# 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]Department of Computer Science, Stony Brook University, Stony Brook, NY
[2]Institut National de Recherche en Informatique and en Automatique, Le Cesnay Cedex, France
[3]Department of Biomedical Sciences, Cornell University, Ithaca, NY
[4]Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY
[5]Department of Computer Science, New York University, New York, NY

***Abstract*—One of the fundamental questions in the treatment of ventricular 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 and global way. By leveraging the Rovergene tool for genetic regulatory networks, we are then able to successfully identify parameter ranges of interest.**

## I. Introduction

A fundamental question in the treatment of cardiac abnormalities, such as ventricular tachycardia and fibrillation [1] (see Figure 1), 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 myriad existing models, (partial) differential-equations 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 [2], [3], [4], [5]. Such models are essential in the understanding of the 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 their elimination [6]. 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

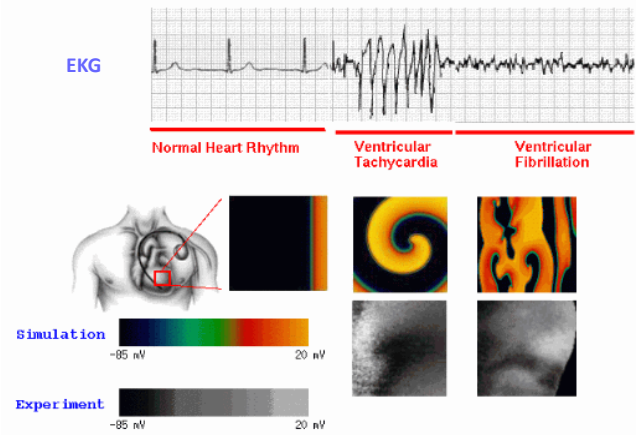


Fig. 1. Emergent behavior in cardiac-cell networks. Top: Electrocardiogram. Middle and bottom: Simulation and experimental mappings of voltage waves occurring in a small rectangular area on the surface of the heart.

and derive a sigmoidal dependence of the product on the log of the substrate, called the Michaelis-Menten equation [7], [8].

Similar to the rectangular (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 [9], [10]. 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 in genetic regulatory networks overcomes this problem by approximating sigmoids with either steps or with ramps [11], [12], [13], [14]. This leads to a piecewise-affine (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 and model-checking techniques [13].

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 [15]. 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 DMM-based simulation results, the MRM is able to accurately reproduce the desired behavior. In fact,

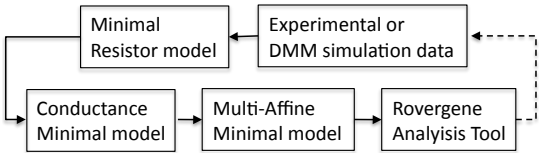


Fig. 2. Overview of our approach

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 DMMs.

The success of the MRM intuitively relies on the same kind of abstraction as discussed 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 [15]. The lumping process is akin to removing and compensating for fast components [8], [6].

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 DMMs to infer the corresponding parameter ranges in the DMMs. The molecular relevance of the DMMs then allows to target treatment strategies to the molecular components responsible for the disorder.

Despite its simplicity compared to DMMs, 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 reciprocal operation*. This allows us to bring the MRM to a canonical form, which we call the *minimal conductance model* (MCM). Intriguingly, this is 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.

While in a qualitative GRM sigmoidal switches are approximated with steps or ramps, this is impossible to do in the MCM without considerably distorting its original behavior. We therefore approximate the shallow sigmoids of the MCM 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 a dynamic programming technique [16], to find the optimal number of segments, typically of different length, minimizing for all sigmoids at the same time, 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-identification problems can be seen as a special case of the GRM parameter-identification problem: find the uncertain (due

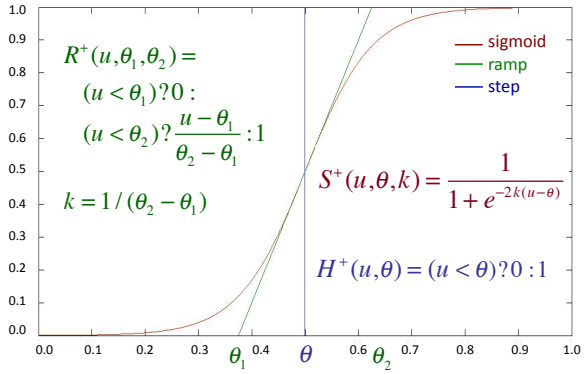


Fig. 3. Threshold-based switching functions.

to experimental barriers or measurement errors) parameter ranges that lead to a robust behavior satisfying a given temporal logic property [13]. Hence, for these disorders, we can leverage tools already developed for GRMs to address the MHA problem. Conversely, qualitative GRM can exploit our linearization technique in order to obtain accurate piecewise-multiaffine models from the experimental data.

The particular cardiac-disorder question addressed by this paper is, *under what circumstances may cardiac-cell excitability be lost?* A region of unexcitable cells within cardiac tissue, can be responsible for tachycardia or fibrillation: 1) The region becomes an obstacle, in the way of the propagating electrical wave; 2) This triggers a spiral rotation of the wave, a condition known as tachycardia; 3) The spiral may then eventually break up into many spirals, a condition known as fibrillation. Studying the parameter ranges for which cardiac cells loose excitability, and identifying the responsible molecular processes, is therefore an important question in the treatment of cardiac disorders. Loss of excitability can be formulated as an LTL formula, for which the Rovergene tool [13] automatically infers nontrivial parameter ranges. To the best of our knowledge, this is the first result of this kind.

Our overall approach is summarized in Figure 2. The rest of this paper is organized accordingly, as follows.. Section II introduces biological switches and their formal description. Section III reviews the MRM. In Section IV, we transform the MRM to an MCM, which is linearized in Section V. Section VI considers the parameter-range-identification problem. Section VII concludes and discusses future work.

## II. Biological Switching

As discussed in Section I, biological switching is sigmoidal. In this paper, we are interested in a particular class of on (+) and off (-) sigmoidal switches, namely the logistic functions. The sigmoidal on-switch is shown in Figure 3. The off-switch is defined as: $S^-(u,k,\theta) = 1 - S^+(u,k,\theta)$.

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

$$S^+(u,k,\theta,u_m,u_M) = u_m + (u_M - u_m)S^+(u,k,\theta)$$
$$S^-(u,k,\theta,u_m,u_M) = S^+(u,k,\theta,u_M,u_m)$$

Note that $S^+(u,k,\theta) = (1+tanh(k(u-\theta)))/2$, which is another way of expressing the logistic function.

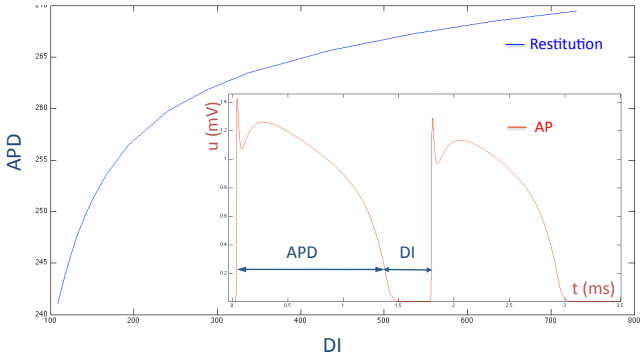


Fig. 4. APD, DI, and restitution at 10% of the maximum value of the AP.

If an on-sigmoid is very steep, then it can be approximated with a Heaviside (or step) switch, as shown in Figure 3. The off-step is defined as: $H^-(u,\theta) = 1 - H^+(u,\theta)$. As with sigmoids, step-switches can be scaled between $u_m$ and $u_M$.

If an on-sigmoid is steep but not very steep, it can be approximated with a ramp, as shown in Figure 3. 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 $u_m$ and $u_M$.

If the sigmoid is slow, then, as shown in Section V, it can be approximated with a sequence of ramps.

## III. The Minimal Resistor Model

Based on previously published data [17], Fenton co-developed a minimal (resistor) model (MRM), of the action potential produced by human ventricular myocytes [15]. An *action potential* (AP) is a change in the cell's transmembrane potential $u$, as a response to an external stimulus (current). If the stimulus $e$ comes from the neighboring cells, then its value is $\nabla(D\nabla u)$, where $D$ is the diffusion coefficient and $\nabla$ 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 Figure 4. Intuitively, the membrane acts like a capacitor, requiring some 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 (DMM) 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 simulation results, the MRM accurately reproduces the behavior in question.

Among fitted parameters are the voltage-controlled resistances $\tau_v$, $\tau_w$, and $\tau_s$, and the equilibrium values $v_\infty$ and $w_\infty$. The differential equations for the state variables are as follows:

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

TABLE I
Parameter values for MRM and MCM

| MRM Par | MRM Val | MCM Par | MCM Val | MRM Par | MRM Val | MCM Par | MCM Val |
|---|---|---|---|---|---|---|---|
| $\theta_o$ | 0.006 | $\theta_o$ | 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 |
| $\theta_w$ | 0.13 | $\theta_w$ | 0.13 | $\tau_{w_1}^-$ | 60 | $g_{w_1}^-$ | 0.01666 |
| $\theta_v$ | 0.3 | $\theta_v$ | 0.3 | $\tau_{w_2}^-$ | 15 | $g_{w_2}^-$ | 0.06666 |
| $u_{sc}$ | 0.65 | $u'_{sc}$ | 1.4824 | $\tau_{o_1}$ | 400 | $g_{o_1}$ | 0.0025 |
| $u_s$ | 0.9087 | $u_s$ | 0.9087 | $\tau_{o_2}$ | 6 | $g_{o_2}$ | 0.16666 |
| $u_u$ | 1.55 | $u_u$ | 1.55 | $\tau_{so_1}$ | 30.0181 | $g_{so_1}$ | 0.03331 |
| $w_\infty^*$ | 0.94 | $w_\infty^*$ | 0.94 | $\tau_{so_2}$ | 0.9957 | $g_{so_2}$ | 1.00431 |
| $k_w^-$ | 65 | $k_w^-$ | 65 | $\tau_{s_1}$ | 2.7342 | $g_{s_1}$ | 0.36453 |
| $k_{sc}$ | 2.0458 | $k_{so}$ | 2.0458 | $\tau_{s_2}$ | 16 | $g_{s_2}$ | 0.0625 |
| $k_s$ | 2.0994 | $k_s$ | 2.0994 | $\tau_{fi}$ | 0.11 | $g_{fi}$ | 9.0909 |
| $\tau_v^+$ | 1.4506 | $g_v^+$ | 0.68936 | $\tau_{si}$ | 1.8875 | $g_{si}$ | 0.5298 |
| $\tau_w^+$ | 200 | $g_w^+$ | 0.005 | $\tau_{w\infty}$ | 0.07 | $g_{w\infty}$ | 142.8571 |

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:

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

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

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

The values of the parameters for the epicardial (surface) myocytes, as fitted in [15], are given in Table I.

## IV. 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) switches to control the state variables, but also uses them to control the value of the resistances. As a consequence, 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:

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

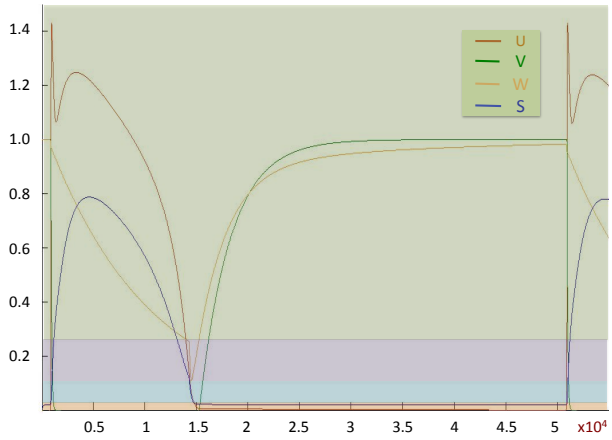


Fig. 5. Response of MCM HA state variables to a stimulus.

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

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

■

Obviously, $H^+(u,\theta,a,b)^{-1}=H^-(u,\theta,b^{-1},a^{-1})$. Table I gives the values of the conductances $g_i=1/\tau_i$ of the MCM for each resistance $\tau_i$. For each threshold $u_i$ in the MRM model, we also provide the associated MCM threshold $u_i'$.

Revising the MRM model by replacing each factor $1/\tau_i$ with $g_i$ and each threshold $u_i$ with the associated MCM threshold $u_i'$ results in the differential equations for the MCM model. Note that in the MCM, sigmoids and steps only appear at the numerator. For further clarity and future reference, we provide here the complete definition of the MCM model. The differential equations for the state variables are:

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

where the three currents are given by the following equations:

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

The voltage-controlled conductances are defined as follows:

$$\begin{aligned}
g_v^-(u) &= H^-(u,\theta_o,g_{v_2}^-,g_{v_1}^-)\\
g_o(u) &= H^+(u,\theta_o,g_{o_1},g_{o_2})\\
g_s(u) &= H^-(u,\theta_w,g_{s_2},g_{s_1})\\
g_w^-(u) &= S^+(u,k_w^-,u_w^{'-},g_{w_1}^-,g_{w_2}^-)\\
g_{so}(u) &= S^+(u,k_{so},u_{so}',g_{so_1},g_{so_2})
\end{aligned}$$

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

$$\begin{aligned}
v_\infty(u) &= H^-(u,\theta_o)\\
w_\infty(u) &= H^-(u,\theta_o)(1-u\,g_{v\infty})+H^+(u,\theta_o)w_\infty^*
\end{aligned}$$

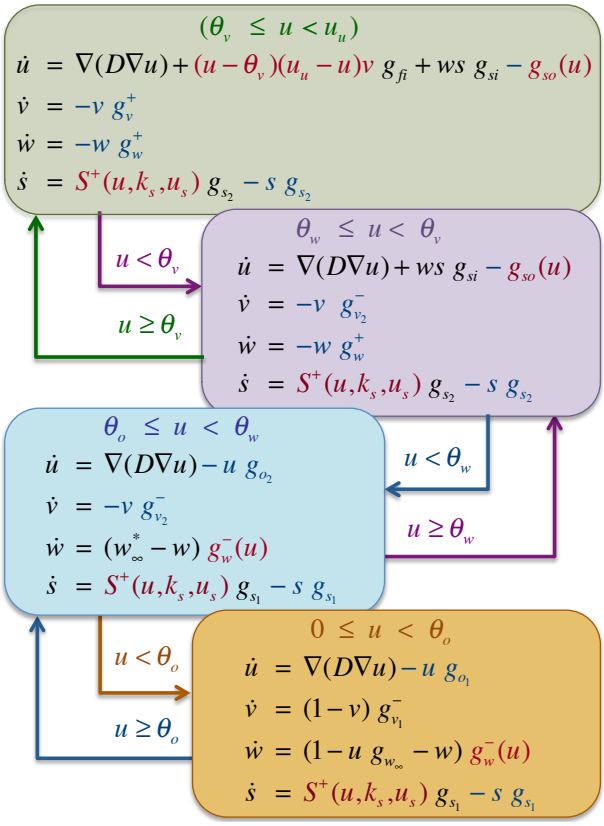


Fig. 6. Hybrid automaton for the MCM model.

An interesting feature of the MCM is that it has the canonical form of a genetic regulatory network (GRN) model.

**Definition 1** (GRM)**.** *The sigmoidal form of a* genetic regulatory network model *(GRM) consists of a set of differential equations in which the i-th equation has the following form:*

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

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

Approximating sigmoids with steps (or ramps) in the GRM, that is, assuming that switching is very steep (or steep), results in a set of piecewise-affine (-multiaffine) differential equations [11], [12] (or [13]). The second summand is often a simple decay term. We believe that that further advances will uncover genetic regulatory networks that contain slow sigmoids as well.

Since each (free) variable on the right-hand side of the differential equations of the MCM corresponds to a ramp, the MCM has the form of a GRM, where the very steep (or steep) sigmoids have been approximated with steps (or ramps). The $u^2$ term in $J_{fi}$ can be understood as the product of two complementary on- and off-ramps. This is not allowed in [13] but, in general, is a valid term of a GRM [9], [10].

The step-switches occurring in the differential equations of the MCM, indicate that the MCM specifies a mixed discrete and continuous behavior. In fact, the MCM is equivalent to a

hybrid automaton (HA). 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. The MCM HA is given in Figure 6. Nonlinear terms are shown in red color and exponential degradation terms in blue. The currents have been expanded and broken according to the modes.

The behavior of the MCM HA state variables in time and as a response to an super-threshold external stimulus is shown in Figure 5. The voltage intervals corresponding to the modes of the HA are highlighted using the same color as 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 transmembrane current is the slow output current $J_{so}(u)$, whose overall behavior mimics the ionic (potassium) K-current. This current causes an exponential decay 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 mode, initiated by a super-threshold 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 $u_u$. At the same time, but 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, providing the cardiac muscle enough 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 a myocyte network.

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 $g_{s_2}$.

## V. The 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. GRN models overcome this problem by assuming that every sigmoid is steep enough to be accurately approximated with either one step or one ramp. While this assumption is debatable for GRMs, it does not hold for MCMs. Its sigmoids are too slow to be approximated by either a single step or a single ramp, without seriously distorting the original MCM behavior.

Slow sigmoids can be accurately approximated with sequences of ramps. This, however, raises a new question: how to choose as few ramps as possible, while still maintaining a desired approximation error? Additionally, since each successive pair of ramp thresholds introduces a new mode in the

```
function [e,a,b,xb] = optimalLinearApproximation(x,y,S)
Input:
   x,y:  Curves given as an x-points vector and a vector of y-points vectors
     S:  Number >= 2 of desired segments
Output:
     e:  Errors matrix
   a,b:  Line-segment-coefficients matrix
    xb:  x-coordinate at breaking point matrix
Initialization
z1  = size(x);  P = z1(2);   Get number of points in each curve
z2  = size(y);  C = z2(1);   Get number of digitized curves
se = zeros(1,C);             Initialize vector of errors, one error for each curve
Cost tables
cost  = ones(P,S) * inf;  cost(30,4) = min cost to pt 30 with 4-segm polyline
error = ones(P,P) * inf;  error(i, n)  = cached error of line segment (i,n)
cost(2,1) = 0;            1-segment-polyline cost of polyline (1,2) = 0
Predecessor table
father = ones(P,S) * inf;  father(30,4) = pred of pt 30 on a 4-segm polyline
Computation of optimal segmentation
Initialize cost and father for 1-segment-polyline, from pt 1 to all other pts
for p = 2:P                Traverse all  other points
   for c = 1:C               Traverse all curves
      se(c) = segmentError(x(1:p), y(c,1:p));  (1,p)-line-segment appr error
   end; for c
   cost(p,1) = max(se);    Maximum error among all curves
   father(p,1) = 1;        All 1-segment polylines have father point 1
end; for p
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 (error(i,p) == Inf)  Error of line segment (i,n) not cashed
        for c = 1:C           Next curve-number to consider
          se(c) = segmentError(x(i:p), y(k,i:p)); (i,p)-segment error
        end; for k
        error(i,p) = max(se);  Maximum line segment error
      end; if
      currErr = cost(i,s-1) + error(i,p);  s-segment-polyline error
      if (currErr < minErr)                Smaller error?
        minErr = currErr; minIndex = i;  Update error and parent
      end; if
    end; for i
    cost(p,s) = minErr;         s-segment-polyline minimal cost
    father(p,s) = minIndex;   Last point's father on the polyline
  end; for p
end; for s
[e,a,b,xb] = ExtractAnswer;
end
```

```
function [e,a,b] = segmentError(x,y)
Input:
   x,y:  Digitized curve-segment as an x-vector and an y-vector
Output:
     e:  Error of the line segment between the first and last point
   a,b:  The coeficients defining this segment
Initialization:
z = size(x); P = s(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
e = 0;           Initialize Error
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
function [e,a,b,xb] = ExtractAnswer
Output:
     e,a,b,xb:  As in the output of optimalLinearApproximation
Initialization:
ib = zeros(S,S+1); xb = zeros(S,S+1);  yb = zeros(C, S,S+1);  Points matrices
a = zeros(C, S,S);   b = zeros(C, S,S);   er = zeros(C, S,S); Coefficients/error
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
   for c = 1:C      Traverse all curves
      yb(c,s,s+1) = y(c,ib(s,s+1));  Get y-value for this point
   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 for segm (x,y)
            segmentError( x(ib(s,i):ib(s,i+1)), y(c,ib(s,i):ib(s,i+1)) );
      end
   end;
end;
end
```

HA, how to choose the *same thresholds* for all sigmoids?

In the following, we show that all of these goals are achievable; i.e., there is an optimal solution, to the slow-sigmoid approximation problem, which minimizes a given approximation error in a global way. Our approach is based on and extends a dynamic programming algorithm developed in the computer graphics community for approximating digitized polygonal curves [16] with minimal error. The MATLAB code for the main function, optimalLinearApproximation, is shown on previous page, where for readability, comments are displayed in green.

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.

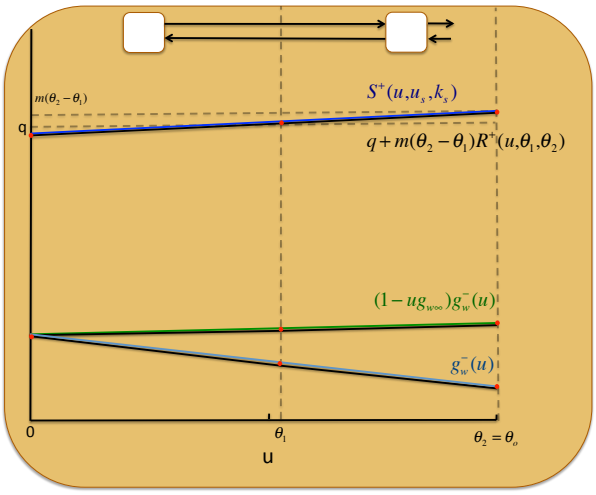


Fig. 7. Linearization of the orange mode of the MCM HA.

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 error(P,P), such that each entry error(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 error(i,p+1).

The value stored in error(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 $y(x) = \mathrm{a}x + \mathrm{b}$ 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

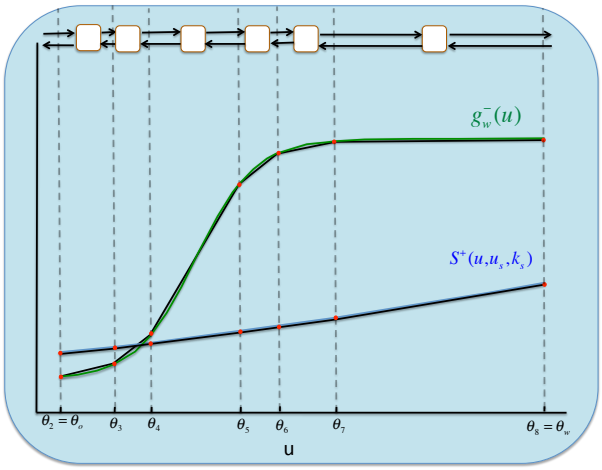


Fig. 8. Linearization of the blue mode of the MCM HA.

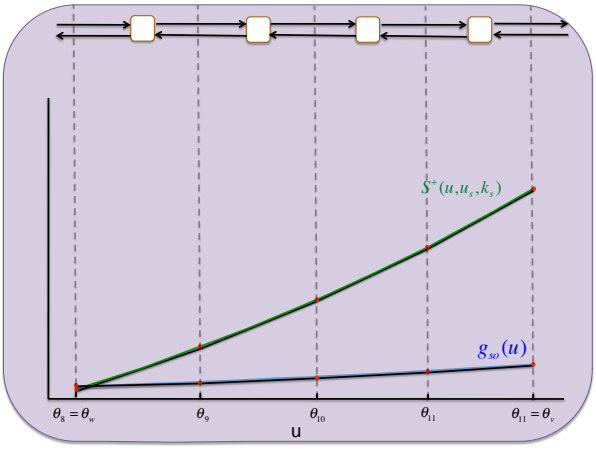


Fig. 9. Linearization of the pink mode of the MCM HA.

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 choose as new breaking points 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 steps and sigmoidal switches are known and fixed. This is a reasonable assumption, as these parameters can be accurately identified from simulation or experimental data. We can thus linearize the MCM HA one mode at a time. The result is a *multiaffine hybrid automaton* (MHA).

Figure 7 presents our linearization of the orange mode of the MHA. There are 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.

Figure 8 presents our linearization of the blue mode of the MHA. There are two nonlinear functions: sigmoids $g_w^-(u)$ and $S^+(u,u_s,k_s)$. 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

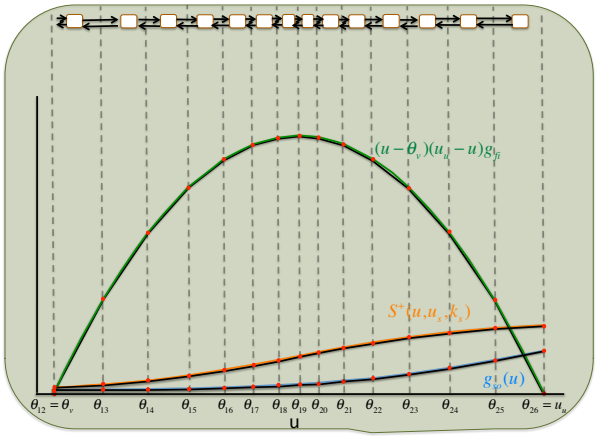


Fig. 10. Linearization of the green mode.

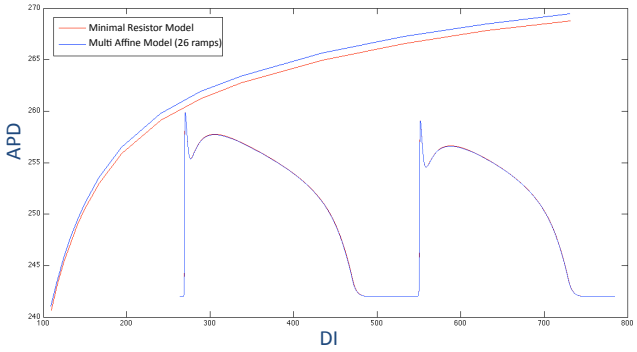


Fig. 11. Comparison of the AP shape and restitution in cable.

behavior to the linearization error is also very important.

Figure 9 presents our linearization of the pink mode of the MHA. There are two nonlinear functions: sigmoids $g_{so}(u)$ and $S^+(u,u_s,k_s)$. A four-segment linearization (four modes in the MHA) achieves a small enough approximation error.

Finally, Figure 10 presents our linearization of the green mode of the MHA. This is the most sophisticated mode. It contains three nonlinear functions: sigmoids $g_{so}(u)$ and $S^+(u,u_s,k_s)$ and the parabolic term $(u-\theta_v)(u_u-u)$. 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 quite costly, but inevitable linearization via 14 segments (14 modes in the MHA). In our experiments, fewer segments have lead to an unacceptable approximation of the MCM behavior.

To asses the accuracy of the MHA, we performed extensive 1D and 2D simulations, in a cable of 100 cells and a grid of $800\times800$ 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 [2], accurately reproduce the AP when simulated in isolation, but fail to reproduce the desired behavior in a cable.

Figure 11 shows the restitution curves of the MCM and MHA models. Each point APD($d$) on these curves, was obtained by first pacing the MCM and the MHA models at

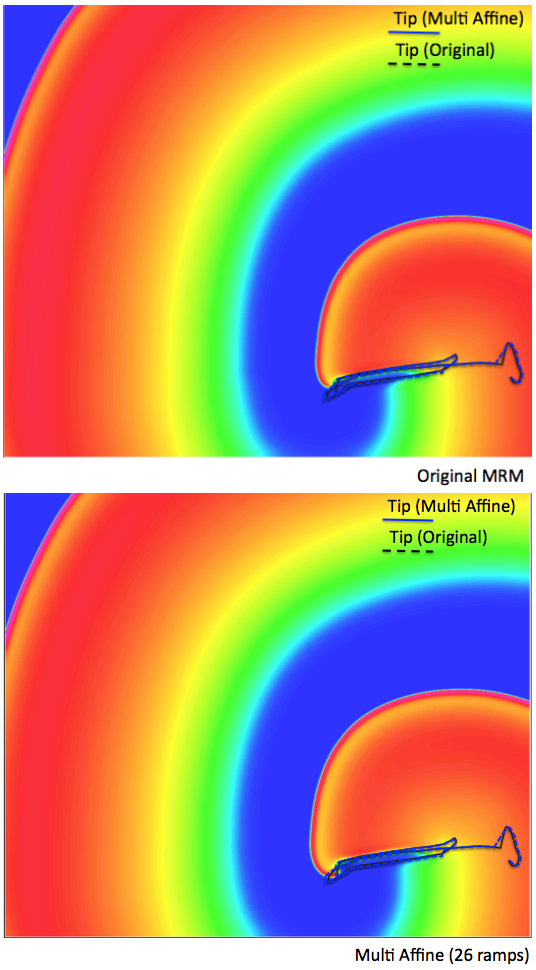


Fig. 12. Spiral wave and tip movement comparison in an $800 \times 800$ grid.

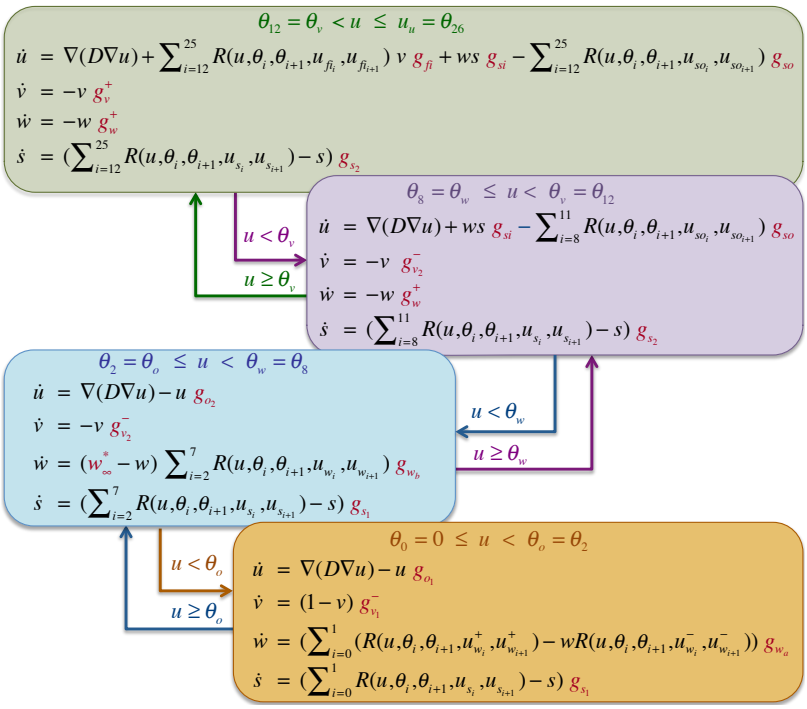


Fig. 13. The multiaffine hybrid automaton.

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($d$). One such comparison is given in Figure 11. In both cases (restitution and AP shape), the MHA approximated the MCM with sufficient accuracy.

In Figure 12, we compare the behavior of the MCM and MHA models on a 2D grid of $800 \times 800$ 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 in Figure 12. The 2D simulation also confirms the very good accuracy of the MHA model.

The simulations where performed on a CUDA PC equipped with four GPU processors. *****EZIO SHOULD COMPLETE THE SETUP DESCRIPTION***** In this setting, we observed a 1.43 speedup in MHA simulation times compared to MCM. Note that it is possible to table sigmoid and parabola values to speedup MCM simulation times [18]. 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.

## VI. Parameter-Range Identification

The linearization algorithm presented in Section V, returns for each mode $[\theta_1, \theta_2)$, the parameter sequences $\mathrm{a}_i$ and $\mathrm{b}_i$, and the threshold sequence $\mathrm{x}_i$. Subscript $i$ ranges over the number of segments chosen, in order to fulfil a desired approximation error $e$. For each $i$, the returned values define a line segment $y(x) = \mathrm{a}x + \mathrm{b}$ 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 $y_i = \mathrm{a}_i x_i + \mathrm{b}_i$ and the maximum value of $y(x)$ is $y_{i+1} = \mathrm{a}_i x_{i+1} + \mathrm{b}_i$. 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 have to be summed up, for each $i > 1$, we have to 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 Figure 13. The remaining parameters of the MHA are now highlighted in red color. The MHA modes have become super-modes, each consisting of as many sub-modes as there are distinct indices in the sums.

The MHA however, is not suitable as input to Rovergene for two reasons: 1) Rovergene expression terms have to 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 $\theta_i$, separating modes $[\theta_{i-1}, \theta_i)$ and $[\theta_i, \theta_{i+1})$, a just-before $\theta_i$ threshold

$\theta_i^-$. 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:

$$\begin{aligned}
\dot{u} = {} & e - R^-(u,\theta_o^-,\theta_o)R^+(u,0,\theta_o^-,0,\theta_o^-)\,g_{o_1} - \\
& R^+(u,\theta_o^-,\theta_o)R^-(u,\theta_w^-,\theta_w)R^+(u,\theta_o,\theta_w^-,\theta_o,\theta_w^-)\,g_{o_2} + \\
& R^+(u,\theta_w^-,\theta_w)R^+(s,0,1)R^+(w,0,1)\,g_{si} - \\
& R^+(u,\theta_w^-,\theta_w)\textstyle\sum_{i=8}^{25} R(u,\theta_i,\theta_{i+1},u_{so_i},u_{so_{i+1}}) + \\
& R^+(u,\theta_o,\theta_v)R^+(v,0,1) \\
& \quad \textstyle\sum_{i=12}^{25} R(u,\theta_i,\theta_{i+1},u_{fi_i},u_{fi_{i+1}})\,g_{fi} \\
\dot{v} = {} & R^-(u,\theta_o^-,\theta_o)R^-(v,0,1)\,g_{v_1}^- - \\
& R^+(u,\theta_o^-,\theta_o)R^-(u,\theta_o,\theta_v)R^+(v,0,1)\,g_{v_2}^- - \\
& R^+(u,\theta_o,\theta_v)\textstyle\sum_{i=12}^{25} R^+(v,0,1)\,g_v^+ \\
\dot{w} = {} & R^-(u,\theta_o^-,\theta_o)\textstyle\sum_{i=0}^{1}(R(u,\theta_i,\theta_{i+1},u_{w_i}^+,u_{w_{i+1}}^+) - \\
& \quad R(w,0,1)R(u,\theta_i,\theta_{i+1},u_{w_i}^-,u_{w_{i+1}}^-))\,g_{w_a} + \\
& R^+(u,\theta_o^-,\theta_o)R^-(u,\theta_w^-,\theta_w) \\
& \quad (w_\infty^* - R(w,0,1))\textstyle\sum_{i=2}^{7} R(u,\theta_i,\theta_{i+1},u_{w_i}^+,u_{w_{i+1}}^+)\,g_{w_b} - \\
& R^+(u,\theta_w^-,\theta_w)R(w,0,1)\,g_w^+ \\
\dot{s} = {} & (R^-(u,\theta_w^-,\theta_w)\,g_{s_1} + R^+(u,\theta_w^-,\theta_w)\,g_{s_2}) \\
& \quad \textstyle\sum_{i=0}^{25} R^+(u,\theta_i,\theta_{i+1},u_{s_i},u_{s_{i+1}}) - R^+(s,0,1)
\end{aligned}$$

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

As discussed in Section I, a biologically relevant question that we would like to answer is, under what circumstances a cell may loose excitability? At 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 above question, as a parameter-range identification question: What are the parameter ranges for which the above MEM fails to generate an AP?

This property may be specified in linear temporal logic (LTL) as follows: G $(u<\theta_v)$, where G is the LTL globally (always) temporal operator. The property states that in all executions of the DEM (an implicit universal quantification of LTL over executions) and in all moments of time (an explicit quantification in LTL with operator G, over all states of a single execution), the voltage value is below $\theta_v$. We would like this property to hold for all stimulus durations. In terms of the MCM HA in Figure 13, 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 automatic fashion we use the Rovergene tool [13] co-developed by one of the co-authors (Batt) with input the above MEM and LTL formula, and initial region: $u\in[0,\theta_1]$, $0\in[0.95,1]$, $w\in[0.95,1]$ and $s\in[0,0.01]$.

The $u$-thresholds and the initial region impose the following partition on the ranges of state variables: $[0,\ldots,\theta_{29}]$ for $u$ (we have added the just-before thresholds), $[0,0.95,1]$ for $v$ and $w$ and $[0,0.01,1]$ for $s$. The parameter ranges with biological

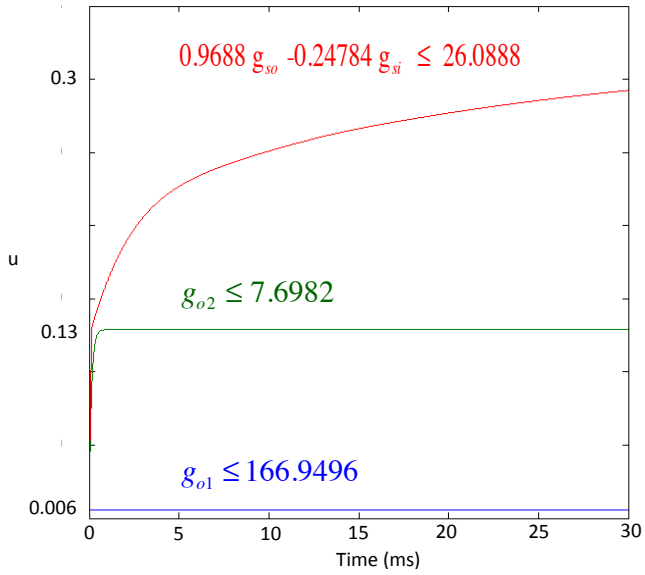


Fig. 14. Simulation results for a sample in each of the above ranges.

significance were taken to be: $[1,180]$ for $g_{o_1}$, $[0,10]$ for $g_{o_2}$, $[0.1,100]$ for $g_{si}$ and $[0.9,50]$ for $g_{so}$.

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 neighbours, 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 these ranges returned are:

$$\begin{aligned}
& 166.9494 \le g_{o_1} \le 180, \quad 7.6982 \le g_{o_1} \le 10 \\
& -0.24784\,g_{si} + 0.9688\,g_{so} \le 26.0888
\end{aligned}$$

They have the following meaning. If $g_{o_1}\ge 166.9494$ then no matter how long a stimulus of magnitude 1 is 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 (note that time is abstracted away by Rovergene), once $u$ enters the blue range, its behavior is completely determined by this mode. If $g_{o_2}\ge 7.6982$ 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 Figure 14. To make sure 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 simulation tool proved to be an invaluable debugging tool, during the model encoding in Rovergene.

## VII. Conclusions and Future Work

Although formal techniques were used before to analyze cardiac-cell properties (see e.g. our work in [19], [20]), this paper presents to the best of our knowledge, for the first time, an approach to automatically identify the parameter ranges of a biologically-relevant cardiac model, guaranteeing that the model accurately reproduces a particular cardiac disorder.

The approach takes the nonlinear cardiac-model in [15], brings it first into a genetic regulatory network sigmoidal form, then linearizes and formats it to a piecewise-multiaffine set of differential equations, and finally leverages the tools previously developed for the automatic parameter-range identification in genetic regulatory networks [13], to automatically check a disorder expressed in linear temporal logic.

In particular, the property we studied 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 singles 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. For example, action potential duration or spiral breakup (fibrillation) are such properties. In future work we therefore plan to investigate novel parameter-range identification approaches, that use timed-automata[21], [22] or linear-automata[23], [24] abstractions instead.

## References


[1] E. Robles de Medina, R. Bernard, P. Coumel, A. Damato, F. C., D. Krikler, N. Mazur, F. Meijler, L. Mogensen, P. Moret, Z. Pisa, and H. Wellens, "Definition of terms related to cardiac rhythm. WHO/ISFC task force." *European Journal of Cardiology*, vol. 8, no. 2, pp. 127–144, 1978.

[2] 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*, vol. 74, no. 6, pp. 1071–1096, Jun. 1994.

[3] L. Priebe and D. Beuckelmann, "Simulation study of cellular electric properties in heart failure," *Circulation Research*, no. 82, pp. 1206–1223, 1998.

[4] K. Ten Tusscher, D. Noble, P. Noble, and A. Panfilov, "A model for human ventricular tissue," *American Journal of Physiology*, no. 286, pp. H1573–H1589, 2004.

[5] V. Iyer, R. Mazhari, and R. Winslow, "A computational model of the human leftventricular epicardial myocytes," *Biophysical Journal*, no. 87, pp. 1507–1525, 2004.

[6] C.M. and S. Orszag, *Advanced Mathematical Methods for Scientists and Engineers: Asymptotic Methods and Perturbation Theory*. Springer Verlag, 1999.

[7] L. Michaelis and M. Menten, "Die kinetik der invertinwirkung," *Biochemische Zeitschrift*, no. 49, pp. 333–3369, 1913.

[8] L. Segel and M. Slemrod, "The quasi-steady-state assumption: A case study in perturbation," *SIAM Review*, vol. 31, no. 3, pp. 446–477, 1989.

[9] G. Yagil and E. E. Yagil, "On the relation between effector concentration and the rate of induced enzyme synthesis," *Biophysical Journal*, vol. 11, no. 1, pp. 11–27, 1971.

[10] J. Keener and J. James Sneyd, *Mathematical Physiology*. Springer, 2008.

[11] L. Glass and S. S. A. Kauffman, "The logical analysis of continuous nonlinear biochemical control networks," *Journal of Mathematical Biology*, vol. 39, no. 1, pp. 103–129, 1973.

[12] T. T. Mestl, E. Plahte, and S. Omholt, "A mathematical framework for describing and analysing gene regulatory networks," *Journal of Theoretical Biology*, vol. 176, no. 2, pp. 291–300, 1995.

[13] G. Batt, C. Belta, and R. Weiss, "Temporal logic analysis of gene networks under parameter uncertainty," *IEEE Transactions of Automatic Control*, vol. 53, pp. 215–229, 2008.

[14] H. de Jong, "Modeling and simulation of genetic regulatory systems: A literature review," *Journal of Computational Biology*, vol. 9, no. 1, pp. 69–105, 2002.

[15] A. Bueno-Orovio, M. Cherry, and F. Fenton, "Minimal model for human ventricular action potentials in tissue," *Journal of Theoretical Biology*, no. 253, pp. 544–560, 2008.

[16] J. Perez and E. Vidal, "Optimum polygonal approximation of digitized curves," *Pattern Recognition Letters*, no. 15, pp. 743–750, 1994.

[17] 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.*, no. 19, pp. 1244–1253, 1992.

[18] 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*, no. 64, pp. 73–96, 2002.

[19] 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*, ser. LNBI. Rostock, Germany: Springer Verlag, October 2008.

[20] 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)*, vol. 52, no. 3, pp. 1–10, March 2009.

[21] R. Alur and D. Dill, "A theory of timed automata," *Theoretical Computer Science*, vol. 2, no. 126, pp. 183–235, Apr. 1994.

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

[23] 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*, ser. Lecture Notes in Control and Information Sciences 199. Springer-Verlag, 1994, pp. 331–351.

[24] G. Frehse, "PHAVer: algorithmic verification of hybrid systems past HyTech," *Springer International Journal of Software Tools fot Technology Transfet*, vol. 10, no. 3, pp. 263–279, 2008.