Computers in Biology and Medicine 81 (2017) 55–63

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Fitting local repolarization parameters in cardiac reaction-diffusion models in the presence of electrotonic coupling

Akshay Kota Aswath Kumar[a,b], Angelina Drahi[a,b], Vincent Jacquemet[a,b,*]

[a] *Université de Montréal, Département de Pharmacologie et Physiologie , Institut de Génie Biomédical, Montréal, Canada*
[b] *Hôpital du Sacré-Coeur de Montréal, Centre de Recherche, Montréal, Canada*

ARTICLE INFO

Keywords:
Computer modeling
Cardiac electrophysiology
Parameter estimation
Action potential duration
Cell coupling

ABSTRACT

Background: Repolarization gradients contribute to arrhythmogenicity. In reaction–diffusion models of cardiac tissue, heterogeneities in action potential duration (APD) can be created by locally modifying an intrinsic membrane kinetics parameter. Electrotonic coupling, however, acts as a confounding factor that modulates APD dispersion.

Method: We developed an algorithm based on a quasi-Newton method that iteratively adjusts the spatial distribution of a membrane parameter to reproduce a pre-defined target APD map in a coupled tissue. The method assumes that the relation between the adjustable parameter and APD is bijective in an isolated cell. Each iteration of the algorithm involved simulating the cardiac reaction–diffusion system with the updated parameter profile for one beat and extracting the APD map. The algorithm was extended to simultaneous estimation of two parameter profiles based on two APD maps at different repolarization thresholds.

Results: The method was validated in 1D, 2D and 3D atrial tissues using synthetic target APD maps with controllable total variation and maximum APD gradient. The adjustable parameter was local acetylcholine concentration. The iterations converged provided that APD gradients were not too steep. Convergence was found to be faster 2–5 iterations) when the maximal gradient was less steep, when APD range was smaller and when tissue conductivity was reduced.

Conclusion: This algorithm provides a tool to automatically generate arrhythmogenic substrates with controllable repolarization gradients and possibly incorporate experimental APD maps into computer models.

1. Introduction

The presence of strong repolarization gradients in a cardiac tissue is an arrhythmogenic factor that promotes wave breaks and reentry [1–4]. The occurrence of functional block has been observed in the presence of action potential duration (APD) gradients above a critical value of the order of 2–12.5 ms/mm [5–8]. Dispersion of action potential duration (APD) may result from intrinsic spatial variations in ion channel density (notably aggravated by the remodeling induced by successive episodes of arrhythmia), from beat-to-beat variability in repolarization eventually exhibiting non-linear dynamics and chaos [9], or from the interplay between geometry, conduction properties, wavelet dynamics [10,11], and mechano-electric feedback [12,13].

Electrotonic currents flowing through gap junctions tend to reduce the differences in APD between neighboring cells [14–16]. As a result, APD measurements in an intact tissue may not exactly reflect the intrinsic local properties of the cells, but rather an average over a surrounding region whose size and shape depends on conduction properties [17,18]. Determination of true intrinsic membrane properties may be obtained through biopsies followed by patch clamp experiments. This approach is however limited in terms of spatial resolution, creates damage to the tissue and possibly changes the dynamics and densities of ionic currents, resulting in an APD that may differ from the APD that would have been measured in situ. Techniques such as electrical stimulation, monophasic action potentials and optical mapping preserve the integrity of the tissue (to some extent), but the resulting APD maps are affected by electrotonicity. Thus, the relationship between measured APD and intrinsic APD is relevant to the non-destructive extraction of cellular intrinsic properties.

In computer models of cardiac arrhythmia, the incorporation of APD dispersion requires designing a spatial profile of intrinsic properties of cardiac cells. Typically, a membrane kinetics parameter is chosen as target and its spatial distribution is used as an input to the model [19,20]. The question arises whether that parameter distribution can be determined from an APD map in the coupled tissue. The existence and uniqueness of the solution has been investigated in a

[*] Correspondence to: Hôpital du Sacré-Coeur de Montréal, Centre de Recherche, 5400 boul. Gouin Ouest, Montreal, Quebec, Canada H4J 1C5.
E-mail address: vincent.jacquemet@umontreal.ca (V. Jacquemet).

http://dx.doi.org/10.1016/j.compbiomed.2016.12.008
Received 27 September 2016; Received in revised form 15 November 2016; Accepted 14 December 2016
0010-4825/ © 2016 Elsevier Ltd. All rights reserved.

simplified model with exponentially-shaped action potentials [21]. Hurtado et al. calibrated a ventricular model to reproduce the relation between activation time and a refractoriness parameter [22]. Defauw et al. proposed a Gaussian Green's function model and a deconvolution approach to estimate the intrinsic APD map [17]. Inspired by their approach, we hypothesized that knowledge about which specific membrane kinetics parameter causes APD variations would enable the development of more accurate methods.

In this paper, we propose an algorithm for iteratively computing the parameter distribution that reproduces a target APD map, based on an idea initially sketched in [21] and tested in [23]. The implementation is described and extensively validated in atrial tissue models with increasing complexity, and its computational performance and accuracy are evaluated.

2. Methods

2.1. Problem statement

In the framework of a monodomain model of cardiac tissue, let us consider that the membrane model depends on a local parameter k that lies within a physiological range $[k_{\min}, k_{\max}]$. This parameter could be an ion channel conductance, an ionic concentration, or a normalized parameter describing the transition between normal and diseased tissue, but is assumed not to affect intercellular coupling (gap junction conductances). After spatial discretization, tissue configuration is described by a vector $\mathbf{k}$ whose size is the number of nodes in the mesh.

When the spatial distribution of k is non-uniform, the simulated APD map is also non-uniform. The forward problem then consists in computing the APD map ($\mathbf{a}$) as a function of $\mathbf{k}$

$$\mathbf{a} = \mathbf{a}_{\text{forw}}(\mathbf{k}; G). \tag{1}$$

Because of electrotonicity, APD distribution depends not only on $\mathbf{k}$ but also on the intercellular coupling matrix G. Practically, the function $\mathbf{a}_{\text{forw}}$ was evaluated by running a monodomain simulation with the distribution of parameters set to $\mathbf{k}$ and by measuring the APD map. Specific simulation methods are described in Section 2.5.

Assuming that the coupling is known, the inverse problem consists in recovering the parameter distribution $\mathbf{k}$ that would reproduce a given APD map $\mathbf{a}_{\text{target}}$

$$\mathbf{k} = \mathbf{a}_{\text{forw}}^{-1}(\mathbf{a}_{\text{target}}; G) \tag{2}$$

provided that the solution exists and is unique for a given stimulation protocol, i.e. $\mathbf{a}_{\text{forw}}$ is invertible. Uniqueness of the solution has been proved in a simple analytical model [21] and is also guaranteed if the APD map in the coupled system can be written as the convolution of the intrinsic APD map with a spatial (e.g. Gaussian) filter [17].

2.2. Parameter identification

The inverse problem is equivalent to solving the equation $\mathbf{a}_{\text{forw}}(\mathbf{k}) - \mathbf{a}_{\text{target}} = 0$. Our approach relies on the fact that the problem is easily solved when cells are uncoupled (G=0). A first approximation $\mathbf{k}^{(0)}$ is obtained by neglecting electrotonicity:

$$\mathbf{k}^{(0)} = \mathbf{a}_{\text{forw}}^{-1}(\mathbf{a}_{\text{target}}; 0). \tag{3}$$

Then, at iteration n, the parameter profile is updated using the quasi-Newton formula

$$\mathbf{k}^{(n+1)} = \mathbf{k}^{(n)} - (\mathbf{Da}_{\text{forw}}(\mathbf{k}^{(n)}; 0))^{-1} \cdot (\mathbf{a}_{\text{forw}}(\mathbf{k}^{(n)}; G) - \mathbf{a}_{\text{target}}), \tag{4}$$

where the Jacobian $\mathbf{Da}_{\text{forw}}(\mathbf{k}^{(n)}; G)$ has been approximated by the (diagonal) Jacobian in the uncoupled tissue $\mathbf{Da}_{\text{forw}}(\mathbf{k}^{(n)}; 0)$ to avoid expensive computations. The Jacobian in the coupled tissue is indeed a fully-populated matrix. The diagonal approximation, which is reminiscent of mass lumping in finite element methods, guarantees that the inverse exists. Moreover, if the simulated local APD is shorter than the target APD, the local parameter k will be updated to increase the intrinsic APD, therefore increasing the updated simulated APD provided that this effect is not compensated by the neighboring cells. This overcompensation will not occur as long as the APD error is a smooth function of space. The iteration process stops when the error $\| \mathbf{a}_{\text{forw}}(\mathbf{k}^{(n)}) - \mathbf{a}_{\text{target}} \|$ falls below a tolerance, typically 1 ms.

2.3. Computational issues

As a preprocessing step, the relation $a = \alpha(k)$ between the parameter k and the APD (a) was studied in an isolated cell. The function α was evaluated (using simulations) at n=8 equally-spaced points in the interval $[k_{\min}, k_{\max}]$. The number of points was then iteratively increased until the maximal error between spline interpolation based on the previous iteration and the new computed data points fell below a threshold, typically 0.5 ms. This provided a piece-wise polynomial interpolation for the function $\alpha(k)$. The monotonicity of $\alpha(k)$ was checked using the coefficient of the polynomials. Spline interpolation on the same data points (reflected across the diagonal) was used to compute the inverse function $k = \alpha^{-1}(a)$. The derivative $\alpha'(k)$ was obtained by analytically differentiating the piece-wise polynomial in each of its segments. To avoid out-of-bound errors, when the argument of the function is out of the domain or the range of α, the value at the bound is returned.

With these notations, we have:

$$\mathbf{a}_{\text{forw}}^{-1}(\mathbf{a}_{\text{target}}; 0) = \alpha^{-1}(\mathbf{a}_{\text{target}}) \tag{5}$$

$$(\mathbf{Da}_{\text{forw}}(\mathbf{k}; 0))^{-1} = \text{diag}(\alpha'(\mathbf{k}))^{-1}, \tag{6}$$

where the functions α^{-1} and α' are applied element-wise and ‘diag’ creates a diagonal matrix from a diagonal vector.

The algorithm was implemented in Matlab on a Linux machine. At each iteration, the Matlab function writes a parameter file, calls an external program to run the simulation, reads the output and continues the execution in Matlab.

2.4. Extension to two parameters

If the membrane model depends on two parameters k and m, two measures of repolarization $a = \alpha(k, m)$ and $b = \beta(k, m)$ are needed for parameter identification. They may represent APD at different repolarization thresholds or at different heart rates. Assuming that the system $a = \alpha(k, m)$ and $b = \beta(k, m)$ has a unique solution in a domain Ω, the inverse solution may be denoted by $k = \alpha^{-1}(a, b)$ and $m = \beta^{-1}(a, b)$.

In a coupled tissue where the two measures $\mathbf{a}_{\text{forw}}(\mathbf{k}, \mathbf{m})$ and $\mathbf{b}_{\text{forw}}(\mathbf{k}, \mathbf{m})$ can be simulated, a first estimate of the parameter vectors $\mathbf{k}$ and $\mathbf{m}$ that solve the inverse problem $\mathbf{a}_{\text{target}} = \mathbf{a}_{\text{forw}}(\mathbf{k}, \mathbf{m})$ and $\mathbf{b}_{\text{target}} = \mathbf{b}_{\text{forw}}(\mathbf{k}, \mathbf{m})$ is obtained by

$$\mathbf{k}^{(0)} = \alpha^{-1}(\mathbf{a}_{\text{target}}, \mathbf{b}_{\text{target}}) \tag{7}$$

$$\mathbf{m}^{(0)} = \beta^{-1}(\mathbf{a}_{\text{target}}, \mathbf{b}_{\text{target}}). \tag{8}$$

Then, at iteration n the update formula reads

$$\begin{pmatrix} \mathbf{k}^{(n+1)} \\ \mathbf{m}^{(n+1)} \end{pmatrix} = \begin{pmatrix} \mathbf{k}^{(n)} \\ \mathbf{m}^{(n)} \end{pmatrix} - J(\mathbf{k}^{(n)}, \mathbf{m}^{(n)})^{-1} \cdot \begin{pmatrix} \mathbf{a}_{\text{forw}}(\mathbf{k}^{(n)}, \mathbf{m}^{(n)}) - \mathbf{a}_{\text{target}} \\ \mathbf{b}_{\text{forw}}(\mathbf{k}^{(n)}, \mathbf{m}^{(n)}) - \mathbf{b}_{\text{target}} \end{pmatrix} \tag{9}$$

where the Jacobian is approximated by

$$J(\mathbf{k}, \mathbf{m}) = \begin{pmatrix} \text{diag}((\partial_k \alpha)(\mathbf{k}, \mathbf{m})) & \text{diag}((\partial_m \alpha)(\mathbf{k}, \mathbf{m})) \\ \text{diag}((\partial_k \beta)(\mathbf{k}, \mathbf{m})) & \text{diag}((\partial_m \beta)(\mathbf{k}, \mathbf{m})) \end{pmatrix} \tag{10}$$

where the functions $\partial_k \alpha$, etc. are applied element-wise. The inverse of J is easily computed thanks to its 2-by-2 diagonal block structure:

Download English Version:

https://daneshyari.com/en/article/4964837

Download Persian Version:

https://daneshyari.com/article/4964837

Daneshyari.com