Computing Compositional Proofs of Input-to-Output Stability Using SOS Optimization and δ-Decidability

Abhishek Murthy[a,*], Md. Ariful Islam[b], Scott A. Smolka[b], Radu Grosu[c]

[a] *Philips Research North America*

[b] *Department of Computer Science, Stony Brook University*

[c] *Department of Computer Engineering, Vienna University of Technology*

Abstract

We present *BFComp*, an automated framework based on Sum-Of-Squares (SOS) optimization and δ-decidability over the reals, to compute Bisimulation Functions (BFs) that characterize Input-to-Output Stability (IOS) of dynamical systems. BFs are Lyapunov-like functions that decay along the trajectories of a given pair of systems, and can be used to establish the stability of the outputs with respect to bounded input deviations.

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: SOSP 1, which enforces the decay requirements on a discretized grid over the input space, and SOSP 2, which covers the input space exhaustively. SOSP 2 is attempted first, and if the resulting error bounds are not satisfactory, SOSP 1 is used to compute a *Candidate BF* (CBF). The decay requirement for the BFs is then encoded as a δ-decidable formula and validated over a level set of the CBF using the dReal tool. If dReal produces a counterexample containing the states and inputs where the decay requirement is violated, this pair of vectors is used to refine the input-space grid and SOSP 1 is iterated.

[*] This research was performed when the author was a graduate student at Stony Brook University, Stony Brook, NY.

Email addresses: amurthy@cs.stonybrook.edu (Abhishek Murthy), mdaislam@cs.stonybrook.edu (Md. Ariful Islam)

URL: http://www.cs.sunysb.edu/~sas/ (Scott A. Smolka), http://ti.tuwien.ac.at/rts/people/grosu/ (Radu Grosu)

By computing BFs that appeal to a small-gain theorem, the *BFComp* framework can be used to show that a subsystem of a feedback-composed system can be replaced–with bounded error–by an approximately equivalent abstraction, thereby enabling approximate model-order reduction of dynamical systems. The BFs can then be used to obtain bounds on the error between the outputs of the original system and its reduced approximation. To this end, 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 I_{Ks} can be safely replaced by a one-variable Hodgkin-Huxley-type approximation. In addition to a detailed performance evaluation of *BFComp*, our case study also presents workarounds for systems with non-polynomial vector fields, which are not amenable to standard SOS optimizers.

Keywords: Model-order reduction, Cardiac cell model, ionic channel, Approximate bisimulation

1. Introduction

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 ISS to systems with output maps. Since the seminal work of Sontag [33, 34, 35], 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 [20]. These Lyapunov-like functions have been used in the small-gain theorems of [38] 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 [10, 11, 12, 1, 17] are Lyapunov-like functions that decay along the trajectories of a given pair of dynamical systems. Sub-level sets of BFs yield approximate bisimulation relations that generalize the classical notion of bisimulation equivalence of finite-state systems [27] 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

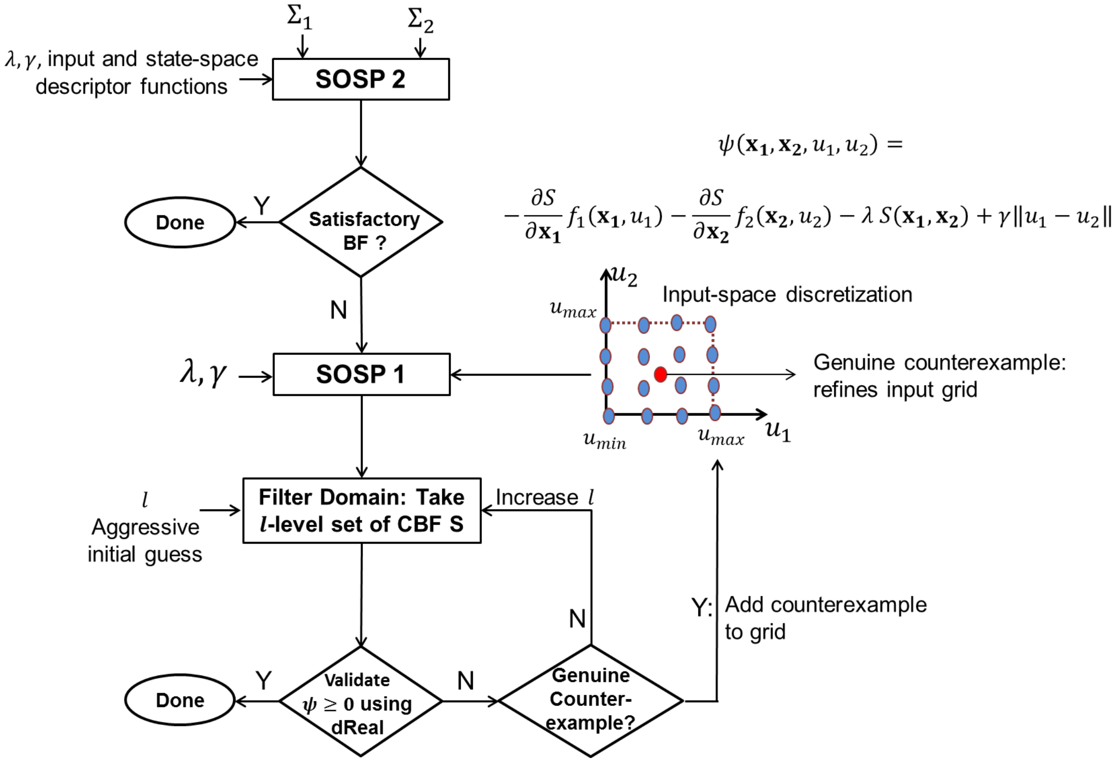

Σ_1
Σ_2
λ, γ, input and state-space descriptor functions
SOSP 2
$\psi(\mathbf{x_1}, \mathbf{x_2}, u_1, u_2) =$
$-\frac{\partial S}{\partial \mathbf{x_1}} f_1(\mathbf{x_1}, u_1) - \frac{\partial S}{\partial \mathbf{x_2}} f_2(\mathbf{x_2}, u_2) - \lambda S(\mathbf{x_1}, \mathbf{x_2}) + \gamma \|u_1 - u_2\|$
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 1: *BFComp*: An Automated Framework for Computing BFs using SOS Optimization and δ-Decidability.

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 [1, 23], 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 [13, 25], reachability analysis [21], and simulation-based verification [14, 5].

In this paper, we present *BFComp*: *an automated framework for computing BFs that characterize IOS of dynamical systems*. *BFComp*, which is illustrated in Fig. 1, leverages Sum-Of-Squares (SOS) optimization and δ-decidability over the reals [9], 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 SOS Problem (SOSP) called SOSP 2 is formulated and solved using MATLAB SOSTOOLS [30]. 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 [23], uses λ and γ to compute a *Candidate BF* (CBF) that satisfies the decay condition of [1] only across a discretized grid over the bounded input space. *BFComp* then appeals to the δ-decidability-based dReal [9] 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 removes the origin of the state space and a finite-sized neighborhood around it, which give rise to the spurious counterexamples, from the domain of ψ. 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 $\mathbf{c}$ 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 [23]. In particular, we use our framework to compute BFs that appeal to the small-gain theorem of [1] 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-Mazhari-Winslow (IMW) ventricular cell model [36]. To the best of our knowledge, this is the first compositional proof of a feedback-based approximate model-order reduction of a biological system.

The rest of the paper develops along the following lines. Section 2 reviews basic definitions and properties of BFs from [1], our previous work on SOSP 1 [23], and the four-variable potassium channel model from [36]. Section 2.4 discusses the one-variable potassium channel abstraction. Section 3 describes our BF-based approach to establishing the substitutivity result within the canonical cell model. Section 6.1 presents SOSP 2, while Section 5 considers our dReal-based validation of SOSP 1 CBFs. Section 6 presents the results of our case study and highlights the implementation issues we faced with *BFComp*. Section 7 presents a performance evaluation of *BFComp* based on our implementations in SOSTOOLS and dReal. Section 8 considers related work. Section 9 contains our concluding remarks and directions for future work.

2. Background

In this section, we review the results on BFs from [1] and our input-space sampling-based algorithm from [23]. Then, 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 (X, X^0, U, f, O, g), where X is the *state space*, $X^0 \subseteq X$ is the set of *initial conditions*, U is the *input space*, $f : X \times U \to X$ is the *vector field* defining the dynamics, O is the set of *outputs*, and $g : X \to O$ maps a state to its output.

We assume f to be Lipschitz continuous, thereby guaranteeing the uniqueness and existence of the trajectories emanating out of X^0. We also assume that the state space $X \subseteq \mathbb{R}^n$ is an invariant for the system; all the trajectories of the system remain within the state space. In most practical settings, X represents the range of values that the state variables can take, and is usually a bounding box. In our case study, the state variables of the ion channel are probabilities, which implies that $X = [0, 1]^n$. This assumption enables us to restrict the BF computation to X, rather than $\mathbb{R}^n$, resulting on better bounds on the error between the original system and its approximation.

2.1. Bisimulation Functions

BFs [1] are non-increasing functions that characterize the joint IOS of two dynamical systems. The following definition, adapted from [1], uses $\parallel . \parallel$ to denote the squared L2-norm.

Definition 1. Let $\Sigma_i = (X_i, X_i^0, U, f_i, Y, g_i)$, $i = 1, 2$, be two dynamical systems such that $X_i \subseteq \mathbb{R}^{n_i}$, $U \subseteq \mathbb{R}^m$ and $Y \subseteq \mathbb{R}^p$. 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 $\mathbf{x}_1 \in X_1$, $\mathbf{x}_2 \in X_2$, $\mathbf{u}_1, \mathbf{u}_2 \in U$:

$$\parallel g_1(\mathbf{x}_1) - g_2(\mathbf{x}_2) \parallel \leq S(\mathbf{x}_1, \mathbf{x}_2), \tag{1}$$

$$\exists \lambda > 0, \gamma \geq 0 \text{ such that } \forall \mathbf{x}_1, \mathbf{x}_2, \mathbf{u}_1, \mathbf{u}_2 :$$
$$\frac{\partial S}{\partial \mathbf{x}_1} f_1(\mathbf{x}_1, \mathbf{u}_1) + \frac{\partial S}{\partial \mathbf{x}_2} f_2(\mathbf{x}_2, \mathbf{u}_2) \leq -\lambda \; S(\mathbf{x}_1, \mathbf{x}_2) + \gamma \parallel \mathbf{u}_1 - \mathbf{u}_2 \parallel . \tag{2}$$

Next, we present a modified version of Theorem 1 of [1], which captures the joint IOS of two systems.

Theorem 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 $t \geq 0$,*

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

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

Proof. See the supplementary document [22]. □

The *feedback composition* $\Sigma_A || \Sigma_B$ of two dynamical systems Σ_A and Σ_B is obtained by feeding the output of Σ_A as the input to Σ_B and vice versa. 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 [1].

Theorem 2. *Let $\Sigma_i = (X_i, X_i^0, U_i, f_i, O_i, g_i)$, $i = 1, 2, A, B$, be dynamical systems such that $U_1 = O_A$, $U_A = O_1$, $U_2 = O_B$ and $U_B = O_2$. Let S_{12}, associated with λ_{12} and γ_{12}, be a BF between Σ_1 and Σ_2. Let S_{AB}, associated with 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:*

$$S(\mathbf{x}_{A1}, \mathbf{x}_{B2}) = \alpha_1 S_{AB}(\mathbf{x}_A, \mathbf{x}_B) + \alpha_2 S_{12}(\mathbf{x}_1, \mathbf{x}_2) \tag{3}$$

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

$$\begin{cases} \frac{\gamma_{12}}{\lambda_{AB}} < \alpha_1 < \frac{\lambda_{12}}{\gamma_{AB}} \quad \textit{and} \quad \alpha_2 = 1 \quad \textit{if} \quad \lambda_{AB} \leq \gamma_{12} \\ \alpha_1 = 1 \quad \textit{and} \quad \frac{\gamma_{AB}}{\lambda_{12}} < \alpha_2 < \frac{\lambda_{AB}}{\gamma_{12}} \quad \textit{if} \quad \lambda_{12} \leq \gamma_{AB} \\ \alpha_1 = 1 \quad \textit{and} \quad \alpha_2 = 1 \quad \textit{in other cases.} \end{cases} \tag{4}$$

Proof. See the supplementary document [22]. □

2.2. Computing CBFs using SOS Optimization and Input-Space Sampling

In [23, 3], 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 [23, 3], we presented the following SOSP for computing BFs using SOS optimization.

Definition 2. *SOSP 1, as per [23, 3], is defined by the following equations.*

$$\text{Minimize } S(\mathbf{x}_1^0, \mathbf{x}_2^0) \tag{5}$$

$$\text{subject to:}$$

$$- S(\mathbf{x}_1, \mathbf{x}_2) + [g_1(\mathbf{x}_1) - g_2(\mathbf{x}_2)]^2 \in \mathbb{S}, \tag{6}$$

$$\exists \lambda > 0, \gamma \geq 0 \text{ such that } \forall \mathbf{x}_1, \mathbf{x}_2, u_1^i \in \mathcal{U}^G, u_2^j \in \mathcal{U}^G : \tag{7}$$

$$\begin{aligned} &- \frac{\partial S}{\partial \mathbf{x}_1} f_1(\mathbf{x}_1, u_1^i) - \frac{\partial S}{\partial \mathbf{x}_2} f_2(\mathbf{x}_2, u_2^j) - \lambda S(\mathbf{x}_1, \mathbf{x}_2) + \\ &\quad \gamma (u_1^i - u_2^j)^2 \in \mathbb{S}. \end{aligned} \tag{8}$$

Typical implementations of Defn. 2 in optimization-based tools, such as SOSTOOLS [30], involve paramterizing S as an ellipsoidal function of $\mathbf{x}_1$ and $\mathbf{x}_2$. The co-efficients of the polynomial function representing S then become the decision variables of the optimization problem.

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 upper-bound of 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. (7) enforces the decay condition for a BF, given by Eq. (2), only on the samples (u_1^i, u_2^j) that comprise the grid $\mathcal{U}$. The validity of Eq. (2) on the entire input space can be verified using delta-decidability, as shown in Sec. 5. In Sec. 6.1, we present an alternative SOSP that enforces Eq. (2) on the entire input space.

Next, we introduce the detailed Markovian potassium-channel model, which is employed as a component in the Iyer-Mazhari-Winslow (IMW) ventricular myocyte model [36].

2.3. The Potassium-Channel Subsystem

Ion channels are special proteins that penetrate the lipid bilayer of an excitable cell's membrane. The channels are selectively permeable to certain ions species. Depending on the conformation of the constituent protein, the channel either allows or inhibits the unidirectional movement of the corresponding ion specie, which results in a transmembrane ionic current. The protein conformation is voltage dependent, thus the name voltage-gated channels.

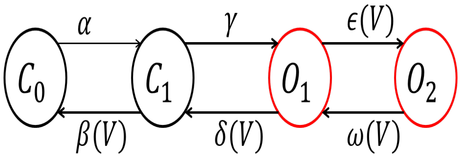


Figure 2: Σ_K: the detailed potassium-channel model, corresponding to the ionic current I_{Ks} in the IMW model.

Definition 3. 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 conformations of the voltage-controlled Continuous Time Markov Chain (CTMC) shown in Fig. 2 in the following order of the labels: $[C_0, C_1, O_1, O_2]$. The dynamics f_K is given by

$$f_K : \dot{\mathbf{x}} = A_K(V)\ \mathbf{x}, \tag{9}$$

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 conformation $\mathbf{x}_j$ to conformation $\mathbf{x}_i$. 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 rates of transitioning out from conformation $\mathbf{x}_i$.

The transition rates, which constitute the entries of $A_K(V)$, are $\alpha = 7.956 \times 10^{-3}, \beta(V) = 0.216 \times exp(-0.00002\ V), \gamma = 3.97 \times 10^{-2}, \delta(V) = (7 \times 10^{-3}) \times exp(-0.15\ V), \epsilon(V) = (7.67 \times 10^{-3}) \times exp(0.087\ V),$ and $\omega(V) = (3.8 \times 10^{-3}) \times exp(-0.014\ V)$.

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 [36].

See [2, 15, 26, 3, 23], for more detailed explanations of similar ion channels models. Next, we define a one-variable abstraction for Σ_K.

2.4. Model-Order Reduction of Σ_K

The curve fitting-based approach of [2, 15] can be used to identify the following one-variable Hodgkin Huxley (HH)-type approximation for Σ_K.

Definition 4. 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

$$f_H : \dot{y} = \alpha_m(V)(1-y) - \beta_m(V)y, \tag{10}$$

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 [2, 15], are as follows.

$$\begin{aligned}\alpha_m(V) &= (-1.331 \times 10^{-10})V^4 - (2.466 \times 10^{-7})V^3 \\ &\quad - (9.723 \times 10^{-6})V^2 - 0.0001231V + 0.001049 \qquad (11)\\ \beta_m(V) &= (4.788 \times 10^{-10})V^6 - (1.547 \times 10^{-8})V^5 \\ &\quad + (1.642 \times 10^{-7})V^4 - (2.85 \times 10^{-6})V^3 \\ &\quad + (6.704 \times 10^{-5})V^2 - (0.0007041)V + 0.003285. \qquad (12)\end{aligned}$$

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 $y_0 = 1.32 \times 10^{-5}$.

3. 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. 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

$$f_C : \dot{V} = -G_K(V - E_K)\ O, \tag{13}$$

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. (13), V_K represents the equilibrium for a fixed-conductance input. Thus, V takes values in $[-35,\ 0]$. Next, we define CCMs Σ_{CK} and Σ_{CH} that reflect this feedback-based composition; see Fig. 3. The models are canonical in the sense that other ion-channel subsystems can be added to obtain the complete IMW model.

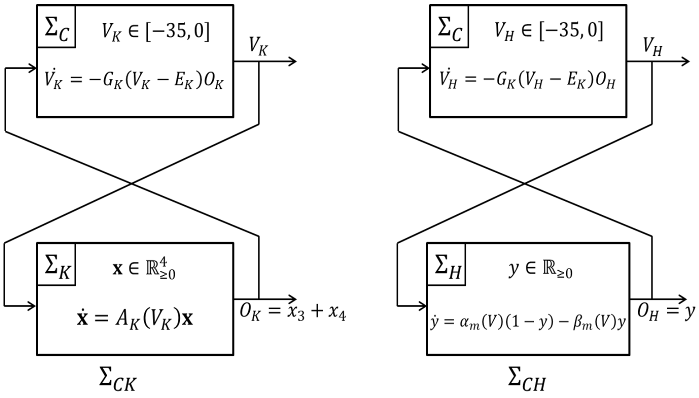

Σ_C
$V_K \in [-35, 0]$
V_K
$\dot{V}_K = -G_K(V_K - E_K)O_K$
Σ_K
$\mathbf{x} \in \mathbb{R}^4_{\geq 0}$
$\dot{\mathbf{x}} = A_K(V_K)\mathbf{x}$
$O_K = x_3 + x_4$
Σ_{CK}
Σ_C
$V_H \in [-35, 0]$
V_H
$\dot{V}_H = -G_K(V_H - E_K)O_H$
Σ_H
$y \in \mathbb{R}_{\geq 0}$
$\dot{y} = \alpha_m(V)(1-y) - \beta_m(V)y$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

Figure 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 [2, 15]; and ii) substituting Σ_H for the detailed model Σ_K within Σ_{CK}.

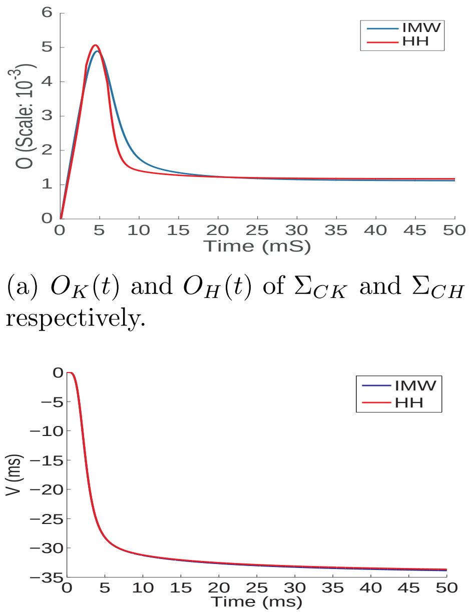
Definition 6. Systems Σ_{CK} and Σ_{CH} (see Fig. 3) are obtained by performing feedback-composition on the voltage subsystem Σ_C with ion-channel subsystems Σ_K

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. (9) and (13). 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.

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. (10) and (13). 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. 4 shows a pair of trajectories of Σ_{CK} and Σ_{CH} that start from nominal initial conditions.



(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 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: $O_{Ks} : 1.1786 \times 10^{-4}$, V: $0.2002\ mV$.

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 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 2; 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.

4. Computing BFs using SOS Optimization

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 2.2, 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:

$$\tau_i(\mathbf{x}_i) = \begin{bmatrix} (x_{i1} - \underline{x_{i1}})(\overline{x_{i1}} - x_{i1}) \\ \vdots \\ (x_{in_i} - \underline{x_{in_i}})(\overline{x_{in_i}} - x_{in_i}) \end{bmatrix}, i = 1, 2. \tag{14}$$

Definition 7. 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.

$$\text{Minimize } S(\mathbf{x}_1^0, \mathbf{x}_2^0) \tag{15}$$

subject to:

$$S(\mathbf{x}_1, \mathbf{x}_2) - [g_1(\mathbf{x}_1) - g_2(\mathbf{x}_2)]^2 \in \mathbb{S}, \tag{16}$$

$$\begin{aligned}
&\forall u_i \in [u_{min}, u_{max}], x_{ij} \in [\underline{x_{ij}}, \overline{x_{ij}}], i = 1, 2, j = 1, \ldots, n_i, \\
&\exists \lambda > 0, \gamma \geq 0, \sigma_1(\mathbf{x}_1, u_1) \in \mathbb{S}, \sigma_2(\mathbf{x}_2, u_2) \in \mathbb{S}, \text{and vectors of} \\
&\text{SOS polynomials } \sigma_3(\mathbf{x}_1, u_1) \text{ and } \sigma_4(\mathbf{x}_2, u_2) \text{ such that :} \\
&-\frac{\partial S}{\partial \mathbf{x}_1} f_1(\mathbf{x}_1, u_1) - \frac{\partial S}{\partial \mathbf{x}_2} f_2(\mathbf{x}_2, u_2) - \lambda S(\mathbf{x}_1, \mathbf{x}_2) + \\
&\gamma(u_1 - u_2)^2 - \sigma_1(\mathbf{x}_1, u_1)\rho(u_1) - \sigma_2(\mathbf{x}_2, u_2)\rho(u_2) - \\
&\sigma_3(\mathbf{x}_1, u_1)\tau_1(\mathbf{x}_1) - \sigma_4(\mathbf{x}_2, u_2)\tau_2(\mathbf{x}_2) \in \mathbb{S}.
\end{aligned} \tag{17}$$

Typical implementations of Defn. 7 in optimization-based tools, such as SOSTOOLS, involve paramterizing S as an ellipsoidal function of $\mathbf{x}_1$ and $\mathbf{x}_2$. The co-efficients of the polynomial function representing S then become the decision variables of the optimization problem.

Note that λ can be handled in two ways: making the value constant or include it in the list of the parameters of S. In the later case, Eq. (17) becomes bilinear in the decision variables and cannot be directly handled by SOSTOOLS. A workaround for this issue can be found in Sec. 4.2 of [28]. We follow the former approach and fix the value of λ, see Sec. 6 for details.

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

Theorem 3. *Consider a feasible solution, $(S, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \lambda, \gamma)$, of the SOSP 2. S satisfies Eqs. (1) and (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. (16):

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

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. (1).

A feasible solution satisfies Eq. (17):

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

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 non-negative when $\mathbf{x}_i \in X_i$, $i = 1, 2$, and $u_1, u_2 \in [u_{min}, u_{max}]$, 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. 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, associated with the constants λ and γ, be a CBF for the two systems; S can be obtained by solving SOSP 1, see Def. 2. A valid solution of SOSP 1 satisfies Eq. (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. (2) over all the inputs, and thus is a BF for the two systems. For this purpose, we use dReal [9], 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. (2) over the entire state and input space, then the following SMT formula must be unsatisfiable:

$$\exists \mathbf{x}_1 \in X_1, \mathbf{x}_2 \in X_2, u_1 \in [u_{min}, u_{max}], u_2 \in [u_{min}, u_{max}] : \psi(\mathbf{x}_1, \mathbf{x}_2, u_1, u_2) < 0. \quad (18)$$

Delta-decidability, which involves relaxing ψ by a parameter $\delta > 0$, can be used to check if Eq. (18) 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 [9, 6, 7] for a formal definition.

When Eq. (18) is presented to dReal, with a pre-determined δ, three possibilities, which are illustrated in Fig. 5, arise. We discuss each of them next.
Case A, dReal returns unsat: Eq. (18) 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

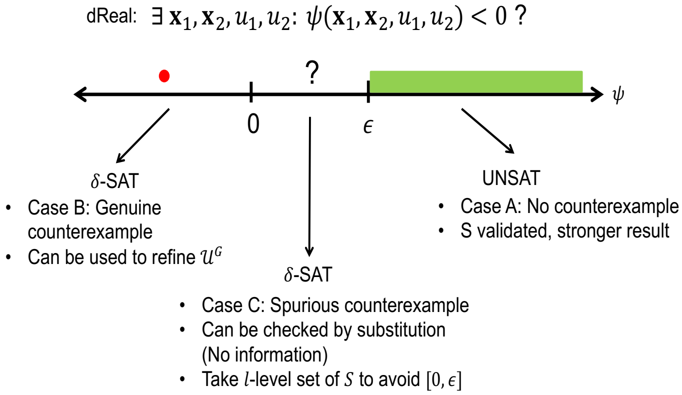

dReal: $\exists\, \mathbf{x}_1, \mathbf{x}_2, u_1, u_2$: $\psi(\mathbf{x}_1, \mathbf{x}_2, u_1, u_2) < 0$?
?
0
ϵ
ψ
δ-SAT
Case B: Genuine counterexample
Can be used to refine $\mathcal{U}^G$
UNSAT
Case A: No counterexample
S validated, stronger result
δ-SAT
Case C: Spurious counterexample
Can be checked by substitution (No information)
Take l-level set of S to avoid $[0, \epsilon]$

Figure 5: Validating SOSP 1-based CBFs using dReal.

fails to satisfy Eq. (2). The input pair is then used to refine $\mathcal{U}^G$ and SOSP 1 is repeated; Eq. (7), 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. (18) 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. (18) is modified as

$$\exists \mathbf{x}_1, \mathbf{x}_2, u_1, u_2 : (S(\mathbf{x}_1, \mathbf{x}_2) \geq l) \wedge (\psi(\mathbf{x}_1, \mathbf{x}_2, u_1, u_2) < 0). \tag{19}$$

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. 7 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: $S^{\geq l} \triangleq \{(\mathbf{x}_1, \mathbf{x}_2) | S(\mathbf{x}_1, \mathbf{x}_2) \geq l\}$. Validating Eq. (18) over $S^{\geq l}$ ensures that Eq. (1) and Eq. (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 4. *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. (1) and Eq. (2) over $S_{12}^{\geq l_1} \times S_{AB}^{\geq l_2}$, 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 2.

Proof. See supplementary document [22]. □

This proposition can be used to show that S will be a level-set-based validated BF if S_{KH} and S_C are both level-set-based validated BFs.

6. 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.

6.1. Computing S_{KH} and S_C using SOSP 2

Automated solvers, such as MATLAB SOSTOOLS [30], 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. (17), 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. (9), 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. 3. 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* [24]. 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. 6.2 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. (17) of Defn. 7. The descriptor functions were obtained from the definitions Σ_K, Σ_H and Σ_C.

SOSP2 was implemented in MATLAB R2013a, SOSTOOLS 2.04 [30] 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* $= 0$.

6.2. Justification for Polynomial Approximations of Rate Functions

Polynomial approximations of the rate functions, which were described in Sec. 6.1, can be justified as follows. We show that the difference in the eigenvalues of $A_K(V)$ and $A_K^p(V)$, 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 1.

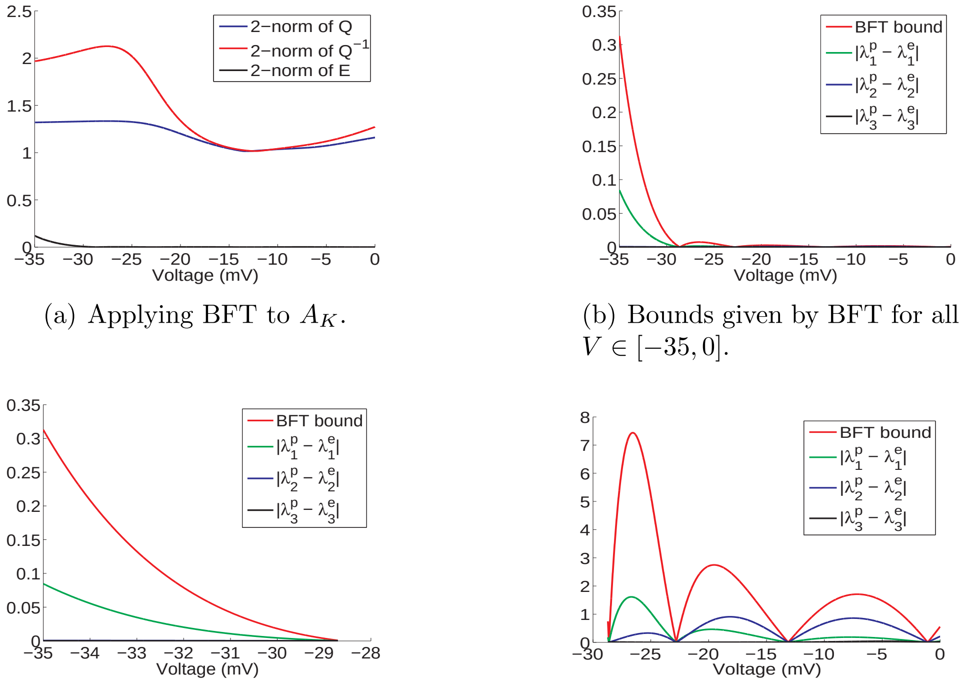
We begin by noting that empirically, the difference between the spectra of $A_K(V)$ and $A_K^p(V)$, which can be made 3×3, after eliminating the redundant state, is minimal. The mean L1 error between the three eigenvalues are $0.005, 4.45 \times 10^{-4}$, and 1.68×10^{-5} for 3000 discrete values of $V \in [-35, 0]$.

Next, we show that the difference between the eigenvalues can be bounded after the fitting process. Note that, in general, Weierstrass Approximation theorem [16] 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) [4] 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. 6 illustrates the BFT on $A_K(V)$ and $A_K^p(V)$. Fig. 6(a) plots the three components that contribute to the upper bound: the norms of the eigenvector matrix and its inverse, and the perturbation matrix, which can be arbitrarily minimized.



(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 6: 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 $A_K^p(V)$.

Fig. 6(b) illustrates the bound obtained by multiplying the aforementioned components. Fig. 6(c) and Fig. 6(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 $A_K^p(V)$.

6.3. 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 [23]. 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; Eq. (19) 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; Eq. (19) was proved to be *unsat* in dReal by choosing $l = 1$.

The validation was implemented using dReal's version 2.14.08-linux [9] 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.

6.4. 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 2, as $\frac{\gamma_{KH}\gamma_C}{\lambda_{KH}\lambda_C} = 0.01 < 1$ in both SOSP 1 and SOSP 2. Applying Theorem 2, 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 [1], λ for S is

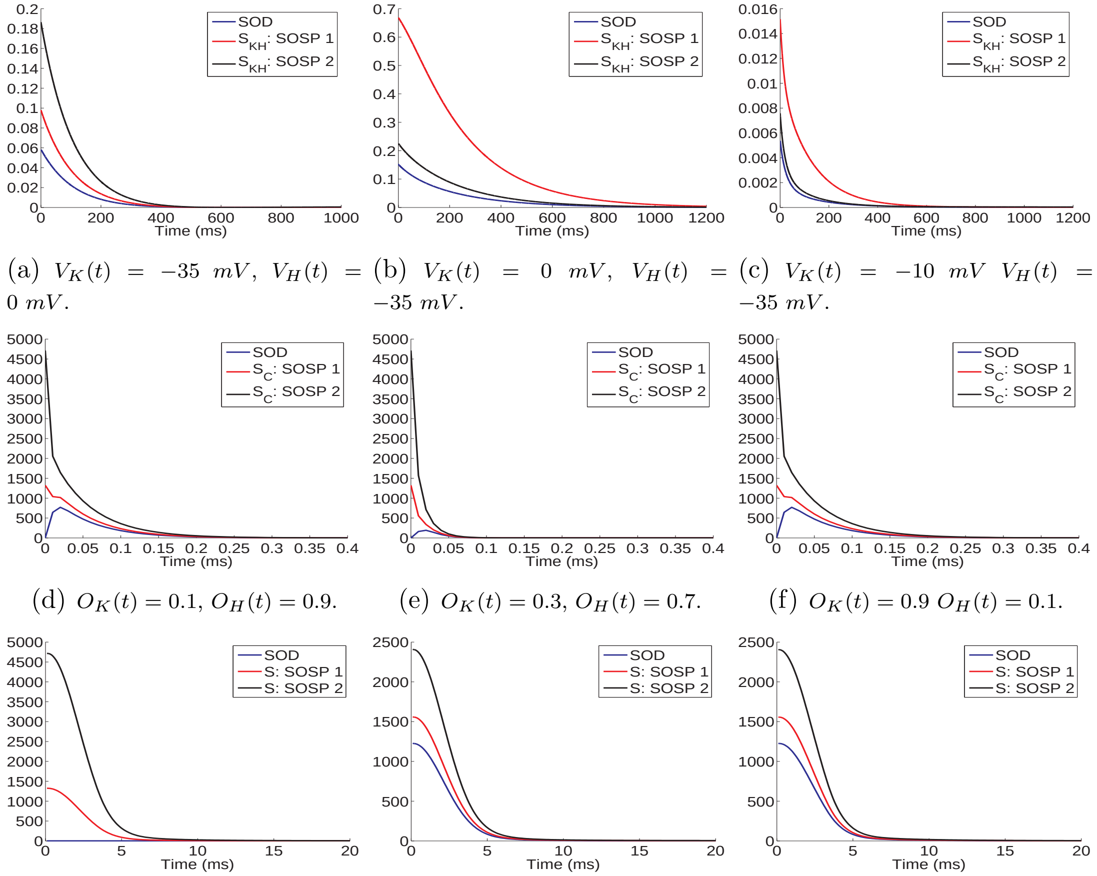
$$\lambda = min\left(\frac{\alpha_1\lambda_{KH} - \alpha_2\gamma_C}{\alpha_1}, \frac{\alpha_2\lambda_C - \alpha_1\gamma_{KH}}{\alpha_2}\right) = 0.0009.$$

6.5. 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 1.

Figs. 7 (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. 3 and 4, and simulated using MATLAB's *ODE45* solver. 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.2.4.



(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 7: 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. The BFs upper bound the SOD and decay along the trajectories.

S_C is plotted in Figs. 7 (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.

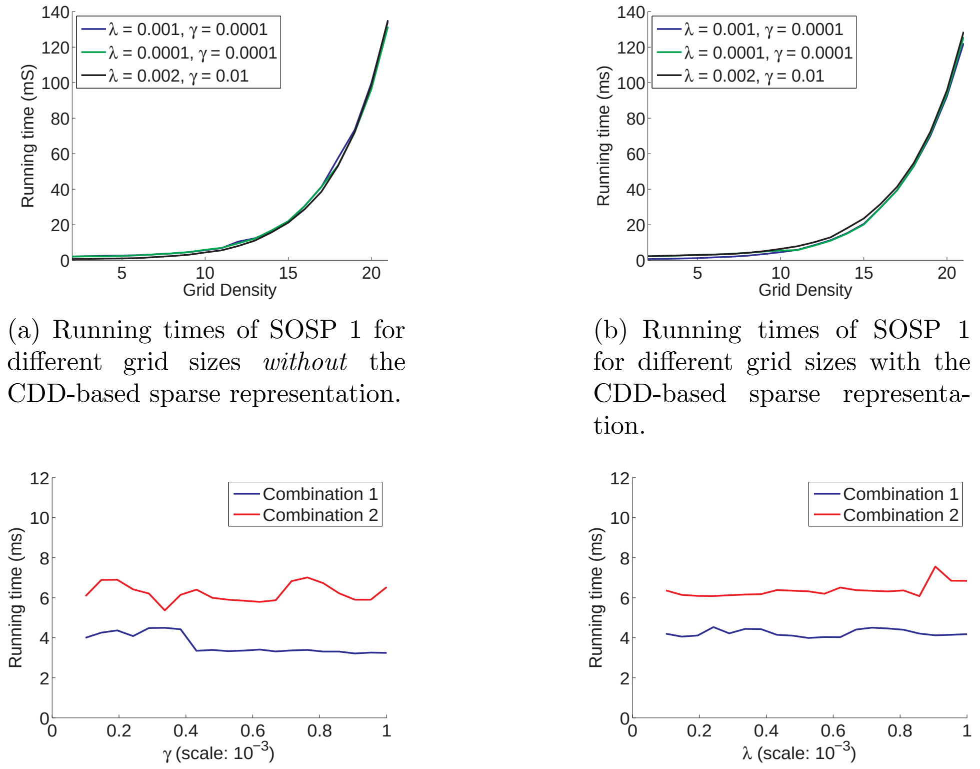
CCMs Σ_{CK} and Σ_{CH} are autonomous dynamical systems and do not receive any external inputs. To visualize the composite BF S, we simulated Σ_{CK} and Σ_{CH} using the Euler method for different initial conditions. Fig. 4 plots the trajectories obtained from these simulations. The corresponding conductance traces of Fig. 4(a) and the voltage traces of Fig. 4(b) empirically validate that the composed models are approximately equivalent as predicted by Theorem 2. BF S along this pair, and two other pairs of trajectories is plotted in Fig. 7 (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.

7. 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. 8.

SOSP 1 uses $\mathcal{U}^G$, the input-space grid; see Def. 2. The size of the grid corresponds to the granularity with which Eq. (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. (2) is satisfied on relatively more points across the input space, which makes the validation step converge faster.

Fig. 8 (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 [30]. This option can



(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 8: Performance evaluation of SOSTOOLS-based implementations of SOSP 1 and SOSP 2.

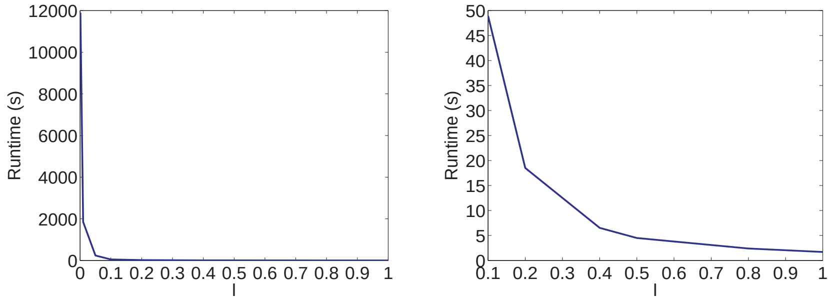
significantly reduce the running time for large SOSPs, but the computation of S_{KH} using SOSP 1 is not affected by this option.

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 [30]. 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. 8(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 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. 19 in dReal depends on level-set parameter l and the δ. A detailed worst case runtime analysis of dReal can be found in [8]. Here, we report empirical run-time statistics for validating the CBF S_{KH} using Eq. 19 for different values of l and δ.

The parameter l defines the domain over which Eq. 19 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. 9 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. 9(b) plots the running time for $l \in [0.1, 1]$. The running time increases exponentially as l decreases. Note that the CBF S_{KH} was successfully validated over all the data points.



(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 9: Running times for dReal across different values of l.

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.

8. Related Work

Initial work on computing BFs, [10, 11, 12, 1, 17, 19], depended primarily on SOS optimization. SOS optimization has also played a crucial role in enabling the automated computation of other Lyapunov-like functions, such as Barrier Certificates [29, 28] and discrepancy functions [5, 14]. In [29, 19], the authors employ an SOSP 2-like approach, which is based on the S-Procedure of [37] 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 SOD bounds. The *simulation-based approach* to analyzing stability of dynamical systems in [18], 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 Functions (CLFs). 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 [18] in three ways. Firstly, we focus on BFs that characterize IOS of dynamical systems, whereas the authors focus on Lyapunov stability in [18]. 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 [23], which proposed SOSP1, in several ways. SOSP1, as a standalone BF-computation technique, suffers from the following limitation: the decay condition of Eq. (2) is enforced only on a grid-based discretized input-space. Therefore, SOSP1 cannot be used to establish incremental input-to-output stability for a continuum of input values. *BFComp* overcomes this limitation in two ways. First, SOSP2, which covers the input space exhaustively, is applied. If the resulting BF fails to provide satisfactory bounds on the SOD, then SOSP1 is used to compute CBFs. The CBFs are then validated using dReal-based delta decidability. In summary, *BFComp* overcomes SOSP1's limitation of input-space discretization, as well as attempts to provide relatively tight SOD bounds.

This paper is an extension of our work in [3]. 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 CBFs, 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 [32, 31], 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.

9. Conclusions

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. 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. As part of the case study, we also presented workarounds for systems with non-polynomial vector fields, which are not amenable to standard SOS optimizers, such as SOSTOOLS. A detailed performance evaluation of *BFComp* was also presented to highlight implementation-level issues.

As future work, we plan to incorporate the SOD bound explicitly in *BFComp*, as a feedback that enables iterative improvement of the BFs. We will also investigate the LP-based approach of [31], which avoids semidefinite programming and is therefore more robust to numerical errors. Finally, we will seek to further generalize our small-gain theorem to enable compositional reasoning with CBFs that are guaranteed to satisfy the decay requirement over level sets, instead of the entire state and input spaces.

Acknowledgments: We would like to thank the anonymous reviewers for their valuable comments. Research supported in part by NSF grants CCF-0926190 and CCF-1018459, and by AFOSR grant FA0550-09-1-0481.

References

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

[2] A. Murthy, M. 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.

[3] 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.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

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

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

[14] 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.

[15] 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.

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

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

[22] M. A. Islam and A. Murthy. Supplementary document. `www.cs.sunysb.edu/~amurthy/hscc15_supp.htm`, 2014.

[23] 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.

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

[25] 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.

[26] Md. A. Islam, A. Murthy, E. Bartocci, S. A. Smolka, and R. Grosu. Compositionality results for cardiac cell dynamics. In *Proceedings of CMSB'13, the 11th Conference on Computational Methods in Systems Biology*, LNCS, Klosterneuburg, Austria, Sept. 2013. Springer.

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

[28] 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.

[29] 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.

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

[31] 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.

[32] M. A. B. Sassi, S. Sankaranarayanan, X. Chen, and E. Abraham. Linear relaxations of polynomial positivity for polynomial Lyapunov function synthesis. *IMA Journal of Mathematical Control and Information*, 2015.

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

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

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

[36] 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.

[37] 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.

[38] 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.