From Cardiac Cells to Genetic Regulatory Networks

R. Grosu[1], G. Batt[2], F. Fenton[3], J. Glimm[4],
C. Le Guernic[5], S.A. Smolka[1], and E. Bartocci[1,4]

[1] Dept. of Comp. Sci., Stony Brook University, Stony Brook, NY, USA
[2] INRIA, Le Cesnay Cedex, France
[3] Dept. of Biomed. Sci., Cornell University, Ithaca, NY
[4] Dept. of Appl. Math. and Sta., Stony Brook University, Stony Brook, NY, USA
[5] Dept. of Comp. Sci., New York University, New York, NY

Abstract. A fundamental question in the treatment of cardiac disorders, such as tachycardia and fibrillation, is under what circumstances does such a disorder arise? To answer to this question, we develop a multiaffine hybrid automaton (MHA) cardiac-cell model, and restate the original question as one of identification of the parameter ranges under which the MHA model accurately reproduces the disorder. The MHA model is obtained from the minimal cardiac model of one of the authors (Fenton) by first bringing it into the form of a canonical, genetic regulatory network, and then linearizing its sigmoidal switches, in an optimal way. By leveraging the Rovergene tool for genetic regulatory networks, we are then able to successfully identify the parameter ranges of interest.

1 Introduction

A fundamental question in the treatment of cardiac abnormalities, such as ventricular tachycardia and fibrillation (see Fig. 1(a)), is under what conditions does such a disorder arise? To answer this question, in vitro and in vivo experimentation is nowadays complemented with the mathematical modeling, analysis and simulation of (networks of) cardiac cells. Among the existing models, differential-equation models (DEMs) are arguably the most popular.

The past two decades have witnessed the development of increasingly sophisticated DEMs, which unravel in great detail the underlying genomic and proteomic processes [14, 18, 21, 11]. Such models are essential in the understanding of molecular interactions, and in the development of novel treatment strategies. However, they also have two significant drawbacks: 1) They often contain too many parameters to be reliably and robustly identified from experimental data. 2) They are often too complex to render their formal analysis or even simulation tractable. We refer to such models as *detailed molecular models* (DMMs).

Approximation (or *abstraction*) is a well-established technique in science and engineering for dealing with complexity. In dynamical systems possessing very fast transient regimes, compared to the rest of the system dynamics, one may use approximation to systematically eliminate these regimes and compensate for
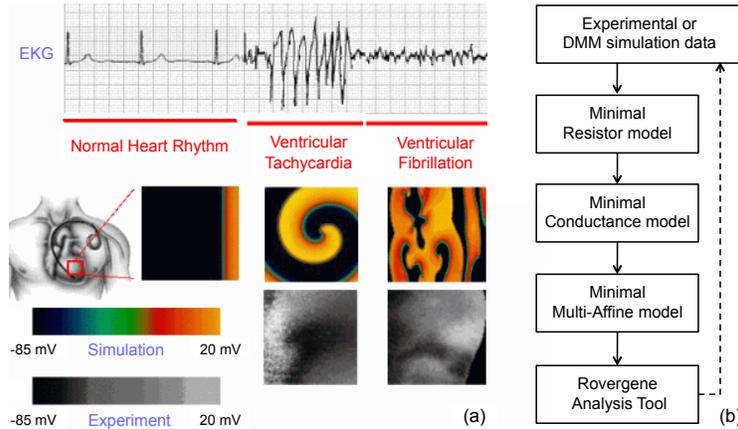
Fig. 1. (a) Emergent behavior in cardiac-cell networks. Top: Electrocardiogram. Middle and bottom: Simulation and experimental mappings of voltage waves occurring in a small rectangular area on the surface of the heart. (b) Overview of our approach.

their elimination [5]. For example, in enzymatic reactions, a substrate reacts very quickly with an enzyme to produce a compound, which subsequently, and much more slowly, breaks down into a product of the reaction and the enzyme itself. In this case, one can use the so-called *quasi-steady-state assumption* to eliminate the fast reaction and derive a sigmoidal dependence of the product on the log of the substrate, called the Michaelis-Menten equation [15, 19].

Similar to the step (or Heaviside) switches used in digital-computer models, sigmoidal switches (dependencies) occur everywhere in biological models: from molecular to cellular models, and from organ to population models [22, 12]. In most cases, they result from the same kind of abstractions discussed above: eliminate fast, transitory components. Unlike in digital-computer models, however, the switching speed of sigmoids plays an important role. Biology is sophisticated!

DEMs with state variables whose rate of change is controlled with sigmoidal switches are still intractable from an analysis point of view. Research of qualitative properties in genetic regulatory networks overcomes this problem by approximating sigmoids with either steps or with ramps [9, 20, 3, 6]. This leads to a piecewise-affine, or piecewise-multiaffine model, respectively. In such models, the dynamics within a hyper-rectangular region is completely determined by the dynamics of its corners, enabling model analysis through the use of powerful discrete abstraction techniques coupled with model-checking techniques [3].

In prior work, one of the authors (Fenton) co-developed an extremely versatile electrical model for cardiac cells involving just 4 state variables and 26 parameters [4]. For reasons to be made clear, we refer to this model as the *minimal resistor model* (MRM). After its parameters are identified from either experimental data or from DMM-based simulation results, the MRM is then able to accurately reproduce the desired behavior. In fact, the MRM identified from experimental data reproduces the experimentally observed behavior with greater accuracy than any of the DMMs. Moreover, its simulation speed is orders of magnitude faster than that of the DMM simulations.

The success of the MRM relies on an abstraction similar to the one above: the large variety of currents traversing the cell membrane are lumped together in only three currents: the fast input current, the slow input current, and the slow output current. These currents are regulated by three gate variables, which together with the voltage, define the MRM's state variables [4]. The lumping process is akin to removing and compensating for fast components [19, 5].

In the MRM context, one may restate the cardiac-disorder question as follows: what are the parameter ranges for which the MRM accurately reproduces cardiac abnormalities? Once these ranges are identified, one may exploit the correspondence between the MRM and DMM models to infer the corresponding parameter ranges in DMMs, and the molecular relevance of the DMMs to target treatment strategies to the components responsible for the disorder.

Despite its simplicity compared to DMM, the MRM is still intractable from an analysis point of view. Its electrical formulation not only uses sigmoidal switches to control the gating variables, but also uses them to model gated resistors. As such, sigmoids occur both as numerators and denominators in the state equations. As part of our effort to simplify the MRM, we prove that *sigmoids are closed under the reciprocal operation*. This allows us to bring the MRM to a canonical form, which we call the *minimal conductance model* (MCM). Intriguingly, the MCM is of the form of a *genetic regulatory network model* (GRM). Hence, this transformation not only exposes the unity of biology, but also allows us to leverage tools developed for GRMs for the analysis of cardiac models.

In GRMs, as well as in MCMs, slow (or shallow) sigmoidal switches cannot be approximated with steps or ramps without considerably distorting the original behavior. We therefore approximate such sigmoids with a succession of ramps, the number of which depends on the desired accuracy. For analysis purposes, it is critical to minimize the number of ramps used and to avoid arbitrary choices. We therefore adapt and extend a dynamic programming technique [17], originally intended for the optimal approximation of digital curves, to find the optimal number of segments (typically of different length) that minimizes, for all sigmoids simultaneously, a sigmoidal-linearization error. This results in a *hybrid-automaton model with multiaffine behavior in each mode* (MHA).

By recasting the intractable parameter-range identification problem for MCMs in terms of MHAs, we now have a tractable problem. Moreover, certain MHA parameter-range identification problems can be seen as GRM parameter-range identification problems: find the parameter ranges that lead to a robust behavior satisfying a given temporal logic property [3]. Hence, for these disorders, we can leverage tools already developed for GRMs to address the MHA problem.

The particular cardiac-disorder question addressed in this paper is *under what circumstances may cardiac-cell excitability be lost?* A region of unexcitable cells can be responsible for ventricular tachycardia or fibrillation: the region becomes an obstacle to a propagating electrical wave, triggering a spiral rotation of the wave (tachycardia); the spiral may then break up into a disordered collection of spirals (fibrillation). Studying the parameter ranges for which cardiac cells loose excitability, and identifying the responsible molecular processes, is thus
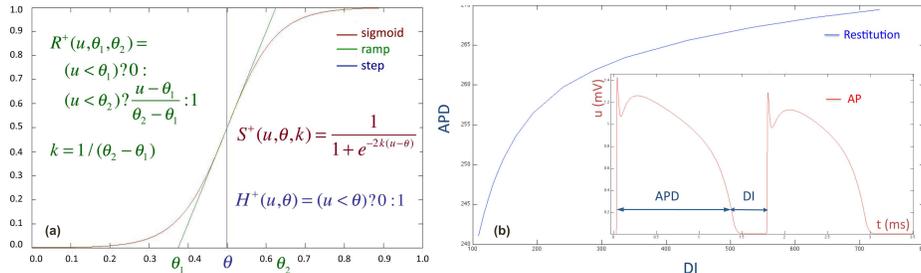
Fig. 2. (a) Threshold-based switching functions. (b) Action potential duration (APD), diastolic interval (DI), and restitution at 10% of the maximum value of the AP.

an important question in the treatment of cardiac disorders. Excitability loss is formulated as an LTL formula, for which the Rovergene tool [3], co-developed by co-author Batt, can automatically infer nontrivial parameter ranges. To the best of our knowledge, this is the first automated parameter-estimation result for cardiac cells.

Our approach is summarized in Fig. 1(b), and the rest of the paper is organized accordingly. Section 2 introduces biological switches and their formal description. Section 3 reviews the MRM. In Section 4, we transform the MRM to an MCM, which is linearized in Section 5. Section 6 considers the parameter-range-identification problem. Section 7 concludes and discusses future work.

2 Biological Switching

As discussed in Section 1, biological switching is sigmoidal. We are interested in a particular class of on $(+)$ and off $(-)$ sigmoidal switches, namely the logistic functions. The sigmoidal on-switch is shown in Fig. 2(a). Equivalently, $S^+(u,k,\theta) = (1 + tanh(k(u - \theta)))/2$. The off-switch is the quantitative complement of the on-switch, and is defined as $S^-(u,k,\theta) = 1 - S^+(u,k,\theta)$.

We typically scale S so that it varies between a minimum value a and maximum value b, both positive:

$$S^+(u,k,\theta,a,b) = a + (b - a)S^+(u,k,\theta), \quad S^-(u,k,\theta,a,b) = S^+(u,k,\theta,b,a)$$

If an on-sigmoid is very steep, then it can be approximated with a Heaviside (or step) switch, as shown in Fig. 2(a). The off-step is given by $H^-(u,\theta) = 1 - H^+(u,\theta)$. As with sigmoids, step-switches can be scaled between a and b.

If an on-sigmoid is steep but not too steep, it can be approximated with a ramp, as shown in Fig. 2(a). The off-ramp is defined as $R^-(u,\theta_1,\theta_2) = 1 - R^+(u,\theta_1, \theta_2)$. Ramps can also be scaled between between a and b. If the sigmoid is shallow, then, as shown in Section 5, it can be approximated with a sequence of ramps.

3 The Minimal Resistor Model

Based on previously published data [16], Fenton co-developed a minimal (resistor) model (MRM) of the action potential produced by human ventricular

myocytes [4]. An *action potential* (AP) is a change in the cell's transmembrane potential u, as a response to an external stimulus (current) e. If the stimulus is delivered from neighboring cells, then its value is $\nabla(D\nabla u)$, where D is the diffusion coefficient and ∇ is the gradient operator. The shape of an AP, its duration (APD), the diastolic interval (DI), and the AP restitution curve (dependence of the APD on the DI) are depicted in Fig. 2(b). Intuitively, the membrane acts like a capacitor, requiring time to recharge after it discharges. The more time it has to recharge, the greater (and longer) the AP. Note that the AP value u is scaled between 0 and 1.5 in the MRM model.

The MRM differs from more complex ionic models (DMMs) in that instead of reproducing a wide range of ion channel currents, it considers the sum of these currents partitioned into three main categories: fast inward J_{fi} (Na-like), slow inward J_{si} (Ca-like), and slow outward J_{so} (K-like). The flow of these total currents is controlled by a fast channel gate v and two slow gates w and s. Together, they retain enough structure such that, with parameters fitted from either experimental data or from DMM simulations, the MRM accurately reproduces the behavior in question.

Among fitted parameters are the voltage-controlled resistances τ_v, τ_w, and τ_s, and the equilibrium values v_∞ and w_∞. The differential equations for the state variables are as follows:

$$
\begin{aligned}
\dot{u} &= e - (J_{fi}(u,v) + J_{si}(u,w,s) + J_{so}(u)) \\
\dot{v} &= H^-(u,\theta_v)\,(v_\infty(u)-v)/\tau_v^-(u) \; - H^+(u,\theta_v)v/\tau_v^+ \\
\dot{w} &= H^-(u,\theta_w)(w_\infty(u)-w)/\tau_w^-(u) - H^+(u,\theta_w)w/\tau_w^+ \\
\dot{s} &= (S^+(u,k_s,u_s)-s)/\tau_s(u)
\end{aligned}
$$

where the three currents are given by the following equations:

$$
\begin{aligned}
J_{fi}(u,v) &= -H^+(u,\theta_v)v(u-\theta_v)(u_u-u)/\tau_{fi} \\
J_{si}(u,w,s) &= -H^+(u,\theta_w)ws/\tau_{si} \\
J_{so}(u) &= +H^-(u,\theta_w)u/\tau_o(u) + H^+(u,\theta_w)/\tau_{so}(u)
\end{aligned}
$$

The voltage-controlled resistances are defined as follows:

$$
\tau_v^-(u) = H^+(u,\theta_o,\tau_{v_1}^-,\tau_{v_2}^-), \; \tau_o(u) = H^-(u,\theta_o,\tau_{o_2},\tau_{o_1}), \; \tau_s(u) = H^+(u,\theta_w,\tau_{s_1},\tau_{s_2})
$$
$$
\tau_w^-(u) = S^-(u,k_w^-,u_w^-,\tau_{w_2}^-,\tau_{w_1}^-), \quad \tau_{so}(u) = S^-(u,k_{so},u_{so},\tau_{so_2},\tau_{so_1})
$$

Finally, the steady state values for gates v and w are:

$$
v_\infty(u) = H^-(u,\theta_o), \quad w_\infty(u) = H^-(u,\theta_o)(1-u/\tau_{v\infty}) + H^+(u,\theta_v^-)w_\infty^*
$$

The values of the parameters for the epicardial (surface) myocytes, as fitted in [4], are given in Fig. 3(a).

4   The Minimal Conductance Model

While much simpler than DMMs, the MRM model is still intractable from an analysis perspective. Its electrical formulation not only uses sigmoidal (and step)

MRM		MCM		MRM		MCM	
Par	Val	Par	Val	Par	Val	Par	Val
θ_o	0.006	θ_o	0.006	$\tau_{v_1}^-$	60	$g_{v_1}^-$	0.01666
u_w^-	0.03	$u_w'^-$	0.0406	$\tau_{v_2}^-$	1150	$g_{v_2}^-$	0.00086
θ_w	0.13	θ_w	0.13	$\tau_{w_1}^-$	60	$g_{w_1}^-$	0.01666
θ_v	0.3	θ_v	0.3	$\tau_{w_2}^-$	15	$g_{w_2}^-$	0.06666
u_{so}	0.65	u_{so}'	1.4824	τ_{o_1}	400	g_{o_1}	0.0025
u_s	0.9087	u_s	0.9087	τ_{o_2}	6	g_{o_2}	0.16666
u_u	1.55	u_u	1.55	τ_{so_1}	30.0181	g_{so_1}	0.03331
w_∞^*	0.94	w_∞^*	0.94	τ_{so_2}	0.9957	g_{so_2}	1.00431
k_w^-	65	k_w^-	65	τ_{s_1}	2.7342	g_{s_1}	0.36453
k_{so}	2.0458	k_{so}	2.0458	τ_{s_2}	16	g_{s_2}	0.0625
k_s	2.0994	k_s	2.0994	τ_{fi}	0.11	g_{fi}	9.0909
τ_v^+	1.4506	g_v^+	0.68936	τ_{si}	1.8875	g_{si}	0.5298
τ_w^+	200	g_w^+	0.005	$\tau_{w\infty}$	0.07	$g_{w\infty}$	142.8571

Fig. 3. (a) Parameter values for the MRM and MCM. (b) Evolution of the MCM HA state variables u, v, w and s in time and as a response to a superthreshold stimulus.

switches to control the state variables, but also uses them to control the value of the resistances. Consequently, sigmoids occur both as numerators and denominators in the state equations.

In order to simplify the MRM model, we prove that scaled sigmoids (or steps) are closed under division; that is, the reciprocal of a scaled sigmoid is also a sigmoid. This result allows us to bring the MRM model to a canonical form, which we call the *minimal conductance model* (MCM).

Theorem 1 (Sigmoid closure). *For $a, b > 0$, scaled sigmoids are closed under multiplicative inverses (division):*

$$S^+(u, k, \theta, a, b)^{-1} = S^-(u, k, \theta + ln(a/b)/2k, b^{-1}, a^{-1})$$

Proof. The proof proceeds by successively transforming the inverse of a scaled sigmoid to a scaled sigmoid. $S^+(u, k, \theta, a, b)^{-1} =$

$$\frac{1}{a + \frac{b-a}{1 + e^{-2k(u-\theta)}}} = \frac{1 + e^{-2k(u-\theta)}}{b + ae^{-2k(u-\theta)}} = \frac{1}{a} \times \frac{a - b + b + ae^{-2k(u-\theta)}}{b + ae^{-2k(u-\theta)}} =$$

$$\frac{1}{a} - \frac{\frac{1}{a} - \frac{1}{b}}{1 + \frac{a}{b}e^{-2k(u-\theta)}} = \frac{1}{a} - \frac{\frac{1}{a} - \frac{1}{b}}{1 + e^{-2k(u - (\theta + \frac{ln\,a - ln\,b}{2k}))}} = S^-(u, k, \theta + \frac{ln\,\frac{a}{b}}{2k}, \frac{1}{b}, \frac{1}{a})$$

Obviously, $H^+(u, \theta, a, b)^{-1} = H^-(u, \theta, b^{-1}, a^{-1})$. Revising the MRM model by replacing each factor $1/\tau_i$ with g_i, and each MRM threshold u_i with the associated MCM threshold u_i', results in the differential equations for the MCM model. Its parameters are given in Fig. 3(a). Note that sigmoids and steps appear only in the numerator. An interesting feature of the MCM is that it has the canonical form of a *genetic regulatory network* (GRN) model (GRM).

Definition 1 (GRM). *The sigmoidal form of a GRM consists of a set of differential equations in which the i-th equation has the following form [22, 12]:*

$(\theta_v \le u < u_u)$

$\dot{u} = e + (u - \theta_v)(u_u - u)v\, g_{fi} + ws\, g_{si} - g_{so}(u)$
$\dot{v} = -v\, g_v^+$
$\dot{w} = -w\, g_w^+$
$\dot{s} = S^+(u, k_s, u_s)\, g_{s_2} - s\, g_{s_2}$

$\theta_w \le u < \theta_v$

$u < \theta_v$ $u \ge \theta_v$

$\dot{u} = e + ws\, g_{si} - g_{so}(u)$
$\dot{v} = -v\, g_{v_2}^-$
$\dot{w} = -w\, g_w^+$
$\dot{s} = S^+(u, k_s, u_s)\, g_{s_2} - s\, g_{s_2}$

$\theta_o \le u < \theta_w$

$u < \theta_w$ $u \ge \theta_w$

$\dot{u} = e - u\, g_{o_2}$
$\dot{v} = -v\, g_{v_2}^-$
$\dot{w} = (w_\infty^* - w)\, g_w^-(u)$
$\dot{s} = S^+(u, k_s, u_s)\, g_{s_1} - s\, g_{s_1}$

$0 \le u < \theta_o$

$u < \theta_o$ $u \ge \theta_o$

$\dot{u} = e - u\, g_{o_1}$
$\dot{v} = (1-v)\, g_{v_1}^-$
$\dot{w} = (1 - u\, g_{w_\infty} - w)\, g_w^-(u)$
$\dot{s} = S^+(u, k_s, u_s)\, g_{s_1} - s\, g_{s_1}$

(a)

$\theta_{12} = \theta_v < u \le u_u = \theta_{26}$

$\dot{u} = e + \sum_{i=12}^{25} R(u, \theta_i, \theta_{i+1}, u_{fi_i}, u_{fi_{i+1}})\, v\, g_{fi} + ws\, g_{si} - \sum_{i=12}^{25} R(u, \theta_i, \theta_{i+1}, u_{so_i}, u_{so_{i+1}})\, g_{so}$
$\dot{v} = -v\, g_v^+$
$\dot{w} = -w\, g_w^+$
$\dot{s} = (\sum_{i=12}^{25} R(u, \theta_i, \theta_{i+1}, u_{s_i}, u_{s_{i+1}}) - s)\, g_{s_2}$

$\theta_8 = \theta_w \le u < \theta_v = \theta_{12}$

$u < \theta_v$ $u \ge \theta_v$

$\dot{u} = e + ws\, g_{si} - \sum_{i=8}^{11} R(u, \theta_i, \theta_{i+1}, u_{so_i}, u_{so_{i+1}})\, g_{so}$
$\dot{v} = -v\, g_{v_2}^-$
$\dot{w} = -w\, g_w^+$
$\dot{s} = (\sum_{i=8}^{11} R(u, \theta_i, \theta_{i+1}, u_{s_i}, u_{s_{i+1}}) - s)\, g_{s_2}$

$\theta_2 = \theta_o \le u < \theta_w = \theta_8$

$u < \theta_w$ $u \ge \theta_w$

$\dot{u} = e - u\, g_{o_2}$
$\dot{v} = -v\, g_{v_2}^-$
$\dot{w} = (w_\infty^* - w) \sum_{i=2}^{7} R(u, \theta_i, \theta_{i+1}, u_{w_i}, u_{w_{i+1}})\, g_{w_h}$
$\dot{s} = (\sum_{i=2}^{7} R(u, \theta_i, \theta_{i+1}, u_{s_i}, u_{s_{i+1}}) - s)\, g_{s_1}$

$\theta_0 = 0 \le u < \theta_o = \theta_2$

$u < \theta_o$ $u \ge \theta_o$

$\dot{u} = e - u\, g_{o_1}$
$\dot{v} = (1-v)\, g_{v_1}^-$
$\dot{w} = (\sum_{i=0}^{1}(R(u, \theta_i, \theta_{i+1}, u_{w_i}^+, u_{w_{i+1}}^+) - wR(u, \theta_i, \theta_{i+1}, u_{w_i}^-, u_{w_{i+1}}^-))\, g_{w_a}$
$\dot{s} = (\sum_{i=0}^{1} R(u, \theta_i, \theta_{i+1}, u_{s_i}, u_{s_{i+1}}) - s)\, g_{s_1}$

(b)

Fig. 4. (a) Hybrid automaton for the MCM model. (b) The multiaffine automaton.

$$\dot{u}_i = \sum_{j=1}^{m_i} a_{ij} \prod_{k=1}^{n_j} S^\pm(u_k, k_k, \theta_k) - \sum_{j=1}^{m_i'} b_{ij} \prod_{k=1}^{n_j'} S^\pm(u_k, k_k, \theta_k)$$

where $S^\pm$ are either on- or off-sigmoidal switches, and a_i and b_i are the expression and inhibition coefficients, respectively.

The second summand is often a simple decay term. Approximating sigmoids with steps (or sequences of ramps) in the GRM and MCM results in a set of piecewise-affine [9, 20] (or piecewise-multiaffine [3]) differential equations.

The steps in the differential equations of the MCM indicate that the MCM specifies a mixed discrete-continuous behavior. In fact, the MCM is equivalent to the MCM hybrid automaton (HA) shown in Fig. 4(a). Consider the partition of the u-axis by the thresholds occurring in step-switches. Each mode of the HA corresponds to the u-interval defined by two successive thresholds, and each transition corresponds to the discrete jump of one of the step-switches. Nonlinear terms are shown in red and exponential degradation terms in blue. The currents have been expanded and partitioned according to the modes.

The behavior of the MCM HA state variables in time and as a response to an super-threshold stimulus is shown in Fig. 3(b). Voltage intervals are highlighted with same color as the one used for corresponding modes in the HA.

Mode $[0\,\theta_o)$ (orange) is a recovering resting mode. In this mode, gates v and w open to their maximum value, and gate s remains closed. Slow sigmoids $S^+(u, k_s, u_s)$ and $g_w^-(u)$ have essentially their minimum value. The only trans-membrane current is the slow output current $J_{so}(u)$, whose overall behavior mimics the ionic (potassium) K-current. This current causes an exponential de-

cay of u. Conductances $g_{v_1}^-$ and $g_{w_1}^-$ control the recovery speed of v and w. Hence, their values are important in properly reproducing AP restitution.

Mode $[\theta_v, u_u)$ (green) is a successful AP initiation mode to a superthreshold stimulus. Factor $(u - \theta_v)(u_u - u)$ in the fast-input current J_{fi} mimics the fast opening of the (sodium) Na channel. This leads to a dramatic membrane depolarization, during which u reaches its peak value u_u. With a slight delay, gate v, which mimics the closing of the Na channel, closes, thus blocking the J_{fi} current. The closing-time of v is solely controlled by the rate constant g_v^+ and the initial value of v. The slow-input (calcium) Ca-like current, J_{si}, is still flowing, which prolongs the duration of the AP. This gives the cardiac muscle time to contract. The value of J_{si} is essentially controlled by gate s, which mimics, through its slow sigmoid, the behavior of the Ca-channel opening-gates. Gate w, which mimics the Ca-channel closing-gate, eventually blocks J_{si}, at rate g_w^+. The slow-output, K-like current, J_{so}, reaches its peak value when the slow sigmoid g_{so} switches on towards its maximum value g_{so_2} $(u > u'_{so})$.

In mode $[\theta_o, \theta_w)$ (blue), gate v starts closing at rate $g_{v_2}^-$, while gate w is still opening. The closing/opening of these gates does not affect the value of u, as this still decays at rate g_{o_2}. It does, however, affect the initial values of v and w for the next AP, which in turn affects the length of this AP. It also affects the AP propagation speed, the so-called AP *conduction velocity* (CV).

In mode $[\theta_w, \theta_v)$ (pink), gate v closes at the same rate as before, but now gate w is also closing, at rate g_w^+. Current J_{so} changes from an exponential decay to a sigmoid, and the slow-input current starts flowing proportional to ws. Gate s adjusts the "expression" coefficient of its slow sigmoid to g_{s_2}.

5 The Piecewise-Multiaffine Model

Although the MCM is simpler than the MRM and considerably simpler than the DMMs, its analysis is still intractable due to the presence of sigmoidal switches. Qualitative GRMs overcome this problem by assuming that every sigmoid is steep enough to be accurately approximated with one step or one ramp. This assumption is generally not appropriate for quantitative GRMs, and therefore not appropriate for the MCM as well: its sigmoids are too shallow to be approximated by either a single step or a single ramp without seriously distorting the original MCM behavior.

A shallow sigmoid can be accurately approximated with a sequence of ramps. This, however, raises a new question: how can one choose as few ramps as possible, while still maintaining a desired approximation error? Additionally, since each ramp introduces a new mode in the HA, how can the *ramp-thresholds* across sigmoids be chosen such that the number of modes is minimized?

In the following, we show that all of these goals are achievable; i.e., there is an optimal solution to the shallow-sigmoid approximation problem, which minimizes a given approximation error in a global way (i.e. simultaneously over a number of sigmoids). Our approach is based on and extends a dynamic programming algorithm developed in the computer graphics community for approximat-

```
function [e,a,b,xb] = optimalLinearApproximation(x,y,S)
Input:   x: Vector of size P        (x coordinates)
         y: Matrix of size P × C  (y coordinates for C curves)
         S: Number >= 2 of desired segments
Output:  e: Errors matrix  - a,b: coefficients matrix
         xb: x-coordinate at breaking point matrix
Initialization of tables
P = size(x, 2);  C = size(y, 1);  P points and C curves
cost  = ones(P,S) * inf;    Cost to a point with n-segments
err  = ones(P,P) * inf;     Cached error of segment i to n
cost(2,1) = 0;              Cost to point 2 with 1 segment
father = ones(P,S) * inf;   Predecessor on opt. s-segm polyline
for p = 2:P                 Traverse all other points

    Get the max (1,p)-line-segment error among all curves
    cost(p,1)=max_{c=1:C}(segmentError(x(1:p),y(c,1:p)));
    father(p,1) = 1; All 1-segment polylines have father point 1
end;
Compute s-segm-polyline cost from point 1 to all other points
for s = 2:S                 Number of segments in the polyline
  for p = 3:P               Next-point-number to consider
    minErr = cost(p-1,s-1); minIndex = p-1; Error of (p-1,p) = 0
    for i = s:p-2           Next-intermediate-point to consider
      if (err(i,p) == Inf)  Error of line segment (i,n) not cashed
        Get the max (i,p)-segment error among all curves
        err(i,p)=max_{c=1:C}(err(i,p), segmentError(x(i:p),y(c,i:p)));
      end; if
      currErr = cost(i,s-1) + err(i,p);  s-segment-polyline error
      Update error and parent
      if (currErr < minErr)  minErr = currErr; minIndex = i;  end;
    end; for i
    cost(p,s) = minErr;          s-segment-polyline minimal cost
    father(p,s) = minIndex;   Last point's father on the polyline
end; end; for p, for s
[e,a,b,xb] = ExtractAnswer;
end
```

```
function [e,a,b,xb] = ExtractAnswer

Output: e,a,b,xb: Like in the optimalLinearApproximation
Initialization of Points matrices and Coefficients/error
ib = zeros(S,S+1); xb = zeros(S,S+1); yb = zeros(C, S,S+1);
a = zeros(C, S,S);    b = zeros(C, S,S);   er = zeros(C, S,S);
Extract error and coefficient matrices
for s = S:-1:1  Traverse polyline segments in inverse order
  ib(s,s+1)  = P;               Get last point number
  xb(s,s+1) = x(ib(s,s+1));    Get x-value for this point
  Traverse all curves - Get y-value for this point
  for c = 1:C   yb(c,s,s+1) = y(c,ib(s,s+1));  end;
  for i = s:-1:1      Traverse predecessor points in inverse order
    ib(s,i) = father(ib(j,i+1),i);   Get predecessor point number
    xb(s,i) = x(ib(s,i));            Get x-value  for this point
    for c = 1:C     Traverse all curves
      yb(c,s,i) = y(c,ib(s,i));      Get y-value for this point
      [er(c,s,i), a(c,s,i), b(c,s,i)] = Compute err, a and b
        segmentError( x(ib(s,i):ib(s,i+1)), y(c,ib(s,i):ib(s,i+1)) );
end; end; end; for c, i, s
end
```

```
function [e,a,b] = segmentError(x,y)
Input: x,y: Digitized curve-segment as x and y vector
Output: e: Error of the line between the first and last point
         a,b: The coeficients defining this segment
e = 0;          Initialize Error
P = size(x,2);  Find out the number of points of x,y
Compute 1-segment linear-interpolation of (x,y) coefficients
a=(y(n)-y(1))/(x(n)-x(1)); b=(y(1)*x(n)-y(n)* x(1))/(x(n) - x(1));
Compute perpendicular-distance error for above line segment
for p = 1:P   Compute error for the each point on the curve
  e = e + (y(p) - a * x(p) - b)^2 / (a^2 +1);  Accumulate least square
end; end
```

Fig. 5. The optimal linear approximation algorithm. Its main function optimalLinear-Approximation uses dynamic programming to compute the optimal segmentation. For each segment, it calls function segmentError to compute the associated error. Function ExtractAnswer is called to extract the answer from the dynamic programming tables.

ing digitized polygonal curves [17] with minimal error. The pseudo MATLAB-code for the main function, optimalLinearApproximation, is shown in Fig. 5, where, for readability, comments are displayed in green.[1]

The function's input is a set of curves (digitized with the same number of points), and a number S of segments to be used by the polylines optimally interpolating the curves. The curves are given as a vector x of P x-coordinates, and a matrix y of C rows, each consisting of P y-coordinates.

The function's output consists of matrices e, a, b and of vector xb. Each entry e(c,s,i), a(c,s,i) and b(c,s,i) gives the error, slope, and y-intercept, respectively, of the i-th segment, in the optimal interpolation polyline of curve c, using s segments. Each entry xb(s,i) is the x-coordinate of breaking point i of the optimal interpolation polylines, using s segments.

---

[1] All MATLAB code and experimental results presented in this paper are available at http://cmacs4heart.pbworks.com/w/page/35180610/CAV2011
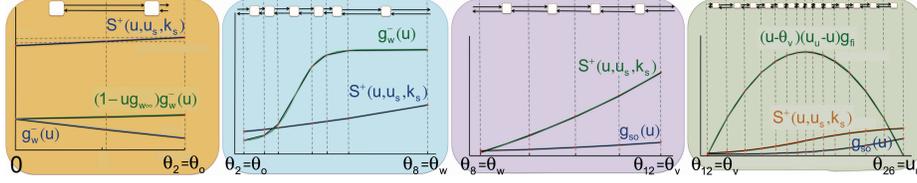
Fig. 6. Linearization of the MCM HA modes with the optimal approximation algorithm. Vertical dashed lines separate the global approximation segments. Each segment corresponds to a mode of the multiaffine hybrid automaton (MHA).

The function first determines the number of points P in each curve, and the number of curves C. It then initializes the dynamic programming storage tables cost(P,S) and father(P,S). Each entry cost(p,s) stores the cost from point 1 to point p of the optimal interpolation polyline consisting of s segments. Each entry father(p,s) stores the predecessor of point p on the optimal-cost polyline consisting of s segments. To speed up the search, we use an error matrix err(P,P), such that each entry err(p,q) caches the maximum error of the segment (p,q) with respect to all of the given curves. Then, in a classic dynamic programming fashion, optimalLinearApproximation fills its solution tables bottom up. First, for all points in the curve (except 1), it computes the cost and father of the 1-segment polyline starting from point 1. Then, knowing the optimal cost of all s-segment polylines from point 1 to any point i that is less than or equal to p, it computes the optimal cost of an $s+1$-segment polyline from point 1 to point $p+1$, by choosing the s-segment polyline, whose cost is minimal when increased with err(i,p+1).

The value stored in err(p,q) is computed with the (nested) function segmentError. Its input consists of vectors x and y, defining a curve segment. Its output consists of error e, and coefficients a and b of the line $y(x) = ax + b$ passing through the first and the last points of the curve segment. Error e is computed by summing up, for each point p on the curve, the square of the perpendicular distance from $(x(p),y(p))$ to y.

Once the solution tables are completely filled, optimalLinearApproximation calls nested function extractAnswer to traverse table father in reverse order, and produce matrices e, a, b, and xb. These matrices have the same format as the output of the caller function, optimalLinearApproximation.

Our implementation of segmentError also allows the use of linear regression instead of linear interpolation. This leads to an optimal approximation that, for the same error, has fewer segments. However, linear regression also introduces discontinuities at the breaking points of the optimal polylines, as the line segment resulting from regression does not typically start and end on the curve. When approximating a single curve, one can use instead the points where the polyline segments intersect. Unfortunately, it is not clear how to generalize this approach to a set of curves, without introducing unnecessary breaking points.

As in GRMs, we assume that the thresholds and slopes of switches (steps and sigmoids) are known and fixed. We can thus linearize the MCM HA one mode at a time. The result is a *multiaffine hybrid automaton* (MHA).
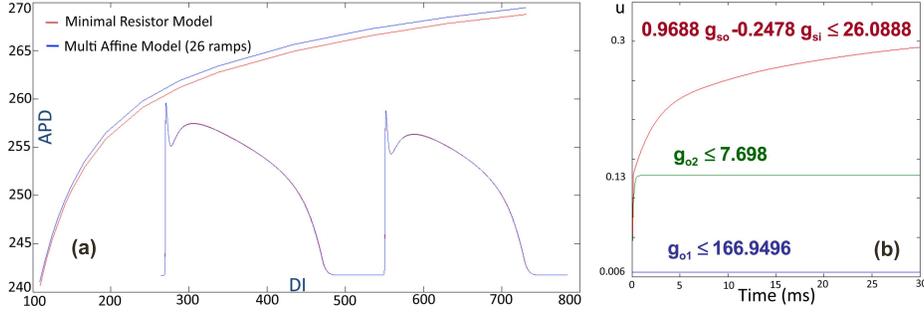
Fig. 7. (a) Comparison of the action potential (AP) shape (the inset figure) and the restitution (dependence of the AP duration (APD) on the diastolic interval (DI)), for the original MRM and the 26 segment MHA, in a cable. (b) Simulation results for the MEM with the values of the conductances g_{o_1}, g_{o_2}, g_{si} and g_{so} taken from the parameter ranges where the MEM robustly satisfies the LTL formula $\mathrm{G}\,(u < \theta_v)$.

Fig. 6 presents our linearization of the MCM modes. Mode $[0\,\theta_o)$ (orange) has three nonlinear functions: sigmoids $g_w^-(u)$ and $S^+(u, u_s, k_s)$, and product $(1 - ug_{w\infty})g_w^-$. The last is treated separately, as the linearization of $g_w^-(u)$ multiplied by $(1 - ug_{w\infty})$ results in a u^2 term. A two-segment linearization (two modes in the MHA) results in a very small error.

Mode $[\theta_o\,,\theta_w)$ (blue) has two nonlinear functions: sigmoids $S^+(u, u_s, k_s)$ and $g_w^-(u)$. In this case, we needed a six-segment linearization (six modes in the MHA) to achieve a small approximation error. Note that the sensitivity of the MCM behavior to the linearization error is also very important.

Mode $[\theta_w\,,\theta_v)$ (pink) has two nonlinear functions: sigmoids $S^+(u, u_s, k_s)$ and $g_{so}(u)$. A four-segment linearization (four modes in the MHA) achieves a small enough approximation error and good overall behavior.

Finally, mode $[\theta_v, u_u)$ (green) has three nonlinear functions: sigmoids $g_{so}(u)$ and $S^+(u, u_s, k_s)$ and the parabolic term $(u - \theta_v)(u_u - u)g_{fi}$. This is the most sophisticated mode. Although gate v closes very rapidly, nullifying the parabolic term in J_{fi}, voltage u traverses in the meantime the entire interval $[\theta_v, u_u)$. Hence, one needs to linearize the parabolic term over this entire interval. This leads to a costly, but inevitable, linearization via 14 segments (modes in the MHA). In our experiments, fewer segments have lead to an unacceptable approximation.

To asses the accuracy of the MHA, we performed extensive 1D and 2D simulations in a cable of 100 cells and a grid of 800×800 cells, respectively. Although the 1D simulation was used to determine the behavior of a single cell only—for example, cell number 50—the use of a cable is necessary, as it is known that cells behave differently when interacting with neighboring cells. Many cardiac models, for example [14], accurately reproduce the AP when simulated in isolation, but fail to reproduce the desired behavior in a cable.

Fig. 7(a) shows the restitution curves of the MRM and MHA models. Each point APD(d) on these curves was obtained by first pacing the MRM and the MHA models at the largest DI value, and then abruptly changing the pacing to DI d. For each value of d, we also compared the AP shapes AP(d). Two such comparison are given as an inset in Fig. 7(a). In both cases (restitution and AP shape), the MHA approximated the MCM with sufficient accuracy.
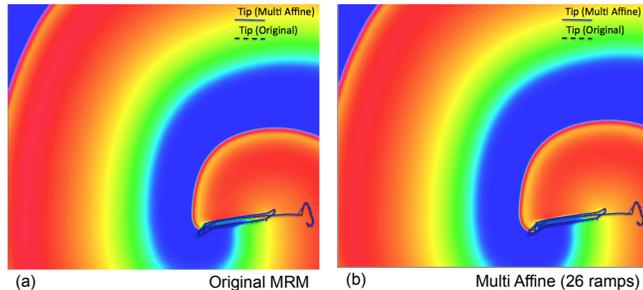
Fig. 8. Snapshot from the simulation of a spiral wave on an 800×800 grid, with isotropic diffusion. (a) The result of the original MRM simulation. (b) The result of the 26-segment MAM simulation. In each figure, we also show the movement over time of the tip of the spiral: dark line for the MAM, and dashed dark line for the MRM.

In Fig. 8, we compare the behavior of the MRM and MHA models on a 2D grid of 800×800 cells. The comparison uses a well-established protocol for the initiation of a spiral wave in cardiac ventricular tissue. We have also tracked the movement of the tip of the spiral over time, which is shown as a dark-blue curve. The 2D simulation confirms the very good accuracy of the MHA model.

Simulations were implemented using CUDA, NVIDIA's parallel computing architecture for GPUs (graphics processing units), and were conducted on a workstation with an Intel Core i7 with 12 Gb RAM, and an NVIDIA Tesla C1060 processor with a 240 GPU processor core and 4GB GDDR3. In this setting, we observed a 1.43 speedup in MHA simulation time compared to MCM simulation time. Note that it is possible to table sigmoid and parabola values to speedup MCM simulation time [7]. This strategy, however, considerably increases the memory demand for the same accuracy and speed, and renders analysis intractable. Our linearization approach can be viewed as an optimal tabling.

6 Cardiac Disorder Parameter-Range Identification

In this section, we show how the MCM HA model can be put into a form suitable for analysis by the Rovergene tool for GRNs [3], thereby allowing us to automatically identify the parameter ranges for a significant cardiac disorder. The linearization algorithm presented in Section 5 returns, for each mode $[\theta_1, \theta_2)$, parameter sequences a_i and b_i, and threshold sequence x_i, where subscript i ranges over the segments chosen in order to fulfil a desired approximation error e. For each i, the returned values define a line segment $y(x) = ax + b$ within the interval $[x_i, x_{i+1})$.

For the first segment, $x_1 = \theta_1$, and for the last segment, $x_n = \theta_2$. Now consider segment $[x_i, x_{i+1})$. The minimum value of $y(x)$ is $y_i = a_i x_i + b_i$ and the maximum value of $y(x)$ is $y_{i+1} = a_i x_{i+1} + b_i$. Together with the threshold values, they define the scaled ramp $R^{\pm}(x, x_i, x_{i+1}, y_i, y_{i+1})$. This is an on $(+)$ ramp, if $y_i \leq y_{i+1}$ and an off $(-)$ ramp if $y_i \geq y_{i+1}$.

Since the ramps must be summed up, for each $i > 1$, we must adjust the y-coordinate by subtracting the maximum value of the previous ramp. Hence, these ramps become $R^{\pm}(x, x_i, x_{i+1}, y_i - y_{i-1}, y_{i+1} - y_{i-1})$.

Once the scaled ramps are computed and summed up, for each mode of the MCM HA, one obtains a multiaffine hybrid automaton (MHA), as shown in Fig. 4(b). The remaining parameters of the MHA are now highlighted in red. The MHA modes have become super-modes, each consisting of as many sub-modes as there are distinct indices in the sums.

The MHA is not suitable as input to the Rovergene GRN analysis tool for two reasons: 1) Rovergene expression terms must be scaled ramps; 2) Rovergene does not support steps. The first problem is overcome by replacing variables with ramps. For example, variable v occurring on the right-hand side of $\dot{u}$ in the green mode, is replaced with the ramp $R^+(v, 0, 1)$. The second problem is overcome by replacing steps with very steep ramps. This amounts to introducing, for each threshold θ_i, separating modes $[\theta_{i-1}, \theta_i)$ and $[\theta_i, \theta_{i+1})$, a just-before θ_i threshold θ_i^-. The equations in mode $[\theta_{i-1}, \theta_i)$ are now multiplied with $R^-(u, \theta_i^-, \theta_i)$ and the ones in mode $[\theta_i, \theta_{i+1})$ are now multiplied with $R^+(u, \theta_i^-, \theta_i)$. The MHA now becomes:

$$
\begin{aligned}
\dot{u} = {}& e - R^-(u, \theta_o^-, \theta_o) R^+(u, 0, \theta_o^-, 0, \theta_o^-)\, g_{o_1} \\
& - R^+(u, \theta_o^-, \theta_o) R^-(u, \theta_w^-, \theta_w) R^+(u, \theta_o, \theta_w^-, \theta_o, \theta_w^-)\, g_{o_2} \\
& + R^+(u, \theta_w^-, \theta_w) R^+(s, 0, 1) R^+(w, 0, 1)\, g_{si} \\
& - R^+(u, \theta_w^-, \theta_w) \sum_{i=8}^{25} R(u, \theta_i, \theta_{i+1}, u_{so_i}, u_{so_{i+1}}) \\
& + R^+(u, \theta_o, \theta_v) R^+(v, 0, 1) \sum_{i=12}^{25} R(u, \theta_i, \theta_{i+1}, u_{fi_i}, u_{fi_{i+1}})\, g_{fi}
\end{aligned}
$$

$$
\begin{aligned}
\dot{v} = {}& R^-(u, \theta_o^-, \theta_o) R^-(v, 0, 1)\, g_{v_1} \\
& - R^+(u, \theta_o^-, \theta_o) R^-(u, \theta_o, \theta_v) R^+(v, 0, 1)\, g_{v_2} \\
& - R^+(u, \theta_o, \theta_v) \sum_{i=12}^{25} R^+(v, 0, 1)\, g_v^+
\end{aligned}
$$

$$
\begin{aligned}
\dot{w} = {}& R^-(u, \theta_o^-, \theta_o) \sum_{i=0}^{1} (R(u, \theta_i, \theta_{i+1}, u_{w_i}^+, u_{w_{i+1}}^+) - R(w, 0, 1) R(u, \theta_i, \theta_{i+1}, u_{w_i}^-, u_{w_{i+1}}^-)) g_{w_a} \\
& + R^+(u, \theta_o^-, \theta_o) R^-(u, \theta_w^-, \theta_w)(w_\infty^* - R(w, 0, 1)) \sum_{i=2}^{7} R(u, \theta_i, \theta_{i+1}, u_{w_i}^+, u_{w_{i+1}}^+)\, g_{w_b} \\
& - R^+(u, \theta_w^-, \theta_w) R(w, 0, 1)\, g_w^+
\end{aligned}
$$

$$
\dot{s} = (R^-(u, \theta_w^-, \theta_w) g_{s_1} + R^+(u, \theta_w^-, \theta_w) g_{s_2}) \sum_{i=0}^{25} (R^+(u, \theta_i, \theta_{i+1}, u_{s_i}, u_{s_{i+1}}) - R^+(s, 0, 1))
$$

where the thresholds (except for the new just-before thresholds θ_i^-), voltages, and conductances match the ones in the MHA. We call this system a *piecewise-multiaffine differential equations model* (MEM).

As discussed in Section 1, a biologically relevant question that we want to answer is, under what circumstances may a cell lose excitability? At the molecular level, this is due to an improper functioning of the cardiac-cell ionic channels.

To identify the molecular processes responsible for this disorder, we first reformulate the question as one of parameter-range identification: What are the parameter ranges for which the MEM fails to generate an AP?

This property may be specified in linear temporal logic (LTL) as $G\,(u < \theta_v)$, where G is the LTL globally (always) temporal operator. The property states that in all executions of the MEM and in all moments of time along a single execution, the voltage value is below θ_v. (Note that in an LTL formula, there is an implicit quantification over all executions.) We would like this property to hold for all stimulus durations. In terms of the MHA of Fig. 4(b), this property is true due to the interplay of the ranges of conductances g_{o_1}, g_{o_2}, g_{si} and g_{so}.

To identify these ranges in an automated fashion, we use the Rovergene tool, with the MEM and LTL formula given above as input, and with the following initial region:

$$u \in [0, \theta_1], \quad v \in [0.95, 1], \quad w \in [0.95, 1], \quad s \in [0, 0.01]$$

The u-thresholds and the initial region impose the following partition on the ranges of state variables (for u we have added the just-before thresholds):

$$u : [0,\ldots,\theta_{29}], \quad v, w : [0, 0.95, 1], \quad s : [0, 0.01, 1]$$

Parameter ranges with biological significance for the conductances were taken as below. They include known values for normal and abnormal cell behavior.

$$g_{o_1} \in [1, 180], \quad g_{o_2} \in [0, 10], \quad g_{si} \in [0.1, 100], \quad g_{so} \in [0.9, 50]$$

The behavior of the MEM in each hypercube of the state-space partition is completely determined by its corners, so the existence of transitions from one hypercube to its neighbors can be computed by evaluating the MEM in the corners. In each corner, the MEM becomes an affine system in the MEM parameters. Solving these systems, one obtains the separating hyperplanes of positive and negative sign of the derivatives in the MEM. Finally, taking into account the desired LTL property, one obtains the parameter ranges for which the property is satisfied. In our case the ranges returned are:

$$166.9494 \leq g_{o_1} \leq 180, \quad 7.6982 \leq g_{o_2} \leq 10$$
$$-0.24784\, g_{si} + 0.9688\, g_{so} \leq 26.0888$$

They have the following meaning. If $g_{o_1} \geq 166.9494$, then, regardless of the duration of the magnitude-1 stimulus applied, the voltage u never leaves the orange interval (mode) $[0, \theta_o]$. If, on the other hand, $g_{o_1} < 166.9494$, then u reaches the blue interval (mode) $[\theta_o, \theta_w)$. Since we are considering stimuli of any width (time is abstracted away by Rovergene), once u enters the blue range, its behavior is completely determined by this mode. If $g_{o_2} \geq 7.6982$, then u can never leave the mode. If $g_{o_2} < 7.6982$, then u will enter the pink interval (mode) $[\theta_w, \theta_v)$. In this mode, the behavior of u is determined by the interplay between g_{so} and g_{si}. If the above linear combination is satisfied, one can never leave this mode.

The corresponding simulation, for a sample of values in the above parameter ranges, is shown in Fig. 7(b). To ensure that we run the same model as Rovergene, we also developed a Rovergene simulation tool that, given a Rovergene model as input, simulates its dynamic behavior in MATLAB. This tool proved to be an invaluable debugging tool during model encoding in Rovergene.

7   Conclusions and Future Work

Although formal techniques were used before to analyze cardiac-cell properties (see e.g. our work in [23, 10]), this paper presents, to the best of our knowledge, the first approach for automatically identifying parameter ranges of a biologically-relevant cardiac model, guaranteeing that the model accurately reproduces a particular cardiac disorder.

Our approach takes the nonlinear cardiac model of [4], brings it first into a genetic regulatory network sigmoidal form, and then linearizes and transforms it into a piecewise-multiaffine set of differential equations. It then leverages tools previously developed for automatic parameter-range identification in genetic regulatory networks [3] to automatically and robustly check a cardiac disorder expressed as a linear temporal logic (LTL) formula.

The particular property we considered in this paper is lack of cardiac-cell excitability. This is an invariant property, where time is abstracted away. In future work, we plan to investigate more sophisticated LTL properties of single cells, as well as reachability properties of cell networks (e.g. spirals).

Many abnormalities responsible for cardiac disorders are time- or rate-dependent properties that cannot be checked with the Rovergene tool, due to its underlying finite-automata abstraction. Action potential duration and spiral breakup (fibrillation) are examples of such properties. We therefore plan to investigate new parameter-range identification approaches that, in contrast, use abstractions based on timed-automata[2, 13] or even linear-automata[1, 8].

References

1. R. Alur, C. Courcoubetis, T. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. In *Proceedings of the 11th International Conference on Analysis and optimization of systems: Discrete Event Systems*, Lecture Notes in Control and Information Sciences 199, pages 331–351. Springer-Verlag, 1994.
2. R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 2(126):183–235, Apr. 1994.
3. G. Batt, C. Belta, and R. Weiss. Temporal logic analysis of gene networks under parameter uncertainty. *IEEE Trans. of Automatic Control*, 53:215–229, 2008.
4. A. Bueno-Orovio, M. Cherry, and F. Fenton. Minimal model for human ventricular action potentials in tissue. *Journal of Theoretical Biology*, 253(3):544–560, 2008.
5. C.M. and S. Orszag. *Advanced Mathematical Methods for Scientists and Engineers: Asymptotic Methods and Perturbation Theory*. Springer Verlag, 1999.

6. H. de Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9(1):69–105, 2002.

7. F. Fenton, E. Cherry, H. Hastings, and S. Evans. Real-time computer simulations of excitable media: Java as a scientific language and as a wrapper for C and Fortran programs. *BioSystems*, 64(1-3):73–96, 2002.

8. G. Frehse. PHAVer: algorithmic verification of hybrid systems past HyTech. *Springer International Journal of Software Tools fot Technology Transfet*, 10(3):263–279, 2008.

9. L. Glass and S. S. A. Kauffman. The logical analysis of continuous nonlinear biochemical control networks. *Journal of Math. Biology*, 39(1):103–129, 1973.

10. R. Grosu, S. Smolka, F. Corradini, A. Wasilewska, E. Entcheva, and E. Bartocci. Learning and detecting emergent behavior in networks of cardiac myocytes. *Communications of the ACM (CACM)*, 52(3):1–10, March 2009.

11. V. Iyer, R. Mazhari, and R. Winslow. A computational model of the human leftventricular epicardial myocytes. *Biophysical Journal*, 87(3):1507–1525, 2004.

12. J. Keener and J. James Sneyd. *Mathematical Physiology*. Springer, 2008.

13. K. Larsen, P. Pettersson, and W. Yi. UPPAAL in a nutshell. *Springer International Journal of Software Tools for Technology Transfer*, 1, 1997.

14. 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.

15. L. Michaelis and M. Menten. Die kinetik der invertinwirkung. *Biochemische Zeitschrift*, 49:333–369, 1913.

16. J. Morgan, D. Cunningham, and E. Rowland. Dispersion of monophasic action potential duration: Demonstrable in humans after premature ventricular extrastimulation but not in steady state. *J. Am. Coll. Cardiol.*, 19:1244–1253, 1992.

17. J. Perez and E. Vidal. Optimum polygonal approximation of digitized curves. *Pattern Recognition Letters*, 15:743–750, 1994.

18. L. Priebe and D. Beuckelmann. Simulation study of cellular electric properties in heart failure. *Circulation Research*, 82:1206–1223, 1998.

19. L. Segel and M. Slemrod. The quasi-steady-state assumption: A case study in perturbation. *SIAM Review*, 31(3):446–477, 1989.

20. T. T. Mestl, E. Plahte, and S. Omholt. A mathematical framework for describing and analysing gene regulatory networks. *Journal of Theoretical Biology*, 176(2):291–300, 1995.

21. K. Ten Tusscher, D. Noble, P. Noble, and A. Panfilov. A model for human ventricular tissue. *American Journal of Physiology*, 286:H1573–H1589, 2004.

22. G. Yagil and E. E. Yagil. On the relation between effector concentration and the rate of induced enzyme synthesis. *Biophysical Journal*, 11(1):11–27, 1971.

23. P. Ye, R. Grosu, S. Smolka, and E. Entcheva. Formal analysis of abnormal excitation in cardiac tissue. In *Proc. of CMSB'08, the 6th International Conference on Computational Methods in Systems Biology*, LNBI, Rostock, Germany, October 2008. Springer Verlag.