x, y) = (r \cos \phi, r \sin \phi)$ , the PRC can be written as

$$\mathcal{Z}(\theta) = \left( -\frac{H'(r_{po})}{G'(r_{po})} \cos \theta - \frac{\sin \theta}{r_{po}} \right) \hat{x} + \left( -\frac{H'(r_{po})}{G'(r_{po})} \sin \theta + \frac{\cos \theta}{r_{po}} \right) \hat{y}. \quad (3.8)$$

At a point  $(x, y) = (r_{po}, 0) \equiv (x_0, y_0)$ , the isochron is in the direction orthogonal to the PRC (surfaces of constant phase are orthogonal to the gradient of the phase). Thus the

eigenvector  $v$  is given as

$$\frac{\left(-\frac{1}{r_{po}}\right) \hat{x} + \left(-\frac{H'(r_{po})}{G'(r_{po})}\right) \hat{y}}{\sqrt{\left(\frac{H'(r_{po})}{G'(r_{po})}\right)^2 + \frac{1}{r_{po}^2}}}. \quad (3.9)$$

We will use this vector in the normalization condition for the IRC below. The IRC (in polar coordinates:  $\frac{\partial \psi}{\partial r} \hat{r} + \frac{\partial \psi}{\partial \phi} \hat{\phi} = \mathcal{I}_r \hat{r} + \mathcal{I}_\phi \hat{\phi}$ ) can be found by solving the adjoint equation subject to  $T$ -periodicity and normalization condition as:

$$\dot{\mathcal{I}}_r = (k - G'(r_{po})) \mathcal{I}_r - H'(r_{po}) \mathcal{I}_\phi, \quad (3.10)$$

$$\dot{\mathcal{I}}_\phi = k \mathcal{I}_\phi, \quad (3.11)$$

$$\Rightarrow \mathcal{I}_\phi = \mathcal{I}_{\phi_0} e^{kt}. \quad (3.12)$$

From the mean of the divergence of the vector field along the periodic orbit, see (3.4), we get  $k = G'(r_{po})$ . Since  $\mathcal{I}_\phi$  and  $\mathcal{I}_r$  are  $T$ -periodic, we must have  $\mathcal{I}_{\phi_0} = 0$ . Thus the IRC in polar and Cartesian coordinates is

$$\mathcal{I}_{r,\phi} = \mathcal{I}_{r_0} \hat{r} + 0 \hat{\phi}, \quad (3.13)$$

$$\mathcal{I}_{x,y} = \mathcal{I}_{r_0} \cos \theta \hat{x} + \mathcal{I}_{r_0} \sin \theta \hat{y}. \quad (3.14)$$

To find the constant  $\mathcal{I}_{r_0}$ , we use the normalization condition at point  $(x_0, y_0)$

$$\mathcal{I}_{x_0, y_0} \cdot v = 1. \quad (3.15)$$

$$\Rightarrow \{\mathcal{I}_{r_0} \hat{x} + 0 \hat{y}\} \cdot \left\{ \frac{\left(-\frac{1}{r_{po}}\right) \hat{x} + \left(-\frac{H'(r_{po})}{G'(r_{po})}\right) \hat{y}}{\sqrt{\left(\frac{H'(r_{po})}{G'(r_{po})}\right)^2 + \frac{1}{r_{po}^2}}} \right\} = 1, \quad (3.16)$$

$$\Rightarrow \mathcal{I}_{r_0} = -\sqrt{1 + \frac{r_{po}^2 H'(r_{po})^2}{G'(r_{po})^2}}. \quad (3.17)$$

This gives the IRC in polar and Cartesian coordinates as

$$\mathcal{I}_{r,\phi} = -\sqrt{1 + \frac{r_{po}^2 H'(r_{po})^2}{G'(r_{po})^2}} \hat{r} + 0 \hat{\phi}, \quad (3.18)$$

$$\mathcal{I}_{x,y} = -\sqrt{1 + \frac{r_{po}^2 H'(r_{po})^2}{G'(r_{po})^2}} (\cos \theta \hat{x} + \sin \theta \hat{y}). \quad (3.19)$$

We see that the Cartesian components of the IRC for a  $\lambda$ - $\omega$  system each take positive and negative values, depending on the value of the phase  $\theta$ . Thus, the same instantaneous, infinitesimal perturbation can either increase or decrease the isostable coordinate (moving the trajectory inward or outward from the periodic orbit, in the sense of isostables), depending on when it is applied.

We now consider two special cases of  $\lambda - \omega$  systems.

### 3.2.2 Hopf Bifurcation

The normal form for a supercritical Hopf bifurcation [29, 58] in Cartesian coordinates is given as:

$$\dot{x} = ax - by + (x^2 + y^2)(cx - dy), \quad (3.20)$$

$$\dot{y} = bx + ay + (x^2 + y^2)(dx + cy). \quad (3.21)$$

This can be written in polar coordinates as:

$$\dot{r} = ar + cr^3, \quad (3.22)$$

$$\dot{\phi} = b + dr^2. \quad (3.23)$$

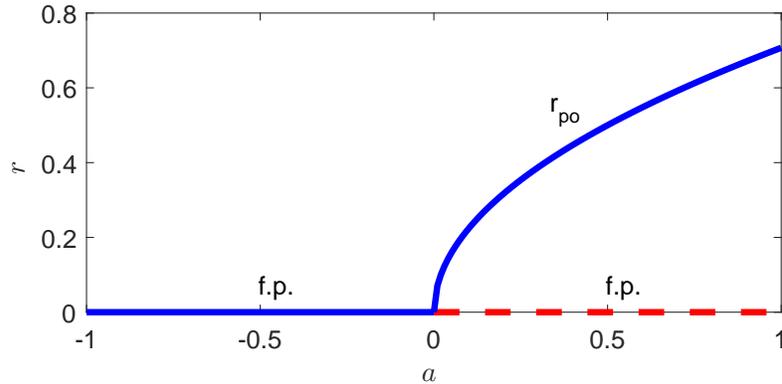


Figure 3.2: Supercritical Hopf normal form bifurcation diagram for  $c = -2$ . Solid blue (resp., dashed red) lines show stable (resp., unstable) solutions.

Thus, the Hopf normal form is a  $\lambda - \omega$  system, with  $G(r) = ar + cr^3$  and  $H(r) = b + dr^2$ . With parameters  $c < 0$ , and  $a < 0$ , the system has a stable fixed point. As  $a$  increases through 0, a stable periodic orbit is born, and the fixed point loses stability. This is shown in Figure 3.2. For  $a > 0$ , the radius of the stable periodic orbit is  $r_{po} = \sqrt{-a/c}$ , and its time period is given by  $T = 2\pi / (b + dr_{po}^2)$ . Using equations (3.18, 3.19), we get the IRC as

$$\mathcal{I}_{x,y} = -\sqrt{1 + \frac{d^2}{c^2}} (\cos \theta \hat{x} + \sin \theta \hat{y}). \quad (3.24)$$

Thus we get a sinusoidal IRC. We note that a special case of this problem was considered using different methods in Example 5.1 from [55].

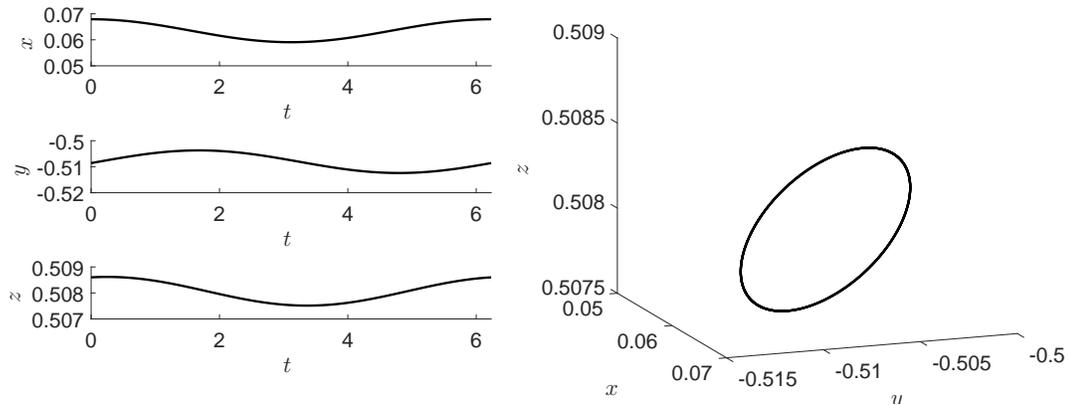


Figure 3.3: Periodic orbit and trajectories for the Rössler system with parameters  $a = 0.125$ ,  $b = 2$ , and  $c = 4$ .

### Rössler system

We use the Rössler system [59] to validate our IRC calculations:

$$\dot{x} = -y - z, \quad (3.25)$$

$$\dot{y} = x + ay, \quad (3.26)$$

$$\dot{z} = b + z(x - c). \quad (3.27)$$

With parameters  $a = 0.125$ ,  $b = 2$ , and  $c = 4$ , we get a stable periodic orbit with time period  $T = 6.2331$ , and nontrivial Floquet multipliers  $\approx 1$ , and  $5.14 \times 10^{-9}$ . Since one of the nontrivial Floquet multipliers is close to 0, we only consider the isostable coordinate corresponding to the larger nontrivial Floquet multiplier in the augmented phase reduction. The nontrivial Floquet exponent comes out to be  $k = -1.0543 \times 10^{-7}$ . Figure 3.3 shows the periodic trajectories and orbit for the Rössler system with the given parameter values. Figure 3.4 compares the numerically computed IRC (blue lines) for the Rössler system near a supercritical Hopf bifurcation with the red sinusoidal curves. We see that the both curves overlap, validating our analytical calculation of a sinusoidal

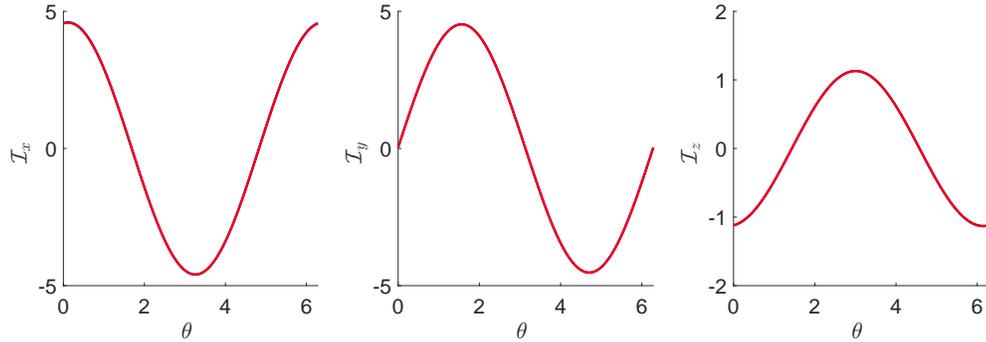


Figure 3.4: IRC for the Rössler system near a supercritical Hopf bifurcation: The blue line (nearly invisible beneath the red line) shows the numerically computed IRCs, while the red line shows the best matching sinusoid curve. The left, middle and right panels show the  $x$ ,  $y$ , and  $z$  components of the IRC, respectively.

IRC for a periodic orbit near a supercritical Hopf bifurcation.

### 3.2.3 Saddle-node bifurcation of periodic orbits

The Bautin normal form [54, 60] can capture a saddle-node bifurcation of periodic orbits, where an unstable branch of periodic orbits born out of a subcritical Hopf bifurcation turns around and gains stability. It is given in Cartesian coordinates as:

$$\dot{x} = ax - by + (x^2 + y^2) (cx - dy + (x^2 + y^2)(fx - gy)) , \tag{3.28}$$

$$\dot{y} = bx + ay + (x^2 + y^2) (dx + cy + (x^2 + y^2)(gx + fy)) . \tag{3.29}$$

This can be written in polar coordinates as:

$$\dot{r} = ar + cr^3 + fr^5, \tag{3.30}$$

$$\dot{\phi} = b + dr^2 + gr^4. \tag{3.31}$$

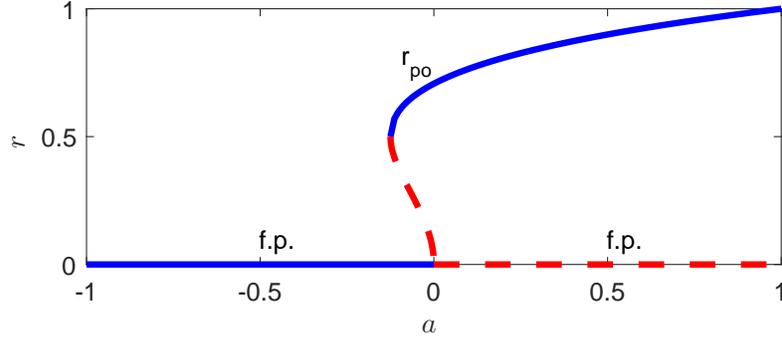


Figure 3.5: Bautin normal form bifurcation diagram for  $c = 1$ , and  $f = -2$ . Solid blue (resp., dashed red) lines show stable (resp., unstable) solutions.

The Bautin normal form is thus a  $\lambda - \omega$  system, with  $G(r) = ar + cr^3 + fr^5$  and  $H(r) = b + dr^2 + gr^4$ . With parameters  $c > 0$ ,  $f < 0$ , and  $a > 0$ , the system has an unstable fixed point, and a stable periodic orbit. As  $a$  decreases through 0, an unstable periodic orbit is born in a subcritical Hopf bifurcation, and the fixed point becomes stable. As  $a$  decreases further, the stable and unstable periodic orbits in a saddle-node bifurcation of periodic orbits at  $a = c^2/4f$ . The bifurcation diagram is shown in Figure 3.5. In this chapter, we consider the stable periodic orbit with radius  $r_{po} = \sqrt{\frac{-c - \sqrt{c^2 - 4af}}{2f}}$ , and time period  $T = 2\pi / (b + dr_{po}^2 + gr_{po}^4)$ . Using equations (3.18, 3.19), we get the IRC in polar and Cartesian coordinates as

$$\mathcal{I}_{r,\phi} = -\sqrt{1 + r_{po}^2 \left( \frac{2dr_{po} + 4gr_{po}^3}{a + 3cr_{po}^2 + 5fr_{po}^4} \right)^2} \hat{r} + 0 \hat{\phi}, \quad (3.32)$$

$$\mathcal{I}_{x,y} = -\sqrt{1 + r_{po}^2 \left( \frac{2dr_{po} + 4gr_{po}^3}{a + 3cr_{po}^2 + 5fr_{po}^4} \right)^2} (\cos \theta \hat{x} + \sin \theta \hat{y}). \quad (3.33)$$

### Thalamic neuron model

The thalamic neuron model [45] introduced in Chapter 2 describes the spiking behavior of neurons inside the thalamus. For details of the model, see Appendix A.1.

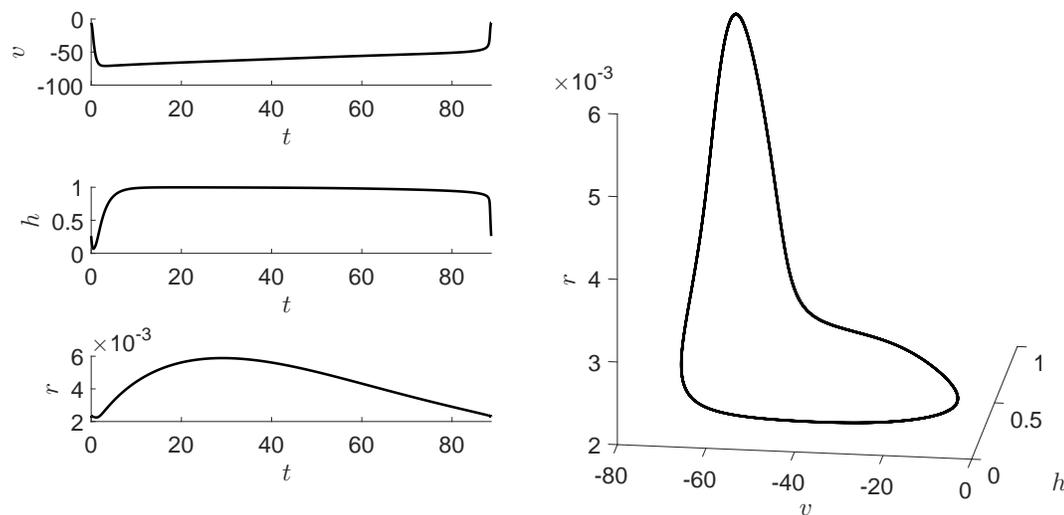


Figure 3.6: Periodic orbit for the thalamic neuron model system with  $I_b = 0.3281$ .

With model parameter  $I_b = 0.3281 \mu A/cm^2$ , there exists a stable periodic orbit near a saddle-node bifurcation of periodic orbits with time period  $T = 88.6816$  ms and nontrivial Floquet multipliers 0.8415 and  $2.584 \times 10^{-8}$ . Since one of the nontrivial Floquet multiplier is close to 0, we only consider the isostable coordinate corresponding to the larger nontrivial Floquet multiplier in the augmented phase reduction. The nontrivial Floquet exponent comes out to be  $k = -0.001946$ . Figure 3.6 shows the periodic orbit for the thalamic neuron model with the given parameter values. Figure 3.7 shows the numerically computed IRC for the thalamic neuron model for these parameters. We see that the IRC along the voltage  $v$  and gating variable  $r$  match closely with a sinusoid, whereas the IRC along the gating variable  $h$  does not. Instead it resembles the IRC for a relaxation oscillator (see section 3.2.6). This is because the Bautin normal form only captures the turning around of an unstable periodic orbit branch born out of a subcritical Hopf bifurcation and gaining stability in a saddle-node bifurcation of periodic orbits. However, it does not capture the relaxation nature of dynamics present in some models,

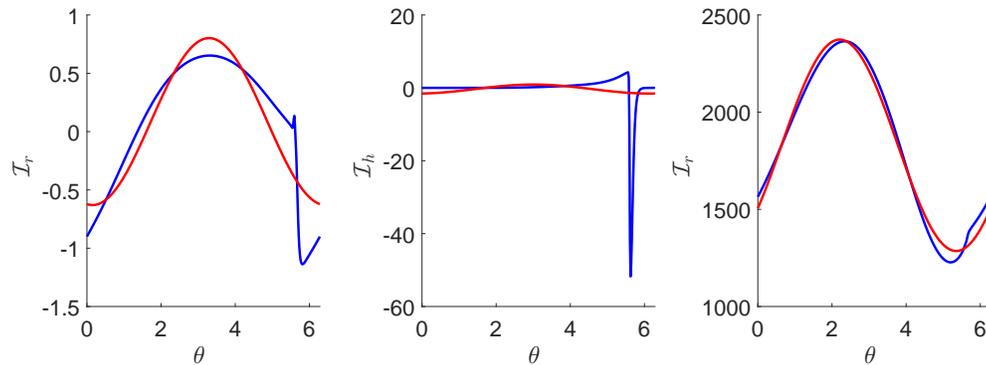


Figure 3.7: IRC for the thalamic neuron model near the saddle-node bifurcation of periodic orbits: The blue line shows the numerically computed IRCs, while the red line shows the best matching sinusoid curve. The left, middle and right panels show the  $v$ ,  $h$ , and  $r$  components of the IRC and its closest matching sinusoid, respectively.

including this one. That is why the IRC computed numerically for such models does not match closely in shape with the derived analytical expression, cf. [61]. The variables  $x$ , and  $y$  in the Bautin normal form vary at a similar rate, but the variables  $v$ , and  $h$  in the thalamic neuron model vary at a much faster rate than the variable  $r$ . Thus one component of the IRC shows a shape similar to that of a relaxation oscillator (see section 3.2.6).

### 3.2.4 SNIPER bifurcation

The SNIPER (Saddle-Node Infinite PERiod) bifurcation [29, 54], also sometimes called SNIC (Saddle-Node on Invariant Circle) bifurcation, takes place when a saddle-node bifurcation of fixed points occurs on a periodic orbit. A simple model for the

SNIPER bifurcation is

$$\dot{x} = x(\rho - x^2 - y^2) - y \left( \eta - \frac{y}{\sqrt{x^2 + y^2}} \right), \quad (3.34)$$

$$\dot{y} = y(\rho - x^2 - y^2) + x \left( \eta - \frac{y}{\sqrt{x^2 + y^2}} \right), \quad (3.35)$$

where  $\rho > 0$ ; this example was considered using different methods in [55]. This can be written in polar coordinates as:

$$\dot{r} = \rho r - r^3, \quad (3.36)$$

$$\dot{\phi} = \eta - \sin \phi. \quad (3.37)$$

Unlike [18], here we use a two-dimensional model, as a minimum of two dimensions is necessary for applying the augmented phase reduction; for the standard phase reduction, a one dimensional model would suffice. As the bifurcation parameter  $\eta$  varies, two fixed points annihilate at  $\eta = 1$ ,  $\phi = \pi/2$ , and  $r = \sqrt{\rho}$ , giving rise to a periodic orbit for  $\eta > 1$ . The stable manifold of the fixed point at the bifurcation is along the radial direction, and the unstable manifold is along the azimuthal direction of the periodic orbit.

The periodic orbit is stable with radius  $r_{po} = \sqrt{\rho}$ , and time period  $T = 2\pi/\sqrt{\eta^2 - 1}$ . We can find the phase variable  $\theta = f(\phi)$  by setting its time derivative equal to the angular frequency. This gives

$$\theta = 2 \arctan \left( \frac{\eta \tan \left( \frac{\phi}{2} \right) - 1}{\sqrt{\eta^2 - 1}} \right) + \pi. \quad (3.38)$$

Thus as  $\phi$  varies from  $-\pi$  to  $\pi$ ,  $\theta$  advances linearly in time from 0 to  $2\pi$ . The bifurcation occurs at  $\phi = \pi/2$ , which corresponds to  $\theta = \pi$ . The periodic trajectory spends most of its time near  $\phi = \pi/2$  near the bifurcation, as shown in Figure 3.8. Thus we expect the PRC to be large near  $\phi = \pi/2$  i.e., ( $\theta = \pi$ ), and small elsewhere. Differentiating

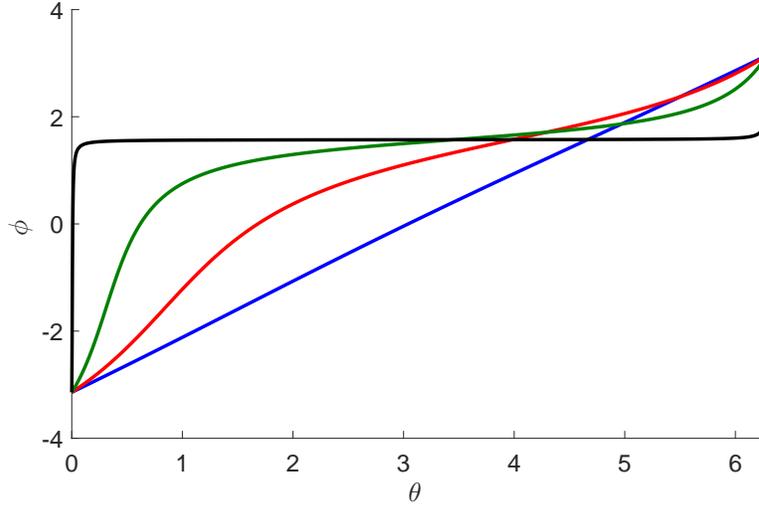


Figure 3.8:  $\phi$  evolution for SNIPER bifurcation model: The figure plots the evolution of azimuthal angle  $\phi$  as a function of the phase  $\theta$  of the periodic orbit, which advances linearly in time. The blue, red, green and black lines corresponds to  $\eta = 20$ , 1.5, 1.05, and 1.00001 respectively.

equation (3.38) with respect to  $r$  and  $\phi$ , we get the PRC as

$$\left( \frac{\partial \theta}{\partial r}, \frac{\partial \theta}{\partial \phi} \right) = \left( 0, \frac{\sqrt{\eta^2 - 1}}{\eta - \sin \phi} \right), \quad (3.39)$$

which simplifies to

$$\left( \frac{\partial \theta}{\partial r}, \frac{\partial \theta}{\partial \phi} \right) = \left( 0, \frac{\eta^2 - \cos \theta - \sqrt{\eta^2 - 1} \sin \theta}{\eta \sqrt{\eta^2 - 1}} \right). \quad (3.40)$$

It is clear from (3.39) that the PRC is always positive, and it blows up to infinity at  $\phi = \pi/2$  (i.e.,  $\theta = \pi$ ) at the bifurcation. This is evident from the Figure 3.9. Note: for  $\eta \gtrsim 1$ , the expression in (3.40) reduces to  $\frac{\partial \theta}{\partial \phi} = \frac{1 - \cos \theta}{\omega}$ , which is consistent with the analysis given in [40, 18]. Transforming to Cartesian coordinates  $(x, y) = (r \cos \phi, r \sin \phi)$ ,

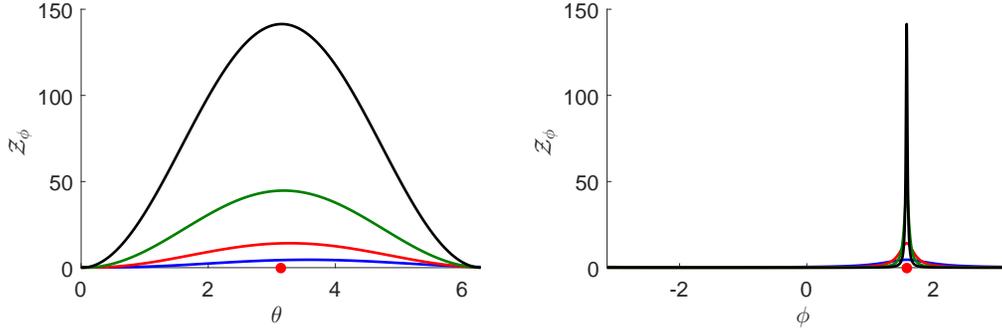


Figure 3.9: PRC for SNIPER bifurcation model: The left (resp., the right) column plots the PRC  $\mathcal{Z}_\phi$  versus  $\theta$  (resp.,  $\phi$ ). In both plots, the blue, red, green and black lines corresponds to  $\eta = 1.1, 1.01, 1.001,$  and  $1.0001$  respectively. The red dots mark the bifurcation point  $\phi = \pi/2 / \theta = \pi$ .

we can write the PRC as

$$\mathcal{Z}(\theta) = \frac{\cos \theta + \sqrt{\eta^2 - 1} \sin \theta - 1}{\sqrt{\rho} \sqrt{\eta^2 - 1}} \hat{x} + \frac{\sin \theta - \sqrt{\eta^2 - 1} \cos \theta}{\sqrt{\rho} \eta} \hat{y}. \quad (3.41)$$

Thus the isochrons are radial lines, with eigenvector  $v = \hat{x} + 0\hat{y}$  at point  $(x, y) = (-r_{po}, 0) \equiv (x_0, y_0)$ . From the mean of the divergence of the vector field along the periodic orbit, we get

$$k = \frac{\int_0^T (2\rho - 4r^2) dt}{T} - \frac{\int_0^T \cos \phi dt}{rT}, \quad (3.42)$$

$$\Rightarrow k = -2\rho - \frac{\int_0^{2\pi} \frac{\sqrt{\eta^2 - 1} \sin \theta - (\eta^2 - 1) \cos \theta}{\sqrt{\eta^2 - 1} \sin \theta + \cos \theta - \eta^2} d\theta}{2\pi r}. \quad (3.43)$$

Since the integral in (3.43) is calculated to be zero, we get  $k = -2\rho$ . Thus the adjoint equation for the IRC becomes:

$$\dot{\mathcal{I}}_r = 0, \quad (3.44)$$

$$\dot{\mathcal{I}}_\phi = (\cos \phi - 2\rho) \mathcal{I}_{\phi_0}. \quad (3.45)$$

Since the IRC is  $T$ -periodic,  $\mathcal{I}_{\phi_0} = 0$ . Thus we get the IRC in polar and Cartesian coordinates:

$$\mathcal{I}_{r,\phi} = \mathcal{I}_{r_0} \hat{r} + 0 \hat{\phi}, \quad (3.46)$$

$$\mathcal{I}_{x,y} = \mathcal{I}_{r_0} \cos \phi \hat{x} + \mathcal{I}_{r_0} \sin \phi \hat{y}. \quad (3.47)$$

To find the constant  $\mathcal{I}_{r_0}$ , we use the normalization condition at point  $(x_0, y_0)$

$$\mathcal{I}_{x_0,y_0} \cdot v = 1 \quad \Rightarrow \mathcal{I}_{r_0} = -1. \quad (3.48)$$

This gives the IRC in polar and Cartesian coordinates as

$$\mathcal{I}_{r,\phi} = -\hat{r} + 0\hat{\phi}, \quad (3.49)$$

$$\mathcal{I}_{x,y} = -\cos \phi \hat{x} - \sin \phi \hat{y}. \quad (3.50)$$

At first glance, it seems that the IRC is sinusoidal. It is, but only far away from the bifurcation point. As we approach the bifurcation,  $\phi$  no longer varies linearly with phase (see Figure 3.8). The “sinusoidal” IRC gets expanded near the bifurcation point, and squeezed away from the bifurcation point. This is seen in Figure 3.10, which plots the IRC as the bifurcation parameter  $\eta$  varies. We see that near the bifurcation point, the IRC stays close to zero in the  $x$  direction, and close to  $-1$  in the  $y$  direction. This observation agrees with the intuitive definition of the IRC. Near the bifurcation point, the periodic trajectory points in the  $x$  direction, so the IRC, which is the gradient of  $\psi$  coordinate, is zero along that direction. On the other hand, the  $y$  coordinate is anti-parallel to the isochron, along which the gradient of  $\psi$  is unity. We can write the IRC as

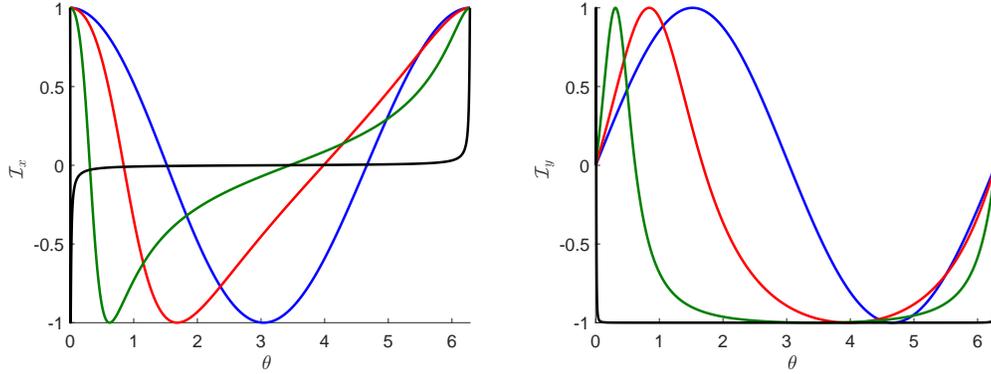


Figure 3.10: IRC for SNIPER bifurcation model: The left (resp., the right) column plots the IRC  $\mathcal{I}_x$  (resp.,  $\mathcal{I}_y$ ) versus  $\theta$ . In both plots, the blue, red, green and black lines corresponds to  $\eta = 20, 1.5, 1.05$ , and  $1.00001$  respectively.

a function of phase as

$$\mathcal{I}_{x,y} = \frac{\sqrt{\eta^2 - 1} \sin \theta - (\eta^2 - 1) \cos \theta}{\sqrt{\eta^2 - 1} \sin \theta + \cos \theta - \eta^2} \hat{x} + \frac{\eta \left( 1 - \sqrt{\eta^2 - 1} \sin \theta - \cos \theta \right)}{\sqrt{\eta^2 - 1} \sin \theta + \cos \theta - \eta^2} \hat{y}. \quad (3.51)$$

### Morris-Lecar Model

The Morris-Lecar model [62], a two-dimensional excitation model, is given as

$$C_M \dot{v} = I_b - g_L(v - E_L) - g_K n(v - E_K) - g_{Ca} m_\infty(v)(v - E_{Ca}), \quad (3.52)$$

$$\dot{n} = \phi(n_\infty(v) - n)/\tau_n(v), \quad (3.53)$$

$$m_\infty(v) = 0.5 \left( 1 + \tanh \left( \frac{v - v_1}{v_2} \right) \right), \quad (3.54)$$

$$\tau_n(v) = \frac{1}{\left( \cosh \left( \frac{v - v_3}{2v_4} \right) \right)}, \quad (3.55)$$

$$n_\infty(v) = 0.5 \left( 1 + \tanh \left( \frac{v - v_3}{v_4} \right) \right). \quad (3.56)$$

Parameters  $\phi$ ,  $g_{Ca}$ ,  $v_3$ ,  $v_4$ ,  $E_{Ca}$ ,  $E_K$ ,  $E_L$ ,  $g_K$ ,  $g_L$ ,  $v_1$ ,  $v_2$ , and  $C_M$  are taken from the column ‘‘SNLC’’ of Table 3.1 in [57]. For  $I_b = 39.9957 \text{ mA}$ , the system has a stable

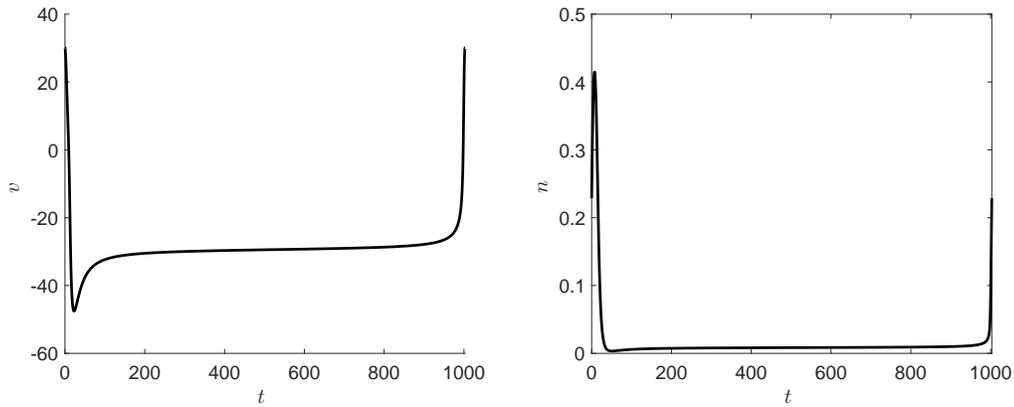


Figure 3.11: Morris-Lecar Model: Time series for the periodic orbit near the SNIPER bifurcation. Here  $I_b = 39.9957 \text{ mA}$ .

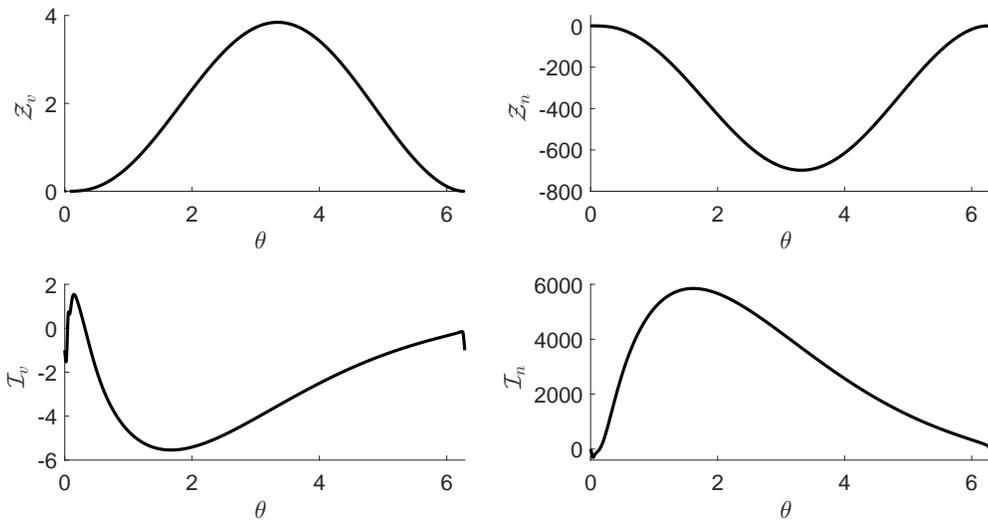


Figure 3.12: Morris-Lecar Model: Top (resp., bottom) row plots the PRC (resp., IRC) near the SNIPER bifurcation. Here  $I_b = 39.9957 \text{ mA}$ .

periodic orbit near a SNIPER bifurcation with time period  $T = 1002.88 \text{ ms}$ , nontrivial Floquet multiplier  $\lambda = 3.632 \times 10^{-45}$ , with corresponding nontrivial Floquet exponent  $k = -0.1020$ . The time series for one period is shown in Figure 3.11. Figure 3.12 plots the PRC and IRC for the Morris-Lecar oscillator. The PRC is sinusoidal and does not change sign, just like the simple model (see Figure 3.9 for comparison). The IRC looks

like a sinusoid skewed to one side, similar to the IRC calculated for the simple model (see Figure 3.10 for comparison).

### 3.2.5 Homoclinic bifurcation

For a homoclinic bifurcation [29, 54], a periodic orbit is born out of a homoclinic orbit to a hyperbolic saddle point  $p$  upon varying a parameter  $\mu$ . If a homoclinic orbit exists for  $\mu = 0$ , then there will be a periodic orbit for, say,  $\mu > 0$ , but not for  $\mu < 0$ , as shown in Figure 3.1(d). We assume that the magnitude of the unstable eigenvalue  $\lambda_u$  of the saddle point is smaller than the stable eigenvalue  $\lambda_s$ , resulting in a stable periodic orbit [29]. For  $\mu$  close to zero, the periodic solution spends most of its time near the saddle point  $p$ , where the vector field can be approximated by its linearization. It can be written in diagonal form as

$$\dot{x} = \lambda_u x, \quad (3.57)$$

$$\dot{y} = \lambda_s y, \quad (3.58)$$

where  $\lambda_u > 0$ , and  $\lambda_s < 0$ . As in [18], we consider a box  $B = [0, \Delta] \times [0, \Delta] \equiv \Sigma_0 \times \Sigma_1$  that encloses the periodic orbit for most of its time period, and within which equations (3.57, 3.58) are accurate. This is shown in the left panel of Figure 3.13. We do not model the periodic orbit outside  $B$ , but assume that trajectory re-enters the box after a time  $\delta T$  at a distance  $\epsilon$  from the  $y$  axis, where  $\epsilon$  varies with the bifurcation parameter  $\mu$ . The time taken for the trajectory to traverse  $B$  can be found as [18]

$$\tau(\epsilon) = \frac{1}{\lambda_u} \log \left( \frac{\Delta}{\epsilon} \right). \quad (3.59)$$

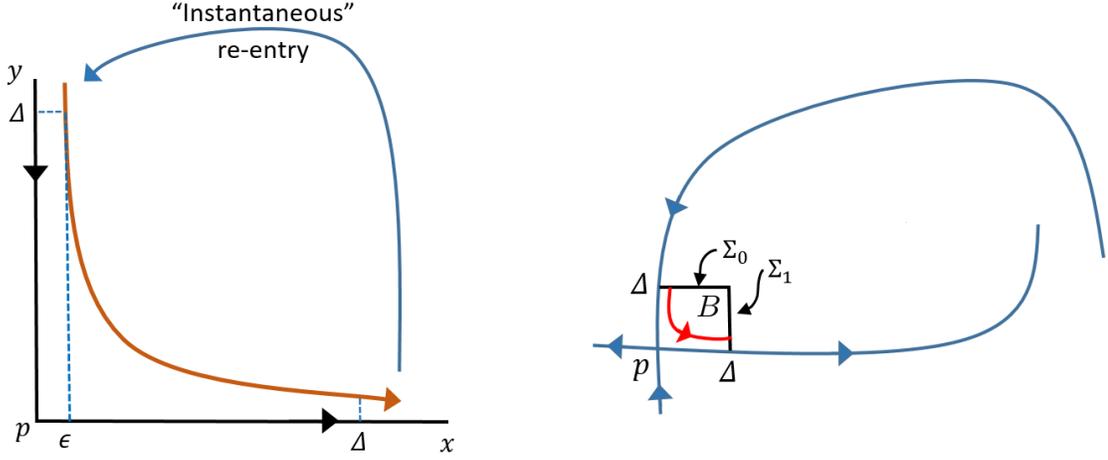


Figure 3.13: Periodic trajectory near a homoclinic bifurcation. The left panel shows the periodic trajectory near the saddle point. The right panel shows the Poincaré sections used in the analysis

Thus the time period of periodic orbit is given as  $\tau(\epsilon) + \delta T$ . As  $\mu$  decreases towards zero, the periodic orbit approaches  $p$ , resulting in  $\epsilon$  approaching 0. Near the bifurcation,  $\delta T \ll \tau(\epsilon)$ , so  $T \approx \tau(\epsilon)$ . Thus we approximate the trajectory as spending all its time within the box  $B$ , and re-injecting into the box instantaneously. Thus we set  $\theta = 0$  at the point where trajectory enters  $B$ , and  $\theta = 2\pi$  where trajectory exits  $B$ . To find the PRC, we solve the adjoint equation (2.7) in  $B$  to get

$$\mathcal{Z}(\theta) = \mathcal{Z}_{x_0} e^{-\lambda_u t} \hat{x} + \mathcal{Z}_{y_0} e^{-\lambda_s t} \hat{y}, \quad (3.60)$$

subject to the initial condition (equation (2.8))

$$\mathcal{Z}_{x_0} \lambda_u \epsilon + \mathcal{Z}_{y_0} \lambda_s \Delta = \frac{2\pi \lambda_u}{\log\left(\frac{\Delta}{\epsilon}\right)}. \quad (3.61)$$

As  $\mu \rightarrow 0$ ,  $\epsilon \rightarrow 0$ , thus the first term in the left hand side and the right hand side term in above equation go to zero. Thus we get  $\mathcal{Z}_{y_0} \approx 0$  near the bifurcation point, and the

PRC is only significant in the  $x$ -direction. Since the isochrons are orthogonal to the PRC on the limit cycle, the eigenvector  $v \approx 0 \hat{x} + 1 \hat{y}$ . We will use this information for the normalization condition of the IRC later. Since the trajectory spends most of its time inside the box  $B$ , we get  $k = \lambda_s + \lambda_u$  by the mean of the divergence of the linear vector field inside  $B$ . We will also prove this by the Poincaré analysis below.

Consider the Poincaré maps

$$P = P_2 \circ P_1 : \Sigma_0 \rightarrow \Sigma_0, \quad \text{where} \quad (3.62)$$

$$P_1 : \Sigma_0 \rightarrow \Sigma_1; \quad (x, \Delta) \rightarrow (\Delta, \Delta e^{\lambda_s T}), \quad (3.63)$$

$$P_2 : \Sigma_1 \rightarrow \Sigma_0; \quad (\Delta, y) \rightarrow (x, \Delta). \quad (3.64)$$

The Poincaré sections  $\Sigma_0$  and  $\Sigma_1$  are shown in the right panel of Figure 3.13. Following the analysis in Chapter 10 in [63], we get the Poincaré map  $P$  as

$$P : \Sigma_0 \rightarrow \Sigma_0, \quad (x, \Delta) \rightarrow (Ax^{-\frac{\lambda_s}{\lambda_u}} + \mu, \Delta), \quad (3.65)$$

where  $A$  is a positive constant, and  $\mu$  is the bifurcation parameter. This gives the nontrivial Floquet multiplier of the periodic orbit as

$$\lambda = A' \epsilon^{-\frac{\lambda_s}{\lambda_u} - 1}, \quad (3.66)$$

where  $A' = -A\lambda_s/\lambda_u$ . From this equation, it is easy to see that  $\lambda \rightarrow 0$  as  $\epsilon \rightarrow 0$ . Note that although the isochrons in the box  $B$  may not be horizontal, we have calculated the nontrivial Floquet multiplier for a horizontal section, as that is more convenient; the value of the nontrivial Floquet multiplier is independent of the Poincaré section [63].  $k$

can be found as

$$k = \frac{\log\left(A'\epsilon^{-\frac{\lambda_s}{\lambda_u}-1}\right)}{T}. \quad (3.67)$$

Near the bifurcation, this can be written as

$$k = \lim_{\epsilon \rightarrow 0} \frac{\log\left(A'\epsilon^{-\frac{\lambda_s}{\lambda_u}-1}\right)}{\frac{1}{\lambda_u} \log\left(\frac{\Delta}{\epsilon}\right)}. \quad (3.68)$$

Since both the numerator and denominator approach plus or minus infinity as  $\epsilon \rightarrow 0$ , the limit can be solved by L'Hospital's rule as

$$k = \lim_{\epsilon \rightarrow 0} \left( \frac{\lambda_u \Delta \epsilon^{-1}}{A' \epsilon^{-\frac{\lambda_s}{\lambda_u}-1}} \right) \left( \frac{A' \left( \frac{\lambda_s}{\lambda_u} + 1 \right) \epsilon^{-\frac{\lambda_s}{\lambda_u}-2}}{\Delta \epsilon^{-2}} \right) = \lambda_s + \lambda_u. \quad (3.69)$$

With this, we get the following adjoint equation for the IRC:

$$\dot{\mathcal{I}}_x = \lambda_s \mathcal{I}_x, \quad (3.70)$$

$$\dot{\mathcal{I}}_y = \lambda_u \mathcal{I}_y, \quad (3.71)$$

$$\Rightarrow \mathcal{I}_x = \mathcal{I}_{x_0} e^{\lambda_s t}, \quad (3.72)$$

$$\mathcal{I}_y = \mathcal{I}_{y_0} e^{\lambda_u t}. \quad (3.73)$$

The normalization condition  $\mathcal{I}_{x_0, y_0} \cdot v = 1$  gives the IRC as

$$\mathcal{I}_{x,y} = \mathcal{I}_{x_0} e^{\frac{\lambda_s \theta}{\omega}} \hat{x} + e^{\frac{\lambda_u \theta}{\omega}} \hat{y}. \quad (3.74)$$

Here  $\mathcal{I}_{x_0}$  remains indeterminate as we do not model the dynamics outside  $B$ . The  $x$  component of the IRC decreases at an exponential rate, while the  $y$  component increases at an exponential rate inside the box  $B$ . We do not implement the condition of  $T$ -

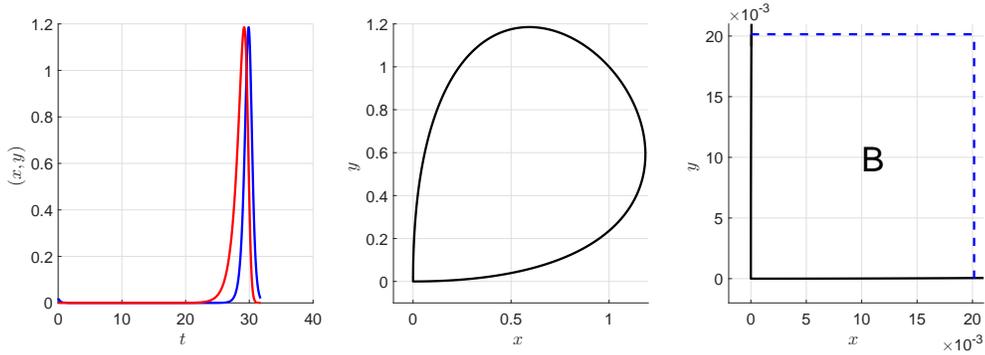


Figure 3.14: Periodic orbit near homoclinic bifurcation with parameters  $\mu = 1 \times 10^{-13}$ ,  $a = -1$ , and  $b = 2$ . The left (resp., middle) panel shows the time series (resp., orbit). The blue and the red lines show the  $x$  and  $y$  component of trajectories respectively. The right panel shows the box  $B$ .

periodicity on equations (3.72, 3.73), as the calculated expressions of the IRC are valid only in the box  $B$ . We expect the IRC to jump back to its initial value as the trajectory re-enters the box.

### A simple model for homoclinic bifurcation

We use a 2-dimensional model derived from [64] to validate our result:

$$\dot{x} = (a + b - 0.5\mu)x - 0.5\mu y - (a/4 + 3b/8)(x + y)^2 - 3a/8(x^2 - y^2), \quad (3.75)$$

$$\dot{y} = 0.5\mu x + (a - b + 0.5\mu)y + (-a/4 + 3b/8)(x + y)^2 + 3a/8(x^2 - y^2). \quad (3.76)$$

This system undergoes a homoclinic bifurcation at  $\mu = 0$ , and has a stable periodic orbit for  $\mu > 0$ ,  $a < 0 < b$ , and  $|b| > |a|$ . With parameters  $\mu = 1 \times 10^{-13}$ ,  $a = -1$ , and  $b = 2$ , we get a stable periodic orbit with the period  $T = 31.7689$ , eigenvalues  $\lambda_s = -3$ ,  $\lambda_u = 1$ , nontrivial Floquet exponent  $k = -1.7579$ , and the eigenvector  $v = 0.0006\hat{x} + 0.9999\hat{y}$ . The periodic trajectory, orbit, and the box  $B$  are shown in Figure 3.14. With  $\Delta = 0.0201$ , the trajectory spends 86.5 % of its period in the box  $B$ . Figure 3.15 compares the

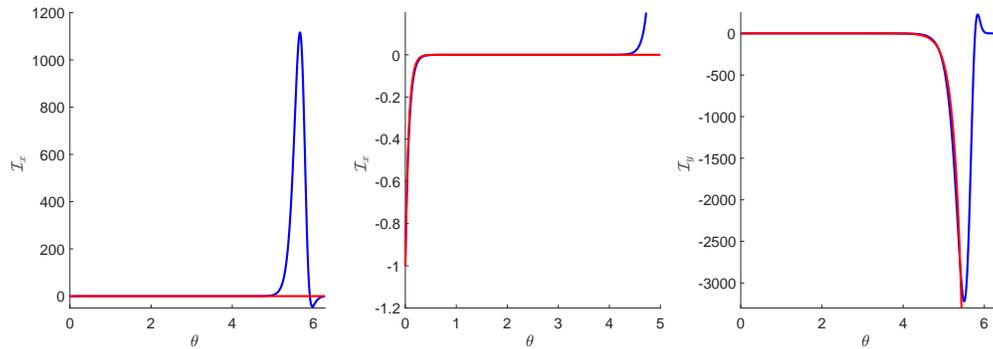


Figure 3.15: IRC for periodic orbit near a homoclinic bifurcation. The left and the right panels shows the  $x$  and  $y$  component of the IRC respectively, the middle panel shows the zoomed in plot of the left panel. The blue line shows the numerically computed IRC, while the red line shows an exponential curve with rate constant given by (3.74).

numerically computed IRC with the exponential curve having rate constants from the analytical IRC (3.74). We see that the numerically computed IRC agrees well with the analytical one in the beginning (inside box  $B$ ), but diverges after. It oscillates quickly back to its initial value at the end of its period, as is expected.

### 3.2.6 Relaxation oscillator

In a relaxation oscillator, at least one variable evolves at a much faster rate than the other variables. Such oscillators are ubiquitous in conductance-based models of cells, where the gating variables evolve at a much slower rate than the cell membrane potential. A two-dimensional relaxation oscillator can be written as

$$\mu \dot{x} = f(x, y), \quad 0 < \mu \ll 1, \quad (3.77)$$

$$\dot{y} = g(x, y). \quad (3.78)$$

In the relaxation limit ( $\mu \rightarrow 0$ ), the PRC is given as [51]

$$\mathcal{Z}(\theta) = -\frac{\omega g_x}{f_x g} \hat{x} + \frac{\omega}{g} \hat{y}. \quad (3.79)$$

Here the functions  $g$ ,  $g_x$ , and  $f_x$  are evaluated on the periodic orbit, and thus are functions of  $\theta$ . The eigenvector  $v$  in the direction of isochron is given as

$$v = \frac{-\hat{x} - \frac{g_x}{f_x} \hat{y}}{\sqrt{1 + \frac{g_x^2}{f_x^2}}}. \quad (3.80)$$

For computing the adjoint equation for IRC in relaxation limit, we do the following analysis in the spirit of [18].

Consider an infinitesimal perturbation  $\Delta \mathbf{x} = (\Delta x, \Delta y)$  to the periodic trajectory  $\mathbf{x} \in \mathbf{x}^\gamma(t)$ . Then the perturbed trajectory evolves as

$$\mu \dot{\Delta x} = f_x \Delta x + f_y \Delta y, \quad (3.81)$$

$$\dot{\Delta y} = g_x \Delta x + g_y \Delta y. \quad (3.82)$$

This can be written as  $A \dot{\Delta \mathbf{x}} = DF \Delta \mathbf{x}$ , where  $A = \begin{bmatrix} \mu & 0 \\ 0 & 1 \end{bmatrix}$ , and  $DF$  is the Jacobian evaluated on the periodic orbit. The isostable shift  $\Delta \psi$  by a perturbation  $A \Delta \mathbf{x}$  is given by  $\Delta \psi = \langle \nabla \psi, A \Delta \mathbf{x} \rangle$ , where  $\langle \cdot, \cdot \rangle$  is the Euclidean inner product. Its time evolution can be written as

$$\dot{\Delta \psi} = \langle \nabla \dot{\psi}, A \Delta \mathbf{x} \rangle + \langle \nabla \psi, A \dot{\Delta \mathbf{x}} \rangle = k \Delta \psi, \quad (3.83)$$

$$\Rightarrow \langle A^T \nabla \dot{\psi}, \Delta \mathbf{x} \rangle = \langle k A^T \nabla \psi, \Delta \mathbf{x} \rangle - \langle \nabla \psi, DF \Delta \mathbf{x} \rangle. \quad (3.84)$$

This can be written as

$$\mu \dot{\mathcal{I}}_x = (k\mu - f_x)\mathcal{I}_x - g_x\mathcal{I}_y, \quad (3.85)$$

$$\dot{\mathcal{I}}_y = -f_y\mathcal{I}_x + (k - g_y)\mathcal{I}_y, \quad (3.86)$$

where  $\mathcal{I}_x = \partial\psi/\partial x$ , and  $\mathcal{I}_y = \partial\psi/\partial y$ . From the mean of the divergence of the vector field along periodic trajectory, we get the nontrivial Floquet exponent and multiplier as

$$\lambda = \exp\left(\int_0^T (f_x/\mu + g_y) dt\right), \quad (3.87)$$

$$k = a/\mu + b, \quad (3.88)$$

where  $a = \frac{\int_0^T f_x dt}{T}$ , and  $b = \frac{\int_0^T g_y dt}{T}$ . We must have  $k < 0$  for a stable periodic orbit. This implies that  $a < 0$ , because otherwise,  $k$  would get positive as  $\mu \rightarrow 0$ . Thus in the relaxation limit,  $k \rightarrow -\infty$  and  $\lambda \rightarrow 0$ . Thus any perturbation from the periodic orbit gets nullified instantly by the vector field. The adjoint equation for the IRC becomes

$$\mu \dot{\mathcal{I}}_x = (a + \mu b - f_x)\mathcal{I}_x - g_x\mathcal{I}_y, \quad (3.89)$$

$$\dot{\mathcal{I}}_y = -f_y\mathcal{I}_x + (a/\mu + b - g_y)\mathcal{I}_y. \quad (3.90)$$

$$\Rightarrow \mathcal{I}_x = \frac{g_x}{a + \mu b - f_x}\mathcal{I}_y + \mathcal{O}(\mu), \quad (3.91)$$

$$\Rightarrow \mu \dot{\mathcal{I}}_y = \left(a + \mu b - \mu g_y - \frac{\mu g_x f_y}{a + \mu b - f_x}\right)\mathcal{I}_y + \mathcal{O}(\mu^2). \quad (3.92)$$

In the relaxation limit ( $\mu \rightarrow 0$ ), we get

$$(a - f_x)\mathcal{I}_y = 0. \quad (3.93)$$

We know from the mean value theorem that there is at least one phase  $\theta_i$  where  $a = f_x$ . Thus the coefficient of  $\mathcal{I}_y$  in (3.93) is nonzero except at  $\theta_i$ . Thus in order to satisfy the equation (3.93),  $\mathcal{I}_y$  has to be zero for all  $\theta$  except at  $\theta_i$  where it can be non-zero. The same can be said about  $\mathcal{I}_x$  from equation (3.91). Thus we can write the the IRC as

$$\mathcal{I}_{x,y} = \left( \sum_i \mathcal{I}_x(\theta_i) \right) \hat{x} + \left( \sum_i \mathcal{I}_y(\theta_i) \right) \hat{y}. \quad (3.94)$$

It makes sense intuitively that the IRC is zero everywhere except at few points because the periodic orbit is very strongly stable in the relaxation limit (the nontrivial Floquet multiplier is close to zero). Therefore, a perturbation from the periodic orbit gets nullified instantaneously by the stabilizing vector field. This renders the isostable coordinate zero near the periodic orbit, and its gradient zero almost everywhere on the periodic orbit.

### van der Pol oscillator

An example of a relaxation oscillator is the van der Pol oscillator [65, 66] which can be written as

$$\mu \dot{x} = -y + x - x^3/3, \quad 0 < \mu \ll 1, \quad (3.95)$$

$$\dot{y} = x. \quad (3.96)$$

In the relaxation limit ( $\mu \rightarrow 0$ ), we find numerically that  $a - f_x$  crosses zero at  $\theta_1 = 1.6567$  and  $\theta_2 = 4.7983$ . Thus we expect the IRC to be zero everywhere except these two  $\theta_i$ 's. We compute periodic orbits and their IRCs for three different values of the parameter  $\mu$  : 0.1, 0.01, and 0.001, as shown in Figure 3.16. We see from Figure 3.16 that as  $\mu$  approaches the relaxation limit, IRC becomes zero everywhere except near the phases  $\theta_1$ , and  $\theta_2$ , thus validating our analytical results. Since the IRC is zero everywhere except

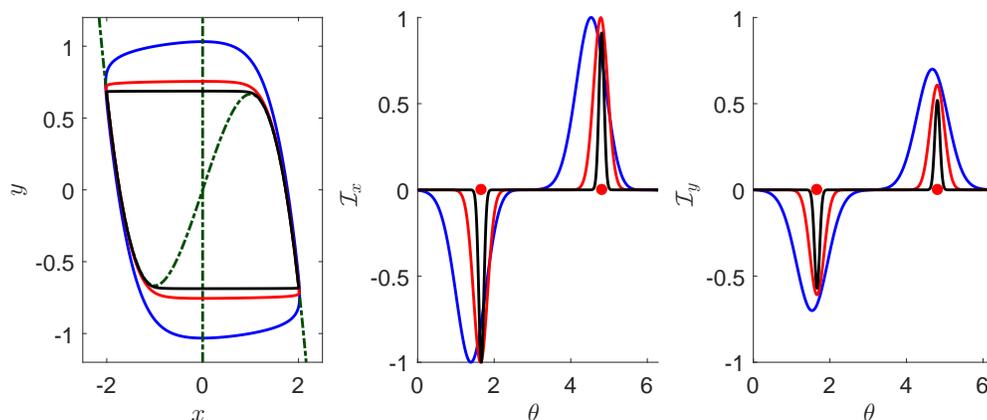


Figure 3.16: van der Pol Oscillator: The left panel plots the periodic orbits and nullclines. The middle (resp., the right) panel plots  $\mathcal{I}_x$  (resp.,  $\mathcal{I}_y$ ). In all plots, the blue, red and black lines correspond to  $\mu = 0.1, 0.01,$  and  $0.001$ , respectively. The two red dots in the middle and right panels mark the phases  $\theta_1$  and  $\theta_2$ .

near 2 points, we do not use the normalization condition of Section 2.3; instead we normalize the IRC by the maximum absolute value of  $\{\mathcal{I}_x(\theta_i), \mathcal{I}_y(\theta_i)\}$ .

### 3.3 Discussion and Conclusions

In this chapter, we have derived expressions for the augmented phase reduction for six distinct systems with a periodic orbit. We found that the  $\lambda - \omega$ , Hopf, and Bautin normal form systems have sinusoidal PRCs and IRCs. For a model near a SNIPER bifurcation, the PRC never changes sign, while the IRC looks like a skewed sinusoid. For a system near homoclinic bifurcation, the IRC is exponential for a large part of its phase. Finally for a relaxation oscillator, the IRC is zero everywhere except at a few points.

We simulated dynamic models which are examples of these six systems, and found that their numerically computed IRCs matches with their analytical counterparts very closely except in the system undergoing a saddle-node bifurcation of periodic orbits, for which we have used the Bautin normal form. This normal form captures a saddle-

node bifurcation of periodic orbits, where an unstable periodic orbit branch born out of a subcritical Hopf bifurcation turns around and gains stability; however, it does not capture the relaxation nature of dynamics present in some models. That is why the IRC computed numerically for such models does not match closely in shape with the derived sinusoidal IRC.

For a strongly stable system, the nontrivial Floquet exponent  $k$  goes to  $-\infty$ . This is the case for relaxation oscillator in the relaxation limit. Thus, any perturbation to the periodic orbit gets nullified instantly. In such a case, it is not necessary to use the augmented phase reduction, instead the standard phase reduction would suffice. On the other hand, for the other five systems, it is better to use the augmented phase reduction over the standard phase reduction, especially when  $k$  is a negative number that is small in magnitude.

Table 3.1 summarizes the analytical expressions for augmented phase reduction derived in Section 3.2.

Table 3.1: Summary of analytical expressions of augmented phase reduction for six dynamically distinct systems

Dynamic Model	PRC	IRC	$k$
$\lambda = \omega$ $\dot{r} = G(r),$ $\dot{\phi} = H(r).$	$\left(-\frac{H'(r_{po})}{G'(r_{po})} \cos \theta - \frac{\sin \theta}{r_{po}}\right) \hat{x}$ $+ \left(-\frac{H'(r_{po})}{G'(r_{po})} \sin \theta + \frac{\cos \theta}{r_{po}}\right) \hat{y}$	$-\sqrt{1 + \frac{r_{po}^2 H'(r_{po})^2}{G'(r_{po})^2}} \cos \theta \hat{x}$ $-\sqrt{1 + \frac{r_{po}^2 H'(r_{po})^2}{G'(r_{po})^2}} \sin \theta \hat{y}$	$G'(r_{po})$
Hopf $\dot{r} = ar + cr^3,$ $\dot{\phi} = b + dr^2.$	$\left(\frac{d}{\sqrt{-ac}} \cos \theta + \frac{c}{\sqrt{-ac}} \sin \theta\right) \hat{x}$ $+ \left(\frac{d}{\sqrt{-ac}} \sin \theta - \frac{c}{\sqrt{-ac}} \cos \theta\right) \hat{y}$	$-\sqrt{1 + \frac{d^2}{c^2}} \cos \theta \hat{x}$ $-\sqrt{1 + \frac{d^2}{c^2}} \sin \theta \hat{y}$	$-2a$
Bautin $\dot{r} = ar + cr^3 + fr^5,$ $\dot{\phi} = b + dr^2 + gr^4.$	$\left(-\frac{2dr_{po} + 4gr_{po}^3}{a + 3cr_{po}^2 + 5fr_{po}^4} \cos \theta - \frac{\sin \theta}{r_{po}}\right) \hat{x}$ $+ \left(-\frac{2dr_{po} + 4gr_{po}^3}{a + 3cr_{po}^2 + 5fr_{po}^4} \sin \theta + \frac{\cos \theta}{r_{po}}\right) \hat{y}$	$-\sqrt{1 + r_{po}^2 \left(\frac{2dr_{po} + 4gr_{po}^3}{a + 3cr_{po}^2 + 5fr_{po}^4}\right)^2} \cos \theta \hat{x}$ $-\sqrt{1 + r_{po}^2 \left(\frac{2dr_{po} + 4gr_{po}^3}{a + 3cr_{po}^2 + 5fr_{po}^4}\right)^2} \sin \theta \hat{y}$	$a + 3cr_{po}^2$ $+ 5fr_{po}^4$
SNIPER $\dot{r} = \rho r - r^3,$ $\dot{\phi} = \eta - \sin \phi.$	$\frac{\cos \theta + \sqrt{\eta^2 - 1} \sin \theta - 1}{\sqrt{\rho} \sqrt{\eta^2 - 1}} \hat{x}$ $+ \frac{\sin \theta - \sqrt{\eta^2 - 1} \cos \theta}{\sqrt{\rho} \eta} \hat{y}$	$\frac{\sqrt{\eta^2 - 1} \sin \theta - (\eta^2 - 1) \cos \theta}{\sqrt{\eta^2 - 1} \sin \theta + \cos \theta - \eta^2} \hat{x}$ $+ \frac{\eta (1 - \sqrt{\eta^2 - 1} \sin \theta - \cos \theta)}{\sqrt{\eta^2 - 1} \sin \theta + \cos \theta - \eta^2} \hat{y}$	$-2\rho$
Homoclinic $\dot{x} = \lambda_u x,$ $\dot{y} = \lambda_s y.$	$\mathcal{Z}_{x_0} e^{-\frac{\lambda_u \theta}{\omega}} \hat{x} + \mathcal{Z}_{y_0} e^{-\frac{\lambda_s \theta}{\omega}} \hat{y}$	$\mathcal{I}_{x_0} e^{\frac{\lambda_s \theta}{\omega}} \hat{x} + e^{\frac{\lambda_u \theta}{\omega}} \hat{y}$	$\lambda_s + \lambda_u$
Relaxation $\mu \dot{x} = f(x, y),$ $\dot{y} = g(x, y).$	$-\frac{\omega g_x}{f_x g} \hat{x} + \frac{\omega}{g} \hat{y}$	$\left(\sum_i \mathcal{I}_x(\theta_i)\right) \hat{x} + \left(\sum_i \mathcal{I}_y(\theta_i)\right) \hat{y}$	$-\infty$

# Chapter 4

## Optimal Phase Control using Augmented Phase Reduction

### 4.1 Introduction

As previously discussed, standard phase reduction is valid only in close proximity to the periodic orbit. Consequently, the magnitude of the allowable control stimulus is limited by the nontrivial Floquet multipliers [29] of the periodic orbit: in systems with a nontrivial Floquet multiplier close to 1, even a relatively small stimulus can drive the trajectory away from the periodic orbit, rendering the phase reduction inaccurate and control based on phase reduction ineffective. In most practical applications, the effectiveness of a control algorithm depends on the size of the allowable stimulus [21, 27, 22]. This suggests that control algorithms based on the augmented phase reduction will be more effective than those based on just the phase coordinate, as they can be designed to allow a larger stimulus without the risk of driving the oscillator away from the periodic orbit [30].

In this chapter, we develop a novel optimal control algorithm based on augmented

phase reduction to advance (resp., delay) the phase of the oscillator, such that the oscillator completes one periodic trajectory sooner (resp., slower). Along with minimizing the total energy consumption, our control algorithm also minimizes a measure of the transversal distance of the oscillator from the unperturbed periodic orbit. This novel aspect of our control algorithm is crucial in ensuring that the controlled oscillator always stays close to the unperturbed periodic orbit, where phase reduction is valid, thus making our control algorithm effective. Note that this way of incorporating closeness of the controlled trajectory to the periodic trajectory in the cost function is possible due to the explicit formulation of transversal dynamics in terms of Floquet multipliers in the augmented phase reduction. This allows us to efficiently keep the perturbed trajectory close to the periodic orbit along weakly stable isostable directions, even in the presence of noise.

Moreover, we develop a novel strategy to eliminate cardiac alternans by connecting our control algorithm with the underlying physiological problem to change the phase of cardiac pacemaker cells. This strategy removes the need to excite the myocardium tissue at multiple sites. We also show how our control algorithm can be used to change the spike timing of neurons, which could be relevant to the problem of desynchronizing neurons for the treatment of essential and parkinsonian tremor [67, 68]. Such an optimal control is expected to consume less energy than the pulsatile current in the present Deep Brain Stimulation (DBS) protocol, thus possibly prolonging the battery life of the stimulator, and also preventing tissue damage caused by the high energy DBS stimuli. Finally, we apply our control algorithm to re-align circadian rhythm with the new light and dark cycle to treat jet lag [69] or adapt to night shift work [70, 71].

We compare our new algorithm with a previous algorithm based on standard phase reduction proposed in [20] by applying it to four different dynamical systems: the Hopf bifurcation normal form, cardiac pacemaker cells (motivated by suppressing alternans),

thalamic neurons (motivated by desynchronizing neurons via spike timing control), and circadian gene regulation in the superchiasmatic nucleus (motivated by controlling circadian rhythm). We show that our algorithm effectively changes the phase in these dynamical systems while keeping the controlled oscillator close to the unperturbed periodic orbit. The previous algorithm drives the oscillator away from the periodic orbit, and thus can fail. We also perform a parametric study to analyze the dependence of the control error on the nontrivial Floquet multiplier of the periodic orbit and on the amount of phase change desired. This study demonstrates the promising potential of our new algorithm over the previous algorithm, especially when a large change in phase is required or when a nontrivial Floquet multiplier of the oscillator is close to 1. In such cases, our algorithm does an order of magnitude better in terms of the calculated control error.

In this chapter, we consider dynamical systems that only have one of the nontrivial Floquet multipliers close to one, and the remaining  $n - 2$  nontrivial Floquet multipliers close to zero. Thus the augmented phase reduction is given by equations (3.2) and (3.3).

This chapter is organized as follows. In Section 4.2, we devise our optimal control algorithm based on augmented phase reduction and also present the previously devised algorithm based on standard phase reduction. In Section 4.3, we compare the two control algorithms by applying them to four different dynamical systems, and in turn develop strategies to suppress cardiac alternans, change the firing time of thalamic neurons, and shift the phase of a circadian rhythm. Section 4.4 analyzes the effect of noise on the performance of our control algorithm. Section 4.5 summarizes the results and gives concluding remarks. Numerical methods used in this chapter are detailed in Appendix B, and Appendix A lists the mathematical models used in this chapter.

The main results of this chapter have been published in [16].

## 4.2 Optimal Phase Control

Suppose we start at the point  $\mathbf{x}_0$  on  $\mathbf{x}^\gamma(t)$ . Without any control input, we expect the trajectory will return to the point  $\mathbf{x}_0$  at time  $t = T$ . Our objective here is to devise a control which returns the trajectory to its initial position after time  $t = T_1$ , where  $T_1 \neq T$ . It should do so using minimal energy input and staying close to the uncontrolled periodic trajectory. An “easy” way of doing this is by taking the control input to be a scalar multiple of the vector field,  $U(t) = sF(\mathbf{x})$ .  $s$  would be positive (resp., negative) when phase advance (resp., delay) is the control objective. However, there are three problems with such a control in an experimental setting: first, the dynamical system under consideration may not be fully actuated (not all the states of the system can be perturbed), which is generally the case in practical situations; second, the entire state of the system may not be experimentally measurable; and third, the function  $F(\mathbf{x})$  might be unknown.

Here we consider dynamical systems which only have one degree of actuation: the control input vector is  $U(t) = [u(t), 0, \dots, 0]^T$ . Such a control input is motivated by the applications we consider in this chapter, where only one of the elements of the state vector is affected directly by the control input. So the standard phase reduction becomes

$$\dot{\theta} = \omega + \mathcal{Z}_{x_1}(\theta)u(t), \quad (4.1)$$

and the augmented phase reduction is

$$\dot{\theta} = \omega + \mathcal{Z}_{x_1}(\theta)u(t), \quad (4.2)$$

$$\dot{\psi} = k\psi + \mathcal{I}_{x_1}(\theta)u(t). \quad (4.3)$$

Here  $\mathcal{Z}_{x_1}$  and  $\mathcal{I}_{x_1}$  correspond to the first component in the  $n$ -dimensional vector functions

$\mathcal{Z}$  and  $\mathcal{I}$ , respectively. Without loss of generality, we will do away with the subscripts and write them as  $\mathcal{Z}$  and  $\mathcal{I}$ . An optimal control law based on the augmented phase reduction is found by using the cost function  $C$ :

$$C = \int_0^{T_1} \left[ \alpha u^2 + \beta \psi^2 + \lambda_1 \left( \dot{\theta} - \omega - \mathcal{Z}(\theta)u(t) \right) + \lambda_2 \left( \dot{\psi} - k\psi - \mathcal{I}(\theta)u(t) \right) \right] dt. \quad (4.4)$$

The first term in the cost function ensures that the control law uses a minimum energy input. The second term minimizes the transversal distance (in the direction of the slow isostable coordinate  $\psi$ ) from the uncontrolled periodic trajectory, thus ensuring that the controlled trajectory stays close to the periodic trajectory where the reduction is valid. The coefficients  $\alpha$  and  $\beta$  give us the freedom to weight energy minimization and transversal distance minimization differently for different problems. The last two terms ensure that the system obeys the augmented phase reduction, with  $\lambda_1$  and  $\lambda_2$  being the Lagrange multipliers. The Euler-Lagrange equations are obtained from

$$\frac{\partial P}{\partial q} = \frac{d}{dt} \left( \frac{\partial P}{\partial \dot{q}} \right), \quad q = \lambda_1, \lambda_2, \theta, \psi, u, \quad (4.5)$$

where  $P$  is the integrand in the cost function  $C$ . This gives

$$\dot{\theta} = \omega + \mathcal{Z}(\theta)u(t), \quad (4.6)$$

$$\dot{\psi} = k\psi + \mathcal{I}(\theta)u(t), \quad (4.7)$$

$$\dot{\lambda}_1 = -u(\lambda_1 \mathcal{Z}'(\theta) + \lambda_2 \mathcal{I}'(\theta)), \quad (4.8)$$

$$\dot{\lambda}_2 = 2\beta\psi - k\lambda_2, \quad (4.9)$$

where

$$u(t) = \frac{\lambda_1 \mathcal{Z}(\theta) + \lambda_2 \mathcal{I}(\theta)}{2\alpha}. \quad (4.10)$$

These equations are solved as a two point boundary value problem (see Appendix B.3) with the boundary conditions

$$\theta(0) = 0, \quad \theta(T_1) = 2\pi, \quad \psi(0) = 0, \quad \psi(T_1) = 0. \quad (4.11)$$

The last boundary condition makes sure that trajectory ends back on the periodic orbit. The previously proposed optimal control problem based on standard phase reduction [20] can be obtained by setting  $\beta = 0$  and  $\lambda_2 = 0$  in the cost function. This gives Euler-Lagrange equations for the variables  $\theta$  and  $\lambda_1$  as

$$\dot{\theta} = \omega + \mathcal{Z}(\theta)u(t), \quad (4.12)$$

$$\dot{\lambda}_1 = -u\lambda_1 \mathcal{Z}'(\theta), \quad (4.13)$$

where

$$u = \frac{\lambda_1 \mathcal{Z}(\theta)}{2\alpha}. \quad (4.14)$$

These control laws (equations (4.10) and (4.14)) can then be applied to the full model  $\dot{\mathbf{x}} = F(\mathbf{x}) + U(t)$  to change the orbit's phase. To compare the control laws, we compute the control energy as

$$\int_0^{T_1} u^2 dt, \quad (4.15)$$

and the control error as the normalized Euclidean distance between the final position and the initial position given as

$$\frac{\|x(T_1) - x(0)\|}{\max(\|x(t)\|)}, \quad (4.16)$$

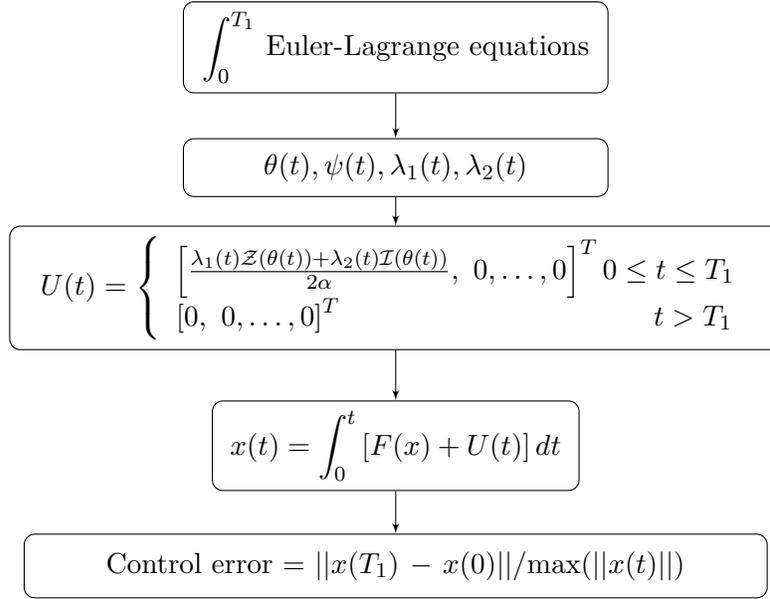


Figure 4.1: Flowchart describing the control algorithm based on augmented phase reduction

where  $\|x\|$  represents the standard Euclidean norm, and  $\max(\|x(t)\|)$  represents the maximum value of the Euclidean norm of the periodic solution  $x(t)$ . The control error arises because we apply the control input (equations (4.10) and (4.14)) based on the reduced model to the full model (equation (2.4)). The control algorithm based on augmented phase reduction is outlined in the flowchart in Figure 4.1. The algorithm based on standard phase reduction is implemented in a similar manner. We will see in Section 4.3 that our new control law is effective in circumstances in which the previously proposed control law fails since the novel attribute of our cost function minimizes the transversal distance, ensuring that the controlled trajectory is always close enough to the periodic orbit so that the phase reduction is valid. On the other hand, with the previously proposed control law, even a small control input can drive the trajectory away from the periodic orbit, thereby rendering the phase reduction invalid and the control law ineffective.

## 4.3 Applications

We apply the new optimal control algorithm (based on the augmented phase reduction) and the previously proposed optimal control algorithm (based on the standard phase reduction) to four different dynamical systems: the Hopf bifurcation normal form, cardiac pacemaker cells, thalamic neurons, and circadian gene regulation in the suprachiasmatic nucleus. For all these systems, the PRC is numerically computed using the software XPP [44]. We solve a two point boundary value problem to obtain the IRC as the periodic solution to equation (3.1) (see Appendix B.3). The control input is obtained by solving the Euler-Lagrange equations (4.6) - (4.9) or (4.12) - (4.13) as a two point boundary value problem numerically. It is then applied to the full model to compute the resulting trajectory  $\mathbf{x}(t)$ . A parametric study is performed to compute this error as a function of the ratio  $T_1/T$  and, for the Hopf bifurcation normal form, the nontrivial Floquet multiplier of the periodic orbit.

### 4.3.1 Hopf Bifurcation Normal Form

#### Motivation

Here we consider the normal form for a supercritical Hopf bifurcation [29], which occurs in several applications including biological and chemical oscillators [19, 72, 56, 73]. This example allows us to explore in detail the interplay between the control objective and the nontrivial Floquet multiplier for the new and the previously proposed control algorithm.

#### Control Strategy

We use our control algorithms to change the phase of a periodic orbit near a supercritical Hopf bifurcation. By varying parameters, we can calculate the control error for

both algorithms as a function of the nontrivial Floquet multiplier and the target phase change, which gives a sense of which control algorithm would work better in what ranges of these quantities.

The normal form of the supercritical Hopf bifurcation with an external control input  $u(t)$  is:

$$\dot{x} = ax - by + (x^2 + y^2)(cx - dy) + u(t), \quad (4.17)$$

$$\dot{y} = bx + ay + (x^2 + y^2)(dx + cy), \quad (4.18)$$

with  $c < 0$ . With zero control input  $u(t)$ , and  $a < 0$ , the system has a stable fixed point. As  $a$  increases through 0, a stable periodic orbit is born, and the fixed point becomes unstable. With parameters  $a = 0.004, b = 1, c = -1, d = 1$ , the system has a stable periodic orbit with the time period  $T = 6.2582$  and the nontrivial Floquet multiplier  $\exp(-2aT) = 0.9512$ . The PRC and the IRC are sinusoidal, see Table 3.1, cf. [47, 38], with amplitudes  $\sqrt{\frac{d^2+c^2}{-ac}}$  and  $\sqrt{1 + \frac{d^2}{c^2}}$ , respectively. Here,  $\theta = 0$  corresponds to the initial condition  $x = -0.0447, y = 0.0447$ . The top row of Figure 4.2 shows the uncontrolled periodic orbit, PRC, and IRC for the given parameter values. The control parameters  $\alpha$  and  $\beta$  are both taken to be unity. We calculate the optimal control with  $T_1 = 1.3T = 8.1356$  both for our new algorithm and the previously proposed algorithm.

The resulting trajectories, time series, and control inputs are shown in the bottom two rows of Figure 4.2. As seen in this figure, the new control algorithm does much better in changing the phase of the periodic orbit while also keeping the trajectory close to the periodic orbit for the uncontrolled system. This is because our algorithm minimizes the transversal distance, ensuring that the controlled trajectory is always close enough to the periodic orbit so that the phase reduction is valid. On the other hand, with the previously proposed control law, the control input drives the trajectory away from the periodic

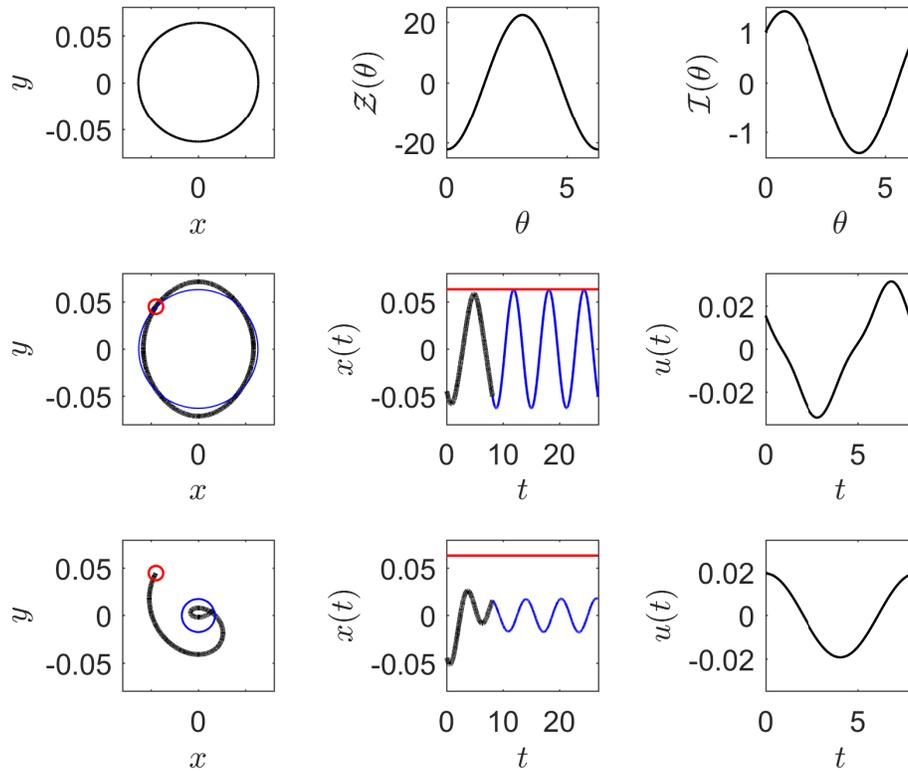


Figure 4.2: Hopf bifurcation normal form: Top row shows the uncontrolled periodic orbit, PRC, and IRC for the Hopf normal form with parameters given in the main text. The middle (resp., bottom) row shows the trajectory, time series, and control input for control based on our new (resp., the previously proposed) algorithm. Control is on (resp., off) for the portion shown by the thick black (resp., thin blue) line. The trajectory starts at the small red circle. The red horizontal line shows the amplitude of the uncontrolled periodic orbit.

orbit, thereby rendering the phase reduction invalid and the control law ineffective. This is apparent from the control error (given by equation (4.16)) as well, which is 0.1435 and 1.1394 for the new and the previous optimal control algorithms, respectively. However, the new control algorithm does better at the expense of consuming more energy (given by equation (4.15)), which comes out to be 0.0032 units, compared with 0.0015 units for the previous control algorithm. We note that the trajectory in the bottom row of Figure

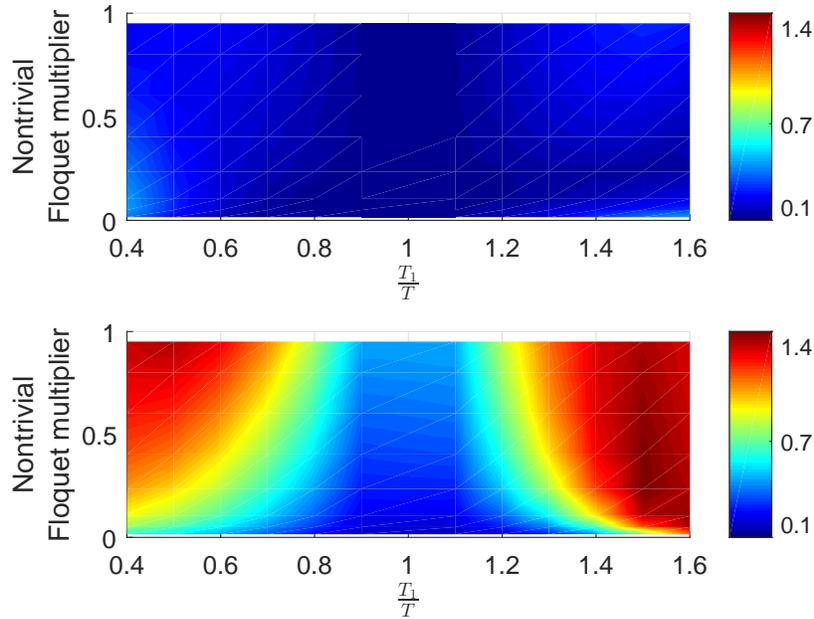


Figure 4.3: Hopf bifurcation normal form: Top (resp., bottom) row shows the control error (equation (4.16)) from the control based on our new (resp., the previously proposed) algorithm as a function of the nontrivial Floquet multiplier and the ratio  $T_1/T$ .

4.2 will eventually return to the stable uncontrolled periodic orbit, but will not have the corresponding desired phase shift.

As the parameter  $a$  is further increased, the system moves away from the bifurcation point, resulting in a decreasing nontrivial Floquet multiplier. A parametric study is performed to analyze the dependence of the control error on the nontrivial Floquet multiplier and the ratio  $T_1/T$ . The top (resp., bottom) row of Figure 4.3 shows this error for the new (resp., the previously proposed) control algorithm. The error for the previously proposed control algorithm increases as the nontrivial Floquet multiplier increases towards 1 and/or ratio  $\frac{T_1}{T}$  moves away from 1 (the control objective becomes more extreme). This is because an extreme control objective requires a large control input, which drives the trajectory away from the periodic orbit, resulting in the phase reduction losing accuracy.

However when the nontrivial Floquet multiplier is close to zero, a trajectory kicked away from the periodic orbit returns quickly back to it, thereby nullifying the effect of a large control input on the accuracy of phase reduction. On the other hand, for the new control algorithm, the error remains small for all values of the ratio  $\frac{T_1}{T}$  and nontrivial Floquet multiplier considered. Thus, we can conclude that our new control algorithm is much more effective than the previously proposed control algorithm, especially when the control objective is extreme and/or the nontrivial Floquet multiplier of the periodic orbit is close to 1.

We expect that the asymmetry in control error, as seen in the bottom panel of Figure 4.3, can be explained by the inherent shear present in the model's dynamics [74]. For the parameters considered, we observe that when phase delay is the desired control objective, the trajectory is kicked inside the periodic orbit, i.e., the amplitude of the transient trajectory decreases. On the other hand, for a phase advance control objective, the trajectory is kicked out of the periodic orbit, i.e., the amplitude of the transient trajectory increases. The difference between this amplitude increase and decrease is magnified for the standard phase reduction-based control with a small Floquet multiplier. Shear present in the dynamics acts differently on these two cases, which is reflected as a small asymmetry in control error seen in the bottom panel of Figure 4.3. For the new control algorithm, the difference between the amplitude increase and decrease stays relatively small, and thus the control error is more symmetric as can be seen in the top panel of Figure 4.3.

### 4.3.2 Controlling Cardiac Pacemaker Cells

#### Motivation

The heartbeat is initiated by a collection of cells in the Sinoatrial node (SA node), which acts as a pacemaker. These cells elicit periodic electrical pulses which polarize a collection of excitable and contractile cells called myocytes. In the process of depolarizing, myocytes contract and propagate action potentials to the neighboring cells. This well-coordinated process of excitation / depolarization and contraction enables the heart to pump blood throughout the body. Under normal conditions, with constant pacing, the action potential duration (APD), that is the time for which an action potential lasts, also remains constant. However, under some conditions, this 1:1 rhythm between pacing and the APD can become unstable, bifurcating into a 2:2 rhythm of alternating long and short APD, known as alternans [75]. Alternans is observed to be a possible first step leading to fibrillation [76]. Thus, a number of researchers have worked on suppressing alternans as a method of preventing fibrillation, thereby preventing the need for painful and damaging defibrillating shocks. Many of these methods [77, 78, 79, 80] operate by exciting the myocardium tissue externally with periodic pulses, and changing the period according to the alternating rhythm. However such a control requires excitation at several sites in the tissue [81].

#### Control Strategy

We devise a novel strategy to suppress alternans by changing the phase of the inherent pacemaker cells. Such a control strategy could eliminate the need to excite the tissue at multiple sites. We make use of the relation between APD, diastolic interval (DI), and basic cycle length (BCL) to devise our control strategy. DI is the time for which a myocyte cell remains depolarized, and BCL is the time between successive action potentials, which

is dictated by the period of the pacemaker cells. In the simplest model [82], APD is a function of the previous DI, given by the *restitution curve*:  $APD_i = f(DI_{i-1})$ . DI is a function of the current APD, given by what we call the *BCL curve*:  $APD_i + DI_i = BCL$ . The intersection of these two curves gives the normal 1:1 rhythm. If the slope of the restitution curve at this intersection is greater than 1, the 1:1 rhythm is unstable, giving rise to alternans. This was first shown by [83] and is illustrated in the top panel of Figure 4.4. Given the current  $DI_i$ , the next  $APD_{i+1}$  is given by the restitution curve. Traversing horizontally from this point to the BCL curve gives the next  $DI_{i+1}$ . Repeating this analysis gives all the successive DIs and APDs. Under constant BCL, it is graphically illustrated in the top panel of Figure 4.4 that when slope of the restitution curve is greater than 1, APDs and DIs alternate between two values, corresponding to alternans. If starting from point 2 in the bottom panel of Figure 4.4 we reduce the BCL for one cycle such that the next DI corresponds to the unstable 1:1 state, this would stabilize the 1:1 rhythm, eliminating alternans. Thus our control strategy to eliminate alternans corresponds to decreasing the BCL for one cycle, i.e., advancing the phase of the SA node cells. Note that in a clinical setting, we may need to apply this control strategy multiple times, as the 1:1 rhythm is unstable. Applying control multiple times is physically realistic as long as the trajectory returns back to the limit cycle before the next stimulus arrives. As we will see later, this is not the the case with the previously proposed optimal control algorithm based on the standard phase reduction. This novel strategy should be clinically feasible as well, since an implanted battery could generate multiple stimuli.

To demonstrate our approach, we consider the SA node cell dynamics, instead of the discrete APD/DI dynamics. Our control objective is to change the phase of SA node cells; the amount of change required is linked to the amount by which the BCL curve needs to be shifted to stabilize the unstable APD/DI dynamics. Here we advance the phase by 20 % as an example. We consider the 7-dimensional YNI model of SA node cells

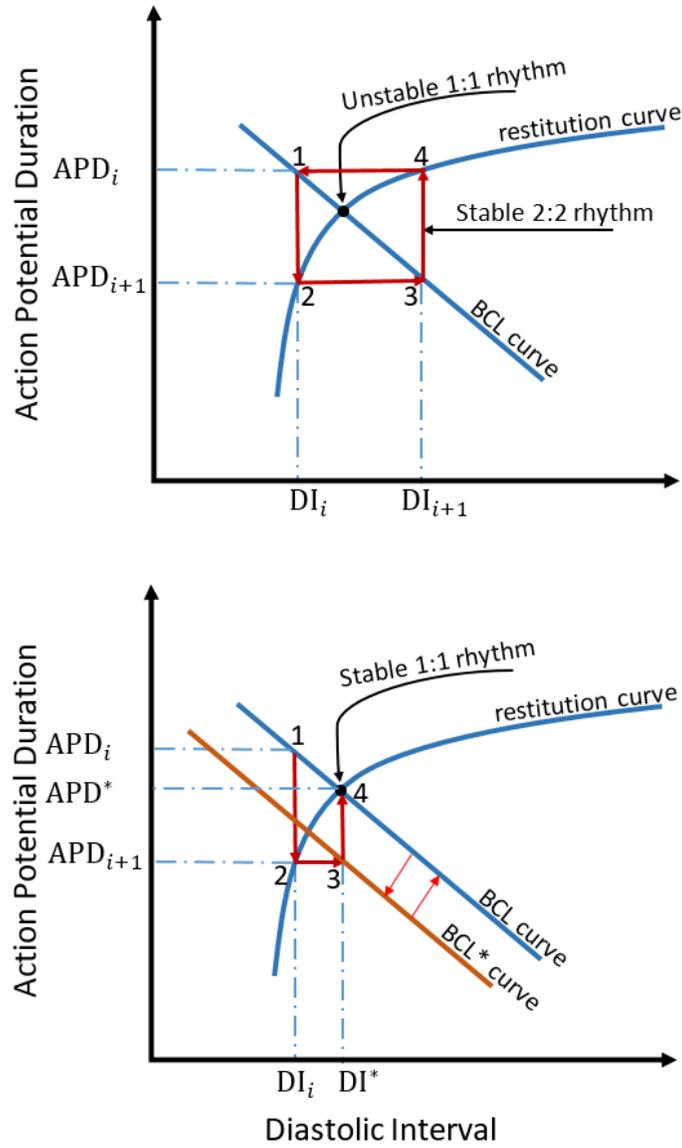


Figure 4.4: Suppression of alternans: The top panel shows the stable 2:2 rhythm of alternans. The bottom panel shows the 1:1 rhythm stabilized by reducing the BCL for one cycle.

in rabbit heart proposed by [84]. The model is of Hodgkin-Huxley type with 6 gating

variables  $d, f, m, h, q, p$  and a transmembrane voltage variable  $V$ . The model is given as

$$\dot{V} = \frac{I_m - I_{Na} - I_k - I_l - I_s - I_h}{C} + u(t), \quad (4.19)$$

$$\dot{y} = \alpha_y(1 - y) - \beta_y y, \quad (4.20)$$

where  $y$  represents the 6 gating variables.  $u(t)$  represents the applied current as the control input. For details of the currents ( $I_{Na}, I_k, I_l, I_s, I_h$ ) and the parameters, see Appendix A.2. With  $I_m = 1.0609$ , we get a stable periodic orbit with time period  $T = 203.4552 \text{ ms}$  and nontrivial Floquet multipliers  $0.7595, 0.1365, 0.0299, \approx 0, \approx 0, \approx 0$ . Since one of the nontrivial Floquet multipliers is considerably larger than others, we only consider the isostable coordinate corresponding to it. The top row of Figure 4.5 shows the uncontrolled periodic orbit, PRC, and IRC for the given parameter values. Control parameters  $\alpha$  and  $\beta$  are taken as 100 and 0.1, respectively. Here we give considerable more weight to minimizing energy, to overcome our new control algorithm's tendency for this problem to require more energy than the previously proposed control algorithm. We calculate optimal control for the new and previously proposed algorithms with  $T_1 = 0.8T = 162.7641 \text{ ms}$ .

The resulting trajectories, time series, and control inputs are shown in the bottom two rows of Figure 4.5. As seen in this figure, our new control algorithm successfully achieves the control objective while keeping the trajectory close to the uncontrolled periodic orbit. It is able to do so while giving considerable importance to energy minimization ( $\alpha$  is significantly bigger than  $\beta$ ). On the other hand, with the previous control algorithm, instead of staying close to the periodic orbit, the trajectory decays to the stable fixed point of the system. This is evident from the control error, which is 0.0858 and 0.3677 for our new and the previous optimal control algorithms, respectively. Our control does better at the expense of consuming more energy (6.3850 units) than the previous control (0.2100

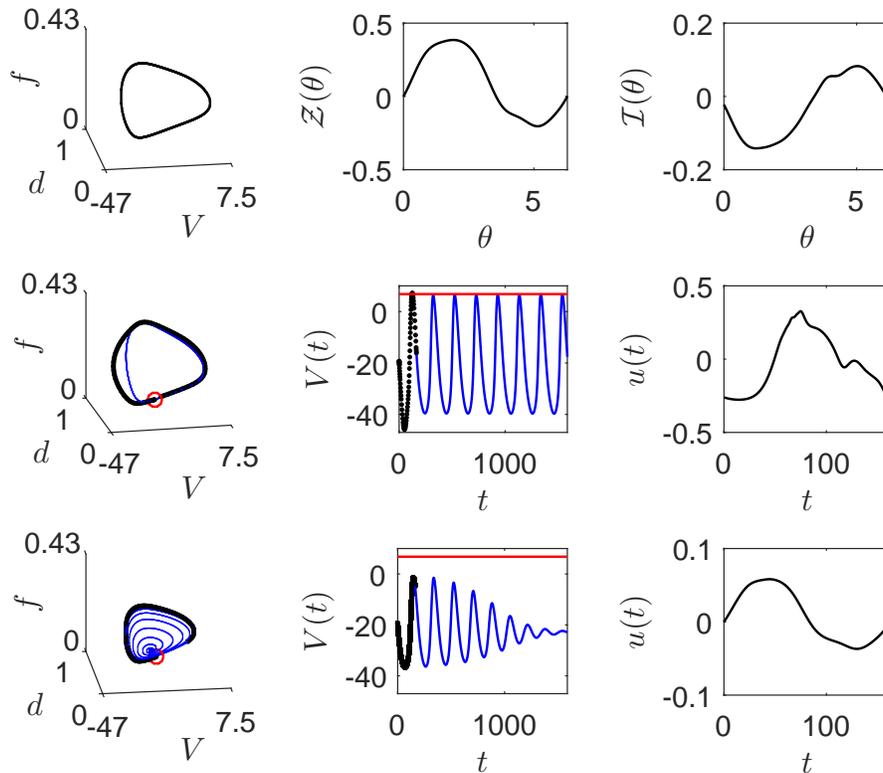


Figure 4.5: YNI model for cardiac pacemaker cells: Top row shows the uncontrolled periodic orbit, PRC, and IRC for the YNI model with parameters given in the main text. The middle (resp., bottom) row shows the trajectory, time series, and control input for control based on our new (resp., the previously proposed) algorithm. Control is on (resp., off) for the portion shown by the thick black (resp., thin blue) line. The trajectory starts at the small red circle. The red horizontal line shows the amplitude of the uncontrolled periodic orbit.

units). Note that here we change the phase by 20% as an example. In a more realistic setting, we would require a more integrated model which combines the discrete APD/DI dynamics together with the dynamics of the SA node cell. This would automatically determine the phase change required.

### 4.3.3 Controlling Neurons

#### Motivation

Essential and parkinsonian tremor, the most common movement disorders, affect millions of people worldwide. These cause involuntary tremors in various parts of the body, disrupting the activities of daily living. Pathological neural synchronization in the thalamus and the STN brain region is hypothesized to be one of the causes of motor symptoms of essential and parkinsonian tremor, respectively [11, 12]. Deep brain stimulation (DBS), an FDA approved treatment, helps to alleviate these symptoms [85, 86] by stimulating the thalamus or the STN brain regions with a high frequency high energy pulsatile waveform. In the process, the high frequency high energy waveform has been hypothesized to desynchronize the synchronized neurons; see, e.g., [87, 15]. This has motivated researchers to come up with efficient control techniques [88, 67] which not only desynchronize the neurons but also consume less energy, thus prolonging the battery life of the stimulator and preventing tissue damage caused by the high energy pulsatile stimuli.

#### Control Strategy

At a single neuron level, desynchronization can be viewed as changing the phase of a neuron to be at a different phase than other neurons [20, 68]. With this in mind, we use our algorithm to change the phase of neuron spikes in thalamic neurons. To see the performance of our algorithm in an extreme scenario, we set the control objective to advance the phase by 60%. We demonstrate this by using the thalamic neuron model introduced in Chapter 2. For details of the model, see Appendix A.1. Under zero control input, these parameters give a stable periodic orbit with time period  $T = 8.3955 \text{ ms}$  and nontrivial Floquet multipliers 0.8275 and 0.0453. Since one of the nontrivial Floquet mul-

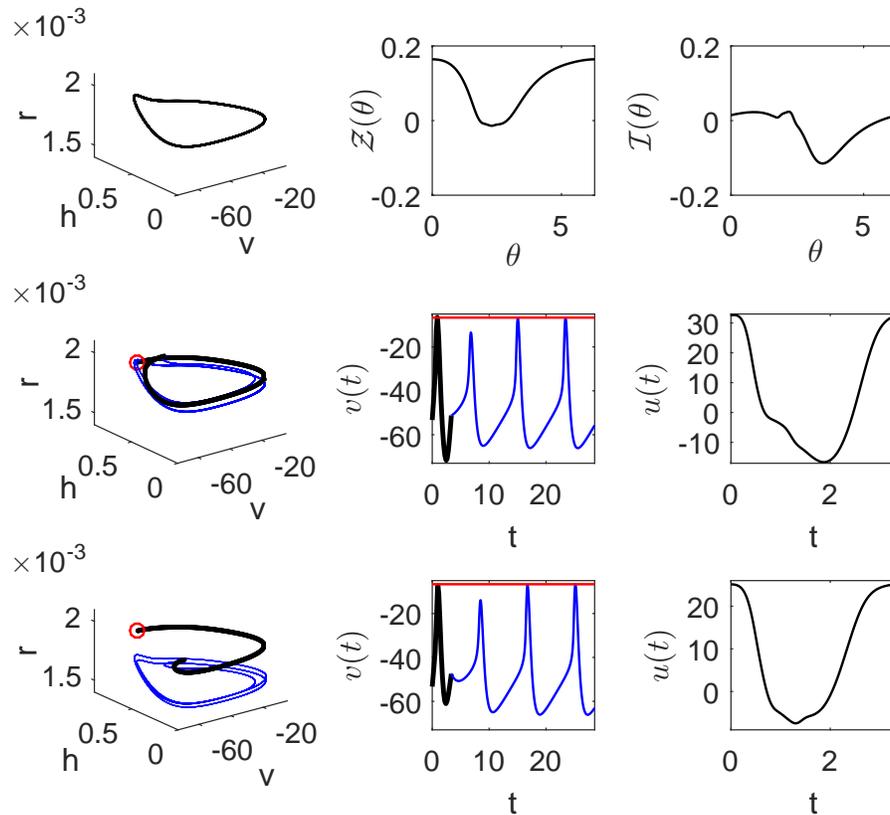


Figure 4.6: Thalamic neuron model: Top row shows the uncontrolled periodic orbit, PRC, and IRC for the thalamic neuron model with parameters given in the main text. The middle (resp., bottom) row shows the trajectory, time series, and control input for our new (resp., the previously proposed) control algorithm. Control is on (resp., off) for the portion shown by the thick black (resp., thin blue) line. The trajectory starts at the small red circle. The red horizontal line shows the amplitude of the uncontrolled periodic orbit.

tiplier is close to 0, we only consider the isostable coordinate corresponding to the larger nontrivial Floquet multiplier in the augmented phase reduction. The top row of Figure 4.6 shows the uncontrolled periodic orbit, PRC, and IRC for the given parameter values. Control parameters  $\alpha$  and  $\beta$  are taken as unity. We calculate the optimal control for our new algorithm and the previously proposed algorithm with  $T_1 = 0.4T = 3.3582$  ms.

The resulting trajectories, time series, and control inputs are shown in the bottom

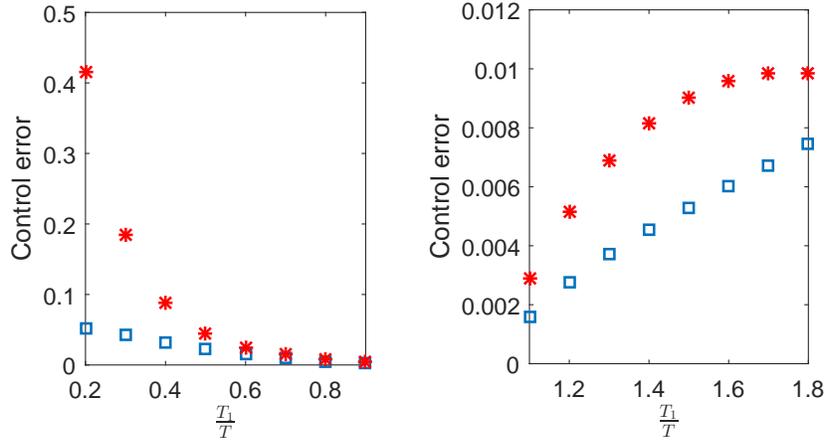


Figure 4.7: Thalamic neuron model: Blue  $\square$  (resp., red  $*$ ) shows the control error for the control from our new (resp., the previously proposed) algorithm as a function of the ratio  $T_1/T$ .

two rows of Figure 4.6. As seen in middle and bottom panels of the left column of this figure, our new control algorithm does better in keeping the trajectory close to the periodic orbit. On the other hand, with the previous control algorithm, the trajectory moves away from the periodic orbit. Looking at the central middle and bottom panels, it seems that the trajectory returns back to the periodic orbit even for the previously proposed optimal control algorithm, but this is not the case. Since one of the Floquet multipliers is close to zero, the voltage state returns back quickly, but the other states still remain far away from the limit cycle. This is evident from the first two panels of the bottom row of Figure 4.6, as well as from the control error, which is 0.032 and 0.088 for our new and the previous optimal control algorithms, respectively. Our new control algorithm does better at the expense of consuming more energy (1119.15 units) compared to (784.16 units) in the previous control algorithm.

We test the control algorithms for various target phase changes, corresponding to the range from  $T_1 = 0.2T$  to  $T_1 = 1.8T$ . Figure 4.7 shows control error for these phase changes for both our new and the previous optimal control algorithms. We see that the

control error grows as the control objective becomes more extreme, which is expected. But it still remains relatively small for our new control algorithm. This again shows that our new control algorithm is more effective in changing the phase than the previously proposed control algorithm.

### 4.3.4 Controlling Circadian Oscillators

#### Motivation

Neurons in the suprachiasmatic nucleus (SCN) of the brain are responsible for maintaining the circadian rhythm in mammals. This rhythm is synchronized with the external day and night cycle under normal conditions. A disruption between these two rhythms can happen due to multiple reasons, such as travel across time zones, starting a night shift job, working in extreme environments (space, earth poles, underwater), etc. Such an asynchrony leads to several physiological disorders like insomnia, improper digestion, and even cancer and cardiovascular diseases [89, 90], thus driving researchers to try to develop ways to remove this asynchrony. One way of doing this is by using a light stimulus, which affects the circadian rhythm [91]. Therefore, many researchers have used appropriately timed exposure to light to entrain circadian rhythm with the new external cycle; see, e.g., [69, 70, 71].

#### Control Strategy

Several control-theoretic approaches have been used in the past to determine timing and intensity of the light stimulus to synchronize the circadian rhythm with a new light - dark cycle [92, 93, 94]. One way of doing this is by changing the phase of one circadian oscillation so that the oscillation gets aligned with the external cycle after the end of the controlled oscillation. As an example, consider a person who is going on a vacation to

London, traveling east from New York City. The day-night cycle in his new environment would be 5 hours behind his internal rhythm. Thus, advancing the phase of his internal circadian rhythm by 20 percent ( $\approx 5$  hours) for one cycle would realign his internal rhythm with the new environment. This would be equivalent of taking  $T_1 = 0.8T$  in our control algorithm.

We use the 3-dimensional model of the clock gene regulation in SCN developed in [95] to demonstrate our control algorithm. This model has a negative feedback loop, where production of one gene leads to the inhibition of the other, thus causing oscillatory behavior. It is given as:

$$\dot{X} = v_1 \frac{K_1^4}{K_1^4 + Z^4} - v_2 \frac{X}{K_2 + X} + L(t), \quad (4.21)$$

$$\dot{Y} = k_3 X - v_4 \frac{Y}{K_4 + Y}, \quad (4.22)$$

$$\dot{Z} = k_5 Y - v_6 \frac{Z}{K_6 + Z}. \quad (4.23)$$

Here  $X$  represents mRNA concentration of a clock gene, *per* or *cry*,  $Y$  represents the resulting protein, PER or CRY [96], and  $Z$  is the active protein which inhibits production of the clock gene.  $L(t)$ , the perturbation in ambient light, acts as the control input. Parameters  $v_1, K_1, v_2, K_2, k_3, v_4, K_4, k_5, v_6, K_6$  are taken from Figure 1 in [95], and are given in Appendix A.3. These parameters give a stable periodic orbit with time period  $T = 23.5398$  hrs and the nontrivial Floquet multipliers 0.9509 and  $\approx 0$ . Since one of the nontrivial Floquet multipliers is approximately 0, we only consider the isostable coordinate corresponding to the larger nontrivial Floquet multiplier in the augmented phase reduction. The top row of Figure 4.8 shows the uncontrolled periodic orbit, PRC, and IRC for the given parameter values. We have taken the control parameters  $\alpha = 10$

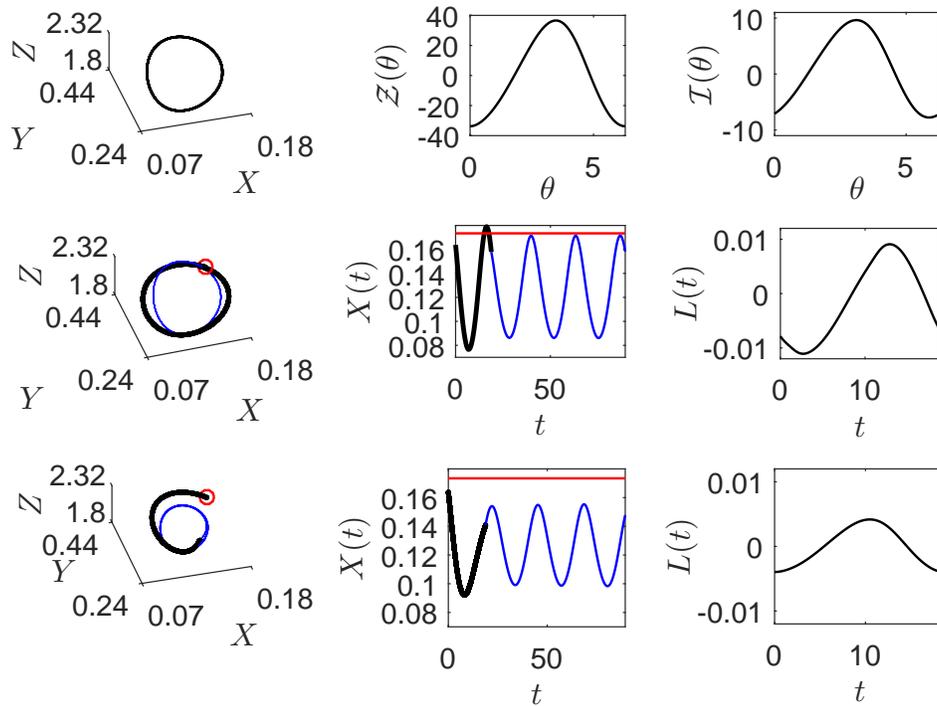


Figure 4.8: Circadian oscillator: Top row shows the uncontrolled periodic orbit, PRC, and IRC for the circadian oscillator model with parameters given in the main text. The middle (resp., bottom) row shows the trajectory, time series, and control input for control based on our new (resp., the previously proposed) algorithm. Control is on (resp., off) for the portion shown by the thick black (resp., thin blue) line. The trajectory starts at the small red circle. The red horizontal line shows the amplitude of the uncontrolled periodic orbit.

and  $\beta = 0.1$ . We again give more weight to minimizing energy to compensate for our new control algorithm's tendency to require more energy than the previously proposed control algorithm for this problem.

The resulting trajectories, time series, and control inputs are shown in the bottom two rows of Figure 4.8. We see that our new control algorithm is able to advance the phase while keeping the trajectory close to the unperturbed periodic orbit. It is able to do so while giving considerable importance to energy minimization ( $\alpha$  is 100 times bigger than  $\beta$ ). On the other hand, with the previous control algorithm, the trajectory

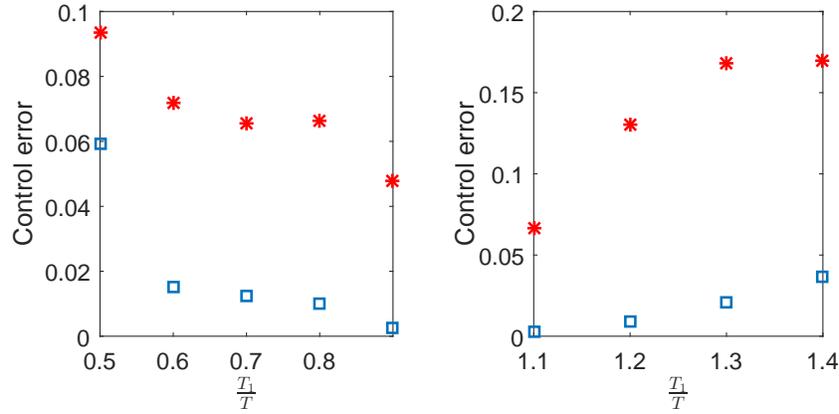


Figure 4.9: Circadian oscillator: Blue  $\square$  (resp., red  $*$ ) shows the control error from the control based on our new (resp., the previously proposed) algorithm as a function of the ratio  $T_1/T$ .

moves away from the unperturbed periodic orbit. This is apparent from the control error as well, which is 0.0099 and 0.0665 for our new and the previous optimal control, respectively. Our new control algorithm does better at the expense of consuming more energy (0.00096 units) than the previous control algorithm (0.00015 units).

We also test our algorithm for more extreme cases of asynchrony, ranging from  $T_1 = 0.5T$  (traveling west and gaining 12 hours in time) to  $T_1 = 1.4T$  (traveling east and losing 9 hours in time). Figure 4.9 shows the control error for these cases for both our new and the previous control algorithm. The control error increases as the control objective becomes more extreme, but it still remains relatively small for our new control algorithm. This again demonstrates the effectiveness of our new control algorithm over the previously proposed control algorithm.

## 4.4 Effect of Noise

So far we have demonstrated that our new control is effective in deterministic systems. However, real systems are subjected to noise, so here we analyze how such noise affects

the performance of our new algorithm. We calculate control from the deterministic phase model ((4.10) and (4.14)), and apply it to the full model with added white noise. So in effect we consider noise to be an external disturbance that affects only the first state variable that we control directly. Thus, we simulate the stochastic dynamical system

$$\frac{d\mathbf{x}}{dt} = F(\mathbf{x}) + [u(t) + \sigma\eta(t), \dots, 0]^T. \quad (4.24)$$

Here  $\sigma\eta(t) = \sigma\mathcal{N}(0,1)$  is zero mean white noise with strength  $\sigma$ . To simulate this equation numerically, we rewrite it as

$$d\mathbf{x} = F(\mathbf{x})dt + [u(t)dt + \sigma dW(t), \dots, 0]^T, \quad (4.25)$$

where  $dW(t) = \eta(t)dt$  and  $W(t)$  is the standard Weiner process. We use the second order Runge-Kutta algorithm developed in [97] to simulate the above equation.

To analyze the effect of noise on the performance of our control algorithm, we perform a parameteric study by calculating control error as a function of the nontrivial Floquet multiplier and the ratio  $T_1/T$  for the Hopf bifurcation normal form. We take the noise strength  $\sigma = 0.1r_{po}$ , where  $r_{po} = \sqrt{-a/c}$  is the radius of the periodic orbit. This ensures that the relative noise strength remains the same as the radius of the periodic orbit varies with the parameter  $a$ . The top (resp., bottom) row of Figure 4.10 shows the control error for the new (resp., the previously proposed) control algorithm in presence of white noise. We see that this figure is very similar to Figure 4.3 where we did not include white noise. The addition of white noise increases control errors for both the algorithms slightly, but the algorithm based on the augmented phase reduction still does much better than the algorithm based on standard phase reduction. Noise doesn't affect the performance of the previous algorithm when the nontrivial Floquet multiplier is close to 0 and the ratio  $T_1/T$

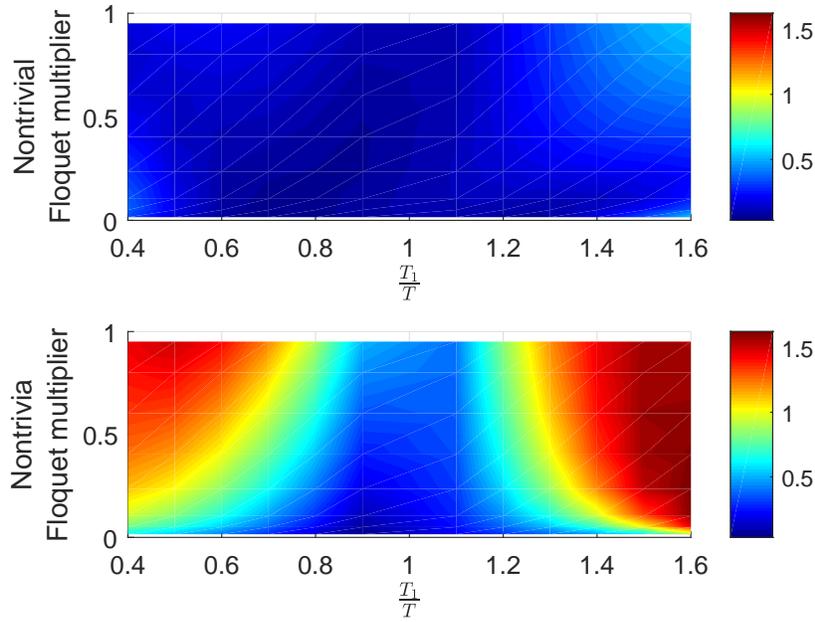


Figure 4.10: Hopf bifurcation normal form with white noise: Top (resp., bottom) row shows the control error (equation (4.16)) from the control based on our new (resp., the previously proposed) algorithm for the system with white noise.

is close to one. This is because any perturbation caused by the noise is nullified quickly under the evolution of the vector field. However the control error for our algorithm based on the augmented phase reduction remains small in the presence of noise for all analyzed values of the nontrivial Floquet multipliers and ratios  $T_1/T$ .

We also present results for controlling the circadian oscillator from Section 4.3.4 with white noise added to the  $X$  equation. Here we take the noise strength  $\sigma = 0.004$ , and the rest of the control parameters are the same as before. The corresponding results displayed in Figure 4.11 show that white noise drives the controlled trajectory slightly further away from the periodic orbit for both algorithms, but our new control is still able to bring the trajectory close to where it started at time  $T_1$ . However, the previously proposed control algorithm fails to do so. Thus these results demonstrate the effectiveness of our new control algorithm in the presence of noise.

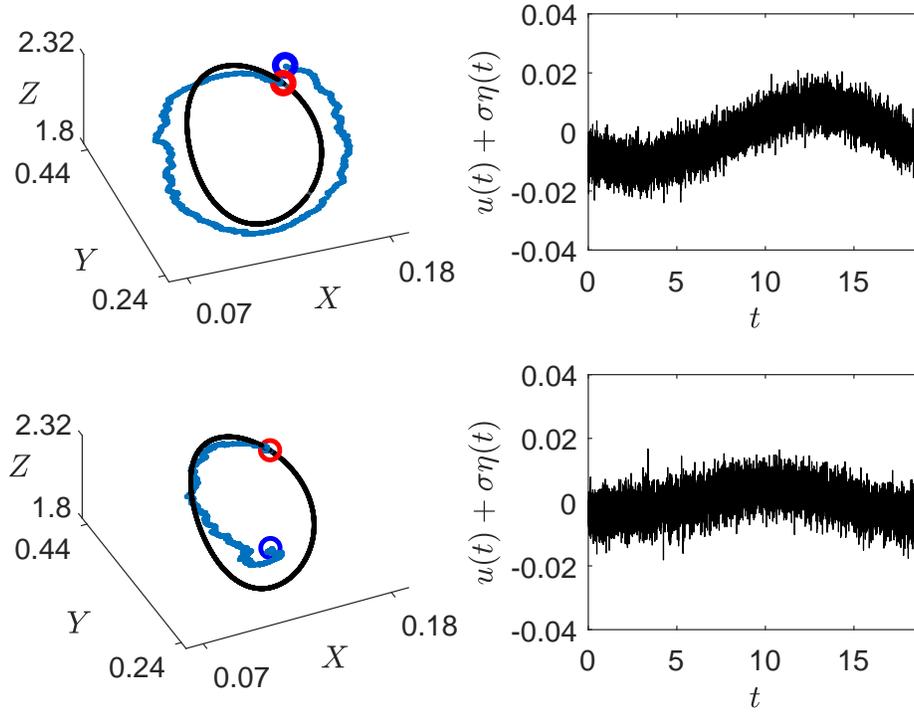


Figure 4.11: Circadian oscillator with noise: Top left (resp., bottom left) panel shows the controlled trajectory in blue for control based on our new (resp., the previously proposed) algorithm, and the periodic orbit in black. The trajectory starts at the small red circle and reaches the small blue circle at time  $T_1 = 0.8T$ . Top right (resp., bottom right) panel shows the control input added to the noise for our new (resp., the previously proposed) algorithm.

## 4.5 Discussion and Conclusion

In this chapter, we have developed a novel optimal control algorithm based on the augmented phase reduction to change the phase of a periodic orbit. Our algorithm not only minimizes the total energy consumption but also reduces the controlled trajectory’s transversal distance from the uncontrolled periodic orbit. This is because of inclusion of both the “energy” ( $u^2$ ) and the “transversal distance” term ( $\psi^2$ ) in the cost function ensures that the control input remains small overall, and keeps the controlled trajectory from getting far away from the unperturbed periodic orbit.

Our algorithm is applicable to generic oscillators, which we have demonstrated for a diversity of applications. We compared the performance of our algorithm as a function of both the Floquet multiplier and the desired phase change by applying it to the normal form for the supercritical Hopf bifurcation. We devised a novel approach to eliminate alternans by changing the phase of the pacemaker cells, and showed how our optimal control algorithm can be tied to the formulated geometrical approach. Such a control strategy could remove the need to excite the myocardium tissue at multiple sites. We also applied our algorithm to change the phase of thalamic neurons, which could be useful for desynchronizing pathologically synchronized neurons, thus holding potential to alleviate the motor symptoms of essential and parkinsonian tremor. Such an optimal control is expected to consume less energy than the pulsatile current used in present DBS protocol, thus prolonging the battery life of the stimulator, and also preventing tissue damage caused by the high energy DBS stimuli. Additionally, we applied the algorithm to change the phase of the clock gene regulation in SCN, which has relevance to treating jet lag or to adapting to night shift work. Finally, we showed that our algorithm performs well even in the presence of noise.

For some systems, the previous control algorithm based on the standard phase reduction could not keep the trajectory close to the unperturbed periodic orbit, and thus failed in achieving the desired control objective. We showed that our new algorithm works much better than the previous algorithm, especially when a nontrivial Floquet multiplier of the periodic orbit is close to 1 and/or a significant change in phase is required. In such cases, our new algorithm can do an order of magnitude better in terms of the calculated control error. From the right column of Figures 4.2, 4.5, 4.6, and 4.8, we see that the control inputs for both of the control algorithms have similar shape, but are shifted in phase. As seen in these figures, for our new control algorithm, the control input is large when the IRC is near zero, and is small when the IRC is large. This diminishes the effect of the

control input on the isostable coordinate, and thus the oscillator's transversal distance from the periodic orbit remains small. This ensures that the augmented phase reduction represents the dynamics accurately, making the control more effective. Our new control algorithm does better at the expense of consuming more energy than the previously proposed control algorithm. We expect that by tuning the control parameters  $\alpha$  and  $\beta$ , this energy difference can be reduced.

# Chapter 5

## Phase Distribution Control

### 5.1 Introduction

Populations of nonlinear oscillators are found in a variety of applications from physics, chemistry, biology, and engineering [1, 2, 3, 4]. The collective behavior of such oscillators varies, and includes synchronization, desynchronization, and clustering. Such diversity of collective behavior has motivated researchers to develop specific control techniques to achieve different behaviors. For example, [94, 93, 98] develop control to promote synchrony, [88, 87, 21] develop control to promote desynchronization, and [13, 14] develop ways to promote clustering. We note that some of these previously proposed algorithms to promote collective behavior are based on individual neuron models [20, 99, 100, 21], and some can face implementation challenges if they require observability of phases of all neurons at all times [100], or demand initial phases to be sufficiently close [21, 23]. There are also population-level algorithms for desynchronization in the literature which use multiple inputs [101, 88, 23], making experimental implementation challenging because they require multiple electrodes to be implanted in a small region of brain tissue.

In this chapter we overcome these difficulties by developing unified control frame-

works which can achieve all of the collective behaviors mentioned above using a single control input. Our algorithms are based on standard phase reduction, which has been instrumental in the development of many of the above control algorithms. The algorithms presented in this chapter use a partial differential equation (PDE) formulation which governs the evolution of the probability distribution of phases (phase distribution) of a population of identical, uncoupled oscillators [18, 23]. We use Fourier analysis to decompose this PDE into a system of ordinary differential equations (ODEs) governing the evolution of the Fourier coefficients of the phase distribution. Thus, to transform the phase distribution of an oscillator population to a desired distribution, we drive the corresponding Fourier coefficients to the Fourier coefficients of the desired distribution. Our first two algorithms are Lyapunov-based, which work by decreasing the  $L^2$  norm difference between the current and the desired phase distributions. Note that a related control algorithm has been published in [102], where we did not employ Fourier analysis to decompose the PDE into a systems of ODEs, but like the present first two algorithms it also decreases the  $L^2$  norm difference between the current and the desired phase distributions. This formulation in Fourier space makes our algorithm suitable for using a pseudospectral method for more accurate numerical simulation of the PDEs, which enables us to realize new control objectives and applications discussed in Section 5.5. Such a formulation also allows us to obtain the degenerate set of phase distributions and phase response curves for which the control would not work.

Our third algorithm is an optimal control algorithm, which unlike the previous two algorithms, minimizes the control energy consumption while achieving the desired control objective. We formulate it by constructing a cost function in terms of the Fourier coefficients of the phase distribution. This formulation results in high dimensional Euler-Lagrange equations that we solve as a two point Boundary Value Problem (BVP) numerically. Since the BVP is high dimensional, we construct a modified Newton Iteration

method that is effective for our problem. To demonstrate the effectiveness of our control algorithms for each of the applications considered, we apply them to a population of 100 uncoupled phase oscillators, and show that the population of phase oscillators with the applied control mimics the desired phase distribution. Other control algorithms based on the probability distribution of phases include [21, 103, 104].

This chapter is organized as follows. In Section 5.2, we develop a pseudospectral framework to write distributions as a finite Fourier series, and devise a Lyapunov-based control algorithm to control their Fourier coefficients. We construct a degenerate set of phase distributions and phase response curves for which the devised control would not work in Section 5.3. In Section 5.4, we detail the pseudospectral method used for numerical simulations throughout the chapter. In Sections 5.5, we demonstrate versatility of our control through several diverse applications and show the corresponding simulation results. In the same section, we formulate the phase difference distribution and prove some of its properties. In Section 5.6, we devise another Lyapunov-based control to take into account the effect of white noise on the oscillator population. We develop our optimal control algorithm in Section 5.7 and compare it with our Lyapunov-based algorithm. Section 5.8 summarizes our work and concludes by suggesting future extensions and tools needed for experimental implementation of our algorithms. Numerical methods used in this chapter are detailed in Appendix B, and Appendix A lists the mathematical models used in this chapter.

The main results of this chapter have been published in [102] and [105].

## 5.2 Control Algorithm

In this section, we devise a control algorithm to change the probability distribution of a population of oscillators. We do this by approximating the probability distribution

as a finite Fourier series and controlling its Fourier coefficients. This algorithm can be applied to a network of noise-free, identical, uncoupled oscillators to achieve any desired traveling-wave probability distribution, as long as the combination of phase distributions and the phase response curve is non-degenerate. A related control algorithm was given in [102], but here we formulate the algorithm in terms of Fourier coefficients; this is better suited for determining the control input using a pseudospectral method which does not have numerical dissipation unlike the method of lines approach used in [102].

Similar to Chapter 4, we consider control inputs of the form  $U(t) = [u(t), 0, \dots, 0]^T$  in this chapter. Thus the standard phase reduction is given by equation (4.1). The control algorithm in this chapter can be formulated for a more general control input as well, but as a matter of convenience, we only consider control input of the above form.

### 5.2.1 Phase density equation

Given a population of noise-free, identical, uncoupled oscillators all receiving the same control input, it is convenient to represent the population dynamics in terms of its probability distribution  $\rho(\theta, t)$ , with the interpretation that  $\rho(\theta, t)d\theta$  is the probability that an oscillator's phase lies in the interval  $[\theta, \theta + d\theta]$  at time  $t$ . This evolves according to the advection equation [18, 23, 102]

$$\frac{\partial \rho(\theta, t)}{\partial t} = -\frac{\partial}{\partial \theta} [(\omega + \mathcal{Z}(\theta)u(t)) \rho(\theta, t)]. \quad (5.1)$$

The desired final probability distribution  $\rho_f(\theta, t)$  will be taken to be a traveling wave which evolves according to [102]

$$\frac{\partial \rho_f(\theta, t)}{\partial t} = -\omega \frac{\partial \rho_f(\theta, t)}{\partial \theta}. \quad (5.2)$$

Note that (5.2) is of the same form as (5.1) with  $u(t) = 0$ . Since these are probability distributions, it is necessary that  $\int_0^{2\pi} \rho(\theta, t) d\theta = \int_0^{2\pi} \rho_f(\theta, t) d\theta = 1$ .

## 5.2.2 Fourier Decomposition

To devise our control laws, we use the approximation of a finite Fourier series to write the phase distributions and the PRC as

$$\rho(\theta, t) = \frac{1}{2\pi} + \sum_{l=1}^{N-1} [A_l(t) \cos(l\theta) + B_l(t) \sin(l\theta)], \quad (5.3)$$

$$\rho_f(\theta, t) = \frac{1}{2\pi} + \sum_{l=1}^{N-1} [\tilde{A}_l(t) \cos(l\theta) + \tilde{B}_l(t) \sin(l\theta)], \quad (5.4)$$

$$\mathcal{Z}(\theta) = C_0 + \sum_{l=1}^{N-1} [C_l \cos(l\theta) + S_l \sin(l\theta)]. \quad (5.5)$$

Here  $N$  is a large number, so the effect of the omitted higher order Fourier modes is negligible. Writing the distribution this way automatically ensures that the phase distribution is  $2\pi$ -periodic, and that the total probability  $\int_0^{2\pi} \rho(\theta, t) d\theta = 1$  at all times. Multiplying equation (5.3) by  $\cos(k\theta)$  and  $\sin(k\theta)$  on both sides and integrating from 0 to  $2\pi$  with respect to  $\theta$ , we obtain

$$A_k(t) = \frac{1}{\pi} \int_0^{2\pi} \rho(\theta, t) \cos(k\theta) d\theta,$$

$$B_k(t) = \frac{1}{\pi} \int_0^{2\pi} \rho(\theta, t) \sin(k\theta) d\theta.$$

Taking the derivative with respect to time of the above equations,

$$\dot{A}_k(t) = \frac{1}{\pi} \int_0^{2\pi} \frac{\partial \rho(\theta, t)}{\partial t} \cos(k\theta) d\theta = -\frac{1}{\pi} \int_0^{2\pi} \frac{\partial}{\partial \theta} [(\omega + \mathcal{Z}(\theta)u(t)) \rho(\theta, t)] \cos(k\theta) d\theta,$$

$$\dot{B}_k(t) = \frac{1}{\pi} \int_0^{2\pi} \frac{\partial \rho(\theta, t)}{\partial t} \sin(k\theta) d\theta = -\frac{1}{\pi} \int_0^{2\pi} \frac{\partial}{\partial \theta} [(\omega + \mathcal{Z}(\theta)u(t)) \rho(\theta, t)] \sin(k\theta) d\theta.$$

Integrating these equations by parts and imposing periodic boundary conditions, we obtain

$$\dot{A}_k(t) = -k\omega B_k - \mathcal{I}_{kA}(t)u(t), \quad (5.6)$$

$$\dot{B}_k(t) = k\omega A_k + \mathcal{I}_{kB}(t)u(t), \quad (5.7)$$

where

$$\mathcal{I}_{kA}(t) = \frac{k}{\pi} \int_0^{2\pi} \mathcal{Z}(\theta)\rho(\theta, t) \sin(k\theta) d\theta, \quad (5.8)$$

$$\mathcal{I}_{kB}(t) = \frac{k}{\pi} \int_0^{2\pi} \mathcal{Z}(\theta)\rho(\theta, t) \cos(k\theta) d\theta. \quad (5.9)$$

Similarly we obtain following equations for time derivatives of  $\widetilde{A}_k$  and  $\widetilde{B}_k$ :

$$\dot{\widetilde{A}}_k(t) = -k\omega \widetilde{B}_k(t), \quad (5.10)$$

$$\dot{\widetilde{B}}_k(t) = k\omega \widetilde{A}_k(t). \quad (5.11)$$

### 5.2.3 Control Design

If for all  $k$ ,  $A_k(\tau) = \widetilde{A}_k(\tau)$  and  $B_k(\tau) = \widetilde{B}_k(\tau)$ , the phase distribution  $\rho$  would be equal to the desired distribution  $\rho_f$  at time  $\tau$ . This motivates us to take our Lyapunov function as the sum of the squared differences of the Fourier coefficients of the current and the desired distribution:

$$V(t) = \frac{1}{2} \sum_{k=1}^{N-1} \left[ \left( A_k(t) - \widetilde{A}_k(t) \right)^2 + \left( B_k(t) - \widetilde{B}_k(t) \right)^2 \right]. \quad (5.12)$$

Thus the Lyapunov function is non-negative, and is zero only when  $\rho(\theta, t) = \rho_f(\theta, t)$ . Its derivative in time evolves as

$$\dot{V}(t) = I(t)u(t), \quad (5.13)$$

where  $I(t)$  is given by the sum

$$I(t) = \sum_{k=1}^{N-1} \left[ \left( B_k(t) - \widetilde{B}_k(t) \right) \mathcal{I}_{kB}(t) - \left( A_k(t) - \widetilde{A}_k(t) \right) \mathcal{I}_{kA}(t) \right]. \quad (5.14)$$

Then by taking the control input  $u(t) = -PI(t)$ , where  $P$  is a positive scalar, we get the time derivative of the Lyapunov function,  $\dot{V}(t) = -PI(t)^2$  as a negative scalar. Thus, according to the Lyapunov theorem, the control law  $u(t) = -PI(t)$  will decrease the Lyapunov function until the current probability distribution becomes equal to the desired distribution. Here we do not consider the degenerate systems where  $I(t) \equiv 0$  when  $\rho(\theta, t) \neq \rho_f(\theta, t)$  (see Section 5.3 for such a system).

For both experimental and numerical reasons, it is more practical to have a bounded control input, so we take a “clipped” proportional control law

$$u(t) = \max(\min(u_{max}, -PI(t)), u_{min}). \quad (5.15)$$

Here  $u_{max}$  and  $u_{min}$  are the upper and lower bounds of the control input, respectively. The max, and min operators find the maximum and minimum of two scalars, respectively.

### 5.3 Degenerate Set

Note that for certain systems where  $\rho(\theta, t) \neq \rho_f(\theta, t)$ , equation (5.14) gives  $I(t) \equiv 0$  for all time  $t$ , and the probability distribution  $\rho(\theta, t)$  would not converge to the desired distribution  $\rho_f(\theta, t)$ . Here we derive the set of such phase distributions and PRCs that

leads to this degeneracy, and give an example of such a system.

We can re-write  $I(t)$  as

$$\begin{aligned} I(t) &= \frac{1}{\pi} \int_0^{2\pi} \sum_{k=1}^{N-1} k \left[ \left( B_k(t) - \widetilde{B}_k(t) \right) \cos(k\theta) - \left( A_k(t) - \widetilde{A}_k(t) \right) \sin(k\theta) \right] \mathcal{Z}(\theta) \rho(\theta, t) d\theta \\ &= \frac{1}{\pi} \int_0^{2\pi} \left( \frac{\partial \rho}{\partial \theta} - \frac{\partial \rho_f}{\partial \theta} \right) \mathcal{Z}(\theta) \rho(\theta, t) d\theta. \end{aligned} \quad (5.16)$$

Now expanding  $\rho(\theta, t)$ ,  $\rho_f(\theta, t)$ , and  $\mathcal{Z}(\theta)$  into their complex Fourier series,

$$\begin{aligned} \rho(\theta, t) &= \sum_{k=1-N}^{N-1} a_k(t) \exp(ik\theta), & \rho_f(\theta, t) &= \sum_{k=1-N}^{N-1} \widetilde{a}_k(t) \exp(ik\theta), \\ \mathcal{Z}(\theta) &= \sum_{k=1-N}^{N-1} c_k \exp(ik\theta), \end{aligned}$$

where

$$\begin{aligned} a_{\pm k}(t) &= \frac{A_k(t) \mp iB_k(t)}{2}, & \widetilde{a}_{\pm k}(t) &= \frac{\widetilde{A}_k(t) \mp i\widetilde{B}_k(t)}{2}, \\ c_{\pm k} &= \frac{C_k \mp iS_k}{2}, & k &= 1, \dots, N-1, \\ a_0(t) &= A_0(t), & \widetilde{a}_0(t) &= \widetilde{A}_0(t), & c_0(t) &= C_0(t), \end{aligned}$$

we can write  $I(t)$  from equation (5.16) as

$$I(t) = \sum_{p=1-N}^{N-1} \sum_{q=1-N}^{N-1} \sum_{r=1-N}^{N-1} \left[ ip(a_p(t) - \widetilde{a}_p(t)) c_q a_r(t) \frac{1}{\pi} \int_0^{2\pi} \exp(i(p+q+r)\theta) d\theta \right]. \quad (5.17)$$

Thus the degenerate set of phase distributions and PRCs can be written in terms of their respective Fourier coefficients as

$$\sum_{p \in M} \sum_{q=1-N}^{N-1} \sum_{r=1-N}^{N-1} [i2p(a_p(t) - \widetilde{a}_p(t)) c_q a_r(t) \delta_{p+q+r,0}] \equiv 0 \quad (5.18)$$

for all time  $t$ , where  $M$  is the subset of integers ranging from  $1 - N$  to  $N - 1$  for which  $a_p(t) \neq \tilde{a}_p(t)$ , and  $\delta_{p+q+r,0}$  is the Kronecker delta, which is equal to 1  $\forall p + q + r = 0$ , and is 0 otherwise.

### 5.3.1 Degenerate System Example

As an example degenerate system, we consider the Type I PRC near a SNIPER bifurcation given by [18]

$$\mathcal{Z}(\theta) = \frac{2}{\omega} (1 - \cos(\theta)).$$

Thus  $c_0 = 2/\omega$ ,  $c_{\pm 1} = 1/\omega$ , while rest of the PRC Fourier coefficients are 0. We take the desired distribution as a uniform distribution,

$$\rho_f(\theta, t) = \frac{1}{2\pi}.$$

Thus  $\tilde{a}_0(t) = 1/2\pi$ , while rest of the Fourier coefficients for  $\rho_f$  are 0 for all times. For the degenerate set,  $I \equiv 0$ , and thus  $\rho(\theta, t)$  is a traveling wave moving in the  $+\theta$  direction.

We take it as

$$\rho(\theta, t) = \frac{\sin^2(\theta - \omega t)}{\pi}.$$

It is a physically realistic distribution since  $\rho(\theta, t) \geq 0$ , and  $\int_0^{2\pi} \rho(\theta, t) d\theta = 1$ . Thus  $a_0(t) = 1/2\pi$ ,  $a_{\pm 2}(t) = -\exp(\mp i2\omega t)/4\pi$ , while rest of its Fourier coefficients are 0.

There are only two nonzero cases to consider in the summation of the degenerate set

(equation (5.18)):

$$\begin{aligned}
 p = 2, q = 0, r = -2; \quad & i2(2) \left( -\frac{\exp(-i2\omega t)}{4\pi} - 0 \right) \left( \frac{2}{\omega} \right) \left( -\frac{\exp(i2\omega t)}{4\pi} \right) = \frac{i}{2\omega\pi^2}, \\
 p = -2, q = 0, r = 2; \quad & -i2(2) \left( -\frac{\exp(i2\omega t)}{4\pi} - 0 \right) \left( \frac{2}{\omega} \right) \left( -\frac{\exp(-i2\omega t)}{4\pi} \right) = -\frac{i}{2\omega\pi^2}, \\
 & I(t) = \frac{i}{2\omega\pi^2} - \frac{i}{2\omega\pi^2} = 0.
 \end{aligned}$$

Thus as  $I$  is zero even though the phase distributions are not equal, this is a degenerate system. This can also be verified by analytically evaluating the integral in equation (5.16) to be zero, i.e.,

$$\begin{aligned}
 I(t) &= \frac{4}{\omega\pi^3} \int_0^{2\pi} \sin^3(\theta - \omega t) \cos(\theta - \omega t) (1 - \cos \theta) d\theta \\
 &= \frac{4}{\omega\pi^3} \left[ \frac{\cos(\theta - 2\omega t) - \cos(2\theta - 2\omega t)}{8} + \frac{\cos(3\theta - 2\omega t)}{24} - \frac{\cos(3\theta - 4\omega t)}{48} \right. \\
 &\quad \left. + \frac{\cos(4\theta - 4\omega t)}{32} - \frac{\cos(5\theta - 4\omega t)}{80} + \frac{3}{32} \right] \Big|_0^{2\pi} = 0.
 \end{aligned}$$

Note that such degeneracy arises due to the inherent simplicity and symmetry present in the system under consideration, and thus should not be considered a limitation of the devised control law. “Real world” systems would have more than two Fourier modes and some sort of asymmetry, which would avoid degeneracy.

## 5.4 Numerical Methods

Here we give details on the numerical methods we use for the simulation results that we present in this chapter. Since we are dealing with periodic probability distributions in  $\theta$ , we take the initial (and, later, the desired distributions) as a von Mises distribution

[106]

$$\rho(\theta, 0) = \frac{e^{\kappa \cos(\theta + \theta_0)}}{2\pi \mathcal{I}_0(\kappa)}, \quad (5.19)$$

with  $\mathcal{I}_0(\kappa)$  the modified Bessel function of first kind of order 0. For such a distribution, the mean is  $\theta_0$ , and the variance is  $1 - \mathcal{I}_1(\kappa)/\mathcal{I}_0(\kappa)$ , where  $\mathcal{I}_1(\kappa)$  is the modified Bessel function of first kind of order 1. The variance decreases as  $\kappa$  increases, and so the distribution becomes narrower and taller. To demonstrate the effectiveness of our control algorithm, we apply the control input given by equation (5.15) to a population of 100 phase oscillators

$$\dot{\Theta}_i(t) = \omega + \mathcal{Z}(\Theta_i(t))u(t), \quad i = 1, 2, \dots, 100. \quad (5.20)$$

For the case where initial distribution  $\rho(\theta, 0)$  is a uniform distribution ( $\kappa = 0$ ), we take the initial value of phase oscillators  $\Theta_i(0) = 2\pi(i - 1)/100$ , and for a non-zero  $\kappa$ , we use the command `randdraw('vonmises', [Theta_0, kappa], 100)` from the circular statistical toolbox developed for Matlab in [107] to initialize the phase oscillators corresponding to a von Mises distribution (equation (5.19)).

We discretize  $\theta$  into a uniform mesh with  $2N = 128$  grid points. We choose this grid size for a good spatial resolution of the probability distribution, and efficient computation of the fast Fourier transform algorithm. For computing the PRCs of the various models presented in next section, we use the XPP package [44] with a time step  $T/N$ . We scale the PRC computed by this package by  $\omega$ , as we consider PRC as  $\mathcal{Z}(\theta) = \frac{\partial \theta}{\partial \mathbf{x}}$ , whereas the computed PRC from the XPP package is  $\tilde{\mathcal{Z}}(t) = \frac{\partial t}{\partial \mathbf{x}}$  [18, 38]. Then we use the Matlab command `fft` to compute the Fourier coefficients of the initial distribution  $\rho(\theta, 0)$ . Note

that the `fft` command computes coefficients of the complex Fourier series given as

$$\rho(\theta, 0) = \sum_{k=1-N}^N a_k(0) \exp(ik\theta),$$

giving an output  $[a_0, a_1(0), \dots, a_N(0), a_{1-N}(0), a_{2-N}(0), \dots, a_{-1}(0)] \times 2N$ . From these coefficients, we then compute the coefficients of the real Fourier series

$$\rho(\theta, 0) = A_0 + \sum_{k=1}^{N-1} A_k(0) \cos(k\theta) + B_k(0) \sin(k\theta),$$

as

$$\begin{aligned} A_0 &= a_0, \\ A_k(0) &= (a_k(0) + a_{-k}(0)), \\ B_k(0) &= i(a_k(0) - a_{-k}(0)). \end{aligned}$$

The same procedure is adopted to compute real Fourier coefficients of  $\rho_f(\theta, 0)$  ( $\widetilde{A}_k(0)$ ,  $\widetilde{B}_k(0)$ ) and the PRC ( $C_k$ ,  $S_k$ ).

To evolve these coefficients over time, ODEs given by equations (5.6), (5.7), (5.10), (5.11) are evolved in time using a fourth order Runge-Kutta method with a fixed time step  $dt = T/(8N)$ . In order to maintain spectral accuracy, the integrals given by equations (5.8)-(5.9) are evaluated in Fourier space, i.e., we take the FFT of the integrand using Matlab command `fft`, and divide the first term of FFT by  $N$  to get the numerical value of the integral at every time step.

## 5.5 Applications

In this section, we apply the control law devised in the previous section to manipulate the population density of uncoupled noise-free oscillators to achieve control objectives in a diversity of applications. These applications are desynchronizing an initially synchronized neuron population for the treatment of parkinsonian and essential tremor, forming neuron clusters from an initial desynchronized neuron population to maximize neural plasticity, and eliminating cardiac alternans by phase shifting a synchronized population of cardiac pacemaker cells. For all these applications, we consider underactuated dynamical systems with only one degree of actuation: the control input vector is  $U(t) = [u(t), 0, \dots, 0]^T$ . We make this assumption because in most conductance-based models of neurons and cardiac pacemaker cells, we can only give a single control input in the form of a current to one of the elements of the state vector, which corresponds to the voltage across the cell membrane.

### 5.5.1 Desynchronizing Neurons

Parkinsonian and essential tremor affect millions of people worldwide, causing involuntary tremors in various parts of the body, and disrupting the activities of daily living. Pathological neural synchronization in the STN and the thalamus brain region is hypothesized to be one of the causes of motor symptoms of parkinsonian and essential tremor, respectively [12, 11]. Deep brain stimulation (DBS), an FDA approved treatment, has proven to alleviate these symptoms [86, 85] by stimulating the STN or the thalamus brain regions with a high frequency, (relatively) high energy pulsatile waveform, which has been hypothesized to desynchronize the synchronized neurons; see, e.g., [87, 15]. This has motivated researchers to come up with efficient model dependent control techniques [88, 67, 21] which not only desynchronize the neurons but also consume less energy,

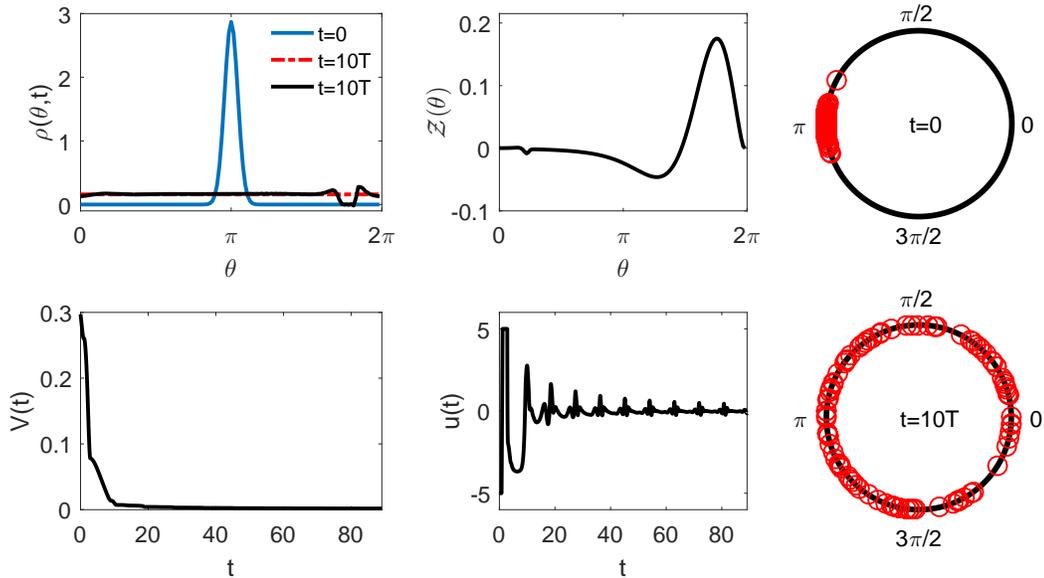


Figure 5.1: Desynchronizing Control: In the top left panel, the solid (resp., red dashed) lines show the probability distribution  $\rho(\theta, t)$  (resp.,  $\rho_f(\theta, t)$ ) at various times. The top middle panel shows the PRC, while the bottom left and middle panels show the Lyapunov function  $V(t)$  (5.12), and the control input, respectively. The top (resp., bottom) right panels show 100 phase oscillators at time  $t = 0$  ms (resp.,  $t = 10T$  ms). Here  $T = 8.91$  ms.

thus prolonging the battery life of the stimulator and preventing tissue damage or side effects caused by the pulsatile stimuli.

Thus, inspired by this treatment of parkinsonian and essential tremor, we employ our algorithm to desynchronize an initially synchronized population of neurons. Here we use our algorithm to change the probability distribution of synchronized neurons with mean  $\pi$  and  $\kappa = 52$ , into a uniform distribution ( $\kappa = 0$ ). As a proof of concept, we use the two-dimensional reduced Hodgkin-Huxley model [4, 108] for calculating the PRC. For details of this model, see Appendix A.4. Under zero control input, this model gives a stable periodic orbit with time period  $T = 8.91$ ms. The top middle panel of Figure 5.1 shows the corresponding PRC. We take the control parameters  $P = 1000$ ,  $u_{min} = -5$ ,  $u_{max} = 5$ , and simulate until  $t = 10T$ . From the top and bottom left panels of Figure 5.1, we see

that the control input is able to flatten out the bell shaped probability distribution, and thus decrease the Lyapunov function towards zero. For  $t > 10 \text{ ms}$ , we see that the decay rate of Lyapunov function decreases, and thus the Lyapunov function asymptotically decreases towards zero. This can be explained from equations (5.13 - 5.15) where we see that control input (decay rate of Lyapunov function) depends on the (square of the) difference between current coefficients and desired coefficients. Thus as the coefficients get closer to their desired value, the magnitude of the control input decreases significantly, which decreases the rate of decay of the Lyapunov function. The top right panel of Figure 5.1 shows 100 phase oscillators synchronized with mean  $\pi$ , and  $\kappa = 52$  extracted through the Matlab circular statistical toolbox. We apply the control input from the middle bottom panel of Figure 5.1 to them in an open loop manner. The bottom right panel of the same figure shows the same oscillators at time  $t = 10T$ . We see that the control input is able to desynchronize these oscillators almost perfectly. In transforming the probability distribution, the total control energy consumed ( $\int_0^{10T} u^2 dt$ ) comes out to be 141.78 units.

### 5.5.2 Clustering Neurons for Maximizing Neural Plasticity

An adult human brain is composed of billions of neurons, and each of these neurons is connected to other neurons. Neural plasticity is a significant factor in forming specific connections by wiring neurons that fire together [109]. Spike time dependent plasticity (STDP) is one type of long-term plasticity, which wires neurons that fire together over a long period of time, thus helping in the regulation of neural synchrony. However, increased neural synchrony is a hallmark of several neurological disorders as discussed in the previous section, and STDP can resynchronize a desynchronized neural population over time in the presence of noise [110]. Thus, desynchronizing control, as considered

in the previous section, may not be the best long-term solution. Recent results [15] suggest another hypothesis that DBS works by forming neural clusters instead of complete desynchronization. Coordinated Reset, a method which promotes clustering, has shown to have long lasting effects even after the control stimulus is turned off [88, 111]. This further motivates clustering as an alternative desynchronizing strategy for the treatment of parkinsonian and essential tremor. This would not only reduce neural synchrony but also promote clustering over long periods of time by re-wiring of neuron connections through STDP. We demonstrate this by first defining the phase difference distribution, and then the STDP curve.

### Phase Difference Distribution

Given a phase distribution  $\rho(\theta, t)$  governing the probability of a population of oscillators at phase  $\theta$  and time  $t$ , a corresponding phase difference distribution  $\rho_d(\phi, t)$  governs the probability that the phase difference between any two set of oscillators in the population is  $\phi$  at time  $t$ , where  $\phi \in [0, 2\pi)$ . We only consider uncoupled oscillators which evolve independently from each other in this chapter. Thus the probability that the phase difference between any two oscillators is  $\phi$  at time  $t$  can be given by the integral of the products of the phase distribution and the phase distribution shifted by  $\phi$  at times  $t$  over the entire domain:

$$\rho_d(\phi, t) = \int_0^{2\pi} \rho(\theta_s, t) \rho(\theta_s + \phi, t) d\theta_s. \quad (5.21)$$

The phase difference distribution satisfies

$$\int_0^{2\pi} \rho_d(\phi, t) d\phi = 1. \quad (5.22)$$

This can be shown from equation (5.21):

$$\begin{aligned}
\int_0^{2\pi} \rho_d(\phi, t) d\phi &= \int_0^{2\pi} \left[ \int_0^{2\pi} \rho(\theta_s, t) \rho(\theta_s + \phi, t) d\theta_s \right] d\phi \\
&= \int_0^{2\pi} \left[ \int_0^{2\pi} \rho(\theta_s + \phi, t) d\phi \right] \rho(\theta_s, t) d\theta_s \\
&= \int_0^{2\pi} 1 \cdot \rho(\theta_s, t) d\theta_s \\
&= 1.
\end{aligned}$$

Note that phase difference distribution for a time-dependent traveling wave  $\rho_f(\theta, t)$  governed by equation (5.2), is stationary and does not depend on time. This can be proven by taking the time derivative of equation (5.21):

$$\begin{aligned}
\frac{d\rho_d}{dt} &= \int_0^{2\pi} \left[ \frac{\partial \rho_f(\theta_s, t)}{\partial t} \rho_f(\theta_s + \phi, t) + \rho_f(\theta_s, t) \frac{\partial \rho_f(\theta_s + \phi, t)}{\partial t} \right] d\theta_s \\
&= -\omega \int_0^{2\pi} \left[ \frac{\partial \rho_f(\theta_s, t)}{\partial \theta_s} \rho_f(\theta_s + \phi, t) + \rho_f(\theta_s, t) \frac{\partial \rho_f(\theta_s + \phi, t)}{\partial \theta_s} \right] d\theta_s \\
&= -\omega \rho_f(\theta_s, t) \rho_f(\theta_s + \phi, t) \Big|_0^{2\pi} + \omega \int_0^{2\pi} \rho_f(\theta_s, t) \frac{\partial \rho_f(\theta_s + \phi, t)}{\partial \theta_s} d\theta_s \\
&\quad - \omega \int_0^{2\pi} \rho_f(\theta_s, t) \frac{\partial \rho_f(\theta_s + \phi, t)}{\partial \theta_s} d\theta_s \\
&= 0.
\end{aligned}$$

Here, the first equality follows from the Leibniz rule from elementary calculus, and the third equality follows from the previous line by applying integration by parts and imposing periodic boundary conditions. Thus, this proves that the phase difference distribution for a time-dependent traveling wave is independent of time. For such a traveling wave phase distribution, we write the phase difference distribution as being independent of time:

$$\rho_d(\phi) = \int_0^{2\pi} \rho_f(\theta_s, t) \rho_f(\theta_s + \phi, t) d\theta_s. \quad (5.23)$$

The Fourier coefficients for the phase difference distribution can be calculated as follows:

$$\begin{aligned} \rho_d(\phi) = & \int_0^{2\pi} \left( \frac{1}{2\pi} + \sum_{k=1}^{N-1} \left[ \widetilde{A}_k(t) \cos(k\theta_s) + \widetilde{B}_k(t) \sin(k\theta_s) \right] \right) \\ & \times \left( \frac{1}{2\pi} + \sum_{l=1}^{N-1} \left[ \widetilde{A}_l(t) \cos(l(\theta_s + \phi)) + \widetilde{B}_l(t) \sin(l(\theta_s + \phi)) \right] \right) d\theta_s. \end{aligned}$$

By expanding  $\cos(l(\theta_s + \phi))$  and  $\sin(l(\theta_s + \phi))$ , and making use of the orthogonality of  $\cos k\theta_s$  and  $\sin k\theta_s$ , we obtain

$$\rho_d(\phi) = \frac{1}{2\pi} + \pi \sum_{k=1}^{N-1} \left( \widetilde{A}_k^2(t) + \widetilde{B}_k^2(t) \right) \cos(k\phi). \quad (5.24)$$

From this formulation of the phase difference distribution in terms of the Fourier coefficients of the desired phase distribution, one can easily verify that  $\rho_d(\phi)$  is  $2\pi$ -periodic,  $\int_0^{2\pi} \rho_d(\phi) d\phi = 1$ , and  $\rho_d(\phi)$  is independent of time, which can be seen by taking the time derivative of equation (5.24):

$$\begin{aligned} \dot{\rho}_d(\phi) &= \pi \sum_{k=1}^{N-1} 2 \left( \widetilde{A}_k(t) \dot{\widetilde{A}}_k(t) + \widetilde{B}_k(t) \dot{\widetilde{B}}_k(t) \right) \cos(k\phi) \\ &= \pi \sum_{k=1}^{N-1} 2k\omega \left( -\widetilde{A}_k(t) \widetilde{B}_k(t) + \widetilde{A}_k(t) \widetilde{B}_k(t) \right) \cos(k\phi) \\ &= 0. \end{aligned}$$

Another property that the phase difference distribution has is that it always has a local maximum at zero phase difference. This can easily be verified from equation (5.24), as  $\frac{d\rho_d(0)}{d\phi} = 0$  and  $\frac{d^2\rho_d(0)}{d\phi^2} < 0$ .

## Spike Time Dependent Plasticity Stabilizes Clusters

STDP is an asymmetric form of Hebbian learning [112] that modifies synaptic connections between neurons when they fire repeatedly in a causal manner [113, 114, 115]. At the single synapse level, STDP potentiates (resp., depresses) the synaptic strength for repeated pre-synaptic action potentials arriving just before (resp., after) the post-synaptic action potential. At the population level, STDP strengthens the synaptic connections between neurons that fire action potentials synchronously and weakens those connections in the out of phase neurons [110]. Plasticity is known to be an important factor in the formation of neural pathways in initial brain development, as well as later in learning and memory storage. Since we consider uncoupled oscillating neurons in this chapter, we reformulate STDP in terms of the phase difference  $\phi$  between two neurons instead of their spike time difference; the distribution of interspike intervals is same as the phase difference distribution for uncoupled oscillating neurons. If the phase difference  $\phi \in [0, \pi)$ , the STDP would increase the synaptic weight, and if the phase difference  $\phi \in [\pi, 2\pi)$ , STDP would depress the synaptic weight. We call this increase or decrease of synaptic weights as a function of phase difference the *STDP curve* given as

$$S(\phi) = \begin{cases} pe^{-\frac{\phi}{\tau_p}}, & \phi \in [0, \pi) \\ -de^{\frac{\phi-2\pi}{\tau_d}}, & \phi \in [\pi, 2\pi) \end{cases}. \quad (5.25)$$

We take the parameters  $p = 0.0096$ ,  $d = 0.0053$  from [114], while  $\tau_p = 0.2$ ,  $\tau_d = 0.365$  are taken so that the integral of the resulting STDP curve ( $\int_0^{2\pi} S(\phi)d\phi$ ) is zero [116]. The top left panel of Figure 5.2 shows the STDP curve  $S(\phi)$  with the above parameters.

Let us suppose that we start with a desynchronized population. The average rate of

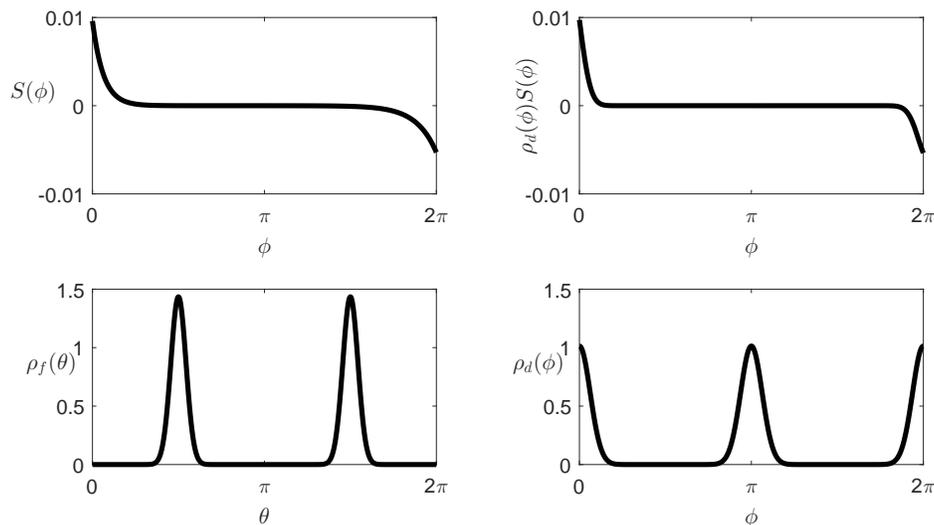


Figure 5.2: The top left panel shows the spike time dependent plasticity curve  $S(\phi)$ . The bottom left (resp., right) panel shows the desired phase (resp., phase difference) distribution. The top right panel shows the change in synaptic weight between two neurons as a function of their phase difference.

synaptic connection change between any two neurons in the population is given by [110]

$$\Delta c = \int_0^{2\pi} \rho_d(\phi) S(\phi) d\phi. \quad (5.26)$$

A uniform phase distribution (desynchronized population) would result in a uniform phase difference distribution, which would lead to a zero average synaptic change. On the other hand, if we promote neural clustering, STDP would potentiate intra-cluster synaptic connections and depress inter-cluster connections. This would thus potentially help in long-term stabilizability of clusters in the presence of noise. We demonstrate this by taking two clusters and calculating the average synaptic change (5.26) intra- and inter-cluster. Thus we take the desired phase distribution as a bi-modal distribution,

which can be realized as a sum of two uni-modal von Mises distributions

$$\rho_f(\theta, t) = \frac{e^{\kappa \cos(\theta + \pi/2)} + e^{\kappa \cos(\theta + 3\pi/2)}}{4\pi\mathcal{I}_0(\kappa)}, \quad (5.27)$$

where  $\kappa = 52$ . From this we calculate the phase difference distribution from equation (5.23) or (5.24), which can then be used to calculate the average synaptic change from equation (5.26). The bottom left, right, and top right panels of Figure 5.2 show the desired phase distribution, phase difference distribution, and the product of the phase difference distribution with the STDP curve, respectively. The average synaptic change for intra- and inter-cluster is calculated as

$$\Delta c_{intra-cluster} = \int_0^{\pi/2} \rho_d(\phi)S(\phi)d\phi + \int_{\frac{3\pi}{2}}^{2\pi} \rho_d(\phi)S(\phi)d\phi = 3.62 \times 10^{-4}, \quad (5.28)$$

$$\Delta c_{inter-cluster} = \int_{\frac{\pi}{2}}^{\frac{3\pi}{2}} \rho_d(\phi)S(\phi)d\phi = -3.96 \times 10^{-7}. \quad (5.29)$$

Thus, STDP would strengthen synapse in each cluster and weaken them between the two clusters, thereby potentially maintaining clusters over a long period of time. This motivates us to transform an initially desynchronized phase distribution ( $\kappa = 0$ ) into a bi-modal phase distribution (5.27). As a proof of concept, here we again use the two-dimensional reduced Hodgkin-Huxley model for calculating the PRC. We take the control parameters  $P = 1200$ ,  $u_{min} = -15$ ,  $u_{max} = 15$ , and simulate until  $t = 15T$ . The results are shown in Figure 5.3. From the top and bottom left panels of Figure 5.3, we see that the control input is able to transform an initial uniform distribution into a bi-modal distribution, and thus the Lyapunov function decreases towards zero. As the Fourier coefficients of the current and desired distribution get closer, the control input decreases in magnitude, which decreases the rate of decay of the Lyapunov function. The top right panel of Figure 5.3 shows 100 desynchronized phase oscillators ( $\Theta_i(0) = 2\pi(i - 1)/100$ )

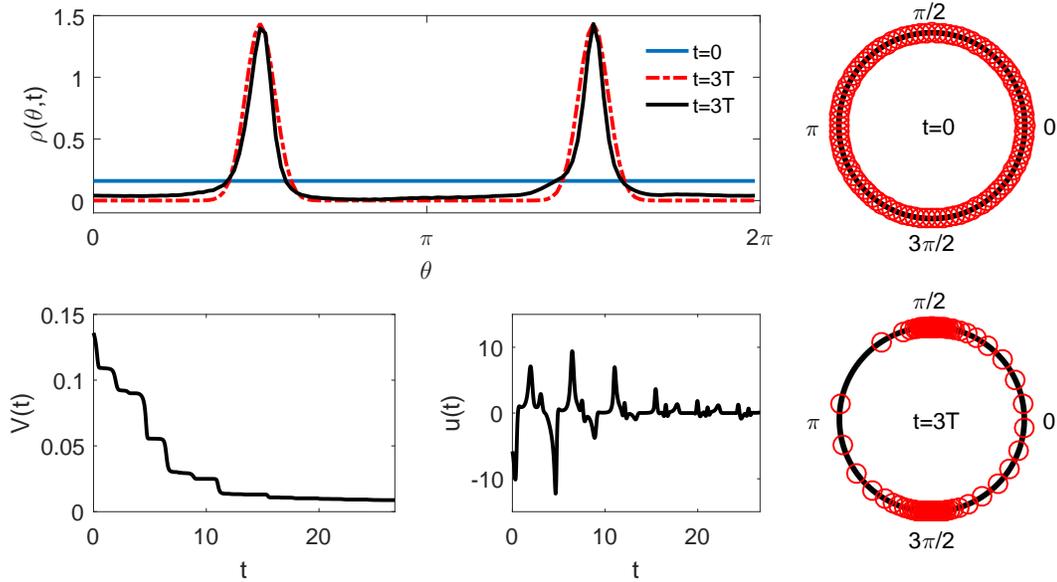


Figure 5.3: Clustering Control: In the top left panel, the solid (resp., red dashed) lines show the probability distribution  $\rho(\theta, t)$  (resp.,  $\rho_f(\theta, t)$ ) at various times. The bottom left and middle panels show the Lyapunov function  $V(t)$  (5.12), and the control input, respectively. The top (resp., bottom) right panels show 100 desynchronized (resp., clustered) phase oscillators at time  $t = 0$  ms (resp.,  $t = 3T$  ms). Here  $T = 8.91$  ms.

to which the control input from the middle bottom panel of Figure 5.3 is applied in an open loop manner. The bottom right panel of the same figure shows the oscillators at time  $t = 3T$ . We see that the control input is able to separate the desynchronized oscillators into two distinct clusters corresponding to the bi-modal phase distribution. In transforming the probability distribution, the total control energy consumed ( $\int_0^{3T} u^2 dt$ ) comes out to be 152.59 units.

### 5.5.3 Eliminating Cardiac Alternans

The collection of cells in the Sinoatrial node (SA node) called cardiac pacemaker cells elicit periodic electrical pulses which polarize a collection of excitable and contractile cells called myocytes. In the process of depolarizing, myocytes contract and propagate

action potentials to the neighboring cells. This well-coordinated process of excitation / depolarization and contraction enables the heart to pump blood throughout the body. Under normal conditions, with constant pacing by the cardiac pacemaker cells, the action potential duration (APD), that is the time for which an action potential lasts in a myocyte cell, also remains constant. However, under some conditions, this 1:1 rhythm between pacing and the APD can become unstable, bifurcating into a 2:2 rhythm of alternating long and short APD, known as alternans [75]. Alternans is observed to be a possible first step leading to fibrillation [76]. Thus, a number of researchers have worked on suppressing alternans as a method of preventing fibrillation, thereby preventing the need for painful and damaging defibrillating shocks. Many of these methods [77, 78, 79, 80] operate by exciting the myocardium tissue externally with periodic pulses, and changing the period according to the alternating rhythm. However, such a control requires excitation at several sites in the tissue [81].

In Chapter 4 (cf. [16]), we developed a novel strategy to suppress alternans by changing the phase of the pacemaker cells. The control strategy was based on a single oscillator model to change the phase of a single cell. However for an effective cardiac alternans elimination, we need to consider the entire population of cardiac pacemaker cells which oscillate in synchrony. So, here we aim to phase shift the population of cardiac pacemaker cells using the control algorithm we developed in Section 5.2.3. Such a control strategy could eliminate the need to excite the tissue at multiple sites. The amount of phase change required to eliminate alternans depends on the discrete APD dynamics [16]. Here we advance the phase by  $\pi/4$  as an example. For the PRC calculation, we consider phase reduction of the 7-dimensional YNI model of SA node cells in rabbit heart proposed in [84]. The model is of Hodgkin-Huxley type with 6 gating variables and a transmembrane voltage variable on which the control input acts. For details of the model, see Appendix A.2. With this model we get a stable periodic orbit with time period

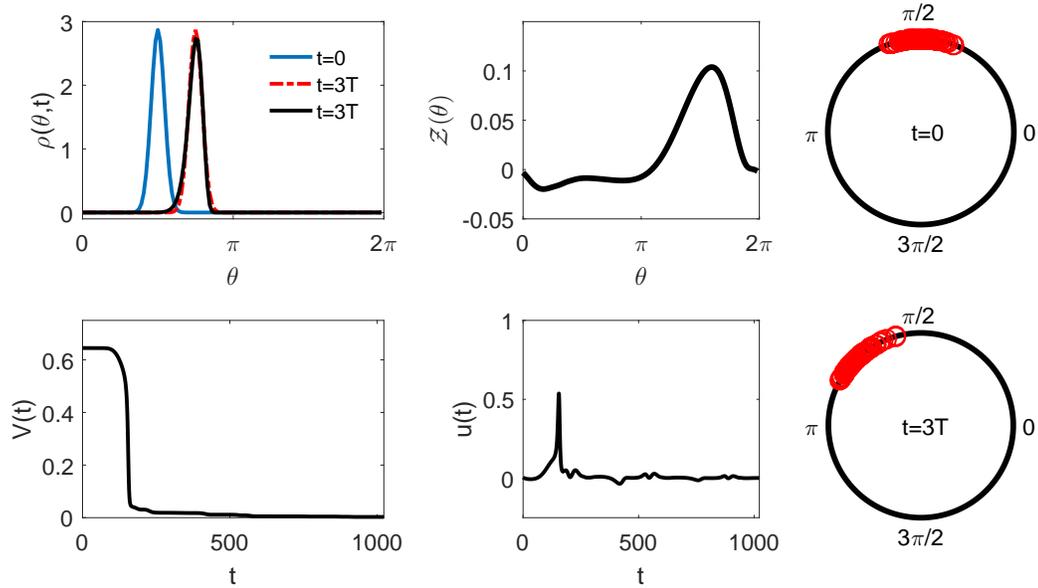


Figure 5.4: Phase shifting cardiac pacemaker cells: In the top left panel, the solid (resp., red dashed) lines show the probability distribution  $\rho(\theta, t)$  (resp.,  $\rho_f(\theta, t)$ ) at various times. The top middle panel shows the PRC, while the bottom left and middle panels show the Lyapunov function  $V(t)$  (5.12), and the control input, respectively. The top (resp., bottom) right panels show 100 phase oscillators at time  $t = 0$  ms (resp.,  $t = 3T$  ms). Here  $T = 340.8$  ms. Note that in the absence of control input, the oscillators would have a mean of  $\pi/2$  at  $t = 3T$ .

$T = 340.8$  ms. We start with a synchronous population distribution with mean  $\pi/2$  and  $\kappa = 52$ . In order to phase shift this distribution by  $\pi/4$ , we take the target population distribution  $\rho_f(\theta, t)$  with same  $\kappa$  value but an initial mean of  $3\pi/4$ . Thus, our control algorithm will push the distribution  $\rho(\theta, t)$  forward until it matches with the desired distribution  $\rho_f(\theta, t)$ . We take the control parameters  $P = 5$ ,  $u_{min} = -1$ ,  $u_{max} = 1$  and apply control input until  $t = 3T$ . From the top and bottom right panels of Figure 5.4, we see that the control input is able to phase shift the probability distribution, and thus decreases the Lyapunov function towards zero. In doing so, it changes the shape of the distribution slightly. The top right panel of Figure 5.4 shows 100 phase oscillators synchronized with mean  $\pi/2$ , and  $\kappa = 52$  extracted through the Matlab circular statistical

toolbox. We apply the control input from the middle bottom panel of the figure to those in an open loop manner. The bottom right panel of the same figure shows the oscillators at time  $t = 3T$ . We see that the control input is able to phase shift these oscillators by  $\pi/4$ . The slight change in shape of the phase distribution is reflected in the final position of phase oscillators, where handful of the oscillators get spread relative to the main cluster. In shifting the phase of the probability distribution, the total control energy consumed ( $\int_0^{3T} u^2 dt$ ) comes out to be 3.21 units.

Here we mention another application for which shifting the phase of an oscillator population is desired: phase shifting circadian oscillators for the treatment of jet lag. Neurons in the suprachiasmatic nucleus (SCN) of the brain are responsible for maintaining the circadian rhythm in mammals. This rhythm is synchronized with the external day and night cycle under normal conditions. A disruption between these two rhythms can happen due to multiple reasons, such as travel across time zones, starting a night shift job, working in extreme environments (space, earth poles, underwater), etc. Such asynchrony can lead to several physiological disorders [89, 90], thus motivating researchers to try to develop ways to remove it. In Chapter 4 (cf. [16]), we developed a strategy to eliminate this asynchrony by changing the phase of a single SCN neuron by using a light stimulus as the control input, since light is known to affect the circadian rhythm [91]. This would change the phase of the circadian rhythm so that it gets aligned with the external cycle after the end of the controlled oscillation. However for a better alignment of the circadian rhythm with the external environment, we need to phase shift the entire population of SCN neurons which oscillate in synchrony, which can be achieved by our control algorithm. This is very similar to the previous application of phase shifting cardiac pacemaker cells.

## 5.6 Addition of White Noise

So far we have demonstrated that our control is effective for a population of uncoupled, noise-free oscillators. However, real systems are subjected to noise; thus, in this section we modify our control to take this into account.

Given  $M$  noisy, uncoupled oscillators with dynamics given by

$$\frac{d\mathbf{x}_j}{dt} = F(\mathbf{x}_j) + \left[ u(t) + \sqrt{2D}\eta_j(t), 0, \dots, 0 \right]^T, \quad j = 1, \dots, M. \quad (5.30)$$

Here each oscillator receives a common input  $u(t)$  modified by a different realization of Gaussian white noise  $\sqrt{2D}\eta_j(t)$  with zero mean, variance  $2D$ , and with  $\langle \eta_i(t)\eta_j(t') \rangle = \delta_{ij}\delta(t-t')$ . Letting  $\theta_j$  be the phase of the  $j^{\text{th}}$  oscillator, to leading order in the noise strength Ito's formula gives [117]

$$\dot{\theta}_j = \omega + \mathcal{Z}(\theta) \left[ u(t) + \sqrt{2D}\eta_j(t) \right], \quad j = 1, \dots, M. \quad (5.31)$$

Assuming  $M$  is large and noise perturbations are small, the population dynamics can be represented in terms of its phase distribution  $\rho(\theta, t)$  with stochastic averaging [118, 103]:

$$\frac{\partial \rho(\theta, t)}{\partial t} = -\frac{\partial}{\partial \theta} [(\omega + \mathcal{Z}(\theta)u(t)) \rho(\theta, t)] + \frac{B^2}{2} \frac{\partial^2 \rho(\theta, t)}{\partial \theta^2}, \quad (5.32)$$

where

$$B^2 = \frac{2D}{2\pi} \int_0^{2\pi} \mathcal{Z}^2(\theta) d\theta.$$

As before, the desired final probability distribution  $\rho_f(\theta, t)$  is taken to be a traveling wave which evolves according to equation (5.2). To devise our control laws, we use the approximation of a finite Fourier series to write the phase distributions (equations (5.3), and (5.4)). The Fourier coefficients of the desired phase distribution evolve as before

(equations (5.10), and (5.11)), whereas the Fourier coefficients of phase distribution evolve as

$$\dot{A}_k(t) = -k\omega B_k - \mathcal{I}_{kA}(t)u(t) - \frac{B^2}{2}k^2 A_k(t), \quad (5.33)$$

$$\dot{B}_k(t) = k\omega A_k + \mathcal{I}_{kB}(t)u(t) - \frac{B^2}{2}k^2 B_k(t). \quad (5.34)$$

### 5.6.1 Control Design

Here as well we take the Lyapunov function as the sum of squared differences of the Fourier coefficients of the current and the desired distributions (equation (5.12)). Its derivative in time evolves as

$$\dot{V}(t) = I(t)u(t) + G(t), \quad (5.35)$$

where

$$G(t) = -\frac{B^2}{2} \sum_{k=1}^{N-1} k^2 \left[ A_k(t) \left( A_k(t) - \widetilde{A}_k(t) \right) + B_k(t) \left( B_k(t) - \widetilde{B}_k(t) \right) \right],$$

and  $I(t)$  is given by equation (5.14). Then by taking the control input

$$u(t) = -PI(t) - \frac{G(t)}{I(t)}, \quad (5.36)$$

where  $P$  is a positive scalar, we get the time derivative of the Lyapunov function to be a negative scalar. Thus, according to the Lyapunov theorem, the control law (5.36) will decrease the Lyapunov function until the current probability distribution becomes equal to the desired distribution. Here we do not consider the degenerate case where  $I(t) \equiv 0$  when  $\rho(\theta, t) \neq \rho_f(\theta, t)$  (see Section 5.3 for cases when  $I(t) \equiv 0$  when  $\rho(\theta, t) \neq \rho_f(\theta, t)$ ).

### 5.6.2 Simulation Results

To demonstrate our control in the presence of noise, we use (5.36) to transform an initial uniform phase distribution into a desired bi-modal distribution (5.27). We take the noise strength  $\sqrt{2D} = 0.03$  in equations (5.32) and (5.31). To simulate the noisy phase oscillators, we write equation (5.31) as

$$d\theta_j = \omega dt + \mathcal{Z}(\theta) \left[ u(t)dt + \sqrt{2D}dW_j(t) \right], \quad j = 1, \dots, M,$$

where  $dW_j(t) = \eta_j(t)dt$  and  $W_j(t)$  is the standard Weiner process. We use the second order Runge-Kutta algorithm developed in [97] to simulate the above equation, and use `randn` with a predefined seed in Matlab for generating the standard Weiner process. In order to be consistent, we evaluate the phase distribution and the control input using a second order Runge-Kutta method as well. As a proof of concept, here we again use the two-dimensional reduced Hodgkin-Huxley model for calculating the PRC. We take the control parameter  $P = 1200$ , and simulate until  $t = 3T$ . The results are shown in Figure 5.5. From the top panel, we see that the control input is able to transform an initial uniform distribution into a bi-modal distribution, and thus the Lyapunov function decreases towards zero. The top right panel of Figure 5.5 shows 100 desynchronized phase oscillators to which the control input from the bottom middle panel of Figure 5.5 is applied in an open loop manner. As seen from the bottom right panel of Figure 5.5, the control input is able to separate the desynchronized oscillators into two distinct clusters corresponding to the bi-modal phase distribution. In transforming the probability distribution, the total control energy consumed ( $\int_0^{3T} u(t)^2 dt$ ) comes out to be 153.30 units. The control input  $u(t)$  used for this energy consumption calculation is taken from equation (5.36), and thus is same for all stochastic realizations with the same noise intensity. The energy consumption is 0.46% more than the similar control without

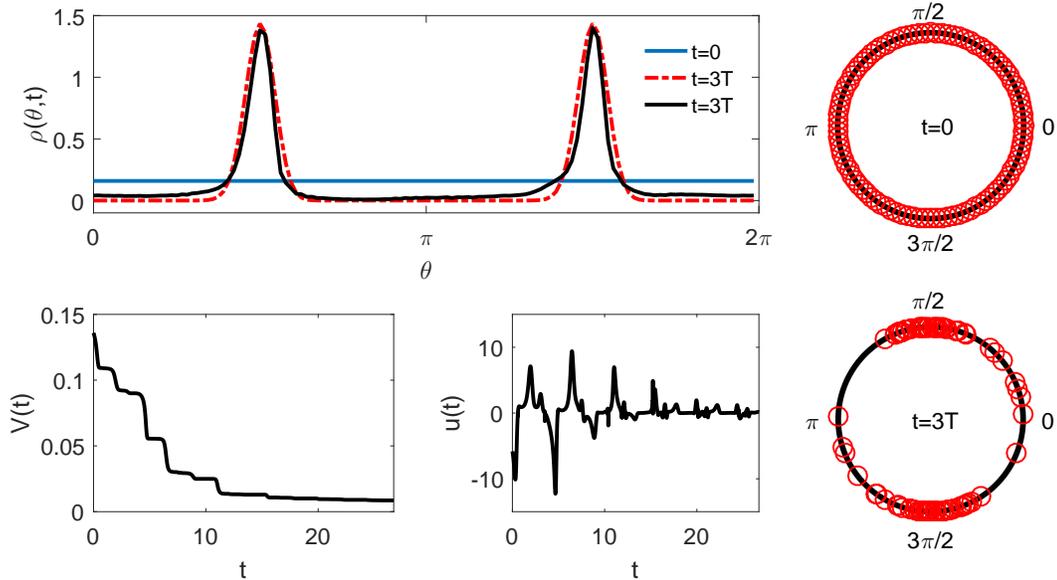


Figure 5.5: Clustering Control in presence of noise: In the top left panel, the solid (resp., red dashed) lines show the probability distribution  $\rho(\theta, t)$  (resp.,  $\rho_f(\theta, t)$ ) at various times. The bottom middle and left panels show the control input (5.36), and the Lyapunov function  $V(t)$  (5.12), respectively. The top (resp., bottom) right panels show 100 desynchronized (resp., clustered) phase oscillators at time  $t = 0$  ms (resp.,  $t = 3T$  ms). Here  $T = 8.91$  ms.

noise. This is expected as the addition of white noise introduces a diffusion term in the phase distribution PDE, and thus causes the phase distribution to decay down towards a uniform distribution with time. Therefore, the control has to expend additional effort in transforming the phase distribution into a bi-modal distribution. We note that non-zero control will be necessary to maintain the bi-modal distribution for all time.

## 5.7 Optimal Control of Phase Distributions

In this section we formulate an optimal control algorithm to transform the phase distribution  $\rho(\theta, t)$  into the desired distribution  $\rho_f(\theta, t)$ . We do this by controlling the Fourier coefficients of the phase distribution. We start with the coefficients  $A_k(0)$  and

$B_k(0)$  of  $\rho(\theta, t)$  at time  $t = 0$ , and want them to match the coefficients  $\widetilde{A}_k(\tau)$  and  $\widetilde{B}_k(\tau)$  of  $\rho_f(\theta, t)$  at time  $t = \tau$ . Thus we pose the optimal control problem as the following Two Point Boundary Value Problem (BVP). We take the cost function  $R$  as

$$R = \int_0^\tau \left\{ u^2 + \sum_{k=1}^{N-1} \left[ \lambda_{kA} \left( \dot{A}_k + k\omega B_k + \mathcal{I}_{kA}u \right) + \lambda_{kB} \left( \dot{B}_k - k\omega A_k - \mathcal{I}_{kB}u \right) \right] \right\} dt. \quad (5.37)$$

The first term in the cost function ensures that the control law uses a minimum energy input. The second term ensures that the phase distribution evolves according to equation (5.1), with  $\lambda_{kA}$  and  $\lambda_{kB}$  being the Lagrange multipliers. The Euler-Lagrange equations are obtained from

$$\frac{\partial P}{\partial q} = \frac{d}{dt} \left( \frac{\partial P}{\partial \dot{q}} \right), \quad q = \lambda_{kA}, \lambda_{kB}, A_k, B_k, u, \quad (5.38)$$

where  $P$  is the integrand in the cost function  $R$ . This gives the Euler-Lagrange equations for  $k = 1, \dots, N - 1$ :

$$\dot{A}_k = -k\omega B_k - \mathcal{I}_{kA}u, \quad (5.39)$$

$$\dot{B}_k = k\omega A_k + \mathcal{I}_{kB}u, \quad (5.40)$$

$$\dot{\lambda}_{kA} = -k\omega \lambda_{kB} + \mathcal{H}_{kA}u, \quad (5.41)$$

$$\dot{\lambda}_{kB} = k\omega \lambda_{kA} + \mathcal{H}_{kB}u, \quad (5.42)$$

where

$$u = \frac{1}{2} \sum_{k=1}^{N-1} [\lambda_{kB} \mathcal{I}_{kB} - \lambda_{kA} \mathcal{I}_{kA}], \quad (5.43)$$

$$\mathcal{H}_{kA} = \frac{1}{\pi} \int_0^{2\pi} \mathcal{Z}(\theta) \Lambda(\theta, t) \cos(k\theta) d\theta, \quad (5.44)$$

$$\mathcal{H}_{kB} = \frac{1}{\pi} \int_0^{2\pi} \mathcal{Z}(\theta) \Lambda(\theta, t) \sin(k\theta) d\theta, \quad (5.45)$$

$$\Lambda(\theta, t) = \sum_{l=1}^{N-1} l [\lambda_{lA} \sin(l\theta) - \lambda_{lB} \cos(l\theta)]. \quad (5.46)$$

We solve the Euler-Lagrange equations as a two point BVP with the boundary conditions:

$$\begin{aligned} A_k(0) &= \frac{1}{\pi} \int_0^{2\pi} \rho(\theta, 0) \cos(k\theta) d\theta, & B_k(0) &= \frac{1}{\pi} \int_0^{2\pi} \rho(\theta, 0) \sin(k\theta) d\theta, \\ A_k(\tau) &= \frac{1}{\pi} \int_0^{2\pi} \rho_f(\theta, \tau) \cos(k\theta) d\theta, & B_k(\tau) &= \frac{1}{\pi} \int_0^{2\pi} \rho_f(\theta, \tau) \sin(k\theta) d\theta. \end{aligned} \quad (5.47)$$

Since  $A_k(0)$ , and  $B_k(0)$  are fixed by the problem, the BVP can be solved by finding appropriate values of  $\lambda_{kA}(0)$  and  $\lambda_{kB}(0)$ . We formulate a modified Newton iteration method to solve this high dimensional  $(2N - 2)$  BVP. For details of the method, see B.4.

We demonstrate the control by considering the application of phase shifting a distribution as we did in Section 5.5.3. Here as well we consider the YNI model of SA node cells in rabbit heart. We start with a synchronous population distribution with mean  $\pi/2$  and  $\kappa = 52$ . We use our optimal control algorithm to phase shift this distribution by  $\pi/4$  in time  $\tau = T$ . So, we take the target distribution  $\rho_f(\theta, t)$  with same  $\kappa$  value but an initial mean of  $3\pi/4$ . We also compute the Lyapunov function  $V(t)$  for comparison with our results from Section 5.5.3, even though our optimal control algorithm is not based on this Lyapunov function. Results are shown in Figure 5.6. From the top and bottom left panels of Figure 5.6, we see that the control input is able to phase shift the phase distribution, and thus decreases the Lyapunov function towards zero (non-monotonically in this case). The top right panel of Figure 5.6 shows 100 phase oscillators synchronized with mean  $\pi/2$ , and  $\kappa = 52$  extracted through the Matlab circular statistical toolbox. We apply the control input from the middle bottom panel of the figure to them in an open loop manner. The bottom right panel of the same figure shows the oscillators at time  $t = T$ . We see that the control input is able to phase shift these oscillators by  $\pi/4$ .

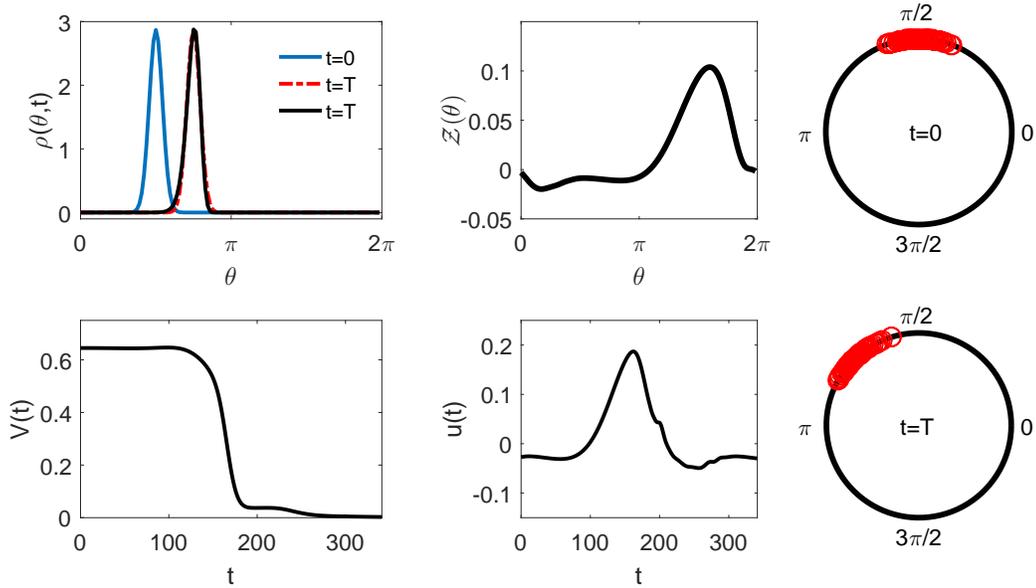


Figure 5.6: Phase shifting cardiac pacemaker cells through optimal control: In the top left panel, the solid (resp., red dashed) lines show the probability distribution  $\rho(\theta, t)$  (resp.,  $\rho_f(\theta, t)$ ) at various times. The top middle panel shows the PRC, while the bottom left and middle panels show the Lyapunov function  $V(t)$  (5.12), and the control input, respectively. The top (resp., bottom) right panels show 100 phase oscillators at time  $t = 0$  ms (resp.,  $t = T$  ms). Here  $T = 340.8$  ms. Note that in the absence of control input, the oscillators would have a mean of  $\pi/2$  at  $t = T$ .

In shifting the phase of the probability distribution, the total control energy consumed ( $\int_0^T u^2 dt$ ) comes out to be 1.56 units, which is less than half of the energy required for the same control objective using our Lyapunov-based control algorithm in Section 5.5.3. We note that this energy comparison is fair as in both cases the control input decreases the Lyapunov function by the same amount (by 99.6%). Thus our optimal control is able to achieve the control objective while simultaneously minimizing the amount of total energy required.

## 5.8 Conclusion

In this chapter we developed a framework to control a population of uncoupled oscillators by controlling their phase distributions. By writing the phase distribution as a finite Fourier series, we were able to decompose the PDE governing the evolution of the phase distribution into a set of ODEs governing the evolution of the corresponding Fourier coefficients. We formulated our control algorithms in Fourier space as well, driving the Fourier coefficients of the current phase distribution to the Fourier coefficients of the desired distribution with a single control input. For our first Lyapunov-based control algorithm, we constructed a degenerate set of phase distributions and the phase response curves in terms of their Fourier coefficients. We extended this algorithm to take into account the effect of white noise on the dynamics of the oscillator population. Finally, we formulated an optimal control algorithm which uses a minimum energy input to achieve the desired phase distribution. Our control algorithms are quite flexible; for the systems considered in this chapter, they have the potential to drive a system of uncoupled oscillators from any initial phase distribution to any traveling-wave final phase distribution, as long as the combination of those distributions is non-degenerate.

We demonstrated the versatility of our control algorithms by using them for three distinct applications. First, motivated by the hypothesis of neural synchronization in the STN and the thalamus brain region as one of the causes of motor symptoms of parkinsonian and essential tremor, respectively, we applied our control algorithm to drive an initial synchronous phase distribution to a uniform distribution. For the second application, we defined the phase difference distribution in terms of the phase distribution, and proved some of its fundamental properties. This formulation of the phase difference distribution was essential in demonstrating the importance of a clustered neural population for enhancing spike time dependent plasticity, and thus re-wiring of neural

connections for better stability of the partially synchronous clustered state. Motivated by the elimination of cardiac alternans, we applied our algorithm to control a population of synchronized cardiac pacemaker cells by advancing their phase distribution by a specified phase. For all these applications, we demonstrated the effectiveness of our control by applying the respective control inputs to a population of 100 uncoupled noise-free phase oscillators.

An experimental setup in general will include effects due to coupling, which are absent in our control algorithm. Our algorithm would still work on such systems as long as the coupling is weak. If synchrony is stable with coupling, then it would be harder for our control algorithm to desynchronize a synchronized population in the presence of coupling. The addition of noise might make this even harder if STDP is present, as STDP promotes synchrony in the presence of noise. However, in the absence of STDP, noise would make it easier for our control algorithm to desynchronize a synchronized oscillator population.

# Chapter 6

## Supervised Learning Based Control

### 6.1 Introduction

Underactuated dynamical systems are systems with fewer actuators/controls than the dimensionality of the state space of the system. Such systems are ubiquitous in a variety of fields including physics, chemistry, biology and engineering. There have been numerous advances made on controlling such systems, with much of the work in robotics [119, 120, 121]. Control in other applications, especially biology [38, 88, 93, 94], is on the rise as it provides promising treatment strategies for several disorders such as Parkinson's disease, cardiac arrhythmias, and jet lag. Most of these control methods, both in robotics and biology, are based on the traditional model based control theory and optimal control.

Such methods work well when it is possible to model the dynamics of the system accurately, which is very difficult as the systems become complicated, especially in neuroscience applications where the dynamics of a single neuron may change rapidly depending on the response from other neurons in the network. Even if an accurate model could be built to describe the dynamics of such a system, developing a classical model based control for such an underactuated system is a challenging task. If the parameters

of the system change with time, or if the model doesn't describe the dynamics accurately, the theoretical control guarantees like stability and boundedness may not apply in real systems [32, 33]. This calls for the development of data driven control algorithms that can learn to control the system without explicitly using a model.

Artificial intelligence algorithms are able to learn to control dynamical systems by using deep neural networks. Such algorithms have been used for a long time [122, 123], but with the availability of large data sets, improvement in deep learning architectures, optimization methods, and cheap computation costs, their use is on the rise [124]. However, the success of such algorithms has been largely dependent on availability of large datasets [34], which can be limited in fields like neuroscience where the cost of obtaining human/animal brain data is very high. Moreover, the black box nature of such methods makes their analysis difficult. Such analysis would be important in tasks where failure is very costly. Another limitation of such methods is their inability to take advantage of the inherent dynamics of the system to achieve the task, which limits their performance.

All these limitations call for a new machine learning control algorithm that doesn't rely on large amounts of data, is easy to understand, and can take advantage of the underlying dynamics in achieving the task. In this chapter, we have developed two related novel supervised learning algorithms based on these three goals. Our algorithms are powerful enough to control a wide variety of complex underactuated dynamical systems, and yet have a simple structure so one can understand how they work using dynamical systems and control theory foundations. Their simple yet intelligent structure also allows them to effectively achieve the control objective by training on a sparse data set, even in the presence of noise. Our algorithms output a bang-bang (binary) control input by taking in feedback of the state of the dynamical system. The algorithms learn this control input by maximizing a reward function in both short and long time horizons. We demonstrate the versatility of our algorithms by applying them to a diverse range of underactuated

dynamical systems including: switching between bistable states, changing the phase of an oscillator, desynchronizing a population of synchronized coupled oscillators, and stabilizing an unstable fixed point of a dynamical system. For most of these applications we are able to reason why our algorithms work by using traditional dynamical systems and control theory. We compare our algorithms with traditional control algorithms and reason why our algorithms work better, especially because they autonomously learn to take advantage of the underlying system dynamics in achieving the control objective. We carry out a robustness analysis to demonstrate the effectiveness of our algorithms even in the presence of noise.

This chapter is organized as follows. In Section 6.2, we develop our supervised learning algorithms and the binary classifier to output a binary control. We demonstrate our first supervised learning algorithm by controlling underactuated bistable dynamical systems in Section 6.3, and compare our algorithm to a fully actuated control. In Section 6.4, we illustrate the effectiveness of our second supervised learning algorithm by controlling the phase of a single oscillator and comparing the algorithm to a model based optimal control algorithm. We further demonstrate the versatility of our second supervised learning algorithm by using it to desynchronize a population of synchronized coupled oscillators in Section 6.5. In Section 6.6, we apply our second algorithm to stabilize an unstable fixed point of an underactuated dynamical system, and compare the algorithm to a model based control algorithm. To demonstrate the applicability of our algorithms in a real setting, we show how their intelligent structure allows them to perform well in the presence of noise in Section 6.7. Section 6.8 summarizes our work and suggests future extensions and tools. Appendix A lists the mathematical models used in this chapter.

This chapter is based on the work originally appearing in [125].

## 6.2 Supervised Learning Algorithm

We consider an underactuated dynamical system with an additive control input  $u(\mathbf{x}(t))$  as

$$\frac{d}{dt}\mathbf{x}(t) = F(\mathbf{x}(t)) + [u(\mathbf{x}(t)), 0_{n-1}]^T, \quad \mathbf{x}(t) \in R^n, \quad (6.1)$$

where  $0_{n-1}$  is an  $n - 1$  dimensional zero vector. Thus, the control input depends on the full state of the dynamical system, and only directly affects the first state of the dynamical system. The control input is binary in nature having two values  $\{u_1 > 0, u_2\}$ , which can be chosen differently for different applications. For our first algorithm, we take  $u_2 = 0$ , and thus the control can be thought of having an “ON” state with the value  $u_1$  and an “OFF” state with value 0. For our second algorithm, we take the control to be a bang bang control with  $u_2 = -u_1$ . Both algorithms learn the control input  $u(\mathbf{x}(t))$  as a function of the state to achieve a particular control objective. They do so by learning from the data generated by sampling a model describing the underlying dynamics of the system. To demonstrate our algorithm in this chapter, we use an analytical model ( $F(\mathbf{x}(t))$ ) which generates our training data. In case a model is not available in an application, one can still use the same algorithms by obtaining training data by direct measurement of the states of the system at different times. Below we describe our supervised learning algorithms in more detail.

### 6.2.1 Supervised Learning Algorithm 1

Our first supervised learning algorithm outputs a “ON” and “OFF” binary control input. The algorithm learns what control input to output to achieve a certain control objective by maximizing a reward function  $\mathcal{R}(\mathbf{x}(t))$  which needs to be carefully designed

to achieve a control objective in a particular application.

We sample a state  $\mathbf{x}(0)$  randomly from the state space of the dynamical model of the system, and evolve the state forward in time for short time  $\Delta t$  with control state OFF. If  $\mathcal{R}(\mathbf{x}(\Delta t)) \geq \mathcal{R}(\mathbf{x}(0))$ , we set the control policy for state  $\mathbf{x}(0)$  as OFF. If it is not, we again evolve the initial state forward for the same time but with control ON, and compare  $\mathcal{R}(\mathbf{x}(\Delta t))$  for both control ON and OFF. Whichever control policy maximizes the reward is set for the sampled state  $\mathbf{x}(0)$ . We repeat the process  $N$  times by sampling more initial states randomly. The algorithm is summarized below:

---

**Algorithm 1**


---

```

Initialize  $\mathcal{X}$  as zeros( $N, \text{length}(\mathbf{x})$ ) and  $\mathcal{U}$  as zeros( $N, 1$ )
for  $i=1, N$  do
  Randomly sample  $\mathbf{x}(0)$ 
  Compute  $\mathbf{x}(\Delta t)$  and  $\mathcal{R}(\mathbf{x}(\Delta t))$  with control OFF
  if  $\mathcal{R}(\mathbf{x}(\Delta t)) \geq \mathcal{R}(\mathbf{x}(0))$  then
    Set policy for  $\mathbf{x}(0)$  as OFF
  else
    Compute  $\mathbf{x}(\Delta t)$  and  $\mathcal{R}(\mathbf{x}(\Delta t))$  with control ON
    if  $\mathcal{R}(\mathbf{x}(\Delta t))$  with control ON  $>$   $\mathcal{R}(\mathbf{x}(\Delta t))$  with control OFF then
      Set policy for  $\mathbf{x}(0)$  as ON
    else
      Set policy for  $\mathbf{x}(0)$  as OFF
    end if
  end if
  Assign  $\mathcal{X}[i, :]$  as  $\mathbf{x}(0)$  and  $\mathcal{U}[i, :]$  as the policy
end for
return  $\mathcal{X}, \mathcal{U}$ 

```

---

Such an algorithm takes advantage of the underlying dynamics by letting the trajectories evolve without any control and only switching “ON” the control when necessary. This makes our algorithm highly energy efficient. Such an algorithm is very suitable for controlling bistable dynamical systems where the objective is for the trajectory to converge to a particular stable state of the system, or to switch from one stable state to another. The control can switch OFF when the trajectory enters the region of attraction

of the desired stable state, and let the dynamics take the trajectory to the desired state.

### 6.2.2 Supervised Learning Algorithm 2

Our second supervised learning algorithm outputs a bang-bang control input which can be used to control a variety of dynamical systems, including coupled oscillators. The algorithm learns what control input to output to achieve a certain control objective by maximizing a reward function  $\mathcal{R}(\mathbf{x}(\Delta t))$ , which needs to be carefully designed to achieve a control objective in a particular application.

We sample a state  $\mathbf{x}(0)$  randomly from the state space of the dynamical model of the system, and evolve the state forward in time for short time  $\Delta t$  with both control  $u_1$  and  $-u_1$ . In both scenarios we let the state evolve further in time with zero control until some event occurs and measure the timing of this event. The reward  $\mathcal{R}(\mathbf{x}(\Delta t))$  is dependent on the timing of this event. Whichever policy ( $u_1$  or  $-u_1$ ) maximizes this reward is set for that sampled state. We repeat the process  $N$  times by sampling more states randomly. The algorithm is summarized below:

---

#### Algorithm 2

---

```

Initialize  $\mathcal{X}$  as zeros( $N$ ,length( $\mathbf{x}$ )) and  $\mathcal{U}$  as zeros( $N$ ,1)
for  $i=1,N$  do
  Randomly sample  $\mathbf{x}(0)$ 
  Compute  $\mathbf{x}(\Delta t)$  and  $\mathcal{R}(\mathbf{x}(\Delta t))$  with control  $u_1$ 
  Compute  $\mathbf{x}(\Delta t)$  and  $\mathcal{R}(\mathbf{x}(\Delta t))$  with control  $-u_1$ 
  if  $\mathcal{R}(\mathbf{x}(\Delta t))$  with control  $u_1 \geq \mathcal{R}(\mathbf{x}(\Delta t))$  with control  $-u_1$  then
    Set policy for  $\mathbf{x}(0)$  as  $u_1$ 
  else
    Set policy for  $\mathbf{x}(0)$  as  $-u_1$ 
  end if
  Assign  $\mathcal{X}[i, :]$  as  $\mathbf{x}(0)$  and  $\mathcal{U}[i, :]$  as the policy
end for
return  $\mathcal{X}, \mathcal{U}$ 

```

---

Such control is useful when the objective is to converge to an unstable state of the

system, because the control needs to stay “ON” (be non-zero) the whole time even when the control objective has been realized, since the trajectory will go back to the stable state otherwise. Here as well the underlying dynamics of the system play a role in our learning algorithm: to determine the control input for a particular initial state, we let the dynamics evolve the trajectory until an event occurs.

Both these algorithms generate data comprising a set of  $N$  sampled states of the dynamical system  $\mathcal{X}$ , and a set of the corresponding control inputs  $\mathcal{U}$ . However, we need to know the control input  $u(\mathbf{x}(t))$  as a function of a general trajectory  $\mathbf{x}(t)$  of the system, since the trajectory is not restricted to these sampled states. This is achieved with our binary classifier, that takes an input as the state of the dynamical system  $\mathbf{x}(t)$  and outputs the corresponding control input  $u(\mathbf{x}(t))$  based on this generated data.

### 6.2.3 Binary Classifier

Our supervised learning algorithms generate data comprising a set  $\mathcal{X}$  of  $N$  sampled states of the dynamical system, and a set of the corresponding control inputs  $\mathcal{U}$ . Based on this information we build a locally weighted binary classifier that takes as input as the instantaneous state of the dynamical system  $\mathbf{x}(t)$ , and outputs the corresponding control input  $u(\mathbf{x}(t))$  to be applied at that instant.

We assign each element of the set  $\mathcal{X}$  with a weight

$$w_i(\mathbf{x}(t)) = \exp\left(-\frac{|\mathbf{x}(t) - \mathcal{X}_i|^2}{2\tau}\right), \quad i = 1, 2, \dots, N,$$

where  $\mathcal{X}_i$  represents the  $i^{\text{th}}$  sampled state stored in the set  $\mathcal{X}$ . Thus a sampled state is given a higher weight if its closer to  $\mathbf{x}(t)$ , and a lower weight if it is further away from

$\mathbf{x}(t)$ . These weights are normalized so that

$$\sum_{i=1}^N w_i(\mathbf{x}(t)) = 1.$$

For the first algorithm, the classifier outputs  $u(\mathbf{x}(t)) = u_1$  (“ON”) if

$$\sum_{i=1}^N w_i(\mathbf{x}(t))\mathcal{U}_i > 0.5u_1, \quad (6.2)$$

and  $u(\mathbf{x}(t)) = 0$  (“OFF”) otherwise. Similarly for the second algorithm, the classifier outputs  $u(\mathbf{x}(t)) = u_1$  if

$$\sum_{i=1}^N w_i(\mathbf{x}(t))\mathcal{U}_i > 0, \quad (6.3)$$

and  $u(\mathbf{x}(t)) = -u_1$  otherwise.

With  $u(\mathbf{x}(t))$  defined to be the output of this binary classifier, we simulate the dynamical system from (6.1) starting from a random initial condition and find that our supervised learning algorithms are able to achieve the desired control objectives, while simultaneously maximizing the designed reward functions. The entire algorithm is depicted in the flowchart in Figure 6.1. Note that in our algorithm, the sets  $\mathcal{X}$ ,  $\mathcal{U}$  need to be computed only once for a given dynamical system, whereas the control input  $u(\mathbf{x}(t))$  is computed by the binary classifier at every timestep.

### 6.3 Bistable Dynamical Systems

Bistability is widely found in neural systems [126] and cardiac arrhythmia [127], and is used in digital electronics for storing binary information, in mechanical switches for transitioning between ON and OFF states, and in multivibrators, Schmitt trigger circuits,

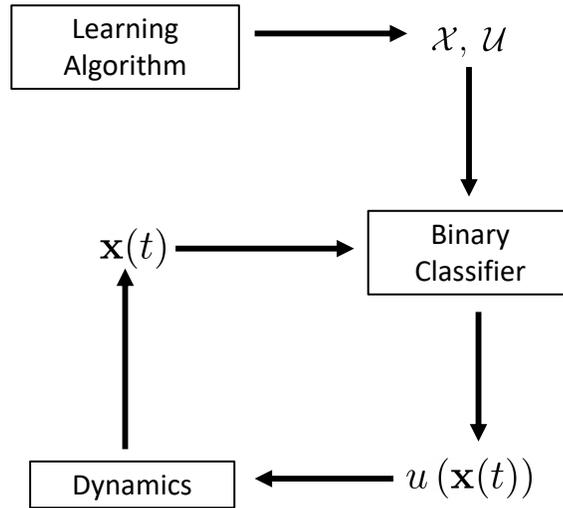


Figure 6.1: Flowchart of the Learning Algorithm.

and even optical systems [128]. It is the key mechanism for understanding several cellular processes including those associated with the onset and treatment of cancer [129]. In this section we apply our first supervised learning algorithm to control underactuated bistable dynamical systems. The control objective is for the trajectory to converge to a particular stable fixed point of the system starting anywhere in the state space (including in the basin of attraction of the other stable state). Such a control objective is relevant for several applications such as biocomputing, gene therapy, and treatment of cancer [130, 131], among others.

### 6.3.1 Duffing System

With the Duffing system [63, 132], we consider the class of bistable dynamical systems having two stable fixed points ( $\mathbf{x}^{s1}$ ,  $\mathbf{x}^{s2}$ ), and an unstable fixed point ( $\mathbf{x}^u$ ). The control objective is for the trajectory to converge to  $\mathbf{x}^{s2}$  starting anywhere in the state space.

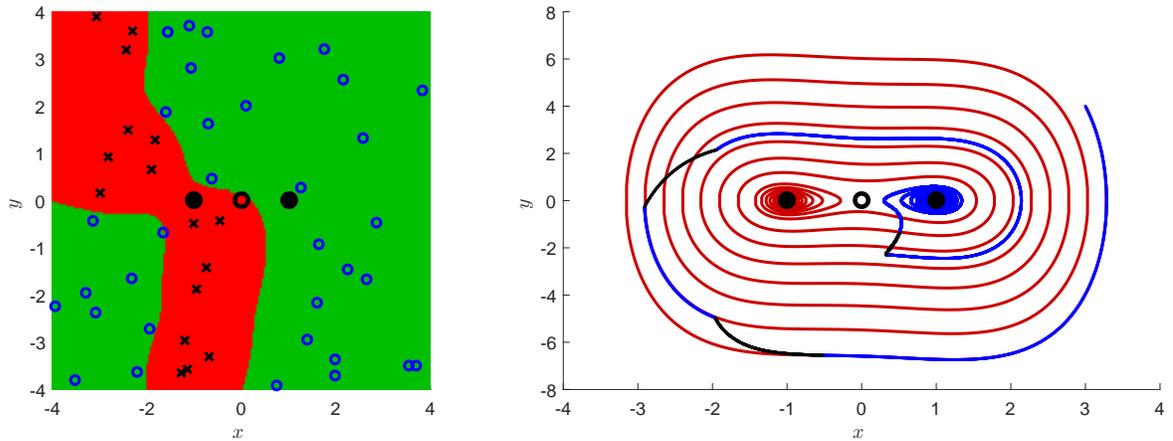


Figure 6.2: Duffing System ( $\delta = 0.1$ ): Solid (resp., open) black circles represent  $\mathbf{x}^{s^1}$ ,  $\mathbf{x}^{s^2}$ , (resp.,  $\mathbf{x}^u$ ). In the left panel, open blue circles (resp., black  $\times$ 's) represent elements of the set  $\mathcal{X}$  where the control policy given by elements of the set  $\mathcal{U}$  is OFF (resp., ON). The green (resp., red) region is where the output  $u(\mathbf{x}(t))$  of the binary classifier is OFF (resp., ON). In the right panel, the trajectory starts in the region of attraction of  $\mathbf{x}^{s^1}$ , and converges to  $\mathbf{x}^{s^2}$  (resp.,  $\mathbf{x}^{s^1}$ ) with (resp., without) control. When the control input is ON (resp., OFF), the trajectory is plotted in black (resp., blue). The uncontrolled trajectory is plotted in red.

The Duffing system is given as:

$$\begin{aligned}\dot{x} &= y + u(\mathbf{x}(t)), \\ \dot{y} &= x - x^3 - \delta y.\end{aligned}$$

For  $\delta > 0$ , the system has two stable fixed points  $\mathbf{x}^{s^1} = (-1, 0)$  and  $\mathbf{x}^{s^2} = (1, 0)$ , and an unstable fixed point  $\mathbf{x}^u = (0, 0)$ , all shown in Figure 6.2. We take  $\delta = 0.1$  in our simulations.

## Learning Algorithm

We choose our reward function to be the negative of the Euclidean distance between the current state and the desired state:

$$\mathcal{R}(\mathbf{x}(t)) = -\|\mathbf{x}(t) - \mathbf{x}^{s2}\|. \quad (6.4)$$

Thus the control will make the trajectory converge towards the desired fixed point while increasing the reward to 0. To converge to  $\mathbf{x}^{s2}$  starting anywhere in the state space, we use our learning algorithm to generate a control policy. The ON (resp., OFF) state of the control policy corresponds to a value of  $u_1 = 4$  (resp., 0). We randomly sample  $N = 50$  points for generating the sets  $\mathcal{X}$ ,  $\mathcal{U}$ , and choose  $\Delta t = 0.001$  and  $\tau = 0.4$ .

## Results

The generated control policy is shown in the left panel of Figure 6.2. Blue open circles (resp., black  $\times$ 's) represent elements of the set  $\mathcal{X}$  where the control policy given by elements of the set  $\mathcal{U}$  is OFF (resp., ON). The green (resp., red) region is where output  $u(\mathbf{x}(t))$  of the binary classifier is OFF (resp., ON). A controlled and an uncontrolled trajectory starting from same  $\mathbf{x}(0)$  is shown in the right panel of Figure 6.2. As can be seen in this figure, the control algorithm gradually converges the trajectory to  $\mathbf{x}^{s2}$  by turning the control ON a few times, whereas the uncontrolled trajectory converges to  $\mathbf{x}^{s1}$ .

The algorithm generates an energy efficient control input as the control is OFF 60.41% of the total time it takes to drive the trajectory within a ball of radius of 0.45 in the region of attraction of  $\mathbf{x}^{s2}$ . We investigate the robustness of our learning algorithm by testing it on 1000 randomly generated initial conditions, and in all the 1000 cases, the control algorithm is able to converge the trajectories to  $\mathbf{x}^{s2}$ , achieving 100% accuracy.

Choosing  $N$  is the crucial task in our learning algorithm. We start with a small  $N$  and keep it increasing until the algorithm achieves 100% effectiveness.  $N = 50$  points turns out to be appropriate for the Duffing system as choosing a lower number of points leads to underfitting, and choosing a higher number of points leads to overfitting. Both underfitting and overfitting can have a detrimental effect on the accuracy of the learning algorithm.

### 6.3.2 Reduced Hodgkin-Huxley model

With the reduced Hodgkin-Huxley model [133, 4, 108] introduced in Chapter 5, we consider the class of bistable dynamical systems having a stable periodic orbit  $\mathbf{x}^{s1}(t)$ , an unstable periodic orbit  $\mathbf{x}^u(t)$ , and a stable fixed point  $\mathbf{x}^{s2}$ . We take the model parameter  $I$  as  $6.69 \mu A/cm^2$ . For the rest of the parameters, see Appendix A.4. In the absence of control input, the system is bistable having  $\mathbf{x}^{s1}(t)$  with period 14.91 ms,  $\mathbf{x}^u(t)$  with period 14.33 ms, and  $\mathbf{x}^{s2} = (-61.04, 0.38)$ , all shown in Figure 6.3.

#### Learning Algorithm

The control objective is for the trajectory to converge to the stable fixed point starting anywhere in the state space. Here as well we choose the reward function (6.4). Without any control input, a trajectory starting outside  $\mathbf{x}^u(t)$  will converge to  $\mathbf{x}^{s1}(t)$ , and a trajectory starting inside  $\mathbf{x}^u(t)$  will converge to  $\mathbf{x}^{s2}$ . To converge to the stable fixed point starting anywhere in the state space, we use our learning algorithm to generate a control policy. The ON (resp., OFF) state of the control policy corresponds to a value of  $u_c = 15$  (resp., 0). We sample  $N = 1000$  points for generating the sets  $\mathcal{X}$ ,  $\mathcal{U}$ , and choose  $\Delta t = 0.001$ , and  $\tau = 0.001$ . Because the two state variables  $v$ ,  $n$  scale differently, it is important to normalize them for calculating the reward function. This is also important

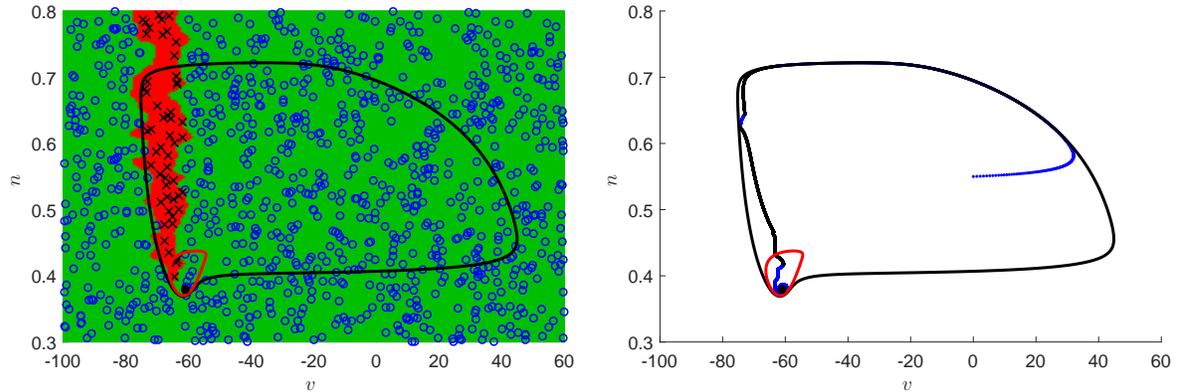


Figure 6.3: Reduced Hodgkin-Huxley model: The black and red curves are  $\mathbf{x}^{s1}(t)$  and  $\mathbf{x}^u(t)$ , respectively. The black point in the bottom left corner of figure panels is  $\mathbf{x}^{s2}$ . In the left panel, small black circles (resp., black  $\times$ 's) represent elements of the set  $\mathcal{X}$  where the control policy given by elements of the set  $\mathcal{U}$  is OFF (resp., ON). Green (resp., red) regions are where the output  $u(\mathbf{x}(t))$  of the binary classifier is OFF (resp., ON). In the right panel, the trajectory starts outside  $\mathbf{x}^u(t)$ , and converges to  $\mathbf{x}^{s2}$ . When the control input is ON (resp., OFF), the trajectory is plotted in black (resp., blue) color.

for the binary classifier to work effectively, since it is based on the Euclidean norm. To do this, we subtract from each element of the set  $\mathcal{X}$  the mean of the set and then divide each element by the variance of the set. We subtract the same mean and divide by the same variance from the state  $\mathbf{x}(t)$  that goes in calculating the reward function and also the binary classifier.

## Results

The generated control policy is shown in the left panel of Figure 6.3. As shown in this figure, the learning algorithm indicates that it is better to have control ON in the left part of the state space in order to maximize the reward function. In all other regions, the learning algorithm indicates that the control policy should be OFF. Since the control policy is ON in only a small region of the state space, we need to sample 1000 points to accurately determine this region. A controlled trajectory using this policy is

shown in the right panel of Figure 6.3. The learning algorithm is able to converge the trajectory to the stable fixed point  $\mathbf{x}^{s2}$  by bypassing the unstable periodic orbit  $\mathbf{x}^u(t)$ . The algorithm generates an energy efficient control input as the control is OFF 23.81% of the time it takes for the algorithm to drive the trajectory inside  $\mathbf{x}^u(t)$ . We investigate the robustness of our learning algorithm by testing it on 1000 randomly generated initial conditions, and in all the 1000 cases, the algorithm is able to converge the trajectories to  $\mathbf{x}^{s2}$ , achieving 100% effectiveness. Note that the learning algorithm has no information about the periodic orbits and fixed points of the system, it only works to maximize the reward function.

### Comparison with fully actuated control

To further demonstrate energy efficiency of our learning algorithm, we compare it with a fully actuated feedback control given as

$$\frac{d}{dt}\mathbf{x}(t) = F(\mathbf{x}(t)) + U(\mathbf{x}(t)), \quad \mathbf{x}(t) \in R^n, \quad (6.5)$$

$$U(\mathbf{x}(t)) = -F(\mathbf{x}(t)) - 0.2(\mathbf{x}(t) - \mathbf{x}^{s2}), \quad (6.6)$$

which also converges the trajectory to the stable fixed point in the same time frame as our learning algorithm. However the energy required by this algorithm calculated as  $\int_0^t \|U(\mathbf{x}(t))\|_2^2 dt$  comes out to be more than 3 orders of magnitude larger compared to the energy taken by the control obtained from our learning algorithm. This is because our learning algorithm takes advantage of the natural dynamics of the system to drive the trajectory close to the desired point, and turns the control ON only for a short amount of time when its really needed. In contrast, the feedback based control is ON the whole time, even when the trajectory reaches inside the unstable periodic orbit.

## 6.4 Phase Control of an Oscillator

In this section, we use our second algorithm to control a class of underactuated dynamical systems having a stable limit cycle solution  $\mathbf{x}^s(t)$ . We seek to maximally increase or decrease the phase of the limit cycle solution by using a bang-bang type control input. The motivation behind such a control objective comes from controlling neurons, where one might want a neuron to spike as quickly as possible subject to a constraint on the magnitude of the allowed input current; this constraint can be due to hardware limitations and/or concern that large inputs might cause tissue damage. Thus, instead of thinking in terms of maximally increasing the phase, one can instead think in terms of maximally decreasing the neuron's spike time.

### 6.4.1 Model

To demonstrate our algorithm, we consider the 3-dimensional thalamic neuron model [45] introduced in Chapter 2 that describes the oscillatory behavior of neurons in the thalamus. Here the state  $\mathbf{x}(t)$  is the tuple  $(v, h, r)$ ,  $v$  is the transmembrane voltage, and  $h, r$  are the gating variables of the neuron. For details of the rest of the parameters, see Appendix A.1. With no control input, these parameters give a stable limit cycle  $\mathbf{x}^s(t)$  with period  $T = 8.40 \text{ ms}$  shown in red in Figure 6.4.

### 6.4.2 Learning Algorithm

Here the control objective is to maximally decrease the spike time of the neuron, meaning we want the oscillation to end sooner than it naturally would. We set the reward function as the negative of the neuron's next spike time (the time when the

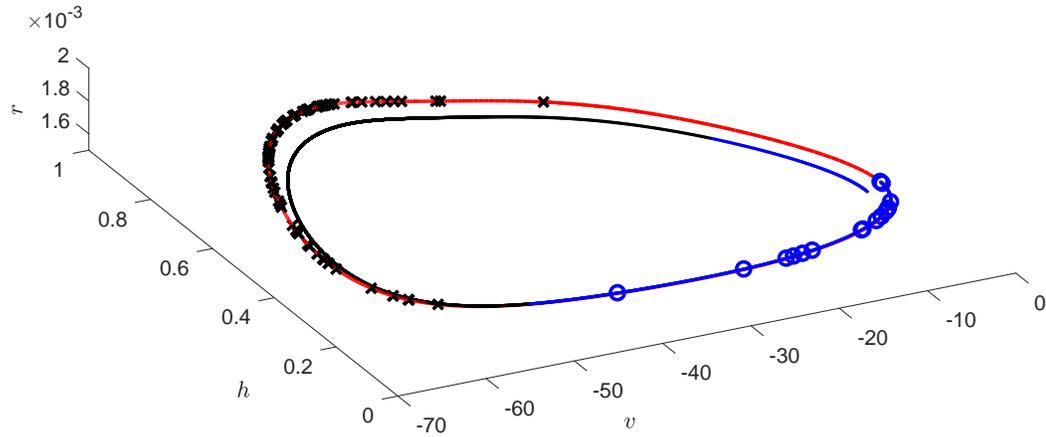


Figure 6.4: Thalamocortical neuron model: Red curve is the stable limit cycle. Small blue circles (resp., black  $\times$ 's) represents elements of set  $\mathcal{X}$  where control policy given by elements of set  $\mathcal{U}$  is  $-u_1$  (resp.,  $u_1$ ). The controlled trajectory is plotted in blue (where  $u(\mathbf{x}(t)) = -u_1$ ) and black (where  $u(\mathbf{x}(t)) = u_1$ ).

transmembrane voltage  $v(t)$  reaches a maximum):

$$\mathcal{R}(\mathbf{x}(t)) = -t_{spike}. \quad (6.7)$$

We sample 100 states randomly along the limit cycle and evolve them with both positive and negative control inputs for time  $\Delta t = 0.001$ , and then evolve them further with zero control input until the neuron spikes. Whichever control input attains the minimal  $t_{spike}$  (maximizes the reward function) is selected as control policy for that sampled state. We choose  $\tau = 0.01$ . Because the state variables  $v$ ,  $h$ ,  $r$  have different dynamic ranges, we normalize the set  $\mathcal{X}$  and the state at every time step similar to in Section 6.3.2.

### 6.4.3 Results

The generated control policy, along with the controlled trajectory, is shown in Figure 6.4. As seen in this figure, most of the sampled states need to have a positive control in order to maximize the reward function. This is evident from the left panel of Figure 6.5 which plots the corresponding control input. Because of the control, the neuron spikes ( $v$  reaches its maximum) in  $t_{spike} = 7.49ms$  which is 10.82% decrease in its natural spike time of  $8.40ms$ . Thus our algorithm is able to achieve the control objective while keeping the controlled trajectory close to the stable limit cycle solution (see Figure 6.4).

### 6.4.4 Model based control comparison

The dynamics of neural oscillations are highly nonlinear and high dimensional, which makes a model based control formulation very challenging. Phase reduction valid close to the limit cycle can overcome these challenges. The neuron spike time control problem was solved as an optimal control problem in [68, 38] using phase reduction, which also resulted in a bang bang control with control input given as

$$u(\mathbf{x}(t)) = -\text{sign}[\mathcal{Z}(\theta)]u_1 \quad \text{for decreasing } t_{spike}, \quad (6.8)$$

where  $u_1$  is the bound chosen by the user and  $\mathcal{Z}(\theta)$  is the phase response curve. Such a control works well, except when the bound  $u_1$  is large, where the controlled trajectory can diverge far away from the limit cycle, decreasing the accuracy of phase reduction and making the control based on phase reduction ineffective. Effectiveness of such a control also relies heavily on accurate measurement of phase response curve, which may not be possible.

We find that our learning based control outputs a control input very similar to the

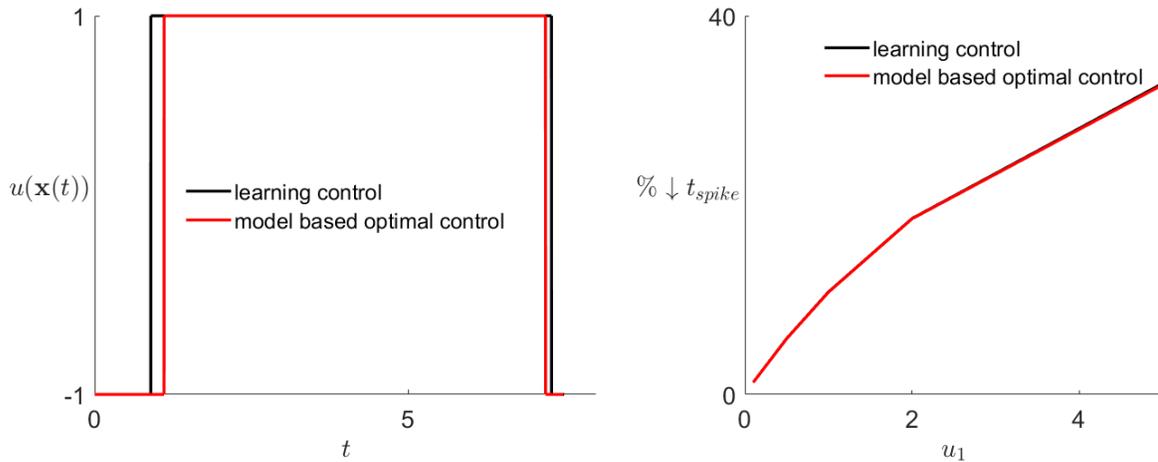


Figure 6.5: Thalamic neuron model: The left panel plots the control input  $u(\mathbf{x}(t))$  for our learning algorithm and optimal control algorithm for  $u_1 = 1$ . The right panel shows the  $\%$  decrease in  $t_{spike}$  as a function of  $u_1$ .

above model based control, both shown in the left panel of Figure 6.5 for  $u_1 = 1$ . We compute  $t_{spike}$  as a function of the bound  $u_1$  and find that our learning based algorithm does slightly better than the model based algorithm in decreasing  $t_{spike}$  (shown in the right panel of Figure 6.5). Both controls are able to decrease  $t_{spike}$  more as  $u_1$  increases.

## 6.5 Desynchronization of a Population of Coupled Oscillators

Pathological synchronization of neural oscillations in the thalamus and the subthalamic nucleus (STN) brain region is hypothesized to be one of the causes of motor symptoms for essential and parkinsonian tremor, respectively [11, 12]. We employ our algorithm to desynchronize an initially synchronized population of  $M$  coupled thalamic neural oscillations inspired by treatment of parkinsonian and essential tremor.

### 6.5.1 Model

We consider the 3-dimensional thalamic neuron model [45] for each individual oscillator with added all-to-all electrotonic coupling:

$$\dot{v}_i = \frac{-I_L(v_i) - I_{Na}(v_i, h_i) - I_K(v_i, h_i) - I_T(v_i, r_i) + I_b}{C_m} + \frac{1}{N} \sum_{j=1}^M \alpha_{ij}(v_j - v_i) + u(\mathbf{x}(t)), \quad (6.9)$$

$$\dot{h}_i = \frac{h_\infty(v_i) - h_i}{\tau_h(v_i)}, \quad (6.10)$$

$$\dot{r}_i = \frac{r_\infty(v_i) - r_i}{\tau_r(v_i)}, \quad (6.11)$$

where  $\mathbf{x}(t)$  represents the full state ( $3 \times M$  dimensional) of the oscillator population. Here,  $i = 1, \dots, M$ , where  $M$  is the total number of oscillators in the neuron population.  $v_i$  is the transmembrane voltage, and  $h_i$ ,  $r_i$  are the gating variables of the  $i^{\text{th}}$  neural oscillator.  $\alpha_{ij}$  is the coupling strength between oscillators  $i$  and  $j$ , which are assumed to be electrotonically coupled [134] with  $\alpha_{ij} = 0.01$  for  $j \neq i$  and  $\alpha_{ii} = 0$  for all  $i$ .  $u(\mathbf{x}(t))$  represents the applied current as the control input. For details of the rest of the parameters, see Appendix A.1. Note that the same control input  $u(\mathbf{x}(t))$  is applied to all of the oscillators. With no control input, these parameters give a synchronized oscillator population with period  $T = 8.40$  ms.

### 6.5.2 Learning Algorithm

We index the individual neural oscillators in the order in which they spike, thus neuron 1 spikes first and neuron  $M$  spikes last. We set the reward function as the absolute value

of spike time difference of neuron 1 and  $M$ :

$$\mathcal{R}(\mathbf{x}(t)) = |t_{spike1} - t_{spikeM}|. \quad (6.12)$$

Since the oscillator population is initially synchronized, this reward is initially a small positive number as all neurons spike very close to each other. We aim to desynchronize the population by maximally increasing this reward function. We consider  $M = 51$  oscillators in the synchronized population and sample 51 states along the synchronized oscillation. Since the state of the oscillator population is very large ( $3 \times M$ ), we take the mean across the population to reduce the dimension of our set  $\mathcal{X}$ . The  $i^{th}$  element of the set  $\mathcal{X}$  is given as

$$\mathcal{X}_i = \frac{\sum_{i=1}^M (v_i, h_i, r_i)}{M}. \quad (6.13)$$

We evolve the oscillator population with both positive and negative control inputs for time  $\Delta t = 0.001$ , and evolve them further with zero control input until all neurons in the population spike. Whichever policy attains the maximum reward function is selected for that sampled state  $\mathcal{X}_i$ .

The binary classifier takes as input the full high dimensional state of the oscillator population. It then computes the mean of the state across the population and compares it with the sampled mean states to output a control input  $u(\mathbf{x}(t))$ . Because the mean of the states  $v_i$ ,  $h_i$ ,  $r_i$  scale differently, it is important to normalize them for the binary classifier to work effectively, since it is based on the Euclidean norm. Thus, we normalize the set  $\mathcal{X}$  and the mean state at every time step similar to in Section 6.3.2. We choose  $\tau = 0.01$ .

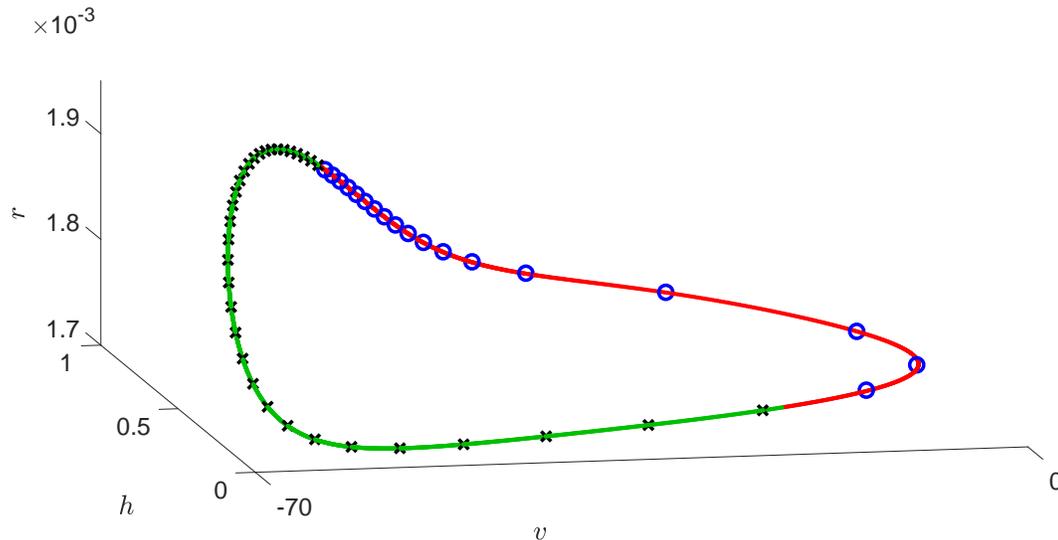


Figure 6.6: Thalamic synchronized population oscillation: The closed curve is the synchronized oscillation. Small blue circles (resp., black  $\times$ 's) represents elements of the set  $\mathcal{X}$  where the control policy given by elements of the set  $\mathcal{U}$  is  $-u_1$  (resp.,  $u_1$ ). The oscillation is plotted in red (resp., green) where  $\mathcal{Z}'(\theta)$  is negative (resp., positive).

### 6.5.3 Results

The generated control policy shown in Figure 6.6 gives a positive control input in the bottom left region of oscillation and a negative control in the top right region of the oscillation. The same figure also plots a model based control input discussed below. Figure 6.7 plots the results of desynchronization of a thalamic neuron population by our learning algorithm. As shown in both the left and right panels of the figure, the control input from our learning algorithm is able to desynchronize an initially synchronized thalamic neuron population in about 90 ms while keeping the oscillators close to the initially synchronized oscillation. It may seem that the population is largely synchronized from the right panel of Figure 6.7 but that is not the case. Since the oscillators spend most of their time near the top of the limit cycle, one naturally observes more of them near the top of the limit cycle even though they are evenly spread out in time (and not space). This becomes clear from the left panel of Figure 6.7.

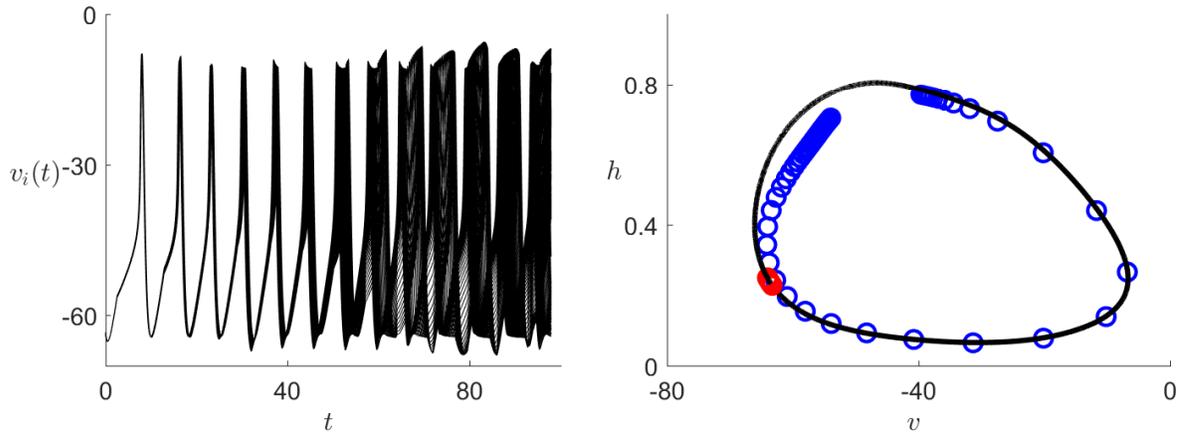


Figure 6.7: Desynchronization of thalamic neuron population: Left panel plots the state  $v_i$  for  $i = 1, \dots, 51$  neurons as a function of time. Right panel plots the initially synchronized (resp. final desynchronized) neurons as small red (resp., blue) circles.

#### 6.5.4 Model based validation of control policy

Here we analyze why the policy predicted by our learning algorithm works. Consider an oscillator population comprised of just 2 oscillators whose dynamics evolve according to phase reduction as

$$\dot{\theta}_1 = \omega + \mathcal{Z}(\theta_1)u(t), \quad (6.14)$$

$$\dot{\theta}_2 = \omega + \mathcal{Z}(\theta_2)u(t). \quad (6.15)$$

The dynamics of their phase difference  $\phi = \theta_1 - \theta_2$  can be written as (cf, [21])

$$\dot{\phi} = \mathcal{Z}'(\theta)u(t)\phi + \mathcal{O}(\phi^3), \quad (6.16)$$

where  $\theta = 0.5(\theta_1 + \theta_2)$  is the mean of the two oscillators' phases, and  $\mathcal{Z}'(\theta)$  is the derivative of the phase response curve with respect to  $\theta$ . If the oscillators are synchronized then their phase difference  $\phi \approx 0$ , thus higher order term in equation (6.16) can be ignored

and the equation can be rewritten as

$$\dot{\phi} = \mathcal{Z}'(\theta)u(t)\phi. \quad (6.17)$$

To desynchronize these two synchronized oscillators the coefficient of  $\phi$  in the above equation should be positive. This can be achieved if  $u(t)$  is of same sign as  $\mathcal{Z}'(\theta)$ . This is exactly what our policy predicts, as is shown in Figure 6.6. The policy predicts the control to be positive in the region of oscillation where  $\mathcal{Z}'(\theta)$  is positive, and it predicts the control to be negative in the region of oscillation where  $\mathcal{Z}'(\theta)$  is negative, thus explaining why our algorithm is able to desynchronize the oscillator population.

## 6.6 Stabilizing an Unstable Fixed Point

In this section we apply our second learning algorithm to stabilize an unstable fixed point of an underactuated dynamical system. This control objective is one of the oldest studied control theory problems that is employed in several fields including robotics, electrochemical systems, and treatment of cardiac arrhythmias [121, 135, 16]. To demonstrate this, we consider the Lorenz system [136] given as:

$$\dot{x} = \sigma(y - x) + u(\mathbf{x}(t)), \quad (6.18)$$

$$\dot{y} = rx - y - xz, \quad (6.19)$$

$$\dot{z} = xy - bz. \quad (6.20)$$

In the absence of control input with parameters  $\sigma = 10$ ,  $b = 8/3$ ,  $r = 1.5$ , the system is bistable with  $\mathbf{x}^{s1} = (-1.15, -1.15, 0.5)$ ,  $\mathbf{x}^u = (0, 0, 0)$ , and  $\mathbf{x}^{s2} = (1.15, 1.15, 0.5)$ , all shown in Figure 6.8.

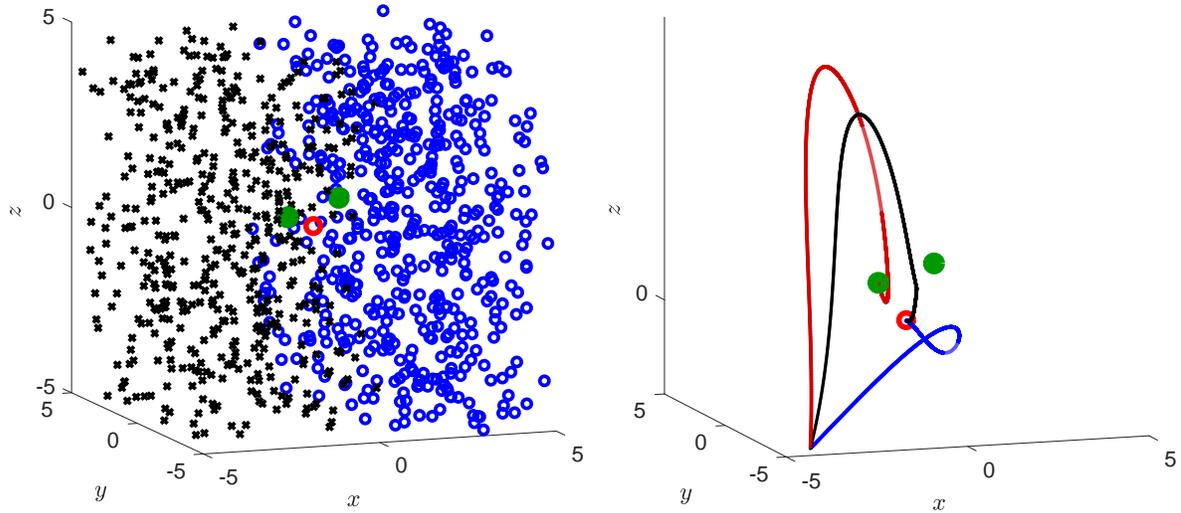


Figure 6.8: Lorenz system: Solid green (resp., open red) circles represent  $\mathbf{x}^{s1}$ ,  $\mathbf{x}^{s2}$ , (resp.,  $\mathbf{x}^u$ ). In the left panel, open blue circles (resp., black  $\times$ 's) represent elements of the set  $\mathcal{X}$  where the control policy given by elements of the set  $\mathcal{U}$  is  $-5$  (resp.,  $5$ ). In the right panel, the uncontrolled trajectory plotted in red converges to  $\mathbf{x}^{s1}$ , and the supervised learning (resp., Lyapanov) based control trajectory plotted in black (resp., blue) converges to  $\mathbf{x}^u$ .

### 6.6.1 Learning Algorithm

The control objective is for a trajectory to converge to the unstable fixed point  $\mathbf{x}^u$  starting anywhere in the state space. We choose our reward function to be the negative of the Euclidean distance between current state and the desired unstable fixed point

$$\mathcal{R}(\mathbf{x}(t)) = -\|\mathbf{x}(t) - \mathbf{x}^u\|. \quad (6.21)$$

Thus the control will make the trajectory converge towards the desired fixed point while increasing the reward to 0. To converge to  $\mathbf{x}^u$  starting anywhere in the state space, we use our learning algorithm to generate a control policy. We take  $u_1 = -u_2 = 5$ , and sample  $N = 1000$  points for generating the sets  $\mathcal{X}$ ,  $\mathcal{U}$ . We choose  $\Delta t = 0.001$  and take

the binary classifier parameter  $\tau = 5$ .

### 6.6.2 Results

The generated control policy is shown in the left panel of Figure 6.8. Blue open circles (resp., black  $\times$ 's) represent elements of the set  $\mathcal{X}$  where the control policy given by elements of the set  $\mathcal{U}$  is -5 (resp., 5). A controlled trajectory using our learning algorithm and an uncontrolled trajectory starting from same  $\mathbf{x}(0)$  are shown in the right panel of Figure 6.8. The learning algorithm converges the trajectory to  $\mathbf{x}^u$ , whereas the uncontrolled trajectory converges to  $\mathbf{x}^{s1}$ . In doing so, the learning based control consumes 150 units of control energy ( $\int_0^6 u(\mathbf{x}(t))^2 dt$ ). We investigate the robustness of our learning algorithm by testing it on 1000 randomly generated initial conditions, and in all the 1000 cases, the learning based control algorithm is able to converge the trajectories within a ball of radius 0.09 units centered at  $\mathbf{x}^u$ , achieving 100% effectiveness.

### 6.6.3 Comparison with Lyapanov based control

To demonstrate energy efficiency of our learning algorithm, we compare it with Lyapunov-based control to stabilize  $\mathbf{x}^u$ . Consider the following positive definite Lyapunov function

$$V(t) = \frac{1}{2}x(t)^2 + \frac{1}{2}y(t)^2 + \frac{1}{2}z(t)^2, \quad (6.22)$$

Its time derivative is given as

$$\dot{V}(t) = -2\sigma x(t)^2 - 2y(t)^2 - 2bz(t)^2 + 2x(t)(u(t) + (\sigma + r)y(t)), \quad (6.23)$$

where  $u(t)$  takes the place of  $u(\mathbf{x}(t))$  in equation (6.18). Then by taking  $u(t) = -(\sigma + r)y(t)$ , one gets a negative definite time derivative of the Lyapunov function. Thus by

the Lyapunov theorem, this control asymptotically stabilizes the unstable fixed point  $\mathbf{x}^u$ . The control trajectory based on this control is plotted in blue in the right panel of Figure 6.8. As seen in the figure, the Lyapunov-based control is able to converge the trajectory towards the unstable fixed point as well. But in doing so, it consumes 1176.8 units of control energy ( $\int_0^6 u(t)^2 dt$ ), which is almost 8 times the energy consumed by our learning based control. This is partly because our learning based control uses the inherent system dynamics to control the trajectory, as the controlled trajectory seems to stay close to the uncontrolled trajectory. In contrast, the Lyapunov based control drives the trajectory far away before it converges to  $\mathbf{x}^u$ , thus it ends up consuming much more energy.

## 6.7 Robustness to Noise

We have demonstrated the effectiveness of our algorithms in several scenarios in which the algorithms were based on data generated from a deterministic dynamical model. However, real data measured from an experimental setup will be noisy. In order for our algorithms to work in an experimental setup it is imperative to investigate their performance when the data is corrupted with noise. We do that by considering the Duffing system in the bistable parameter regime from Section 6.3.1. The control objective is still for the trajectory to converge to  $\mathbf{x}^{s2}$  starting anywhere in the state space.

### 6.7.1 Learning Algorithm

To replicate the effect of noise in an experimental setup, we use exactly the same parameters as before to generate the sets  $\mathcal{X}$ ,  $\mathcal{U}$  and corrupt the set  $\mathcal{X}$  by adding Gaussian white noise with mean 0 and standard deviation  $\sigma$ , resulting in the set  $\tilde{\mathcal{X}}$ . Thus each element in the dataset will be offset from its true value. We also add Gaussian white noise with the same mean and standard deviation to the state  $\mathbf{x}(t)$  resulting in  $\tilde{\mathbf{x}}(t)$ ,

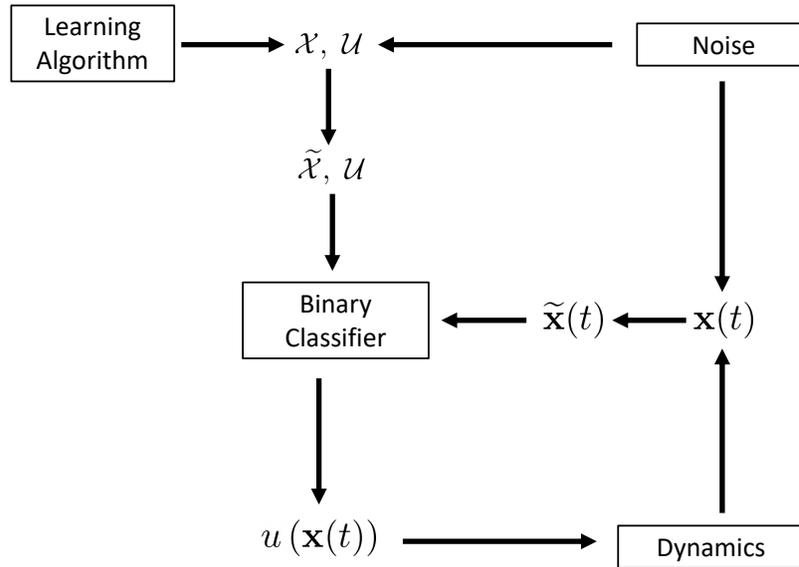


Figure 6.9: Flowchart of the Learning Algorithm with added noise.

which the binary classifier takes as input at every time step. This accounts for the noise in estimation of the state by the classifier in a real system. The flowchart from Figure 6.1 with added noise is modified and shown in Figure 6.9

## 6.7.2 Results

The generated control policy corrupted with noise of standard deviation  $\sigma = 0.2$  is shown in the left panel of Figure 6.10. Blue open circles (resp., black  $\times$ 's) represent elements of the noise corrupted set  $\tilde{\mathcal{X}}$  where the control policy given by elements of the set  $\mathcal{U}$  is OFF (resp., ON). The green (resp., red) region is where output  $u(\tilde{\mathbf{x}}(t))$  of the binary classifier is OFF (resp., ON). The elements of the original set  $\mathcal{X}$  are plotted in white to show the shifting of the elements due to the noise. Besides shifting the elements, the addition of white noise blurs the decision boundary between the ON and OFF policy region. A controlled and an uncontrolled trajectory starting from the same  $\mathbf{x}(0)$  is shown in the right panel of Figure 6.10. As can be seen in this figure, the control algorithm

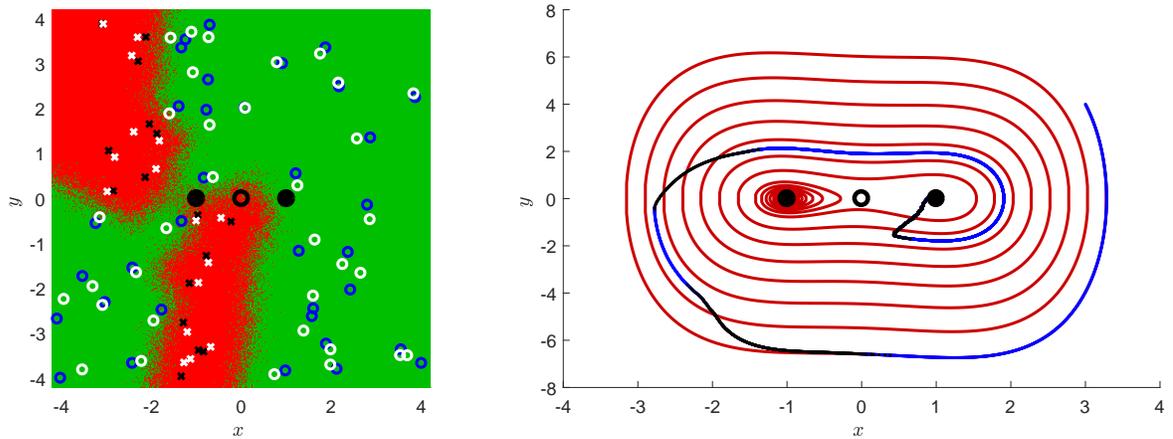


Figure 6.10: Duffing System with noise ( $\delta = 0.1$ ): Solid (resp., open) black circles represent  $\mathbf{x}^{s1}$ ,  $\mathbf{x}^{s2}$ , (resp.,  $\mathbf{x}^u$ ). In the left panel, open blue circles (resp., black  $\times$ 's) represent elements of the set  $\tilde{\mathcal{X}}$  where the control policy given by elements of the set  $\mathcal{U}$  is OFF (resp., ON). The green (resp., red) region is where the output  $u(\tilde{\mathbf{x}}(t))$  of the binary classifier is OFF (resp., ON). The elements of the original set  $\mathcal{X}$  are plotted in white. In the right panel, the trajectory starts in the region of attraction of  $\mathbf{x}^{s1}$ , and converges to  $\mathbf{x}^{s2}$  (resp.,  $\mathbf{x}^{s1}$ ) with (resp., without) control. When the control input is ON (resp., OFF), the trajectory is plotted in black (resp., blue). The uncontrolled trajectory is plotted in red.

converges the trajectory to  $\mathbf{x}^{s2}$  even though it has been corrupted by adding noise both to the training dataset and to the input of the binary classifier. On the other hand, the uncontrolled trajectory converges to  $\mathbf{x}^{s1}$ . Note that the controlled trajectory follows a different route when compared to the noiseless case from Figure 6.2. This is because the noise distorts the decision boundary between ON and OFF states of the policy, resulting in a slightly different path.

We investigate the robustness of our learning algorithm in the presence of noise by testing it on 1000 randomly generated initial conditions, and in all the 1000 cases the control algorithm is able to converge the trajectories to  $\mathbf{x}^{s2}$ , achieving 100% accuracy. Choosing  $\sigma$  does have an effect on the effectiveness of our algorithm. We observed that for  $\sigma$  up to 0.2 the algorithm achieves 100% effectiveness. This is because of the way

our binary classifier is designed. It outputs a control based on all the elements of the dataset  $\mathcal{X}$  instead of just a few nearest neighbors (see equation (6.2)). Because of noise, the elements get shifted to  $\tilde{\mathcal{X}}$ , but since all the elements are used in the summation in equation (6.2) for making a decision, the slight shifting of each element is averaged out by this summation. For  $\sigma > 0.25$ , the algorithm's effectiveness starts dropping as the shifts in  $\tilde{\mathcal{X}}$  become too big to be offset by the summation of  $N = 50$  points. Thus, if the noise intensity is very high, the algorithm can perform well by taking a higher number of points ( $N$ ) in the dataset to effectively offset the shifting of the elements.

## 6.8 Conclusion

In this chapter we have developed two novel supervised learning algorithms to control a underactuated dynamical systems. The algorithms output a bang bang (binary) control input to achieve the desired control objectives which maximize a reward function. A simple yet intelligent structure allows the algorithms to be energy efficient as they autonomously learn to take advantage of the inherent dynamics. We demonstrated the versatility of our algorithms by applying them to a diverse range of applications including: switching between bistable states, changing the phase of an oscillator, desynchronizing a population of synchronized coupled oscillators, and stabilizing an unstable fixed point. For most of these applications we were able to reason why our algorithms work by using traditional dynamical systems and control theory. We compared our algorithms to some traditional nonlinear model control algorithms and showed that our algorithms work better. We also carried out a robustness analysis to demonstrate the effectiveness of our algorithms even with noisy data. Note that having an additive control input doesn't restrict our algorithms. Since the algorithms work by maximizing the reward function, the structure of control input coming into the dynamics does not matter.

# Chapter 7

## Conclusions and Experimental Implementation

Nonlinear oscillators are ubiquitous in biology, some examples being the beating of pacemaker cells in the heart, the firing of action potentials in neurons, and circadian rhythms. This dissertation develops several control algorithms for such biological oscillators, and demonstrates their significance in devising treatment for Parkinson's disease, cardiac alternans, and jet lag.

Standard phase reduction is a crucial tool in the analysis and control of these biological oscillators. It reduces the dimensionality of a system, and can make its control experimentally amenable. However, a recently developed reduction technique called augmented phase reduction, could be better in some control applications. Thus, in the first part of the dissertation, we investigated under what dynamical regimes is the use of the augmented phase reduction better than the standard phase reduction. To do this, in Chapter 3 we derived expressions for the augmented phase reduction for six distinct systems with a periodic orbit:  $\lambda - \omega$  systems, relaxation oscillators, and systems in which periodic orbits are born out of four codimension one bifurcations. We found that for a re-

laxation oscillator, it is not necessary to use the augmented phase reduction, instead the standard phase reduction would suffice. On the other hand, for the other five systems, it is better to use the augmented phase reduction over the standard phase reduction, especially when a nontrivial Floquet exponent of the periodic orbit has small magnitude.

We continued this investigation further in Chapter 4, where we developed a novel optimal control algorithm based on the augmented phase reduction to change the phase of a single oscillator. Our algorithm not only minimized the total energy consumption but also reduced the controlled trajectory's transversal distance from the uncontrolled periodic orbit. We showed that our new algorithm works much better than a similar algorithm based on standard phase reduction, especially when a nontrivial Floquet multiplier of the periodic orbit is close to 1 and/or a significant change in phase is required. In such cases, our new algorithm can do an order of magnitude better in terms of the calculated control error.

PRCs are measured experimentally by giving perturbations to the oscillator at various phases, and recording the phase change caused by the perturbation as a function of the stimulation phase. We propose that IRCs can be measured in a similar way. One can apply perturbations at various phases, and record the resulting "amplitude" change as a function of the stimulation phase, or one can record the time required for the trajectory to return back to the periodic orbit as a function of the stimulation phase. Either of these approaches will give a measure of the IRC, which can be appropriately scaled to give the true IRC. Thus, just like the control algorithm based on standard phase reduction, we propose that our new algorithm from this chapter can be applied in an electrophysiological setting.

This finished our investigation of control of a single oscillator, and we moved on to devising control algorithms for a population of oscillators in Chapter 5. There, we developed a framework based on the phase distribution of a population of uncoupled oscillators

to control their collective behavior. We devised a Lyapunov-based control algorithm, and extended it to take into account the effect of white noise on the dynamics of the oscillator population. Finally, we formulated an optimal control algorithm which uses a minimum energy input to achieve the desired phase distribution. Our control algorithms are quite flexible; for the systems considered in this chapter, they have the potential to drive a system of uncoupled oscillators from any initial phase distribution to any traveling-wave final phase distribution, as long as the combination of those distributions is non-degenerate.

Since these algorithms require knowledge of the current phase distribution, one would need to measure neuronal/cardiac pacemaker cell activity in order to back out the phase distribution in real time. This measurement would require good spatial and temporal resolution, so for both neuroscience and cardiovascular experiments we suggest that the use of Micro-Electrode arrays (MEA) would be a good fit. Note that for *in vivo* experiments, fMRI and EEG are unlikely to be the right tools since fMRI has poor temporal resolution, while EEG is susceptible to noise and poorly measures neural activity beneath the cortex.

All these control algorithms were model dependent, which may not be suitable in situations for which it is difficult to obtain an accurate model of a system. Motivated by this, we developed data-based control algorithms in Chapter 6. We developed two novel supervised learning algorithms to control a range of underactuated dynamical systems. The algorithms output a bang bang (binary) control input to achieve the desired control objectives which maximizes a reward function. A simple yet intelligent structure allows the algorithms to be energy efficient as they autonomously learn to take advantage of the inherent dynamics. We demonstrated the versatility of our algorithms by applying them to a diverse range of applications beyond biological oscillators.

In an experimental setting such an algorithm can be implemented by stimulating the system with binary control inputs at different states of the system and determining

which control input works best for the different sampled states, and ultimately using that information in constructing a binary classifier. The data generated from an experimental setting might be corrupted with noise, thus to demonstrate the potential of our algorithm in a real setting we showed that our algorithm works even in the presence of noise. In the future, we plan to explore how to modify our algorithm if some of the states are not observable, and how to adapt our algorithm for very high dimensional dynamical systems.

Closing the loop on this dissertation, we conclude that the control algorithms were designed so that the required parameters/training data could be measured experimentally. We tested the robustness of the algorithms by demonstrating their resilience to noise, and thus showed their suitability for controlling living biological tissue. This work truly holds great potential in devising treatments for Parkinson's disease, cardiac alternans, and jet lag, and thus is a step towards improving the health and well-being of the many patients suffering from these conditions.

# Appendix A

## Models

In this appendix, we give details of the dynamical models which are necessary to reproduce the results of this dissertation.

### A.1 Thalamic neuron model

The thalamic neuron model used in Chapters 2, 3, 4, and 6 is given as

$$\begin{aligned}\dot{v} &= \frac{-I_L(v) - I_{Na}(v, h) - I_K(v, h) - I_T(v, r) + I_b}{C_m} + u(t), \\ \dot{h} &= \frac{h_\infty(v) - h}{\tau_h(v)}, \\ \dot{r} &= \frac{r_\infty(v) - r}{\tau_r(v)}.\end{aligned}$$

where

$$\begin{aligned}h_\infty(v) &= 1/(1 + \exp((v + 41)/4)), \\ r_\infty(v) &= 1/(1 + \exp((v + 84)/4)), \\ \alpha_h(v) &= 0.128 \exp(-(v + 46)/18), \\ \beta_h(v) &= 4/(1 + \exp(-(v + 23)/5)), \\ \tau_h(v) &= 1/(\alpha_h + \beta_h), \\ \tau_r(v) &= (28 + \exp(-(v + 25)/10.5)), \\ m_\infty(v) &= 1/(1 + \exp(-(v + 37)/7)), \\ p_\infty(v) &= 1/(1 + \exp(-(v + 60)/6.2)), \\ I_L(v) &= g_L(v - e_L), \\ I_{Na}(v, h) &= g_{Na}(m_\infty^3)h(v - e_{Na}), \\ I_K(v, h) &= g_K((0.75(1 - h))^4)(v - e_K),\end{aligned}$$

$$\begin{aligned}
I_T(v, r) &= g_T(p_\infty^2)r(v - e_T), \\
C_m = 1, \quad g_L = 0.05, \quad e_L = -70, \quad g_{Na} = 3, \quad e_{Na} = 50, \\
g_K = 5, \quad e_K = -90, \quad g_T = 5, \quad e_T = 0, \quad I_b = 5.
\end{aligned}$$

## A.2 YNI model

The YNI model [84] used in Chapters 4, and 5 is given as

$$\begin{aligned}
\dot{V} &= \frac{I_m - I_{Na} - I_k - I_l - I_s - I_h}{C} + u(t), \\
\dot{d} &= \alpha_d(1 - d) - \beta_d d, \\
\dot{f} &= \alpha_f(1 - f) - \beta_f f, \\
\dot{m} &= \alpha_m(1 - m) - \beta_m m, \\
\dot{h} &= \alpha_h(1 - h) - \beta_h h, \\
\dot{q} &= \alpha_q(1 - q) - \beta_q q, \\
\dot{p} &= \alpha_p(1 - p) - \beta_p p,
\end{aligned}$$

where

$$\begin{aligned}
\alpha_d &= \frac{0.01045(V + 35)}{(1 - \exp(-(V + 35)/2.5)) + \frac{0.03125V}{(1 - \exp(-V/4.8))}}, \\
\beta_d &= 0.00421(V - 5)/(-1 + \exp((V - 5)/2.5)), \\
\alpha_f &= 0.000355(V + 20)/(-1 + \exp((V + 20)/5.633)), \\
\beta_f &= 0.000944(V + 60)/(1 + \exp(-(V + 29.5)/4.16)), \\
\alpha_m &= (V + 37)/(1 - \exp(-(V + 37)/10)), \\
\beta_m &= 40 \exp(-0.056(V + 62)), \\
\alpha_h &= 0.001209(\exp(-(V + 20)/6.534)), \\
\beta_h &= 1/(1 + \exp(-(V + 30)/10)), \\
\alpha_q &= 0.0000495 + \frac{0.00034(V + 100)}{(-1 + \exp((V + 100)/4.4))}, \\
\beta_q &= 0.0000845 + 0.0005(V + 40)/(1 - \exp(-(V + 40)/6)), \\
\alpha_p &= 0.0006 + 0.009/(1 + \exp(-(V + 3.8)/9.71)), \\
\beta_p &= 0.000225(V + 40)/(-1 + \exp((V + 40)/13.3)), \\
i_s &= 12.5(\exp((V - 30)/15) - 1), \\
I_s &= (0.95d + 0.05)(0.95f + 0.05)i_s, \\
I_{Na} &= 0.5m^3h(V - 30),
\end{aligned}$$

$$\begin{aligned}
I_h &= 0.4q(V + 25), \\
I_k &= 0.7p(\exp(0.0277(V + 90)) - 1) / \exp(0.0277(V + 40)), \\
I_l &= 0.8(-\exp(-(V + 60)/20) + 1), \\
C &= 1.
\end{aligned}$$

$I_m = 1.0609$  for Chapter 4, and 0 for Chapter 5.

### A.3 Clock gene regulation model

The clock gene regulation model [95] used in Chapter 4 is given as

$$\begin{aligned}
\dot{X} &= v_1 \frac{K_1^4}{K_1^4 + Z^4} - v_2 \frac{X}{K_2 + X} + L(t), \\
\dot{Y} &= k_3 X - v_4 \frac{Y}{K_4 + Y}, \\
\dot{Z} &= k_5 Y - v_6 \frac{Z}{K_6 + Z}, \\
v_1 &= 0.7, \quad v_2 = 0.35, \quad v_4 = 0.35, \quad v_6 = 0.35, \\
K_1 &= 1, \quad K_2 = 1, \quad K_6 = 1, \quad k_3 = 0.7, \quad k_5 = 0.7.
\end{aligned}$$

### A.4 Reduced Hodgkin-Huxley model

Here we give the reduced Hodgkin-Huxley model [133, 4, 108] used in Chapters 5, and 6:

$$\begin{aligned}
\dot{v} &= (I - g_{Na}(m_\infty(v))^3(0.8 - n)(v - v_{Na}) - g_K n^4(v - v_K) - g_L(v - v_L)) / c + u(t), \\
\dot{n} &= a_n(v)(1 - n) - b_n(v)n,
\end{aligned}$$

where  $v$  is the trans-membrane voltage, and  $n$  is the gating variable.  $I$  is the base-line current, which we take as  $6.69 \mu A/cm^2$ , and  $\pi(\mathbf{x}(t))$  represents the applied control current.

$$\begin{aligned}
a_n(v) &= 0.01(v + 55) / (1 - \exp(-(v + 55)/10)), \\
b_n(v) &= 0.125 \exp(-(v + 65)/80), \\
a_m(v) &= 0.1(v + 40) / (1 - \exp(-(v + 40)/10)), \\
b_m(v) &= 4 \exp(-(v + 65)/18), \\
m_\infty(v) &= a_m(v) / (a_m(v) + b_m(v)), \\
c &= 1, \quad g_L = 0.3, \quad g_{Na} = 120, \quad v_{Na} = 50 \\
g_K &= 36, \quad v_K = -77, \quad v_L = -54.4 \quad I = 20.
\end{aligned}$$

# Appendix B

## Numerical Methods

In this appendix, we give details on the numerical methods we used to compute the Floquet multipliers, PRC, and IRC, and solve the Euler Lagrange equations and the full model equations.

### B.1 Computation of PRC

For the normal form of the Hopf bifurcation, we can compute the PRC and its derivative w.r.t.  $\theta$  analytically, see, e.g., [18]. For computing the PRCs (and their derivatives w.r.t.  $\theta$ ) of the YNI, thalamic neuron, and the clock gene regulation model, we use the XPP package [44], which is widely used by the community working on nonlinear oscillators. This package solves the appropriate adjoint equation backward in time along the periodic orbit to compute the PRC as a function of time. We scale the PRC computed by this package by  $\omega$ , as we consider PRC as  $\mathcal{Z}(\theta) = \frac{\partial \theta}{\partial \mathbf{x}}$ , whereas the computed PRC from the XPP package is  $\tilde{\mathcal{Z}}(t) = \frac{\partial t}{\partial \mathbf{x}}$ . Note that the XPP computes the derivative of the PRC w.r.t. time  $\left(\dot{\tilde{\mathcal{Z}}}(t) = \frac{\partial^2 t}{\partial \mathbf{x} \partial t}\right)$ , which is numerically equivalent to its derivative w.r.t.  $\theta$   $\left(\mathcal{Z}'(\theta) = \frac{\partial^2 \theta}{\partial \mathbf{x} \partial \theta}\right)$ . The XPP package gives the PRC and its derivative as a timeseries. After appropriately scaling the time series, we write them as an analytical expression of  $\theta$  by approximating them as a finite Fourier series, to be used in the numerical computation of the Euler-Lagrange equations.

### B.2 Computation of Floquet multipliers

Once the PRC has been computed, we choose an arbitrary point on the periodic orbit as  $\theta = 0$ , and approximate the isochron  $\Gamma_0$  as an  $n - 1$  dimensional hyperplane orthogonal to the PRC at that point. To compute the Jacobian  $DF$ , we compute  $\mathbf{x}_\Gamma^j$  (as defined beneath equation (2.14) in the main text) for a large  $j$ , for a number of initial conditions  $\mathbf{x}_0$  spread out on the isochron. Eigenvector decomposition of  $DF$  gives us the Floquet

multipliers of the periodic orbit and the corresponding Floquet exponents  $k_i$ . Note that for planar systems, the nontrivial Floquet exponent can be directly computed from the divergence of the vector field as [52]

$$k = \frac{\int_0^T \nabla \cdot F(\mathbf{x}^\gamma(t)) dt}{T}. \quad (\text{B.1})$$

### B.3 Two point BVP with Newton Iteration

We calculate the IRC and solve the Euler-Lagrange equations as a two point boundary value problem using Newton iteration in Chapter 4, which we briefly summarize here. Consider a general two point boundary value problem

$$\dot{y} = f(t, y), \quad y \in \mathbb{R}^n, \quad 0 \leq t \leq b, \quad (\text{B.2})$$

with the linear boundary condition

$$B_0 y(0) + B_b y(b) = a, \quad B_0, B_b \in \mathbb{R}^{n \times n}.$$

To solve such a boundary value problem, we integrate equation (B.13) with the initial guess  $c = y(0)$ , and calculate the function  $g(c)$ :

$$g(c) = B_0 c + B_b y(b) - a,$$

where  $y(b)$  is the solution at time  $b$  with the initial condition  $c$ . If we had chosen the correct initial condition  $c$ ,  $g(c)$  would be 0. Based on the current guess  $c^\nu$ , and the  $g(c^\nu)$  value, we choose the next initial condition by the Newton Iteration as

$$c^{\nu+1} = c^\nu - \left( \frac{\partial g}{\partial c} \Big|_{c^\nu} \right)^{-1} g(c^\nu). \quad (\text{B.3})$$

We compute the Jacobian  $J = \frac{\partial g}{\partial c} \Big|_{c^\nu}$  numerically as

$$J_i = \frac{g^+ - g^-}{2\epsilon},$$

where

$$\begin{aligned} g^+ &= g(c^\nu + e_i \epsilon), \\ g^- &= g(c^\nu - e_i \epsilon), \end{aligned}$$

$J_i$  is the  $i^{\text{th}}$  column of  $J$ ,  $\epsilon$  is a small number, and  $e_i$  is a column vector with 1 in the  $i^{\text{th}}$  position and 0 elsewhere.

### B.3.1 Computation of IRC

To calculate the IRC, we first compute and save the periodic solution  $\mathbf{x}^\gamma(t)$  using Matlab's ODE solver ode45 with a relative error tolerance of  $3e - 12$ , and an absolute error tolerance of  $1e - 15$ . The next step is to solve the adjoint equation

$$\dot{\mathcal{I}} = (k_i I - DF(\mathbf{x}^\gamma(t))^T) \mathcal{I}, \quad 0 \leq t \leq T,$$

with periodic boundary conditions

$$\mathcal{I}(0) = \mathcal{I}(T).$$

We choose an initial guess  $\mathcal{I}(0)$ , and integrate the adjoint equation using Matlab's ODE solver ode45 with a relative error tolerance of  $3e - 12$ , and an absolute error tolerance of  $1e - 15$ . For Newton iteration, we take

$$c^\nu = \mathcal{I}(0), \tag{B.4}$$

$$\begin{aligned} g(c^\nu) &= \underbrace{I}_{B_0} \mathcal{I}(0) - \underbrace{I}_{B_b} \mathcal{I}(T), \\ \Rightarrow g(c^\nu) &= \mathcal{I}(0) - \mathcal{I}(T), \end{aligned} \tag{B.5}$$

$$\left. \frac{\partial g}{\partial c} \right|_{c^\nu} = I - J, \tag{B.6}$$

where  $I$  is the identity matrix, and  $J$  is the Jacobian matrix

$$J = \frac{\partial \mathcal{I}(T)}{\partial \mathcal{I}(0)},$$

which we compute numerically. We use equations (B.4) - (B.6) together with equation (B.14) to compute the next initial condition. Once a periodic solution is obtained, the computed IRC is scaled by the normalization condition  $\nabla_{\mathbf{x}_0} \psi_i \cdot v_i = 1$  [30]. Its derivative w.r.t.  $\theta$  is obtained numerically by a central difference scheme

$$\mathcal{I}'(\theta_i) = \frac{\mathcal{I}(\theta_{i+1}) - \mathcal{I}(\theta_{i-1})}{\theta_{i+1} - \theta_{i-1}}.$$

The obtained IRC and its derivative w.r.t.  $\theta$  are written as analytical expressions of  $\theta$  by a finite Fourier series approximation, which is used in the computation of the Euler-Lagrange equations.

### B.3.2 Solving Euler-Lagrange equations

For Euler Lagrange equations based on augmented phase reduction, we set the boundary conditions as  $\theta(0) = 0$ ,  $\theta(T_1) = 2\pi$ ,  $\psi(0) = 0$ ,  $\psi(T_1) = 0$ . We can write this as a

two point boundary value problem with the function  $g$  as

$$g(c) = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{B_0} \underbrace{\begin{bmatrix} \theta(0) \\ \psi(0) \\ \lambda_1(0) \\ \lambda_2(0) \end{bmatrix}}_c + \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}}_{B_b} \begin{bmatrix} \theta(T_1) \\ \psi(T_1) \\ \lambda_1(T_1) \\ \lambda_2(T_1) \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 2\pi \\ 0 \end{bmatrix},$$

$$\Rightarrow g(c) = \begin{bmatrix} 0 \\ 0 \\ \theta(T_1) - 2\pi \\ \psi(T_1) - 0 \end{bmatrix}.$$

Since  $\theta(0)$ , and  $\psi(0)$  are fixed by our problem,  $g$  can be influenced by changing  $\lambda_1(0)$  and  $\lambda_2(0)$  only. So we get the following matrices for Newton Iteration:

$$c^\nu = \begin{bmatrix} \lambda_1(0) \\ \lambda_2(0) \end{bmatrix}, \quad (\text{B.7})$$

$$g(c^\nu) = \begin{bmatrix} \theta(T_1) - 2\pi \\ \psi(T_1) \end{bmatrix}, \quad (\text{B.8})$$

$$\left. \frac{\partial g}{\partial c} \right|_{c^\nu} = \begin{bmatrix} \frac{\partial \theta(T_1)}{\partial \lambda_1(0)} & \frac{\partial \theta(T_1)}{\partial \lambda_2(0)} \\ \frac{\partial \psi(T_1)}{\partial \lambda_1(0)} & \frac{\partial \psi(T_1)}{\partial \lambda_2(0)} \end{bmatrix}. \quad (\text{B.9})$$

In a similar way, we get the following matrices for Euler-Lagrange equations based on standard phase reduction:

$$c^\nu = \lambda_1(0), \quad (\text{B.10})$$

$$g(c^\nu) = \theta(T_1) - 2\pi, \quad (\text{B.11})$$

$$\left. \frac{\partial g}{\partial c} \right|_{c^\nu} = \frac{\partial \theta(T_1)}{\partial \lambda_1(0)}. \quad (\text{B.12})$$

All the integrations are done with Matlab ODE solver ode45 with relative error tolerance  $\leq 1e - 10$  and absolute error tolerance  $\leq 1e - 10$ .

## B.4 Two point BVP with modified Newton Iteration

We solve the Euler-Lagrange equations as a two point boundary value problem using a modified Newton iteration method, which we briefly summarize. Consider a general two point boundary value problem

$$\dot{y} = f(t, y), \quad y \in \mathbb{R}^n, \quad 0 \leq t \leq \tau, \quad (\text{B.13})$$

with the linear boundary condition

$$B_0 y(0) + B_\tau y(\tau) = a, \quad B_0, B_\tau \in \mathbb{R}^{n \times n}.$$

To solve such a boundary value problem, we integrate equation (B.13) with the initial guess  $c = y(0)$ , and calculate the function  $g(c)$ :

$$g(c) = B_0 c + B_\tau y(\tau) - a,$$

where  $y(\tau)$  is the solution at time  $\tau$  with the initial condition  $c$ . If we had chosen the correct initial condition  $c$ ,  $g(c)$  would be 0. Based on the current guess  $c^\nu$ , and the  $g(c^\nu)$  value, we choose the next initial condition by the modified Newton Iteration as an element-wise update

$$c_i^{\nu+1} = c_i^\nu - \left( \frac{\partial g_i}{\partial c_i} \Big|_{c^\nu} \right)^{-1} g_i(c^\nu), \quad \text{for } i = 1, \dots, n \quad (\text{B.14})$$

where  $g_i$ , and  $c_i^\nu$  represent the  $i^{\text{th}}$  element of vectors  $g$ , and  $c^\nu$  respectively. We compute the derivative  $J_{ii} = \frac{\partial g_i}{\partial c_i} \Big|_{c^\nu}$  numerically as

$$J_{ii} = \frac{g_i^+ - g_i^-}{2\epsilon},$$

where

$$\begin{aligned} g_i^+ &= g_i(c^\nu + e_i \epsilon), \\ g_i^- &= g_i(c^\nu - e_i \epsilon), \end{aligned}$$

$\epsilon$  is a small number, and  $e_i$  is a column vector with 1 in the  $i^{\text{th}}$  position and 0 elsewhere.

### B.4.1 Solving Euler-Lagrange equations

For the Euler-Lagrange equations devised in Section 5.7,  $A_k(0)$ , and  $B_k(0)$  are fixed by the initial distribution, so the only way to control the distribution is by choosing appropriate values of  $\lambda_{kA}(0)$  and  $\lambda_{kB}(0)$ . Thus our BVP can be reduced to  $2N - 2$  dimensions even though the Euler-Lagrange equations are  $4N - 4$  dimensional. The  $i^{\text{th}}$  element of the vector  $c$  is taken as

$$c_i = \begin{cases} \lambda_{kA}(0), & \text{for } i = k = 1, \dots, N - 1 \\ \lambda_{kB}(0), & \text{for } i = k + N - 1 = N, \dots, 2N - 2. \end{cases}$$

The  $i^{\text{th}}$  element of the vector  $g(c)$  for  $i = k = 1, \dots, N - 1$  is taken as

$$g_i(c) = A_k(0) + A_k(\tau) - \frac{1}{\pi} \int_0^{2\pi} (\rho(\theta, 0) + \rho_f(\theta, \tau)) \cos(k\theta) d\theta,$$

and, for  $i = k + N - 1 = N, \dots, 2N - 2$ ,

$$g_i(c) = B_k(0) + B_k(\tau) - \frac{1}{\pi} \int_0^{2\pi} (\rho(\theta, 0) + \rho_f(\theta, \tau)) \sin(k\theta) d\theta.$$

The derivative  $J_{ii}$  is given as

$$\frac{\partial g_i}{\partial c_i} = \begin{cases} \frac{\partial A_k(\tau)}{\partial \lambda_{kA}(0)}, & \text{for } i = k = 1, \dots, N - 1 \\ \frac{\partial B_k(\tau)}{\partial \lambda_{kB}(0)}, & \text{for } i = k + N - 1 = N, \dots, 2N - 2. \end{cases}$$

This information is used in equation (B.14) to iteratively find the appropriate value of the vector  $c$ .

# Bibliography

- [1] A. Pikovsky, M. Rosenblum, and J. Kurths, *Synchronization: A Universal Concept in Nonlinear Sciences*. University Press, 2003.
- [2] Y. Kuramoto, *Phase-and center-manifold reductions for large populations of coupled oscillators with application to non-locally coupled systems*, *International Journal of Bifurcation and Chaos* **7** (1997), no. 04 789–805.
- [3] A. Winfree, *Biological rhythms and the behavior of populations of coupled oscillators*, *Journal of Theoretical Biology* **16** (1967), no. 1 15–42.
- [4] J. Keener and J. Sneyd, *Mathematical Physiology*. Springer-Verlag New York, Inc., New York, NY, USA, 1998.
- [5] D. C. Michaels, E. P. Matyas, and J. Jalife, *Mechanisms of sinoatrial pacemaker synchronization: a new hypothesis.*, *Circulation Research* **61** (1987), no. 5 704–714, [<http://circres.ahajournals.org/content/61/5/704.full.pdf>].
- [6] C. S. Peskin and C. I. of Mathematical Sciences, *Mathematical aspects of heart physiology*. Courant Institute of Mathematical Sciences, New York University, 1975.
- [7] C. Gray, P. König, A. Engel, and W. Singer, *Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties*, *Nature* **338** (1989), no. 6213 334–337.
- [8] R. Friedrich and M. Stopfer, *Recent dynamics in olfactory population coding*, *Current Opinion in Neurobiology* **11** (2001), no. 4 468–474.
- [9] G. Stent, *A physiological mechanism for Hebb’s postulate of learning*, *Proceedings of the National Academy of Sciences* **70** (1973), no. 4 997–1001.
- [10] W. Klimesch, *Memory processes, brain oscillations and EEG synchronization*, *International Journal of Psychophysiology* **24** (1996), no. 1 61–100.
- [11] A. Kane, W. Hutchison, M. Hodaie, A. Lozano, and J. Dostrovsky, *Enhanced synchronization of thalamic theta band local field potentials in patients with essential tremor*, *Experimental Neurology* **217** (2009), no. 1 171–176.

- [12] A. Kühn, A. Tsui, T. Aziz, N. Ray, C. Brücke, A. Kupsch, G.-H. Schneider, and P. Brown, *Pathological synchronisation in the subthalamic nucleus of patients with Parkinson's disease relates to both bradykinesia and rigidity*, *Experimental Neurology* **215** (2009), no. 2 380–387.
- [13] T. Matchen and J. Moehlis, *Real-time stabilization of neurons into clusters*, in *2017 American Control Conference (ACC)*, pp. 2805–2810, May, 2017.
- [14] T. Matchen and J. Moehlis, *Phase model-based neuron stabilization into arbitrary clusters*, *Journal of Computational Neuroscience* **44** (Jun, 2018) 363–378.
- [15] D. Wilson and J. Moehlis, *Clustered desynchronization from high-frequency deep brain stimulation*, *PLoS Comput. Biol.* **11** (2015), no. 12 e1004673.
- [16] B. Monga and J. Moehlis, *Optimal phase control of biological oscillators using augmented phase reduction*, *Biological Cybernetics* **113** (Apr, 2019) 161–178.
- [17] J. Guckenheimer, *Isochrons and phaseless sets*, *Journal of Mathematical Biology* **1** (1975), no. 3 259–273.
- [18] E. Brown, J. Moehlis, and P. Holmes, *On the phase reduction and response dynamics of neural oscillator populations*, *Neural Comp.* **16** (2004) 673–715.
- [19] A. Winfree, *The Geometry of Biological Time, Second Edition*. Springer, New York, 2001.
- [20] J. Moehlis, E. Shea-Brown, and H. Rabitz, *Optimal inputs for phase models of spiking neurons*, *ASME Journal of Computational and Nonlinear Dynamics* **1** (2006), no. 4 358–367.
- [21] D. Wilson and J. Moehlis, *Optimal chaotic desynchronization for neural populations*, *SIAM Journal on Applied Dynamical Systems* **13** (2014), no. 1 276.
- [22] A. Zlotnik, Y. Chen, I. Kiss, H. Tanaka, and J.-S. Li, *Optimal waveform for fast entrainment of weakly forced nonlinear oscillators*, *Physical Review Letters* **111** (2013), no. 2 024102.
- [23] P. Tass, *Phase Resetting in Medicine and Biology: Stochastic Modelling and Data Analysis*, vol. 172. Springer-Verlag, Berlin, 2007.
- [24] D. Minors, J. Waterhouse, and A. Wirz-Justice, *A human phase-response curve to light*, *Neuroscience Letters* **133** (1991), no. 1 36–40.
- [25] T. Stigen, P. Danzl, J. Moehlis, and T. Netoff, *Controlling spike timing and synchrony in oscillatory neurons*, *Journal of Neurophysiology* **105** (2011), no. 5 2074.

- [26] A. Nabi, T. Stigen, J. Moehlis, and T. Netoff, *Minimum energy control for in vitro neurons*, *Journal of Neural Engineering* **10** (2013), no. 3 036005.
- [27] R. Snari, M. Tinsley, D. Wilson, S. Faramarzi, T. Netoff, J. Moehlis, and K. Showalter, *Desynchronization of stochastically synchronized chemical oscillators*, *Chaos: An Interdisciplinary Journal of Nonlinear Science* **25** (2015), no. 12 123116.
- [28] T. Netoff, M. Schwemmer, and T. Lewis, *Experimentally estimating phase response curves of neurons: theoretical and practical issues*, in *Phase Response Curves in Neuroscience* (N. Schultheiss, A. Prinz, and R. Butera, eds.), pp. 95–129. Springer, New York, 2012.
- [29] J. Guckenheimer and P. J. Holmes, *Nonlinear Oscillations, Dynamical Systems and Bifurcations of Vector Fields*. Springer-Verlag, New York, New York, 1983.
- [30] D. Wilson and J. Moehlis, *Isostable reduction of periodic orbits*, *Physical Review E* **94** (2016), no. 5 052213.
- [31] A. Mauroy, I. Mezić, and J. Moehlis, *Isostables, isochrons, and Koopman spectrum for the action–angle representation of stable fixed point dynamics*, *Physica D: Nonlinear Phenomena* **261** (2013) 19–30.
- [32] Z.-S. Hou and Z. Wang, *From model-based control to data-driven control: Survey, classification and perspective*, *Information Sciences* **235** (2013) 3 – 35. Data-based Control, Decision, Scheduling and Fault Diagnostics.
- [33] B. D. O. Anderson, *Failures of adaptive control theory and their resolution*, *Commun. Inf. Syst.* **05** (2005), no. 1 1–20.
- [34] E. Kaiser, J. N. Kutz, and S. L. Brunton, *Sparse identification of nonlinear dynamics for model predictive control in the low-data limit*, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **474** (2018), no. 2219 20180335.
- [35] I. Malkin, *The Methods of Lyapunov and Poincare in the Theory of Nonlinear Oscillations*. Gostekhizdat, Moscow-Leningrad, 1949.
- [36] E. A. Coddington and N. Levinson, *Theory of Ordinary Differential Equations*. McGraw-Hill, New York, 1955.
- [37] Y. Kuramoto, *Chemical Oscillations, Waves, and Turbulence*. Springer, Berlin, 1984.
- [38] B. Monga, D. Wilson, T. Matchen, and J. Moehlis, *Phase reduction and phase-based optimal control for biological systems: A tutorial*, *Biological Cybernetics* **113** (Apr, 2019) 11–46.

- [39] D. Hansel, G. Mato, and C. Meunier, *Synchrony in excitatory neural networks*, *Neural Comp.* **7** (1995) 307–337.
- [40] B. Ermentrout, *Type I membranes, phase resetting curves, and synchrony*, *Neural Computation* **8** (1996), no. 5 979–1001.
- [41] L. Glass and M. C. Mackey, *From Clocks to Chaos: The Rhythms of Life*. Princeton University Press, Princeton, 1988.
- [42] G. Ermentrout and N. Kopell, *Multiple pulse interactions and averaging in coupled neural oscillators*, *J. Math. Biol.* **29** (1991) 195–217.
- [43] F. C. Hoppensteadt and E. M. Izhikevich, *Weakly Connected Neural Networks*. Springer-Verlag, New York, 1997.
- [44] B. Ermentrout, *Simulating, Analyzing, and Animating Dynamical Systems: A Guide to XPPAUT for Researchers and Students*, vol. 14. SIAM, Philadelphia, 2002.
- [45] J. Rubin and D. Terman, *High frequency stimulation of the subthalamic nucleus eliminates pathological thalamic rhythmicity in a computational model*, *Journal of Computational Neuroscience* **16** (2004), no. 3 211–235.
- [46] S. Shirasaka, W. Kurebayashi, and H. Nakao, *Phase-amplitude reduction of transient dynamics far from attractors for limit-cycling systems*, *Chaos: An Interdisciplinary Journal of Nonlinear Science* **27** (2017), no. 2 023119.
- [47] O. Castejón, A. Guillamon, and G. Huguet, *Phase-amplitude response functions for transient-state stimuli*, *The Journal of Mathematical Neuroscience* **3** (2013), no. 1 1.
- [48] K. Wedgwood, K. Lin, R. Thul, and S. Coombes, *Phase-amplitude descriptions of neural oscillator models*, *The Journal of Mathematical Neuroscience* **3** (Jan, 2013) 2.
- [49] J. Guckenheimer, *Phase portraits of planar vector fields: Computer proofs*, *Experimental Mathematics* **4** (1995), no. 2 153–165.
- [50] J. Moehlis, *Improving the precision of noisy oscillators*, *Physica D: Nonlinear Phenomena* **272** (2014) 8–17.
- [51] E. Izhikevich, *Phase equations for relaxation oscillators*, *SIAM Journal on Applied Mathematics* **60** (2000), no. 5 1789–1804.
- [52] P. Glendinning, *Stability, Instability and Chaos: An Introduction to the Theory of Nonlinear Differential Equations*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 1994.

- [53] B. Monga and J. Moehlis, *Augmented phase reduction for homoclinic bifurcation and relaxation oscillators*, *In preparation*.
- [54] Y. Kuznetsov, *Elements of Applied Bifurcation Theory*, vol. 112. Springer-Verlag, New York, 2013.
- [55] A. Guillamon and G. Huguet, *A computational and geometric approach to phase resetting curves and surfaces*, *SIAM Journal on Applied Dynamical Systems* **8** (2009), no. 3 1005–1042.
- [56] N. Kopell and L. Howard, *Plane wave solutions to reaction-diffusion equations*, *Studies in Applied Mathematics* **52** (1973), no. 4 291–328.
- [57] G. Ermentrout and D. Terman, *Mathematical Foundations of Neuroscience*. Springer-Verlag New York, 2010.
- [58] Y. A. Kuznetsov, *Andronov-Hopf bifurcation*, *Scholarpedia* **1** (2006), no. 10 1858. revision 90964.
- [59] O. Rössler, *An equation for continuous chaos*, *Physics Letters A* **57** (1976), no. 5 397–398.
- [60] J. Guckenheimer and Y. A. Kuznetsov, *Bautin bifurcation*, *Scholarpedia* **2** (2007), no. 5 1853. revision 91035.
- [61] E. Izhikevich, *Neural excitability, spiking and bursting*, *International Journal of Bifurcation and Chaos* **10** (2000), no. 06 1171–1266.
- [62] C. Morris and H. Lecar, *Voltage oscillations in the barnacle giant muscle fiber*, *Biophysical Journal* **35** (1981), no. 1 193 – 213.
- [63] S. Wiggins, *Introduction to Applied Nonlinear Dynamical Systems and Chaos*. Springer-Verlag New York, 2003.
- [64] B. Sandstede, *Constructing dynamical systems having homoclinic bifurcation points of codimension two*, *Journal of Dynamics and Differential Equations* **9** (1997), no. 2 269–288.
- [65] T. Kanamaru, *Van der Pol oscillator*, *Scholarpedia* **2** (2007), no. 1 2202. revision #138698.
- [66] B. Van der Pol, *A theory of the amplitude of free and forced triode vibrations*, *Radio Review* **1** (1920), no. 1920 701–710.
- [67] A. Nabi, M. Mirzadeh, F. Gibou, and J. Moehlis, *Minimum energy desynchronizing control for coupled neurons*, *Journal of Computational Neuroscience* **34** (2013), no. 2 259–271.

- [68] A. Nabi and J. Moehlis, *Time optimal control of spiking neurons*, *Journal of Mathematical Biology* **64** (2012), no. 6 981–1004.
- [69] R. Wever, *Use of light to treat jet lag: differential effects of normal and bright artificial light on human circadian rhythms*, *Annals of the New York Academy of Sciences* **453** (1985), no. 1 282–304.
- [70] C. Czeisler, M. Johnson, J. Duffy, E. Brown, J. Ronda, and R. Kronauer, *Exposure to bright light and darkness to treat physiologic maladaptation to night work*, *New England Journal of Medicine* **322** (1990), no. 18 1253–1259.
- [71] C. Eastman and S. Martin, *How to use light and dark to produce circadian adaptation to night shift work*, *Annals of Medicine* **31** (1999), no. 2 87–98.
- [72] J. Marsden and M. McCracken, *The Hopf Bifurcation and Its Applications*, vol. 19. Springer-Verlag, New York, 2012.
- [73] E. M. Izhikevich, *Dynamical Systems in Neuroscience*. MIT press, Cambridge, 2007.
- [74] Q. Wang and L.-S. Young, *Strange attractors in periodically-kicked limit cycles and Hopf bifurcations*, *Communications in Mathematical Physics* **240** (Sep, 2003) 509–529.
- [75] G. Mines, *On dynamic equilibrium in the heart*, *The Journal of Physiology* **46** (1913), no. 4-5 349–383.
- [76] J. Pastore, S. Girouard, K. Laurita, F. Akar, and D. Rosenbaum, *Mechanism linking t-wave alternans to the genesis of cardiac fibrillation*, *Circulation* **99** (1999), no. 10 1385–1394.
- [77] G. Hall and D. Gauthier, *Experimental control of cardiac muscle alternans*, *Physical Review Letters* **88** (2002), no. 19 198102.
- [78] D. Christini, M. Riccio, C. Cuiianu, J. Fox, A. Karma, and R. Gilmour Jr, *Control of electrical alternans in canine cardiac Purkinje fibers*, *Physical Review Letters* **96** (2006), no. 10 104101.
- [79] K. Hall, D. Christini, M. Tremblay, J. Collins, L. Glass, and J. Billette, *Dynamic control of cardiac alternans*, *Physical Review Letters* **78** (1997), no. 23 4518.
- [80] D. Wilson and J. Moehlis, *Extending phase reduction to excitable media: theory and applications*, *SIAM Review* **57** (2015), no. 2 201–222.
- [81] W.-J. Rappel, F. Fenton, and A. Karma, *Spatiotemporal control of wave instabilities in cardiac tissue*, *Physical Review Letters* **83** (1999), no. 2 456.

- [82] M. Guevara, G. Ward, A. Shrier, and L. Glass, *Electrical alternans and period doubling bifurcations*, *IEEE Comp. Cardiol* **562** (1984) 167–170.
- [83] J. Nolasco and R. Dahlen, *A graphic method for the study of alternation in cardiac action potentials*, *Journal of Applied Physiology* **25** (1968), no. 2 191–196.
- [84] K. Yanagihara, A. Noma, and H. Irisawa, *Reconstruction of sino-atrial node pacemaker potential based on the voltage clamp experiments.*, *The Japanese Journal of Physiology* **30** (1980), no. 6 841–857.
- [85] A. Benabid, P. Pollak, D. Hoffmann, C. Gervason, M. Hommel, J. Perret, J. De Rougemont, and D. Gao, *Long-term suppression of tremor by chronic stimulation of the ventral intermediate thalamic nucleus*, *The Lancet* **337** (1991), no. 8738 403–406.
- [86] A. Benabid, S. Chabardes, J. Mitrofanis, and P. Pollak, *Deep brain stimulation of the subthalamic nucleus for the treatment of Parkinson’s disease*, *The Lancet Neurology* **8** (2009), no. 1 67–81.
- [87] C. Wilson, B. Beverlin, and T. Netoff, *Chaotic desynchronization as the therapeutic mechanism of deep brain stimulation*, *Front. Syst. Neurosci.* **5** (2011).
- [88] P. Tass, *A model of desynchronizing deep brain stimulation with a demand-controlled coordinated reset of neural subpopulations*, *Biological Cybernetics* **89** (2003), no. 2 81–88.
- [89] M. Rea, A. Bierman, M. Figueiro, and J. Bullough, *A new approach to understanding the impact of circadian disruption on human health*, *Journal of Circadian Rhythms* **6** (2008), no. 1 7.
- [90] E. Klerman, *Clinical aspects of human circadian rhythms*, *Journal of Biological Rhythms* **20** (2005), no. 4 375–386.
- [91] C. Czeisler, R. Kronauer, J. Allan, J. Duffy, M. Jewett, E. Brown, and J. Ronda, *Bright light induction of strong (type 0) resetting of the human circadian pacemaker*, *Science* **2** (1989) 4.
- [92] D. Dean, D. Forger, E. Klerman, *et. al.*, *Taking the lag out of jet lag through model-based schedule design*, *PLoS Comput Biol* **5** (2009), no. 6 e1000418.
- [93] D. Forger and D. Paydarfar, *Starting, stopping, and resetting biological oscillators: in search of optimum perturbations*, *Journal of Theoretical Biology* **230** (2004), no. 4 521–532.
- [94] J. Zhang, J. Wen, and A. Julius, *Optimal circadian rhythm control with light input for rapid entrainment and improved vigilance*, in *Proceedings of the 51st IEEE Conference on Decision and Control (CDC)*, pp. 3007–3012, IEEE, 2012.

- [95] D. Gonze, S. Bernard, C. Waltermann, A. Kramer, and H. Herzel, *Spontaneous synchronization of coupled circadian oscillators*, *Biophysical Journal* **89** (2005), no. 1 120–129.
- [96] A. Gad, G. David, S. Markus, R. Hans, D. Charna, K. Florian, I. M. Rau, W. Frederick, and S. Ueli, *{SIRT1} regulates circadian clock gene expression through {PER2} deacetylation*, *Cell* **134** (2008), no. 2 317 – 328.
- [97] R. Honeycutt, *Stochastic runge-kutta algorithms. i. white noise*, *Phys. Rev. A* **45** (Jan, 1992) 600–603.
- [98] K. Kuritz, W. Halter, and F. Allgöwer, *Passivity-based ensemble control for cell cycle synchronization*, pp. 1–13. Springer International Publishing, Emerging Applications of Control and Systems Theory, 2018.
- [99] P. Danzl and J. Moehlis, *Event-based feedback control of nonlinear oscillators using phase response curves*, in *Decision and Control, 2007 46th IEEE Conference on*, pp. 5806–5811, IEEE, 2007.
- [100] A. Nabi and J. Moehlis, *Nonlinear hybrid control of phase models for coupled oscillators*, in *American Control Conference (ACC), 2010*, pp. 922–923, IEEE, 2010.
- [101] P. Tass, *Effective desynchronization by means of double-pulse phase resetting*, *EPL (Europhysics Letters)* **53** (2001), no. 1 15.
- [102] B. Monga, G. Froyland, and J. Moehlis, *Synchronizing and desynchronizing neural populations through phase distribution control*, in *2018 American Control Conference (ACC)*, pp. 2808–2813, June, 2018.
- [103] D. Wilson and J. Moehlis, *Isostable reduction with applications to time-dependent partial differential equations*, *Phys. Rev. E* **94** (Jul, 2016) 012211.
- [104] K. Kuritz, S. Zeng, and F. Allgöwer, *Ensemble controllability of cellular oscillators*, *IEEE Control Systems Letters* **3** (April, 2019) 296–301.
- [105] B. Monga and J. Moehlis, *Phase distribution control of a population of oscillators*, *Physica D: Nonlinear Phenomena* **398** (2019) 115 – 129.
- [106] D. Best and N. Fisher, *Efficient simulation of the von mises distribution*, *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **28** (1979), no. 2 152–157.
- [107] P. Berens, *Circstat: A Matlab toolbox for circular statistics*, *Journal of Statistical Software, Articles* **31** (2009), no. 10 1–21.

- [108] A. L. Hodgkin and A. F. Huxley, *A quantitative description of membrane current and its application to conduction and excitation in nerve*, *The Journal of Physiology* **117** (1952), no. 4 500–544.
- [109] C. Shatz, *The developing brain*, *Scientific American* **267** (1992), no. 3 60–67.
- [110] O. Popovych, S. Yanchuk, and P. Tass, *Self-organized noise resistance of oscillatory neural networks with spike timing-dependent plasticity*, *Scientific Reports* **3**.
- [111] I. Adamchic, C. Hauptmann, U. Barnikol, N. Pawelczyk, O. Popovych, T. Barnikol, A. Silchenko, J. Volkmann, G. Deuschl, W. Meissner, M. Maarouf, V. Sturm, H. Freund, and P. Tass, *Coordinated reset neuromodulation for parkinson’s disease: Proof-of-concept study*, *Movement Disorders* **29** (2014), no. 13 1679–1684.
- [112] D. Hebb, *The Organization of Behavior*. Wiley: New York, 1949.
- [113] G. Bi and M. Poo, *Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type*, *Journal of Neuroscience* **18** (1998), no. 24 10464–10472.
- [114] G. Bi and M. Poo, *Synaptic modification by correlated activity: Hebb’s postulate revisited*, *Annual Review of Neuroscience* **24** (2001), no. 1 139–166.
- [115] H. Markram, W. Gerstner, and P. Sjstrm, *Spike-Timing-Dependent Plasticity: A Comprehensive Overview*. Frontiers Research Foundation, 2012.
- [116] L. I. Zhang, H. Tao, C. Holt, W. Harris, and M. Poo, *A critical window for cooperation and competition among developing retinotectal synapses*, *Nature* **395**.
- [117] C. Gardiner, P. Zoller, and P. Zoller, *Quantum Noise: A Handbook of Markovian and Non-Markovian Quantum Stochastic Methods with Applications to Quantum Optics*. Springer Series in Synergetics. Springer, 2004.
- [118] M. I. Freidlin and A. D. Wentzell, *Random Perturbations*, pp. 15–43. Springer New York, New York, NY, 1998.
- [119] I. Fantoni and R. Lozano, *Non-linear Control for Underactuated Mechanical Systems*. Springer-Verlag London, 2002.
- [120] M. Reyhanoglu, A. van der Schaft, N. H. Mcclamroch, and I. Kolmanovsky, *Dynamics and control of a class of underactuated mechanical systems*, *IEEE Transactions on Automatic Control* **44** (1999), no. 9 1663–1671.

- [121] M. Spong, *Energy based control of a class of underactuated mechanical systems*, *IFAC Proceedings Volumes* **29** (1996), no. 1 2828 – 2832. 13th World Congress of IFAC, 1996, San Francisco USA, 30 June - 5 July.
- [122] K. Hunt, D. Sbarbaro, R. bikowski, and P. Gawthrop, *Neural networks for control systemsa survey*, *Automatica* **28** (1992), no. 6 1083 – 1112.
- [123] A. G. Barto, R. S. Sutton, and C. W. Anderson, *Neuronlike adaptive elements that can solve difficult learning control problems*, *IEEE Transactions on Systems, Man, and Cybernetics* **SMC-13** (Sep., 1983) 834–846.
- [124] M. I. Jordan and T. M. Mitchell, *Machine learning: Trends, perspectives, and prospects*, *Science* **349** (2015), no. 6245 255–260.
- [125] B. Monga and J. Moehlis, *Supervised learning algorithms for controlling underactuated dynamical systems*, *arXiv preprint arXiv:1909.11119*.
- [126] A. Longtin, A. Bulsara, and F. Moss, *Time-interval sequences in bistable systems and the noise-induced transmission of information by sensory neurons*, *Phys. Rev. Lett.* **67** (Jul, 1991) 656–659.
- [127] D. Wilson and J. Moehlis, *An energy-optimal methodology for synchronization of excitable media*, *SIAM Journal on Applied Dynamical Systems* **13** (2014), no. 2 944–957.
- [128] C. Bowden, M. Ciftan, and H. Robl, *Optical Bistability*. Springer US, 1981.
- [129] R. Lefever and W. Horsthemke, *Bistability in fluctuating environments. implications in tumor immunology*, *Bulletin of Mathematical Biology* **41** (Jul, 1979) 469–490.
- [130] T. S. Gardner, C. R. Cantor, and J. J. Collins, *Construction of a genetic toggle switch in escherichia coli*, *Nature* **403** (2000), no. 2219 339–342.
- [131] A. Fiasconaro, A. Ochab-Marcinek, B. Spagnolo, and E. Gudowska-Nowak, *Monitoring noise-resonant effects in cancer growth influenced by external fluctuations and periodic treatment*, *The European Physical Journal B* **65** (Oct, 2008) 435–442.
- [132] T. Kanamaru, *Duffing oscillator*, *Scholarpedia* **3** (2008), no. 3 6327. revision #91210.
- [133] J. Moehlis, *Canards for a reduction of the Hodgkin-Huxley equations*, *Journal of Mathematical Biology* **52** (Feb, 2006) 141–153.
- [134] D. Johnston and S. M.-S. Wu, *Foundations of Cellular Neurophysiology*. MIT Press, Cambridge, MA, 1995.

- [135] K. Pyragas, V. Pyragas, I. Z. Kiss, and J. L. Hudson, *Adaptive control of unknown unstable steady states of dynamical systems*, *Phys. Rev. E* **70** (Aug, 2004) 026215.
- [136] E. N. Lorenz, *Deterministic nonperiodic flow*, *Journal of the Atmospheric Sciences* **20** (1963), no. 2 130–141.