

Department of Computer Science

**Estimation and Verification of Hybrid Heart Models for
Personalised Medical and Wearable Devices**

**Benoît Barbot, Marta Kwiatkowska, Alexandru Mereacre
and Nicola Paoletti**

CS-RR-15-05



Department of Computer Science, University of Oxford
Wolfson Building, Parks Road, Oxford, OX1 3QD

Abstract. We are witnessing a huge growth in popularity of wearable and implantable devices equipped with sensors that are capable of monitoring a range of physiological processes and communicating the data to smartphones or to medical monitoring devices. Examples include fitness bands that can monitor the heartbeat and estimate the time you spend in deep sleep, wristbands that authenticate users based on the unique shape of their electrocardiogram (ECG), implantable glucose sensors that send measurements to an insulin pump, and leadless cardiac pacemakers that are implanted inside the heart. Applications include not only medical diagnosis and treatment, but also biometric authentication systems. An important requirement is personalisation of the devices, namely, their ability to adapt to the physiology of the human wearer and to faithfully reproduce the characteristics in real-time for the purposes of authentication or optimisation of medical therapies. In view of the complexity of the embedded software that controls such devices, model-based frameworks have been advocated for their design, development, verification and testing. In this paper, we focus on applications that exploit the unique characteristics of the heart rhythm. We introduce a hybrid automata model of the electrical conduction system of a human heart, adapted from Lian et al [22], and present a framework for the estimation of personalised parameters, including the generation of synthetic ECGs from the model. We demonstrate the usefulness of the framework on two applications, ensuring safety of a pacemaker against a personalised heart model and ECG-based user authentication. The paper concludes by discussing the challenges and opportunities in this field.

1 Introduction

Recent technological advances have spurred a huge growth in apps and wearables for use in health monitoring. They employ a multiplicity of noninvasive sensors, e.g. accelerometers and miniature cameras, that can read physiological indicators, wirelessly send data to smartphones and analyse it not only to record trends, but also to support decision making for diagnosis and intervention. Examples include fitness bands, such as Jawbone UP3¹, which can count steps, monitor the heartbeat, and estimate the calories intake and the time you spend in deep sleep. The success in miniaturisation of electronics has led to novel variants of traditional medical devices being introduced on the market, to mention leadless cardiac pacemakers that can be implanted inside the human heart (e.g. Nanostim²) and implantable glucose monitors that transmit data to a wristwatch to alert the wearer about any undesirable trends (e.g. Minimed³). Applications are not limited to the medical field, and include also emerging technologies for biometric user authentication and security, such as wristbands

¹ <https://jawbone.com/store/buy/up3>

² <http://www.sjm.com/leadlesspacing/intl/options/leadless-pacing>

³ <http://www.medtronicdiabetes.com/treatment-and-products/minimed-revel-insulin-pump>

that periodically check the electrocardiogram (ECG) of the user to produce a template authentication signal (e.g. the Nymi band⁴).

An important requirement for wearables is their personalisation, namely, the ability for the device to adapt to the physiology of the human wearer based on the person's individual characteristics. This is viewed as a key enabler of personalised healthcare, achieved through automation of medical intervention strategies uniquely adapted to the individual. Personalisation is typically implemented via an appropriate parameterisation of a model of the physiological process, through parameter estimation and parameter synthesis techniques. Automation of personalised delivery of medical treatment is a major challenge; for example, rate-adaptive pacemakers are able to vary the rate of pacing depending on the activity and age of the patient [18], but insulin pumps still rely on human supervision. Another important role of personalised devices is in device safety assurance, where they can be used to faithfully reproduce the unique characteristics of the wearer in real-time for the purposes of testing.

Undoubtedly, personalised medical wearable and implantable devices are an important step towards achieving personalised healthcare. However, major advances are necessary to realise this vision, ranging from technological (miniaturisation, low-power circuits), software technologies (design automation, code generation, integration), to regulatory and legal frameworks (FDA approval, certification). This paper is concerned with model-based design and verification techniques for ensuring safety and effectiveness of personalised devices based on the bioelectrical activity of the heart.

We focus on the hybrid automata framework for closed-loop quantitative verification of cardiac pacemakers introduced in [9,19], where a pacemaker is modelled as a timed automaton, composed in closed loop with a heart model represented as a hybrid automaton. The heart and pacemaker models are provided in Simulink, and their safety and energy efficiency is analysed for a range of heart conditions, including chronotropic deficiency that can be treated with the rate-adaptive pacemaker. This was extended in [11] with techniques to automatically synthesise optimal timing delays to minimise energy consumption, and in [4] with a hardware-in-loop simulator to evaluate embedded pacemaker software on low-power hardware similar to that of a real pacemaker. However, personalisation was not supported.

In this paper, we extend the framework of [9,19] as follows. We introduce a new hybrid heart model, an adaptation of the model of Lian et al [22], encoded in Simulink/Stateflow and develop techniques to personalise the model through parameter estimation based on ECG data. We implement methods to produce synthetic ECGs that are characteristic for the given individual, and also to compare different ECG patterns. We consider two applications: verification of safety properties for a pacemaker against a personalised heart model, and biometric authentication based on matching the wearer's signature with ECG data acquired for recognition.

⁴ <https://www.nymi.com/>

2 Background

In this section, we introduce our main modelling formalism, *timed I/O automata with priorities and data* [20,3], which extend the model of [11] with data variables, and show how to equip it with probabilistic dynamics, which we will need in the parameterization of the heart model from ECG data. The hybrid dynamics of the electrical conduction of a human heart is relatively simple, and in fact can be modelled using timed automata with data. To model the conduction, we use data variables in the guards of the automaton and, each time an edge is taken, we compute the valuation that the data variables are updated to in the target location. Priorities are needed to impose determinism.

2.1 Timed I/O automata with priorities and data

Let \mathcal{X} be a set of *non-negative* real-valued variables, called *clocks*. Let \mathcal{D} be a set of real-valued variables, called *data*. A variable valuation is a function $\eta = \eta_{\mathcal{X}} \cup \eta_{\mathcal{D}}$, where $\eta_{\mathcal{X}} : \mathcal{X} \rightarrow \mathbb{R}_{\geq 0}$ and $\eta_{\mathcal{D}} : \mathcal{D} \rightarrow \mathbb{R}$. We denote the set of variables with $V = \mathcal{X} \cup \mathcal{D}$. Let Γ be a set of real-valued *parameters*. A parameter valuation is a function $\gamma : \Gamma \rightarrow \mathbb{R}$ mapping each parameter p to a value in its domain $\text{dom}(p) \subseteq \mathbb{R}$.

Let \mathcal{Y} be a set and $\mathcal{V}(\mathcal{Y})$ denote the set of all valuations over \mathcal{Y} . We consider guard constraints of the form $\bigwedge_i v_i \bowtie_i f_i$, where $v_i \in \mathcal{X}$ is a clock, $\bowtie_i \in \{<, \leq, >, \geq\}$ and $f_i : \mathcal{V}(\mathcal{D}) \times \mathcal{V}(\Gamma) \rightarrow \mathbb{R}$ is a real-valued function over data variable and parameter valuations. Compared to the usual TAs, these constraints involve real-valued data and parameters whose dynamics are more complex. A variable valuation η and a parameter valuation γ satisfy the above constraint iff $\bigwedge_i \eta(v_i) \bowtie_i f_i(\eta_{\mathcal{D}}, \gamma)$ holds. We denote with $\mathcal{B}(V)$ the set of guard constraints over V . We extend the TA notion of “reset” to describe both the reset of the clocks and the update of data variables. The reset of a set of variables $V' \subseteq V$ is thus a function $r : V' \times \mathcal{V}(V) \times \mathcal{V}(\Gamma) \rightarrow \mathbb{R}$. Given valuations η and γ , η is updated by reset r to the valuation $\eta[r] = \{v \mapsto r(v, \eta, \gamma) \mid v \in V'\} \cup \{v \mapsto \eta(v) \mid v \notin V'\}$ that applies the reset r to the variables in V' and leaves the other variables unchanged. We denote with \mathcal{R} the set of reset functions. The valuation η after time $\delta \in \mathbb{R}_{\geq 0}$ has elapsed is denoted by $\eta + \delta$ and is such that $\eta + \delta(v) = \eta(v) + \delta$ if $v \in \mathcal{X}$ and $\eta + \delta(v) = \eta(v)$ otherwise. This implies that all clocks proceed at the same speed and data variables are not affected by the passage of time. However, when an edge is taken the reset function on data can perform an arbitrary computation over the current valuation of all clock variables and parameters, which in particular enables solving simple differential equations needed to model hybrid dynamics.

Definition 1 (Deterministic Timed I/O Automaton with Priority and Data). *A deterministic timed I/O automaton (TIOA) with priority and data $\mathcal{A} = (\mathcal{X}, \Gamma, \mathcal{D}, Q, q_0, \Sigma_{\text{in}}, \Sigma_{\text{out}}, \rightarrow)$ consists of:*

- *A finite set of clocks \mathcal{X} , data variables \mathcal{D} and parameters Γ .*

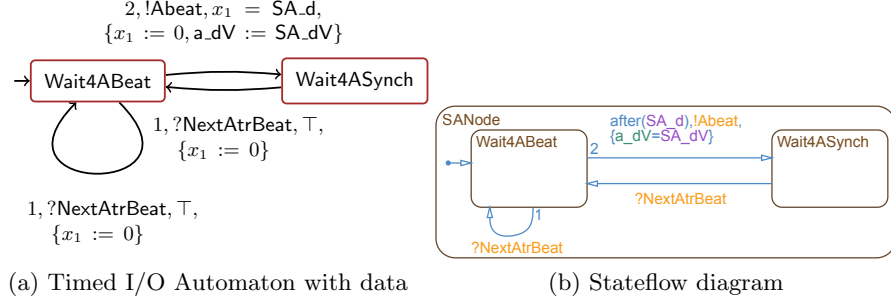


Fig. 1: Automata for the SA node

- A finite set of locations Q , with the initial location $q_0 \in Q$.
- A finite set of input actions Σ_{in} and a finite set of output actions Σ_{out} .
- A finite set of edges $\rightarrow \subseteq Q \times (\Sigma_{\text{in}} \cup \Sigma_{\text{out}}) \times \mathbb{N} \times \mathcal{B}(V) \times \mathcal{R} \times Q$. Each edge $e = (q, a, pr, g, r, q')$ is described by a source location q , an action a , a priority pr , a guard g , a reset r and a target q' .

We require that priorities define a total ordering of the edges out of any location, and that output actions have higher priority than input actions.

Fig. 1a depicts an automaton modelling the sino-atrial node. The automaton contains one clock, x_1 , one data variable, a_dV , and two parameters SA_d and SA_dV . The set of locations is $Q = \{\text{Wait4ABeat}, \text{Wait4ASynch}\}$, Wait4ABeat being the initial state. The action sets are $\Sigma_{\text{in}} = \{\text{?NextAtrBeat}\}$ and $\Sigma_{\text{out}} = \{\text{!Abeat}\}$. Digits 1, 2 labelling edges denote priorities.

The TIOAs as defined above are able to synchronise on matching input and output actions, thus forming *networks of communicating automata*. We say that an output edge is *enabled* when the associated guard holds. On the other hand, an input edge is enabled when both its guard holds and it can synchronise with a matching output action fired by another component of the network. A component of a network of TIOAs is enabled if, from its current location, there is at least one outgoing edge enabled. Also, we assume that output edges are *urgent*, meaning that they are taken as soon as they become enabled. As shown in [11], priority and urgency imply that *the TIOA is deterministic*.

Definition 2 (Network of TIOAs). A network of TIOAs with m components is a tuple $\mathcal{N} = (\{\mathcal{A}^1, \dots, \mathcal{A}^m\}, \mathcal{X}, \Gamma, \mathcal{D}, \Sigma_{\text{in}}, \Sigma_{\text{out}})$ of TIOAs, where

- for $j = 1, \dots, m$, $\mathcal{A}^j = (\mathcal{X}, \Gamma, \mathcal{D}, Q^j, q_0^j, \Sigma_{\text{in}}, \Sigma_{\text{out}}, \rightarrow^j)$ is a TIOA,
- $\mathcal{X}, \Gamma, \mathcal{D}, \Sigma_{\text{in}}, \Sigma_{\text{out}}$ are the common sets of clocks, parameters, data variables, input and output actions, respectively.

We define the set of network modes by $\mathbf{Q} = Q^1 \times \dots \times Q^m$, with initial mode $\mathbf{q}_0 = (q_0^1, \dots, q_0^m)$ and the initial variable valuation η_0 . A state of the network is a pair (\mathbf{q}, η) , where $\mathbf{q} \in \mathbf{Q}$ and $\eta \in \mathcal{V}(V)$.

The execution of a network of TIOAs $\mathcal{N}(\gamma)$ is described by a sequence $\rho = (\mathbf{q}_0, \eta_0) \xrightarrow{t_0} (\mathbf{q}_1, \eta_1) \xrightarrow{t_1} \dots$, where, for each i , $\rho[i] = (\mathbf{q}_i, \eta_i)$ is a state of the network and t_{i-1} is the time spent in that state. A step in the path occurs as soon as at least one component is enabled and evolves as follows. Each component can (i) have an output edge with maximum priority enabled, in which case the component fires the output edge and moves to the next location accordingly, or (ii) if no output edge is enabled then it fires the enabled input edge with maximum priority, thus synchronising with an output edge taken by another component, or (iii) if no input edge is enabled, it lets time pass.

Probabilistic resets. Let $Distr(S)$ be the set of all probability distribution functions over a space $S \subseteq \mathcal{V}(V)$. We implement probabilistic dynamics by enabling the reset of data and clock variables with values sampled from distributions in $Distr(S)$. In this case, the reset of a set of variables $V' \subseteq V$ becomes a function $r : V' \times \mathcal{V}(V) \times \mathcal{V}(\Gamma) \rightarrow Distr(S)$. Given valuations η and γ , r induces an updated valuation $\eta[r]$ such that, for any $s \in S$:

$$\begin{aligned} \Pr(\eta[r](v) = s) &= r(v, \eta, \gamma)(s), \text{ for } v \in V' \\ \Pr(\eta[r](v) = s) &= \delta_{\eta(v)}(s), \text{ for } v \notin V' \end{aligned}$$

In other words, the reset $r(v, \eta, \gamma)$ is now a probability distribution, which determines the probability that a variable $v \in V'$ is assigned a value $s \in S$ in the updated valuation $\eta[r]$. Note that deterministic updates can still be expressed by choosing a Dirac distribution for $r(v, \eta, \gamma)$. Furthermore, as timing delays can depend on the variables, they can be made probabilistic too.

The second equation states that, when v is not affected by the reset, its value stays the same with probability 1, since $\delta_{\eta(v)}$ denotes the Dirac distribution centered at $\eta(v)$, i.e. such that $\delta_{\eta(v)}(s)$ is 1 for $s = \eta(v)$ and 0 otherwise.

In Fig. 2 we extend the TIOA for the SA node in Fig. 1a with probabilistic resets. In this version, the waiting time in location `Wait4ABeat` is uniformly distributed in the interval $[a, b]$. This is implemented through an additional data variable `rand_delay`, which is updated with a randomly sampled value (assignment `rand_delay := $\mathcal{U}(a, b)$`) every time before entering location `Wait4ABeat`. For the sake of simplicity, we use the shortcut $x_1 := 0$ to denote the deterministic update $x_1 := \delta_0$.

Note that the probabilistic choices affect directly only updates of variables. When an edge (q_1, a, p, r, g, q_2) is taken, the distribution r is sampled and variables are updated accordingly. As timing delays depend on the variables, they indirectly become probabilistic.

Modelling of Stateflow charts. The formalism of TIOAs conveniently captures a subset of the Simulink/Stateflow modelling language. In particular, we do not consider state hierarchy, meaning that automata locations cannot be specialised as sub-components. We also exclude state actions, used in Stateflow to model updates that occur without firing a transition, e.g. continuous flows. TIOAs

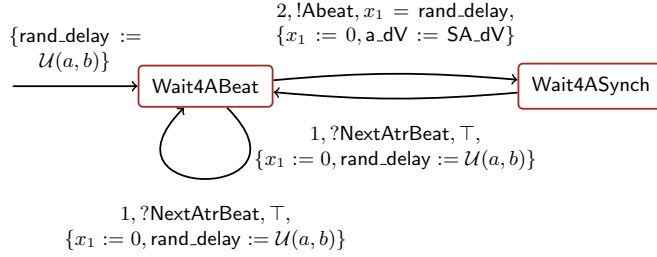


Fig. 2: Automaton in Fig. 1a extended with probabilistic resets.

employ broadcast communication, where synchronisation is not restricted, while Stateflow supports also *local events*, that is, two-party communication where the receiver is pre-determined. Nevertheless, local communication can be easily encoded as broadcast communication by means of appropriate action names. In addition, TIOAs do not support junctions and multiple actions associated to a single edge. Each Stateflow component possesses an implicit clock, which is reset to 0 whenever an edge is taken. In Stateflow charts, the keyword `after(t)` is equivalent to the time guard $x = t$, where x is the implicit clock.

Fig. 1 shows the same model both as a timed I/O automaton and a Stateflow chart.

3 Heart Model

The heart model describes the electrical conduction system of the heart and is used to reproduce the propagation of the cardiac action potential from the atria to the ventricles through the AV node. It is a TIOA translation of the model by Lian et al [22]. Model files are available⁵ in the Simulink/Stateflow file format for both the heart and the pacemaker.

In [22], the heart model is coded as a C program. In contrast, we provide a heart model as a network comprising hybrid automata and probabilistic timed automata represented in Stateflow, within a model-based plug-and-play framework that supports modular designs. Thus, for example, a novel pacemaker model can be added and composed in a closed loop with a heart model for pacemaker verification, or the heart model can be adapted to model a specific disease. The main advantage of a model-based framework is that it is rigorous, and therefore supports automated verification, e.g. checking safety of pacemakers [9] and parameter synthesis that minimises pacemaker energy consumption [4]. In addition, a range of additional analyses can be provided for the same model, including generation of executable or simulatable code.

The high-level structure of the model is depicted in Figure 3. It comprises 9 conduction nodes and two main conduction pathways: the *antegrade* conduction,

⁵ <http://qav.cs.ox.ac.uk/subm/HeartModel.zip>

that is, the normal situation where an electrical impulse generated by the sinoatrial (SA) node stimulates the atria and is conducted to the ventricles through the atrio-ventricular (AV) node; and the *retrograde* conduction, where the impulse travels in the opposite direction (from the ventricles to the atria through the AV node). Generally, retrograde conduction is less frequent and is originated when the ventricular myocardium is stimulated artificially by a pacemaker or by an ectopic action potential. The conduction between nodes is implemented through synchronisation between the involved components. In this way, the model can be easily extended with other accessory conduction pathways, in order to reproduce heart conditions such as the Wolff–Parkinson–White syndrome [6] which is characterised by an abnormal pathway between the atria and the ventricles that bypasses the AV node.

In the figure, we also illustrate the optional connection with a generic dual-chamber pacemaker component: the device sends impulses to the (right) atrium and (right) ventricle through actions AP and VP, respectively, and senses intrinsic impulses from the atrium and ventricle by synchronising on actions Vget and Aget, respectively. Model parameters are summarised in Table 2.

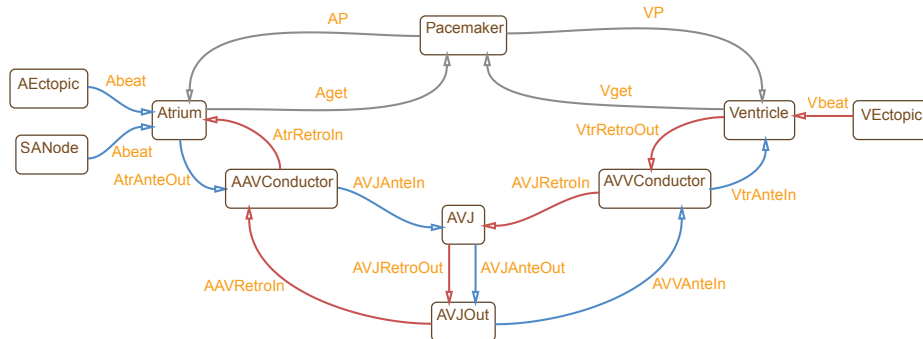


Fig. 3: TIOA components in the heart model. Arrows indicate synchronisations, with the target component synchronising with the output action performed by the source component, and are labelled with the action name. The antegrade conduction (blue arrows) passes from the Atrium component to the Ventricle component. The retrograde conduction (red arrows) follows the opposite path. Grey arrows indicate the connection with the pacemaker.

Below we provide a detailed description of the TIOA components of the heart model, expressed as Stateflow charts.

Atrial rhythm generation (Figure 4a). Impulses in the atria are generated by the components **SANode**, which models the sinoatrial node with a firing rate equal to **SA_d**, and **AECtopic**, which generates ectopic beats with rate **AEcto_d**. While the sinoatrial node is located in the right atrium, atrial ectopic beats (also called premature atrial contractions or PAC) are originated by the spontaneous excitation

of different portions of the atrial tissue, mainly from the pulmonary veins [24]. The potential is transmitted to the Atrium component through the output action !Abeat, when stepping from location *Wait4ABeat* to location *Wait4ASynch*. The strength of the action potential is stored in variable *a_dV* and is determined by parameter *SA_dV* for intrinsic impulses from the SANode, or *AEcto_dV* for ectopic ones.

Atrium (Figure 4b). This component provides an abstraction of the left and right atria and their stimulation dynamics. Initially, it stays in the *Refractory* location for *Atr_refrD* time units, before switching location to *Excitable*. In this mode, it can be stimulated by three types of impulses: intrinsic or ectopic from the SANode or the AEctopic components through action *Abeat*; paced impulses from the pacemaker through *AP*; or a retrograde signal from the ventricles through *AtrRetroIn*. These trigger a contraction of the atria. When the atrial stimulus is generated by the SA node or the pacemaker, this is propagated towards the ventricles through the AV node by performing action *AtrAnteOut*. When the stimulus is not artificial, i.e., it is either an intrinsic or ectopic beat, the component outputs an *Aget* action for the pacemaker to read. Before reaching the *Refractory* state again, an intermediate action is required (*NextAtrBeat*) to synchronise the SANode component.

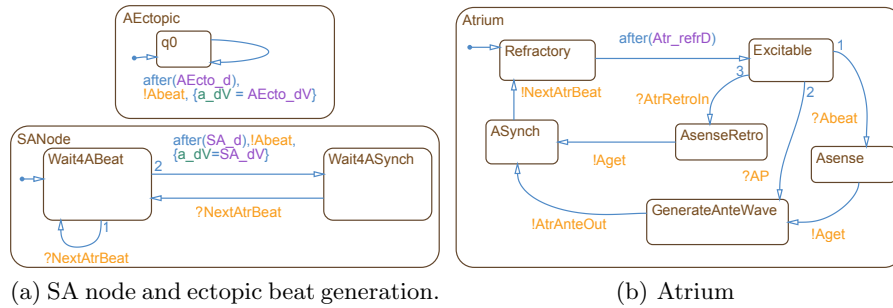


Fig. 4: Atrial rhythm generation components (a) and Atrium component (b).

Conductors (Figure 5). The AAVConductor (Atrium-AV node conductor) and RV conductor (AV node-Ventricle conductor) are structurally equivalent components that apply a propagation delay to the transmission of the action potential in both directions between the Atrium and the AV node (component AVJ), and between the AV node (component AVJOut) and the Ventricle, respectively. From the physiological viewpoint, the AAVConductor component reproduces the behaviour of the so-called internodal tracts, while the AVVConductor is an abstraction of the nodes connecting the AV node and the ventricles, namely, the bundle of His and the Purkinje fibres. When both antegrade and retrograde

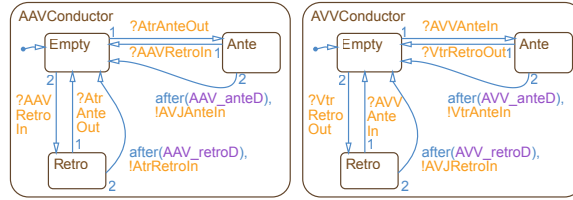


Fig. 5: AAVConductor (left) and AVVConductor (right) components.

waves reach one conductor, a so called fusion beat happens, i.e. they annihilate each other, bringing the conductor back to the *Empty* state.

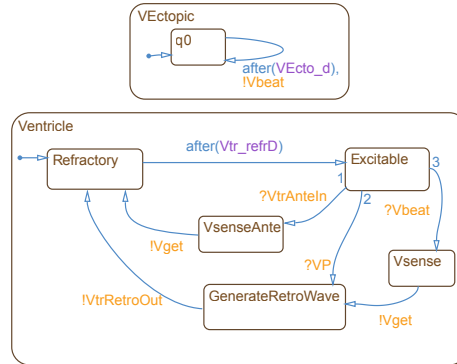


Fig. 6: Ventricular ectopic rhythm generation (top) and Ventricle component (bottom).

Ventricle and ventricular rhythm generation (Figure 6). The structure of the Ventricle component is similar to that of the Atrium component. The main difference is that we do not explicitly model the spontaneous depolarisation of the ventricles, which can occur with very slow rate in the absence of external electrical impulses. However, we include the VEctopic component for generating ectopic ventricular beats (also called premature ventricular contractions or PVC), which typically originate in the Purkinje cells or in the ventricular myocardium. Therefore, escape beats can still be reproduced by choosing an appropriate firing rate of ectopic beats (parameter VEcto_d) in the VEctopic component. In the ventricle automaton, the activation of the retrograde pathway is triggered from location *GenerateRetroWave* by firing the action *VtrRetroOut* after receiving a stimulus from the pacemaker (action *VP*) or an ectopic beat (action *Vbeat*).

AV node (Figure 7). The AV node plays a crucial role in the conduction of cardiac waves, and thus requires precise modelling of its action potential. It is

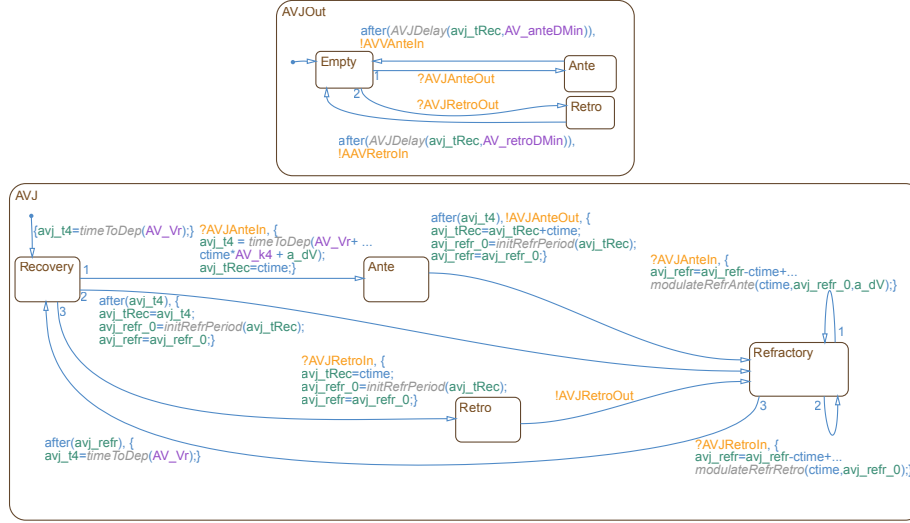


Fig. 7: Components modelling AV node.

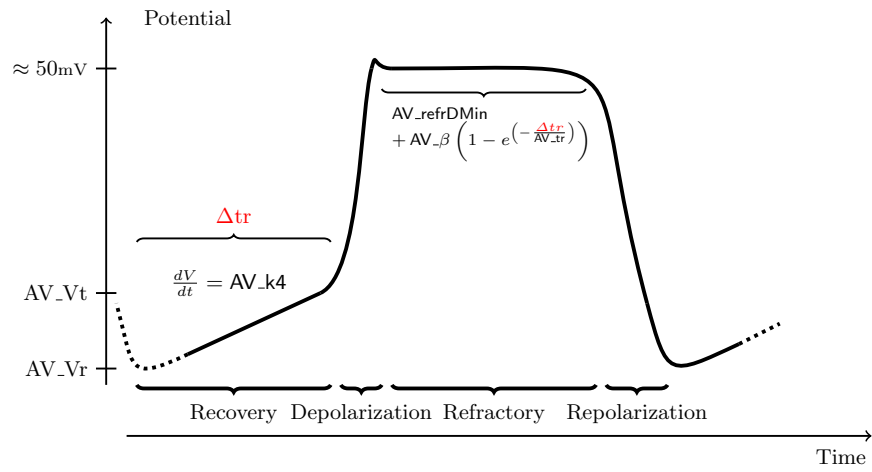
implemented through the components AVJ (atrio-ventricular junction), which models the action potential of cells in the AV node, and AVJOut, which applies additional delays depending on the state of the action potential. In the figure, `ctime` denotes the implicit automaton clock.

$$\begin{aligned}
 \text{timeToDep}(V_m) &= \max(0, \frac{AV_Vt - V_m}{AV_k4}) \\
 \text{initRefrPeriod}(t) &= AV_refrDMin + AV_beta \cdot (1 - \exp(-\frac{t}{AV_tr})) \\
 \text{modulateRefrAnte}(t, t_0) &= AV_refrDMin \cdot \left(\frac{t}{t_0}\right)^{AV_theta} \cdot \min\left(1, \frac{dV}{AV_Vt - AV_Vr}\right)^{AV_delta} \\
 \text{modulateRefrRetro}(t, t_0) &= AV_refrDMin \cdot \left(\frac{t}{t_0}\right)^{AV_theta} \\
 AVJDelay(t, t_{min}) &= t_{min} + AV_alpha \cdot \exp(-\frac{t}{AV_tau_c})
 \end{aligned}$$

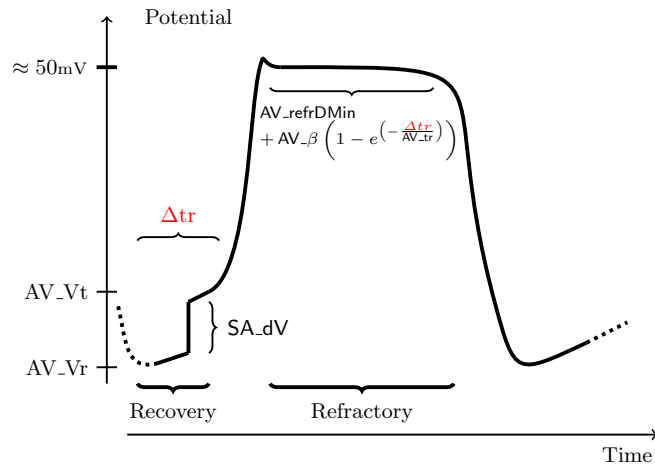
Table 1: Equations regulating the conduction delay and action potential in the AV node

In Table 1, we report the main functions used in the model to implement the AV conduction delay. Figure 8 illustrates the evolution of action potential in two possible cases: intrinsic depolarisation and excitation from atrial stimulus.

In the first case (Figure 8a), the AV node does not receive any external stimuli. In the *Recovery* state the action potential increases linearly starting from the resting potential `AV_Vr` with rate `AV_k4`. This slow depolarisation phase ends when the action potential reaches the threshold `AV_Vt`, after which we register a fast increase of the potential (rapid depolarisation). In the AVJ component, this



(a) Action potential with intrinsic depolarisation



(b) Action potential with stimulus from the atria

Fig. 8: Action potential of the AV node

behaviour is implemented by the transition from *Recovery* to *Refractory*. Given the current membrane potential V_m , function $\text{timeToDep}(V_m)$ computes the time left to reach rapid depolarisation during the linear increase phase.

In the refractory period, the action potential is roughly constant and does not affect the time spent in this state. Rather, this is affected by the time previously spent in the *Recovery* state (variable avj_tRec) through function $\text{initRefrPeriod}(\text{avj_tRec})$, according to which the initial refractory period (variable avj_refr_0) decays exponentially with avj_tRec .

Signals arriving at this stage prolong the refractory time according to functions modulateRefrAnte or modulateRefrRetro , depending of whether the signal originates from the Atrium or the Ventricle. This phenomenon is known as AV concealed conduction [26,17,23]. At the end of the refractory period, the action potential decreases to AV_Vr . This is modelled by the switching from the *Refractory* to the *Recovery* location.

The second case is the most common one, where a signal reaches the AVJ during the recovery period. Figure 8b shows an example of action potential receiving a stimulus from the Atrium. Depending on the intensity of the wave, the action potential increases promptly and is usually sufficient to trigger the depolarisation by exceeding the value AV_Vt . The atrial potential is stored in the a_dV variable, but in the figure we assume it comes from an intrinsic beat with potential SA_dV . Depending on the type of conduction, in the AVJ component the stimulus is detected through input action AVJAnteIn or AVJRetroIn , after which the component transitions into state *Ante* or *Retro*, respectively. In particular, we assume that signals from the Ventricle are always strong enough to trigger the depolarisation, while the impulse strength from the Atrium is used to update the time left to spend in the recovery period (stored in variable a_dV) through function initRefrPeriod . We also suppose that, during the recovery period, only one signal at a time can reach the AV node.

When it is depolarised, the AV node transmits the potential outside to the AVJOut component by performing output actions AVJAnteOut or AVJRetroOut . Through function $\text{AVJDelay}(\text{avj_tRec}, t_{min})$, the latter component applies an additional conduction delay, which decays exponentially with avj_tRec .

Fig. 9 illustrates some possible dynamics that can be reproduced by the model (with no pacemaker attached) in the form of stem plots. For these examples, timings and resets are deterministic. Fig. 9a shows a healthy behaviour, obtained with the default parameters, while Fig. 9b shows a bradycardia episode (slow heart rate) reproduced by increasing the SA node firing rate (parameter SA_d).

In Fig. 9c, we simulate a conduction defect of the AV node known as Wenckebach phenomenon [8]. This is characterized by the progressive prolongation of the atrio-ventricular conduction delay, until one wave fails to conduct to the ventricles. The AV delay is longest immediately before the dropped beat and shortest immediately after the dropped beat. In the figure, we can observe that the delay between atrial and ventricular events keeps increasing until the atrial event occurring at approximately 5 seconds is not conducted to the ventricles.

Parameter	Value	Description
SA_d	1000 ms	SA node firing rate
AEcto_d	10400 ms	Rate of atrial ectopic beats
SA_dV	50 mV	Voltage increment in AV node action potential due to atrial beat
AEcto_dV	50 mV	Voltage increment in AV node potential due to atrial ectopic beat
Atr_refrD	50 ms	Atrial refractory period
AAV_anteD	30 ms	Conduction delay from Atrium to AAVConductor
AAV_retroD	30 ms	Conduction delay from AAVConductor to Atrium
AVV_anteD	50 ms	Conduction delay from AVVConductor to Ventricle
AVV_retroD	50 ms	Conduction delay from Ventricle to AVVConductor
AV_Vr	-90 mV	AV node resting potential
AV_Vt	-40 mV	AV node depolarisation threshold potential
AV_k4	0.03 mV·ms ⁻¹	AV node spontaneous depolarisation slope
AV_refrDMin	50 ms	minimum AV node refractory period
AV_α	150 ms	maximum AV node delay time
AV_β	250 ms	maximum AV node refractory period
AV_tr	500 ms	AV node refractory time constant
AV_δ	10	electrotonic modulation constant for concealed impulse strength
AV_θ	10	electrotonic modulation constant for concealed impulse timing
AV_τ _c	100 ms	AV node conduction time constant
AV_anteDMin	50 ms	minimum AV node antegrade conduction time
AV_retroDMin	50 ms	minimum AV node retrograde conduction time
VEcto_d	30450 ms	Rate of ventricular ectopic beats
Vtr_refrD	200 ms	Ventricular refractory period

Table 2: Default parameters of the heart model (taken from [22]).

This behaviour is reproduced by increasing the AV threshold potential parameter `AV_Vt`. Indeed, when `AV_Vt` is too high, the potential from the atria is not sufficient by itself to trigger the depolarization. This causes an increased recovery time (i.e. waiting time in location *Ante*, see Fig. 7) and hence the refractory time is also increased according to the `initRefrPeriod` function. In turn, the increased time in the refractory state shortens the interval between the beginning of the next recovery period and the arrival of the next atrial stimulus, leading to a recovery time even longer than that of the previous cycle and to an increased conduction delays. The delays keep increasing until the atrial stimulus occurs during a refractory period and is thus dropped.

4 Estimation from ECG data

In this section, we describe the workflow and the methods to estimate from ECG data personalised parameters for the heart model introduced in Section 3. The workflow is depicted in Figure 10.

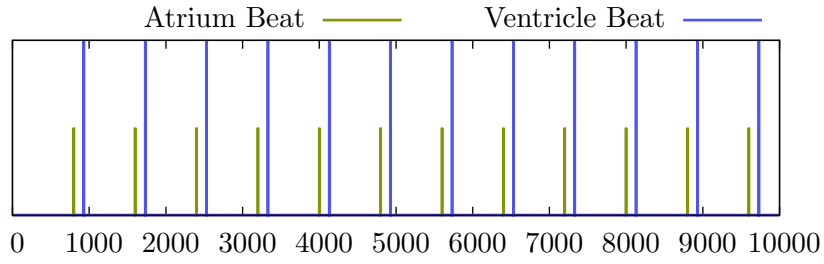
The first step is the filtering and analysis of the input ECG recordings and the detection of the characteristic ECG waves, which is described in Section 4.1. Information on the time intervals between ECG waves is directly mapped to a subset of the parameters, called *explicit*. However, there are also some *implicit* parameters that cannot be directly estimated from the ECG, e.g. those affecting the conduction delay within the AV node. For such parameters, we adopt Gaussian Process (GP) optimisation (described in Section 4.4) to find the values that maximise the probability that the input ECG is generated from the personalised heart model. In particular, the objective function is the statistical distance between the input ECG and the synthetic ECG, i.e. the ECG signal generated by the heart model. The derivation of the synthetic ECG and the statistical distance are described in Section 4.2 and 4.3, respectively.

4.1 ECG detection and mapping of explicit parameters

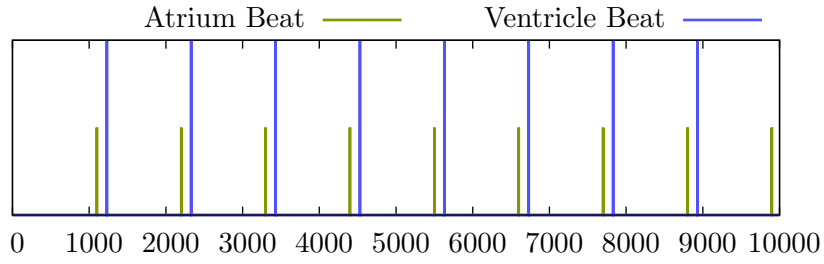
The electrocardiogram (ECG) is a signal acquired from the patient’s skin through electrodes and provides information on the electrical activity of the heart [1]. In Figure 11 we depict an example ECG for one cardiac cycle. It comprises five waves, P, Q, R, S and T, which map to specific heart events:

- *P wave*: activation of the atrium;
- *QRS complex*: activation of the ventricles; and
- *T wave*: ventricular repolarisation.

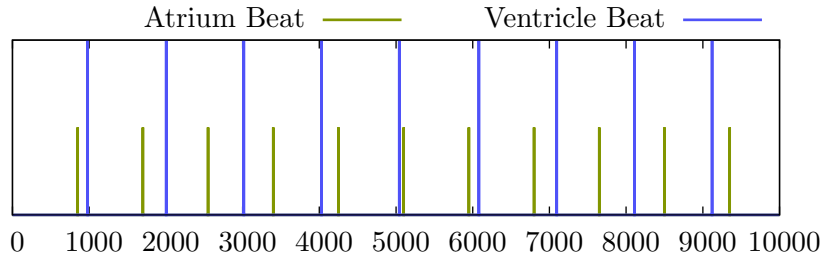
From ECG to model. The time interval between the P and Q waves, or *PQ interval*, represents the time needed for the impulse to travel from the atrium to the AV node. In the model, it is the time between an `Aget` event and the following `AVJAnteReached` event, which is determined by parameter `RA_anteD`. The distance between two consecutive R waves, called *RR interval*, gives the



(a) Healthy behaviour obtained with default parameters (Table 2).



(b) Bradycardia episode. Parameter $SA.d$ is set to 1100 ms.



(c) Wenckebach AV Block. Parameter $AV.Vt$ is set to -17 mV.

Fig. 9: Examples of reproducible heart dynamics. A short vertical stem (green) corresponds to an atrial event, whereas a longer one (blue) to a ventricular event. Time (x-axis) is in milliseconds.

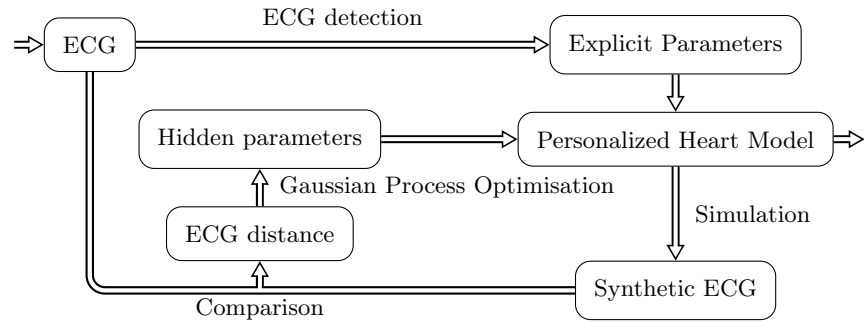


Fig. 10: Workflow for the estimation of personalised heart models from ECG data

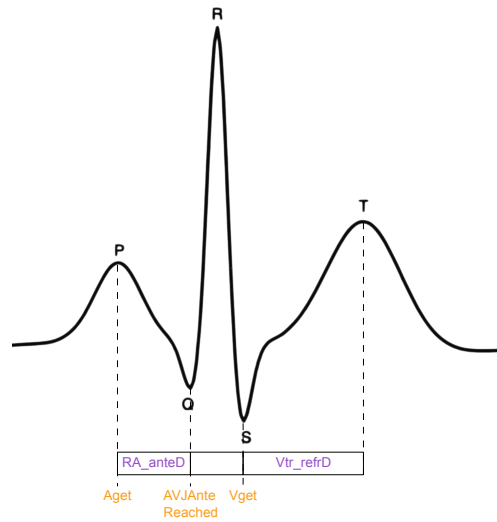


Fig. 11: Illustration of example ECG for one cardiac cycle and the corresponding P, Q, R, S, T peaks. On the bottom, we show the mappings with events (orange) and parameters (purple) of the heart model.

time between two ventricular impulses, i.e. the duration of the cardiac cycle. In general, this can be mapped to the SA node frequency (parameter `SA_d`). The *ST interval* describes the period of zero potential between ventricular depolarisation and repolarisation, during which the ventricular myocardial cells are refractory, that is, they cannot be stimulated by other impulses. Hence, this interval corresponds to parameter `Vtr_refrD`. Finally, the *QS interval* represents the time for the impulse to reach the ventricles from the AV node. In the model, this corresponds to the time between an `AVJAnteReached` event and a `Vget` event, which results from the dynamics of the components `AVJ`, `AVJOut` and `RVConductor`.

In Table 3 we summarise the mappings between ECG features and parameters or events of the heart model. Some of these mappings are explicit (PQ, RR and ST), i.e. when the corresponding ECG interval can be directly mapped to a parameter. Implicit mappings (QS) occur when the interval spans multiple events and is affected by multiple parameters.

Remark 1. Explicit mappings are implemented in the TIOA model in a *probabilistic* fashion. We first derive a discrete distribution from the detected intervals. Then, in the TIOA model, the corresponding parameter is replaced by a data variable that is updated with values uniformly sampled from the distribution. Some example distributions of PQ intervals are depicted in Fig. 19.

Interval	Explicit	Parameter(s)
RR	✓	<code>SA_d</code>
PQ	✓	<code>RA_anteD</code>
ST	✓	<code>Vtr_refrD</code>
QS		<code>AV_Vr</code> , <code>AV_Vt</code> , <code>AV_k4</code> , <code>AV_refrDMin</code> , <code>AV_β</code> , <code>AV_tr</code> , <code>AV_δ</code> , <code>AV_θ</code> , <code>AV_τc</code> , <code>AV_anteDMin</code> , <code>RV_anteD</code>

Table 3: Mapping of ECG intervals to heart model parameters (see Table 2 for the description of parameters).

ECG detection algorithm. The method for extracting peaks and intervals from the ECG signal consists of the following three steps:

1. *Pre-processing.* At this stage, the ECG signal is normalised in order to ensure a 0 mV baseline, and filtered to remove noise and baseline drift. Specifically, we employ a lowpass Butterworth filter.
2. *R peaks detection.* ECG waves are detected starting from the R wave, which indicates the presence of a heart beat. Here, R peaks are detected using the Pan-Tompkins algorithm [27].
3. *ECG detection.* The location and amplitudes of the other peaks and the duration of RR, PQ, ST and QS intervals are derived through an adapted

version of the ECG detection algorithm in [30]. For each wave, we also compute the *full width at half maximum*, i.e. the width of the wave at half of its amplitude, which will be used in the generation of the synthetic ECG, as explained in Section 4.2.

4.2 Generation of model-based synthetic ECGs

The derivation of the synthetic signal is inspired by [25] and obtained by summing, for each wave kind $i \in \{P, Q, R, S, T\}$ and for each wave location $l_i \in \text{Peaks}_i$ produced during the simulation of the model (see mapping with automata actions in Sect. 4.1), a Gaussian function centred at l_i , with height a_i and standard deviation c_i . For each peak, a_i and c_i are uniformly sampled from the sets of amplitudes and widths extracted in the ECG detection phase. In particular, c_i is related to the detected full width at half maximum (FWHM) by $c_i = \frac{\text{FWHM}}{2\sqrt{2\log 2}}$. Thus, the synthetic ECG at time t is given by:

$$\text{synthECG}(t) = \sum_{i \in \{P, Q, R, S, T\}} \sum_{l_i \in \text{Peaks}_i} a_i \cdot \exp\left(-\frac{(t - l_i)^2}{2c_i^2}\right).$$

Starting from the input raw signal, the process of ECG detection and generation of synthetic signal is illustrated in Figure 12.

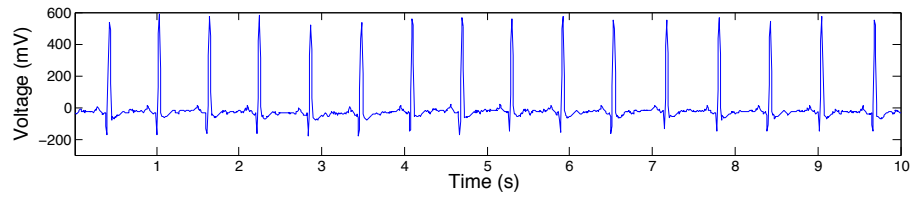
4.3 Measuring ECGs distance

In this section we illustrate how we compute the statistical distance between the filtered input ECG signal and the synthetic ECG.

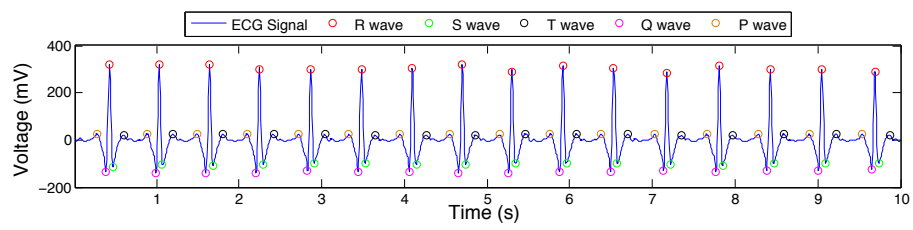
Phase assignment and statistical waveform. For each signal, we first need to derive a statistical description of the corresponding ECG waveform. To facilitate the comparison of the two signals, their statistical waveforms should be independent of the heart rate and represented so that the main peaks are synchronised between the signals.

To achieve this, we consider the *linear phase assignment* approach of [29] and implemented in [28]. It consists in assigning a periodic phase value to each sample in the ECG, starting from one R-peak (phase 0) and ending with the next R-peak (phase 2π). Then, the other samples between the two R-peaks are assigned linearly a phase value in $(0, 2\pi)$, as shown in Figure 13.

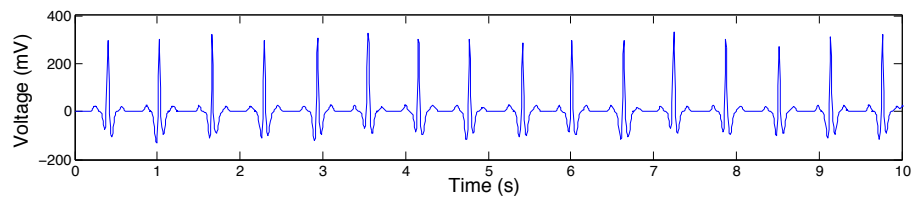
The resulting statistical waveform is a function $w : [0, 2\pi) \rightarrow 2^{\mathbb{R}}$ mapping phase values to the set of ECG samples registered at those values. In Figure 14, we depict the mean value and the standard deviation of an example statistical waveform, where we can clearly distinguish the characteristic ECG waves (with the R wave centred at phase 0).



(a) Raw signal



(b) Processed signal and detected peaks



(c) Synthetic signal

Fig. 12: ECG detection and generation of synthetic signal

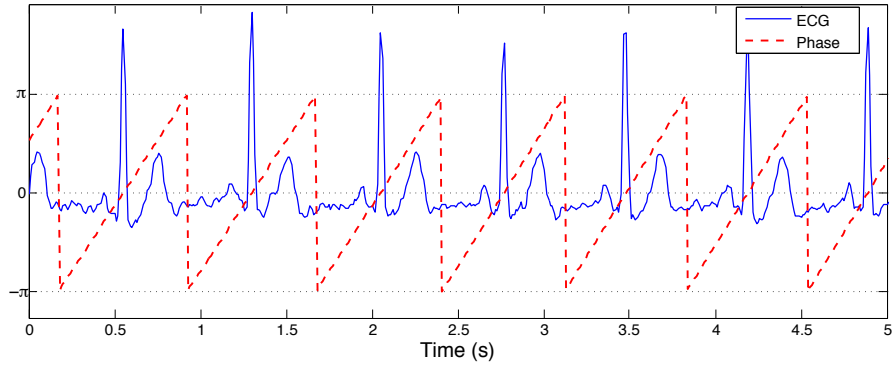


Fig. 13: Phase assignment for ECG signals.

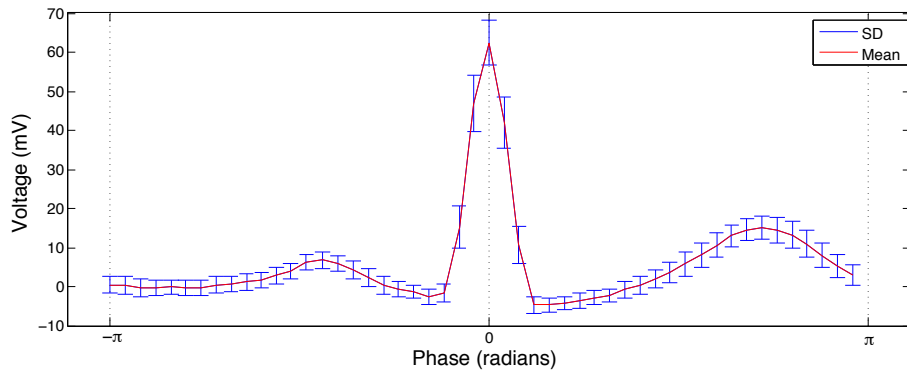


Fig. 14: Mean and standard deviation of statistical ECG waveform.

Statistical distance. We now want to provide a measure of similarity between two statistical ECG waveforms w_i and w_j . For this purpose, we define a statistical distance function $d(w_i, w_j)$ that returns a score from 0, when the waveforms are equivalent, to 1, when they are completely different.

We first divide the phase space of the waveforms into a finite number of right-open intervals and re-assign the samples accordingly. Specifically, if we assume k intervals, $[0, 2\pi)$ would be divided into $\{[0, \frac{2\pi}{k}), [\frac{2\pi}{k}, 2 \cdot \frac{2\pi}{k}), \dots, [(k-1) \cdot \frac{2\pi}{k}, 2\pi)\}$. Let $P = \{0, \frac{2\pi}{k}, \dots, (k-1) \cdot \frac{2\pi}{k}\}$. Then, a waveform w is transformed into a discretised waveform $w' : P \rightarrow \mathbb{N}^{\mathbb{R}}$ such that, for each $p' \in P$, $w'(p')$ is the multiset of samples recorded within the interval $[p', p' + \frac{2\pi}{k})$: $w'(p') = \bigsqcup_{p \in [p', p' + \frac{2\pi}{k})} w(p)$.

Let w'_i and w'_j be the discretised versions of w_i and w_j , respectively. These can be easily transformed into discrete probability distributions, which will be used as the inputs of the statistical distance function. For each interval p in the phase space, we fix a number of intervals k_p for dividing the sample space of $w'_i(p)$ and $w'_j(p)$. Let $m = \min\{w'_i(p), w'_j(p)\}$ and $M = \max\{w'_i(p), w'_j(p)\}$. For $x \in X = \{m, m + \frac{M-m}{k_p}, \dots, M\}$, we define the function $\mu_{i,p}(x)$ ($\mu_{j,p}(x)$) as

$$\mu_{i,p}(x) = \frac{|\{w'_i(p) \mid w'_i(p) \in [x, x + \frac{M-m}{k_p}]\}|}{|w'_i(p)|}.$$

It is easy to show that $\mu_{i,p}$ is a discrete probability distribution, since $\mu_{i,p}(x)$ is the relative frequency that the samples in $w'_i(p)$ take values within $[x, x + \frac{M-m}{k_p})$. The same derivation can be done for $\mu_{j,p}$.

In our experiments, to compare distributions $\mu_{i,p}$ and $\mu_{j,p}$ we employ the total variation distance [21], which is defined as:

$$d(\mu_{i,p}, \mu_{j,p}) = \frac{1}{2} \sum_{x \in X} |\mu_{i,p} - \mu_{j,p}|.$$

Finally, the distance $d(w_i, w_j)$ we seek is computed as the mean of the distances between distributions $\mu_{i,p}$ and $\mu_{j,p}$ for each point p in the discretised phase space:

$$d(w_i, w_j) = \frac{\sum_{p \in P} d(\mu_{i,p}, \mu_{j,p})}{|P|}$$

4.4 GP optimisation for synthesising heart parameters

Gaussian process optimisation is a black-box optimisation method, where samples are drawn by querying a statistical model of the unknown system, updated at run-time as new simulation results or measurements are obtained. It alternates between two main phases: compute an approximate solution x^* by optimising the statistical model; then perform actual measurements to obtain the objective function value at x^* , $f(x^*)$, and use it to improve *online* the accuracy of the model. In our case, the inputs of the statistical model are the implicit heart

parameters and the output to optimize is the statistical distance between the synthetic ECGs generated through the heart model and the training ECG signals. The optimisation algorithm is based on [16] and consists of the following steps:

- i) select n initial samples (by e.g. Latin hypercube) and compute their objective values;
- ii) estimate a statistical model from the current samples;
- iii) use the model to predict the point x^* that maximises the *expected improvement* (EI) and obtain $f(x^*)$;
- iv) add the latter to the set of samples and go to step ii).

The statistical model is built following the *Gaussian process regression* (GPR) method, which can be seen as a stochastic generalisation of linear regression. Given n samples $x^{(1)}, x^{(2)} \dots, x^{(n)}$, and their respective objective function values $f(x^{(1)}), f(x^{(2)}) \dots, f(x^{(n)})$, the method assumes that they are drawn from a model of the form:

$$f(x^{(i)}) = \mathbf{g}(x^{(i)})^T \cdot \boldsymbol{\beta} + \epsilon(x^{(i)}) \quad i = 1, 2, \dots, n \quad (1)$$

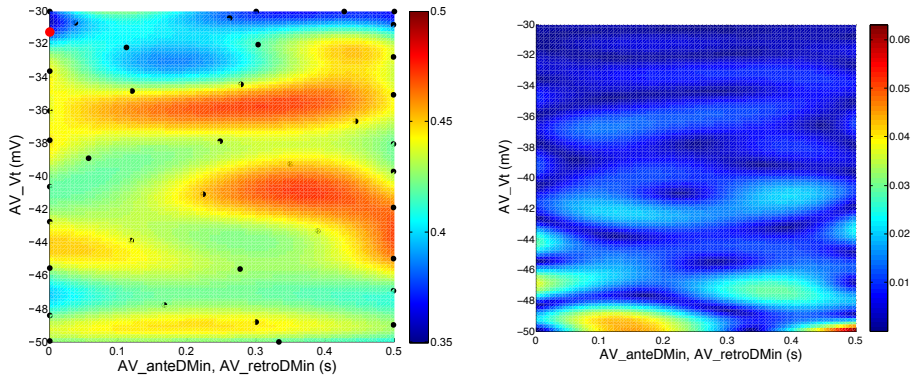
$\mathbf{g}(x^{(i)})^T \cdot \boldsymbol{\beta}$ is called the *regression part*, where $\mathbf{g}(x^{(i)})$ is the vector of basis functions and $\boldsymbol{\beta}$ is the vector of unknown coefficients estimated through classical regression techniques. ϵ is normally distributed with zero mean and correlation dependent on a weighted Euclidean distance of the n samples. Such weights are the parameters of the statistical model, and are estimated by maximising the likelihood function. For a point x^* , GPR is able to predict both an approximate value for $f(x^*)$, assuming it is randomly distributed according to Eq.(1), and an estimate of the prediction standard error. For our experiments, we implement GP optimisation by using the MATLAB SUMO toolbox [14], and employ training ECG from the MIT-BIH Normal Sinus Rhythm database (Id: 1, Table 4).

Id	Experiment	Database	Record Time (s)
1	Training	MIT-BIH Normal Sinus Rhythm	16265m [0,59]
2	Authentication	MIT-BIH Normal Sinus Rhythm	16265m [60,119]
3	Failed authentication	MIT-BIH Normal Sinus Rhythm	17453m [0,59]
4	Bradycardia verification	MIT-BIH Arrhythmia	202m [900,1020]

Table 4: Summary of ECG recordings extracted from [13] and used in the experiments.

Figure 15 shows the results of the estimation process through GP optimisation (30 samples). We consider three implicit heart parameters (see Table 3): the minimum times for antegrade and retrograde conduction in the AV node, *AV_anteDMin* and *AV_retroDMin*, respectively, and the AV node depolarisation

threshold AV_Vt . $AV_anteDMin$ and $AV_retroDMin$ are updated at each iteration with the same value, thus yielding a 2-dimensional parameter space for the optimisation problem. The search space is $[1, 500]$ ms for $AV_anteDMin$ and $AV_retroDMin$, and $[-50, -30]$ mV for AV_Vt . From plot a), we observe good distance values at low AV_Vt (approximately in the interval $[-30, -32]$ mV), while the distance does not seem affected by $AV_anteDMin$ and $AV_retroDMin$ in a consistent way. The standard deviation (plot b) is below 0.02 for a large majority of the parameter space, which indicates good accuracy of the statistical estimation. The accuracy can be further improved by increasing the number of samples.

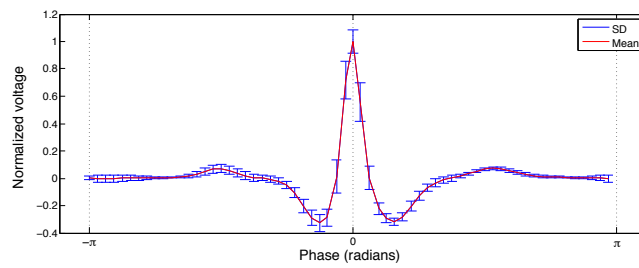


(a) Mean. Min ECGs statistical distance: 0.37 at $AV_anteDMin, AV_retroDMin = 1$ ms and $AV_Vt = -31.17$ mV.

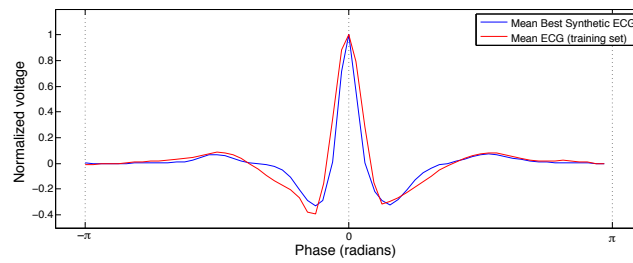
(b) Standard deviation

Fig. 15: Estimation of implicit parameters by GP optimisation of statistical distance between training and synthetic ECGs waveforms. Synthetic waves are generated by sampling parameters $AV_anteDMin$ and $AV_retroDMin$ (x-axis, updated with the same value) and AV_Vt (y-axis). a) The heat-map shows the mean values of the estimated Gaussian process (red: large distance values, blue: small distance values). Black dots indicate the simulated samples, the red dot the optimal sample. c) Standard deviation (blue: low, red: high).

In Figure 16a we report the mean and standard deviation of the synthetic ECG associated with the optimal parameters. The comparison with the mean ECG waveform of the training set (plot b) provides evidence that the estimated parameters can reproduce the patient’s electro-physiology well.



(a) Best synthetic ECG



(b) Training set ECG

Fig. 16: Mean and standard deviation of synthetic ECG (a) and comparison with training ECG (b).

5 Applications

5.1 Pacemaker Verification

In this section we describe the pacemaker model and give examples of properties that can be checked in combination with the heart model. The pacemaker has the role of maintaining the synchronisation between the atrium and the ventricle. In particular, we consider the DDD pacemaker specification [7], that is, pacing and sensing both the atrium and the ventricle, and provide a TIOA network model adapted from the timed automata model of Jiang et al [15].

We briefly describe the components of the pacemaker model shown in Fig. 17 (see [9] for details): AVI maintains the synchronisation between the atrium and the ventricle; LRI sets a lower bound for the heart rate; URI sets an upper bound for the heart rate; PVARP detects intrinsic atrial events; and VRP detects intrinsic ventricle events. The pacemaker communicates with the heart model by means of four actions: AP (atrial pace), VP (ventricle pace), Aget (atrial sense) and Vget (ventricle sense). Every component has associated a timing parameter. By changing these parameters one can control, for instance, the pacing rate in the atrium or ventricle, or the signal propagation delay from the atrium to the ventricle.

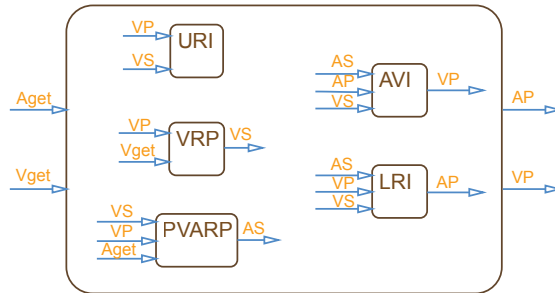


Fig. 17: Pacemaker model

Properties. We use the notation of temporal logic to express the properties of interest, see e.g. [9] for details on the syntax. The first property that we study characterises the probability that, within the first minute (60000 milliseconds), the time between any two successive ventricular contractions is always less than 1.1 seconds:

$$P_1 := P_{=?} G^{<60000} ((Vget \vee VP) \implies F^{<1100}(Vget \vee VP))$$

The proposition Vget holds when a Vget action is fired, i.e. when the ventricle contracts naturally and VP when the ventricle contracts due to pacemaker stimulation. Note that the subformula enclosed by the probabilistic operator $P_{=?}$

is falsified when the heart suffers bradycardia. We used our approach to build a personalised model from an ECG recording including a bradycardia episode (record 202 of MIT-BIH Arrhythmia Database on PhysioBank [13]).

Property	Healthy	Arrhythmia	With Pacemaker
P_1	0.99997 ± 0.0012	0.360607 ± 0.000015	$1 - 0.00003$
P_2	0.946454 ± 0.0005	$0.0 + 0.000005$	0.875494 ± 0.0008

Table 5: Results of pacemaker verification. For each property we report the satisfaction probability and the confidence interval considering healthy parameters (first column), arrhythmia parameters (second column, bradycardia for property P_1 and AV block for P_2) and arrhythmia parameters in combination with the pacemaker (third column). Properties are checked with the statistical model checker Cosmos [2] through 1 million simulations of the code generated from the model. Confidence intervals are built using the Clopper Pearson algorithm with a confidence level of 0.99.

The results reported in Table 5 provide evidence that bradycardia is fully corrected by the pacemaker since the verification of P_1 yields a satisfaction probability of 1. By detaching the pacemaker, we demonstrate that the probability of satisfying P_1 is much lower (nearly 0.36).

The second property asks for the probability that after, each atrial beat, a ventricular beat always occurs within [100, 200] ms:

$$P_2 := P_{=?} G^{<60000} \left((Aget \vee AP) \implies F^{[100,200]}(Vget \vee VP) \right)$$

In this case, we would expect a consistently low satisfaction probability in presence of an atrio-ventricular block (AVB) that delays the propagation from the atrium to the ventricle. Specifically, we simulate a first degree AVB, i.e. the fixed lengthening of the AV conduction time, through increasing both `AV_anteDmin` and `RV_anteD` by 60 ms. In the results table, we observe that the pacemaker does not make the property hold with probability 1, but the obtained probability is close to that of a healthy heart. This probability is lower than that of property P_1 due to instabilities such as atrial ectopic beats occurring during the refractory period of the AV node.

5.2 Authentication

Recently, biometrics based on the ECG and heart rhythm have been proposed for use in authentication, demonstrating a good level of accuracy (see e.g. [33,32]). Another advantage of this method is that the ECG signal cannot be captured without cooperation from the person, and currently it is difficult to falsify a person’s ECG. Fingerprint authentication, in comparison, can be compromised using high quality pictures [12].

A variety of methods exist to perform authentication based on ECG [32], with several of them based on feature extraction, e.g. delays between peaks or relative widths of peaks. Another class of methods is based on representing the ECG with simpler objects (wavelet decomposition, Fourier decomposition, decomposition into polynomials). Other techniques rely on the use of statistical distances similar to that presented in Sec. 4.3, or neural networks.

In general, the process consists of two phases:

1. *Enrolment*: the system captures and processes one or several ECG readings from the user and generates a biometric template.
2. *Authentication*: the system assesses the user’s identity by capturing her/his ECG, called recognition ECG, and matching it with the biometric template.

Our framework can be directly applied to authentication as follows. For the enrolment phase, we use synthetic ECGs generated by the *personalised* heart model as the biometric template. For the authentication, we compute its distance with the acquired recognition ECG. Or, from the recognition ECG we can build a personalised model and generate synthetic ECGs to match with the template. If the obtained score is small enough (e.g. not exceeding 50% of the score obtained in the parameter estimation phase), the authentication is successful.

Fig. 18a shows an example of successful authentication when the ECGs for model estimation and authentication come from the same patient⁶, while Fig. 18b shows how authentication fails with a recognition ECG from a different patient⁷. Alternatively, the extracted ECG features can be directly used for biometric authentication. As visible in Fig. 19, the template ECG and the recognition ECG from the same patient have associated almost identical distributions of PQ intervals. On the contrary, we observe a large discrepancy with the PQ intervals from a different patient.

Thus, our framework employs both feature extraction and statistical distance and, unlike other methods, does not rely solely on the PQRST waves. Indeed, our framework can also model abnormal rhythms, such as the presence of ectopic beats or retrograde conduction.

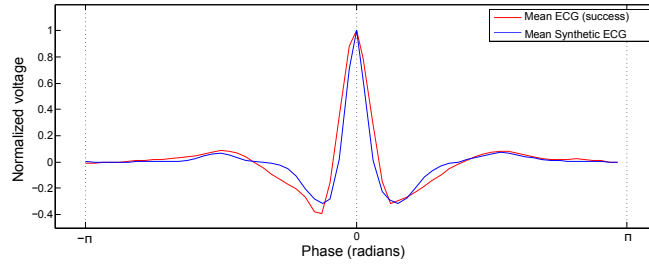
We remark that, by the construction of the statistical ECGs through the linear phase assignment approach, the distance between two ECG waveforms is not affected by the heart rate. This is necessary to avoid false negatives due to emotional or physical stress that increase the heart rate.

6 Conclusion

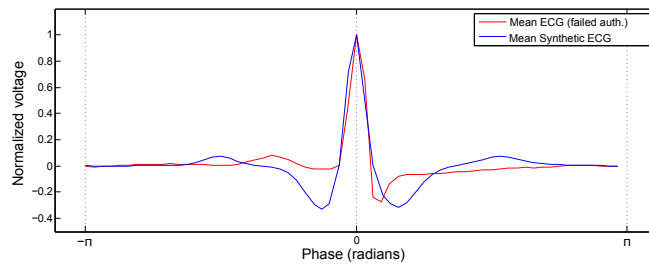
In this work, we presented methods to derive personalised heart models from data and showed their usefulness in the safety verification of pacemaker devices and in the ECG-based authentication. Besides enabling formal verification and

⁶ Dataset 2, Table 4

⁷ Dataset 3, Table 4



(a) Same patient (dataset 2, Table 4). Distance: 0.42.



(b) Different patient (dataset 3, Table 4). Distance: 0.78.

Fig. 18: Authentication tests by statistical distance between template ECG and recognition ECG from same (a) and different patients (b).

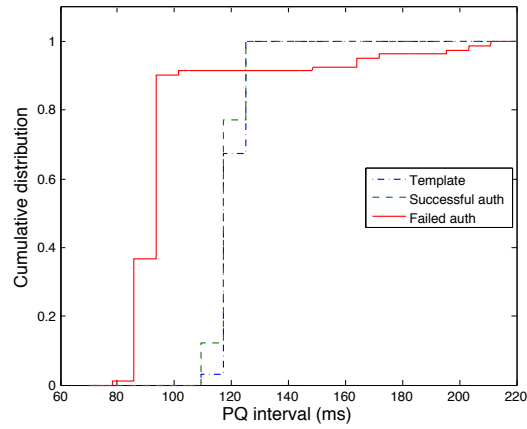


Fig. 19: Cumulative distributions of PQ intervals detected from the template ECG (blue line) and the recognition ECGs from same (green) and different (red) patients.

synthesis [10,5], code generation and modularity, our formal model-based framework is sufficiently general to support, at the same time, other kinds of physiological systems and medical devices. This would enable improvement of the authentication performance by combining the ECG with other biometrics (e.g. fingerprints or iris) [31], and ultimately verification of the collective behaviour of multiple interconnected devices in a closed-loop with a highly-personalised model of the human physiological system.

There are several directions for future work. Firstly, more accurate synthetic ECGs can be obtained by using Gaussian regression also to estimate some of the explicit parameters. For example, when waves are close to each other as in the QRS complex, their overlap makes the estimation of the amplitudes less accurate. In addition, the Gaussian functions used in the synthetic ECG could be replaced by non symmetric curves, given that most waves in real ECGs are not symmetric, see for instance the S waves in Fig. 13.

On a final note, ECG authentication is currently believed to be secure because it is difficult to obtain or copy someone's ECG recording. The same belief was held until recently for fingerprint and iris authentication, but in [12] it was shown that a picture taken with a smartphone is sufficient to reconstruct a fingerprint. As more and more devices record ECG data (smart watches, wearables), the risk of breaking the ECG authentication mechanism is growing, not least because wearables, which are not designed with security as a primary concern, may broadcast poorly secured ECG data via radio.

Acknowledgments. This work is supported by the ERC AdG VERIWARE and ERC PoC VERIPACE.

References

1. Ashley, E.A., Niebauer, J.: Cardiology explained, chap. Conquering the ECG. Remedica (2004)
2. Ballarini, P., Barbot, B., Duflot, M., Haddad, S., Pekergin, N.: Hasl: A new approach for performance evaluation and model checking from concepts to experimentation. *Performance Evaluation* 90(0), 53 – 77 (2015)
3. Barbot, B., Kwiatkowska, M., Mereacre, A., Paoletti, N.: Building power consumption models from executable timed I/O automata specifications. Submitted to *IEEE Real-Time Systems Symposium (RTSS 2015)* (2015)
4. Barker, C., Kwiatkowska, M., Mereacre, A., Paoletti, N., Patanè, A.: Hardware-in-the-loop simulation and energy optimization of cardiac pacemakers. In: *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, to appear (2015)
5. Barker, C. et al.: Hardware-in-the-loop simulation and energy optimization of cardiac pacemakers. In: *IEEE EMBC*, to appear (2015)
6. Bennett, D.H.: Wolff–parkinson–white syndrome. *Bennett's Cardiac Arrhythmias: Practical Notes on Interpretation and Treatment*, Eighth Edition pp. 67–76 (2012)
7. Boston Scientific: Pacemaker system specification. http://sqr1.mcmaster.ca/_SQRDocuments/PACEMAKER.pdf (2007)

8. Burns, E.: Av block: 2nd degree, mobitz i (wenckebach phenomenon). <http://lifeinthefastlane.com/ecg-library/basics/wenckebach/> (2015)
9. Chen, T., Diciolla, M., Kwiatkowska, M., Mereacre, A.: Quantitative verification of implantable cardiac pacemakers over hybrid heart models. *Information and Computation* 236, 87–101 (2014)
10. Chen, T. et al.: Quantitative verification of implantable cardiac pacemakers over hybrid heart models. *Information and Computation* 236, 87–101 (2014)
11. Diciolla, M., Kim, C.H.P., Kwiatkowska, M., Mereacre, A.: Synthesising optimal timing delays for timed I/O automata. In: 14th International Conference on Embedded Software (EMSOFT’14). ACM (2014)
12. Fiebig, T., Krissler, J., Hänsch, R.: Security impact of high resolution smartphone cameras. In: 8th USENIX Workshop on Offensive Technologies (WOOT 14). USENIX Association, San Diego, CA (Aug 2014)
13. Goldberger, A.L., et al: Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals. *Circulation* 101(23), e215–e220 (2000)
14. Gorissen, D., Couckuyt, I., Demeester, P., Dhaene, T., Crombecq, K.: A surrogate modeling and adaptive sampling toolbox for computer based design. *The Journal of Machine Learning Research* 11, 2051–2055 (2010)
15. Jiang, Z., Pajic, M., Moarref, S., Alur, R., Mangharam, R.: Modeling and verification of a dual chamber implantable pacemaker. In: *Tools and Algorithms for the Construction and Analysis of Systems*, pp. 188–203. Springer (2012)
16. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. *Journal of Global optimization* 13(4), 455–492 (1998)
17. Jørgensen, P., Schäfer, C., Guerra, P.G., Talajic, M., Nattel, S., Glass, L.: A mathematical model of human atrioventricular nodal function incorporating concealed conduction. *Bulletin of mathematical biology* 64(6), 1083–1099 (2002)
18. Kwiatkowska, M., Lea-Banks, H., Mereacre, A., Paoletti, N.: Formal modelling and validation of rate-adaptive pacemakers. In: *Healthcare Informatics (ICHI), 2014 IEEE International Conference on*. pp. 23–32. IEEE (2014)
19. Kwiatkowska, M., Mereacre, A., Paoletti, N.: On quantitative software quality assurance methodologies for cardiac pacemakers. In: *Proc. 6th International Symposium On Leveraging Applications of Formal Methods, Verification and Validation (ISoLA)*. LNCS, vol. 8803, pp. 365–384. Springer (2014)
20. Kwiatkowska, M., Mereacre, A., Paoletti, N., Patanè, A.: Synthesising robust and optimal parameters for cardiac pacemakers using symbolic and evolutionary computation techniques. In: *Proceedings of the 4th International Workshop on Hybrid Systems Biology (HSB 2015)*, to appear (2015)
21. Levin, D.A., Peres, Y., Wilmer, E.L.: *Markov chains and mixing times*. American Mathematical Soc. (2009)
22. Lian, J., Krätschmer, H., Müssig, D., Stotts, L.: Open source modeling of heart rhythm and cardiac pacing. *Open Pacing Electrophysiol Ther J* 3, 4 (2010)
23. Lian, J., Müssig, D., Lang, V.: Computer modeling of ventricular rhythm during atrial fibrillation and ventricular pacing. *Biomedical Engineering, IEEE Transactions on* 53(8), 1512–1520 (2006)
24. Manolis, A.S.: Supraventricular premature beats. In: *UpToDate* (<http://www.uptodate.com/contents/supraventricular-premature-beats>) (2015)
25. McSharry, P.E., Clifford, G.D., Tarassenko, L., Smith, L., et al.: A dynamical model for generating synthetic electrocardiogram signals. *Biomedical Engineering, IEEE Transactions on* 50(3), 289–294 (2003)

26. Meijler, F.L., Jalife, J., Beaumont, J., Vaidya, D.: Av nodal function during atrial fibrillation. *Journal of cardiovascular electrophysiology* 7(9), 843–861 (1996)
27. Pan, J., Tompkins, W.J.: A real-time qrs detection algorithm. *Biomedical Engineering, IEEE Transactions on* (3), 230–236 (1985)
28. Sameni, R.: Oset: The open-source electrophysiological toolbox. <http://spc.shirazu.ac.ir/products/Featured-Products/oset/>
29. Sameni, R., Shamsollahi, M.B., Jutten, C., Clifford, G.D.: A nonlinear Bayesian filtering framework for ECG denoising. *Biomedical Engineering, IEEE Transactions on* 54(12), 2172–2185 (2007)
30. Sedghamiz, H.: An online algorithm for R, S and T wave detection. <https://sites.google.com/a/student.liu.se/h-sedghamiz/projects-and-portofilo-matlab/portfolio/> (2013)
31. Singh, Y.N., Singh, S.K.: Evaluation of electrocardiogram for biometric authentication (2011)
32. Sufi, F., Khalil, I., Hu, J.: ECG-based authentication. In: *Handbook of Information and Communication Security*, pp. 309–331. Springer (2010)
33. Wang, Y., Agrafioti, F., Hatzinakos, D., Plataniotis, K.N.: Analysis of human electrocardiogram for biometric recognition. *EURASIP journal on Advances in Signal Processing* 2008, 19 (2008)