Formal Verification of Nonlinear Biological Systems

A Dissertation presented

by

Md. Ariful Islam

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

Stony Brook University

December 2015

Stony Brook University

The Graduate School

Md. Ariful Islam

We, the dissertation committe for the above candidate for the

Doctor of Philosophy degree, hereby recommend

acceptance of this dissertation

Scott A. Smolka - Dissertation Advisor
Professor, Computer Science

Allen Tannenbaum - Chairperson of Defense
Professor, Computer Science

Radu Grosu - Dissertation Co-Advisor
Research Professor, Computer Science

Sayan Mitra
Associate Professor, Electrical and Computer Engineering, UIUC

This dissertation is accepted by the Graduate School

Charles Taber
Dean of the Graduate School

Abstract of the Dissertation

Formal Verification of Nonlinear Biological Systems

by

Md. Ariful Islam

Doctor of Philosophy

in

Computer Science

Stony Brook University

2015

Abstraction and composition have proved to be particularly useful in extending the reach of formal verification. Abstraction reduces the size of the system under investigation by neglecting details irrelevant to the properties of interest. Compositionality allows us to decompose large-scale system into smaller components and verify each component individually and reason about the verification of entire system from verified components. Together, these two techniques permit us to substitute a component with its equivalent abstraction such that the overall system retains the property of interest.

In this thesis, we first show that in the context of the Iyer et al. (IMW) 67-variable cardiac myocycte model, it is possible to replace the detailed 13-state probabilistic model of the sodium channel dynamics with a much simpler Hodgkin-Huxley (HH)-like two-state model, while only incurring a bounded approximation error. We then extend our technique to the 10-state model of the fast recovering calcium-independent potassium channel. The basis of our results is the construction of an approximate bisimulation between the HH-type abstraction and the corresponding detailed ion channel model, both of which are input-controlled (voltage in this case) CTMCs.

We then present BFComp, an automated framework based on Sum-Of-Squares (SOS) optimization and delta-decidability over the reals to compute Bisimulation Functions (BF). BF formalizes the notion of abstraction in dynamical systems by establishing approximate equivalence between the original system and its abstract equivalent. In addition, BFComp framework supports compositional reasoning. In this work, we are particularly interested infeedback composition, where output of one component is the input to another, and vice versa. By appealing to small-gain theorem of BFs, it can be shown that an abstract component with lower complexity can replace a detailed component in a feedback composition. Such substitution is safe in the sense that approximation error between the detailed and abstract component will not be amplified in the feedback composition. We will illustrate the utility of BFComp on a canonical cardiac-cell model, showing that the four-variable Markovian model for the slowly activating potassium current component can be safely replaced by a one-variable Hodgkin-Huxley-type approximation.

Finally, we present what we believe to be the first formal verification of a biologically realistic (nonlinear ODE) model of a neural circuit in a multicellular organism: Tap Withdrawal (TW) in C. Elegans, the common roundworm. TW is a reflexive behavior exhibited by C. Elegans in response to vibrating the surface on which it is moving; the neural circuit underlying this response is the subject of this investigation. Specially, we perform reach-tube-based reachability analysis on the TW circuit model of Wicks et al. (1996) to estimate key model parameters. Underlying our approach is the use of Fan and Mitra's recently developed technique for automatically computing local discrepancy (convergence and divergence rates) of general nonlinear systems.

The results we obtain are a significant extension of those of Wicks et al. (1996), who equip their model with fixed parameter values that reproduce the predominant TW response they observed experimentally in a population of 590 worms. In contrast, our techniques allow us to much more fully explore the model's parameter space, identifying in the process the parameter ranges responsible for the predominant behavior as well as the non-dominant ones. The verification framework we developed to conduct this analysis is model-agnostic, and can thus be re-used on other complex nonlinear systems.

Dedication Page

This dissertation is dedicated to my parents, ABM Abdul Awal and Lutfun Nahar, for their enormous love, limitless sacrifices, endless support, and dedication.

Contents

List of Figures

List of Tables

List of Abbreviations

ODE Ordinary Differential Equation

IMW Iyer-Mazhari-Winslow

MM Minimal Model

AP Action Potential

HH Hodgkin-Huxley

VCE Voltage Clamp Experiment

CTMC Continuous Time Markov Chain

PEFT Parameter Estimation from Finite Traces

RFI Rate Function Identification

BF Bisimulation Function

IOS Input-to-Output Stability

SOS Sum-of-Squares

SOD Squared Output Difference

DEM Differential Equation Model

ISS Input-to-State Stability

IOS Input-to-Output Stability

SOSP Sum Of Square Programming

BF Bisimulation Function

PDE Partial Differential Equation

ODE Ordinary Differential Equation

SOD Squared Output Difference

TWC Tap Withdrawal Circuit

LP Linear Programming

CBF Candidate Bisimulation Function

CLF Candidate Lyapunov Function

List of Symbols

$\alpha_h(V)$ The rate of opening of an inactivating (h-type) subunit in an HH-type model.

$\alpha_m(V)$ The rate of opening of an activating (m-type) subunit in an HH-type model.

$\beta_h(V)$ The rate of closing of an inactivating (h-type) subunit in an HH-type model.

$\beta_m(V)$ The rate of closing of an activating (m-type) subunit in an HH-type model.

λ The degree of activation in an HH-type model, 3 for $\mathcal{H}_{Na}$ and 4 for $\mathcal{H}_K$.

$\mathcal{H}$ The HH model for Squid neurons.

$\mathcal{H}_K$ The 2-state HH-type abstraction identified for $\mathcal{M}_K$ using Parameter Estimation from Finite Traces (PEFT) and Rate Function Identification (RFI).

$\mathcal{H}_K^v$ The constant-voltage version of $\mathcal{H}_K$, where $V = v$.

$\mathcal{H}_{Na}$ The 2-state HH-type abstraction identified for $\mathcal{M}_{Na}$ using PEFT and RFI.

$\mathcal{H}_{Na}^v$ The constant-voltage version of $\mathcal{H}_{Na}$, where $V = v$.

$\mathcal{M}_K$ The 10-state voltage-controlled CTMC for the sodium current $I_{Kv4.3}$.

$\mathcal{M}_K^v$ The constant-voltage version of $\mathcal{M}_K$, where $V = v$.

$\mathcal{M}_{Na}$ The 13-state voltage-controlled CTMC for the sodium current I_{Na}.

$\mathcal{M}_{Na}^v$ The constant-voltage version of $\mathcal{M}_{Na}$, where $V = v$.

$\overline{g}_{Na}$ Maximum conductance of the sodium channel.

h Variable used in HH-type models for measuring the extent of inactivation.

$I_{Kv4.3}$ The potassium current, component of the I_{to1} current, and modeled using the voltage-controlled CTMC $\mathcal{M}_K$ in the IMW model.

I_{Na} Sodium current, modeled using the voltage-controlled CTMC $\mathcal{M}_{Na}$ in the IMW model.

m Variable used in HH-type models for measuring the extent of activation.

O_H Conductance of the HH-type channel, given by $\{m(V,t)\}^{\lambda} h(V,t)$.

O_K The conductance (observable output) of $\mathcal{M}_K$, given by $O_K = O$, the occupancy probability of the state labeled O.

O_{Na} The conductance (observable output) of $\mathcal{M}_{Na}$, given by $O_{Na} = O_1 + O_2$, the occupancy probabilities of the states labeled O_1 and O_2.

V Transmembrane potential of a cardiac myocyte.

V_{max} The maximum potential reached by cardiac myocytes during an AP.

V_{Na} Nernst potential for sodium.

V_{res} The resting potential for cardiac myocytes.

A_K The 10×10 rate matrix for $\mathcal{M}_K$.

A_{Na} The 13×13 rate matrix for $\mathcal{M}_{Na}$.

Acknowledgements

Over the past five years, I have been privileged to work with a highly intellectual, extremely supportive and enormously inspiring group of people. I have learned a lot from everyone of them, and would like to express my gratitude for their help and support.

I would like to begin by expressing my deepest gratitude to my advisers Prof. Scott A. Smolka and Prof. Radu Grosu. Prof. Smolka's patience, rigor, and effective communication style, combined with his unsurpassed knowledge of formal verification have enabled me to mature as a researcher. He will always be an inspiration. Prof. Grosu's passion and enthusiasm for research was contagious and motivational for me, even during the tough times of my research. I could not be prouder of my academic roots, and will endeavor to pass on the research values and the dreams that they have given to me.

My advisers and I collaborated with Prof. Sayan Mitra on model checking of a neural circuit in C. Elegans. Along with Prof. Smolka and Prof. Grosu, Prof. Mitra patiently guided me through our interdisciplinary research. I am grateful for all his help. I would like to thank Prof. Allen R. Tannenbaum, for agreeing to serve on my dissertation committee. I have learned something new from each of our meetings, and I look forward to working with him.

I would like to gratefully acknowledge the funding source that made my Ph.D. work possible. The research presented in this thesis was performed under the US National Science Foundation (NSF) Expedition in Computing, entitled "Computational Modeling and Analysis of Complex Systems", which is funded by the grant: NSF CCF-0926190 and another NFS project on CPS:Frontiers entitled as " Collaborative Research: Compositional, Approximate, and Quantitative Reasoning for Medical Cyber-Physical Systems", which is funded by the grant: NSF CNS-1446832.

During my initial years, Ezio Bartocci helped me with starting my my research. His diligence and dedication have continued to inspire me. I have always enjoyed our interactions and look forward to our sustained collaboration. I am immensely grateful to Abhishek Murthy, my fellow-graduate student. As a junior PhD-student, I have learnt a lot from him over the years. He not only helped in my research, but also gave me valuable advice over the years. I will always cherish the memories that I made with him over the years.

I am grateful to Prof. Flavio H. Fenton and Prof. Elizabeth M. Cherry for

their help on understanding cardiac models. Prof. Antoine Girard helped us with the compositionality results for cardiac cell dynamics. I am very thankful for his insights on the sum-of-squares computation procedure for Bisimulation Functions. My seniors Mohammad T. Irfan and Tushar Deshpande have been very helpful. The support of my fellow-Ph.D. students Richard Defrancisco, Junxing Yang, and Dung Phan has been instrumental during my Ph.D. They have helped me rehearse presentations, critiqued my papers, and given me vital research advice.

I am grateful to the department's academic and administrative staff: Betty Knittweis, Cynthia Scalzo and Kathy Germana. In the past five years, their cogent and efficient style of working has been critical for my success, and inspiring at the same time. I would like to mention, especially, Brian Tria, who was extremely helpful with the department's computing infrastructure. He has also been a very good friend over the years.

I would also like to thank all of my undergrad professors at Bangladesh University of Engineering and Technology. I feel that their influence and constant encouragement motivated me tremendously in pursuing my PhD.

I am extremely lucky to make many friends during my PhD. Their amiable presence and friendly company always kept me energetic and refreshed. I would like to thank them all.

Finally, I would like to give special thanks to my parents and family members. They have been a constant source of encouragement and inspiration all my life. Thank you.

Md. Ariful Islam
Stony Brook University
December 2015

Publications

1. Md. Ariful Islam, Richard DeFrancisco, Chuchu Fan, Radu Grosu, Sayan Mitra, Scott A. Smolka. Model Checking Tap Withdrawal in C. Elegans. In *Proceedings of the Fourth International Workshop on Hybrid Systems Biology* (HSB 2015), Madrid, Spain, September, 2015.

2. Abhishek Murthy, Md. Ariful Islam, Scott A. Smolka, and Radu Grosu. Computing Compositional Proofs of Input-to-Output Stability Using SOS Optimization and δ-Decidability. Invited to submit in *Nonlinear Analysis: Hybrid Systems.*

3. Konstantin Selyunin, Denise Ratasich, Ezio Bartocci, Md. Ariful Islam, Scott Smolka, Radu Grosu. Neural Programming: Towards Adaptive Control in Cyber-Physical Systems. In *Proceedings of the 54th IEEE Conference on Decision and Control* (CDC 2015), Osaka, Japan, December, 2015.

4. Abhishek Murthy, Md. Ariful Islam, Scott A. Smolka, and Radu Grosu. Computing bisimulation functions using SOS optimization and δ- decidability over the reals. In *Proceedings of the 18th InternationalConference on Hybrid Systems: Computation and Control* (HSCC 2015), Seattle, USA, April 2015.

5. Md. Ariful Islam, Abhishek Murthy, Ezio Bartocci, Elizabeth M. Cherry, Flavio H. Fenton, James Glimm, Scott A. Smolka, and Radu Grosu. Model-Order Reduction of Ion Channel Dynamics using Approximate Bisimulation. *Theoretical Computer Science*, Volume 599, 27 September 2015, Pages 34-46.

6. Md. Ariful Islam, Abhishek Murthy, Tushar Deshpande, Scott D. Stoller, Scott A. Smolka, Ezio Bartocci, and Radu Grosu. Tracking Action Potentials of Nonlinear Excitable Cells Using Model Predictive Control. In *Proceedings of the 6th International Conference on Bioinformatics, Biocomputational Systems and Biotechnologies* (BIOTECHNO 2014), Chamonix, France, April 2014.

7. Md. Ariful Islam, Abhishek Murthy, Ezio Bartocci, Antoine Girard, Scott A. Smolka, and Radu Grosu. Compositionality Results for Cardiac Cell Dynamics. In *Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control* (HSCC 2014), Berlin, Germany, April 2014.

8. Abhishek Murthy, Md. Ariful Islam, Ezio Bartocci, Elizabeth M. Cherry, Flavio H. Fenton, James Glimm, Scott A. Smolka, and Radu Grosu. Approximate Bisimulations for Sodium Channel Dynamics. In *Proceedings of the 10th International Conference on Computational Methods in Systems Biology* (CMSB 2012), London, UK, October 2012.

Chapter 1

Introduction

Biological systems are extremely complex systems. The complexity exists at different levels of organization that ranges from the subatomic realm to individual organisms to whole populations and beyond. Despite the enormous complexity, biological systems operate in a harmonious fashion with milions of entities with a great precision. This harmony, however, is disrupted in disease states. Determining the root cause of diseases states is the key question in biological and medical research. Due to their inherent complexity, the comprehension and analysis of biological systems is a major challenge

Over the last few decades, the technological advances in data acquisition and data processing have revolutionized the way we understand biological processes. From the molecular level to the organ level, new mechanisms underlying these processes are uncovered on a regular basis, and then formalized mathematically. Such processes are, typically, described as *Differential Equations Model* (DEM) such as Partial Differential Equation (PDE)s and Ordinary Differential Equation (ODE)s.

Despite significant inroads into capturing biological processes using DEMs in recent years, we do not yet have a mature science to support verification and validation of Differential Equation Model (DEM). Traditional analysis tools are unable to cope with the full complexity of DEM or adequately predict behaviors. Additionally, approaches to modularity and composability that enable reliable and verifiable assembly of individual models to represent interacting systems of systems are needed.

The traditional approach in verification and validation of DEM relies mainly on simulation, which ensures the system works as intended only on a few scenarios selected carefully by the domain expert. Formal verification, on the other hand, promises to catch bugs in corner cases by exploring the space of all possible executions of the system. Recently, researchers have made steady progress in developing methodology, algorithms, and tools for the verification of nonlinear ODE models.

In particular, reachability analysis, an automatic technique for exploring the entire state space of a system model, have been studied in several papers [12, 2, 11, 50].

Reachability analysis, however, run into the well known *State Explosion Problem* when large DEMs are used for verification. In the case of myocytes, PDE models have evolved from 4 state variables in the Noble model [73] to 67 state variables in the Iyer-Mazhari-Winslow (IMW) model [47], pushing simulation efforts, not to mention formal analysis, up against the tractability boundary. The original benchmark for neurons was the 4-variable ODE model of Hodgkin and Huxley [42]. Thus, deriving guarantees for large-scale DEM is a grand challenge. Given a property of interest, the DEM i) must have the requisite level of detail, and ii) must be amenable to tractable formal verification.

Two techniques have proved to be particularly useful in extending the reach of formal analysis: *abstraction* and *decomposition*. Abstraction reduces the size of the system under investigation by neglecting details irrelevant to the properties of interest. Decomposition, as the name suggests, decomposes the system into smaller pieces, which are then analyzed on their own. Together, these two techniques permit *compositional reasoning*: if A_1 abstracts I_1 and A_2 abstracts I_2, then their composition $A_1 \| A_2$ abstracts $I_1 \| I_2$.

Compositional reasoning has proved to be especially useful in the non-numerical setting, with *simulation* and *bisimulation* among the most widely used abstraction techniques. Intuitively, simulation is a game requiring that each move of I from state x_1, with observation o_1, to a state x_2, with observation o_2, can be matched by a move of A from state y_1, with observation o_1, to a state y_2, with observation o_2. Both I and A start in their initial states, and iteratively continue from the successor states. Bisimulation is the symmetric form of this game. Simulation and bisimulation have the very salient property of supporting compositional reasoning in the sense discussed above.

In the numerical setting, the notions of simulation and bisimulation were extended in two ways. First, the moves of the game are assumed to take some (equal) amount of time. Second, identical observations are replaced with the more robust notions of *δ-simulation* and *δ-bisimulation*, where δ is the maximum distance (error) between observations.

In chapter 3, we first show that in the context of the Iyer et al. (IMW) 67-variable cardiac myocycte model, it is possible to replace the detailed 13-state probabilistic model of the sodium channel dynamics with a much simpler Hodgkin-Huxley (HH)-like two-state model, while only incurring a bounded approximation error. We then extend our technique to the 10-state model of the fast recovering calcium-independent potassium channel. The basis of our results is the construction of a δ-bisimulation between the HH-type abstraction and the corresponding detailed

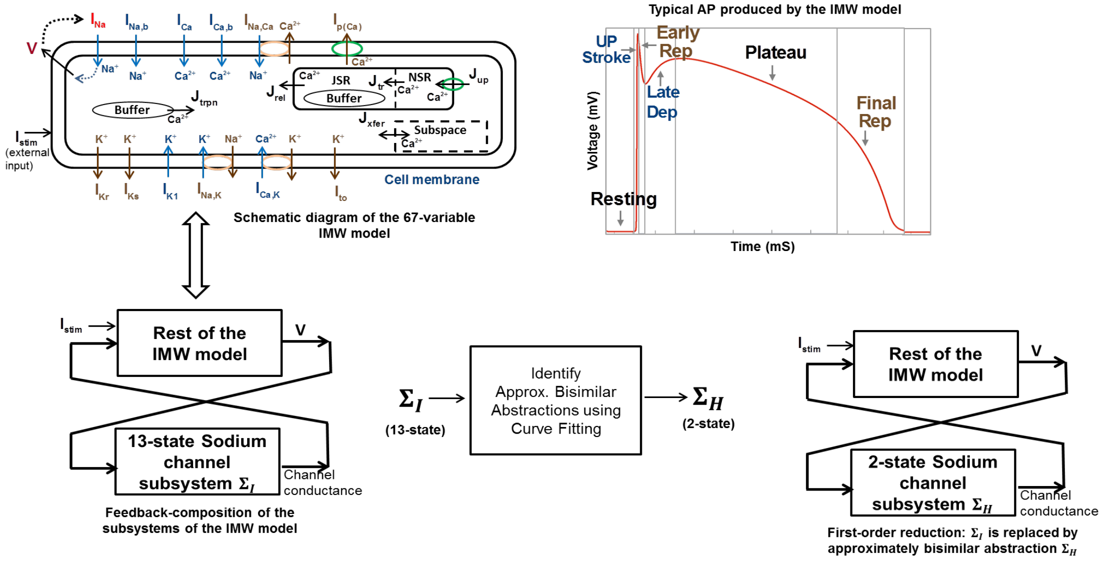


Figure 1.1: Component-wise model-order reduction of IMW model using abstraction and compositional reasoning.

ion channel model.

Unfortunately, δ-bisimulation does not automatically support compositional reasoning. When parallel composition feeds the outputs of subsystem A_1 to the inputs of subsystem A_2 (and vice versa), the error of A_1 with respect to I_1 can get amplified by A_2, which may lead to an even larger error of A_2 with respect to I_2, which is then again amplified by A_1, and so on. For example, in the context of the IMW model (see Fig. 5.1), let this model with its sodium channel removed be denoted by Σ_R (the "rest" of the IMW model). Then, replacing Σ_I within $\Sigma_I \| \Sigma_R$ with its δ-bisimilar channel Σ_H may lead to widely divergent behavior by $\Sigma_H \| \Sigma_R$ with respect to $\Sigma_I \| \Sigma_R$ in terms of their APs, even for a very small δ. Careful examination of the problem reveals that this is due to Σ_H not properly closing as the AP reaches the conclusion of its upstroke phase.

To find the proper conditions under which the δ-bisimulation abstraction Σ_H of Σ_I behaves properly with respect to its feedback composition with Σ_R, we enlist the support of *Bisimulation Functions* (BFs) [31, 33, 35, 29, 49]. Similar to Kamke and Lyapunov functions, *Bisimulation Functions* (BFs) have played a transformative role in extending the control-theoretic notions of Lyapunov Stability and ISS [82, 83, 84] to system verification. BFs are Lyapunov-like functions that decay along the trajectories of a given pair of dynamical systems. Level sets of BFs yield approximate bisimulation relations that generalize the classical notion of bisimulation equivalence of finite-state systems [68] to real-valued continuous-time dynamical systems. BFs also allow one to show that a system is robust to bounded deviations in the input signals. As shown in [29, 62], one can appeal to a small-gain theorem to compute BFs that *bound the error* that is introduced when substituting Σ_H for Σ_I within Σ_R.

In chapter 5, we present ***BFComp***: *an automated framework for computing BFs that characterize IOS of dynamical systems.* In addition to establishing IOS, ***BFComp*** is designed to provide tight bounds on the squared output errors between systems whenever possible. For this purpose, two SOS optimization formulations are employed: SOSP1, which enforces the decay requirements on a discretized grid over the input space, and SOSP2, which covers the input space exhaustively. SOSP2 is attempted first, and if the resulting error bounds are not satisfactory, SOSP1 is used to compute a Candidate BF (CBF). The decay requirement for the BFs is then encoded as an SMT formula and validated over a level set of the CBF using the dReal tool. We illustrate the utility of ***BFComp*** on a canonical cardiac-cell model, showing that the four-variable Markovian model for the slowly activating Potassium current component can be safely replaced by a one-variable Hodgkin-Huxley-type approximation.

In chapter 6, we present the formal verification of a nonlinear ODE model of a neural circuit in a multicellular organism: Tap Withdrawal (TW) in C. Elegans, the common roundworm. Specifically, we perform reachability analysis on the TW circuit model of Wicks et al. (1996), which enables us to estimate key circuit parameters. Underlying our approach is the use of Fan and Mitra's recently developed technique for automatically computing local discrepancy (convergence and divergence rates) of general nonlinear systems. We show that the results we obtain are in agreement with the experimental results of Wicks et al. (1995). As opposed to the fixed parameters found in most biological models, which can only produce the predominant behavior, our techniques characterize ranges of parameters that produce all three observed behaviors: reversal of movement, acceleration, and lack of response.

Finally, in chapter 8, we will summarize all the results and discuss some exciting future work directions.

Chapter 2

Background

In this chapter, we will first present the mathematical model of cardiac myocytes. We will show both markovian [47] and HH-type [42] ionic chanel dynamics. Later, we will present the mathematical model of tap withdrawal circuit in C. Elegans [92].

2.1 Dynamics of Cardiac Myocytes

Cardiac myocytes belong to the class of excitable cells, which also includes neurons. Such cells respond to an external electrical stimulus in the form of an AP, the characteristic change of the transmembrane potential in time as the cell responds to the stimulus. A typical ventricular myocyte AP and its associated phases are shown in Fig. 2.1(Right). Either an external stimulus, or the diffusing charge from the neighboring cells can excite the myocyte, causing an AP to quickly depolarize the membrane from a negative resting potential of V_{res} mV to a maximum of V_{max} mV followed by gradual repolarization.

2.1.1 The IMW Cardiac Cell Model

The IMW DEM, a physiologically detailed model, describes the ionic processes responsible for the generation of an AP in human ventricular myocytes:

$$\begin{aligned} -C\dot{V} &= I_{Na}+I_{Nab}+I_{Ca}+I_{Cab}+I_{Kr}+I_{Ks}+I_{K1}+I_{to1}+I_{p(Ca)}+ \\ &\quad I_{NaCa}+I_{NaK}+I_{CaK}+I_{st} \end{aligned} \tag{2.1}$$

where V is the membrane potential, $\dot{V}$ is its first-order time derivative, C is the membrane's capacitance, and I_v are the ionic currents shown in Fig. 2.1(Left), except

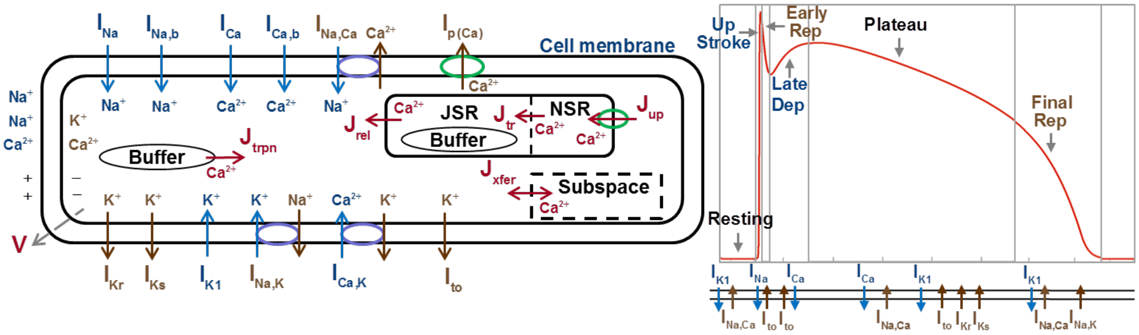


Figure 2.1: (Left) Currents in IMW: Blue and brown arrows show ionic currents flowing through channels. Blue circles and arrows correspond to ionic exchanger currents and green circles denote ionic pumps. Intra-cellular currents are shown in Magenta. (Right) The AP, its phases, and associated currents. (Right-Inlay) The sodium current I_{Na} dominates other currents during the upstroke phase. The calcium-independent transient outward Potassium current I_{to1} influences the AP's notch in the early repolarization phase. We focus on one of its two components, the $I_{Kv4.3}$ current.

for I_{st}. This is the stimulus current, which could be either an external stimulus or the diffused charge from neighboring cells.

Ionic currents result from the flow of sodium, potassium and calcium ions, across the myocyte's membrane. Three types of transport mechanisms are responsible for these flows: channels, pumps, and exchangers. Ion channels are special proteins that penetrate the membrane's lipid bi-layer, and are selectively permeable to certain ion species. Depending on the conformation of the constituent protein, the channel either allows or inhibits the movement of ions. The protein conformation is voltage dependent, thus the name voltage-gated channels. All the transmembrane currents in Fig. 2.1 result from voltage-gated ionic channels, except for I_{NaK}, I_{NaCa}, and $I_{p(Ca)}$, which are exchanger or pump currents.

We focus on the behavior of the sodium channel, which regulates the flow of I_{Na}, a dominant current in the upstroke phase of the AP. The other focus of the paper will be the channel corresponding to the potassium current, $I_{Kv4.3}$. Along with the $I_{Kv1.4}$ current, it constitutes the I_{to1} current, which influences the notch of the AP at the end of early repolarization [39].

2.1.2 The HH Model for Ion Channels

Hodgkin and Huxley, in their seminal work of [42], model the squid neuron's sodium channel behavior using two independent processes: activation and inactivation.

Starting from the resting potential, if the cell is depolarized to a constant voltage, *activation* is responsible for the sudden increase in the channel's conductance. This is followed by *inactivation*, which gradually brings the conductance down, before reaching a steady state. The resulting ionic current is modeled as

$$I_{Na}(V,t) = \overline{g}_{Na}\, m^3(V,t)\, h(V,t)(V - V_{Na}) \tag{2.2}$$

where $\overline{g}_{Na}$ is the maximum conductance of the sodium channel, and V_{Na} is sodium's Nernst potential. The gating variables $m(V)$ and $(1 - h(V))$ measure the extent of activation and inactivation respectively, with $m = 0$ $(h = 1)$ representing complete deactivation (recovery from inactivation) and $m = 1$ $(h = 0)$ representing complete activation (inactivation).

The dynamics of m and h are given by the DEM $\mathcal{H}$ as follows.

Definition 2.1.1. *The DEM $\mathcal{H}$ gives the* ***dynamics of*** *m* ***and*** *h, both $\in \mathbb{R}_{\geq 0}$:*

$$\dot{m} = \alpha_m(V)(1-m) - \beta_m(V)m, \qquad \dot{h} = \alpha_h(V)(1-h) - \beta_h(V)h$$

The linear system obtained by fixing $V = v$ will be denoted as $\mathcal{H}^v$. The conductance of a channel modeled using by m and h is given by:

$$O_H(V,t) = m(V,t)^{\lambda} h(V,t) \tag{2.3}$$

where λ is the degree of activation[1]*, and is 3 in [42]. The rate functions $\alpha_x(V)$ and $\beta_x(V), x \in \{m,h\}$ are continuous in V, and for the sodium channel are given by Eqs. (20)-(21) and Eqs. (23)-(24) of [42].*

In the Voltage Clamp Experiment (VCE) performed by the authors in [42], the membrane potential was initially maintained at the resting potential V_{res}. In each of the VCE, the membrane potential was suddenly changed to a clamp potential V_c and the emanating ionic current and conductance was observed till steady state. The dynamics of the gating variables m and h, during a VCE is given by:

$$x(V = V_c, t) = x_\infty - (x_\infty - x_0)exp(-t/\tau_x), \quad x \in \{m,h\} \tag{2.4}$$

where the parameters $x_\infty(V_c) = \alpha_x(V_c)/(\alpha_x(V_c) + \beta_x(V_c))$, $\tau_x = 1/(\alpha_x(V_c) + \beta_x(V_c))$ and $x_0(V_c) = x_\infty(V_{res})$, $x \in \{m,h\}$. Eq. (2.3) was then fit to the observed current to estimate the parameters.

[1] In this paper, we consider the degree of inactivation to be 1 for reducing the sodium and potassium channels.

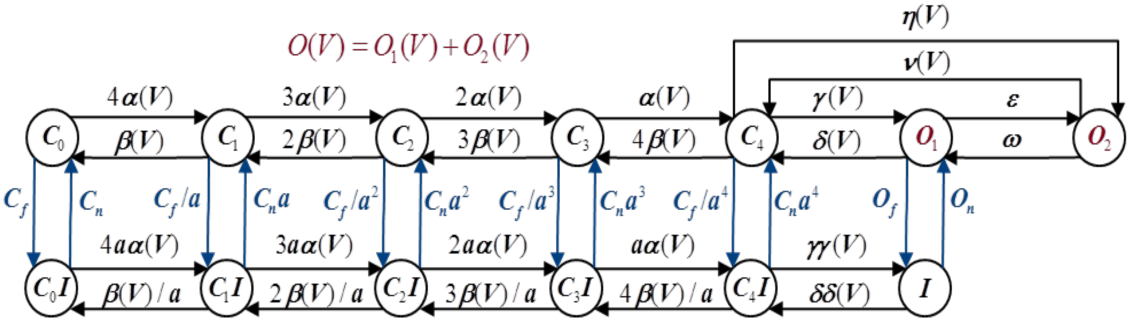


Figure 2.2: The 13-variable voltage-controlled CTMC, $\mathcal{M}_{Na}$, for the sodium channel component of the IMW model. The conductance of the channel is the sum of occupancy probabilities $O = O_1 + O_2$. The transition rates are defined in Table 2.1.

2.1.3 Voltage-Controlled CTMC Models

The IMW model uses physiologically detailed models to capture the behavior of ion channels. The voltage-dependent conformations of the constituent protein are captured as states of a stochastic model. Continuous functions of voltage are employed for transfer rates between the states.

In the IMW model, the sodium current, I_{Na}, is modeled by an equation similar to Eq. (2.2). The conductance of the HH-type channel, m^3h, is replaced by $O_{Na}(V,t) = O_1(V,t) + O_2(V,t)$, the sum of the occupancy probabilities of the two states of the stochastic model shown in Fig. 2.2 [56, 58] and defined as follows.

Definition 2.1.2. *Consider the 13-state model for sodium-channel dynamics shown in Fig. 2.2. Let p_j denote the j^{th} state occupancy probability from the vector $\mathbf{p} = [C_0, C_1, C_2, C_3, C_4, O_1, O_2, C_0I, C_1I, C_2I, C_3I, C_4I, I]^T \in \mathbb{R}^{13}_{\geq 0}$. The **evolution of** $\mathbf{p}$ is described by the model $\mathcal{M}_{Na}$ as:*

$$\dot{p}_j = \sum_{i \neq j} k_{ij}(V) p_i - \sum_{i \neq j} k_{ji}(V) p_j \qquad i, j = 1, \ldots, 13 \tag{2.5}$$

where $\dot{p}_j$ denotes the time derivative of p_j, V is the transmembrane potential, and $k_{ij}(V) \in \mathbb{R}_{\geq 0}$ is the transition rate from the i^{th} to the j^{th} state as defined in Table 2.1. This system can be re-written as:

$$\dot{\mathbf{p}} = A^{Na}(V).\mathbf{p}, \tag{2.6}$$

where $A^{Na}(V)$ is a 13×13 matrix with $A^{Na}_{j,i}(V) = k_{ij}(V)$ $i \neq j$, $A^{Na}_{j,j}(V) = -\sum\limits_{i \neq j} k_{ji}$. The linear system $\mathcal{M}^v_{Na}$ is obtained by fixing $V = v$ in Eq. 2.6. The conductance (output) of the model is given by $O_1 + O_2$.

rate	function	rate	function	rate	function
$\alpha(V)$	$c.e^{-19.6759+0.0113V}$	$\delta\delta(V)$	$c.e^{-38.4839-0.1440V}$	ϵ	0.0227
$\beta(V)$	$c.e^{-26.2321-0.0901V}$	$\gamma\gamma(V)$	$c.e^{-21.9493+0.0301V}$	ω	1.0890
$\gamma(V)$	$c.e^{-16.5359+0.1097V}$	$\eta(V)$	$c.e^{-19.6729+0.0843V}$	c_n	0.7470
$\delta(V)$	$c.e^{-27.0926-0.0615V}$	$O_n(V)$	$c.e^{-20.6726+0.0114V}$	c_f	0.2261
$\nu(V)$	$c.e^{-26.3585-0.0678V}$	$O_f(V)$	$c.e^{-39.7449+0.0027V}$	a	1.4004

Table 2.1: Transfer rates of $\mathcal{M}_{Na}$, which is shown in Fig. 2.2. Values were instantiated from Table 6 of [47] at temperature T = 310K, and $c = 8.513 \times 10^9$.

The potassium current $I_{Kv4.3}$ in the IMW DEM is modeled by

$$I_{Kv4.3} = \overline{g}_{Kv4.3}\left(O_K(V)\right)(V - V_K) \tag{2.7}$$

where $\overline{g}_K$ is the maximum conductance of the channel, V_K is the Nernst potential for potassium, $O_K(V) = O(V)$ is the occupancy probability of the open state in the 10-state voltage-controlled CTMC shown in Fig. 2.3 [39] and defined as follows.

Definition 2.1.3. *Consider the 10-state model for the potassium channel component shown in Fig. 2.3. Let q_j denote the j^{th} state occupancy probability from the vector $\mathbf{q} = [C_0, C_1, C_2, C_3, O, CI_0, CI_1, CI_2, CI_3, OI]^T \in \mathbb{R}^{10}_{\geq 0}$. The* ***evolution of*** $\mathbf{q}$*, as described by the model $\mathcal{M}_K$, is given by*

$$\dot{q}_j = \sum_{i \neq j} k_{ij}(V) q_i - \sum_{i \neq j} k_{ji}(V) q_j \qquad i, j = 1, \ldots, 10 \tag{2.8}$$

where $\dot{q}_j$ denotes the time derivative of q_j, V is the transmembrane potential, and $k_{ij}(V) \in \mathbb{R}_{\geq 0}$ is the transition rate from the i^{th} to the j^{th} state as defined in Table 2.2. This system can be re-written as:

$$\dot{\mathbf{q}} = A^K(V).\mathbf{q}, \tag{2.9}$$

where $A^K(V)$ is a 10×10 matrix with $A^K_{j,i}(V) = k_{ij}(V)$ $i \neq j$, $A^K_{j,j}(V) = -\sum\limits_{i \neq j} k_{ji}$. The linear system $\mathcal{M}^v_K$ is obtained by fixing $V = v$ in Eq. 2.9. The conductance (output) of the model is given by O.

2.2 Tap Withdrawal Circuit of C. Elegans

In this section, we describe the Tap Withdrawal (TW) neural circuit of *C. Elegans*. We then present a mathematical model of the circuit.

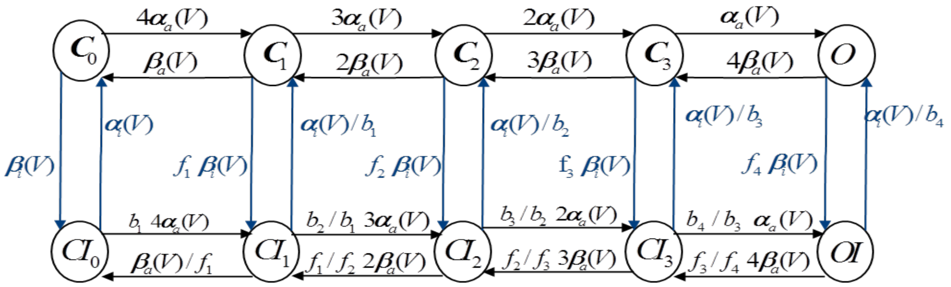


Figure 2.3: The 10-variable voltage-controlled CTMC, $\mathcal{M}_K$, for the potassium channel component of the IMW model. The conductance of the channel is the occupancy probability O. The transition rates are defined in Table 2.2.

rate	function	rate	function	rate	function
$\alpha_a(V)$	$0.5437e^{0.029V}$	f_1	1.8936	b_1	6.7735
$\beta_a(V)$	$0.0802e^{0.0468V}$	f_2	14.2246	b_2	15.6213
$\alpha_i(V)$	$0.0498e^{0.0004V}$	f_3	158.5744	b_3	28.7533
$\beta_i(V)$	$0.0008e^{(5.374\times10^{-8}V)}$	f_4	142.9366	b_4	524.5762

Table 2.2: Transfer rates of $\mathcal{M}_K$, shown in Fig. 2.3. $c = 8.513 \times 10^9$. Note that these values are different from the ones given in Table 9 of [47]. Corrections were made to the parameters to match the observed APs of the IMW model.

2.2.1 The Tap Withdrawal Circuit

In *C. Elegans*, there are three classes of neurons: *sensory*, *inter*, and *motor*. For the TW circuit, the sensory neurons are *PLM*, *PVD*, *ALM,* and *AVM*, and the interneurons are *AVD*, *DVA*, *PVC*, *AVA*, and *AVB*. The model we are using abstracts away the motor neurons as simply forward and reverse movement.

Neurons are connected in two ways: electrically via bi-directional *gap junctions*, and chemically via uni-directional chemical *synapses*. Each connection has varying degrees of throughput, and each neuron can be *excitatory* or *inhibitory*, governing the polarity of transmitted signals. These polarities were experimentally determined in [92], and used to produce the circuit shown in Fig. 2.4.

The TW circuit produces three distinct locomotive behaviors: *acceleration*, *reversal* of movement, and a *lack of response*. In [91], Wicks et al. performed a series of laser ablation experiments in which they knocked out a neuron in a group of animals (worms), subjected them to a tapped surface, and recorded the magnitude and direction of the resulting behavior. In the control group with no neurons knocked out, 98% of subjects reacted to a tap with a reversal of locomotion, but there were

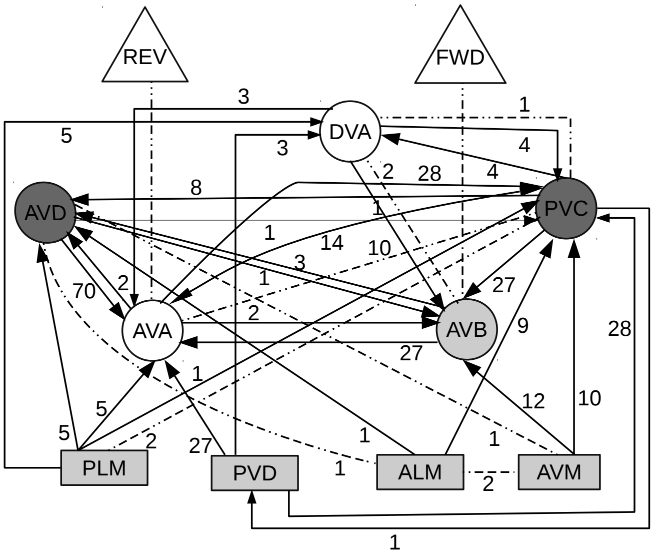


Figure 2.4: Tap Withdrawal Circuit of C. Elegans. Rectangle: Sensory Neurons; Circle: Inter-neurons; Dashed Undirected Edge: Gap Junction; Solid Directed Edge: Chemical Synapse; Edge Label: Number of Connections; Dark Gray: Excitatory Neuron; Light Gray: Inhibitory Neuron; White: Unknown Polarity. FWD: Forward Motor system; REV: Reverse Motor System.

still measured cases of acceleration and "no response" behavior. Fig. 2.5 shows the response types for each of their experiments.

2.2.2 Mathematical Model for the Tap Withdrawal Circuit

In [92], Wicks et al. presented a mathematical model for the TW circuit. This subsection provides an overview of this model.

Modeling the Neuron

The dynamics of a neuron's membrane potential, V, is determined by the sum of all input currents, written as:

$$C_m\dot{V} = \frac{1}{R_m}(V_l - V) + \sum I^{gap} + \sum I^{syn} + I^{stim}$$

where C_m is the *membrane capacitance*, R_m is the *membrane resistance*, V_l is the *leakage potential*, I^{gap} and I^{syn} are gap-junction and the chemical synapse currents, respectively, and I^{stim} is the applied external *stimulus* current. The summations are over all neurons with which this neuron has a (gap-junction or synaptic) connection.

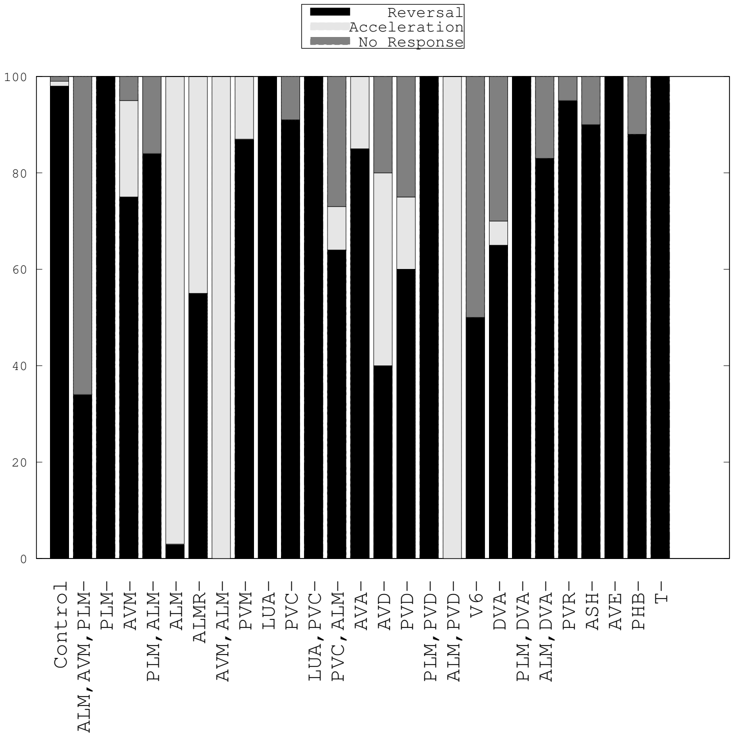


Figure 2.5: Effect of ablation on Tap Withdrawal reflex. The length of the bars indicate the fraction of the population demonstrating the particular behavior. [91]

Modeling Gap-Junction Currents

The current flow between neuron i and j via a gap-junction is given by:

$$I_{ij}^{gap} = n_{ij}^{gap} g_m^{gap} (V_j - V_i)$$

where the constant g_m^{gap} is the *maximum conductance* of the gap junction, and n_{ij}^{gap} is the number of gap-junction connections between neurons i and j. The conductance g_m^{gap} is one of the key circuit parameters of this model that dramatically affects the behavior of the animal.

Modeling Chemical Synapse Currents

The synaptic current flowing from pre-synaptic neuron j to post-synaptic neuron i is described as follows:

$$I_{ij}^{syn} = n_{ij}^{syn} g_{ij}^{syn}(t)(E_j - V_i)$$

where $g_{ij}^{syn}(t)$ is the time-varying synaptic conductance of neuron i, n_{ij}^{syn} is the number of synaptic connections from neuron j to neuron i, and E_j is the *reversal* *potential* of neuron j for the synaptic conductance.

The chemical synapse is characterized by a synaptic sign, or polarity, specifying if said synapse is excitatory or inhibitory. The value of E_j is assumed to be constant for the same synaptic sign; its value is higher if the synapse is excitatory rather than inhibitory.

Synaptic conductance is dependent only upon the membrane potential of presynaptic neuron V_j, given by:

$$g_{ij}^{syn}(t) = g_{\infty}^{syn}(V_j)$$

where g_{∞}^{syn} is the steady-state *post-synaptic conductance* in response to a pre-synaptic membrane potential.

The steady-state post-synaptic membrane conductance is modeled as:

$$g_{\infty}^{syn}(V_j) = \frac{g_m^{syn}}{1 + \exp\left(-4.3944 \frac{V_j - V_{EQ_j}}{V_{Range}}\right)}$$

where g_m^{syn} is the *maximum post-synaptic membrane conductance* for the synapse, V_{EQ_j} is the *pre-synaptic equilibrium potential*, and V_{Range} is the *pre-synaptic voltage range* over which the synapse is activated.

Modeling the Tap Withdrawal Circuit

Combining all of the above pieces, the mathematical model of the TW circuit is a system of nonlinear ODEs, with each state variable defined as the membrane potential of a neuron in the circuit. Consider a circuit with N neurons. The dynamics of the i^{th} neuron of the circuit is given by:

$$C_{m_i}\dot{V_i} = \frac{V_{l_i} - V_i}{R_{m_i}} + \sum_{j=1}^{N} I_{ij}^{gap} + \sum_{j=1}^{N} I_{ij}^{syn} + I_i^{stim} \tag{2.10}$$

$$I_{ij}^{gap} = n_{ij}^{gap} g_m^{gap}(V_j - V_i) \tag{2.11}$$

$$I_{ij}^{syn} = n_{ij}^{syn} g_{ij}^{syn}(E_j - V_i) \tag{2.12}$$

$$g_{ij}^{syn} = \frac{g_m^{syn}}{1 + \exp\left(-4.3944 \frac{V_j - V_{EQ_j}}{V_{Range}}\right)}. \tag{2.13}$$

The equilibrium potentials (V_{EQ}) of the neurons are computed by setting the left-hand side of Eq. (2.10) to zero. This leads to a system of linear equations, that can be solved as follows:

$$V_{EQ} = A^{-1}b \tag{2.14}$$

where matrix A is given by:

$$A_{ij} = \begin{cases} -R_{m_i} n_{ij}^{gap} g_m^{gap} & \text{if } i \neq j \\ 1 + R_{m_i} \sum_{j=1}^{N} n_{ij}^{gap} g_{ij}^{gap} g_m^{syn}/2 & \text{if } i = j \end{cases}$$

and vector b is written as:

$$b_i = V_{l_i} + R_{m_i} \sum_{j=1}^{N} E_j n_{ij}^{syn} g_m^{syn}/2.$$

Output of the Tap Withdrawal Circuit

The potential of the motor neurons AVB and AVA determine the observable behavior of the animal. If the integral of the difference between V_{AVA} - V_{AVB} is large, the animal will reverse movement. By extension, if the difference is a large negative value, the animal will accelerate, and if the difference is close to zero there will be no response. The equation that converts the membrane potential of AVB and AVA to a behavioral property, (e.g. *reversal*), is given by:

$$\text{Propensity to Reverse} \propto \int (V_{AVA} - V_{AVB}) dt \tag{2.15}$$

where the integration is computed from the beginning of tap stimulation until either the simulation ends or the integrand changes sign. To allow initial transients after the tap, the test for a change of integrand sign occurs only after a grace period of 100 ms.

2.2.3 Normalization of Tap Withdrawal Circuit Model

For the purpose of reachability analysis (Section 6.1), we normalize the system of equations with respect to the capacitance. Combining Eqs.(2.10) and (2.13) and taking C_{m_i} to the right-hand side, we have:

$$\dot{V}_i = \frac{V_{l_i} - V_i}{R_{m_i} C_{m_i}} + \frac{g_m^{gap}}{C_{m_i}} \sum_{j=1}^{N} n_{ij}^{gap} (V_j - V_i) + \frac{g_m^{syn}}{C_{m_i}} \sum_{j=1}^{N} \frac{n_{ij}^{syn} (E_j - V_i)}{1 + \exp(-4.3944 \frac{V_j - V_{EQ_j}}{V_{Range}})} + \frac{1}{C_{m_i}} I_i^{stim}$$

Now letting $g_i^{leak} = \frac{1}{R_{m_i} C_{m_i}}$, $g_i^{gap} = \frac{g_m^{gap}}{C_{m_i}}$, $g_i^{syn} = \frac{g_m^{syn}}{C_{m_i}}$ and $I_i^{ext} = \frac{1}{C_{m_i}}$ the system dynamics can be written as:

$$\dot{V}_i = g_i^{leak} (V_{l_i} - V_i) + g_i^{gap} \sum_{j=1}^{N} n_{ij}^{gap} (V_j - V_i) + g_i^{syn} \sum_{j=1}^{N} \frac{n_{ij}^{syn} (E_j - V_i)}{1 + \exp(-4.3944 \frac{V_j - V_{EQ_j}}{V_{Range}})} + I_i^{ext} \tag{2.16}$$

This is the 9 dimensional ODE model of the TW circuit. The key circuit parameters are the gap conductances, g_i^{gap}, and we aim to characterize the ranges of these conductances that produce acceleration, reversal, and no response.

Chapter 3

Approximate Bisimulation of Ion Channel Dynamics

The emergence of high throughput data acquisition techniques has changed cell biology from a purely wet lab-based science to also an engineering and information science. The identification of a mathematical model from cellular experimental data, and the use of this model to predict and control the cell's behavior, are nowadays indispensable tools in cell biology's arsenal [71, 8].

Improved data acquisition has led to the creation of increasingly sophisticated partial DEMs for cardiac cells (myocytes). Their main purpose is to elucidate the biological laws governing the electric behavior of cardiac myocytes, i.e., their underlying ionic processes [20].

Inspired by the squid-neuron DEM [42] developed by HH, Luo and Rudy devised one of the first myocyte DEMs, for guinea pig ventricular cells [60]. Adapting this model to human myocytes led to the ten Tusscher-Noble^2-Panfilov DEM [85], which has 17 state variables and 44 parameters. Based on updated experimental data, IMW subsequently developed a DEM comprising of 67 state variables and 94 parameters [47]. This DEM reflects a highly detailed physiological view the electrochemical behavior of human myocytes.

From 17 to 67 variables, all such DEMs capture myocytic behavior at a particular level of abstraction, and hence all of them play an important role in the modeling hierarchy. It is essential, however, to maintain focus on the purpose of a particular DEM; that is, of the particular cellular and ionic processes whose behavior the DEM is intended to capture. Disregarding this purpose may lead to the use of unnecessarily complex DEMs, which may render not only analysis, but also simulation, intractable.

If the only entity of interest is the myocyte's transmembrane voltage, coauthors Cherry and Fenton have experimentally shown that a Minimal Model (MM) consisting of only 4 variables and 27 parameters can accurately capture voltage propagation properties in 1D, 2D, and 3D networks of myocytes [7]. The MM has allowed us to obtain dramatic simulation speedups [5], and to use its linear hybridization as the basis for formal symbolic analysis [40].

Since new technological advances are expected to lead to further insights into myocytic behavior, it is likely that the IMW model will be further refined by adding new variables. As in model checking and controller synthesis, one would therefore like to compute the smallest approximation that is observationally equivalent to the state-of-the-art DEM with respect to the property of interest, modulo some bounded approximation error. This, however, is not easily accomplished, as it implies the automatic approximation of very large nonlinear DEMs.

A first step toward the desired automation is to identify a set of approximation techniques that allow one to systematically remove unobservable variables from say, a detailed model such as IMW to end up with the MM, if the only observable variable is the voltage. This is one of the goals of the project computational modeling and analysis of complex systems [72]. A byproduct of this work is to establish a long-missing formal relation among the existing myocyte DEMs, facilitating the transfer of properties established at one layer of abstraction to the other layers. Building such *towers of abstraction* is becoming increasingly prevalent in systems biology [22, 24].

In this chapter, we focus on model-order reduction and abstraction of ion channel dynamics. The main question posed in this paper is the following: *Assuming that the conductance of the ion channel is the only observable, is the behavior of an HH-type channel equivalent to the behavior of the IMW channel, modulo a well-defined approximation error?* Specifically, we answer this question for the sodium and the calcium-independent potassium channels. Rather than dealing with behavioral equivalence explicitly, we ask if it is possible to construct an approximate bisimulation [30, 34, 32, 35] between the HH and IMW channel models? This notion of equivalence is stronger than the conventional behavioral equivalence, which compares the observed behaviors (trajectories) of two systems.

The answer to the above-posed question is of broad interest, as it reduces to showing the existence of an approximate bisimulation between two continuous time Markov decision processes; that is, two input-controlled (voltage in this case) CTMCs. We answer this question in the positive, by explicitly constructing such a bisimulation. See Fig. 3.1 for an overview of our approach.

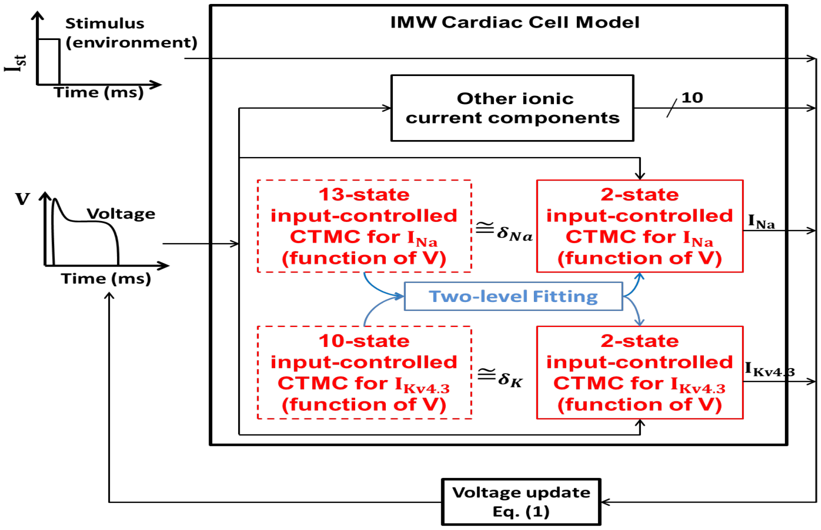


Figure 3.1: A modular view of the IMW DEM that composes various concurrently evolving components corresponding to the different ionic currents. We replace the 13-state I_{Na} and the 10-state $I_{Kv4.3}$ components with corresponding 2-state HH-type abstractions. A two-level fitting process, described in Section 3.1, is used to identify the abstractions. Approximate bisimilarity of the detailed components and their corresponding abstractions allows us to substitute them for each other within the whole-cell model. The stimulus current affects the overall voltage update and is not an input to the ionic current components. The system outputs the 13 currents in Eq. (1).

3.1 Model-Order Reduction of Ion Channel Dynamics

We construct two HH-type DEMs, $\mathcal{H}_{Na}$ and $\mathcal{H}_K$, that can be substituted for the components $\mathcal{M}_{Na}$ and $\mathcal{M}_K$ respectively within the IMW cardiac-cell model. We perform the following abstractions in the process:

- The abstractions $\mathcal{H}_{Na}$ and $\mathcal{H}_K$ employ three and four activating subunits respectively. A single subunit is used to model inactivation.
- We abstract away the conditional dependence between activation and inactivation. This is done by abstracting away the scaling factors: a of $\mathcal{M}_{Na}$ and $f_1 - f_4$, $b_1 - b_4$ of $\mathcal{M}_K$.

After identifying $\mathcal{H}_{Na}$, its conductance, m^3h, is substituted for $\mathcal{M}_{Na}$'s conductance, O_1+O_2, in the IMW model's version of Eq. (2.2). Similarly $\mathcal{H}_K$'s conductance, m^4h, replaces $\mathcal{M}_K$'s conductance, O_K in Eq. (2.7). Note that this leads to two levels of substitution. First, the conductance of the detailed ionic current components is substituted by the abstract model conductances. Then the modified current component replaces the original term in Eq. (2.1).

Our approach to obtaining the 2-state HH-type abstractions $\mathcal{H}_{Na}$ and $\mathcal{H}_K$ from the detailed models, $\mathcal{M}_{Na}$ and $\mathcal{M}_K$ respectively, is summarized in Fig. 3.2, and described as follows.

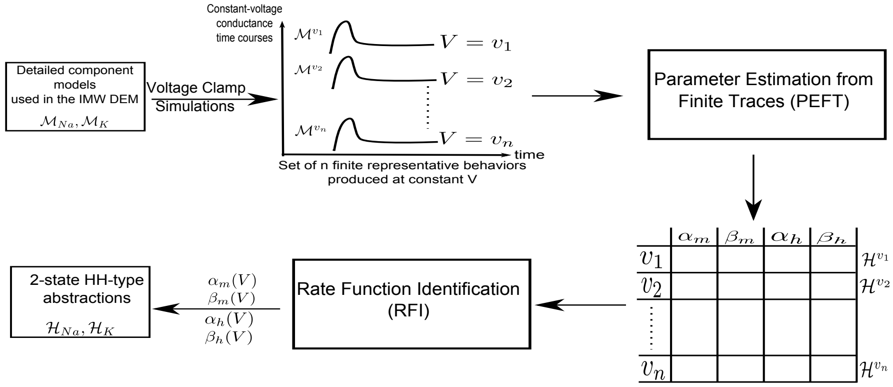

Detailed component models used in the IMW DEM
$\mathcal{M}_{Na}, \mathcal{M}_K$
Voltage Clamp Simulations
Constant-voltage conductance time courses
$\mathcal{M}^{v_1}$
$V = v_1$
$\mathcal{M}^{v_2}$
$V = v_2$
$\mathcal{M}^{v_n}$
$V = v_n$
time
Set of n finite representative behaviors produced at constant V
Parameter Estimation from Finite Traces (PEFT)
α_m
β_m
α_h
β_h
v_1
v_2
v_n
$\mathcal{H}^{v_1}$
$\mathcal{H}^{v_2}$
$\mathcal{H}^{v_n}$
Rate Function Identification (RFI)
$\alpha_m(V)$
$\beta_m(V)$
$\alpha_h(V)$
$\beta_h(V)$
2-state HH-type abstractions
$\mathcal{H}_{Na}, \mathcal{H}_K$

Figure 3.2: Abstraction process for ion channel dynamics. The voltage-controlled CTMC components are simulated at constant voltages (clamp potentials) using the steady state values corresponding to $V = V_{res}$ as the initial conditions. The conductance time courses are then fit as per Eq. (2.3) to obtain the parameters $\alpha_m, \beta_m, \alpha_h, \beta_h$ at the clamp potentials used for voltage clamp simulations. The four parameter values, along with the initial conditions determine the $\mathcal{H}^v$ abstractions. The parameters are then fit in the RFI step to obtain parameter functions $\alpha_m(V), \beta_m(V), \alpha_h(V), \beta_h(V)$.

1. **Voltage clamp simulations**
   VCEs, pioneered by Hodgkin and Huxley in their seminal work of [42], are intended to expose the activation and inactivation processes governing a channel's behavior. The experiments involve stimulating the channel by changing the membrane potential suddenly and then holding it constant, starting from appropriate initial conditions. As the ion channel reacts by opening, and then closing, the resulting ionic currents are recorded. The corresponding conductance time courses characterize the channel's response to varying the membrane potential.

   We simulated VCEs by simulating the detailed models, $\mathcal{M}_{Na}$ and $\mathcal{M}_K$, for various values of V. In other words, the systems of linear differential equations, $\mathcal{M}^v_{Na}$ and $\mathcal{M}^v_K$, were simulated for different values of $V_{res} \leq v \leq V_{max}$. We used 20,000 uniformly spaced voltage values for v, and the corresponding $\mathcal{M}^v_{Na}$ and $\mathcal{M}^v_K$ were simulated using MATLAB's ODE45 solver [64], starting from the initial conditions specified in Table 4 of [47]. The initial conditions correspond to the steady state of the models at $v = V_{res}$. This is exactly the initially conditions used by Hodgkin and Huxley in [42]. The resulting conductance time courses, $O_1(t) + O_2(t)$ for $\mathcal{M}_{Na}^v$ and $O(t)$ for $\mathcal{M}_K^v$, were recorded until steady state was reached. As per Theorem 3 in Appendix A, for all $v \in [V_{res}, V_{max}]$, $\mathcal{M}_{Na}^v$ and $\mathcal{M}_K^v$ have stable equilibria and therefore steady state is guaranteed. Simulating the models at constant voltage values corresponds to the clamp potentials to which the membrane was excited in [42] to uncover the activation and inactivation process.

2. **PEFT**

   PEFT is a procedure that identifies $\mathcal{H}_{Na}^v$ and $\mathcal{H}_K^v$ models corresponding to $\mathcal{M}_{Na}^v$ and $\mathcal{M}_K^v$ respectively. The parameters $\alpha_m^v, \beta_m^v, \alpha_h^v, \beta_h^v$ are estimated such that the resulting conductance time courses, produced by $\mathcal{H}_{Na}^v$ and $\mathcal{H}_K^v$, match the conductance time courses observed in the voltage clamp simulations for $\mathcal{M}_{Na}^v$ and $\mathcal{M}_K^v$ respectively.

   In our implementation, we fit $\mathcal{M}_{Na}^v$'s conductance time series, $O_1(t)+O_2(t)$, to $\mathcal{H}_{Na}^v$'s $m^3(t)h(t)$. Time series $O(t)$ observed from $\mathcal{M}_K^v$ was fit with $m^4(t)h(t)$ to identify $\mathcal{H}_{Na}^v$. At constant voltage v, the trajectories $m(t)$ and $h(t)$ of $\mathcal{H}_{Na}^v$ and $\mathcal{H}_K^v$ are given by

   $$z(t) = \frac{\alpha_z^v}{\alpha_z^v + \beta_z^v} + \left(z(0) - \frac{\alpha_z^v}{\alpha_z^v + \beta_z^v} \right) \exp\left(- \left(\alpha_z^v + \beta_z^v \right) t \right) \tag{3.1}$$

   where $z \in \{m, h\}$ and $x \in \{Na, K\}$. The fitting was performed using MATLAB's curve fitting utility *cftool* [65] for each voltage value v used in the voltage clamp simulations. Two aspects of our implementation deserve further elaboration:

   - **Choosing $m(0)$ and $h(0)$** - The initial conditions were chosen such that for $\mathcal{H}_{Na}^v$, $m(0)^3h(0)$ and for $\mathcal{H}_K^v$, $m(0)^4h(0)$ was approximately equal to $O_{V_{res}}$, the steady state conductances of $\mathcal{M}_{Na}^{V_{res}}$ and $\mathcal{M}_K^{V_{res}}$ respectively. As per convention, we also ensured that $m^v(0) \approx 0$ and $h(0) \approx 1$ for both the models. We chose $m(0) = 0.0027, h(0) = 0.95$ for $\mathcal{H}_{Na}^v$; $m(0) = 0.00138, h(0) = 0.98$ for $\mathcal{H}_{Na}^v$.
   - **Providing seed-values** - For each voltage-value v, *cftool* needs seed values of α_x^v, β_x^v, $x \in \{m, h\}$, to start optimizing over the parameter space. The parameters estimated for the i^{th} voltage v_i were used as seed-values for v_{i+1}. For $i = 0$, when $v = v_{res}$, the parameters were calculated by trial and error.

3. **RFI**

   RFI is a procedure that fits the parameter values, α_m^v, β_m^v, α_h^v and β_h^v, as functions of voltage to produce the parameter functions $\alpha_m(V)$, $\beta_m(V)$, $\alpha_h(V)$ and $\beta_h(V)$. We fit exponential functions using *cftool* [65]. The parameter functions defining $\mathcal{H}_{Na}$ and $\mathcal{H}_K$ are as follows.

Rate functions that define $\mathcal{H}_{Na}$:

$$\alpha_m(V) = \begin{cases} 13.63 - \frac{14.3}{1+\exp(0.061V+1.72)} & V \leq 19.98 \\ 20.76 - \frac{7.89}{1+\exp(464.1V-13920)} & V > 19.98 \end{cases}$$

$$\beta_m(V) = \begin{cases} 9.925 & V \leq -65 \\ 4.7 - \frac{2.58}{1+\exp(0.61V+38.19)} & -65 < V \leq 5.8 \\ 12.24 - \frac{7.77}{1+\exp(1877V-56314)} & V > 5.8 \end{cases}$$

$$\alpha_h(V) = \frac{0.1745}{1+\exp(269.8V+17720)}$$

$$\beta_h(V) = 10.1 - \frac{10}{1+\exp(0.0579V+0.71)}$$

Rate functions that define $\mathcal{H}_K$

$$\alpha_m(V) = \begin{cases} 0.45\exp(0.026V) & V \leq 24.5 \\ 0.85 - \frac{0.048}{1+\exp(-0.2V+9.5)} & V > 24.5 \end{cases}$$

$$\beta_m(V) = \begin{cases} 0.029\exp(0.065V) & V \leq 24.5 \\ 0.1839 - \frac{0.05}{1+\exp(0.19V-9.32)} & V > 24.5 \end{cases}$$

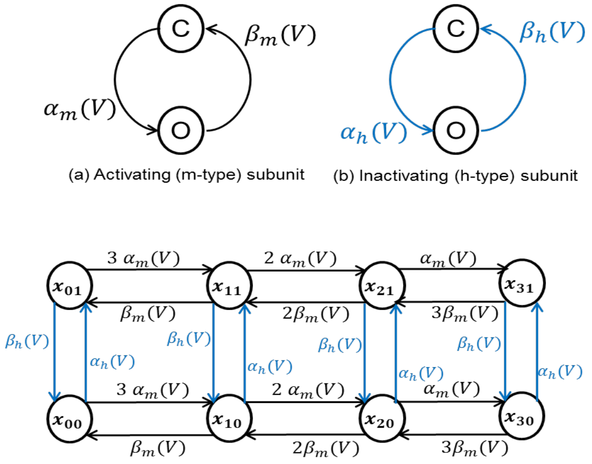
$$\alpha_h(V) = 0.0015 - \frac{0.0014}{1+\exp(0.027V-2.54)}$$

$$\beta_h(V) = \begin{cases} 0.12 - \frac{0.06}{1+\exp(-0.054V+2.62)} & V \leq 24.5 \\ 0.109 + \frac{0.015}{1+\exp(-0.033V+10.83)} & V > 24.5 \end{cases}$$

Adapting the Abstraction Process to Arbitrary Observable Functions of $\mathcal{M}_{Na}$ and $\mathcal{M}_{\mathcal{K}}$: The two-step abstraction process, consisting of PEFT and RFI, assumed that the conductance of the detailed models was the observable function for them. In other words, $O_1(V,t)+O_2(V,t)$ for $\mathcal{M}_{Na}$, and $O(V,t)$ for $\mathcal{M}_K$ mapped a state of the corresponding model to its output. The abstraction methodology described above is not restricted by these observable functions and can be adapted to arbitrary functions that map a state to its output.

Suppose we are given a stochastic (detailed) ion channel model $\mathcal{M}$, with a function that maps the state occupancy probability vector to a real-valued output. The goal is to reduce it to an HH-type abstraction, $\mathcal{H}$, that has a degree of activation λ and 1 as the degree of inactivation. We provide details for modifying PEFT, such that the resulting set of constant-voltage $\mathcal{H}^v$ systems are behaviorally equivalent to constant-voltage versions, $\mathcal{M}^v$, of the detailed model.

The first step is to establish a mapping between the states of $\mathcal{M}$ and a $2(\lambda+1)$-state stochastic model corresponding to the HH-type model, denoted by $\mathcal{H}_{stoch}$. We will label the states of $\mathcal{H}_{stoch}$ as x_{ij}, and denote the corresponding occupancy probability by p_{ij} where $i = 0, \dots, \lambda$ and $j = 0, 1$. The model $\mathcal{H}_{stoch}$ interprets the degrees of activation and inactivation as the number of independent activating and inactivating subunits of the channel. In our case, the state x_{ij} corresponds to the conformation of the channel where i activating and j inactivating subunits are in an "open" state that allow ion flow. Fig. 3.3 shows the model $\mathcal{H}_{stoch}$ corresponding to an HH-type model with $\lambda = 3$ and a degree of inactivation of 1. In the model, the state x_{21} corresponds to the a conformation where the inactivating subunit and two of the activating subunits are open. The inactivating subunit can close at a rate of $\beta_h(V)$ and change the conformation to x_{20}. The remaining activating subunit can open at the rate of $\alpha_m(V)$ to change the state to x_{31}. In this conformation, all the three activating subunits and the inactivating subunit are open. Thus, this state corresponds to the conformation of the channel which allows ion flow. From the state x_{21}, any of the two independent activating subunits could close at a rate of $2\beta_m(V)$ to change the state to x_{11}. Once we have mapped the states of the given stochastic model to the states of $\mathcal{H}_{stoch}$, PEFT can be modified to identify $\mathcal{H}^v_{stoch}$ systems that can match the behavioral traces observed from $\mathcal{M}^v$.



(a) Activating (m-type) subunit

(b) Inactivating (h-type) subunit

(c) Stochastic model for a channel 3 activating (λ = 3) and 1 inactivating subunits.

Figure 3.3: Invariant manifolds can be used to map the states of a $2(\lambda + 1)$-state stochastic model to an HH-type model with a degree of activation λ and degree of inactivation 1 ($\lambda = 3$ in the example).

The two-state HH-type model forms an invariant manifold [53] of $\mathcal{H}_{stoch}$. The occupancy probability of the state x_{ij} is given by $m^i h^j$. This correspondence helps us map a state vector of $\mathcal{H}_{stoch}$ to a state of the HH-type model (the vector $[m, h]^T$). Note that this mapping is exact and provides an output function that can be matched to the output function of $\mathcal{M}$. Suppose the output function maps states of $\mathcal{M}$ corresponding to the states x_{11} and x_{31} under the established mapping. Then the output function of $\mathcal{H}$ will take as arguments mh and m^3h. PEFT can be used to minimize divergence of this new observation function to identify $\mathcal{H}^v$ systems.

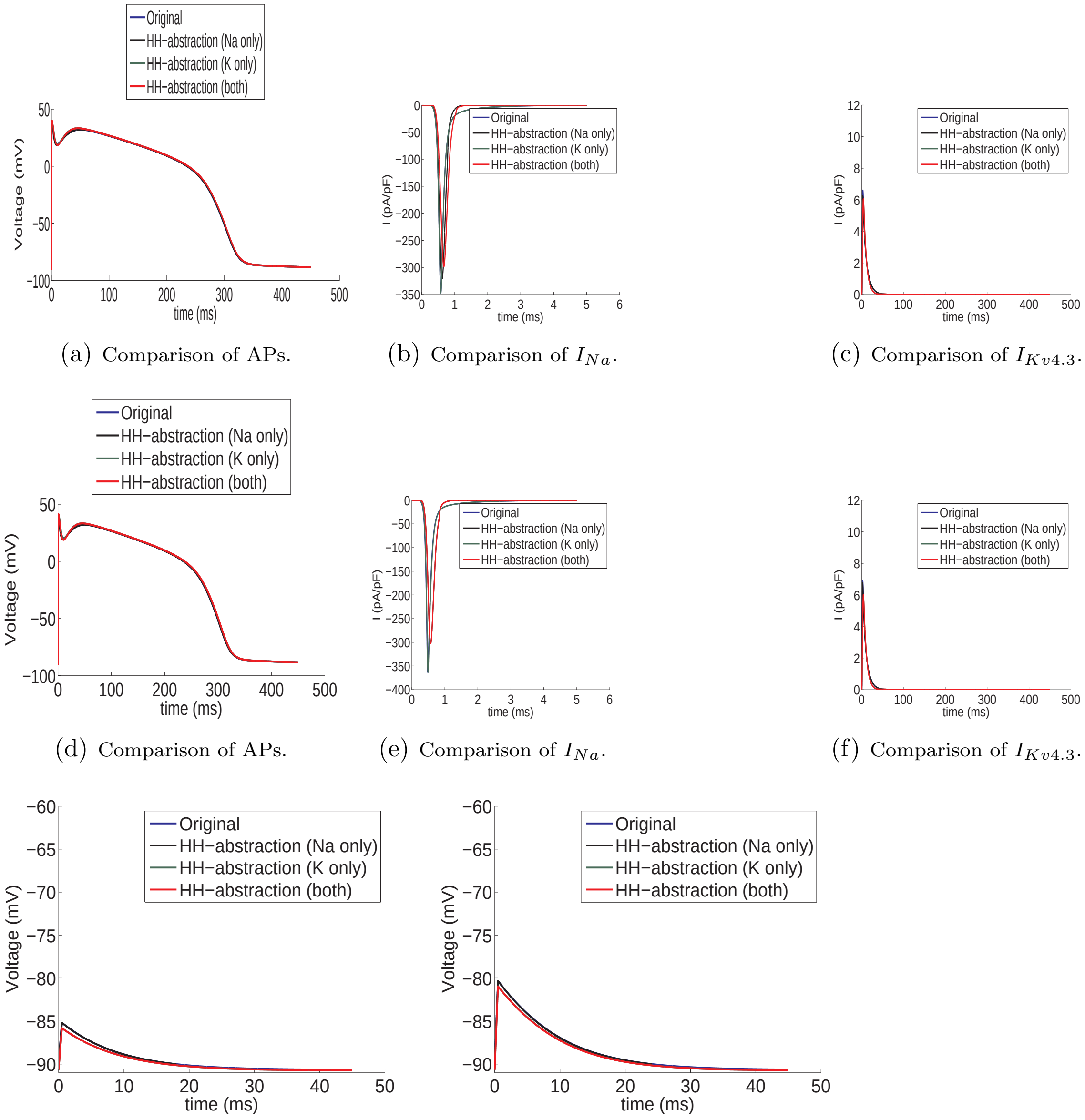
3.2 Results

The component models, $\mathcal{M}_{Na}$ and $\mathcal{M}_K$, were substituted by their respective HH-type abstractions $\mathcal{H}_{Na}$ and $\mathcal{H}_K$ within the whole-cell IMW model. The substitutions were done in three combinations: 1) substitution of $\mathcal{M}_{Na}$ only, 2) substitution of $\mathcal{M}_K$ only, and 3) substitution of both $\mathcal{M}_{Na}$ and $\mathcal{M}_K$. The modified IMW models were simulated in FORTRAN in all three cases with an integration time step of 0.001 ms. Both supra- and sub-threshold stimuli, lasting 0.5 ms, were used to excite the cardiac cell. Supra-threshold stimuli used were: $S1 = -100$ pA/pF, and $S2 = -120$ pA/pF. Sub-threshold stimuli employed were: $S3 = -10$ pA/pF, and $S4 = -20$ pA/pF.

S	V (mV)			I_{Na} (pA/pF)			I_K (pA/pF)		
	Na Only	K Only	Both	Na Only	K Only	Both	Na Only	K only	Both
S1	7.73×10^{-4}	1.4×10^{-3}	1.6×10^{-3}	2.7×10^{-3}	8.3×10^{-6}	5.2×10^{-3}	6.2×10^{-5}	1.72×10^{-4}	1.1×10^{-3}
S2	7.3×10^{-4}	1.4×10^{-3}	1.5×10^{-3}	5.2×10^{-3}	5.4×10^{-5}	5.2×10^{-2}	4.67×10^{-5}	2.54×10^{-4}	2.5×10^{-4}
S3	1.61×10^{-5}	1.1×10^{-3}	1.1×10^{-3}	-	-	-	-	-	-
S4	1.39×10^{-4}	1.2×10^{-3}	1.3×10^{-3}	-	-	-	-	-	-

Table 3.1: Mean L2 errors incurred in the simulations after substituting $\mathcal{M}_{Na}$ by $\mathcal{H}_{Na}$ and $\mathcal{M}_K$ by $\mathcal{H}_K$ in the IMW model. The first column, S, stands for the stimulus used to excite the cell at the beginning of the simulation. Only the voltage errors were recorded for the sub-threshold stimuli S3 and S4 as the currents were negligible.

Fig. 3.4 provides empirical evidence of the modified whole cell models being behaviorally equivalent to the original models. The model retains both normal and



(a) Comparison of APs.

(b) Comparison of I_{Na}.

(c) Comparison of $I_{Kv4.3}$.

(d) Comparison of APs.

(e) Comparison of I_{Na}.

(f) Comparison of $I_{Kv4.3}$.

(g) Comparison of APs ($S3$).

(h) Comparison of APs ($S4$).

Figure 3.4: Comparison of the original and the modified IMW models when $\mathcal{M}_{Na}$ and $\mathcal{M}_K$ are substituted by $\mathcal{H}_{Na}$ and $\mathcal{H}_K$ respectively. Subfigures (a)-(c) are obtained for the S1 stimulus, (d)-(f) for the S2 stimulus, (g) for the S3 stimulus and (h) for the S4 stimulus. S1 and S2 are supra-threshold stimuli and lead to an AP, whereas S3 and S4 being subthreshold stimuli fail to produce the AP.

anomalous cell-level behaviors on replacing the 13-state sodium channel and/or 10-state potassium channel components with the corresponding 2-state abstraction(s). In the next section, we formalize the equivalence of $\mathcal{M}_{Na}$ and $\mathcal{M}_K$ to $\mathcal{H}_{Na}$ and $\mathcal{H}_K$ respectively using approximate bisimulation.

3.3 Approximate Bisimulation Equivalence of M_I and M_H

In this section, we formalize the equivalence of the the detailed ion channel models, $\mathcal{M}_{Na}$ and $\mathcal{M}_K$, and their HH-type abstractions, $\mathcal{H}_{Na}$ and $\mathcal{H}_K$. Proposed in [35], the concept of *approximate bisimulation* enables us to formalize the approximate equivalence of two dynamical systems. The conventional notion of behavioral equivalence compares the trajectories of the two systems. Approximate bisimulation is stronger than approximate behavioral equivalence in that it allows for compositional reasoning. We will use the following two dynamical systems to define approximate bisimulation in a way similar to [34]:

$$\Delta_i : \begin{cases} \dot{\mathbf{x}}_i(t) & = f_i(\mathbf{x}_i(t), \mathbf{u}_i(t)) \\ \mathbf{y}_i(t) & = g_i(\mathbf{x}_i(t)) \end{cases}, i = 1, 2. \tag{3.2}$$

with the state vectors $\mathbf{x}_i \in \mathbb{R}^{n_i}$, the input vectors $\mathbf{u}_i \in \mathbb{R}^m$ and the output vectors $\mathbf{y}_i \in \mathbb{R}^p$. We assume that the dynamics are given by Lipschitz-continuous functions $f_i : \mathbb{R}^{n_i} \times \mathbb{R}^{n_i} \to \mathbb{R}^{n_i}$, $i = 1, 2$. We use $\mathbf{X}_i^0 \subseteq \mathbb{R}^{n_i}, i = 1, 2$, to denote the set of initial conditions for the two dynamical systems.

In our case, the ion channel models $\mathcal{M}_{Na}$ and $\mathcal{M}_K$ are dynamical systems with 13 and 10-dimensional state vectors, $\mathbf{p} \in \mathbb{R}^{13}_{\geq 0}$ and $\mathbf{q} \in \mathbb{R}^{10}_{\geq 0}$, respectively. The dynamics of $\mathbf{p}$ and $\mathbf{q}$, corresponding to $f(.,.)$ in Eq. (3.2), are given by Eqs. (2.6) and (2.9) respectively. Both models receive voltage, a 1-dimensional input and produce conductance, a 1-dimensional output. The output functions are $g_{Na}(\mathbf{p}) = O_1 + O_2$ and $g_{Na}(\mathbf{q}) = O$ as per Definitions 2.1.2 and 2.1.3. Table 4 of [47] provides the initial conditions for the two models. Both models have a single initial condition.

The HH-type abstractions, $\mathcal{H}_{Na}$ and $\mathcal{H}_K$, on the other hand have 2-dimensional state vectors, $[m_{Na}, h_{Na}]^T$ and $[m_K, h_K]^T$ respectively. The dynamics are as per Definition 2.1.1. The rate functions defining the two models were identified in the RFI step and are given in Section 3.1. Voltage, the input, and conductance, the output, are both 1-dimensional. The output function, as per Definition 2.1.1 is $m_i^\lambda h_i$, $i \in \{Na, K\}$, where $\lambda = 3$ for $\mathcal{H}_{Na}$ and $\lambda = 4$ for $\mathcal{H}_K$. In Section 3.1, PEFT identified the singleton initial conditions for them.

Definition 3.3.1. *The two dynamical systems defined in Eq. (3.2) are said to be* approximately bisimilar with precision δ*, denoted by* $\Delta_1 \cong_\delta \Delta_2$*, if there exists a relation* $B_\delta \subseteq \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ *such that:*

1. *For every* $\mathbf{x}_1 \in \mathbf{X}_1^0$*, there exists* $\mathbf{x}_2 \in \mathbf{X}_2^0$ *such that* $(\mathbf{x}_1, \mathbf{x}_2) \in B_\delta$*.*
2. *For every* $(\mathbf{x}_1, \mathbf{x}_2) \in B_\delta$*,* $d(\mathbf{y}_1, \mathbf{y}_2) \leq \delta$*, where* $\mathbf{y}_1$ *and* $\mathbf{y}_2$ *are the corresponding outputs and* d *is a metric distance defined on* $\mathbb{R}^p$*.*
3. *For every* $(\mathbf{x}_1, \mathbf{x}_2) \in B_\delta$*, for all* $T > 0$*,*
   (a) *If on the input signal* $\mathbf{u}(t)$*,* Δ_1 *produces the trajectory* $\mathbf{x}_1(t)$*, where* $\mathbf{x}_1(0) = \mathbf{x}_1$*, then* Δ_2 *evolves as per the trajectory* $\mathbf{x}_2(t)$*, where* $\mathbf{x}_2(0) = \mathbf{x}_2$*, such that* $(\mathbf{x}_1(t), \mathbf{x}_2(t)) \in B_\delta$ *for all* $t \in [0, T]$*.*
   (b) *If on the input signal* $\mathbf{u}(t)$*,* Δ_2 *produces the trajectory* $\mathbf{x}_2(t)$*, where* $\mathbf{x}_2(0) = \mathbf{x}_2$*, then* Δ_1 *evolves as per the trajectory* $\mathbf{x}_1(t)$*, where* $\mathbf{x}_1(0) = \mathbf{x}_1$*, such that* $(\mathbf{x}_1(t), \mathbf{x}_2(t)) \in B_\delta$ *for all* $t \in [0, T]$*.*

It should be noted that Definition 3.3.1 also applies when Δ_1 and Δ_2 are autonomous systems, i.e. they do not evolve according to an explicit input signal. In that case, the input signal can be considered as time. B_δ, which is used to denote the approximate bisimulation relation, contains corresponding states from trajectories of the two systems. These trajectories must be initialized from states related by B_δ. The constant voltage versions of the ion channel models, $\mathcal{M}_{Na}^v$ and $\mathcal{M}_K^v$, and their corresponding HH-type abstractions, $\mathcal{H}_{Na}^v$ and $\mathcal{H}_K^v$, are examples of autonomous dynamical systems. For all voltage values, the initial conditions of $\mathcal{M}_{Na}^v$ and $\mathcal{M}_K^v$ are the same as that of $\mathcal{M}_{Na}$ and $\mathcal{M}^K$ respectively. Similarly, for all voltage values, the initial conditions of $\mathcal{H}_{Na}^v$ and $\mathcal{H}_K^v$ are the same as that of $\mathcal{H}_{Na}$ and $\mathcal{H}_K$ respectively. Next, we state a lemma that leads to the approximate bisimilarity of the constant-voltage abstractions, $\mathcal{H}_{Na}^v$ and $\mathcal{H}_K^v$, and their corresponding detailed models $\mathcal{M}_{Na}^v$ and $\mathcal{M}_K^v$.

Lemma 1. *Consider the voltage values* $v \in [V_{res}, V_{max}]$ *that were used to simulate and fit the trajectories of,* $\mathcal{M}_{Na}^v$ *and* $\mathcal{M}_K^v$ *in PEFT. The corresponding HH-type abstractions* $\mathcal{H}_{Na}^v$ *and* $\mathcal{H}_K^v$ *are approximately bisimilar to the detailed models, i.e.* $\mathcal{M}_{Na}^v \cong_{\delta_{Na}^v} \mathcal{H}_{Na}^v$ *and* $\mathcal{M}_K^v \cong_{\delta_K^v} \mathcal{H}_K^v$*. The errors* δ_{Na}^v *and* δ_K^v *are the maximum errors incurred by the PEFT procedure while fitting the trajectories of* $\mathcal{M}_{Na}^v$ *and* $\mathcal{M}_K^v$ *respectively.*

Proof. Theorem 3 in Appendix A proves that for all the voltage values $v \in [V_{res}, V_{max}]$, the corresponding $\mathcal{M}_{Na}^v$ and $\mathcal{M}_K^v$ have stable equilibria. The fitting performed by

PEFT ensures that the corresponding HH-type abstractions $\mathcal{H}_{Na}^v$ and $\mathcal{H}_K^v$ also reach their steady state.

Let δ_{Na}^v and δ_K^v be the maximum error incurred by cftool while fitting the trajectories of $\mathcal{M}_{Na}^v$ and $\mathcal{M}_K^v$, respectively. The approximate bisimulation relation $B_{\delta_{Na}^v}$ relates the corresponding states on the unique trajectories of the two systems. Two states on the two trajectories are related if they are reached from their respectively initial conditions in the same time.

We are guaranteed that the errors in the observations will not deviate more than δ_{Na}^v and δ_K^v as the respective pairs of systems reach their steady state, thus ensuring $\mathcal{M}_{Na}^v \cong_{\delta_{Na}^v} \mathcal{H}_{Na}^v$ and $\mathcal{M}_K^v \cong_{\delta_K^v} \mathcal{H}_K^v$. □

Theorem 2. *Let $\mathcal{V}$ be the set of input signals, denoted by $V(t)$, such that $V(0) = V_{res}$ and the signals reach the resting potential, V_{res}, infinitely often. If the type of input signals are restricted to $\mathcal{V}$, then $\mathcal{M}_{Na} \cong_{\delta_{Na}} \mathcal{H}_{Na}$ and $\mathcal{M}_K \cong_{\delta_K} \mathcal{H}_K$, where $\delta_{Na} = 6 \times 10^{-3}$ and $\delta_K = 0.15$.*

Proof. Consider the input signal $V(t) \in \mathcal{V}$. As this input signal is provided to $\mathcal{M}_{Na}$ and $\mathcal{H}_{Na}$, we will compare the resulting trajectories $\mathbf{p}(t)$ and $[m_{Na}(t), h_{Na}(t)]^T$. A similar treatment will be given to the trajectories, $\mathbf{q}(t)$ and $[m_K(t), h_K(t)]^T$, of $\mathcal{M}_K$ and $\mathcal{H}_K$ respectively.

Consider the time interval $[t, t + \delta t]$. In the limit $\delta t \to 0$, we can consider the input $V(t) = v$ for $t \in [t, t + \delta t]$. The evolution of $\mathcal{M}_{Na}$ and $\mathcal{H}_{Na}$ in the interval $[t, t + \delta t]$ can be approximated by the evolution of the constant voltage dynamical systems $\mathcal{M}_{Na}^v$ and $\mathcal{H}_{Na}^v$ respectively.

For every voltage value v that the input signal $V(t)$ can take over the interval $[t, t + \delta t]$, there will be a value $v^* \in [V_{res}, V_{max}]$, which is closest to it among the voltage values used to simulate $\mathcal{M}_{Na}^{v^*}$ during PEFT. Thus, the divergence between the trajectories of $\mathcal{M}_{Na}^v$ and $\mathcal{H}_{Na}^v$ in the interval $[t, t + \delta t]$ can be broken down into three components:

- Divergence between $\mathcal{M}_{Na}^v$ and $\mathcal{M}_{Na}^{v^*}$
- Divergence between $\mathcal{M}_{Na}^{v*}$ and $\mathcal{H}_{Na}^{v^*}$
- Divergence between $\mathcal{H}_{Na}^{v*}$ and $\mathcal{H}_{Na}^v$

As per Lemma 1, $\mathcal{M}_{Na}^{v^*} \cong_{\delta_{Na}^{v^*}} \mathcal{H}_{Na}^{v^*}$. This ensures that the divergence between $\mathcal{M}_{Na}^{v^*}$ and $\mathcal{H}_{Na}^{v^*}$ is bounded by $\delta_{Na}^{v^*}$.

The divergence between $[m_{Na}^{v^*}(t), h_{Na}^{v^*}(t)]^T$, the trajectory of $\mathcal{H}_{Na}^{v*}$, and $[m_{Na}^v(t), h_{Na}^v(t)]^T$, the trajectory of $\mathcal{H}_{Na}^v$, can be bound using sensitivity analysis. We bound the sensitivity of the solutions $m_{Na}(t)$ and $h_{Na}(t)$ to a change in voltage from v to v^*. The sensitivity is bounded by taking the partial derivative of the solution in Eq. (2.4) with respect to the voltage V. Using chain rule we have,

$$\frac{\partial m}{\partial V} = \frac{\partial m}{\partial m_\infty}.\frac{\partial m_\infty}{\partial V} + \frac{\partial m}{\partial \tau_m}.\frac{\partial \tau_m}{\partial V} \qquad \frac{\partial h}{\partial V} = \frac{\partial h}{\partial h_\infty}.\frac{\partial h_\infty}{\partial V} + \frac{\partial h}{\partial \tau_h}.\frac{\partial \tau_h}{\partial V}$$

The maximum change in voltage, i.e. the difference between v^* and v, is given by the granularity of voltage values that were used in PEFT. When 20000 uniformly spaced values are taken in $[V_{res}, V_{max}]$, $max(|v-v^*|) \approx 0.007$. For $\mathcal{H}_{Na}$, $\frac{\partial m_\infty}{\partial V} \leq 0.001$, $\frac{\partial tau_m}{\partial V} \leq 0.0015$, $\frac{\partial h_\infty}{\partial V} \approx 0.0001$ and $\frac{\partial tau_h}{\partial V} \leq 0.0014$. For $\mathcal{H}_K$, $\frac{\partial m_\infty}{\partial V} \leq 4.84 \times 10^{-5}$, $\frac{\partial tau_m}{\partial V} \leq 6.73 \times 10^{-5}$, $\frac{\partial h_\infty}{\partial V} \approx 2.79 \times 10^{-9}$ and $\frac{\partial tau_h}{\partial V} \leq 2.3 \times 10^{-6}$.

Performing a similar analysis for bounding the divergence between $\mathcal{M}^v_{Na}$ and $\mathcal{M}^{v^*}_{Na}$ is slightly more complicated. The solutions of the two 13×13 systems depends the matrix exponentials $e^{A(v)}$ and $e^{A(v^*)}$, where A is the rate matrix defined in Definition 2.1.2. The matrix exponentials are determined by the spectrum (eigenvalues) of A. Thus the sensitivity of the solutions of $\mathcal{M}^v_{Na}$ and $\mathcal{M}^{v^*}_{Na}$ can be bound by bounding the change in the spectrum of A due to a change in voltage from v to v^*. Bauer-Fike theorem [6] can be used to bound the perturbation in the spectrum. The maximum change in the eigenvalue is bound by a factor of $max(|v - v^*|) \approx 0.007$.

Combining the three components of errors, we get $\delta_{Na} = 6 \times 10^{-3}$. Note that this is a conservative bound and is much more than the the empirical evidence shown in Section 4. A similar analysis can be performed to bound the divergence of $\mathcal{M}_K$ and $\mathcal{H}_K$, to get $\delta_K = 0.15$. □

Chapter 4

Bisimulation Function

Incremental *Input-to-State Stability* (ISS) of a pair of dynamical systems refers to the property that bounded differences in their input signals lead to bounded differences in their resulting state trajectories. Incremental *Input-to-Output Stability* (IOS) generalizes incremental Input-to-State Stability (ISS) to systems with output maps. Since the seminal work of Sontag [82, 83, 84], the $\mathcal{K}$, $\mathcal{KL}$, and $\mathcal{K}_\infty$ classes of Kamke functions have been used to characterize ISS of dynamical systems as extensions of Lyapunov stability; see [57]. These Lyapunov-like functions have been used in the small-gain theorems of [94] to establish stability of feedback-based interconnected systems, thereby enabling compositional design of nonlinear control systems.

Similar to Kamke and Lyapunov functions, *Bisimulation Functions* (BFs) have played a transformative role in extending the control-theoretic notions of Lyapunov Stability and ISS to system verification. BFs [31, 33, 35, 29, 49] are Lyapunov-like functions that decay along the trajectories of a given pair of dynamical systems. Level sets of BFs yield approximate bisimulation relations that generalize the classical notion of bisimulation equivalence of finite-state systems [68] to real-valued continuous-time dynamical systems. BFs also allow one to show that a system is robust to bounded deviations in the input signals.

BFs can also be used to reason *compositionally* about dynamical systems. Consider a dynamical system D with a subsystem S connected to the rest of D through a feedback loop. Moreover, suppose we have an approximately equivalent version S' of S that uses fewer state variables than S. That is, S' is an *abstraction* or model-order reduction of S, and by substituting S' for S in D one would hope to obtain the corresponding model-order reduction in D. Care must be taken in this situation, however, as the approximation error between S and S' may get amplified by the feedback context in which S resides.

As shown in [29, 63], one can appeal to a small-gain theorem to compute BFs that *bound the error* that is introduced when substituting S' for S within D. BFs can also be used in other system design and verification settings, including controller design [36, 67], reachability analysis [61], and simulation-based verification [43, 14].

The following definition of BFs is adapted from [29], where $\parallel . \parallel$ is used as $\mathcal{L}2$-norm.

Definition 4.0.1. *Let $\Sigma_i = (\mathcal{X}_i, \mathcal{X}_i^0, \mathcal{U}, f_i, \mathcal{Y}, g_i)$, $i = 1, 2$, be two dynamical systems such that $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$, $\mathcal{U} \subseteq \mathbb{R}^m$ and $\mathcal{Y} \subseteq \mathbb{R}^p$.*

A bisimulation function *(BF) is a smooth function $S : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}_{\geq 0}$ such that for every $\mathbf{x}_1 \in \mathcal{X}_1$, $\mathbf{x}_2 \in \mathcal{X}_2$, $\mathbf{u}_1, \mathbf{u}_2 \in \mathcal{U}$:*

$$\parallel g_1(\mathbf{x}_1) - g_2(\mathbf{x}_2) \parallel \leq S(\mathbf{x}_1, \mathbf{x}_2), \tag{4.1}$$

$$\begin{aligned} \exists \lambda > 0, \gamma \geq 0 : \frac{\partial S}{\partial \mathbf{x}_1} f_1(\mathbf{x}_1, \mathbf{u}_1) &+ \frac{\partial S}{\partial \mathbf{x}_2} f_2(\mathbf{x}_2, \mathbf{u}_2) \\ &\leq -\lambda S(\mathbf{x}_1, \mathbf{x}_2) + \gamma \parallel \mathbf{u}_1 - \mathbf{u}_2 \parallel \end{aligned} \tag{4.2}$$

4.1 IOS Theorem of BFs

The Theorem 1 presented in [29] is modified to incorporate the adapted definition of BFs and presented below.

Theorem 4.1.1. *Let S be a BF with parameters λ and γ between dynamical systems Σ_i, $i = 1, 2$, and let $\mathbf{x}_1(t)$ and $\mathbf{x}_2(t)$ be two trajectories of the systems. For all $t \geq 0$,*

$$\begin{aligned} \parallel g_1(\mathbf{x}_1(t)) - g_2(\mathbf{x}_2(t)) \parallel &\leq S(\mathbf{x}_1(t), \mathbf{x}_2(t)) \\ &\leq e^{-\lambda t} S(\mathbf{x}_1(0), \mathbf{x}_2(0)) + \\ &\quad \frac{\gamma}{\lambda} \parallel \mathbf{u}_1 - \mathbf{u}_2 \parallel_\infty \end{aligned}$$

where $\parallel \mathbf{u}_1 - \mathbf{u}_2 \parallel_\infty = sup_{t \geq 0} \parallel \mathbf{u}_1(t) - \mathbf{u}_2(t) \parallel$ denotes the maximum difference in the input signals being fed to the two systems.

Proof. From Eq. (4.1), we have the first inequality. From Eq. (4.2), we have

$$\begin{aligned} \frac{dS(\mathbf{x}_1(t), \mathbf{x}_2(t))}{dt} &\leq -\lambda S(\mathbf{x}_1(t), \mathbf{x}_2(t)) + \gamma \parallel \mathbf{u}_1(t) - \mathbf{u}_2(t) \parallel \\ &\leq -\lambda S(\mathbf{x}_1(t), \mathbf{x}_2(t)) + \gamma \parallel \mathbf{u}_1 - \mathbf{u}_2 \parallel_\infty \end{aligned}$$

Let $\eta(t) = e^{-\lambda t} S(\mathbf{x}_1(0), \mathbf{x}_2(0)) + \frac{\lambda}{\gamma} \parallel \mathbf{u}_1 - \mathbf{u}_2 \parallel_\infty$. It is a solution of the differential equation $\dot{\eta}(t) = -\lambda \eta(t) + \gamma \parallel \mathbf{u}_1 - \mathbf{u}_2 \parallel_\infty$. Moreover, $S(\mathbf{x}_1(0), \mathbf{x}_2(0)) \leq \eta(0)$; then from the funnel theorem, it follows that $\forall t \geq 0, S(\mathbf{x}_1(t), \mathbf{x}_2(t)) \leq \eta(t)$. □

It follows from Theorem 4.1.1 that the existence of a BF between two dynamical systems is a proof of the following IOS property: when *approximately equal* signals are fed as inputs to the two systems, they produce *approximately equal* outputs. When applied to one system, a BF demonstrates the *robustness* of the system to small variations in the input signal. The Lyapunov-like functions-based approach is an extension of *input-to-state* stability of [84].

4.2 Compositionality Theorem of BFs

When subsystems are connected using feedback, their respective BFs can be composed subject to a small-gain condition. We formalize this idea by stating a result based on Theorem 2 of [29].

Theorem 4.2.1. *Let $\Sigma_i = (\mathcal{X}_i, \mathcal{X}_i^0, \mathcal{U}_i, f_i, \mathcal{O}_i, g_i)$, $i = 1, 2, A, B$, be dynamical systems such that $\mathcal{U}_1 = \mathcal{O}_A$, $\mathcal{U}_A = \mathcal{O}_1$, $\mathcal{U}_2 = \mathcal{O}_B$ and $\mathcal{U}_B = \mathcal{O}_2$. Let S_{12}, parameterized by λ_{12} and γ_{12}, be a BF between Σ_1 and Σ_2. Let S_{AB}, parameterized by λ_{AB} and γ_{AB}, be a BF between Σ_A and Σ_B.*

Let $\Sigma_{A1} = \Sigma_A || \Sigma_1$ and $\Sigma_{B2} = \Sigma_B || \Sigma_2$. If the small-gain condition *(SGC) $\frac{\gamma_{AB}\gamma_{12}}{\lambda_{AB}\lambda_{12}} < 1$ is met, then a BF S can be constructed between Σ_{A1} and Σ_{B2} by composing S_{AB} and S_{12} as follows:*

$$S(\mathbf{x}_{A1}, \mathbf{x}_{B2}) = \alpha_1 S_{AB}(\mathbf{x}_A, \mathbf{x}_B) + \alpha_2 S_{12}(\mathbf{x}_1, \mathbf{x}_2) \tag{4.3}$$

where $\mathbf{x}_{A1} = [\mathbf{x}_A, \mathbf{x}_1]^T$ and $\mathbf{x}_{B2} = [\mathbf{x}_B, \mathbf{x}_2]^T$ and the constants α_1 and α_2 are given by:

$$\begin{cases} \frac{\gamma_{12}}{\lambda_{AB}} < \alpha_1 < \frac{\lambda_{12}}{\gamma_{AB}} \quad \textit{and} \quad \alpha_2 = 1 \quad \textit{if} \quad \lambda_{AB} \leq \gamma_{12} \\ \alpha_1 = 1 \quad \textit{and} \quad \frac{\gamma_{AB}}{\lambda_{12}} < \alpha_2 < \frac{\lambda_{AB}}{\gamma_{12}} \quad \textit{if} \quad \lambda_{12} \leq \gamma_{AB} \\ \alpha_1 = 1 \quad \textit{and} \quad \alpha_2 = 1 \quad \textit{in other cases} \end{cases} \tag{4.4}$$

Proof. Consider S be a function of the form (5.4.1). We will find conditions on α_1 and α_2 such that S is a BF between Σ_{A1} and Σ_{B2}. If $\alpha_1 \geq 1$ and $\alpha_2 \geq 1$, then

$$\begin{aligned} S(\mathbf{x}_{A1}, \mathbf{x}_{B2}) &\geq S_{AB}(\mathbf{x}_A, \mathbf{x}_B) + S_{12}(\mathbf{x}_1, \mathbf{x}_2) \\ &\geq \parallel g_A(\mathbf{x}_A) - g_B(\mathbf{x}_B) \parallel + \parallel g_1(\mathbf{x}_1) - g_2(\mathbf{x}_2) \parallel \end{aligned}$$

because S_{AB} and S_{12} satisfy Eq. (4.1). The observational difference of Σ_{A1} and Σ_{B2}:

$$\begin{aligned} \parallel g_{A1}(\mathbf{x}_{A1}) - g_{B2}(\mathbf{x}_{B2}) \parallel &= \\ \sqrt{\parallel g_A(\mathbf{x}_A) - g_B(\mathbf{x}_B) \parallel^2 + \parallel g_1(\mathbf{x}_1) - g_2(\mathbf{x}_2) \parallel^2} & \\ \leq \parallel g_A(\mathbf{x}_A) - g_B(\mathbf{x}_B) \parallel + \parallel g_1(\mathbf{x}_1) - g_2(\mathbf{x}_2) \parallel & \end{aligned}$$

Therefore, it shows that S satisfies Eq. (4.1). Applying similar steps as in [29], we can write the following:

$$\begin{aligned} &\frac{\partial S}{\partial \mathbf{x}_{A1}} f_{A1}(\mathbf{x}_{A1}, \mathbf{u}_{A1}) + \frac{\partial S}{\partial \mathbf{x}_{B2}} f_{B2}(\mathbf{x}_{B2}, \mathbf{u}_{B2}) \\ \leq -(\alpha_1 \lambda_{AB} - \alpha_2 \gamma_{12}) S_{AB}(\mathbf{x}_A, \mathbf{x}_B) &- (\alpha_2 \lambda_{12} - \alpha_1 \gamma_{AB}) S_{12}(\mathbf{x}_1, \mathbf{x}_2) \end{aligned}$$

If $(\alpha_1 \lambda_{AB} - \alpha_2 \gamma_{12}) > 0$ and $(\alpha_2 \lambda_{12} - \alpha_1 \gamma_{AB}) > 0$ and $\lambda = \min(\frac{(\alpha_1 \lambda_{AB} - \alpha_2 \gamma_{12})}{\alpha_1}, \frac{(\alpha_2 \lambda_{12} - \alpha_1 \gamma_{AB})}{\alpha_2})$, then

$$\frac{\partial S}{\partial \mathbf{x}_{A1}} f_{A1}(\mathbf{x}_{A1}, \mathbf{u}_{A1}) + \frac{\partial S}{\partial \mathbf{x}_{B2}} f_{B2}(\mathbf{x}_{B2}, \mathbf{u}_{B2}) \leq -\lambda S(\mathbf{x}_{A1}, \mathbf{x}_{B2}).$$

Therefore, S will be a BF if $\alpha_1 \geq 1$, $\alpha_2 \geq 1$, $(\alpha_1 \lambda_{AB} - \alpha_2 \gamma_{12}) > 0$ and $(\alpha_2 \lambda_{12} - \alpha_1 \gamma_{AB}) > 0$. As shown in [29], these four conditions can be expressed as $\frac{\lambda_{AB} \lambda_{12}}{\gamma_{AB} \gamma_{12}} < 1$. □

Chapter 5

BFComp Framework

In this chapter, we present ***BFComp***: *an automated framework for computing BFs that characterize IOS of dynamical systems.* ***BFComp***, which is illustrated in Fig. 5.1, leverages acSOS optimization and δ-decidability over the reals [28], a new form of Satisfiability Modulo Theory (SMT), to compute BFs. In addition to establishing IOS, ***BFComp*** is designed to provide tight bounds on the squared output errors between systems whenever possible.

An overview of ***BFComp*** is as follows. Given a pair of dynamical systems Σ_1 and Σ_2, an Sum Of Square Programming (SOSP) called SOSP 2 is formulated and solved using MATLAB SOSTOOLS [76]. SOSP 2 requires the decay parameter λ, the gain parameter γ, and so-called *descriptor functions* that characterize the bounded state and input spaces. If the resulting BF provides satisfactory bounds on the output error, then the BF computation terminates.

Otherwise, an alternative SOSP formulation, SOSP 1, is called upon. SOSP 1, which we recently proposed in [63], uses λ and γ to compute a *Candidate BF* (CBF) that satisfies the decay condition of [29] only across a discretized grid over the bounded input space. ***BFComp*** then appeals to the δ-decidability-based dReal [28] to verify that the decay requirement, which is encoded by the SMT formula ψ, is exhaustively satisfied over the exterior of the CBF's l-level set.

Level sets are used here because dReal relies fundamentally on the technique of δ-relaxation, which may lead to spurious counterexamples. Taking the level set of the CBF filters out the origin and a finite-sized neighborhood around it, which gives rise to the spurious counterexamples. Starting from a relatively small (aggressive) value, which retains most of the state space in the domain of ψ, the parameter l is (iteratively) tuned to filter ψ's domain to avoid such counterexamples. A positive result by dReal implies that the CBF is actually a valid BF everywhere outside the l-level set. If a (genuine) counterexample $\mathbf{c} = (\mathbf{x}_1, \mathbf{x}_2, u_1, u_2)$ to ψ is found, then $\mathbf{c}$

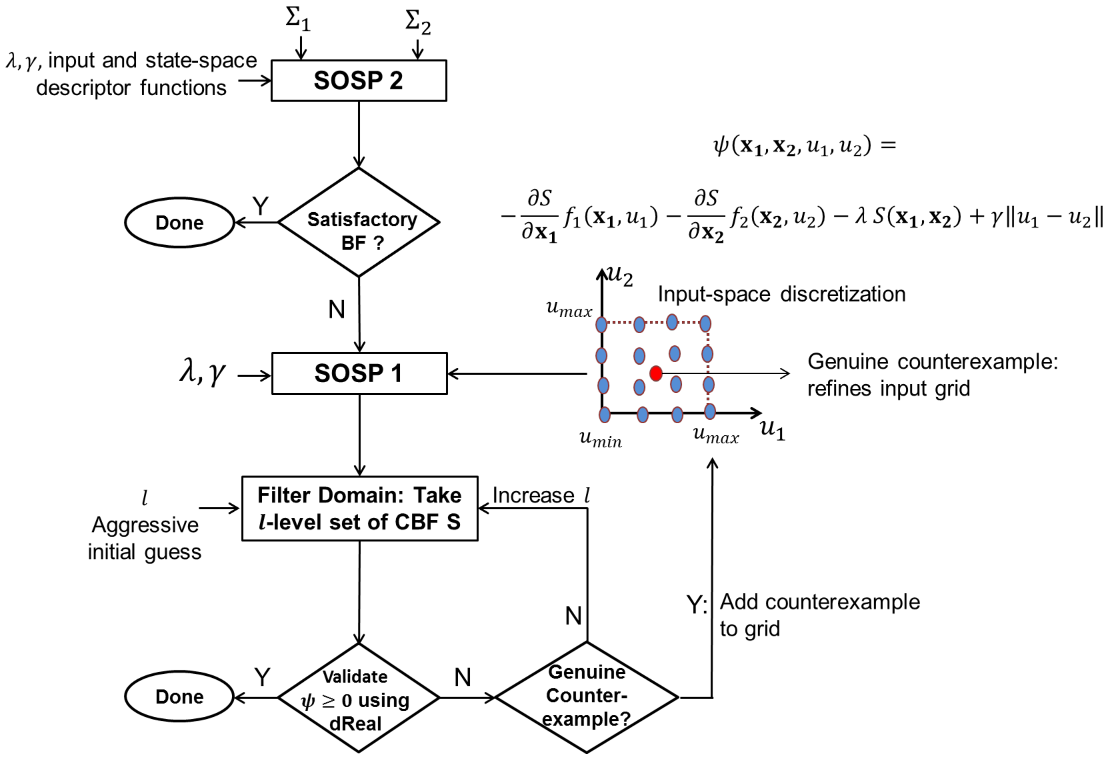

Σ_1
Σ_2
λ, γ, input and state-space descriptor functions
SOSP 2
$\psi(\mathbf{x_1}, \mathbf{x_2}, u_1, u_2) =$
$-\frac{\partial S}{\partial \mathbf{x_1}} f_1(\mathbf{x_1}, u_1) - \frac{\partial S}{\partial \mathbf{x_2}} f_2(\mathbf{x_2}, u_2) - \lambda\, S(\mathbf{x_1}, \mathbf{x_2}) + \gamma \|u_1 - u_2\|$
Done
Y
Satisfactory BF ?
N
u_2
Input-space discretization
u_{max}
λ, γ
SOSP 1
Genuine counterexample: refines input grid
u_{min}
u_{max}
u_1
l
Aggressive initial guess
Filter Domain: Take l-level set of CBF S
Increase l
N
Y: Add counterexample to grid
Done
Y
Validate $\psi \geq 0$ using dReal
N
Genuine Counter-example?

Figure 5.1: ***BFComp***: An Automated Framework for Computing BFs using SOS Optimization and δ-Decidability.

is used to refine the input-space grid.

To illustrate the utility of ***BFComp***, we apply it to the model-order reduction of a canonical cardiac-cell model [63]. In particular, we use our framework to compute BFs that appeal to the small-gain theorem of [29] to establish that the four-variable Markovian potassium-channel component of the cell model can be safely replaced by an approximately equivalent one-variable abstraction. The canonical model captures the feedback-based interconnection of the four-variable model within the detailed 67-variable Iyer-Ma-zhari-Winslow (IMW) ventricular cell model [47]. To the best of our knowledge, this is the first compositional proof of a feedback-based approximate model-order reduction of a biological system.

5.1 Background

In this section, we present physiological background by introducing the four-variable Markovian subsystem for the I_{Ks} current of the IMW model.

We define dynamical systems using a 6-tuple $(\mathcal{X}, \mathcal{X}^0, \mathcal{U}, f, \mathcal{O}, g)$, where $\mathcal{X}$ is the *state space*, $\mathcal{X}^0 \subseteq \mathcal{X}$ is the set of *initial conditions*, $\mathcal{U}$ is the *input space*, $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ is the *vector field* defining the dynamics, $\mathcal{O}$ is the set of *outputs*, and $g : \mathcal{X} \to \mathcal{O}$ maps a state to its output.

Next, we introduce the detailed Markovian potassium-channel model, which is employed as a component in the Iyer-Mazhari-Winslow (IMW) ventricular myocyte model [47].

5.1.1 The Potassium-Channel Subsystem

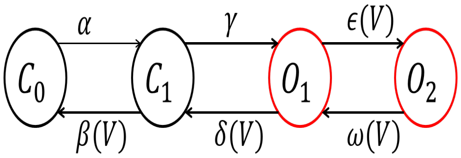


Figure 5.2: Σ_K: the detailed potassium-channel model, corresponding to the ionic current I_{Ks} in the IMW model.

Definition 5.1.1. The *potassium channel model* Σ_K is given by $(X, X^0, \mathcal{V}, f_K, \mathcal{O}, g_K)$. A *state* $\mathbf{x} \in X \subseteq \mathbb{R}^4_{\geq 0}$ is the occupancy probability distribution over the four states of the voltage-controlled Continuous Time Markov Chain (CTMC) shown in Fig. 5.2 in the following order of the state labels: $[C_0, C_1, O_1, O_2]$. The dynamics f_K is given by

$$f_K : \dot{\mathbf{x}} = A_K(V)\, \mathbf{x}, \tag{5.1}$$

where $V \in \mathcal{V} \subseteq \mathbb{R}$, the transmembrane voltage, is the input to the system and $A_K(V)$ is the 4×4 voltage-controlled rate matrix. The off-diagonal entry $A_K(i,j), i \neq j$, is the transition rate from state $\mathbf{x}_j$ to state $\mathbf{x}_i$. For example, $A_K(3,4) = \omega(V)$, the transition rate from O_2 to O_1. The diagonal entry $A_K(i,i)$ is the negative of the sum of all the outgoing rates from state $\mathbf{x}_i$. The transition rates are

$$\begin{aligned}\alpha &= 7.956 \times 10^{-3},\\ \beta(V) &= 0.216 \times exp(-0.00002\ V),\\ \gamma &= 3.97 \times 10^{-2},\\ \delta(V) &= (7 \times 10^{-3}) \times exp(-0.15\ V),\\ \epsilon(V) &= (7.67 \times 10^{-3}) \times exp(0.087\ V), \text{and}\\ \omega(V) &= (3.8 \times 10^{-3}) \times exp(-0.014\ V).\end{aligned}$$

The set of outputs $\mathcal{O} \subseteq \mathbb{R}_{\geq 0}$ contains the conductance values for the states. Given a state $\mathbf{x}$, $g_K(\mathbf{x}) \triangleq \mathbf{x}_3 + \mathbf{x}_4$ maps it to its conductance given by the sum of the occupancy probabilities of the states labeled O_1 and O_2. The system has a single initial condition $\mathbf{x}_0 = [0.9646, 0.03543, 2.294 \times 10^{-7}, 4.68 \times 10^{-11}] \in X^0$, as per Table 4 of [47].

Next, we define a one-variable abstraction for Σ_K.

5.1.2 Model-Order Reduction of Σ_K

The curve fitting-based approach of [70, 45] can be used to identify the following one-variable Hodgkin Huxley (HH)-type approximation for Σ_K.

Definition 5.1.2. *The* HH-type abstraction Σ_H *is given by* $(Y, Y^0, \mathcal{V}, f_H, \mathcal{O}, g_H)$. *A* state $y \in Y \subseteq \mathbb{R}_{\geq 0}$ *denotes the value of an activating (m-type) subunit. The dynamics* f_H *is given by*

$$f_H : \dot{y} = \alpha_m(V)(1-y) - \beta_m(V)y, \tag{5.2}$$

where $V \in \mathcal{V} \subseteq \mathbb{R}$*, the transmembrane voltage, is the input to the system. The rate functions* $\alpha_m(V)$ *and* $\beta_m(V)$*, identified using the two-step curve fitting-based approach of [70, 45], are as follows.*

$$\begin{aligned}\alpha_m(V) &= (-1.331 \times 10^{-10})V^4 - (2.466 \times 10^{-7})V^3\\ &\quad - (9.723 \times 10^{-6})V^2 - 0.0001231V + 0.001049 \qquad (5.3)\\ \beta_m(V) &= (4.788 \times 10^{-10})V^6 - (1.547 \times 10^{-8})V^5\\ &\quad + (1.642 \times 10^{-7})V^4 - (2.85 \times 10^{-6})V^3\\ &\quad + (6.704 \times 10^{-5})V^2 - (0.0007041)V + 0.003285. \qquad (5.4)\end{aligned}$$

The set of outputs $\mathcal{O} \subseteq \mathbb{R}_{\geq 0}$ *contains the conductance values for the states. Given a state* $\mathbf{y}$*,* $g_H(y) \triangleq y$ *maps it to its conductance. The system has a single initial condition* $y_0 = 1.32 \times 10^{-5}$*.*

5.2 Canonical Cell Models and Compositional Reasoning

In this section, we setup our case study on approximate model-order reduction within feedback loops. We first introduce the voltage subsystem Σ_C representing the cell membrane, which we compose with Σ_K and Σ_H to obtain two *Canonical Cell Models* (CCMs). We then state our compositionality result in terms of the two CCMs, and show how BFs can be used to prove the result.

Definition 5.2.1. *The* voltage subsystem Σ_C *is a capacitor-like model given by* $(\mathcal{V}, \mathcal{V}^0, \mathcal{O},$
$f_C, \mathcal{V}, g_C)$*. State* $V \in \mathcal{V} \subseteq \mathbb{R}$ *is the voltage. The dynamics of* Σ_C *is given by*

$$f_C : \dot{V} = -G_K(V - E_K)\ O, \tag{5.5}$$

where $G_K = 90.58$ *and* $E_K = -35\ mV$ *are the parameters of the model, and* $O \in \mathcal{O} \subseteq \mathbb{R}_{\geq 0}$*, the conductance of the potassium channel, is* Σ_C*'s input. The system outputs its state, i.e., for* $V \in \mathcal{V}$*,* $g_C(V) = V$*, and the initial condition is* $V_0 = 0\ mV$*.*

As per Eq. (5.5), V_K represents the equilibrium for a fixed-conductance input. Thus, V takes values in $[-35,\ 0]$.

In the case of detailed cardiac cell models, such as the IMW model, ion-channel subsystems such as Σ_K and Σ_H take voltage as input from the rest of the model and provide the conductance of the channel as the output. The rest of the model takes the channel conductance as input and outputs the voltage, which is then fed back to the ion-channel subsystems. Next, we define CCMs Σ_{CK} and Σ_{CH} that reflect this feedback-based composition; see Fig. 5.3. The models are canonical in the sense that other ion-channel subsystems can be added to obtain the complete IMW model.

Definition 5.2.2. *Systems* Σ_{CK} *and* Σ_{CH} *(see Fig. 5.3) are obtained by performing feedback-composition on the voltage subsystem* Σ_C *with ion-channel subsystems* Σ_K *and* Σ_H*, respectively; i.e.,* $\Sigma_{CK} = \Sigma_C || \Sigma_K$ *and* $\Sigma_{CH} = \Sigma_C || \Sigma_H$*. The state spaces, initial conditions, dynamics and outputs are inherited from the subsystems, as explained below. Both* Σ_{CK} *and* Σ_{CH} *are autonomous systems and do not receive any external inputs.*

A state of Σ_{CK} *is given by* $[\mathbf{x}, V_K]^T$*, where* $\mathbf{x}$ *is a state of* Σ_K *and* V_K *is a state of* Σ_C*. The subscript* K *in* V_K *is used to denote the copy of* Σ_C *composed with* Σ_K*. The system dynamics are given by Eqs. (5.1) and (5.5). The output is given by* $[g_K(\mathbf{x}), V_K]^T$*. The initial condition is the pair of the initial conditions of* Σ_K *and* Σ_C*.*

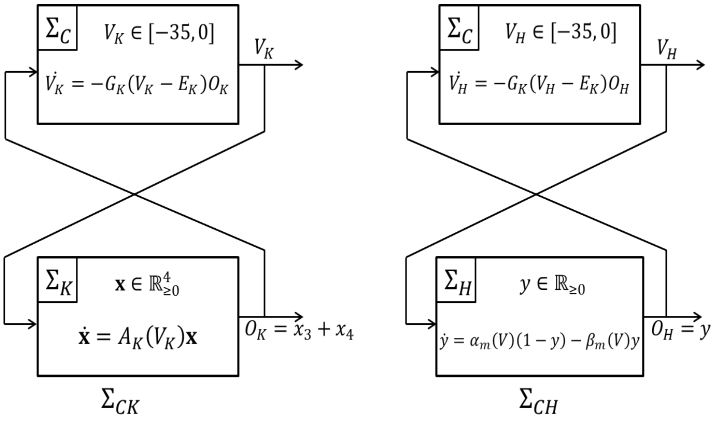

Σ_C
$V_K \in [-35, 0]$
$\dot{V_K} = -G_K(V_K - E_K)O_K$
V_K
Σ_K
$\mathbf{x} \in \mathbb{R}^4_{\geq 0}$
$\dot{\mathbf{x}} = A_K(V_K)\mathbf{x}$
$O_K = x_3 + x_4$
Σ_{CK}
Σ_C
$V_H \in [-35, 0]$
$\dot{V_H} = -G_K(V_H - E_K)O_H$
V_H
Σ_H
$y \in \mathbb{R}_{\geq 0}$
$\dot{y} = \alpha_m(V)(1-y) - \beta_m(V)y$
$O_H = y$
Σ_{CH}

Figure 5.3: Σ_{CK} and Σ_{CH}: ion-channel subsystems Σ_K and Σ_H are feedback-composed with Σ_C, which represents the cell membrane. Σ_{CH} is obtained by i) identifying the one-variable abstraction Σ_H of Σ_K using the curve-fitting procedure given in [70, 45]; and ii) substituting Σ_H for the detailed model Σ_K within Σ_{CK}.

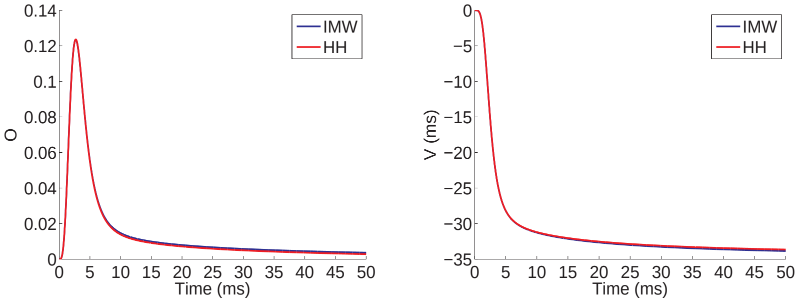
A state of Σ_{CH} is given by $[y, V_H]^T$, where y denotes a state of Σ_H and V_H denotes a state of Σ_C. The subscript H in V_H is used to denote the copy of Σ_C composed with Σ_H. The system dynamics are given by Eqs. (5.2) and (5.5). The output is given by $[g_H(y), V_H]^T$. The initial condition is the pair of the initial conditions of Σ_H and Σ_C.

When Σ_K in Σ_{CK} is replaced by Σ_H to obtain Σ_{CH}, the behaviors of the composite CCMs might diverge. This is due to the feedback composition that tends to amplify deviations in the outputs of either of the subsystems. Fig. 5.4 shows a pair of trajectories of Σ_{CK} and Σ_{CH} that start from nominal initial conditions.

The goal of the paper is to compute BFs that prove that the composite CCMs are indeed approximately equivalent, i.e., the following statement holds.

Compositionality Result: There exists a BF S between Σ_{CK} and Σ_{CH} that renders the two CCMs to be approximately equivalent as characterized by Theorem 4.1.1.

S is computed compositionally as follows. First, the components Σ_K and Σ_H are proved to be approximately equivalent by computing a BF S_{KH} between the two systems. Then, the context Σ_C is proved to be robust to input deviations by computing a BF S_C for it. The computation procedure ensures that the prerequisite small-gain condition is satisfied by S_{KH} and S_C, thereby enabling the application of Theorem 4.2.1; this results in a BF S between Σ_{CK} and Σ_{CH}. Next, we describe ***BFComp*** and apply it for computing S_{KH} and S_C in the following sections.



(a) $O_K(t)$ and $O_H(t)$ of Σ_{CK} and Σ_{CH} respectively.

(b) $V_K(t)$ and $V_H(t)$ of Σ_{CK} and Σ_{CH} respectively.

Figure 5.4: Simulations of Σ_{CK} and Σ_{CH}: when Σ_K is replaced by Σ_H, feedback composition tends to accumulate error incurred due to the abstract component. The mean L1 errors: O_{Ks} : 1.1786×10^{-4}, V: $0.2002\ mV$.

5.3 Computing BFs using SOS Optimization

In this section, we will present both SOSP 1 and SOSP 2 formulation to compute BFs.

5.3.1 Computing CBFs using SOSP 1

In [63, 1], we presented SOSP 1, a computation procedure based on SOS optimization for computing CBFs. In this subsection, we review the algorithm and comment on its input-space sampling approach.

A multivariate polynomial $p(x_1, x_2, \ldots, x_n) = p(\mathbf{x})$ is an *SOS polynomial* if there exist polynomials $f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})$ such that $p(\mathbf{x}) = \sum_{i=1}^{m} f_i^2(\mathbf{x})$. For example, $p(x, y) = x^2 - 6xy + 12y^2$ is an SOS polynomial; it can be expressed as $(x - 3y)^2 + (\sqrt{3}y)^2$. We denote the set of all SOS polynomials by $\mathbb{S}$.

An *SOS optimization Problem* (SOSP), involves finding an $S \in \mathbb{S}$ such that a linear objective function, whose decision variables are the coefficients of S, is optimized. The constraints of the problem are linear in the decision variables. A formal definition of an SOSP can be found in the SOSTOOLS user guide (p. 7).

Consider two dynamical systems $(X_i, \{\mathbf{x}_i^0\}, [u_{min}, u_{max}], f_i, O, g_i)$, $i = 1, 2$, with u_1 and u_2 being the scalar inputs of the two systems. Let $\mathcal{U}^G$ represent a discretized grid for u_1 and u_2. The grid is formed by dividing the input space $[u_{min}, u_{max}]$ into a finite number of uniformly spaced intervals, and (u_1^i, u_2^j) denotes the pair of inputs where u_1 takes the i^{th} value and u_2 takes the j^{th} value. In [63, 1], we presented the following SOSP for computing BFs using SOS optimization.

Definition 5.3.1. SOSP 1*, as per [63, 1], is defined by the following equations.*

$$\begin{aligned}
&\textit{Minimize } S(\mathbf{x}_1^0, \mathbf{x}_2^0) && (5.6)\\
&\textit{subject to:}\\
&-S(\mathbf{x}_1, \mathbf{x}_2) + [g_1(\mathbf{x}_1) - g_2(\mathbf{x}_2)]^2 \in \mathbb{S}, && (5.7)\\
&\exists \lambda > 0, \gamma \geq 0 \textit{ such that } \forall \mathbf{x}_1, \mathbf{x}_2, u_1^i \in \mathcal{U}^G, u_2^j \in \mathcal{U}^G : && (5.8)\\
&-\frac{\partial S}{\partial \mathbf{x}_1} f_1(\mathbf{x}_1, u_1^i) - \frac{\partial S}{\partial \mathbf{x}_2} f_2(\mathbf{x}_2, u_2^j) - \lambda S(\mathbf{x}_1, \mathbf{x}_2) +\\
&\quad \gamma(u_1^i - u_2^j)^2 \in \mathbb{S}. && (5.9)
\end{aligned}$$

The CBF S starts at its maximum value at the pair of initial conditions $(\mathbf{x}_1^0, \mathbf{x}_2^0)$, and then decays along various trajectories of the two systems. Thus, along any pair of trajectories, the gap between $S(\mathbf{x}_1(t), \mathbf{x}_2(t))$ and the Squared Output Difference (SOD) $[g_1(\mathbf{x}_1(t)) - g_2(\mathbf{x}_2(t))]^2$ is maximum at $t = 0$, i.e. at the initial states. To improve the bound on the SOD given by S, we minimized $S(\mathbf{x}_1(0), \mathbf{x}_2(0))$ as the objective function of the SOSP.

Eq. (5.8) enforces the decay condition for a BF, given by Eq. (4.2), only on the samples (u_1^i, u_2^j) that comprise the grid $\mathcal{U}$. The validity of Eq. (4.2) on the entire input space can be verified using delta-decidability, as shown in Sec. 5.4. In Sec. 5.5.1, we present an alternative SOSP that enforces Eq. (4.2) on the entire input space.

5.3.2 Computing BFs using SOSP 2

In this section, we describe SOSP 2, an SOSP formulation that can be used to compute BFs. SOSP 2, in contrast to SOSP 1, which was reviewed in Section 5.3.1, exhaustively covers the input-space. First, we present the problem formulation and then we show that the solutions are indeed BFs.

We assume that the input spaces are described using sets, such as $\mathcal{U} = \{u \in \mathbb{R} : \rho(u) \geq 0\}$, where $\rho(u)$ is called a *descriptor function*. For example, $\rho(u) = (u - u_{min})(u_{max} - u)$ describes the input-space $\mathcal{U} = [u_{min}, u_{max}]$. We denote the components of the state vectors as $\mathbf{x}_1 = [x_{11}, x_{12}, \ldots, x_{1n_1}]$ and $\mathbf{x}_2 = [x_{21}, x_{22}, \ldots, x_{2n_2}]$. Each of these components take values in a closed interval, i.e. $x_{11} \in [\underline{x_{11}}, \overline{x_{11}}], \ldots, x_{1n_1}$
$\in [\underline{x_{1n_1}}, \overline{x_{1n_1}}]$ and $x_{21} \in [\underline{x_{21}}, \overline{x_{21}}], \ldots, x_{2n_2} \in [\underline{x_{2n_2}}, \overline{x_{2n_2}}]$. We introduce vectors of

polynomials τ_1 and τ_2 as descriptor functions of the state vectors:

$$\tau_i(\mathbf{x}_i) = \begin{bmatrix} (x_{i1} - \underline{x_{i1}})(\overline{x_{i1}} - x_{i1}) \\ \vdots \\ (x_{in_i} - \underline{x_{in_i}})(\overline{x_{in_i}} - x_{in_i}) \end{bmatrix}, i = 1, 2. \tag{5.10}$$

Definition 5.3.2. *Consider two dynamical systems $\Sigma_i = (X_i, \{\mathbf{x}_i^0\}, [u_{min}, u_{max}], f_i, \mathcal{O}, g_i)$, $i = 1, 2$. SOSP 2 is given by the following equations.*

$$\begin{aligned} &\textit{Minimize } S(\mathbf{x}_1^0, \mathbf{x}_2^0) && (5.11) \\ &\textit{subject to:} \\ &S(\mathbf{x}_1, \mathbf{x}_2) - [g_1(\mathbf{x}_1) - g_2(\mathbf{x}_2)]^2 \in \mathbb{S}, && (5.12) \\ &\forall u_i \in [u_{min}, u_{max}], x_{ij} \in [\underline{x_{ij}}, \overline{x_{ij}}], i = 1, 2, j = 1, \ldots, n_i, \\ &\exists \lambda > 0, \gamma \geq 0, \sigma_1(\mathbf{x}_1, u_1) \in \mathbb{S}, \sigma_2(\mathbf{x}_2, u_2) \in \mathbb{S}, \textit{and vectors of} \\ &\textit{SOS polynomials } \sigma_3(\mathbf{x}_1, u_1) \textit{ and } \sigma_4(\mathbf{x}_2, u_2) \textit{ such that:} \\ &-\frac{\partial S}{\partial \mathbf{x}_1} f_1(\mathbf{x}_1, u_1) - \frac{\partial S}{\partial \mathbf{x}_2} f_2(\mathbf{x}_2, u_2) - \lambda S(\mathbf{x}_1, \mathbf{x}_2) + \\ &\gamma(u_1 - u_2)^2 - \sigma_1(\mathbf{x}_1, u_1)\rho(u_1) - \sigma_2(\mathbf{x}_2, u_2)\rho(u_2) - \\ &\sigma_3(\mathbf{x}_1, u_1)\tau_1(\mathbf{x}_1) - \sigma_4(\mathbf{x}_2, u_2)\tau_2(\mathbf{x}_2) \in \mathbb{S}. && (5.13) \end{aligned}$$

Next, we show that the feasible solutions of SOSP 2 are indeed BFs for the systems.

Theorem 5.3.3. *Consider a feasible solution, $(S, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \lambda, \gamma)$, of the SOSP 2. S satisfies Eqs. (4.1) and (4.2), and thus is a BF between Σ_1 and Σ_2.*

Proof. S, from a feasible solution $(S, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \lambda, \gamma)$, satisfies Eq. (5.12):

$$\forall \mathbf{x}_1, \mathbf{x}_2 : S(\mathbf{x}_1, \mathbf{x}_2) - [g_1(\mathbf{x}_1) - g_2(\mathbf{x}_2)]^2 \in \mathbb{S}.$$

As an SOS polynomial is always non-negative, we get

$$S(\mathbf{x}_1, \mathbf{x}_2) - [g_1(\mathbf{x}_1) - g_2(\mathbf{x}_2)]^2 \geq 0,$$

which implies $[g_1(\mathbf{x}_1) - g_2(\mathbf{x}_2)]^2 \leq S(\mathbf{x}_1, \mathbf{x}_2)$, S satisfies Eq. (4.1).

A feasible solution satisfies Eq. (5.13):

$$\begin{aligned} &-\frac{\partial S}{\partial \mathbf{x}_1} f_1(\mathbf{x}_1, u_1) - \frac{\partial S}{\partial \mathbf{x}_2} f_2(\mathbf{x}_2, u_2) - \lambda S(\mathbf{x}_1, \mathbf{x}_2) + \gamma(u_1 - u_2)^2 - \sigma_1(\mathbf{x}_1, u_1)\rho(u_1) \\ &- \sigma_2(\mathbf{x}_2, u_2)\rho(u_2) - \sigma_3(\mathbf{x}_1, u_1)\tau_1(\mathbf{x}_1) - \sigma_4(\mathbf{x}_2, u_2)\tau_2(\mathbf{x}_2) \in \mathbb{S}. \end{aligned}$$

Non-negativity of SOS polynomials leads to

$$\begin{aligned}&-\frac{\partial S}{\partial \mathbf{x}_1} f_1(\mathbf{x}_1, u_1) - \frac{\partial S}{\partial \mathbf{x}_2} f_2(\mathbf{x}_2, u_2) - \sigma_1(\mathbf{x}_1, u_1)\rho(u_1) - \sigma_2(\mathbf{x}_2, u_2)\rho(u_2)\\ &- \sigma_3(\mathbf{x}_1, u_1)\tau_1(\mathbf{x}_1) - \sigma_4(\mathbf{x}_2, u_2)\tau_2(\mathbf{x}_2) \geq \lambda S(\mathbf{x}_1, \mathbf{x}_2) - \gamma(u_1 - u_2)^2.\end{aligned}$$

Multiplying both sides by -1 and then reversing the inequality, we get

$$\begin{aligned}&\frac{\partial S}{\partial \mathbf{x}_1} f_1(\mathbf{x}_1, u_1) + \frac{\partial S}{\partial \mathbf{x}_2} f_2(\mathbf{x}_2, u_2) + \sigma_1(\mathbf{x}_1, u_1)\rho(u_1) + \sigma_2(\mathbf{x}_2, u_2)\rho(u_2)\\ &+ \sigma_3(\mathbf{x}_1, u_1)\tau_1(\mathbf{x}_1) + \sigma_4(\mathbf{x}_2, u_2)\tau_2(\mathbf{x}_2) \leq -\lambda S(\mathbf{x}_1, \mathbf{x}_2) + \gamma(u_1 - u_2)^2.\end{aligned}$$

As $\sigma_1(\mathbf{x}_1, u_1)\rho(u_1) + \sigma_2(\mathbf{x}_2, u_2)\rho(u_2) + \sigma_3(\mathbf{x}_1, u_1)\tau_1(\mathbf{x}_1) + \sigma_4(\mathbf{x}_2, u_2)\tau_2(\mathbf{x}_2)$ is always non-negative, we can eliminate the sum and still retain the inequality to get

$$\frac{\partial S}{\partial \mathbf{x}_1} f_1(\mathbf{x}_1, u_1) + \frac{\partial S}{\partial \mathbf{x}_2} f_2(\mathbf{x}_2, u_2) \leq -\lambda S(\mathbf{x}_1, \mathbf{x}_2) + \gamma(u_1 - u_2)^2.$$

□

5.4 Validating SOSP 1 CBFs using δ-Decidability

Consider two dynamical systems $(X_i, \{\mathbf{x}_i^0\}, [u_{min}, u_{max}], f_i, O, g_i)$, $i = 1, 2$. Let S, parameterized by λ and γ, be a CBF, which can be obtained by solving SOSP 1, see Def. 5.3.1. A valid solution of SOSP 1 satisfies Eq. (4.2) over the input grid $\mathcal{U}^G$ that is used in SOSP 1. The focus of this section is to validate if S satisfies Eq. (4.2) over all the inputs, and thus is a BF for the two systems. For this purpose, we use dReal [28], which implements δ-decidability, to validate S. Consider the function ψ:

$$\psi(\mathbf{x}_1, \mathbf{x}_2, u_1, u_2) \triangleq -\frac{\partial S}{\partial \mathbf{x}_1} f_1(\mathbf{x}_1, u_1) - \frac{\partial S}{\partial \mathbf{x}_2} f_2(\mathbf{x}_2, u_2) - \lambda S(\mathbf{x}_1, \mathbf{x}_2) + \gamma(u_1 - u_2)^2.$$

If S satisfies Eq. (4.2) over the entire state and input space, then the following SMT formula must be unsatisfiable:

$$\exists \mathbf{x}_1, \mathbf{x}_2, u_1, u_2 : \psi(\mathbf{x}_1, \mathbf{x}_2, u_1, u_2) < 0. \tag{5.14}$$

Delta-decidability, which involves relaxing ψ by a parameter $\delta > 0$, can be used to check if Eq. (5.14) is indeed unsatisfiable. The open-source tool *dReal* implements δ-decision procedures and can be used for our problem. A decision procedure is said to be δ-complete if for any SMT formula, it returns either *unsat*, if the formula is unsatisfiable, or returns *δ-sat*, if the formula's δ-relaxation is satisfiable, see [28, 25, 26] for a formal definition.

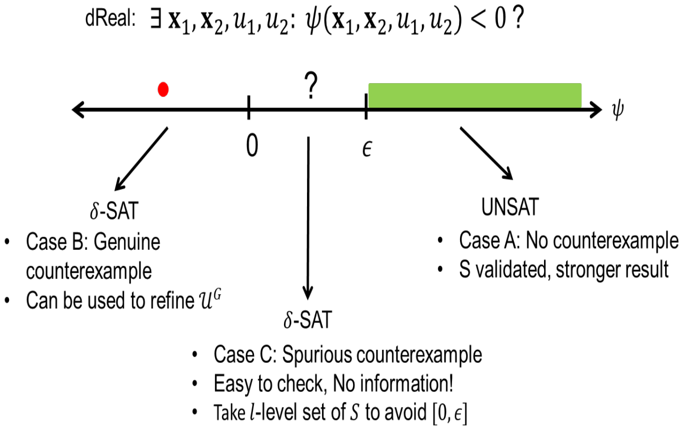

dReal: ∃ x1, x2, u1, u2: ψ(x1, x2, u1, u2) < 0 ?
?
ψ
0
ϵ
δ-SAT
• Case B: Genuine counterexample
• Can be used to refine 𝒰G
UNSAT
• Case A: No counterexample
• S validated, stronger result
δ-SAT
• Case C: Spurious counterexample
• Easy to check, No information!
• Take l-level set of S to avoid [0, ϵ]

Figure 5.5: Validating SOSP 1-based CBFs using dReal.

When Eq. (5.14) is presented to dReal, with a pre-determined δ, three possibilities, which are illustrated in Fig. 5.5, arise. We discuss each of them next.

Case A, dReal returns unsat: Eq. (5.14) is unsatisfiable, and therefore S is a valid BF. Delta-relaxation ensures a stronger result, we may claim that $\psi \geq \epsilon$, where ϵ is a function of δ and other internal parameters of dReal.

Case B, dReal returns δ-sat, with a counterexample, where $\psi < 0$. The tuple of states and inputs that is returned as the counterexample contains an input pair, where S fails to satisfy Eq. (4.2). The input pair is then used to refine $\mathcal{U}^G$ and SOSP 1 is repeated; Eq. (5.8), which represents a family of inequalities, is instantiated over $\mathcal{U}^G$, as well as the input pair from the counterexample.

Case C, dReal returns δ-sat, with a counterexample, where $\psi \geq 0$. This possibility arises due to the delta-relaxation. The formula ψ evaluates to a very small value in the range $[0, \epsilon]$. The counterexample is spurious and does not provide any information about S. Thus, we present a workaround to avoid this case.

To avoid Case C, we filter the domain of the state variables $\mathbf{x}_1$ and $\mathbf{x}_2$ in Eq. (5.14) to eliminate the pairs of states that contribute to the spurious counterexamples. Specifically, we validate the CBF S only on, and outside its l-level set. Eq. (5.14) is modified as

$$\exists \mathbf{x}_1, \mathbf{x}_2, u_1, u_2 : (S(\mathbf{x}_1, \mathbf{x}_2) \geq l) \wedge (\psi(\mathbf{x}_1, \mathbf{x}_2, u_1, u_2) < 0). \qquad (5.15)$$

The intuition behind level-set-based filtering of the domain is as follows. The function ψ goes to 0 when the states and the inputs go to 0. Also, ψ bounds the derivative of S with respect to time. The time-derivative of S takes very low values near the origin of the state-space, as the the origin is an equilibrium for our systems. Fig. 5.8

illustrates this property. When the inputs to the subsystem are held constant, the derivative of a BF becomes very low as the trajectories decay to the origin, which is a stable equilibrium. Thus, the derivative of S with respect to time, and consequently ψ, will have relatively larger values outside the l-level set of S.

The parameter l may be tuned till we avoid Case C completely and cover as many states as possible. Starting from an aggressive small value of $l \geq 0$, it may be incremented in small steps till Case C is completely avoided.

Our level-set-based approach can be justified as follows. We define the exterior of the l-level set of S as: $S^{\geq l} \triangleq \{(\mathbf{x}_1, \mathbf{x}_2) | S(\mathbf{x}_1, \mathbf{x}_2) \geq 0\}$. Validating Eq. (5.14) over $S^{\geq l}$ ensures that Eq. (4.1) and Eq. (4.2) are satisfied for all states within $S^{\geq l}$. In a practical setting, where we want to establish IOS between two systems, the sets of initial conditions become important. Given the decaying nature of a BF, the maximum value of the BF over a given pairing of the initial states is the best bound on the SOD that the BF can provide. Approximate bisimilarity of two systems can be established by minimizing the maximum value of the BF over all pairings of the initial states. For a given CBF, if this value is greater than the level set l, at which the CBF is validated, then the CBF can be used to be provide practical bounds on the SOD.

CBFs validated using the level-set-based approach also enable compositionality arguments, albeit in a weaker setting. To this end, we state the following proposition.

Proposition 5.4.1. *Let $\Sigma_i = (\mathcal{X}_i, \mathcal{X}_i^0, \mathcal{U}_i, f_i, \mathcal{O}_i, g_i)$, $i = 1, 2, A, B$, be dynamical systems such that $\mathcal{U}_1 = \mathcal{O}_A$, $\mathcal{U}_A = \mathcal{O}_1$, $\mathcal{U}_2 = \mathcal{O}_B$ and $\mathcal{U}_B = \mathcal{O}_2$. Let S_{12}, parameterized by λ_{12} and γ_{12}, be a BF between Σ_1 and Σ_2 in $S_{12}^{\geq l_1}$. Let S_{AB}, parameterized by λ_{AB} and γ_{AB}, be a BF between Σ_A and Σ_B in $S_{AB}^{\geq l_2}$.*

Let $\Sigma_{A1} = \Sigma_A || \Sigma_1$ and $\Sigma_{B2} = \Sigma_B || \Sigma_2$. If the small gain condition *(SGC) $\frac{\gamma_{AB}\gamma_{12}}{\lambda_{AB}\lambda_{12}} < 1$ is met, then a BF S between Σ_{A1} and Σ_{B2}, which satisfies Eq. (4.1) and Eq. (4.2) over $S_{12}^{\geq l_1} \times S_{AB}^{\geq l_1}$, can be constructed as follows.*

$$S(\mathbf{x}_{A1}, \mathbf{x}_{B2}) = \alpha_1 S_{AB}(\mathbf{x}_A, \mathbf{x}_B) + \alpha_2 S_{12}(\mathbf{x}_1, \mathbf{x}_2)$$

where $\mathbf{x}_{A1} = [\mathbf{x}_A, \mathbf{x}_1]^T$ and $\mathbf{x}_{B2} = [\mathbf{x}_B, \mathbf{x}_2]^T$ and the constants α_1 and α_2 are as per Theorem 4.2.1.

Proof. Consider S be the function as per the theorem. We will find conditions on α_1 and α_2 such that S is a BF between Σ_{A1} and Σ_{B2} in $S_{12}^{\geq l_1} \times S_{AB}^{\geq l_1}$. If $\alpha_1 \geq 1$ and $\alpha_2 \geq 1$, then

$$\begin{aligned} S(\mathbf{x}_{A1}, \mathbf{x}_{B2}) &\geq S_{AB}(\mathbf{x}_A, \mathbf{x}_B) + S_{12}(\mathbf{x}_1, \mathbf{x}_2) \\ &\geq \| g_A(\mathbf{x}_A) - g_B(\mathbf{x}_B) \| + \| g_1(\mathbf{x}_1) - g_2(\mathbf{x}_2) \| \end{aligned}$$

because S_{AB} and S_{12} satisfy first conditon of BF. The observational difference of Σ_{A1} and Σ_{B2}:

$$\begin{aligned} &\parallel g_{A1}(\mathbf{x}_{A1}) - g_{B2}(\mathbf{x}_{B2}) \parallel = \\ &\sqrt{\parallel g_A(\mathbf{x}_A) - g_B(\mathbf{x}_B) \parallel^2 + \parallel g_1(\mathbf{x}_1) - g_2(\mathbf{x}_2) \parallel^2} \\ &\leq \parallel g_A(\mathbf{x}_A) - g_B(\mathbf{x}_B) \parallel + \parallel g_1(\mathbf{x}_1) - g_2(\mathbf{x}_2) \parallel \end{aligned}$$

Therefore, it shows that S satisfies first condition of BF too. As $(\mathbf{x}_1, \mathbf{x}_2) \in S_{12}^{\geq l_1}$ and $(\mathbf{x}_A, \mathbf{x}_B) \in S_{AB}^{\geq l_2}$, by applying similar steps as in [29], we can write the following:

$$\begin{aligned} &\frac{\partial S}{\partial \mathbf{x}_{A1}} f_{A1}(\mathbf{x}_{A1}, \mathbf{u}_{A1}) + \frac{\partial S}{\partial \mathbf{x}_{B2}} f_{B2}(\mathbf{x}_{B2}, \mathbf{u}_{B2}) \\ &\leq -(\alpha_1 \lambda_{AB} - \alpha_2 \gamma_{12}) S_{AB}(\mathbf{x}_A, \mathbf{x}_B) - (\alpha_2 \lambda_{12} - \alpha_1 \gamma_{AB}) S_{12}(\mathbf{x}_1, \mathbf{x}_2) \end{aligned}$$

If $(\alpha_1 \lambda_{AB} - \alpha_2 \gamma_{12}) > 0$ and $(\alpha_2 \lambda_{12} - \alpha_1 \gamma_{AB}) > 0$ and
$\lambda = \min(\frac{(\alpha_1 \lambda_{AB} - \alpha_2 \gamma_{12})}{\alpha_1}, \frac{(\alpha_2 \lambda_{12} - \alpha_1 \gamma_{AB})}{\alpha_2})$, then

$$\frac{\partial S}{\partial \mathbf{x}_{A1}} f_{A1}(\mathbf{x}_{A1}, \mathbf{u}_{A1}) + \frac{\partial S}{\partial \mathbf{x}_{B2}} f_{B2}(\mathbf{x}_{B2}, \mathbf{u}_{B2}) \leq -\lambda S(\mathbf{x}_{A1}, \mathbf{x}_{B2}).$$

Therefore, S will be a BF in the domain $S_{12}^{\geq l_1} \times S_{AB}^{\geq l_2}$ if $\alpha_1 \geq 1$, $\alpha_2 \geq 1$, $(\alpha_1 \lambda_{AB} - \alpha_2 \gamma_{12}) > 0$ and $(\alpha_2 \lambda_{12} - \alpha_1 \gamma_{AB}) > 0$. As shown in [29], these four conditions can be expressed as $\frac{\lambda_{AB} \lambda_{12}}{\gamma_{AB} \gamma_{12}} < 1$. □

5.5 Results

In this section, we elaborate on computing the BFs S_{KH}, S_C, and the composed BF S between Σ_{CK} and Σ_{CH} using ***BFComp***. The BFs computed using SOSP 1 and SOSP 2 are then visualized along pairs of trajectories obtained by feeding constant-input signals to the corresponding systems.

5.5.1 Computing S_{KH} and S_C using SOSP 2

Automated solvers, such as MATLAB SOSTOOLS [76], which can be used to solve SOSP 2, have the following restriction: only polynomial vector fields, denoted by $f_i(\mathbf{x}_i, u_i)$, $i = 1, 2$ in Eq. (5.13), can be specified. In other words, f_i must be a polynomial function of $\mathbf{x}_i$ and u_i.

The potassium-channel subsystem Σ_K does not satisfy the above-mentioned requirement. The dynamics, see Eq. (5.1), is specified by $\dot{\mathbf{x}} = A_K(V).\mathbf{x}$, where $\mathbf{x}$ is the occupancy-probability vector and $A_K(V)$ is the rate matrix, whose entries are *exponential functions of the input membrane potential* V, see Defn. 5.1.1. Thus, the dynamics of Σ_K are not polynomial in the input.

As a workaround, we transformed the rate matrix $A_K(V)$ to an approximately equivalent matrix $A^p_K(V)$ by fitting the entries of A with polynomial functions using MATLAB *cftool* [66]. The polynomial approximations of the voltage-dependent rate functions, denoted by the superscript p are as follows.

$$\begin{aligned}
\beta^p(V) &= -(4.322 \times 10^{-6})V + 0.216,\\
\delta^p(V) &= (2.125 \times 10^{-10})V^6 - (9.322 \times 10^{-9})V^5 + (8.964 \times 10^{-8})V^4\\
&\quad - (1.716 \times 10^{-6})V^3 + (8.87 \times 10^{-5})V^2 - 0.001284V + 0.006744,\\
\epsilon^p(V) &= (4.435 \times 10^{-9})V^4 + (5.191 \times 10^{-7})V^3 + (2.539 \times 10^{-5})V^2\\
&\quad + (0.0006507)V + 0.007652, \text{and}\\
\omega^p(V) &= (3.771 \times 10^{-7})V - (5.415 \times 10^{-5})V + 0.0038.
\end{aligned}$$

See Sec. 5.5.2 for a detailed justification for the polynomial approximations.

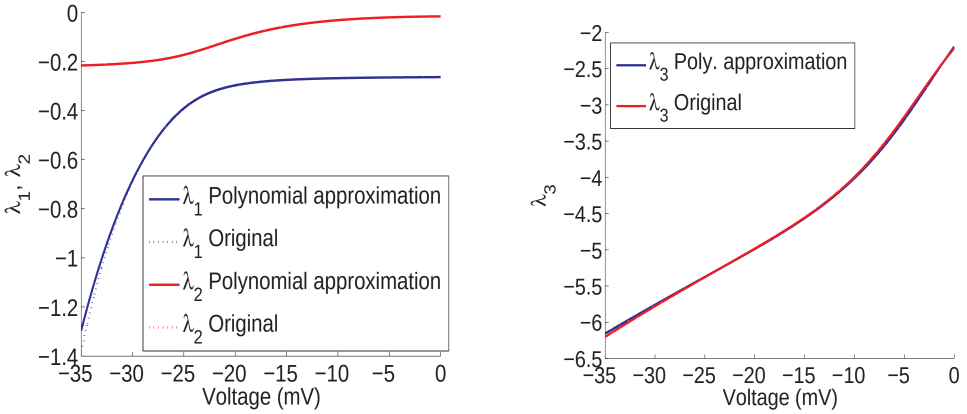
Computing S_{KH} and S_C using SOSP 2 begins with declaring the form of the BFs. We chose ellipsoidal forms using the `sossosvar` function provided by SOSTOOLS: $S_{KH}(\mathbf{x},\mathbf{y}) = [\mathbf{x},\mathbf{y}].Q_{KH}.[\mathbf{x},\mathbf{y}]^T$ and $S_C(V_K,V_H) = [V_K,V_H].Q_C.[V_K,V_H]^T$. Variables $\mathbf{x},\mathbf{y},V_K$, and V_H are declared using the `pvar` polynomial variable toolbox. The coefficients of the BFs, which form the decision variables of the SOSPs, are contained in the positive semidefinite matrices Q_{KH} (4×4) and Q_C (2×2). We chose ellipsoidal forms, using the `sossosvar`, for the $\sigma(,.,)$ functions that strengthen the decay requirement in Eq. (5.13) of Defn. 5.3.2. The descriptor functions were obtained from the definitions Σ_K, Σ_H and Σ_C.

SOSP2 was implemented in MATLAB R2013a, SOSTOOLS 2.04 [76] on an Intel Core i7-4770K 3.5 GHz CPU with 32 GB of memory. For S_{KH}, SOSTOOLS terminated in 5.95 seconds with the following flags: *feas ratio* $= 0.3147$, *pinf* $=$ *dinf* $= 0$, *numerr* $= 1$. For S_C, SOSTOOLS terminated in 5.95 seconds with the following flags: *feas ratio* $= 1.03$, *pinf* $=$ *dinf* $=$ *numerr* $= 0$.

5.5.2 Justification for Polynomial Approximations of Rate Functions

Polynomial approximations of the rate functions, which were described in Sec. 5.5.1, can be justified as follows. We show that the difference in the eigenvalues of $A_K(V)$ and $A^p_K(V)$, which controls the difference between the corresponding trajectories of the two systems, is negligible for $V \in [-35, 0]$. Moreover, the difference can be bounded after the polynomial curve fitting is completed. The resulting error between the trajectories can then be used to relax the bound on SOD provided by Theorem 4.1.1.

We begin with empirical evidence of the difference between the spectra of $A_K(V)$ and $A_K^p(V)$ being minimal. Fig. 5.6 plots the spectrum of the two matrices $A_K(V)$ and $A_K^p(V)$ as a function of V. Only three eigenvalues are plotted as one of the four state-variables of Σ_K can be eliminated, as they sum to 1, resulting in a 3×3 rate matrix.



(a) First and second eigenvalues: mean L1 errors are 0.005 and 4.45×10^{-4}.

(b) Third eigenvalue, scale of y-axis is 10^{-3}. Mean L1 error is 1.68×10^{-5}.

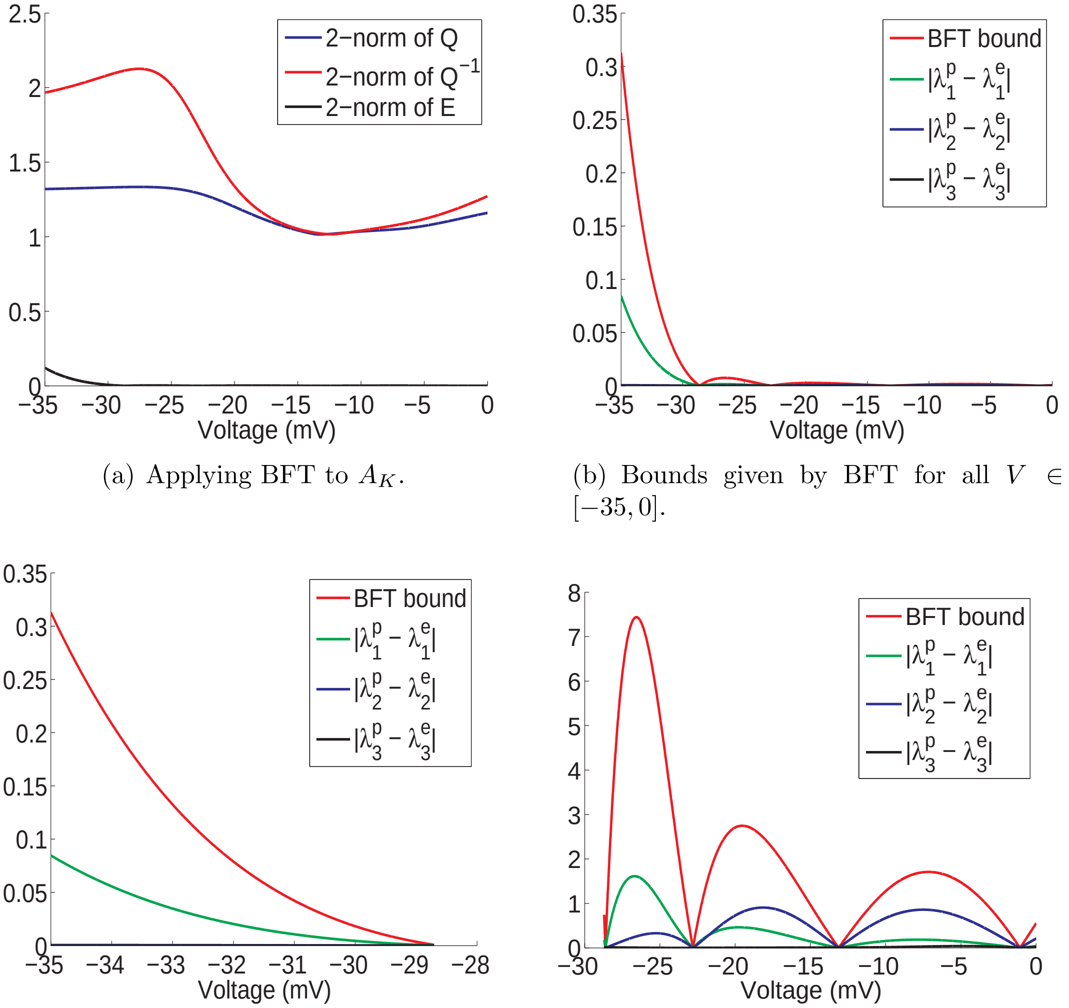
Figure 5.6: Spectra of $A_K(V)$ and $A_K^p(V)$.

Next, we show that the difference between the eigenvalues can be bounded after the fitting process. Note that, in general, Weierstrass Approximation theorem [48] allows us to find polynomial approximations of the continuous exponential functions, like $\beta(V)$, $\delta(V)$, $\epsilon(V)$, and $\omega(V)$, to any degree of accuracy for $V \in [-35, 0]$. Once the polynomial approximations have been identified, the Bauer-Fike Theorem (BFT) [6] can be used to bound the corresponding error in the eigenvalues.

BFT relates the eigenvalues of an $n \times n$ diagonalizable matrix A, where $A = QDQ^{-1}$, to the matrix $A + E$, where E is an $n \times n$ perturbation. Every eigenvalue μ of the matrix $A + E$ satisfies the following inequality: $|\mu - \lambda| \leq ||Q||.||Q^{-1}||.||E||$, where λ is some eigenvalue of A and $||.||$ denotes the 2-norm.

In our case, the original rate matrix A_K is perturbed during the polynomial approximation to A_K^p. As mentioned above, this perturbation can be minimized arbitrarily. BFT can be applied to bound the difference between the eigenvalues after the exact value of the perturbation is determined after the fitting process.

Fig. 5.7 illustrates the BFT on $A_K(V)$ and $A_K^p(V)$. Fig. 5.7(a) plots the three components that contribute to the upper bound: the norms of the eigenvector matrix



(a) Applying BFT to A_K.

(b) Bounds given by BFT for all $V \in [-35, 0]$.

(c) Bounds given by BFT for all $V \in [-35, 28.5]$.

(d) Bounds given by BFT for all $V \in [-28.5, 0]$.

Figure 5.7: Bounding the difference between the spectra of $A_K(V)$ and $A_K^p(V)$ using BFT. The superscript p is used to denote the eigenvalue of $A_K^p(V)$.

and its inverse, and the perturbation matrix, which can be arbitrarily minimized. Fig. 5.7(b) illustrates the bound obtained by multiplying the aforementioned components. Fig. 5.7(c) and Fig. 5.7(d) show the bounds for two partitions of the voltage range for better clarity. We observe that BFT is not overly conservative and can be used to obtain reasonable bounds between the original rate matrix $A_K(V)$ and its polynomial version $A_K^p(V)$.

5.5.3 Computing S_{KH} and S_C using SOSP 1 and dReal

The details of implementing SOSP 1 in MATLAB SOSTOOLS can be found in Sec. 3 of [63]. We provide details on dReal-based validation of the CBFs.

For S_{KH}, $\mathcal{V} = [-35, -25, -15, -5, 0]$. $\mathcal{V} \times \mathcal{V}$ was used as the input grid to compute the CBF S_{KH} using SOSP 1. S_{KH} was parameterized by $\lambda_{KH} = 0.001$ and $\gamma_{KH} = 0.0001$. The CBF was validated as per Sec. 5.4; Eq. (5.15) was proved to be *unsat* in dReal by choosing $l = 0.001$.

For S_C, we considered $\mathcal{O} \times \mathcal{O}$ as the input grid, where $\mathcal{O} = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$. S_C was parameterized by $\lambda_C = 0.001$ and $\gamma_C = 0.0001$. The CBF was validated as per Sec. 5.4; Eq. (5.15) was proved to be *unsat* in dReal by choosing $l = 1$.

The validation was implemented using dReal's version 2.14.08-linux [28] on an Intel Core i7-4770K 3.5 GHz CPU with 32 GB of memory. The running time was 416 minutes and 58.64 seconds for S_{KH} and 7 seconds for S_C.

5.5.4 Composing S_{KH} and S_C using the Small-Gain Theorem

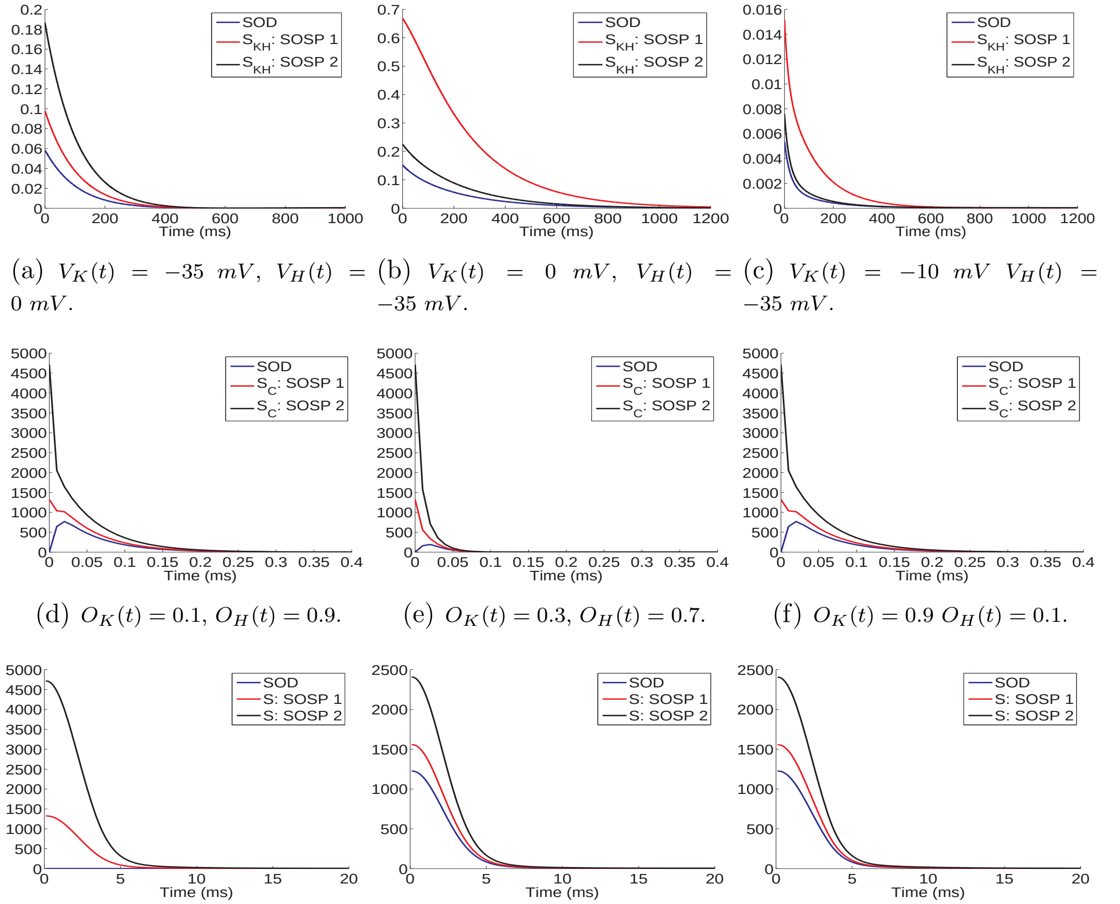
The parameters of S_{KH} and S_C satisfy the SGC condition of Theorem 4.2.1, as $\frac{\gamma_{KH}\gamma_C}{\lambda_{KH}\lambda_C} = 0.01 < 1$ in both SOSP 1 and SOSP 2. Applying Theorem 4.2.1, we linearly composed S_{KH} and S_C to obtain $S = \alpha_1 S_{IH} + \alpha_2 S_C$, where $\alpha_1, \alpha_2 = 1$. S is a BF between the composite systems Σ_{CK} and Σ_{CH}. As per Theorem 2 of [29], the parameter λ of S is given by

$$\lambda = min\left(\frac{\alpha_1\lambda_{KH} - \alpha_2\gamma_C}{\alpha_1}, \frac{\alpha_2\lambda_C - \alpha_1\gamma_{KH}}{\alpha_2}\right) = 0.0009.$$

5.5.5 Visualizing the BFs

Empirical validation of the BFs is provided by plotting them in 2D along the time axis. As the time proceeds in the same manner in both systems, the corresponding BF is plotted for the pair of states occurring at the same time along the trajectories of the systems. The SOD observed for the pair of states is also plotted in the same graph. The resulting plots show that the BFs bound the SOD and decay in time along the pairs of trajectories, as per Theorem 4.1.1.

Figs. 5.8 (a) - (c) show S_{KH} plotted along three pairs of trajectories of Σ_K and Σ_H. Each pair was generated by supplying a pair of constant voltage signals $(V_1(t), V_2(t))$ as inputs to Σ_K and Σ_H, respectively. The two subsystems were initialized as per Defs. 5.1.1 and 5.1.2, and simulated using MATLAB's *ODE45* solver.



(a) $V_K(t) = -35\ mV,\ V_H(t) = 0\ mV.$ (b) $V_K(t) = 0\ mV,\ V_H(t) = -35\ mV.$ (c) $V_K(t) = -10\ mV\ V_H(t) = -35\ mV.$

(d) $O_K(t) = 0.1$, $O_H(t) = 0.9$. (e) $O_K(t) = 0.3$, $O_H(t) = 0.7$. (f) $O_K(t) = 0.9\ O_H(t) = 0.1$.

(g) $V_K(0) = 0\ mV, V_H(0) = 0\ mV.$ (h) $V_K(0) = 0\ mV,\ V_H(0) = -35\ mV.$ (i) $V_K(0) = -35\ mV\ V_H(0) = 0\ mV.$

Figure 5.8: BFs S_{KH}, S_C, S, and their corresponding SOD plotted along trajectories of the respective systems. In subfigures (a) - (c), S_{KH} and SOD are plotted along three pairs of trajectories of Σ_K and Σ_H generated using constant voltage (input) signals. In subfigures (d) - (f), S_C and SOD are plotted along three pairs of trajectories of Σ_C generated using constant conductance (input) signals. In subfigures (g) - (h), the composed BF S and SOD are plotted along three pairs of trajectories of Σ_{CK} and Σ_{CH} generated using different initial conditions. In all three cases, the BFs upper bound the SOD and decay along the trajectories.

S_{KH} was then evaluated along the resulting pair of trajectories after shifting the origin to the equilibrium defined by $(V_1(t), V_2(t))$. In two cases, S_{KH} computed using SOSP 2 provides slightly better error bound than that of using SOSP 1.

S_C characterizes the ability of Σ_C to tolerate small changes in the input signals. In the composite systems Σ_{CK} and Σ_{CH}, these signals are provided by subsystems

Σ_K and Σ_H, and thus vary slightly due to the fitting errors incurred by the model-order reduction as described in Sec.5.1.2.

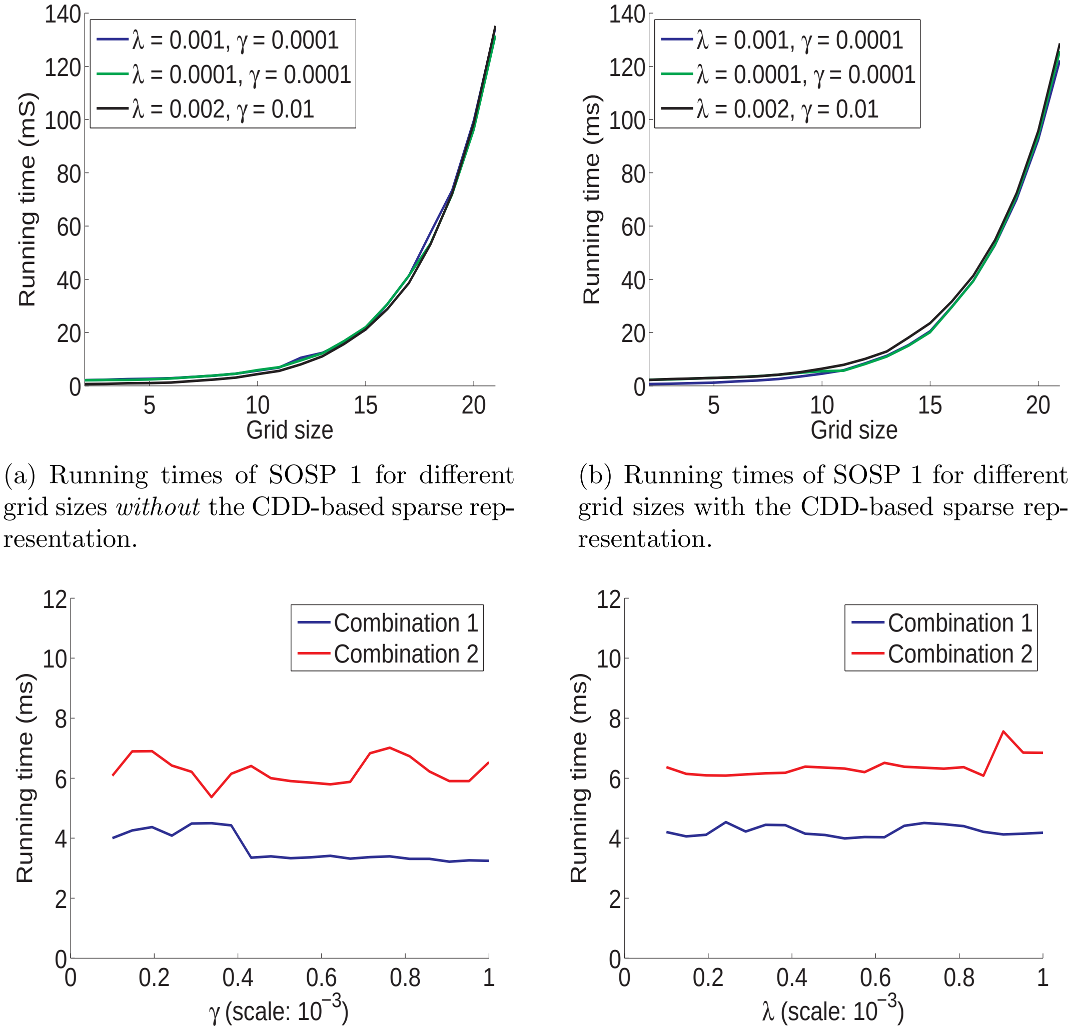
S_C is plotted in Figs. 5.8 (d) - (f) along three pairs of trajectories of Σ_C. Each pair of trajectories was generated by supplying constant conductance (input) signals $(O_1(t), O_2(t))$. Σ_C was initialized at 0 mV and simulated using the Euler method. S_C was evaluated along the resulting trajectories after shifting the origin to the equilibrium, -35 mV (E_K). We observed that S_C computed using SOSP 1 gives a tighter SOD bound compared to SOSP 2.

CCMs Σ_{CK} and Σ_{CH} are autonomous dynamical systems and do not receive any external inputs. To visualize the composite BF S, we simulated Σ_{CK} and Σ_{CH} using the Euler method for different initial conditions. Fig. 5.4 plots the trajectories obtained from these simulations. The corresponding conductance traces of Fig. 5.4(a) and the voltage traces of Fig. 5.4(b) empirically validate that the composed models are approximately equivalent as predicted by Theorem 4.2.1. BF S along this pair, and two other pairs of trajectories is plotted in Fig. 5.8 (g) - (i). The value of S is dominated by the value of S_C, as it bounds the squared difference of voltages and is much larger than S_{KH}, which bounds differences in probabilities. This is reasonable as voltage is the primary entity of interest when analyzing excitable cells. One could scale subsystem Σ_C such that its output lies in $[0, 1]$ and is thus comparable to the outputs of Σ_K and Σ_H. In all three cases, S computed using SOSP 1 performs much better than the one computed using SOSP 2.

5.6 Performance Evaluation of ***BFComp***

In this section, we present performance evaluation results for ***BFComp***. The parameters of SOSP 1 and SOSP 2 were varied while computing the BF S_{KH} using SOSTOOLS. The validation of the SOSP-1-based CBF in dReal was also analyzed by varying the level set parameter and δ. The running time for the SOSTOOLS and dReal-based implementations are presented to illustrate the tradeoffs posed by the different parameters. We begin with the performance of SOSTOOLS-based implementations SOSP 1 and SOSP 2, which is illustrated in Fig. 5.9.

SOSP 1 uses $\mathcal{U}^G$, the input-space grid; see Def. 5.3.1. The size of the grid corresponds to the granularity with which Eq. (4.2) is enforced across the input space. Each input pair on the grid corresponds to one constraint in the SOSP. Therefore, a relatively denser grid leads to a larger instance of SOSP 1, which takes a longer duration of time to be solved in SOSTOOLS. On the other hand, a denser grid ensures that Eq. (4.2) is satisfied on relatively more points across the input space, which makes the validation step converge faster.



(a) Running times of SOSP 1 for different grid sizes *without* the CDD-based sparse representation.

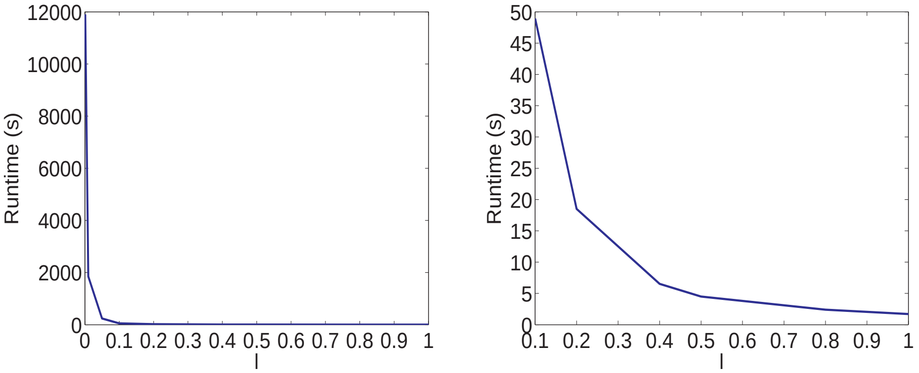
(b) Running times of SOSP 1 for different grid sizes with the CDD-based sparse representation.

(c) Running times of SOSP 2 for different values of γ and the two combinations of the $\sigma(.,.)$ functions.

(d) Running times of SOSP 2 for different values of λ and the two combinations of the $\sigma(.,.)$ functions.

Figure 5.9: Performance evaluation of SOSTOOLS-based implementations of SOSP 1 and SOSP 2.

Fig. 5.9 (a)-(b) reflects this behavior. SOSP 1 was used to compute S_{KH} using different grid sizes. Three different values of λ and γ were also used. The running times increase exponentially with the grid size. The values of λ and γ do not affect this trend. Moreover, SOSTOOLS offers the option of using the CDD package to exploit the sparsity of multivariate polynomials; see Sec. 2.4.3 of [76]. This option can significantly reduce the running time for large SOSPs, but the computation of S_{KH} using SOSP 1 is not affected by this option.



(a) Running time for dReal: l varies from 0.001 to 1.

(b) Running time for dReal: l varies from 0.1 to 1.

Figure 5.10: Running times for dReal across different values of l.

Performance of SOSP 2 was analyzed for different values of λ and γ. The form of the $\sigma(,.,)$ functions is an important parameter in SOSP 2. We chose an $\mathbf{x}^T Q\mathbf{x}$ form for these functions using `sossosvar` provided by SOSTOOLS; see Sec. 2.3.4 of [76]. Two combinations of $\sigma(,.,)$ functions were tested. In *Combination 1*, i) σ_1 and σ_2 were quartic in V_K and V_H, ii) σ_3 was of degree 8 in V_K, and iv) σ_4 was quadratic in V_H. In *Combination 2*, i) σ_1 and σ_2 were of degree 8 in V_K and V_H, ii) σ_3 was of degree 12 in V_K, and iv) σ_4 was quartic in V_H.

The relatively higher degree of the polynomials in Combination 2 results in a larger instance of SOSP 2. Consequently, SOSTOOLS-based implementations take longer to compute S_{KH}, as illustrated in Fig. 5.9(c) - (d). The figures also show that the exact value of λ and γ do not affect the running times. CDD-based sparse representations were used for both generating both these plots.

Next, we focus on the validation of SOSP-1-based CBFs in dReal. The running time required to prove the formula in Eq. 5.15 in dReal depends on level-set parameter l and the δ. A detailed worst case runtime analysis of dReal can be found in [27]. Here, we report empirical run-time statistics for validating the CBF S_{KH} using Eq. 5.15 for different values of l and δ.

The parameter l defines the domain over which Eq. 5.15 is validated. A smaller value of l corresponds to the formula being validated over a relatively larger portion of the state-space. Consequently, dReal requires more running time. Fig. 5.10 illustrates the dependence of dReal's runnning time for different values of l, while keeping δ to a fixed value 1×10^{-3}. For clarity, Fig. 5.10(b) plots the running time for $l \in p[0.1, 1]$. The running time increases exponentially as l decreases. Note that the CBF S_{KH} was successfully validated over all the data points.

The performance of dReal was also analyzed by varying δ. The parameter was varied from 1×10^{-2} to 1×10^{-8}, while keeping l fixed at 5×10^{-2}. The running time varied non-deterministically between 130 seconds and 190 seconds.

Chapter 6

Model Checkign Tap Withdrawal in C. Elegans

Although neurology and brain modeling/simulation is a popular field of biological study, formal verification has yet to take root. There has been cursory study into neurological model checking (see chapter 7), but not with the nonlinear ODE models used by biologists. The application of verification technology to hardware circuits has played a key role in the *Electronic Design Automation* (EDA) industry; perhaps it will play a similar role with neural circuits.

For our initial neurological study, we have selected the round worm, *Caenorhabditis Elegans*, due to the simplicity of its nervous system (302 neurons, ∼5,000 synapses) and the breadth of research on the animal. The complete connectome of the worm is documented, and there have been a number of interesting experiments on its response to stimuli.

For model-checking purposes, we were particularly interested in the *tap withdrawal* (TW) neural circuit. The TW circuit governs the reactionary motion of the animal when the petri dish in which it swims is perturbed. (A related circuit, *touch sensitivity*, controls the reaction of the worm when a stimulus is applied to a single point on the body.) Studies of the TW circuit have traditionally involved using lasers to ablate the different neurons in the circuit of multiple animals and measuring the results when stimuli are applied.

A model of the TW circuit was presented by Wicks, Roehrig, and Rankin in [92]. Their model is in the form of a system of nonlinear ODEs with an indication of polarity (inhibitory or excitatory) of each neuron in the TW circuit. Additionally, Wicks and Rankin had a previous paper in which they measure the three possible reactions of the animals to TW with various neurons ablated [91]; see also Fig. 2.5. The three behaviors—acceleration, reversal of movement, and no response—are logged with the percentage of the experimental population to display that behavior.

The [92] model has a number of circuit parameters, such as gap-junction conductance, capacitance, and leakage current, that crucially affect the behavior of the organism. A single value for each parameter is given in [92]. With this single set of parameter values, the model produces predominant behavior in most ablation groups with a few exceptions.

While the experimental work in [91, 92] and the model presented in [92] were by no means insubstantial, the exploration of the model is vastly incomplete. The fixed parameter values fit through experimentation cause the model to replicate the predominant behavior seen in said experiments, but little can be said about the model beyond that. The ranges that can produce the predominant behavior, as well as the two other behaviors, are completely missing. This is not to fault the authors of [92], however, as the technology needed to uncover these ranges simply did not exist at the time.

The missing technology was the ability to automatically generate local discrepancy functions [15], and has only recently been developed [19]. With this technique, we can theoretically compute reach tubes used in verification. In reality, this is not a simple plug-and-play situation. To make use of [19], we needed to create the verification framework in Fig. 6.1. Through careful model engineering (Fig. 6.1 (1-3)) and verification engineering (Fig. 6.1 (4-6)) we were able to explore and verify the full parameter ranges in the Wicks et al. model to produce all three behaviors in the TW circuit. Such an understanding of the model is critical to morphospace exploration [88] of the animal. A detailed description of our framework and its application to the [92] model (Fig. 6.1(b)) is given in Sec. 6.1.

This verification framework has the additional benefit of being model agnostic. It can be reused to verify other complex nonlinear ODE models.

The rest of the chapter develops as follows. Section 6.1 describes our reach-tube reachability analysis and associated property checking. Section 6.2 presents our extensive collection of model-checking/parameter-estimation results.

6.1 Reachability Analysis of Nonlinear TW Circuit

Reachability analysis for verifying properties for general nonlinear dynamical systems is a well-known hard problem. The verification framework introduced in Fig. 6.1 combines model and verification engineering to perform reachability analysis

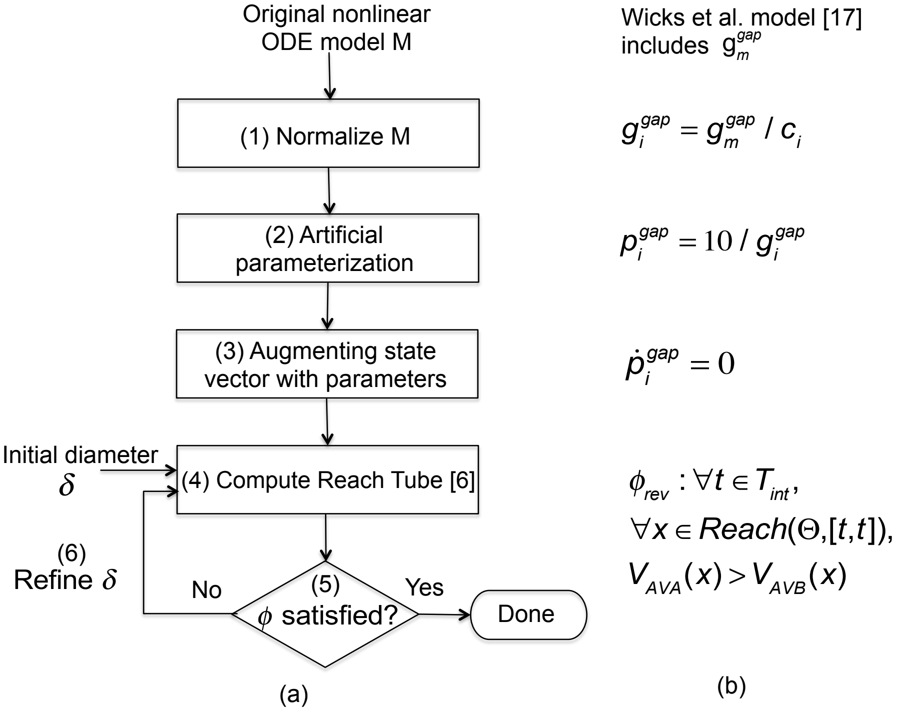

Original nonlinear ODE model M
(1) Normalize M
(2) Artificial parameterization
(3) Augmenting state vector with parameters
Initial diameter δ
(4) Compute Reach Tube [6]
(6) Refine δ
No
(5) ϕ satisfied?
Yes
Done
(a)
Wicks et al. model [17] includes g_m^{gap}
$g_i^{gap} = g_m^{gap} / c_i$
$p_i^{gap} = 10 / g_i^{gap}$
$\dot{p}_i^{gap} = 0$
$\phi_{rev} : \forall t \in T_{int}$,
$\forall x \in Reach(\Theta, [t,t])$,
$V_{AVA}(x) > V_{AVB}(x)$
(b)

Figure 6.1: Verification framework of nonlinear ODE model based on automatic computation of discrepancy function. (a) The general framework, (b) Application to [92] model.

on the Wicks et al. [92] model, discovering crucial parameter ranges to produce all three behaviors of the TW circuit. Our framework can be applied to any nonlinear ODE model.

6.1.1 Background on Reachability using Discrepancy

Consider an n-dimensional *autonomous dynamical system*:

$$\dot{x} = f(x), \tag{6.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is a Lipschitz continuous function. A *solution* or a *trajectory* of the system is a function $\xi : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ such that for any initial point $x_0 \in \mathbb{R}^n$ and at any time $t > 0$, $\xi(x_0, t)$ satisfies the differential equation (6.1). A state x in $\mathbb{R}^n$ is *reachable from the initial set* $\Theta \subseteq \mathbb{R}^n$ *within a time interval* $[t_1, t_2]$ if there exists an initial state $x_0 \in \Theta$ and a time $t \in [t_1, t_2]$ such that $x = \xi(x_0, t)$. The set of all reachable states in the interval $[t_1, t_2]$ is denoted by $\mathsf{Reach}(\Theta, [t_1, t_2])$. If $t_1 = 0$, we write $\mathsf{Reach}(t_2)$ when set Θ is clear from the context. If we can compute or approximate the reach set of such a model, then we can check for invariant or temporal properties of the model. Specifically, *C. Elegans* TW properties such as accelerated forward movement or reversal of movement fall into these categories. Our core reachability algorithm [15, 44, 16] uses a simulation engine that gives sampled numerical simulations of (6.1).

Definition 6.1.1. *A* (x_0, τ, ϵ, T)-simulation *of (6.1) is a sequence of time-stamped sets* (R_0, t_0), $(R_1, t_1) \ldots, (R_n, t_n)$ *satisfying:*

1. *Each* R_i *is a compact set in* $\mathbb{R}^n$ *with* $dia(R_i) \leq \epsilon$.
2. *The last time* $t_n = T$ *and for each* i, $0 < t_i - t_{i-1} \leq \tau$, *where the parameter* τ *is called the* sampling period.
3. *For each* t_i, *the trajectory from* x_0 *at* t_i *is in* R_i, *i.e.,* $\xi(x_0, t_i) \in R_i$, *and for any* $t \in [t_{i-1}, t_i]$, *the solution* $\xi(x_0, t) \in hull(R_{i-1}, R_i)$.

The algorithm for reachability analysis uses a key property of the model called a *discrepancy function.*

Definition 6.1.2. *A uniformly continuous function* $\beta : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ *is a* discrepancy function *of (6.1) if*

1. *for any pair of states* $x, x' \in \mathbb{R}^n$, *and any time* $t > 0$,
$$\|\xi(x,t) - \xi(x',t)\| \leq \beta(x, x', t), \text{ and} \tag{6.2}$$
2. *for any* t, *as* $x \to x'$, $\beta(.,.,t) \to 0$.

If a function β meets the two conditions for any pair of states x, x' in a compact set K then it is called a *K-local discrepancy function.* Uniform continuity means that $\forall \epsilon > 0, \forall x, x' \in K, \exists \delta$ such that for any time t, $\|x - x'\| < \delta \Rightarrow \beta(x, x', t) < \epsilon$. The verification results in [15, 44, 17, 16] required the user to provide the discrepancy function β as an additional input for the model. A Lipschitz constant of the dynamic function f gives an exponentially growing β, contraction metrics [59] can give tighter bounds for incrementally stable models, and sensitivity analysis gives tight bounds for linear systems [13], but none of these give an algorithm for computing β for general nonlinear models. Therefore, finding the discrepancy can be a barrier in the verification of large models like the TW circuit.

Here, we use Fan and Mitra's recently developed approach that automatically computes local discrepancy along individual trajectories [19]. Using the simulations and discrepancy, the reachability algorithm for checking properties proceeds as follows: Let the U be the set of states that violate the invariant in question.

First, a δ-cover $\mathcal{C}$ of the initial set Θ is computed; that is, the union of all the δ-balls around the points in $\mathcal{C}$ contain Θ. This δ is chosen to be large enough so that the cardinality of $\mathcal{C}$ is small. Then the algorithm iteratively and selectively refines $\mathcal{C}$ and computes more and more precise over-approximations of $\mathsf{Reach}(\Theta, T)$ as a union $\cup_{x_0 \in \mathcal{C}} \mathsf{Reach}(B_\delta(x_0), T)$. Here, $\mathsf{Reach}(B_\delta(x_0), T)$ is computed by first generating a (x_0, τ, ϵ, T)-simulation and then bloating it by a factor that maximizes $\beta(x, x', t)$ over $x, x' \in B_\delta(s_0)$ and $t \in [t_{i-1}, t_i]$. If $\mathsf{Reach}(B_\delta(x_0), T)$ is disjoint from U or is (partly) contained in U, then the algorithm decides that $B_\delta(x_0)$ satisfies and violates U, respectively. Otherwise, a finer cover of $B_\delta(x_0)$ is added to $\mathcal{C}$ and the iterative selective refinement continues. We refer to this in this paper as δ-refinement. In [15], it is shown that this algorithm is sound and relatively complete for proving bounded time invariants.

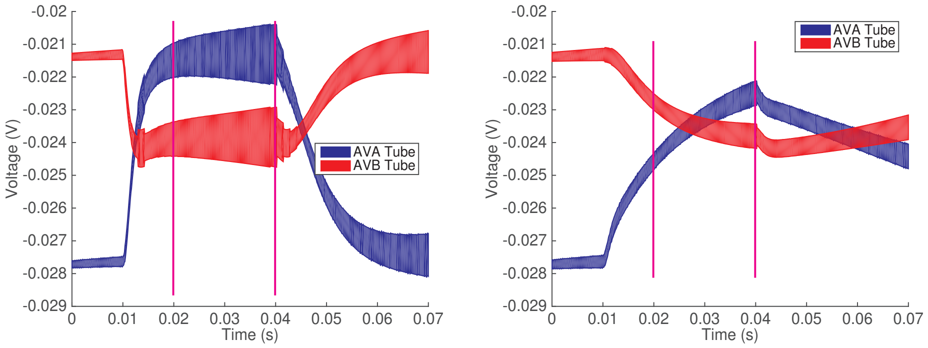
6.1.2 Applying Local Discrepancy to TW Circuit

Fan and Mitra's algorithm (see details in [19]) for automatically computing local discrepancy relies on the Lipschitz constant and the Jacobian of the dynamic function, along with simulations. The Lipschitz constant is used to construct a coarse, one-step over-approximation S of the reach set of the system along a simulation. Then the algorithm computes an upper bound on the maximum eigenvalue of the symmetric part of the Jacobian over S, using a theorem from matrix perturbation theory. This gives a piecewise exponential β, but the exponents are tight as they are obtained from the maximum eigenvalue of the linear approximation of the system in S. This means that for models with convergent trajectories, the exponent of β over S will be negative, and the $\mathsf{Reach}(T)$ approximation will quickly become very accurate. In the rest of this section, we describe key steps involved in making this approach work with the TW circuit.

The model of the TW circuit from Section **??** can be written as $\dot{V} = f(V)$, where $V \in \mathbb{R}^9$ has components V_i giving the membrane potential of neuron i. The Jacobian of the system is the matrix of partial derivatives with the ij^{th} term given by:

$$\begin{aligned}
\frac{\partial f_i}{\partial V_j} &= -g_i^{leak} - g_i^{gap} \sum_{j=1, j \neq i}^{N} n_{ij}^{gap} - g_i^{syn} \sum_{j=1, j \neq i}^{N} \frac{n_{ij}^{syn}}{1 + \exp(k \frac{V_j - V_j^{eq}}{V_{Range}})} \\
&= g_i^{gap} n_{ij}^{gap} - g_i^{syn} n_{ij}^{syn} \frac{\frac{k}{V_{Range}} \exp(k \frac{V_j - V_j^{eq}}{V_{Range}})(E_j - V_i)}{(1 + \exp(k \frac{V_j - V_j^{eq}}{V_{Range}}))^2} \qquad (6.3)
\end{aligned}$$

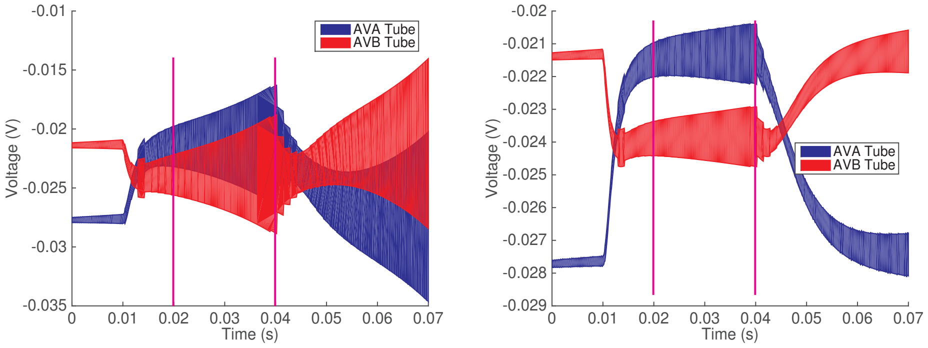
For parameter-range estimation of the TW circuit, each parameter p of interest is added as a new variable with constant dynamics ($\dot{p} = 0$). Computing the reach-set from initial values of p is then used to verify or falsify invariant properties for a continuous range of parameter values, and therefore a whole family of models, instead of analyzing just a single member of that family. Here the parameters of interest are the quantities $p_i^{leak} = 1/g_i^{leak}, p_i^{gap} = 10/g_i^{gap}, p_i^{syn} = 1/g_i^{syn}$.



(a) Rev. property satisfied with $g_{AVM}^{gap} = 1000$.

(b) Rev. unknown with $g_{AVM}^{gap} = 33.33$.

Figure 6.2: Model Checking Reversal Property of Control Group, with $\delta = 5{\times}10^{-5}$, varying g_{AVM}^{gap}.



(a) Rev. property unknown with $\delta = 1{\times}10^{-4}$.

(b) Rev. property satisfied with $\delta = 5{\times}10^{-5}$.

Figure 6.3: Model Checking Reversal Property of Control Group by refining δ.

Consider, for example, $1/g_i^{leak}$ as a parameter:

$$\begin{bmatrix} \dot{V} \\ 1/g_i^{leak} \end{bmatrix} = \begin{bmatrix} f(V) \\ 0 \end{bmatrix}.$$

In this case the Jacobian matrices for the system with parameters will be singular because of the all-zero rows that come from the parameter dynamics. The zero eigenvalues of these singular matrices are taken into account automatically by the algorithm for computing local discrepancy. In this paper we focus on p_i^{gap}, leaving the others for future work.

6.1.3 Checking Properties

Once the reach sets are computed, checking the *acceleration*, *reversal*, and *no-response* properties are conceptually straightforward. For instance, Equation (2.15) gives a method to check reversal movement. Instead of computing the integral of $(V_{AVA} - V_{AVB})$, we use the following sufficient condition to check it:

$$\phi_{rev} : \forall\ t \in T_{int}, \forall\ x \in \mathsf{Reach}(\Theta, [t,t]), V_{AVA}(x) > V_{AVB}(x).$$

Here, T_{int} is a specific time interval after the stimulation time, Θ is the initial set with parameter ranges, and recall that $\mathsf{Reach}(\Theta, [t,t])$ is the set of states reached at time t from Θ. We implement this check by scanning the entire reach-tube and checking that its projection on $V_{AVB}(x)$ is above that of $V_{AVA}(x)$ over all intervals. If this check succeeds (as in Figure 6.5(a)), we conclude that the range of parameter values produce the reversal movement. If the check fails, then the reversal movement is not provably satisfied (Figure 6.3(a)) and in that case we δ-refine the initial partition (Figure 6.3(b)). In some cases, such as Figure 6.5(b), δ-refinement can not prove the property satisfied or unsatisfied. This often occurs when two tubes intersect within the interval of interest. In this case, the property is considered to be unknown.

Fig. 6.4 helps paint a picture of how the δ-refinement process works with two parameters. We consider 4 refinement steps: $\delta = 7{\times}10^{-5}$, $\delta = 6{\times}10^{-5}$, $\delta = 5.5{\times}10^{-5}$, and $\delta = 5{\times}10^{-5}$. For $\delta = 7{\times}10^{-5}$, the property of interest is unknown at all points. With $\delta = 6{\times}10^{-5}$ the property is considered unknown for all red areas in the figure, including red and blue areas. Blue areas show where $\delta = 5.5 \times 10^{-5}$ are satisfied, and in the blue and yellow area both $\delta = 6 \times 10^{-5}$ and $\delta = 5.5 \times 10^{-5}$ have a satisfied property. The property is satisfied for the entire range of the graph when $\delta = 5 \times 10^{-5}$. Thus, the refinement process stops at $\delta = 5 \times 10^{-5}$, and the entire range of the parameter space is characterized.

6.2 Experimental Results

In this section, we apply our verification framework to the [92] model to estimate parameter ranges that produce three different behaviors (*reversal*, *acceleration*, *no response*) in the control and four ablation groups. We vary the gap-junction conductance of the sensory neurons (g_i^{gap}, $i \in \{AVM, ALM, PLM\}$) and keep all other parameters constant, as per [92]. Additionally, in the case of the no response behavior, we must lower the gap-junction conductance of the other neurons by a factor of 10^3. In Sec. 6.1, we explain that we use p_i^{gap} as our parameter in the state vector instead of g_i^{gap}, where $p_i^{gap} = 10/g_i^{gap}$. The parameter space we explore can be considered a bounding box, where each p_i^{gap} ranges over $[0.01, 1]$. As exploring the entire

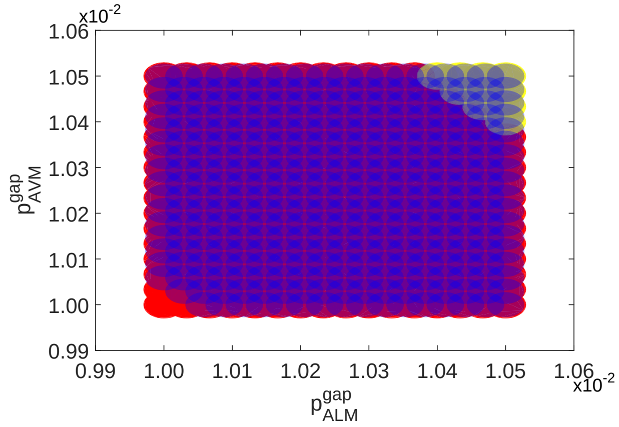


Figure 6.4: Example of 2-D Parameter Refinement. Red Regions are Unknown for both $\delta = 6{\times}10^{-5}$ and $\delta = 5.5{\times}10^{-5}$, Red/Blue Regions are Unknown for $\delta = 6{\times}10^{-5}$, but Satisfied for $\delta = 5.5 \times 10^{-5}$, and Yellow/Blue Regions are Satisfied for both.

parameter space is computationally intensive, we intelligently select a subspace to cover that lets us estimate contiguous ranges of parameters for each behavior. In Table 6.2.2, we present these ranges in terms of g_i^{gap}.

In the following subsections, we will present our results for parameter range estimation for all three behaviors of the control and ablation groups. This process requires three experiments per group.

6.2.1 1-D Parameter Space

Here we vary p^{gap}_{AVM} in all groups, except the AVM,ALM- group. By varying this parameter, we are able to produce reversal behavior in all four groups. We are also able to produce acceleration in all groups but PLM-. The PLM neuron drives acceleration in the TW circuit [91]. Hence, its absence in the PLM- group prevents acceleration from being produced, justifying the result.

For the AVM,ALM- group, we vary p^{gap}_{PLM} and produce acceleration and no response behaviors. As both AVM and ALM, responsible for reversal of movement, are ablated, reversal cannot be produced by this group.

6.2.2 2-D Parameter Space

In this set of experiments, we vary two parameters simultaneously. First we vary p^{gap}_{AVM} and p^{gap}_{ALM} for the control and PLM- groups. In both cases we produce reversal behavior. For the same reasons given in the previous subsection, we are unable to produce acceleration in the PLM- group and no response behavior in both these groups.

Next, we vary p^{gap}_{AVM} and p^{gap}_{PLM} for the ALM- and ALM,DVA- groups. We are able to produce both all three behaviors in both groups.

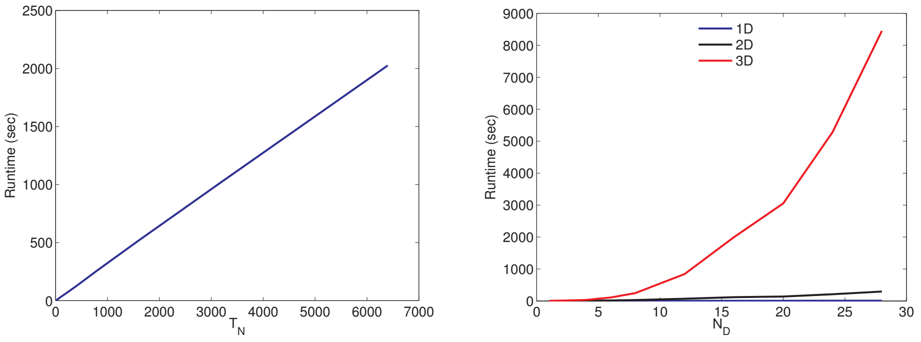
Group Name	Property	Parameters	Ranges	δ	Runtime (sec)
Control	REV	g^{gap}_{AVM}	$[46.2, 1000]$	1×10^{-6}	6324.4
	REV	g^{gap}_{AVM}, g^{gap}_{ALM}	$[952.38, 1000]^2$	2×10^{-5}	776.5
	REV	g^{gap}_{AVM}, g^{gap}_{ALM}, g^{gap}_{ALM}	$[990.01, 1000]^3$	2×10^{-5}	314.23
	ACC	g^{gap}_{AVM}	$[15.87, 10]$	1×10^{-5}	1110.01
	ACC	g^{gap}_{AVM}, g^{gap}_{ALM}	$[15.86, 15.87]^2$	2×10^{-5}	1619.8
	ACC	g^{gap}_{AVM}, g^{gap}_{ALM}, g^{gap}_{ALM}	$[15.85, 15.87]^3$	2×10^{-5}	320.12
	NR	g^{gap}_{AVM}	-	-	-
	NR	g^{gap}_{AVM}, g^{gap}_{ALM}	-	-	-
	NR	g^{gap}_{AVM}, g^{gap}_{ALM}, g^{gap}_{ALM}	$[10.005, 10]^3$	5×10^{-5}	124.23
PLM-	REV	g^{gap}_{AVM}	$[467.3, 1000]$	1×10^{-5}	718.08
	REV	g^{gap}_{AVM}, g^{gap}_{ALM}	$[952.38, 1000]^2$	2×10^{-5}	775.12
	ACC	g^{gap}_{AVM}	-	-	-
	ACC	g^{gap}_{AVM}, g^{gap}_{ALM}	-	-	-
	NR	g^{gap}_{AVM}	-	-	-
	NR	g^{gap}_{AVM}, g^{gap}_{ALM}	$[15.84, 15.87]^2$	5×10^{-5}	124.23
ALM-	REV	g^{gap}_{AVM}	$[467.3, 1000]$	1×10^{-5}	718.08
	REV	g^{gap}_{AVM}, g^{gap}_{PLM}	$[952.38, 1000]^2$	2×10^{-5}	785.01
	ACC	g^{gap}_{AVM}	$[15.38, 15.87]$	$2e-5$	660.87
	ACC	g^{gap}_{AVM}, g^{gap}_{PLM}	$[14.91, 14.93]^2$	2×10^{-5}	782.3
	NR	g^{gap}_{AVM}	-	-	-
	NR	g^{gap}_{AVM}, g^{gap}_{PLM}	$[10, 10.05]^2$	5×10^{-5}	125.01
ALM,DVA-	REV	g^{gap}_{AVM}	$[250, 500]$	1×10^{-5}	1085.74
	REV	g^{gap}_{AVM}, g^{gap}_{PLM}	$[487.80, 500]^2$	2×10^{-5}	779.75
	ACC	g^{gap}_{AVM}	$[13.88, 14.28]$	1×10^{-5}	1084.23
	ACC	g^{gap}_{AVM}, g^{gap}_{PLM}	$[15.84, 15.87]^2$	2×10^{-5}	782.3
	NR	g^{gap}_{AVM}	-	-	-
	NR	g^{gap}_{AVM}, g^{gap}_{PLM}	$[15.86, 15.87]^2$	2×10^{-5}	779.01
ALM,AVM-	REV	g^{gap}_{PLM}	-	-	-
	ACC	g^{gap}_{PLM}	$[33.33, 1000]$	5×10^{-5}	3619.19
	NR	g^{gap}_{PLM}, g^{gap}_{ALM}	$[10, 13.33]$	5×10^{-5}	3118.45

Table 6.1: Parameter ranges for all experiments, including δ and runtime information. REV=Reversal, ACC=Acceleration, NR=No Response.

6.2.3 3-D Parameter Space

Since the ablation groups we have used in this paper all feature at least one of the primary sensory neurons (*ALM*, *AVM*, and *PLM*) ablated, we can only show the 3-D case for the original animal.

For the 3-D case, in addition to p^{gap}_{AVM} and p^{gap}_{ALM}, we have the p^{gap}_{PLM} conductance. Finally, we get a non-zero value for *no response* in the control, but Table 6.2.2 shows that this value is an order of magnitude smaller than *acceleration* and several orders smaller than *reversal*.



(a) Runtime analysis over T_N

(b) Runtime analysis over dimension

Figure 6.5: Experiment on runtime analysis.

6.2.4 Runtime and Memory Complexity Analysis

The time and memory needed for the procedure depends upon the value of δ used and the size of the parameter space. Assume L_d to be the interval length in the d^{th} dimension. The total number of δ-balls required to cover the parameter space completely is:

$$T_N = \Pi_{d=1}^{D} N_d$$

where D is the number of parameters added to the state vector and $N_d = 2L_d/\delta$. If L_d is the same in all dimensions, $T_N = N_d^D$. We can analyze both runtime and memory complexity based on T_N. If we consider the time and memory required for verifying each δ-ball to be $O(1)$, then the time and memory complexity will both be $O(T_N) = O(N_d^D)$. Note that the complexity also depends on the value of the δ-refinement loop counter. Since we can safely assume that the loop will iterate only a constant number of times, this is not an issue.

Fig. 6.5 illustrates how runtime relates to T_N in one (a) and multiple (b) dimensions. The graph from (a) is the same as the 1D line in (b), but for a larger range of T_N. This increased range more clearly illustrates the linear relationship of runtime to T_N when $D = 1$. Part (b) shows the rates for $D = 1$, $D = 2$ and $D = 3$ over a much smaller range of T_N but helps to demonstrate the effect of dimensionality on time complexity. Since runtime grows at a trinomial rate when $d = 3$, we use the largest δ values (smallest T_N) that correctly cover the parameter space. This is what makes the δ-refinement process imperative; it allows us to correctly verify a property while avoiding runtime blow-up.

Chapter 7

Related Work

In response to the increasing complexity of biological models, *model reduction* and *abstraction* have become active areas of research in computational biology. In [22, 24], the authors propose the idea of *towers of abstraction*, consisting of a hierarchy of models that capture biological details at varying scales of space and complexity. Compositionality and approximation are vital aspects of constructing such a hierarchy, as pointed out in [23, 41].

Singular perturbation [54, 69] and invariant manifold reduction [9, 37] are two popular approaches to reducing multi-scale state-space models of chemical reaction kinetics [18, 38, 77]. The quasi steady state assumption is central to singular perturbation techniques used in [77]. The derivative of fast variables, which evolve on relatively short time scales, is approximated to be zero, resulting in model reduction. Despite being successful for chemical kinetics models, such techniques are not well-suited for Markovian ion channel models. The former involves a constant rate matrix A that renders the system linear, where as in our Markovian models, the rate matrix A is a function of the transmembrane voltage V. The voltage V is itself dependent on the evolution of the Markovian model and this circular dependency causes the overall model to be nonlinear.

Reduction of Markovian ion-channel models, which is the central topic of chapter 3, has been explored in [89, 90]. The focus is on reducing the simulation time, rather than obtaining a formal reduction. In [81], Smith et al. reduce a stochastic model for the sodium-potassium pump by lumping the states of their model. In [21], Fink et al. use mixed formulations of an HH-type model and a Markovian model to reduce the number of state variables for the calcium current. In this thesis, we provide a systematic reduction of the sodium channel. Conventional approaches like [55] use behavioral equivalence to validate the reduced models. Approximate bisimulation, used in this paper, formalize equivalence in a compositional setting and also help in insightful analysis.

Approximate bisimulation equivalence of a detailed model to an abstract model supports such compositional reasoning. In [3], the authors use bisimulation to constrain the *U-projections* of deterministic systems of algebraic differential equations of biochemical networks. The projected automaton is *U-bisimilar* to the original system and thus satisfies the same temporal logic formulae. Compositionality of the models-of-interest is not investigated.

In other related work, a number of efforts, including [10, 78, 4], have been devoted to developing process algebras that are capable of describing biological systems (e.g., interactions among bacteria and bacteriophage viruses) via special biologically motivated operators. Bisimulation in these calculi are typically congruences, thereby allowing compositional reasoning through substitution of equal for equals. These approaches typically do not consider continuous system dynamics and feedback through composition.

Initial work on computing BFs, [31, 33, 35, 29, 49, 52], depended primarily on Sum-of-Squares (SOS) optimization. SOS optimization has also played a crucial role in enabling the automated computation of other Lyapunov-like functions, such as Barrier Certificates [75, 74] and discrepancy functions [14, 43]. In [75, 52], the authors employ an SOSP 2-like approach, which is based on the S-Procedure of [93] and entails strengthening the Lyapunov-like inequalities over the region-of-interest in the state and input spaces.

Despite the success of the above-mentioned approaches, SOS-optimization-based techniques suffer from various drawbacks, such as numerical errors and choosing the forms of the unknown polynomials, which may be crucial for getting good Squared Output Difference (SOD) bounds. The *simulation-based approach* to analyzing stability of dynamical systems in [51], which is closely related to our work, addresses some of these issues. Simulation traces of a given dynamical system are used to compute so-called Candidate Lyapunov Function (CLF). The authors then use an SMT-based ensemble of tools, which includes dReal, to validate the decay requirements over level sets of the CLF. The ***BFComp*** framework differs from the work of [51] in three ways. Firstly, we focus on BFs that characterize IOS of dynamical systems, whereas the authors focus on Lyapunov stability in [51]. Secondly, as shown in our case study, our framework places emphasis on SOD to enable bounding the error that is incurred when a detailed subsystem is replaced by an abstraction within a feedback loop. Lastly, our framework is completely based on Sum-of-Square optimization, whereas the authors use a Linear Programming (LP)-based approach to computing the CLFs.

BFComp builds upon our previous work of [63], which proposed SOSP 1, in several ways. SOSP 1, as a standalone BF-computation technique, suffers from the following limitation: the decay condition of Eq. (4.2) is enforced only on a grid-based discretized input-space. Therefore, SOSP 1 cannot be used to establish incremental input-to-output stability for a continuum of input values. ***BFComp*** overcomes this limitation in two ways. First, SOSP 2, which covers the input space exhaustively, is applied. If the resulting BF fails to provide satisfactory bounds on the SOD, then SOSP 1 is used to compute CBFs. The CBFs are then validated using dReal-based delta reachability. In summary, ***BFComp*** overcomes SOSP 1's limitation of input-space discretization, as well as attempts to provide relatively tight SOD bounds.

BFComp is an extension of our work in [1]. Based on the insightful feedback from the reviewers, we elaborate further on approximating the rate functions of Σ_K with polynomials for computing S_{KH} using SOSP 2. This technique can also be applied to other systems with non-polynomial vector fields. We also provide a detailed performance evaluation of the tool and highlight the tradeoffs between the various parameters, such as the size of input-space grid in SOSP 1, the level set parameter l for validating Candidate Bisimulation Function (CBF)s, and the choice of the $\sigma(,.,)$ functions of SOSP 2.

LP-based computation of Lyapunov-like functions is a promising alternative to SOS optimization. In [80, 79], the authors present LP formulations, based on Handelman representations of polynomials, to compute Lyapunov functions. Consequently, the computation avoids semi-definite programming, which enables SOS optimization, and is therefore more robust to numerical errors. Incorporating such LP-based approaches into our framework is part of the future work.

In chapter 6, we presented a discrepancy function-based reachability analysis of Tap Withdrawal Circuit (TWC) in C. Elegans. Recently, researchers have made steady progress in developing methodology, algorithms, and tools for the verification of such nonlinear ODE models. Reachability analysis, in particular, have been studied in several papers [12, 2, 11, 50].

Discrepancy function-based reachability analysis of nonlinead ODE model has been proposed in [15, 44, 16]. In our work, we, however, apply the technique presented in [19]. Compared with [15, 44, 16], where safety verification algorithms for dynamical switched hybrid systems were proposed, [19] focuses on one crucial component that is missing in [15, 44, 16]: computing discrepancy functions for general nonlinear systems. To the best of our knowledge, other methods could not provide a less conservative over-approximation reach-tube in reasonable time.

Moreover, successfully applying this technique to the verification of TWC model poses a major challenge. It requires both modeling and verification engineering. The TWC model in chapter 6 had to be transformed to normalized gap-junction conductances, the computation of the Jacobian had to be adjusted to take into account the singular matrices, and the routines for checking acceleration and reversal properties from the reach-tubes had to be written.

In other related work on verifying neural circuit, Iyengar et al. present a Pathway Logic (PL) model of neural circuits in the marine mollusk *Aplysia* in [46]. Specifically, the circuits they focus on are those involved in neural plasticity and memory formation. PL systems do not use differential equations, favoring qualitative symbolic models. They do not argue that they can replace traditional ODE systems, but rather that their qualitative insights can support the quantitative analysis of such systems. Neurons are expressed in terms of rewrite rules and data types. Their simulations, unlike our reachability analysis, do not provide exhaustive exploration of the state space. Additionally, PL models are abstractions usually made in collaboration between computer scientists and biologists. Our work meets the biologists on their own terms, using the pre-existing ODE systems developed from physiological experiments.

Tiwari and Talcott [86] build a discrete symbolic model of the neural circuit Central Pattern Generator (CPG) in *Aplysia*. The CPG governs rhythmic foregut motion as the mollusk feeds. Working from a physiological (non-linear ODE) model, they abstract to a discrete system and use the Symbolic Analysis Laboratory (SAL) model checker to verify various properties of this system. They cite the complexity of the original model and the difficulty of parameter estimation as motivation for their abstraction. Neuronal inputs can be positive, negative, or zero and outputs are boolean: a pulse is generated or not. Our approach uses the original biological model of the TW circuit of *C. Elegans* [92], and through reachability analysis, we obtain the parameter ranges of interest.

Chapter 8

Conclusion and Future Work

In chapter, we will first summarize our work and then, present some interesting future work directions.

8.1 Conclusion

In chapter 3, we constructed two-state HH-type models, $\mathcal{H}_{Na}$ and $\mathcal{H}_K$, that can replace the corresponding detailed ion channel models, $\mathcal{M}_{Na}$ (13-state) and $\mathcal{M}_K$ (10-state) respectively, within the IMW model. The reduction was formalized by proving the abstract and the concrete models to be approximately bisimilar. This notion of system equivalence can be used for compositional reasoning.

In chapter 4, we presented Bisimulation Functions and relevant theorems from proving compositional results. In chapter 5, we presented ***BFComp***, an automated framework based on SOS optimization and δ-decidability over the reals for computing BFs that characterize IOS of dynamical systems and provide reasonable bounds on the SOD between the systems.

Given the pair of systems and descriptor functions for the states and input spaces, ***BFComp*** uses SOSP 2, an instance of SOS optimization, to compute a BF. If the BF does not provide satisfactory bounds on the Squared Output Difference (SOD), then SOSP 1, an alternative SOS optimization formulation, is used to compute CBFs, which satisfy the Lyapunov-like decay requirement only over a discretized grid in the input-space. We then appeal to dReal to validate the CBF over the entire input-space. Validation is restricted to the exterior region of the CBF's l-level set to avoid the problem of spurious counterexamples, which can be attributed to dReal's δ-relaxation-based techniques.

We applied ***BFComp*** to compute BFs that appeal to a small-gain theorem, thereby compositionally showing that a detailed four-variable potassium-channel model can be safely replaced by an approximately equivalent one-variable abstraction within a feedback-composed system.

In chapter 6, we performed reachability analysis with discrepancy to automatically determine parameter ranges for three fundamental reactions by *C. Elegans* to tap-withdrawal stimulation: reversal of movement, acceleration, and no response. We followed the lead of the *in vivo* experimental results of [91] to obtain parameter-estimation results for gap-junction conductances for a number of neural-ablation groups. The ranges we present are a significant expansion of the results in [92], where all of the parameters are constant and only the predominant behavior is produced. To the best of our knowledge, these results represent the first formal verification of a biologically realistic (nonlinear ODE) model of a neural circuit in a multicellular organism.

The verification framework we develop is model-agnostic, and allows the techniques of [19] to be applied to general nonlinear ODE models. This is only possible through the careful model and verification engineering developed in this paper.

8.2 Future Work

In chapter 5, we present SOSP-based ***BFComp***. One major drawbacks for this approach is that it uses semidefinite programming, which is numerically more unstable than LP-based method. In [51], Kapinski et al. present LP-based technique to compute Lyapunov functions for nonlinear dynamical systems.

Fig. 8.1 shows our revised ***BFComp***, where we replace the SOSP with LP formulation. The main idea is as follows. We generate a collection of simulation traces of both systems generated by considering the grid points as the input pair (u_1^i, u_2^i), i is the index of the grid point.

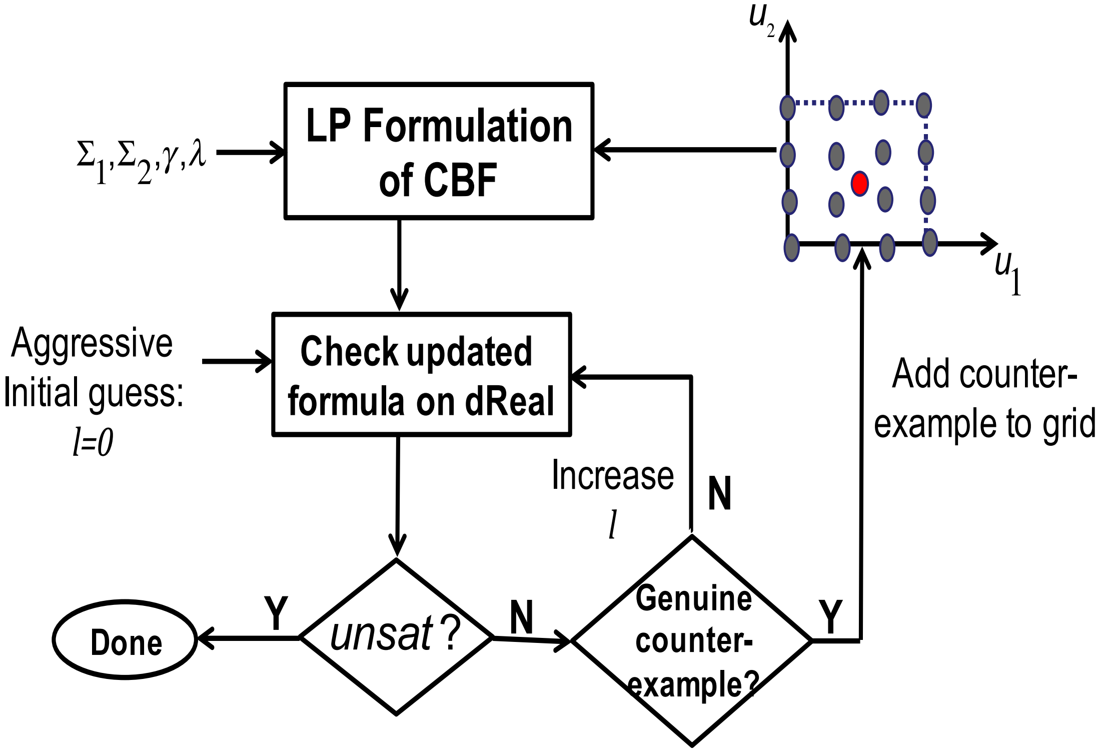

u_2
$\Sigma_1, \Sigma_2, \gamma, \lambda$
LP Formulation of CBF
u_1
Aggressive Initial guess: *l=0*
Check updated formula on dReal
Add counter-example to grid
Increase *l*
N
Y
Done
unsat?
N
Genuine counter-example?
Y

Figure 8.1: Revised ***BFComp***: An Automated Framework for Computing BFs using LP and δ-Decidability.

Assume, traces $\mathbf{x}_{i1}(t_j)$ and $\mathbf{x}_{i2}(t_j)$, $j = 0, 1, 2, ..., t_N$, N is number of simulation steps, are the i-th simulation trace of system 1 and system 2, respectively and the form of BF is $S(\mathbf{x}_1, \mathbf{x}_2) = z^T Q z$, where z is some vector of monomials of $\mathbf{x}_1$ and $\mathbf{x}_2$ and Q is a symmetric matrix. Now for each pair of simulation traces, we construct the following LP probelm:

$$\min_{Q} S(\mathbf{x}_{i1}(t_0), \mathbf{x}_{i2}(t_0)) \tag{8.1}$$

subject to:

$$- S(\mathbf{x}_{i1}(t_j), \mathbf{x}_{i2}(t_j)) + [g_1(\mathbf{x}_{i1}(t_j)) - g_2(\mathbf{x}_{i2}(t_j))]^2 \geq 0, \tag{8.2}$$

$$\begin{aligned} &- \frac{\partial S}{\partial \mathbf{x}_{i1}} f_1(\mathbf{x}_{i1}(t_j), u_1^i) - \frac{\partial S}{\partial \mathbf{x}_{i2}} f_2(\mathbf{x}_{i2}(t_j), u_2^j) - \lambda S(\mathbf{x}_{i1}(t_j), \mathbf{x}_{i2}(t_j)) + \\ &\quad \gamma(u_1^i - u_2^j)^2 \geq 0 \end{aligned} \tag{8.3}$$

$$\gamma > 0 \tag{8.4}$$

$$\lambda \geq 0 \tag{8.5}$$

As alluded in chapter. 6, our results cannot necessarily cover the entire parameter space due to the T_N required, but still enough to verify the properties in question. A potential solution to the incomplete coverage is parallelizing our approach. Luckily, calculating reach-tubes is a data-parallel computation and considered "trivially parallel" for the GPGPU (General-Purpose computing on a Graphics Processing Unit) architecture. This should allow us to run verification experiments in a fraction of the current required time, giving us a potential expansion of coverage.

Finally, we think our techniques can be used to generate new hypotheses for biological experiments. For example, two ablation groups not included in the Wicks et al. experiments are PLM, AVM- and AVD, PVC-. Using reachability analysis we can determine the predominant behaviors for such groups and verify our results experimentally.

Appendix A

Stability properties of voltage-controlled CTMCs for Ion Channels

In this section, we use compartmental systems theory [87] to state and prove stability properties of the constant-voltage ion channel models $\mathcal{M}^v_{Na}$ and $\mathcal{M}^v_K$. This in turn justifies the simulation strategy used in PEFT, i.e. finite-length simulations of $\mathcal{M}^v_{Na}$ and $\mathcal{M}^v_K$ are sufficient to obtain the approximately bisimilar HH-type abstractions $\mathcal{H}^v_{Na}$ and $\mathcal{H}^v_K$.

Theorem 3. *Let $A_x(v)$, be the rate matrix of the corresponding dynamical system M_x, $x \in \{Na, K\}$, as per Definitions 2.1.2 and 2.1.3. For all the values of the bounded input $v \in [V_{res}, V_{max}]$, $A_x(v)$ has exactly one eigenvalue that is 0 and the real part of all the other eigenvalues is negative.*

Proof. First, we will prove a lemma showing that the rate matrices $A_x(v)$ are *compartmental matrices* for $v \in [V_{min}, V_{max}]$.

Lemma 4. *The rate matrix $A_x(v)$ is a compartmental matrix for $v \in [V_{min}, V_{max}]$.*

Proof. A square matrix $M \in \mathbb{R}^{n \times n}$ is called a compartmental system if it satisfies the following properties:

1. All the non-diagonal entries are greater than or equal to 0, i.e $M_{ij} \geq 0$ for $i = 1, \ldots, n, j = 1, \ldots, n, i \neq j$.

2. Sum of the entries along all the columns is less than or equal to 0, i.e. $\sum_{i=1}^{n} M_{ij} \leq 0$, $j = 1, \ldots, n$.

The M_{ij} entry of the matrix is interpreted as the rate of flow of mass from the j^{th} compartment to the j^{th} compartment, $i \neq j$. The diagonal entry M_{ii} is the total rate of outflow from the i^{th} compartment.

For the rate matrices $A_x(v)$, the first property is satisfied as the non-diagonal entries on the i^{th} row of $A_x(v)$ represent incoming transfer rates for state i. These rates are positive as they are exponential functions of the input v. For $\mathcal{M}_{Na}$ and $\mathcal{M}_K$, these functions are listed in Tables 2.1 and 2.2 respectively.

Consider the j^{th} column of $A_x(V)$. The entry $A_{x,ij}$, $i \neq j$ denotes the transfer rate from state j to state i. The diagonal entry of this column $A_{x,jj}$ is the negated sum of all the outgoing rates from state j. Thus, the sum of every column is 0. □

Lemma 5. *The rate matrix $A_x(v)$ is irreducible for $v \in [V_{min}, V_{max}]$.*

Proof. Irreducibility of $A_x(v)$ can be proved using a graph-theoretic argument. We construct a directed graph $G_x^v(W, E)$, where the set of vertices W corresponds to the states of M_x. The set of edges E is constructed as follows. An edge, e_{ij}, from vertex i to vertex j, $i, j = 1, \ldots, n$ exists if $A_{x,ij}(v) \neq 0$.

From linear algebra, we know that the matrix $A_x(v)$ is irreducible if and only if $G_x^v(V, E)$ is connected, i.e. there is a path between every pair of vertices.

If there is an edge from state i to state j of M_x, then the transfer rate $A_{x,ij}(v)$ does not become 0 for any value of v as it an exponential function of v. Also, for a given value of $v \in [V_{min}, V_{max}]$, the corresponding graph G_x^v always remains connected. Thus $A_x(v)$ is irreducible for all $v \in [V_{min}, V_{max}]$. □

Now we introduce the concept of a *trap* of a compartmental system. A trap is a compartment or a set of compartments from which there are no transfers or flows to the environment nor to the compartments that are not in that set. A formal definition is as follows. Let S be a linear compartmental system consisting of compartments $C_1, C_2, \ldots, C_n$. Let $T \subseteq S$, be a subset of the compartments. We number the compartments such that T consists of the compartments $C_m, C_{m+1}, \ldots, C_n$ for $m \leq n$. Let $F \in \mathbb{R}^{n \times n}$ be the rate matrix consistent with the new numbering. The subset T is a trap if and only if $F_{ij} = 0$ for (i, j) such that $j = m, m+1, \ldots, n$ and $i = 0, 1, \ldots, m-1$. A trap is said to be *simple* is it does not strictly contain any traps.

Lemma 6. *The only trap in $A_x(v)$ is the set of all states.*

Proof. As $A_x(v)$ is irreducible, as per Lemma 5, flow between any pair of compartments is nonzero. Thus the only trap is the set of all compartments, □

The proof of the Theorem 3 now follows from Theorems 2.2.4 and 2.2.6 of [87]. Prerequisite conditions have been proved in Lemmas 5 and 6. □

A corollary of Theorem 3 is that for all values of the input $V \in [V_{min}, V_{max}]$ the constant voltage autonomous dynamical systems $\mathcal{M}_{Na}$ and $\mathcal{M}_K$ have stable equilibria. A 0 eigenvalue does not make $A_{Na}(v)$ and $A_K(v)$ Hurwitz, and thus does not lead to asymptotic stability of the equilibria.

Bibliography

[1] Abhishek Murthy, Md. Ariful Islam, Scott A. Smolka, and Radu Grosu. Computing bisimulation functions using sos optimization and δ-decidability over the reals. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, pages 78 – 87. ACM, 2015.

[2] M. Althoff, O. Stursberg, and M. Buss. Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 4042–4048. IEEE, 2008.

[3] M. Antoniotti, C. Piazza, A. Policriti, M. Simeoni, and B. Mishra. Taming the complexity of biochemical models through bisimulation and collapsing: Theory and practice. *Theoretical Computer Science*, 325:2004, 2004.

[4] R. Barbuti, A. Maggiolo-Schettini, P. Milazzo, and A. Troina. Bisimulation congruences in the calculus of looping sequences. In *International Colloquium on Theoretical Aspects of Computing (ICTAC'06), LNCS 4281*, pages 93–107. Springer, 2006.

[5] E. Bartocci, E. Cherry, J. Glimm, R. Grosu, S. A. Smolka, and F. Fenton. Toward real-time simulation of cardiac dynamics. In *Proceedings of the 9th International Conference on Computational Methods in Systems Biology, CMSB'2011*, pages 103–112. ACM, 2011.

[6] F. L. Bauer and C. T. Fike. *Norms and Exclusion Theorems.* Numerische Matematik, 1960.

[7] A. Bueno-Orovio, E. M. Cherry, and F. H. Fenton. Minimal model for human ventricular action potentials in tissue. *Journal of Theoretical Biology*, 253(3):544–560, 2008.

[8] E. M. Cherry and F. H. Fenton. Visualization of spiral and scroll waves in simulated and experimental cardiac tissue. *New Journal of Physics*, 10:125016, 2008.

[9] E. Chiavazzo, A. N. Gorban, and I. V. Karlin. Comparisons of invariant manifolds for model reduction in chemical kinetics. *Comm Comp Phys*, 2:964 –992, 2007.

[10] M. Curti, P. Degano, C. Priami, and C. T. Baldari. Modelling biochemical pathways through enhanced pi-calculus. *Theoretical Computer Science*, 325(1):111–140, 2004.

[11] T. Dang, A. Donzé, O. Maler, and N. Shalev. Sensitive state-space exploration. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 4049–4054. IEEE, 2008.

[12] A. Donzé and O. Maler. Systematic simulation using sensitivity analysis. In *Hybrid Systems: Computation and Control*, pages 174–189. Springer, 2007.

[13] A. Donzé and O. Maler. Systematic simulation using sensitivity analysis. In *Hybrid Systems: Computation and Control*, pages 174–189. Springer, 2007.

[14] P. S. Duggirala, S. Mitra, and M. Viswanathan. Verification of annotated models from executions. In *Proceedings of the Eleventh ACM International Conference on Embedded Software*, EMSOFT '13, pages 26:1–26:10, Piscataway, NJ, USA, 2013. IEEE Press.

[15] P. S. Duggirala, S. Mitra, and M. Viswanathan. Verification of annotated models from executions. In *Proceedings of the International Conference on Embedded Software, EMSOFT 2013*, Montreal, Canada, Sep.-Oct. 2013. IEEE.

[16] P. S. Duggirala, S. Mitra, M. Viswanathan, and M. Potok. C2E2: A verification tool for Stateflow models. In *21st International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2015*, 2015.

[17] P. S. Duggirala, L. Wang, S. Mitra, M. Viswanathan, and C. Muñoz. Temporal precedence checking for switched models and its application to a parallel landing protocol. In *FM 2014: Formal Methods, 19th International Symposium, Proceedings*, volume 8442 of *Lecture Notes in Computer Science*, pages 215–229. Springer, May 2014.

[18] I. R. Epstein and J. A. Pojman. *An Introduction to Nonlinear Chemical Dynamics.* Oxford University Press, London, 1998.

[19] C. Fan and S. Mitra. Bounded verification using on-the-fly discrepancy computation. Technical Report UILU-ENG-15-2201, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Feb. 2015.

[20] F. H. Fenton and E. M. Cherry. Models of cardiac cell. *Scholarpedia*, 3:1868, 2008.

[21] M. Fink and D. Noble. Markov models for ion channels: Versatility versus identifiability and speed. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1896):2161–2179, 2009.

[22] J. Fisher, D. Harel, and T. A. Henzinger. Biology as reactivity. *Communications of the ACM*, 54(10):72–82, October 2011.

[23] J. Fisher and N. Piterman. The executable pathway to biological networks. *Briefings in Functional Genomics*, 9(1):79–92, Jan. 2010.

[24] J. Fisher, N. Piterman, and M. Y. Vardi. The only way is up. In *Proceedings of the 17th International Conference on Formal methods*, FM'11, pages 3–11, Berlin, Heidelberg, 2011. Springer-Verlag.

[25] S. Gao, J. Avigad, and E. M. Clarke. Delta-complete decision procedures for satisfiability over the reals. In *Proceedings of the 6th International Joint Conference on Automated Reasoning*, IJCAR'12, pages 286–300, Berlin, Heidelberg, 2012. Springer-Verlag.

[26] S. Gao, J. Avigad, and E. M. Clarke. Delta-decidability over the reals. In *Proceedings of the 27th Annual IEEE Symposium on Logic in Computer Science (LICS), 2012*, pages 305–314. IEEE, 2012.

[27] S. Gao, J. Avigad, and E. M. Clarke. δcompletee decision procedures for satisfiability over the reals. In *Proceedings of the 6th International Joint Conference on Automated Reasoning*, IJCAR'12, pages 286–300, Berlin, Heidelberg, 2012. Springer-Verlag.

[28] S. Gao, S. Kong, and E. M. Clarke. dreal: An SMT solver for nonlinear theories over the reals. In *Automated Deduction–CADE-24*, pages 208–214. Springer, 2013.

[29] A. Girard. A composition theorem for bisimulation functions. *Pre-print*, 2007. arXiv:1304.5153.

[30] A. Girard. Controller synthesis for safety and reachability via approximate bisimulation. *Automatica*, 48:947–953, 2012.

[31] A. Girard and G. J. Pappas. Approximate bisimulations for nonlinear dynamical systems. In *Proceedings of 44th IEEE Conference on Decision and Control*, Serville, Spain, December 2005.

[32] A. Girard and G. J. Pappas. Approximate bisimulations for nonlinear dynamical systems. In *Proceedings of CDC'05, the 44th International Conference on Decision and Control*, Seville, Spain, December 2005. IEEE.

[33] A. Girard and G. J. Pappas. Approximate bisimulation relations for constrained linear systems. *Automatica*, 43(8):1307 – 1317, August 2007.

[34] A. Girard and G. J. Pappas. Approximate bisimulation relations for constrained linear systems. *Automatica*, 43:1307–1317, 2007.

[35] A. Girard and G. J. Pappas. Approximation metrics for discrete and continuous systems. *Automatic Control, IEEE Transactions on*, 52(5):782 –798, May 2007.

[36] A. Girard and G. J. Pappas. Hierarchical control system design using approximate simulation. *Automatica*, 45(2):566 – 571, 2009.

[37] A. N. Gorban and I. V. Karlin. Method of invariant manifold for chemical kinetics. *Chem. Eng. Sci.*, 58:4751 –4768, 2003.

[38] A. N. Gorban, N. Kazantzis, I. G. Kevrekidis, H. C. Ottinger, and C. Theodoropoulos. *Model reduction and coarse-graining approaches for multiscale phenomena.* Springer, 2006.

[39] J. L. Greenstein, R. Wu, S. Po, G. F. Tomaselli, and R. L. Winslow. Role of the calcium-independent transient outward current I_{to1} in shaping action potential morphology and duration. *Circulation Research*, 87(11):1026–1033, Nov. 2000.

[40] R. Grosu, G. Batt, F. Fenton, J. Glimm, C. L. Guernic, S. A. Smolka, and E. Bartocci. From cardiac cells to genetic regulatory networks. In *Proc. of CAV'11, the 23rd International Conference on Computer Aided Verification*, LNCS, Cliff Lodge, Snowbird, Utah, USA, July 2011. Springer.

[41] R. Grosu, G. Batt, F. H. Fenton, J. Glimm, C. L. Guernic, S. A. Smolka, and E. Bartocci. From cardiac cells to genetic regulatory networks. In *Proceedings of the 23rd International Conference on Computer Aided Verification*, pages 396–411. Springer, 2011.

[42] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, 117:500–544, 1952.

[43] Z. Huang, C. Fan, A. Mereacre, S. Mitra, and M. Kwiatkowska. Invariant verification of nonlinear hybrid automata networks of cardiac cells. In *Proceedings of 26th International Conference on Computer Aided Verification (CAV)*, volume 8559 of *LNCS*, pages 373–390. Springer, 2014.

[44] Z. Huang, C. Fan, A. Mereacre, S. Mitra, and M. Z. Kwiatkowska. Invariant verification of nonlinear hybrid automata networks of cardiac cells. In *Computer Aided Verification, 26th International Conference, CAV 2014, Proceedings*, volume 8559 of *Lecture Notes in Computer Science*, pages 373–390, Vienna, Austria, July 2014. Springer.

[45] M. A. Islam, A. Murthy, E. Bartocci, E. M. Cherry, F. H. Fenton, J. Glimm, S. A. Smolka, and R. Grosu. Model-order reduction of ion channel dynamics using approximate bisimulation. *Theoretical Computer Science*, 2014.

[46] S. M. Iyengar, C. Talcott, R. Mozzachiodi, E. Cataldo, and D. A. Baxter. Executable symbolic models of neural processes. *Network Tools and Applications in Biology (NETTAB07)*, 2007.

[47] V. Iyer, R. Mazhari, and R. L. Winslow. A computational model of the human left-ventricular epicardial myocytes. *Biophysical Journal*, 87(3):1507–1525, 2004.

[48] H. Jeffreys and B. Jeffreys. *Methods of Mathematical Physics*. Cambridge Mathematical Library, 2000.

[49] A. A. Julius and G. J. Pappas. Approximate equivalence and approximate synchronization of metric transition systems. In *Proceedings of 45th IEEE Conference on Decision and Control*, San Diego, CA, 2006, December 2006.

[50] A. A. Julius and G. J. Pappas. Trajectory based verification using local finite-time invariance. In *Hybrid Systems: Computation and Control*, pages 223–236. Springer, 2009.

[51] J. Kapinski, J. V. Deshmukh, S. Sankaranarayanan, and N. Arechiga. Simulation-guided Lyapunov analysis for hybrid dynamical systems. In *Hybrid Systems: Computation and Control (HSCC)*, pages 133–142. ACM Press, 2014.

[52] J. Kapinski, A. Donzé, F. Lerda, H. Maka, S. Wagner, and B. H. Krogh. Control software model checking using bisimulation functions for nonlinear systems. In *47th IEEE Conference on Decision and Control (CDC)*, pages 4024–4029, 2008.

[53] J. P. Keener. Invariant manifold reductions for Markovian ion channel dynamics. *Journal of Mathematical Biology*, 58(3):447–457, July 2009.

[54] J. Kevorkian and J. D. Cole. *Multiple Scale and Singular Perturbation Methods*. Springer-Verlag, 1996.

[55] P. Kienker. Equivalence of aggregated markov models of ion-channel gating. *Proceedings of the Royal Society of London. B. Biological Sciences*, 236(1284):269–309, 1989.

[56] C.-C. Kuo and B. P. Bean. Na channels must deactivate to recover from inactivation. *Neuron*, 12:819–829, 1994.

[57] D. Liberzon. *Switching in Systems and Control.* Springer, 2003.

[58] Lisa A. Irvine, M. Saleet Jafri, and Raimond L. Winslow. Cardiac sodium channel markov model with tempretature dependence and recovery from inactivation. *Biophysical Journal*, 76:1868–1885, 1999.

[59] W. Lohmiller and J. J. E. Slotine. On contraction analysis for non-linear systems. *Automatica*, 1998.

[60] C. H. Luo and Y. Rudy. A dynamic model of the cardiac ventricular action potential. I. Simulations of ionic currents and concentration changes. *Circulation Research*, 74(6):1071–1096, June 1994.

[61] J. Lygeros, G. Pappas, and S. Sastry. An introduction to hybrid systems modeling, analysis and control. In *Preprints of the First Nonlinear Control Network Pedagogical School*, pages 307–329, 1999.

[62] M. A. Islam, A. Murthy, A. Girard, S. A. Smolka, and R. Grosu. Compositionality results for cardiac cell dynamics. In *Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control.* ACM, 2014.

[63] M. A. Islam, A. Murthy, A. Girard, S. A. Smolka, and R. Grosu. Compositionality results for cardiac cell dynamics. In *Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control.* ACM, 2014.

[64] MATLAB ODE45 solver. *Version 7.10.0 (R2010a).* The MathWorks Inc., Natick, Massachusetts, 2010.

[65] MATLAB Open curve fitting toolbox (cftool). *Version 7.10.0 (R2010a).* The MathWorks Inc., Natick, Massachusetts, 2010.

[66] MATLAB Open curve fitting toolbox (cftool). *Version 7.10.0 (R2010a).* The MathWorks Inc., Natick, Massachusetts, 2010.

[67] Matthias Rungger and Majid Zamani. Compositional construction of approximate abstractions. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, pages 68 – 77. ACM, 2015.

[68] R. Milner. *Communication and Concurrency.* Prentice Hall, 1989.

[69] J. D. Murray. *Mathematical Biology.* Springer-Verlag, 1990.

[70] A. Murthy, Md. A. Islam, E. Bartocci, E. Cherry, F. H. Fenton, J. Glimm, S. A. Smolka, and R. Grosu. Approximate bisimulations for sodium channel dynamics. In *Proceedings of CMSB'12, the 10th Conference on Computational Methods in Systems Biology*, LNCS, London, U.K., October 2012. Springer.

[71] C. J. Myers. *Engineering Genetic Circuits.* CRC Press, 2010.

[72] National Science Foundation (NSF). Computational Modeling and Analysis of Complex Systems (CMACS). http://cmacs.cmu.edu.

[73] D. Noble. A modification of the Hodgkin-Huxley equations applicable to Purkinje fibre action and pace-maker potentials. *Journal of Physiology*, 160(2):317–352, 1962.

[74] S. Prajna and A. Jadbabaie. Safety verification of hybrid systems using barrier certificates. In *In Hybrid Systems: Computation and Control*, pages 477–492. Springer, 2004.

[75] S. Prajna, A. Jadbabaie, and G. J. Pappas. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Transactions on Automatic Control*, 52(8):1415–1429, 2007.

[76] S. Prajna, A. Papachristodoulou, P. Seiler, and P. A. Parrilo. *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*, 2004.

[77] O. Radulescu, A. N. Gorban, A. Zinovyev, and A. Lilienbaum. Robust simplifications of multiscale biochemical networks. *BMC Systems Biology*, 2(1):86, 2008.

[78] A. Regev, E. M. Panina, W. Silverman, L. Cardelli, and E. Shapiro. Bioambients: An abstraction for biological compartments. *Theoretical Computer Science*, 325(1):141–167, Sept. 2004.

[79] S. Sankaranarayanan, X. Chen, and E. Abraham. Lyapunov function synthesis using Handelman representations. In *IFAC conference on Nonlinear Control Systems (NOLCOS)*, pages 576–581, 2013.

[80] M. A. B. Sassi, S. Sankaranarayanan, X. Chen, and E. Abraham. Linear relaxations of polynomial positivity for polynomial Lyapunov function synthesis, 2014.

[81] N. Smith and E. Crampin. Development of models of active ion transport for whole-cell modelling: Cardiac sodium-potassium pump as a case study. *Progress in Biophysics and Molecular Biology*, 85:387 – 405, 2004. Modelling Cellular and Tissue Function.

[82] E. D. Sontag. Smooth stabilization implies coprime factorization. *IEEE Transactions on Automatic Control*, 34(4), 1989.

[83] E. D. Sontag. On the input-to-state stability property. *Systems and Control Letters*, 24:351–359, 1995.

[84] E. D. Sontag. Input to state stability: Basic concepts and results. In *Nonlinear and Optimal Control Theory*, pages 163–220. Springer, 2006.

[85] K. H. W. J. ten Tusscher, D. Noble, P. J. Noble, and A. V. Panfilov. A model for human ventricular tissue. *American Journal of Physiology*, 286:H1573–H1589, 2004.

[86] A. Tiwari and C. L. Talcott. Analyzing a discrete model of Aplysia central pattern generator. In *Proceedings of the 6th Conference on Computational Methods in Systems Biology (CMSB)*, pages 347–366. Springer, 2008.

[87] J. van den Hof. *System theory and system identification of compartmental systems*. PhD thesis, University of Groningen, Netherlands, November 1996.

[88] L. R. Varshney. Individual differences. `http://blog.openworm.org/post/107263481195/individual-differences`, 2015.

[89] C. Wang, P. Beyerlein, H. Pospisil, A. Krause, C. Nugent, and W. Dubitzk. An efficient method for modeling kinetic behavior of channel proteins in cardiomyocytes. *IEEE/ACM Trans. on Computational Biology and Bioinformatics*, 9(1):40–51, Jan/Feb 2012.

[90] J. P. Whiteley. Model reduction using a posteriori analysis. *Mathematical Biosciences*, 225(1):44 – 52, 2010.

[91] S. R. Wicks and C. H. Rankin. Integration of mechanosensory stimuli in Caenorhabditis Elegans. *The Journal of Neuroscience*, 15(3):2434–2444, 1995.

[92] S. R. Wicks, C. J. Roehrig, and C. H. Rankin. A dynamic network simulation of the nematode tap withdrawal circuit: Predictions concerning synaptic function using behavioral criteria. *The Journal of Neuroscience*, 16(12):4017–4031, 1996.

[93] V. A. Yakubovich, G. A. Leonov, and A. K. Gelig. Stationary sets in control systems with discontinuous nonlinearities (series on stability, vibration and control of systems, series a, vol. 14), 2004.

[94] Z. P. Jiang, I. M. Y. Mareels, and Y. Wang. A Lyapunov formulation of the nonlinear small-gain theorem for interconnected ISS systems. *Automatica*, 32(8):1211 – 1215, 1996.